LNBIP 421

Adriano Augusto · Asif Gill ·
Selmin Nurcan · Iris Reinhartz-Berger ·
Rainer Schmidt · Jelena Zdravkovic (Eds.)

Enterprise, Business-Process and Information Systems Modeling

22nd International Conference, BPMDS 2021 and 26th International Conference, EMMSAD 2021 Held at CAiSE 2021, Melbourne, VIC, Australia, June 28–29, 2021 Proceedings

Springer

Lecture Notes in Business Information Processing 421

More information about this series at http://www.springer.com/series/7911

Adriano Augusto · Asif Gill ·
Selmin Nurcan · Iris Reinhartz-Berger ·
Rainer Schmidt · Jelena Zdravkovic (Eds.)

Enterprise, Business-Process and Information Systems Modeling

22nd International Conference, BPMDS 2021
and 26th International Conference, EMMSAD 2021
Held at CAiSE 2021, Melbourne, VIC, Australia, June 28–29, 2021
Proceedings

Springer

Editors
Adriano Augusto
University of Melbourne
Parkville, VIC, Australia

Asif Gill
University of Technology Sydney
Ultimo, NSW, Australia

Selmin Nurcan
University Paris 1 Panthéon-Sorbonne
Paris, France

Iris Reinhartz-Berger
University of Haifa
Haifa, Israel

Rainer Schmidt
Munich University of Applied Sciences
Munich, Germany

Jelena Zdravkovic
Stockholm University
Kista, Sweden

ISSN 1865-1348 ISSN 1865-1356 (electronic)
Lecture Notes in Business Information Processing
ISBN 978-3-030-79185-8 ISBN 978-3-030-79186-5 (eBook)
https://doi.org/10.1007/978-3-030-79186-5

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

Preface

This book contains the proceedings of two long-running events held along side the CAiSE conference relating to the areas of enterprise, business-process, and information systems modeling: the 22nd International Working Conference on Business Process Modeling, Development and Support (BPMDS 2021) and the 26th International Working Conference on Exploring Modeling Methods for Systems Analysis and Development (EMMSAD 2021).

The two working conferences had a joint keynote given by Didar Zowghi, Professor of Software Engineering at the Faculty of Engineering and IT at the University of Technology Sydney. The two events also shared an industrial talk given by Nigel Adams, a consultant researching process mining and business process compliance.

This year both conferences, originally planned to be held in Melbourne, Australia, during June 28–29, 2021, took a virtual form. More information on the individual events and their selection processes can be found on the following pages.

BPMDS 2021

BPMDS has been held as a series of workshops devoted to business process modeling, development, and support since 1998. During this period, business process analysis and design have been recognized as a central issue in the area of information systems (IS) engineering. The continued interest in these topics on behalf of the IS community is reflected by the success of the previous BPMDS events and the recent emergence of new conferences and workshops devoted to the theme. In 2011, BPMDS became a two-day working conference attached to the International Conference on Advanced Information Systems Engineering (CAiSE).

The goals, format, and history of BPMDS can be found on the website http://www.bpmds.org/.

The BPMDS working conference deals with and promotes research on business process modeling, development, and support, and has been a platform for a multitude of influential research papers. In keeping with its tradition, the working conference covers a broad range of theoretical and application-based research on BPMDS.

In 2021, BPMDS took place virtually as an online event, whilst keeping the general spirit and principles of BPMDS.

The intention of BPMDS is to solicit papers related to business process modeling, development, and support in general, using quality as the main selection criterion. As a working conference, we aim to attract papers describing mature research, but we still give place to industrial reports and visionary idea papers. To encourage new and emerging challenges and research directions in the area of business process modeling, development, and support, we have a unique focus theme every year. Papers submitted as *idea papers* must be relevant to the focus theme, thus providing a mass of new ideas around a relatively narrow but emerging research area. *Full research papers* and *experience reports* do not necessarily need to be directly connected to this theme (although they still needed to be explicitly relevant to BPMDS).

The focus theme for the BPMDS 2021 idea papers, *Business Process Improvement*, originates from the opportunities unleashed by the advancements in the fields of machine-learning and artificial intelligence, which enable the transition from the traditional process improvement led by human-experts to new process improvement methods based on intelligent software and systems, with the goal of reducing the effort and time required to achieve process improvements.

BPMDS 2021 received 26 submissions from authors in 12 countries (Australia, Austria, Belgium, France, Germany, Indonesia, Israel, Italy, Portugal, Spain, Sweden, Uruguay). The management of paper submission and reviews was supported by the EasyChair conference system. Each paper received at least three reviews from the members of the international Program Committee. Eventually, 10 high-quality full papers and 1 short paper were selected, which included one experience report.

The accepted papers cover a wide spectrum of issues related to business process development, modeling, and support, and also fit with this year's focus theme, *Business Process Improvement*, even though none of these papers were submitted as an idea paper. They are organized under the following section headings:

- Improving event data quality in coherence with business requirements
- Enhancing the value of data in processes improvement
- Event stream and predictive monitoring
- Modeling languages and reference models

We wish to thank all the people who submitted papers to BPMDS 2021 for having shared their work with us, as well as the members of the BPMDS 2021 Program Committee, who made a remarkable effort in reviewing submissions.

We also thank the organizers of CAiSE 2021 for their help with the organization of the event, particularly adjusting to the changing circumstances during the global COVID-19 crisis and facilitating the transformation to a virtual event. We would have liked to visit the lovely city of Melbourne. We also thank IFIP WG8.1 for its sustainable support and Springer, in particular Ralf Gerstner and Christine Reiss for their assistance during the production of the proceedings.

May 2021

Selmin Nurcan
Rainer Schmidt
Adriano Augusto

EMMSAD 2021

The objective of the EMMSAD conference series is to provide a forum for researchers and practitioners interested in modeling methods for Systems Analysis and Development (SA&D) to meet and exchange research ideas and results. The conference aims to provide home for a rich variety of modeling paradigms, including software modeling, business process modeling, enterprise modeling, capability modeling, service modeling, ontology modeling, and domain-specific modeling. These important modeling paradigms, and specific methods following them, continue to be enriched with extensions, refinements, and even new languages, to address new challenges. Even with some attempts at standardization, new modeling paradigms and methods are constantly being introduced, especially in order to deal with emerging trends and challenges. Ongoing changes significantly impact the way systems are analyzed and designed in practice. Moreover, they challenge the empirical and analytical evaluation of the modeling methods, which contributes to the knowledge and understanding of their strengths and weaknesses. This knowledge may guide researchers towards the development of the next generation of modeling methods and help practitioners to select the modeling methods most appropriate to their needs.

This year, EMMSAD 2021 continued its tradition and accepted papers in five tracks that emphasize the variety of EMMSAD topics: (1) Foundations of modeling and method engineering – chaired by Mahdi Fahmideh, Jolita Ralyté, and Janis Stirna; (2) Enterprise, business process, and capability modeling – chaired by Dominik Bork, Jānis Grabis, and Paul Grefen; (3) Information systems and requirements modeling – chaired by Aneesh Krishna, Roman Lukyanenko, and Marcela Ruiz; (4) Domain-specific and ontology modeling – chaired by Georg Grossmann, Dimitris Karagiannis, and Arnon Sturm; and (5) Evaluation of modeling approaches – chaired by Lubna Alam, Oscar Pastor, and Geert Poels. More details on the current and previous editions of EMMSAD can be found at http://www.emmsad.org/.

In total, 34 submissions were received from authors in 24 countries (Australia, Austria, Belgium, Canada, China, Colombia, Estonia, France, Germany, Greece, Israel, Italy, Japan, Latvia, Morocco, Netherlands, Norway, Pakistan, Qatar, Spain, Sweden, Switzerland, UK, USA). The division of submissions between the tracks was as follows: 4 submissions related to foundations of modeling and method engineering, 10 related to enterprise, business process, and capability modeling, 10 related to information systems and requirements modeling, 5 related to domain-specific and ontology modeling, and 5 related to evaluation of modeling approaches. After a rigorous review process, which included three reviews per submission, 14 high-quality papers, comprising 13 long papers and 1 short paper, were selected. The have been divided into four sections as follows:

1. Enterprise Modeling:
   - Anne Gutschmidt, Birger Lantow, Ben Helmanzik, Ben Ramforth, and Matteo Wiese. *Participatory Modeling From A Stakeholder Perspective: On the*

Influence of Collaboration and Revisions On Psychological Ownership and Perceived Model Quality.
- Daniela Pöhn and Peter Hillmann. *Reference Service Model for Federated Identity Management.*
- Wilco Engelsman, Roel Wieringa, Jaap Gordijn, Marten van Sinderen, and Timber Haaker. *Traceability from the Business Value Model to the Enterprise Architecture: A Case Study.*
- Hasan Koç, Kurt Sandkuhl, and Janis Stirna. Design Thinking and Enterprise Modeling: *An Investigation of Eight Enterprise Architecture Management Projects.*

2. Handling Models and Modeling Methods:
   - Victoria Döller and Dimitris Karagiannis. *Formalizing Conceptual Modeling Methods with MetaMorph.*
   - Sebastian Gottschalk, Enes Yigitbas, Alexander Nowosad, and Gregor Engels. *Situation-specific Business Model Development Methods for Mobile App Developers.*
   - Maxim Bragilovski, Yifat Makias, Moran Shamshila, Roni Stern, and Arnon Sturm. *Searching for Class Models.*
   - Wolfgang Maass, Roman Lukyanenko, and Veda C. Storey. *From Mental Models to Machine Learning Models via Conceptual Models (short paper).*
3. Threat and Evidence Modeling:
   - Nicklas Hersén, Simon Hacks, and Konrad Fögen. *Towards Measuring Test Coverage of Attack Simulations.*
   - Dirk van der Linden, Hava Dayan, Anna Zamansky, and Irit Hadar. *Murder, She Modeled: Modeling to Support Crimino-Forensic Processes.*
4. Model-Driven Engineering and Applications:
   - Charlotte Verbruggen and Monique Snoeck. *Model-Driven Engineering: a State of Affairs and Research Agenda.*
   - Flavio Corradini, Arianna Fedeli, Fabrizio Fornari, Andrea Polini, and Barbara Re. *FloWare: an Approach for IoT Support and Application Development.*
   - José Fabián Reyes Román, Alejandro Marco Palomares, Alberto García, and Oscar Pastor. *A Model-based Application for the Effective and Efficient Management of Data associated with Retina-Macula Pathology.*
   - Michiel Overeem, Slinger Jansen, and Max Mathijssen. *API Management Maturity of Low-Code Development Platforms.*

We wish to thank all the authors who shared their work with us, as well as the members of EMMSAD 2021 Program Committee for their valuable reviews in the difficult times of the COVID-19 pandemic. Special thanks go to the track chairs for their help in EMMSAD advertising and the review process. Finally, we thank the organizers of CAiSE 2021 for their help with the organization of the event, IFIP WG8.1 for its support, and Springer staff (especially Ralf Gerstner and Christine Reiss).

May 2021

Iris Reinhartz-Berger
Jelena Zdravkovic
Asif Gill

BPMDS 2021 Organization

Program Chairs

Adriano Augusto	University of Melbourne, Australia
Selmin Nurcan	University of Paris 1 Panthéon-Sorbonne, France
Rainer Schmidt	Munich University of Applied Sciences, Germany

Steering Committee

Ilia Bider	Stockholm University and IbisSoft, Sweden
Selmin Nurcan	University of Paris 1 Panthéon-Sorbonne, France
Rainer Schmidt	Munich University of Applied Sciences, Germany
Pnina Soffer	University of Haifa, Israel

Industrial Advisory Board

Ilia Bider	Stockholm University and IbisSoft, Sweden
Pascal Negros	Arch4IE, France
Gil Regev	EPFL and Itecor, Switzerland

Program Committee

João Paulo A. Almeida	Federal University of Espírito Santo, Brazil
Saïd Assar	Institut Mines-Télécom Business School, France
Judith Barrios Albornoz	University de Los Andes, Venezuela
Karsten Boehm	FH KufsteinTirol, Austria
Cristina Cabanillas	University of Seville, Spain
Claudio di Ciccio	Sapienza University of Rome, Italy
Dirk Fahland	Eindhoven University of Technology, the Netherlands
Renata Guizzardi	Federal University of Espírito Santo, Brazil
Jens Gulden	Utrecht University, the Netherlands
Amine Jalali	Stockholm University, Sweden
Anna Kalenkova	University of Melbourne, Australia
Marite Kirikova	Riga Technical University, Latvia
Agnes Koschmider	Kiel University, Germany
Sander J. J. Leemans	Queensland University of Technology, Australia
Henrik Leopold	Kühne Logistics University, Germany
Jan Mendling	Vienna University of Economics and Business, Austria
Michael Möhring	Munich University of Applied Sciences, Germany
Pascal Negros	Arch4IE, France
Oscar Pastor	Universitat Polytechnica de Valencia, Spain
Gil Regev	EPFL and Itecor, Switzerland

Manfred Reichert	University of Ulm, Germany
Hajo Reijers	Utrecht University, the Netherlands
Colette Rolland	University of Paris 1 Panthéon-Sorbonne, France
Michael Rosemann	Queensland University of Technology, Australia
Stefan Schönig	University of Bayreuth, Germany
Samira Si-Said Cherfi	CNAM, France
Pnina Soffer	University of Haifa, Israel
Irene Vanderfeesten	Eindhoven University, the Netherlands
Han van der Aa	Humboldt University of Berlin, Germany
Barbara Weber	Technical University of Denmark, Denmark
Moe Thandar Wynn	Queensland University of Technology, Australia
Jelena Zdravkovic	Stockholm University, Sweden
Alfred Zimmermann	Reutlingen University, Germany

Additional Reviewers

Adam Banham	Queensland University of Technology, Australia
Iris Beerepoot	Utrecht University, the Netherlands
Adam Burke	Queensland University of Technology, Australia
Jelmer Jan Koorn	Utrecht University, the Netherlands
Andrés Jiménez Ramírez	University of Seville, Spain

EMMSAD 2021 Organization

Program Chairs

Iris Reinhartz-Berger	University of Haifa, Israel
Jelena Zdravkovic	Stockholm University, Sweden
Asif Gill	University of Technology, Sydney

Track Chairs

Lubna Alam	Deakin University, Australia
Dominik Bork	TU Wien, Austria
Mahdi Fahmideh	University of Wollongong, Australia
Jānis Grabis	Riga Technical University, Latvia
Paul Grefen	Eindhoven University of Technology, the Netherlands
Georg Grossmann	University of South Australia, Australia
Dimitris Karagiannis	University of Vienna, Austria
Aneesh Krishna	Curtin University, Australia
Roman Lukyanenko	HEC Montreal, Canada
Oscar Pastor Lopez	Universitat Politècnica de València, Spain
Geert Poels	Ghent University, Belgium
Jolita Ralyté	University of Geneva, Switzerland
Marcela Ruiz	Zurich University of Applied Sciences, Switzerland
Janis Stirna	Stockholm University, Sweden
Arnon Sturm	Ben-Gurion University, Israel

Program Committee

Pär Ågerfalk	Uppsala University, Sweden
Sultana Lubna Alam	Deakin University, Australia
Raian Ali	Hamad Bin Khalifa University, Qatar
Said Assar	Institut Mines-Télécom Business School, France
Fatma Başak Aydemir	Boğaziçi University, Turkey
Giuseppe Berio	Université de Bretagne Sud and IRISA, France
Drazen Brdjanin	University of Banja Luka, Bosnia and Herzegovina
Tony Clark	Aston University, UK
Dolors Costal	Universitat Politècnica de Catalunya, Spain
Marne de Vries	University of Pretoria, South Africa
Rebecca Deneckere	Centre de Recherche en Informatique, France
Mehdi Fahmideh	University of New South Wales, Australia
Michael Fellmann	University of Rostock, Germany
Christophe Feltus	Luxembourg Institute of Sci. and Tech.
Peter Fettke	Saarland University, Germany

Hans-Georg Fill	University of Fribourg, Germany
Frederik Gailly	Ghent University, Belgium
Mohamad Gharib	University of Florence, Italy
Cesar Gonzalez-Perez	Incipit CSIC, Spain
Andreas Gregoriades	Cyprus University of Technology, Cyprus
Renata Guizzardi	Universidade Federal do Espirito Santo, Brazil
Jennifer Horkoff	Chalmers University of Technology, Sweden
Manfred Jeusfeld	University of Skövde, Sweden
Ivan Jureta	University of Namur, Belgium
Jānis Kampars	Riga Technical University, Latvia
Dimitris Karagiannis	University of Vienna, Austria
Elena Kornyshova	CNAM, France
Georgios Koutsopoulos	Stockholm University, Sweden
Thomas Kuehne	Victoria University of Wellington, New Zealand
Birger Lantow	University of Rostock, Germany
Tong Li	Beijing University of Technology, China
Patricia Martin-Rodilla	Spanish National Research Council, Spain
Raimundas Matulevicius	University of Tartu, Estonia
Haralambos Mouratidis	University of Brighton, UK
Alex Norta	Tallinn University of Technology, Estonia
Andreas L. Opdahl	University of Bergen, Norway
Jose Ignacio Panach Navarrete	Universitat de València, Spain
Vik Pant	University of Toronto, Canada
Francisca Pérez	Universidad San Jorge, Spain
Klaus Pohl	University of Duisburg-Essen, Germany
Fethi Rabhi	University of New South Wales, Australia
Pilar Rodríguez	Universidad Politécnica de Madrid, Spain
Ben Roelens	Open University of the Netherlands, the Netherlands
Kurt Sandkuhl	University of Rostock, Germany
Estefanía Serral	Katholieke Universiteit Leuven, Belgium
Monique Snoeck	Katholieke Universiteit Leuven, Belgium
Stefan Strecker	University of Hagen, Germany
Michael Verdonck	University Ghent, Belgium
Yves Wautelet	Katholieke Universiteit Leuven, Belgium
Hans Weigand	Tilburg University, the Netherlands
Carson Woo	University of British Columbia, Canada
Eric Yu	University of Toronto, Canada
Anna Zamansky	University of Haifa, Israel

Additional Reviewers

Achim Reiz	University of Rostock, Germany
Madhushi Bandara	University of New South Wales, Australia
Markus Fischer	University of Hagen, Germany
Simon Curty	University of Fribourg, Germany

Achim Reiz	University of Rostock, Germany
Johannes Wagner	University of Hagen, Germany
Mubashar Iqbal	University of Tartu, Estonia

Contents

Improving Event Data Quality in Coherence with Business Requirements (BPMDS 2021)

From Network Traffic Data to Business Activities: A Process Mining Driven Conceptualization 3
Gal Engelberg, Moshe Hadad, and Pnina Soffer

A NLP-Oriented Methodology to Enhance Event Log Quality 19
Belén Ramos-Gutiérrez, Ángel Jesús Varela-Vaca, F. Javier Ortega, María Teresa Gómez-López, and Moe Thandar Wynn

Enhancing the Value of Data in Processes Improvement (BPMDS 2021)

Detecting and Understanding Branching Frequency Changes in Process Models 39
Yang Lu, Qifan Chen, and Simon Poon

Detecting the "Split-Cases" Workaround in Event Logs 47
Yael Dubinsky and Pnina Soffer

Towards a Content-Based Process Mining Approach in Personal Services . . . 62
Felix Holz, Birger Lantow, and Michael Fellmann

Event Stream and Predictive Monitoring (BPMDS 2021)

Generating Reliable Process Event Streams and Time Series Data Based on Neural Networks 81
Tobias Herbert, Juergen Mangler, and Stefanie Rinderle-Ma

Inter-case Properties and Process Variant Considerations in Time Prediction: A Conceptual Framework. . . . 96
Avihai Grinvald, Pnina Soffer, and Osnat Mokryn

Towards Reliable Business Process Simulation: A Framework to Integrate ERP Systems. . . . 112
Gyunam Park and Wil M. P. van der Aalst

Modeling Languages and Reference Models (BPMDS 2021)

Supporting Data-Aware Processes with MERODE. . . . 131
Monique Snoeck, Johannes De Smedt, and Jochen De Weerdt

Evaluating Perceived Usefulness and Ease of Use of CMMN and DCR 147
Amin Jalali

Subject-Oriented Reference Model for Virtual Factory Operations Commissioning 163
Matthes Elstermann, Stefanie Betz, Matthias Lederer, Werner Schmidt, and Lisa Bührer

Enterprise Modeling (EMMSAD 2021)

Participatory Modeling from a Stakeholder Perspective: On the Influence of Collaboration and Revisions on Psychological Ownership and Perceived Model Quality 181
Anne Gutschmidt, Birger Lantow, Ben Hellmanzik, Ben Ramforth, Matteo Wiese, and Erko Martins

Reference Service Model for Federated Identity Management 196
Daniela Pöhn and Peter Hillmann

Traceability from the Business Value Model to the Enterprise Architecture: A Case Study 212
Wilco Engelsman, Jaap Gordijn, Timber Haaker, Marten van Sinderen, and Roel Wieringa

Design Thinking and Enterprise Modeling: An Investigation of Eight Enterprise Architecture Management Projects 228
Hasan Koç, Kurt Sandkuhl, and Janis Stirna

Handling Models and Modeling Methods (EMMSAD 2021)

Formalizing Conceptual Modeling Methods with MetaMorph 245
Victoria Döller and Dimitris Karagiannis

Situation-Specific Business Model Development Methods for Mobile App Developers 262
Sebastian Gottschalk, Enes Yigitbas, Alexander Nowosad, and Gregor Engels

Searching for Class Models 277
Maxim Bragilovski, Yifat Makias, Moran Shamshila, Roni Stern, and Arnon Sturm

From Mental Models to Machine Learning Models via Conceptual Models 293
Wolfgang Maass, Veda C. Storey, and Roman Lukyanenko

Threat and Evidence Modeling (EMMSAD 2021)

Towards Measuring Test Coverage of Attack Simulations 303
Nicklas Hersén, Simon Hacks, and Konrad Fögen

Murder, She Modeled: Modeling to Support Crimino-Forensic Processes 318
Dirk van der Linden, Hava Dayan, Anna Zamansky, and Irit Hadar

Model-Driven Engineering and Applications (EMMSAD 2021)

Model-Driven Engineering: A State of Affairs and Research Agenda. 335
Charlotte Verbruggen and Monique Snoeck

FloWare: An Approach for IoT Support and Application Development 350
Flavio Corradini, Arianna Fedeli, Fabrizio Fornari, Andrea Polini, and Barbara Re

A Model-Based Application for the Effective and Efficient Management of Data Associated with Retina-Macula Pathology. 366
José Fabián Reyes Román, Alejandro Marco Palomares, Alberto García Simón, and Oscar Pastor

API Management Maturity of Low-Code Development Platforms 380
Michiel Overeem, Slinger Jansen, and Max Mathijssen

Author Index 395

Improving Event Data Quality in Coherence with Business Requirements (BPMDS 2021)

From Network Traffic Data to Business Activities: A Process Mining Driven Conceptualization

Gal Engelberg[1,2](✉), Moshe Hadad[1,2](✉), and Pnina Soffer[1](✉)

[1] Department of Information Systems, University of Haifa, Haifa, Israel
spnina@is.haifa.ac.il
[2] Accenture Labs, Tel Aviv, Israel
{gal.engelberg,moshe.hadad}@accenture.com

Abstract. Event logs are the main source for business process mining techniques. However, they are produced by part of the systems and are not always available. Furthermore, logs that are created by a given information system may not span the full process, which may entail actions performed outside the system. We suggest that data generated by communication network traffic associated with the process can fill this gap, both in availability and in span. However, traffic data is technically oriented and noisy, and there is a huge conceptual gap between this data and business meaningful event logs. Addressing this gap, this work develops a conceptual model of traffic behavior in a business activity. To develop the model, we use simulated traffic data annotated by the originating activity and perform an iterative process of abstracting and filtering the data, along with application of process discovery. The results include distinct process models for each activity type and a generic higher-level model of traffic behavior in a business activity. Conformance checking used for evaluating the models shows high fitness and generalization across different organizational domains.

Keywords: Process discovery · Conceptual modeling · Network traffic

1 Introduction

Business process management (BPM) has been gaining growth in popularity in recent years due to its contribution to performance of organizations [1, 2]. BPM enables overseeing how work is performed, ensuring consistent outcomes and taking advantage of improvement opportunities [1]. Using process mining to discover the process model, one can perform tasks such as: identify the bottlenecks in a process, understand how to remove them, verify deviations from the process model, and more. The main data sources for process mining are event logs which are recorded data, produced by information systems, in the form of at least a timestamp, a case ID and an activity. Process mining algorithms use the case ID as a unique identifier for one specific business process execution (case) and the timestamp and activities to construct the business process model. Nevertheless, not all information systems are tuned to produce event logs of

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 3–18, 2021.
https://doi.org/10.1007/978-3-030-79186-5_1

this form. In many kinds of information systems, standard event logs are not always available, or available in different forms [3].

Consequently, many efforts have been devoted to exploiting data from a variety of sources, such as data bases, machine logs, Simple Object Access Protocol (SOAP) messages, and more [3]. For example, [4, 5] suggested techniques for extracting event logs from relational databases, [6] using database event data to discover a process model.

While extensive research considered various data sources, little attention was given to network traffic data. We claim it would be beneficial to use traffic data for process mining for the following reasons. First, while event logs may not always be available, network traffic can be obtained for processes using all kinds of information systems. Second, an event log may not provide full coverage of processes which may span more than one information system or include steps that are not managed by an information system (e.g., exchange of emails). Using network traffic data can provide additional aspects to help achieving a better coverage.

However, using such data for gaining business meaningful analysis is highly challenging. There is a huge gap in the abstraction level, and the gap does not end there. Traffic data is focused on technical network operations. It is partly generated by the business process, but it also contains many network activities that are not related to the business process, such as synching the network session, sending acknowledgment for receiving a data, or broadcasting a message to find the address of a specific computer. It is huge in its volume and comprised of atomically small building blocks of different protocols called packets. For some protocols, a message can be spread on several packets while for others a packet may contain several messages. According to our data, an average amount of 2,000 packets are generated for each business process activity. Furthermore, since in an enterprise network the traffic at any moment relates to many activities that are performed in parallel, attributing the packets and the messages to specific business activities is particularly challenging. One network stream may contain several business processes and activities that are running simultaneously. Since network traffic is a stream by nature, the start or the end of an activity instance is masked.

With these challenges, our aim is to show the gap between technical-level traffic data and conceptual-level business activities can be bridged. Since a business activity is a conceptual notion, our aim is to provide a conceptual model that depicts network behavior for a business activity in a generic manner. We pose the following research question: What are the behavioral patterns of network traffic that correspond to business activities and enable inference of business activity boundaries in network traffic terms?

To address this question, we take the following approach. We create simulated traffic data, annotated by the originating business activities. We then apply process discovery methods for identifying the traffic behavior that corresponds to a business activity. Due to the high granularity and noisy nature of the data, this is done iteratively, at an increasing level of abstraction, obtained by filtering and abstraction operations that are performed to the data. Eventually, distinct behavioral models are obtained for each activity in the simulated process. Moreover, we show that these models can further be abstracted into a generic conceptual model of traffic behavior in a business activity, which is then successfully evaluated against validation setting of a process spread across various organizational domains.

The remainder of the paper is structured as follows: Sect. 2 provides a background, then the main concepts of business processes and network traffic and their relations are presented, as a basis for the analysis process and its resulting behavioral patterns and conceptual model, which are then evaluated. Last, conclusions are drawn.

2 Background and Related Work

TCP/IP Communication Protocols. Hosts in a computer network communicate using network protocols. They send messages from host to host and the network protocols are responsible for transporting, encoding and decoding them. There are two widely used communication models: The ISO Open System Model (OSI) [7] and the Department of Defense (DoD) Internet Architecture Model [8]. For the purpose of this paper the DoD model was explored since it is commonly used, and the analysis of the DoD model can also be applicable for the OSI model. The DoD model has four layers: application layer, host to host layer, internet layer and the network access layer. Two layers are particularly interesting for the scope of this work. The application layer, which contains the high-level protocols in which the applications communicate with each other, and the host to host layer, which contains the transmission protocol in which the data is transported. Specifically, the Transmission Control Protocol/Internet Protocol (TCP/IP) is a fundamental network protocol vastly used in many computer networks. Each layer encapsulates the data of the previous layer, so by analyzing the TCP/IP protocol one can also get the data for the application layer.

Utilizing Low Level Events for Process Mining Tasks. An extensive research was done utilizing low-level events for process mining tasks, tackling a few challenges. One of the key challenges is to map between different abstraction levels. For that purpose [9, 10] presented supervised event abstraction methods to classify low-level events to their corresponding higher-level activities, for composing a higher level event log, while [11, 12] presented an unsupervised approach for this. Another challenge is identifying a case ID. [13] presented a semi-automatic method for discovering event's case and activity identifiers over an event log sourced from a distributed system. [14] presented an approach using communication protocol behavior for process mining, by recording a network traffic stream, converting it to an event log and discovering protocol behavior. In that case the communication session was utilized as a case identifier. However, the discovered process remained at a low level, showing network events without bringing them up to a business activity level.

One of the main use cases of utilizing low-level events for process mining tasks is analyzing software system logs to perform reverse engineering or discover low-level behavioral patterns that correspond to a higher-level behavior. For instance, in [15] a model driven reverse engineering technique was proposed to correlate and convert distributed system's events to a standard event log, which was mined to discover real life business transactions utilizing TCP/IP communication for the correlation. The work done in [16, 17] is another example for extracting a model from a software system. In [16] a sequence miner algorithm was presented, converting system logs into a sequence diagram. In [17] a sequence diagram was constructed from system's interactions by observing network traffic and analyzing TCP/IP packets. Messages were aligned using their sender and receiver and results were transformed into a sequence diagram by applying a k-tail algorithm.

While similar challenges are addressed in this paper, our aim is to bridge the conceptual gap between traffic data and business activities, and thus to establish a basis for utilizing this data for business-level process mining and analysis.

3 Conceptualizing Business Processes and Network Traffic

This section sets a basis for discovering behavioral patterns associated with a business activity by structuring concepts and relations of business processes and network traffic. We focus on a common communication infrastructure, which typically exists in enterprises. It includes communication between end-users and web-applications, utilizing a relational database backend, enterprise systems, and email exchange servers. Accordingly, the network has TCP/IP transportation, and application level client-server communication between web-applications, mail-servers, and database-servers over Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP) and Postgre Structured Query Language (PGSQL) application layer protocols, respectively. We note that additional scenarios and protocols exist. Yet, here we use the basic and most common ones as a basis. Figure 1 depicts a conceptual model of this environment, formed of two main parts: business abstract level and network traffic abstract level.

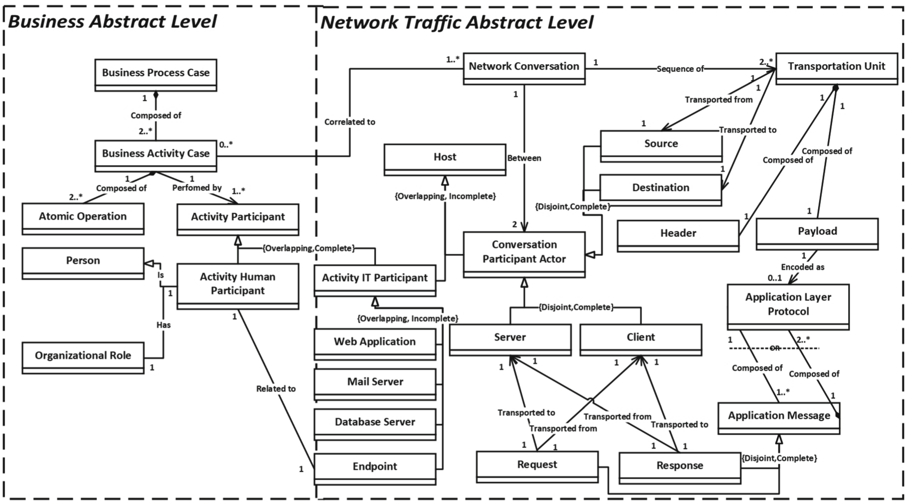


Fig. 1. Conceptual model of enterprise business and network traffic abstract levels

Business Abstract Level. A main entity in this level is a business process case which is an instantiation of a business process abstract class. A business process case is composed of two or more business activity cases, a business activity case is composed of at least two instances of atomic operation and is performed by at least one instance of activity participant. An activity participant is instantiated by an activity human participant and an activity IT participant, where an instantiation could be overlapping. An activity human participant is a person and has an organizational role.

An activity IT participant is a joint point of the business abstract level and the network traffic abstract level. It can be overlappingly instantiated by a web application, a mail server, a database server, or an endpoint, namely a single instance of an activity IT participant may be a database server and a mail server.

Network Traffic Abstract Level. The main entity in this part is network conversation, which is defined as a communication between two instances of a conversation participant actor. A network conversation is correlated to a business activity case. The relation between the two classes is defined as a correlation, since the first is comprised of atomically small building blocks of different protocols, transported in a concurrent manner. At any moment, those events could be related to many business activity cases, but this relation is not evident with certainty upfront. An instance of a business activity case can be correlated to at least one instance of a network conversation, while a network conversation instance can be correlated to multiple instances of a business activity case. Furthermore, a network conversation is described as a sequence of transportation units, defined as an information package (a packet) transported between a source and a destination which are disjoint instantiations of conversation participant actor.

Transportation unit is composed of a header that holds transportation unit's metadata such as timestamp, sequence number, etc. and a payload, which holds a data unit that is encoded as an application layer protocol. Moreover, a transportation unit has a transportation role which is instantiated to multiple instances such as a connection established request, a connection established acknowledge, a data transmission, etc., the instantiation could be overlapping.

An application message instance can be composed or decomposed by instantiations of a payload encoded as an application layer protocol, i.e., one transportation unit can hold many instances of application message, or act as a fragment that will be composed later on with other instances to a single application message. Furthermore, an application message is a disjoint instance of a request or a response, i.e., each instance of an application message can be a request transported from a client to a server, or a response transported from a server to a client. Moreover, each instance of a conversation participant actor in the context of a network conversation could acts as a server or a client.

4 Network Traffic Behavioral Patterns for an Activity

Towards answering the research question, simulated traffic data were generated as a foundation to discover the high-level behavioral patterns that correspond to a business activity. This section presents the behavioral analysis methodology and its results.

4.1 Simulation of Business Process Cases

As a first step, an enterprise simulation environment was created, to simulate execution of business process cases over an ERP web application with a relational database backend and an email server. The simulation environment included endpoint machines, where each was related to an activity human participant (e.g. HR manager). Each endpoint communicates with an Odoo ERP web application [18] over an HTTP application layer protocol, where the latter communicates with a PostgreSQL database server [19] and a mail server over a PGSQL and an SMTP application layer protocols, respectively (database server and mail server are located on same host).

Figure 2 depicts an HR (human resources) recruitment process that was executed via the simulation environment. A recruitment process starts with a generate job application activity (GJA) and holds two final states: application dropped (AD) or contract signed (CS). It includes the following activities: resume review (RR), schedule a phone interview (SPI), perform a phone interview (PPI), schedule an on-site interview (SOI), perform an on-site interview (POI), and contract proposal (CP). The HR recruitment process spans across a single organizational domain (HR).

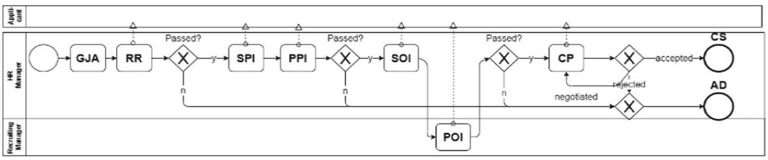


Fig. 2. HR recruitment process

Figure 3 depicts the purchase-to-pay process that was executed via the simulation environment. It starts with a create purchase request (CPR) and has two final states: purchase failed (PF) or purchase succeeded (PS). It includes the following activities: create call for tender (CFT), create RFQs (CRF), vendor response (VR), bid selection (BS), create purchase order (CPO), receive goods (RG), and submit payment (SP). The purchase-to-pay process spans across multiple organizational domains (sales, purchase, stock, accounting).

The following types of variance could be introduced in the simulation: (1) Time between two consecutive business activities of the same business process case. (2) Control-flow variance by randomization of decisions at decision points. (3) Activity human participant and its corresponding Activity IT participant are varied across activity's cases. (4) Data variance, e.g. in generate Job Application applicant details (name, email, and required job) are randomized across different cases. (5) Variance among the internal workflow of the business activity across different business process cases, e.g. negotiation mechanism applied in contract Proposal.

Three simulation scenarios were created: a training environment based on HR recruitment process (denoted as *TRAIN*) holds 250 cases, a validation environment based on purchase-to-pay process (denoted as *VALID*) holds 67 cases, and a combined test environment based on both business processes (denoted as *TEST*) holds 134 cases for HR recruitment, and 29 cases for purchase-to-pay. Network traffic data were recorded via

Wireshark[1] when executing activities to the form of a single packet capture (PCAP) file per business activity case. The set of PCAP files for each activity formed an initial event log for that activity[2]. Table 1 presents average amount of packets per business activity case in the recorded data.

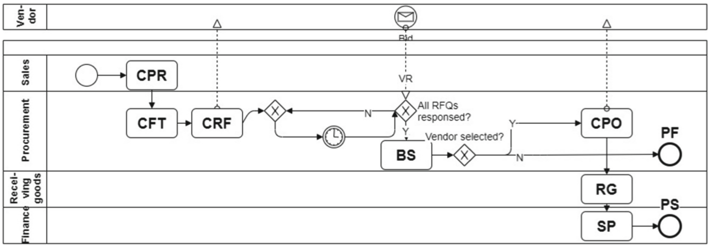


Fig. 3. Purchase-to-pay process

Table 1. Average packets per business activity case

Business process	Business activity	TCP Packets	PGSQL Packets	HTTP Packets	SMTP Packets	Total
HRR	GJA	111	235	10	-	355
	RR	206	597	11	20	834
	SPI	326	650	18	22	1,017
	PPI	183	435	14	5	638
	SOI	364	823	18	39	1,244
	POI	256	664	16	14	951
	CP	502	1,143	25	48	1,718
PTP	CPR	693	1,138	12	-	1,843
	CFT	850	696	12	-	1,557
	CRF	1,467	5,529	24	130	7,150
	VR	580	354	2	-	935
	BS	1,056	702	13	-	1,770
	CPO	852	2,996	12	46	3,906
	RG	709	1,141	16	-	1,866
	SP	856	2,678	18	-	3,552
Average		601	1,319	15	41	1,956

4.2 Packet Capture to Event Log

Transformation of packet capture data to a meaningful set of events is intended to conceptualize correlation between business and network traffic abstract levels. It is a challenging task due to the granular and technical nature of the latter. To overcome this issue, we took an iterative abstraction approach of transforming packet capture data to form event log representations of different abstraction levels, where the most suitable one served as a foundation for behavioral analysis by process discovery techniques.

This part was performed using *TRAIN* environment that yielded scenarios of 250 consecutive business process cases, denoted as $B_i = \{i \in [1..250]\}$, including cases of business activities, $j \in [1..7]$, which formed a PCAP file per each business activity case, denoted as A_{ij}. Furthermore, the network was recorded for 24 h without any execution

[1] www.wireshark.org.
[2] Available upon contacting gal.engelberg@accenture.com.

of business process cases to the form of a PCAP file. Prior knowledge, regarding IP addresses of hosts participating in business activities, included a mapping of a host to its related activity participants, denoted as *P* (e.g. an endpoint identified by IP is related to an HR manager, etc.).

Four representations of the event logs per business activity type were created, denoted as EA_R_{kj}; $k \in [1..4]$; $j \in [1..7]$. For assessing the convergence of the generated log to an abstracted pattern, we used a variant ratio, which is the ratio between the number of unique traces and the number of cases in the log. When all cases in an event log are different from one another, this ratio equals 1, and likely that process discovery would yield a spaghetti like process model; when only one unique trace exists, the ratio equals 1/#cases. Each representation *k* was measured by the average variant ratio across all business activities, denoted as V_k. In what follows we describe the abstraction steps aimed to minimize V_k.

Message Type Level Event Log. This step transformed packet captures to a message type level event log, which is the first abstraction level. This step takes multiple PCAP files per business activity, where each file holds packets related to a single business activity case, and creates a message type level event log per business activity, denoted as EA_R_{1j}; $j \in [1..7]$. Each PCAP file transformation holds two main steps: filtering a pcap file to reduce packets that are unrelated to a business activity execution and mapping the resulted packets to an event log standard form, as follows:

Filtering Steps: 1) Keep only TCP packets: Messages between all activity IT participants in the scope of this work are transported over a TCP transportation protocol. Hence packets of other transportation protocols (e.g., User Datagram Protocol (UDP)) are not related to the explored phenomenon. 2) Filter out packets related to irrelevant IP addresses: The enterprise simulation environment holds multiple hosts that communicate with each other routinely, some of which do not take part in business activities. Hence, at this step only packets that are related to activity IT participants (based on prior knowledge *P*) are kept in a PCAP file. 3) Filter out data retransmission packets: One key feature of a TCP protocol is a data retransmission mechanism, to assure reliability in cases where it is possible that a packet was not transmitted to its destination. In our setting, since network traffic was recorded by Wireshark, which listens to all network components, retransmission packets create duplicate messages. Hence, filtering out retransmission packets while keeping the original packets solves this issue. 4) Keep only TCP reassembled data: This filtering mechanism utilizes Wireshark capabilities of composing fragments of packet application layer data to a complete message. At this step only composed packets or packets that hold a complete message are kept.

Mapping Steps: 1) Classify message type: This step is applied to a filtered PCAP file and defines logic for classifying message types out of packets. A packet is classified according to its highest layer protocol, namely a packet can be described as a TCP packet for cases where no application layer data is transmitted between hosts. For example, a TCP connection acknowledgment packet is classified by the packet's transportation role, i.e. connection established acknowledge. Furthermore, for cases where application layer data is transmitted between hosts, a packet is classified according to an application layer request or response type. For example, in HTTP Request, the method followed by a

URI was used; in HTTP response the response code description was used; in PGSQL request the request type (e.g., a simple query) was used. 2) Extract message attributes: At this step all attributes related to a message type of an application layer protocol are extracted to the form of a set of (attribute, value) pairs. For instance, a PGSQL request simple query message type holds attributes for its query text, length, etc. 3) Map hosts to their related activity participants: This step uses prior knowledge P to map source and destination hosts of a packet to their related activity participants. This step is done for conceptualization proposes and can be replaced by using the IP of conversation participant actor if prior knowledge is not accessible. 4) Formulate event labels: This step sets the labels of events, as transportation source, destination, and message type. For example, connection synchronization between an HR manager and Odoo application is denoted as: *End Point (HR Manager)-> Odoo Application:[Connection establish request (SYN)]*. 5) Formulate timestamp attribute: Since time between network traffic events can have a delta of nano-seconds, which does not fit the time granularity of most process mining frameworks, there is a need to scale up timestamps as collected ones that would avoid unreal concurrency of events on one hand, and maintain correct relations of time between two consecutive events on the other hand. Therefore, timestamp is scaled up by 10^6 corresponding to the packet's relative transportation time since the beginning of traffic capture. 6) Formulate case ID attribute: All events related to a PCAP file A_{ij}, are marked by a corresponding case ID.

Calculation of variant ratio for this representation resulted with $V_1 = 0.66$.

Data Access Level Event Log. Since message type level event log held a fine-grained level of network traffic events of concurrent network conversations ($V_1 = 0.66$), conceptualizing behavioral patterns via process discovery could yield unusable results. Therefore, abstraction of the event logs to a data access level was performed. At this level only application layer events related to a user request, a mail message, and a database access via a query of SELECT, UPDATE, or INSERT were kept in event log. The intuition underlying this abstraction is that those events hold high connection to business semantics, unlike other events related to procedural network behavior such as TCP connection events, PGSQL authentication request, etc.

This abstraction takes EA_R_{1j} as an input and outputs EA_R_{2j}. The following message types were kept: HTTP request, HTTP response, SMTP reassembled message (holds a complete content of mail sent). For PGSQL, only messages of querying the database via SELECT, INSERT, or UPDATE were kept, including information regarding which tables were accessed, which was extracted from the message attributes and was concatenated to the event label. In addition, accessed tables were mapped to the event's resource attribute. Timestamp and case ID conventions were kept as in the previous representation. Calculation of variant ratio for this representation resulted with $V_2 = 0.29$, which is lower than V_1, but may still yield complicated models.

Data Access Level Event Log and Noise Cancelation. As mentioned, the network has been recorded for 24 h without any execution of business process cases. The recorded events, namely, messages that are transported when no business activity is performed, are related to recurrent database procedures. Those events were also observed during the execution of business activity cases, thus filtering them would keep-in events that are only related to a business activity case. This abstraction was done using EA_R_{2j} as an input and outputs EA_R_{3j}, where conventions regarding case ID, timestamp, event, and resource were kept. Calculation of variant ratio for this representation resulted in $V_3 = 0.14$.

Database Impact Level Event Log. Further abstraction was made to ease conceptualization, converting data access level event log with noise cancelation to database impact level event log, namely for HTTP and SMTP keep all events, and for a PGSQL filter out events whose query type is SELECT. The intuition behind this abstraction is that network traffic events that cause database impact (i.e. INSERT and UPDATE) would hold the 'bottom-line' of a business behavior. This abstraction is done using EA_R_{3j} as an input and outputs EA_R_{4j}, where conventions regarding case ID, timestamp, event, and resource were kept. Calculation of variant ratio for this representation resulted in $V_4 = 0.07$. With this variant ratio, this representation would serve as a base for the analysis of behavioral patterns, as described in the following.

4.3 Discovery of Network Traffic Behavioral Patterns for Business Activities

To infer business activity behavior and boundaries in terms of network traffic events, two analysis steps were performed using the database impact level event log of the *TRAIN* environment. The first step aimed to identify the main business entities in the process, namely, the central database tables that were accessed in the log. The second step was process discovery, to produce process models describing network behavior for each activity. Then a generic conceptual model was constructed utilizing the shared observed behaviors among the different business activities in the *TRAIN* environment.

Discovery and Definition of Main Business Entities. A business central table is defined as a table that is related directly to the ERP module that the business process is performed on, this information is common for ERP systems as ODOO[3], and SAP[4]. In HR recruitment all activities are performed via HR module. In purchase-to-pay activities span across sales, purchase, stock, and accounting modules. The rest of the tables are referred to as reference tables (users log table is referred as an audit trail across all business activities).

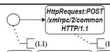
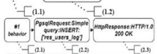
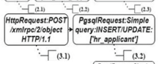
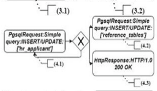
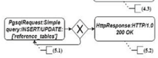
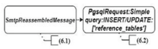
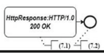
Process Discovery. At this step, a process model was discovered per each network traffic log of a business activity in the *TRAIN* environment (EA_R_{4j}) separately, using the Heuristic Miner [20]. Process model quality was measured in terms of fitness, precision, and F_1 [21]. While the obtained fitness was high (93% in average) and accordingly F_1 (80% in average) was acceptable, precision for most activity types was moderate (66% in average). This is understandable since the cases in our event logs were composed of long sequences (40 in average), containing multiple loops of different sizes. Yet, as a base for conceptualization we consider these results to be acceptable.

3 https://useopenerp.com/v8.
4 https://www.system-overload.org/sap/tables.html.

Conceptual Behavioral Model. Based on the discovered process models of the different activity types, we created a generalized conceptual model of traffic behavior in a business activity. Table 2 depicts all shared behaviors observed in the different process models (OBs), and their transformation to conceptual notions. For instance, a process model start was transformed to a business activity start, a database impact on an hr_applicant table was transformed to a database impact on a business central table, and so on. These conceptual notions were combined to a conceptual model describing a general behavior of network traffic when business activities take place. Figure 4 depicts the model as a BPMN model, composed of four pools. Activity human participant pool stands for the human actor executing atomic operations as part of a business activity, while the rest of the pools stand for activity IT participants.

As a general pattern we observe that a business activity starts with an HTTP request for user connection (OB1, OB2), followed by a sequence of business logic requests and responses, primarily inserting and updating the central table (OB3, OB4), and additionally inserting and updating other reference tables (OB5). Optionally, email messages can be triggered along this sequence (OB6). What marks the ending of an activity is an HTTP response that does not trigger further requests, transported from a web application to an endpoint (OB7).

Table 2. Observed behaviors

#	Observed behavior (OB)	Conceptual notion
1.		(1.1) Business activity starts >> (1.2) HTTP request: user connection.
2.		(2.1) OB1 >> (2.2) DB impact: insert user connection to audit trail>> (2.3) HTTP response.
3.		(3.1) HTTP request: business logic >> (3.2) DB impact: insert/update business central table.
4.		(4.1) DB impact: insert/update business' central table >> {(4.1) DB impact: insert/update business central table OR (4.2) DB impact: insert/update reference tables OR (4.3) HTTP response}.
5.		(5.1) DB impact: insert/update reference tables >> {(5.1) DB impact: insert/update reference tables OR (5.2) HTTP response}.
6.		(6.1) SMTP: send mail >> (6.2) DB impact: insert/update reference tables.
7.		(7.1) HTTP response >> (7.2) Business activity ends.

Business activity types differ from one another in the sequence of these operations, in the involved activity human participants, and in the tables and attributes that are accessed. Business activity cases differ from one another in the sequence of these operations for activities that hold internal workflow variance, and in values of attributes.

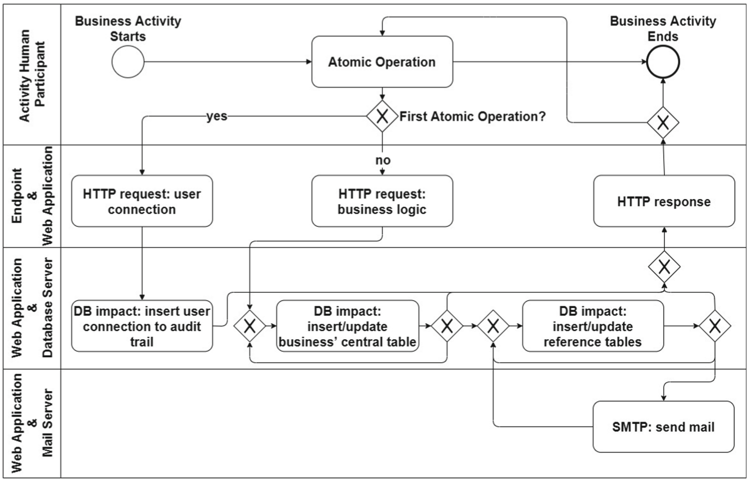


Fig. 4. Conceptual behavioral model of an activity

5 Evaluation

This section reports our evaluation, considering three questions as follows:

1) To what extent does the generic conceptual model fit the data of all activities, and generalizes well to other processes?

First, to measure model-data fitness, conformance checking of the conceptual model against an abstracted log of *TRAIN* environment was performed. The abstracted log included the logs of all the activity types, standing for all network traffic events. The case ID in this log marked a business activity case. Event labels were abstracted according to the OBs depicted in Table 2. Conformance checking shows that the model fits the log well (fitness = 91%). However, precision is moderate (57%) and lower than achieved for the process discovery of each activity type separately. This is as expected, since the model was created and evaluated against an event log containing events related to several business activities which differ from one another. Furthermore, abstraction reduced the amount of event classes, and resulted in increased recurrent transitions, which again reduced precision.

Second, to measure the models' generalization to other processes, conformance checking of the conceptual model against an abstracted log of the *VALID* environment was performed using the same approach as above. Results[5] shows that the model fits the log well (fitness = 86%), and precision is moderate (54.1%) similarly to the conformance checking over the *TRAIN* environment. Namely, the conceptual model is generalized well to another business process, which holds a higher level of complexity than the one used for training, with higher numbers of activity participant types, endpoints; and business domains (HR only vs. sales, purchase, stock, and accounting).

2) To what extent are the models obtained for each activity type distinct?

To address this question, the process models discovered for each activity of HR recruitment process (*TRAIN* environment) and Purchase-to-pay (*VALID* environment)

[5] Excluding vendor response activity an external communication activity which is out of model's scope.

were compared using a graph similarity method [22]. The method calculates Graph Edit Distance (denoted as GED), which measures distance between two causal nets as the minimal cost of transforming one into the other. Then a normalized similarity measure was calculated by dividing GED by the total number of constructs of the two causal nets. When the normalized similarity measure converges to 0, the causal nets are different; when it converges to 1 the causal nets are similar. Aggregated results described in Table 3 shows that models of different business activity types are distinct. Models of HR recruitment's activity types are moderately similar to each other, models of purchase-to-pay's activity types hold low similarity to each other, models of different business activity types across both processes hold remarkably low similarity.

3) To what extent could process models obtained for each activity be used to classify new cases?

To address this question, a conformance-based classifier was used to classify new business activity cases related both to HR recruitment and purchase-to-pay processes (*TEST* environment). The approach utilized the process models discovered in the previous step (based on the *TRAIN* and *VALID* environments). Each new case was conformance-checked against all process models, measuring its log-model fitness and precision, then the model holding the highest F_1 was determined as the predicted class. The results are evaluated via multi-class recall, precision and F_1 [23].

Conformance-based classifier results as specified in Table 4 yielded high recall (0.87) and precision (0.9) in average across all business activities. Log-model classification criteria shows that fitness is remarkably high in average for the predicted classes (0.98), log-model precision (0.49) and log-model F_1 (0.63) are moderate.

Table 3. Normalized similarity aggregated results

Models Comparison	Mean Normalized Similarity	STD Normalized Similarity
HRR vs. HRR	0.64	0.18
PTP vs. PTP	0.25	0.13
HRR vs. PTP	0.14	0.06

Table 4. Conformance-based classifier results

Business Activity	Recall	Precision	F_1
GJA	0.99	0.99	0.99
RR	0.04	0.86	0.09
SPI	0.99	1	0.99
PPI	1	0.34	0.51
SOI	0.98	1	0.99
POI	1	1	1
CP	0.95	0.95	0.95
CPR	1	0.91	0.95
CFT	0.93	1	0.96
CRF	1	0.97	0.98
BS	0.97	0.64	0.77
CPO	0.33	1	0.5
RG	1	1	1
SP	1	1	1
Average	0.87	0.90	0.83

6 Concluding Discussion

Network traffic data can be useful for process mining, but large conceptual gaps exist between this data and business meaningful event logs that are usable for process mining. Our aim in this paper was to bridge this conceptual gap and develop a generic conceptual model that depicts network behavior corresponding to a business activity.

The approach taken was to use simulated data for mining process models of traffic behavior that correspond to each activity type and generalize these models into a generic conceptual model. The results show that the conceptual model has a high fitness with the traffic data and that the model is generalized well to another business process which holds a higher level of complexity in terms of activity participant types, endpoints, and business domains involved. Furthermore, the models discovered for each activity type show distinct internal behavior that can be utilized for activity classification based on network traffic data.

The developed artefacts of this work, namely the generic conceptual model of an activity and a distinct behavioral model per activity type can be utilized in future research. For example, for identifying business activity occurrences and their boundaries within a network traffic stream and splitting network traffic stream to business activity case candidates towards a classification task. An additional contribution stems from the analysis process we took – the four-level iterative process of abstracting the data can generically serve for preparing traffic data for mining.

As an initial step in the direction of utilizing traffic data for process mining, the work is of a limited scope and suffers the following limitations. First, it is based on simulated data which might not hold all types of variations as in real life data. Second, the simulated environment targets a typical enterprise environment, and does not cover additional communication protocols and behaviors that may exist in more complicated environments. Last, as we recorded each activity separately in an artificial setting, we did not encounter the full complexity of an environment where many activities are performed in parallel. However, as mentioned above – the developed artefacts could be utilized to overcome this challenge later on.

Yet, our results show the possibility and pave a way towards transforming traffic data to an event log that lends itself to business meaningful process mining. Future research will address additional processes, environments, and protocols in more realistic settings. We intend to develop machine learning methods that will discover business activities in traffic data and will be able to distinguish traffic related to different processes and cases.

Acknowledgment. This work was done in collaboration with Accenture Labs, Israel, and supported by the Center for Cyber Law & Policy (CCLP), established by the University of Haifa in collaboration with the Israeli National Cyber Bureau.

References

1. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A. (eds): Fundamentals of Business Process Management, Springer, Berlin (2013)

2. Škrinjar, R., Indihar-Štemberger, M., Hernaus, T.: The impact of business process orientation on organizational performance. In: Proceedings of the 2007 Informing Science and IT Education Joint Conference (2007)
3. Van Der Aalst, W.: Process mining: discovery. Conformance and Enhancement of Business Processes. Springer, Berlin (2011). https://doi.org/10.1007/978-3-642-28108-2_19
4. Calvanese, D., Kalayci, T.E., Montali, M., Tinella, S.: Ontology-based data access for extracting event logs from legacy data: the onprom tool and methodology. In: Abramowicz, W. (eds.) Business Information Systems. BIS 2017. Lecture Notes in Business Information Processing, vol. 288, pp. 220–236. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59336-4_16
5. González López de Murillas, E., Reijers, H.A., van der Aalst, W.M.P.: Connecting databases with process mining: a meta model and toolset. Softw. Syst. Model. **18**, 1209–1247 (2019). https://doi.org/10.1007/s10270-018-0664-7
6. Van der Aalst, W.M.: Extracting event data from databases to unleash process mining. In: vom Brocke, J., Schmiedel, T. (eds.) BPM – Driving Innovation in a Digital World Management for Professionals, pp. 105–128. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-14430-6_8
7. Zimmermann, H.: OSI reference model - the ISO model of architecture for open systems interconnection. IEEE Trans. Commun. **28**, 425–432 (1980)
8. The DoD internet architecture model. Comput. Netw. (1976) **7**, 307–318 (1983)
9. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P., Toussaint, P.J.: From low-level events to activities - a pattern-based approach. In: La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNCS, vol. 9850, pp. 125–141. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45348-4_8
10. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.M.: Event abstraction for process mining using supervised learning techniques. In: Bi, Y., Kapoor, S., Bhatia, R. (eds.) IntelliSys 2016. LNNS, vol. 15, pp. 251–269. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-56994-9_18
11. Günther, C.W., Rozinat, A., van der Aalst, W.M.P.: Activity mining by global trace segmentation. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) Business Process Management Workshops, pp. 128–139. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12186-9_13
12. van Dongen, B.F., Adriansyah, A.: Process mining: fuzzy clustering and performance visualization. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 158–169. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12186-9_15
13. Bala, S., Mendling, J., Schimak, M., Queteschiner, P.: Case and activity identification for mining process models from middleware. In: Buchmann, R.A., Karagiannis, D., Kirikova, M. (eds.) PoEM 2018. LNBIP, vol. 335, pp. 86–102. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02302-7_6
14. Wakup, C., Desel, J.: Analyzing a TCP/IP-protocol with process mining techniques. In: Fournier, F., Mendling, J. (eds.) BPM 2014. LNBIP, vol. 202, pp. 353–364. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15895-2_30
15. Leemans, M., van der Aalst, W.M.: Process mining in software systems: discovering real-life business transactions and process models from distributed systems. In: 2015 ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS), pp. 44–53. IEEE (2015)
16. Wolny, S., Mazak, A., Wimmer, M.: Automatic reverse engineering of interaction models from system logs. In: 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 57–64 (2019)
17. Ackermann, C., Lindvall, M., Cleaveland, R.: Recovering views of inter-system interaction behaviors. In: 2009 16th Working Conference on Reverse Engineering, pp. 53–61 (2009)

18. Ganesh, A., Shanil, K.N., Sunitha, C., Midhundas, A.M.: OpenERP/Odoo-an open source concept to ERP solution. In: 2016 IEEE 6th International Conference on Advanced Computing (IACC), pp. 112–116. IEEE (2016)
19. PostgreSQL: The world's most advanced open source database. https://www.postgresql.org/ Accessed 28 June 2020
20. Mannhardt, F., de Leoni, M., Reijers, H.A.: Heuristic mining revamped: an interactive, data-aware, and conformance-aware miner. In: BPM (Demos) (2017)
21. Mannhardt, F., De. Leoni, M., Reijers, H.A.: The Multi-perspective process explorer. BPM (Demos). **1418**, 130–134 (2015)
22. Dijkman, R., Dumas, M., García-Bañuelos, L.: Graph matching algorithms for business process model similarity search. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. Lecture Notes in Computer Science, vol. 5701, pp. 48–63. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03848-8_5
23. Brownlee, J.: Imbalanced classification with Python: better metrics, balance skewed classes, cost-sensitive learning. Machine Learning Mastery (2020)

A NLP-Oriented Methodology to Enhance Event Log Quality

Belén Ramos-Gutiérrez[1(✉)], Ángel Jesús Varela-Vaca[1], F. Javier Ortega[1], María Teresa Gómez-López[1], and Moe Thandar Wynn[2]

[1] IDEA & ITALICA Research Groups, Universidad de Sevilla, Seville, Spain
{brgutierrez,ajvarela,javierortega,maytegomez}@us.es
[2] Queensland University of Technology (QUT), Brisbane, Australia
m.wynn@qut.edu.au
http://www.idea.us.es

Abstract. The quality of event logs is a crucial cornerstone for the feasibility of the application of later process mining techniques. The wide variety of data that can be included in an event log refer to information about the activity, such as what, who or where. In this paper, we focus on event logs that include textual information written in a natural language that contains exhaustive descriptions of activity executions. In this context, a pre-processing step is necessary since textual information is unstructured and it can contain inaccuracies that will provoke the impracticability of process mining techniques. For this reason, we propose a methodology that applies Natural Language Processing (NLP) to raw event log by relabelling activities. The approach let the customised description of the measurement and assessment of the event log quality depending on expert requirements. Additionally, it guides the selection of the most suitable NLP techniques for use depending on the event log. The methodology has been evaluated using a real-life event log that includes detailed textual descriptions to capture the management of incidents in the aircraft assembly process in aerospace manufacturing.

Keywords: Natural Language Processing · Event log quality · Process mining

1 Introduction

Event logs include the footprints generated by an organisation's information systems, being possible to store a wide variety of information [4,6] related to the tracked events, e.g., textual descriptions, timestamps or used resources. In general, event logs need to be adapted for a later (process mining) analysis, for instance, to discover processes. Thereby, the assessment of the quality of an event log [7] is the very first and crucial step for any subsequent analysis. The application of any process mining technique over incorrect or inaccurate event logs, e.g. process discovery, will produce incorrect or inaccurate process models [32].

Several authors have defined criteria to assess the data quality in general [9,21] and event log quality in particular [7,32], such as completeness, correctness, security and

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 19–35, 2021.
https://doi.org/10.1007/978-3-030-79186-5_2

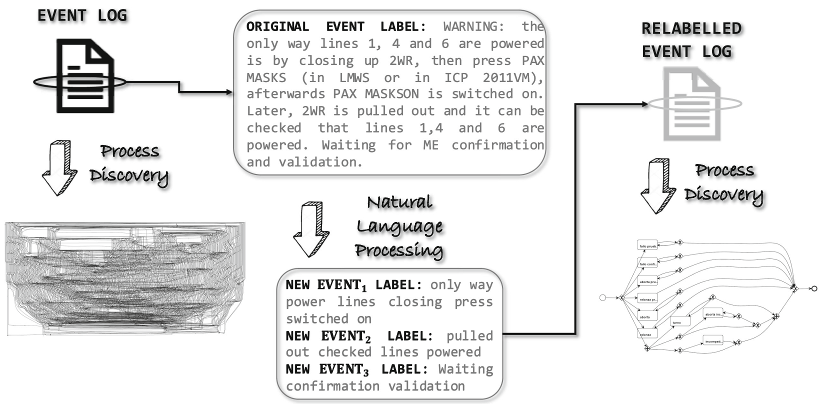


Fig. 1. Relabelling application example.

trustworthiness. The Process Mining manifesto [7] introduces the quality of an event log as a quality maturity level.

The imperfections than can produce a low event log quality might be improved analysing the activity labelling, timestamps, case identification, etc. In this paper, we focus on event logs that include some textual descriptions which detail what happened in various moments of process execution. We propose to first identify these activities and their inter-relations from textual constraint descriptions using NLP techniques and then relabelling activities in the event log to extract these details in an easy to handle way.

Figure 1 illustrates how the use of Natural Language Processing (NLP) techniques and the relabelling of activities in an event log, can improve the results of automated process discovery. For this reason, we not only adapt a general methodology to measure and assess the data quality of event logs [27], but also propose a decision-support system to assist in the selection of the most suitable NLP techniques according to the current quality level and the expected assessment. The research question is: *What are the most suitable NLP techniques to use for relabelling an event log in order to improve its quality?* This question does not have a simple answer, since it depends on the current quality of the event log and the dimension or dimensions that must be improved and how.

To answer this research question, it is necessary to define metrics to measure the event log quality and a mechanism to assess how good is this quality level in each context. This is well-known as the fitness-per-purpose [19], where the level of quality must be customised according to the needs, as for example: (1) determining the average length of the label of the activities; (2) the level of noise allowed; and (3) the usual number of activities per trace.

With this goal in mind, we propose a methodology, called LOADING-NLP, which assists in the decision-making for the application of NLP techniques over raw event logs for relabelling activities in accordance with the decision rules about data quality described by experts. A set of fitness-per-purpose metrics and dimensions are proposed to measure and assess the quality of an event log. Both, the metrics and dimensions can be customised, or extended for other examples. In this regard, the measurement and assessment can be adjusted and alternative NLP techniques can be applied. Our methodology also assists the user to select the most suitable NLP techniques. To validate the proposed methodology, we applied it to a real case study based on the management of the incidents produced during the aircraft assembly processes in aerospace manufacturing.

The rest of the paper is organised as follows: Sect. 2 includes the related work in the area. Section 3 introduces the methodology; the measurement and the assessment of the event log quality are discussed in Sect. 4. Section 5 reviews the NLP techniques that can be applied in this context and outlines to what extent they can affect the quality assessment. Section 6 presents the evaluation results and Sect. 7 concludes the paper.

2 Related Work

In order to understand the advantages that this proposal offers, it is necessary to know the level of maturity in the following areas.

Event Log Quality. The necessity to have data with suitable quality is crucial for any process and necessary for later analysis, such as process mining [10]. How to measure and assess the possibility of leveraging their quality is an important topic which has been a focus of study during the last decades. However, event logs appear in new contexts [30] and include features that make it necessary to define new metrics to measure, extend and adapt the dimensions (e.g., completeness, accuracy, simplicity) to the business process context [8,31].

Event Log Improvement. Once the data quality level can be assessed, various are the techniques that can be applied to improve the event log quality [26]. Some solutions are based on timestamp [12,13,15], case identification, and activity relabelling. Regarding the activity relabelling, the solution presented in [25] proposes to detect synonymous and polluted labels in event logs, but no techniques were proposed to improve the quality in accordance to the previous detection, and [24] uses a gamification approach to repair the labels. The types of improvements over event logs depend on the case and the later use of them [18].

Use of NLP in Business Process Management. Previous works have studied the extraction of declarative [2] and imperative business process models [3] from texts. In those works, the NLP techniques have been used in order to facilitate the automation of tasks that require a significant effort detecting patterns of relational order between the activities involved [1]. In addition, the detection of activities and their associated labels is crucial for further analysis and refactoring of the terms to enable an automatic analysis [17]. The text analysis in the context of the business process has also been focused on the detection of inconsistencies between the textual descriptions and the graphical representation [3], as a mechanism of misalignment detection.

Use of NLP to Improve Process Discovery Results. Some works have studied the preprocessing of the event data to improve the discovery task when using real-life logs that are written in natural language. In [23], is done by automatically detecting and classifying eight different semantic roles in event data. In [14], semantic-based techniques are applied to aggregate and normalise event log text information. Other types of analysis have been made to improve the labels of the activities in a process model by detecting erroneous ways of labelling activity that lead to ambiguity and inconsistency [22]. Contrary to our proposal, in all cases, these proposals start from event data in natural language with a very process-oriented construction and a simple and correct syntax.

To sum up, to the best of our knowledge, this is the first work where NLP techniques have been used to improve the event log quality according to a set of proposed metrics, guided by a decision-support system to ascertain the best techniques to apply depending on the event log and its quality level.

3 LOADING-NLP: Methodology for Assessing and Improving Event Log Quality with NLP

When the labels of the events within the log include natural language texts, it is necessary to analyse and treat them to become the log useful. The NLP techniques used to this aim will cause a direct impact on the different number of labels in the log, the number of events per trace, or the similarity between the labels.

Therefore, the best NLP techniques to use depends on the meaning of event log quality in each context, the event log quality before the application of the techniques, and how the experts want it to evolve after the application of the techniques. To support these three aspects, we propose the methodology presented in Fig. 2, described through a BPMN model.

The first step is related to the definition of when the event log is usable (cf., Determine the usability of the event log quality), according to the measurement and the assessment described by a set of decision rules about data quality. If the event log has sufficient quality, it can be used for a process mining analysis. However, if the event log is deemed unsuitable, the expert must determine the dimension or dimensions that must be adjusted and the required assessment (cf., Introduce dimensions and assessment to achieve). Using this information, we propose a decision-support system (cf., Infer NLP techniques to apply) that provides possible NLP techniques to use for improving the event log quality according to the described decision rules about data quality and the requirements of the experts. This process can be repeated until the resulting event log achieves the required level of quality.

4 Determine the Usability of the Event Log Quality

The question of when an event log has sufficient quality does not have a single answer. It will depend on the meaning of event log quality in each context. There are solutions as [27] that provide mechanisms to describe the decision rules related to the measurements and assessments adapted to each context and requirements. At first, we need to define

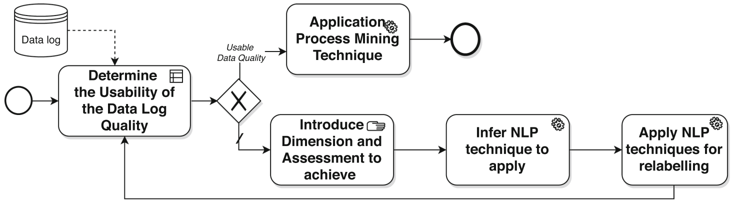


Fig. 2. Methodology for the application of NLP techniques.

the decision rules according to data quality using a set of metrics extracted from the event log. Thereby, it is necessary to define a set of metrics to evaluate the dimensions, as detailed in Subsects. 4.1 and 4.2. These measures enable us to perform an assessment according to the meaning of quality defined by the expert, as described in Subsect. 4.3.

4.1 Metrics to Measure the Quality of Event Logs

Based on [11], an ***event log*** is a set of traces that represent different instances of the same process. A ***trace*** is an ordered sequence of events that represents a process instance. Every trace is associated with a unique case identifier. The execution of an activity in a business process is represented as an ***event*** in an event log. Similarly, an event is the representation of the execution of an activity in a business process. Each event is associated with a case identifier, one timestamp and can also have many other contextual attributes. Usually, each event has associated at most one timestamp, which represents the start or the end of the execution of an activity.

To measure the dimensions, some metrics must be extracted from the event log [5, 11], as the mentioned in [8]. In our case, the used metrics are:

- **Number of traces.** Total number of traces in the log, and the trace j is represented by τ_j.
- **Number of events** (ε). Total number of events in the log, and the event i is represented by ε_i. It helps to know the size of the log.
- **Number of different labels.** Number of different labels that occur in every trace.
- **Number of unique labels.** Number of single (unique) labels that appear in the log.

4.2 Quality Dimensions for Event Logs

In general, the data quality dimensions describe the relevant aspects for a data set and typically consist of accuracy, completeness, consistency and uniqueness. However, we cannot guarantee that an event log with a high level of quality in those dimensions will produce valuable business processes. For this reason, other dimensions are included to assess the event log quality in process mining, as was defined in [31]. Those dimensions can be affected by the application of NLP techniques. Based on them, we propose the following dimensions albeit others can also be used together with our methodology:

$$m_{Uniqueness} = \left(\frac{Number\ of\ Unique\ Labels}{Number\ of\ Events}\right) \quad m_{Complexity} = \left(\frac{Avg.\ Number\ of\ Events}{Number\ of\ Traces}\right)$$

$$m_{Relevancy} = \left(\frac{Number\ of\ Different\ Labels}{Number\ of\ Events}\right) \quad m_{Consistency} = \sum_{i=1}^{\varepsilon} \left(\frac{|l(\varepsilon_i) - \overline{l(\varepsilon)}|}{Number\ of\ Events}\right)$$

Uniqueness. If every label in an event log is the same, the discovered process will only include one activity. However, if each label is unique in the traces, the discovered process will have one branch per trace, thus not very useful. The uniqueness dimension, that we propose in a range between $[0..1]$, measures the percentage of single labels regarding the total number of labels. When the values are in the extremes (i.e., 0 or 1), it implies that the process may be too simple (low uniqueness) or too complex (high uniqueness).

Consistency. When the labels of activities are dissimilar, especially for textual formats, they can imply that the descriptions have different granularity. For this reason, the measurement of consistency that we propose is based on the average length[1] of strings in these textual descriptions, and the mean of the distance to the average. Therefore, this dimension is bounded by the length of the longest string.

Relevancy. The relevancy of each label depends on the number of times that it occurs. It is important to analyse the number of different labels according to the total number. It is related to the uniqueness, but it is not exactly the same. The dimension is bound between $[0..1]$.

Complexity. There are several metrics that can represent the complexity of an event log [26], such as the average of events per trace. We propose to measure the complexity by the mean of the number of events per trace. A higher mean implies a higher concentration of events per trace, therefore representing a more complex process.

4.3 Customising the Measurement and Assessment of Event Log Quality

As commented previously, the data quality is an aspect highly related to the later use of the data, hence it must be customised according to the necessities. Following the DMN4DQ proposed in [27,28], DMN (Decision Model and Notation) [20] can be used for facilitating the description of data quality divided into measurement and assessment rules. DMN is a declarative language proposed by OMG to describe decision rules applied to a tuple of input data to obtain a tuple of outputs according to the evaluation of a set of conditions described in FEEL. A DMN table is composed of rows that describe a decision rule as an if-then condition so that, if it is satisfied, the output is returned. Also, DMN permits a hierarchical structure where the output of a DMN table can be the input of another. Using the methodology DMN4DQ, we propose to split up the measurement and assessment in two different levels for each involved dimension. Additionally, the final assessment is obtained by aggregating the assessment of every dimension, as described in Fig. 3.

[1] We use $l()$ function to define the length of a label description.

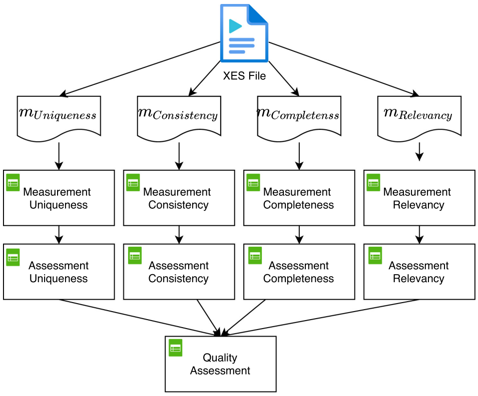


Fig. 3. DMN for describing the Quality Assessment.

DMN tables have various types of columns (`orders`, `inputs`, and `outputs`). The first column establishes the `order` by assigning an index to each row, and includes the hit policy to determine how to act when more than one is satisfied, (cf., *F* to describe that the evaluation of the condition is in order). Each `input` column represents the input variables that are evaluated the condition of the row. An example of DMN tables for the measurement of each dimension is detailed in Tables 1a, 1b, 1c and 1d, that illustrate the four dimensions proposed in this paper. Each dimension is described by a set of domains of the metric values, where the measurement of the metrics is the output value of the table. For example, the consistency dimension, in this case, describes that if the average of the numbers of characters is greater than 30, the measurement of the consistency will be *Very low*. For the measurement of each dimension, only one metric is required as input. In each row of the input column, the conditions are described in FEEL, for instance, the first row in Table 1b establishes that the valid range for the metric $m_{Uniqueness}$ is between 0 and 1. We use the metrics defined in the previous subsection as the inputs. The `outputs` represent the obtained values depending on the condition satisfied, for instance, if the input for $m_{Uniqueness}$ is 0.5 the outputs is *High*. The values and conditions established in Table 1 have been adjusted according to the know-how of experts for the use case at hand.

We should bear in mind that the measurement of a metric does not represent whether the metric is good or not, this is why a later assessment is necessary. Table 2 includes a proposal for the assessment of each dimension, for example, both a *Low* and *Very Low* number of events will imply an *Excellent* assessment in the event log according to the Complexity metric. As previously commented, this assessment has been defined based on the experts' knowledge of the event logs but other assessments can be accommodated. The assessment of each dimension needs to be aggregated to determine a global one. A possible set of decision rules for the aggregation of the assessment of the four dimensions is described in Table 3, albeit another combination can be applied according to the necessity of the organisation.

Table 1. Decision tables for measuring each dimension.

(a) Measurement of Uniqueness Dimension

F	Input $m_{Uniqueness}$	$Output_U$
1	[0, 0.1]	Very Low
2	(0.1, 0.2]	Low
3	(0.2, 0.4]	Medium
4	(0.4, 0.6]	High
5	(0.6, 1]	Very High

(b) Measurement of Consistency Dimension

F	Input $m_{Consistency}$	$Output_{Cs}$
1	[0, 5]	Very High
2	(6, 14]	High
3	(14, 20]	Medium
4	(20, 30]	Low
5	(30, ∞)	Very Low

(c) Measurement of Relevancy Dimension

F	Input $m_{Relevancy}$	$Output_R$
1	[0, 0.1]	Very High
2	(0.1, 0.2]	High
3	(0.2, 0.4]	Medium
4	(0.4, 0.6]	Low
5	(0.6, 1]	Very Low

(d) Measurement of Complexity Dimension

F	Input $m_{Complexity}$	$Output_{Cx}$
1	[0, 4]	Very Low
2	(4, 6]	Low
3	(7, 10]	Medium
4	(11, 15]	High
5	(16, ∞)	Very High

Table 2. Decision tables for the assessment of each dimension.

(a) Assessment of Uniqueness Dimension

F	Input $Output_U$	$Assess_U$
1	Very Low	Fair
2	Low $\vee$ Medium	Excellent
3	High	Poor
4	Very High	Very Poor

(b) Assessment of Consistency Dimension

F	Input $Output_{Cs}$	$Assess_{Cs}$
1	Very Low	Very Poor
2	Low	Poor
3	Medium	Fair
4	High	Good
5	Very High	Excellent

(c) Assessment of Relevancy Dimension

F	Input $Output_R$	$Assess_R$
1	Very Low $\vee$ Very High	Very Poor
2	High $\vee$ Medium	Fair
3	Low	Poor

(d) Assessment of Complexity Dimension

F	Input $Output_{Cx}$	$Assess_{Cx}$
1	Very Low $\vee$ Low	Excellent
2	Medium	Good
3	High	Poor
4	Very High	Very Poor

Table 3 is designed in such a way that, when at least 3 assessment values for the dimensions is qualified as Excellent, and the remaining one is qualified as Good or Fair, the quality outcome of the log is Excellent. Similarly, when we have three dimensions qualified as Excellent or Good and one as Fair or Poor, the quality outcome of the log will be Good. On the other hand, when we find out two dimensions as Excellent or Good and other two as Fair or Poor, the quality outcome of the log will be Fair, while,

Table 3. Aggregation of the Dimensions for the Quality Assessment

	Inputs				Output
F	$Assess_U$	$Assess_{Cs}$	$Assess_R$	$Asses_{Cx}$	$Quality_{assessment}$
1	Excellent	Excellent	Excellent	Excellent ∨ Good ∨ Fair	Excellent
2	Excellent	Excellent	Good ∨ Fair	Excellent	Excellent
3	Excellent	Good ∨ Fair	Excellent	Excellent	Excellent
4	Good ∨ Fair	Excellent	Excellent	Excellent	Excellent
5	Good ∨ Excellent	Good ∨ Excellent	Good ∨ Excellent	Good ∨ Excellent	Good
6	Poor ∨ Fair	Good ∨ Excellent	Good ∨ Excellent	Good ∨ Excellent	Good
7	Good ∨ Excellent	Poor ∨ Fair	Good ∨ Excellent	Good ∨ Excellent	Good
8	Good ∨ Excellent	Good ∨ Excellent	Poor ∨ Fair	Good ∨ Excellent	Good
9	Good ∨ Excellent	Good ∨ Excellent	Good ∨ Excellent	Poor ∨ Fair	Good
10	Poor ∨ Fair	Poor ∨ Fair	Good ∨ Excellent	Good ∨ Excellent	Fair
11	Good ∨ Excellent	Poor ∨ Fair	Poor ∨ Fair	Good ∨ Excellent	Fair
12	Good ∨ Excellent	Good ∨ Excellent	Poor ∨ Fair	Poor ∨ Fair	Fair
13	Poor ∨ Fair	Good ∨ Excellent	Good ∨ Excellent	Poor ∨ Fair	Fair
14	Very Poor	Very Poor	Very Poor	Very Poor	Very Poor
15	Very Poor	Very Poor	Very Poor ∨ Poor ∨ Fair	Very Poor ∨ Poor ∨ Fair	Very Poor
16	Very Poor ∨ Poor	Very Poor	Very Poor	Very Poor ∨ Poor ∨ Fair	Very Poor
17	Very Poor ∨ Poor ∨ Fair	Very Poor ∨ Poor ∨ Fair	Very Poor	Very Poor	Very Poor
18	-	-	-	-	Poor

when, at least, two dimensions are qualified as Very Poor, the quality outcome of the log will be Very Poor. In any other case, the quality outcome of the log will be Poor.

5 Improving Event Log Quality: NLP Techniques for Relabelling Activities

Our proposal aims to guide selection the NLP techniques for the relabelling of activities to extract the most meaningful and representative words for each process activity, but first, we need to introduce some NLP techniques.

For the sake of clarity, we take the following description of an incident as an example to show the effects of each NLP technique proposed in the paper: *"WARNING: the only way lines 1, 4 and 6 are powered is by closing up 2WR, then press PAX MASKS (in LMWS or in ICP 2011VM), afterwards PAX MASKS ON is switched on. Later, 2WR is pulled out and it can be checked that lines 1, 4 and 6 are powered. Waiting for ME confirmation and validation."*

The NLP techniques that are being proposed to be applied are described below:

Sentence Detection. This technique splits the text into its main components (i.e., sentences) to make it easier for the next steps to extract rich information from them. For our example, the application of this technique provides the next output:

- *Sentence 1.* WARNING: the only way lines 1, 4 and 6 are powered is by closing up 2WR, then press PAX MASKS (in LMWS or in ICP 2011VM), afterwards PAX MASKS ON is switched on
- *Sentence 2.* Later, 2WR is pulled out and it can be checked that lines 1, 4 and 6 are powered.
- *Sentence 3.* Waiting for ME confirmation and validation.

Each sentence is shorter and contains fewer verbs (actions) than the original, so they should be easier to analyse afterwards.

Part-Of-Speech (POS) Tagging. It consists of determining the grammatical function of each word in a text (i.e., noun, verb, adjective, preposition, pronoun, etc.), choosing for each word its corresponding class from a set of predefined tags[2]. The result of a POS-tagging on our example, selecting those words that are tagged as "NOUN", "VERB" or "ADJ" (adjective), therefore excluding the rest:

- *Sentence 1.* Only way power lines closing press switched on
- *Sentence 2.* Pulled out checked lines powered
- *Sentence 3.* Waiting confirmation validation

With this technique, we can keep those words that we consider relevant according to their grammatical category.

Lemmatisation. This technique normalises or substitutes the inflected forms by its lemma. In this way, it is easier to compare texts or even to group together different inflexions of the same lexeme. Lemmatisation can provide a more normalised text which can be better suited for relabelling. The results of the lemmatisation of our example are:

- *Sentence 1.* The only way that power the line 1 , 4 and 6 to be close 2WR, then press PAX MASKS (in LMWS or in ICP 2011VM), afterwards this switch on in PAX MASKS
- *Sentence 2.* Later pull out 2WR and check that the line 1 , 4 and 6 now to be power
- *Sentence 3.* Wait ME confirm valid.

Dependency Parsing. It determines the syntactic relationships between the words in a sentence by obtaining a *dependency tree* which provides information about the root verb of the sentence, its subject, the different objects and complements that it could contain. These are the roots of each sentence in our example detected by a dependency parser: *Sentence 1:* press; *Sentence 2:* pull; *Sentence 3:* Waiting. In this case, the dependency parser is used to extract the main verb of each sentence, so we could identify the action that characterises the corresponding activity.

Acronyms Detection. We have implemented a simple rule-based acronym detection that retrieves those words that are written in upper-cases and their lower-case form do not exist in the target language. Next, we show the acronyms detected by our approach

[2] All the tag sets used in this work come from the community open project called *Universal Dependencies* (https://universaldependencies.org/).

Table 4. Expected impact of NLP techniques on the log quality dimensions.

Dimensions	Sentence detection	POS tagging	Lemmat.	Dependency parsing	Acronym detection
$m_{Uniqueness}$	↓	↓	↓	↓	↑
$m_{Consistency}$	↓	↓	↓	↓	-
$m_{Relevancy}$	↓	↓	↓	↓	↑
$m_{Complexity}$	↑	-	-	-	-

in the example: *2WR, LMWS, 2011VM*. Depending on the context, these acronyms could be useful for detecting relevant entities in the domain.

It is important to bear in mind that the application of some of these techniques to certain texts may lead to the generation of void labels. When this happens, those events with an empty label are not included in the new log. On the other hand, when the NLP technique splits a label into several new ones (e.g., the acronyms 2WR, LMWS, 2011VM), one new event is generated for each new label, but maintaining the other attributes from the original event (e.g., the timestamp attribute).

5.1 Decision-Making for the Application of NLP Techniques

Our proposal for relabelling an event log consists of the application of the aforementioned NLP techniques to the incident descriptions, filtering out or editing them, to produce new simplified texts that are used to replace the original activity labels.

As previously commented in Sect. 5, bear in mind that the application of NLP techniques may generate or delete events according to the new labels generated. We can observe in Table 4 how the detection of sentences produces shorter texts, reducing their diversity but increasing the number of events (we will have one event per sentence, in the same order they appear within the description). Therefore, it can decrease all the proposed dimensions, except the $m_{Complexity}$ which will be increased. Concerning detection of acronyms, it can increase the $m_{Uniqueness}$ and the $m_{Relevancy}$, while the $m_{Complexity}$ and the $m_{Consistency}$ may not be noticeably affected due to the usually short length of the acronyms. The rest of NLP techniques (POS Tagging, lemmatisation, and dependency parsing) are applied to filter out irrelevant elements of the texts, keeping the important ones, and also to unify different inflexions of words into a common meaningful form. Therefore, their effects on the dimensions would be fairly similar. The $m_{Uniqueness}$ and the $m_{Relevancy}$ can be decreased since the normalisation of texts achieved by these techniques should decrease the number of unique events as well as the number of different events. $m_{Consistency}$ can be decreased as well because the length of the texts will be reduced and so will be the difference in length between them. $m_{Complexity}$ is not significantly affected because the number of traces and the number of events stays almost the same.

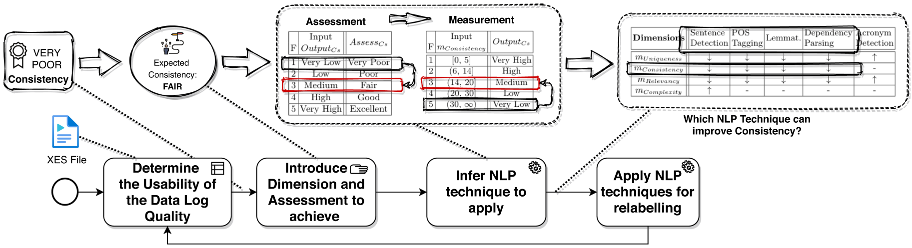


Fig. 4. Inferring the NLP technique to apply.

5.2 Inferring NLP Techniques to Improve Quality Dimensions

The relabelling of event logs through the application of NLP techniques and the definition of the dimensions and metrics discussed in previous sections provide a useful tool that guides us in the selection of the most suitable set of techniques to be applied achieving a certain assessment of quality.

Let's go back to the example proposed in Sect. 4.3. Let assume that there is an event log with a Consistency assessment ($Assess_{Cs}$) equal to $Very\ Poor$ and we want to improve it to $Fair$ as shown in Fig. 4. According to Table 2(d), we will need to take our event log from $Very\ Low$ to, at least, $Medium$ in terms of the measurement of the Consistency ($m_{Consistency}$). Then, looking at Table 1(d), it is necessary to reduce the value of the $m_{Consistency}$. Finally, according to Table 4, we can find out the NLP techniques that decrease the $m_{Consistency}$, and hence can be applied to our event log to achieve our objective. In this case they are *Sentence Detection*, *POS tagging*, *Lemmatisation*, and *Dependency parsing*.

In summary, given an event log with an assessment of quality and the dimensions to be improved, our proposal can help us to choose and apply the proper set of NLP techniques to achieve our objective.

6 Evaluation of the Proposal

For the evaluation, we use an event log (hereinafter Log_{desc}[3]) that represents the description of the evolution of the incidents in the aircraft assembly process which was presented in [29]. For instance, the following text represents a real incident description: *"When reading, the F1 error appears, wiring is verified according to FAQ and there is no continuity in any pinning in sections from 1509VC (pin 16) to 1599VC (pin 12). It is also appreciated that the colour coding concerning the plane (P1) does not correspond. Between FLKC1 and 250VC the wiring is correct"*. It can be easily observed that in this description of an incident, 3 sub-incidents are recorded: *(i) the F1 error appears*, *(ii) there is no continuity in any pinning from 16 to 12*, and; *(iii) the colour coding is incorrect*. For these reasons, textual descriptions can be useful to improve event logs

[3] Characteristics of the event log: 11.342 cases, 114.473 events, number of different labels 78.012, and 10.811 variants.

Table 5. Metrics of the event log used in the example, tagged as Log_{desc}.

Description	Total
Total number of textual descriptions	4,022
Total number of words	72,435
Out-Of-Vocabulary (OOV) words	10,832
Number of descriptions with OOVs	3,468
Grammatical and syntactic errors	1,642
Number of descriptions with errors	1,233

and discovered processes quality, but they must also be carefully processed to obtain relevant results.

The NLP is carried out with the aid of *spaCy* [16], a state-of-the-art Python library with pre-trained language models. Specifically, we have used for this work the largest Spanish pre-trained spaCy model, "es_core_news_lg"[4].

In order to illustrate the complexity of the problem, we show some metrics about Log_{desc} used for our evaluation in Table 5. We can see that 14.95% of the terms in the log are Out-Of-Vocabulary (OOV), which means that those words do not belong to the language at hand (they do not appear in the language model). In this point, domain-dependent enhancements could have been applied to the log, but the evaluation of our technique could have been biased by the nature of the domain, so we decided to keep the log as is. This complicates proper processing of the texts since 86.22% of the descriptions contain such terms. An additional difficulty from the point of view of NLP is the length of the descriptions, since too short texts may be insufficiently informative, and too long texts may be noisy for the task at hand. In this sense, Log_{desc} contains 110 descriptions with 3 or fewer words and 212 descriptions with more than 50 words. Finally, the event log has also been analysed using a grammar and spell checking tool[5], detecting a total of 1,642 errors (apart from the errors provoked by OOV words), which affects the $30,65\%$ of descriptions.

In order to evaluate the proposed steps, they have been applied to Log_{desc}, performing different sets of NLP techniques. At first, we have applied five techniques, thus, five new event logs have been created: Log_{acro}, the acronym detection is used to keep only these keywords; Log_{dep}, the dependency analysis is applied to keep only the root word of each description; Log_{lemma} contains the lemmatisation of the words; Log_{pos} applying POS tagging and keeping only those words tagged as "NOUN", "VERB" or "ADJ" (nouns, verbs and adjectives); and Log_{sent}, the sentence detection is used to split up each description into its constituent sentences. Second, we propose several pipelines of NLP techniques for the improvement of the event log quality. The Log_{sent} is used as the first step for all the proposed pipelines: Log_{sent_dep}, we simplify sentences only maintaining the root word; $Log_{sent\_dep\_lemma}$, we apply a lemmatisation to the root word previously obtained; $Log_{sent\_dep\_lemma\_acro}$, in addition to the lemmatised

[4] https://github.com/explosion/spacy-models/releases//tag/es_core_news_lg-3.0.0.

[5] Language-Tool: https://github.com/languagetool-org/languagetool.

form of the root words of each description, we also keep the acronyms within them; for Log_{sent_pos}, the POS tagging is applied to keep the nouns, verbs and adjectives of each sentence; $Log_{sent\_pos\_acros}$ the acronyms detected are added to the previous log; with $Log_{sent\_pos\_lemma}$ we keep the lemmatised forms of the words within Log_{sent_pos}; finally, $Log_{sent\_pos\_lemma\_acros}$ adds the acronyms to the previous log.

We have obtained the results of the quality assessment for each log previously described as shown in Table 6. The results show the value for each metric and the final quality reached. The results presented support how the application of the NLP techniques affect the measurements as estimated in Table 4. However, there exist dimensions, such as Relevancy, whose relation among the metric and the assessment is not lineal, e.g., when an NLP is applied to increase the relevancy metric, the assessment can become *Very Poor* instead of *Fair*.

Finally, the implementation of our framework used in this evaluation is available on a website [6].

Table 6. Dimensions values for the event logs applying NLP techniques.

Event log	Complexity	Uniqueness	Relevancy	Consistency	Quality assessment
Log_{desc}	6.734 (Excellent)	0.621 (Very Poor)	0.734 (Very Poor)	58.916 (Very Poor)	Poor
Log_{acro}	2.708 (Excellent)	0.229 (Excellent)	0.399 (Fair)	3.818 (Excellent)	Good
Log_{dep}	6.384 (Excellent)	0.177 (Excellent)	0.291 (Fair)	4.373 (Excellent)	Good
Log_{lemma}	6.707 (Excellent)	0.474 (Poor)	0.620 (Very Poor)	36.091 (Very Poor)	Poor
Log_{pos}	6.697 (Excellent)	0.439 (Poor)	0.586 (Poor)	26.671 (Poor)	Poor
Log_{sent}	8.886 (Good)	0.500 (Poor)	0.643 (Very Poor)	30.104 (Very Poor)	Very Poor
Log_{sent_dep}	8.227 (Good)	0.090 (Fair)	0.183 (Fair)	1.831 (Excellent)	Fair
$Log_{sent\_dep\_lemma}$	8.227 (Good)	0.057 (Fair)	0.126 (Fair)	1.733 (Excellent)	Fair
$Log_{sent\_dep\_lemma\_acro}$	8.325 (Good)	0.104 (Excellent)	0.190 (Fair)	2.560 (Excellent)	Good
Log_{sent_pos}	8.533 (Good)	0.413 (Poor)	0.566 (Poor)	18.924 (Fair)	Poor
$Log_{sent\_pos\_acro}$	8.564 (Good)	0.422 (Poor)	0.576 (Poor)	19.415 (Fair)	Poor
$Log_{sent\_pos\_lemma}$	8.533 (Good)	0.399 (Excellent)	0.552 (Poor)	18.563 (Fair)	Fair
$Log_{sent\_pos\_lemma\_acro}$	8.564 (Good)	0.410 (Poor)	0.562 (Poor)	19.051 (Fair)	Poor

7 Conclusions and Future Work

The preparation of an event log by carefully paying attention to its quality is crucial for the later (process mining) analysis. One of the difficulties is to ascertain when the quality is sufficient for a specific purpose, and which techniques to use to improve the quality. In this paper, we focus on the improvement of the event log quality by using NLP techniques that affect both the measurement and assessment. We propose: (1) a set of metrics to measure the quality of an event log; (2) a mechanism to describe both, data and process rules, for assessing the event log quality; and, (3) a guide for selecting the more proper NLP techniques to apply. The viability of the proposal has been demonstrated by an implementation applied to a real event log from an industrial context. As an extension of the paper, we plan to analyse how the quality of the event

[6] http://www.idea.us.es/loading-nlp.

log can be aligned with the quality of the process discovered. In addition, we will extend the number of metrics and mechanisms to improve the quality level of the event log, not only contextualised to the data textual analysis.

Acknowledgement. Projects (RTI2018-094283-B-C33, RTI2018-098062-A-I00), funded by: FEDER/Ministry of Science and Innovation - State Research, and the Junta de Andalucía via the COPERNICA (P20_01224) project.

References

1. van der Aa, H., Carmona, J., Leopold, H., Mendling, J., Padró, L.: Challenges and opportunities of applying natural language processing in business process management. In: Proceedings of the 27th COLING 2018, Santa Fe, New Mexico, USA, 20-26 August 2018, pp. 2791–2801 (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. van der Aa, H., Leopold, H., Reijers, H.A.: Comparing textual descriptions to process models - the automatic detection of inconsistencies. Inf. Syst. **64**, 447–460 (2017)
4. Van der Aalst, W.: Process Mining Discovery Conformance and Enhancement of Business Processes. Springer-Verlag, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19345-3
5. van der Aalst, W.: Extracting event data from databases to unleash process mining. In: BPM - Driving Innovation in a Digital World, pp. 105–128 (2015)
6. van der Aalst, W.: Process Mining - Data Science in Action, 2nd edn. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4_1
7. 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
8. Andrews, R., van Dun, C.G.J., Wynn, M.T., Kratsch, W., Röglinger, M., ter Hofstede, A.H.M.: Quality-informed semi-automated event log generation for process mining. Decis. Support Syst. **132**, 113265 (2020)
9. Batini, C.: Data quality assessment. In: Liu, L., Özsu, M.T. (eds.) Encyclopedia of Database Systems, 2nd edn. Springer, New York (2018). https://doi.org/10.1007/978-1-4614-8265-9
10. Bose, R.J.C., Mans, R.S., van der Aalst, W.M.: Wanna improve process mining results? In: 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), pp. 127–134. IEEE (2013)
11. Chapela-Campa, D., Mucientes, M., Lama, M.: Discovering infrequent behavioral patterns in process models. In: Carmona, J., Engels, G., Kumar, A. (eds.) BPM 2017. LNCS, vol. 10445, pp. 324–340. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65000-5_19
12. Conforti, R., La Rosa, M., ter Hofstede, A.: Timestamp repair for business process event logs (2018). http://hdl.handle.net/11343/209011
13. Denisov, V., Fahland, D., van der Aalst, W.M.P.: Repairing event logs with missing events to support performance analysis of systems with shared resources. In: Janicki, R., Sidorova, N., Chatain, T. (eds.) PETRI NETS 2020. LNCS, vol. 12152, pp. 239–259. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-51831-8_12
14. Deokar, A.V., Tao, J.: Semantics-based event log aggregation for process mining and analytics. Inf. Syst. Front. **17**(6), 1209–1226 (2015). https://doi.org/10.1007/s10796-015-9563-4

15. Fischer, D.A., Goel, K., Andrews, R., van Dun, C.G.J., Wynn, M.T., Röglinger, M.: Enhancing event log quality: detecting and quantifying timestamp imperfections. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) BPM 2020. LNCS, vol. 12168, pp. 309–326. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58666-9_18
16. Honnibal, M., Montani, I., Van Landeghem, S., Boyd, A.: spaCy: Industrial-strength Natural Language Processing in Python (2020). https://doi.org/10.5281/zenodo.1212303
17. Leopold, H., Pittke, F., Mendling, J.: Ensuring the canonicity of process models. Data Knowl. Eng. **111**, 22–38 (2017)
18. Martin, N., Martinez-Millana, A., Valdivieso, B., Fernández-Llatas, C.: Interactive data cleaning for process mining: a case study of an outpatient clinic's appointment system. In: Di Francescomarino, C., Dijkman, R., Zdun, U. (eds.) BPM 2019. LNBIP, vol. 362, pp. 532–544. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-37453-2_43
19. Mocnik, F.B., Fan, H., Zipf, A.: Data quality and fitness for purpose (2017). https://doi.org/10.13140/RG.2.2.13387.18726
20. OMG: Decision Model and Notation (DMN), Version 1.2 (2019). https://www.omg.org/spec/DMN
21. Otto, B., Lee, Y.W., Caballero, I.: Information and data quality in networked business. Electron. Mark. **21**(2), 79–81 (2011). https://doi.org/10.1007/s12525-011-0062-2
22. Pittke, F., Leopold, H., Mendling, J.: When language meets language: anti patterns resulting from mixing natural and modeling language. In: Fournier, F., Mendling, J. (eds.) BPM 2014. LNBIP, vol. 202, pp. 118–129. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15895-2_11
23. Rebmann, A., van der Aalst, H.: Extracting semantic process information from the natural language in event logs. CoRR abs/2103.11761 (2021)
24. Sadeghianasl, S., ter Hofstede, A.H.M., Suriadi, S., Turkay, S.: Collaborative and interactive detection and repair of activity labels in process event logs. In: 2nd ICPM, pp. 41–48 (2020)
25. Sadeghianasl, S., ter Hofstede, A.H.M., Wynn, M.T., Suriadi, S.: A contextual approach to detecting synonymous and polluted activity labels in process event logs. In: Panetto, H., Debruyne, C., Hepp, M., Lewis, D., Ardagna, C.A., Meersman, R. (eds.) OTM 2019. LNCS, vol. 11877, pp. 76–94. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33246-4_5
26. Suriadi, S., Andrews, R., ter Hofstede, A., Wynn, M.: Event log imperfection patterns for process mining: towards a systematic approach to cleaning event logs. Inf. Syst. **64**, 132–150 (2017)
27. Valencia-Parra, A., Parody, L., Varela-Vaca, A.J., Caballero, I., Gómez-López, M.T.: DMN4DQ: when data quality meets DMN. Decis. Support Syst. **141**, 113450 (2020)
28. Valencia-Parra, Á., Parody, L., Varela-Vaca, Á.J., Caballero, I., Gómez-López, M.T.: DMN for data quality measurement and assessment. In: Di Francescomarino, C., Dijkman, R., Zdun, U. (eds.) BPM 2019. LNBIP, vol. 362, pp. 362–374. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-37453-2_30
29. Valencia-Parra, Á., Ramos-Gutiérrez, B., Varela-Vaca, A.J., Gómez-López, M.T., Bernal, A.G.: Enabling process mining in aircraft manufactures: extracting event logs and discovering processes from complex data. In: Proceedings of the Industry Forum at BPM, Vienna, pp. 166–177 (2019)
30. Vanbrabant, L., Martin, N., Ramaekers, K., Braekers, K.: Quality of input data in emergency department simulations: framework and assessment techniques. Simul. Model. Pract. Theory **91**, 83–101 (2019)

31. Verhulst, R.: Evaluating quality of event data within event logs: an extensible framework. Master's thesis, Rijksuniversiteit Groningen, echnische Universiteit Eindhoven (2016)
32. Wynn, M.T., Sadiq, S.: Responsible process mining - a data quality perspective. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) BPM 2019. LNCS, vol. 11675, pp. 10–15. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26619-6_2

Enhancing the Value of Data in Processes Improvement (BPMDS 2021)

Detecting and Understanding Branching Frequency Changes in Process Models

Yang Lu(✉), Qifan Chen, and Simon Poon

School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia
{yalu8986,qche8411}@uni.sydney.edu.au,
simon.poon@sydney.edu.au

Abstract. Business processes are continuously evolving in order to adapt to changes due to various factors. One type of process changes are branching frequency changes, which are related to changes in frequencies between different options when there is an exclusive choice. Existing methods either cannot detect such changes or cannot provide accurate and comprehensive results. In this paper, we propose a method which takes both event logs and process models as input and generates a choice sequence for each exclusive choice in the process model. The method then identifies change points based on the choice sequences. We evaluate our method on a real-life event log. Results show that our method can identify branching frequency changes in process models and provide comprehensive results to users.

Keywords: Process science · Data science · Process drift · Concept drift · Branching frequency changes

1 Introduction and Motivation

Business processes are continuously changing in order to adapt changes to its execution environment, and such changes are called process drifts. Process drifts can be caused by different factors. Detecting and understanding such drifts can give us valuable information for process improvement.

One type of process drifts are changes of branching frequencies [2], which is related to changes in frequencies between different options when there is an exclusive choice. Detecting and understanding branching frequency changes is important. For example, in a hospital triage process, if the proportion of emergency patients is found to be increasing, more resources can be sent to the emergency department to optimise the process.

Figure 1 shows an example branching frequency change. Before the change, the frequency of choosing A and C after S are equal, and after the change, the frequency of choosing C becomes much higher. Such a change can be detected by some process drift detection methods such as [1–3]. However, it is difficult to understand such changes even if they can be detected.

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 39–46, 2021.
https://doi.org/10.1007/978-3-030-79186-5_3

Firstly, these process drift detection methods do not report if the returned process change points are related to process control-flow structure changes or branching frequency changes. A possible solution is to discover a process model based on sub-logs between every two consecutive detected process drift points, and if two consecutive process models are identical, we may conclude the detected process drift is related to branching frequency changes. However, the same process discovery algorithm can return different process models on the two sub-logs even if the process control-flow structure is unchanged. This is often due to the impacts of infrequent behaviours or the way for process discovery algorithms to deal with infrequent behaviours. We illustrate the problem in Fig. 2. Secondly, most current process discovery algorithms can only return non-stochastic process models (i.e. they show all possible activity sequences, but they do not tell their probabilities). As a result, branching frequency changes cannot be reflected on discovered process models. Lastly, for real-life event logs, different models can be discovered when different process discovery algorithms are applied. Consequently, we need to obtain a process model before analyzing branching frequency changes.

In this paper, we narrow down the problem of branching frequency changes in process models to the changes of frequencies between different options when there is an exclusive choice. We propose a method which can detect such changes and provide comprehensive results for users. The rest of the paper is structured as follows: Sect. 2 is a literature review of related work. The proposed method is presented in Sect. 3. Our method is then evaluated in Sect. 4. We finally conclude our paper in Sect. 5.

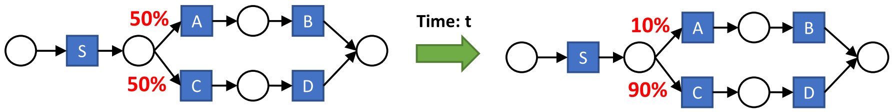


Fig. 1. An example branching frequency change from model 1 (left) to model 2 (right)

2 Background

Recent studies in process drifts define a process drift as a time point when there is a significant change among process behaviours before/after the time point [1–3]. Based on the definition, [1–3] use a sliding window to obtain two consecutive samples in the event log. Features are extracted from each sample. Then statistical hypothesis tests are performed among the consecutive samples, and if a significant change is found, a process drift can be reported. Those methods can successfully detect process control-flow structure changes and branching frequency changes, but are unable to produce a comprehensive results (i.e. they only return the time of process drifts).

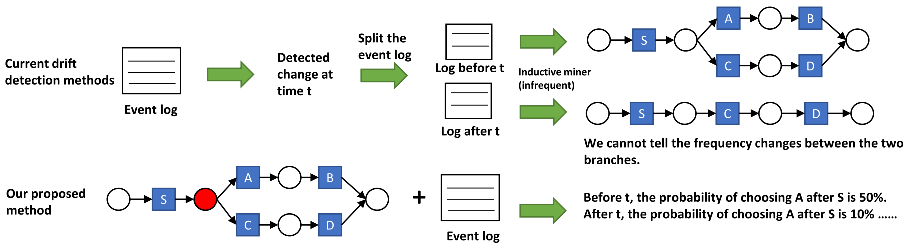


Fig. 2. Our method vs. current process drift detection methods when analysing the process drift in Fig. 1

Based on the process drift points detected by [3,8] applies the inductive miner to get a process model between every two consecutive drift points and uses nature languages to describe the difference between discovered process models. [8] can also report branching frequency changes in process models. However, as described in the previous section, we could get different process models even if the significant behaviour change is caused by a branching frequency change due to the impacts of infrequent behaviours or the ways for the process discovery algorithm to deal with infrequent behaviours. In such a case, [8] could mistakenly describe the process drift as a change in the process control-flow structure (see Fig. 2 for an example). In addition, [8] can only be applied to block-structured [9] process models.

Some process drift detection methods are not based on statistical tests. For example, the TPCDD [10] reports a process drift whenever a new behaviour is observed or an existing behaviour is removed for a certain amount of time. As branching frequency changes will not bring new behaviours or remove existing behaviours to/from the process, TPCDD [10] cannot report such changes.

Other process drift detection methods aim at providing comprehensive results to users such as [6,7,11,12]. [11] mines process models for different time periods and compares graph matrices of different models. [12] mines models for the first period of time and performs conformance checking on each new trace. A drift point is reported if there is a significant change on the conformance results. [6,7] use Declare miners to represent the process, and a comprehensive visualisation is provided for users to understand process drifts. Those methods bring research about process drifts to a new stage, but still cannot provide comprehensive results for branching frequency changes in process models.

3 The Proposed Method

Figure 3 shows an overview of our proposed method. The inputs of the method are a Petri net and event log. Based on the inputs, the proposed method will detect branching frequency change points and generate a comprehensive report. More specifically, only sound Petri nets can be used as the input. For the definition of soundness, we refer to [13].

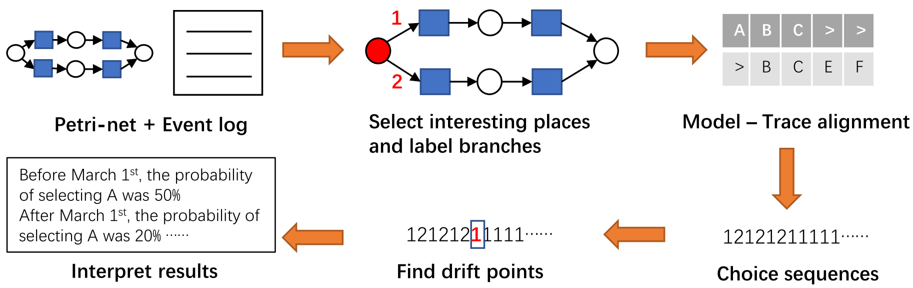


Fig. 3. An overview of the proposed method

3.1 Step 1: Discovering Process Models and Selecting Interesting Places

In the first step, users should discover a process model based on the event log. The process model can be discovered by any process discovery algorithms or manually created by domain experts. In the method proposed in this paper, all process models should be transformed to sound Petri nets.

In a sound Petri net, an exclusive choice split is caused by a place with more than one outgoing arcs. If users are interested in frequency changes in one or multiple exclusive choices, they can select such places and only focus on process drifts on the places they select. Otherwise all places with more than one outgoing arcs can be selected. In this paper, we call selected places "interesting places".

3.2 Step 2: Discovering Choice Sequences

To describe the sequence of decisions each time a choice is needed to be made on the process model, we propose a new concept named choice sequence. A choice sequence for a place p is a sequence of categorical variables, each refers to an outgoing arc of p and corresponds to a timestamp. We firstly give a label to each outgoing arc of an interesting place. Then each time a choice is made, a new element is added into the choice sequence. In our implementation, we use an integer to label each arc. Figure 4 shows a simple example. For place p, the first outgoing arc is labeled as 1 while the second arc is labeled as 2. Each time the first (second) arc is selected, 1 (2) is added into the choice sequence.

However, when dealing with real-life event logs and process models, some traces may not conform with the process model. In addition, Petri nets can also contain hidden transitions (i.e. transitions without activity labels.). Those factors make it hard for us to determine which outgoing arc of the interesting place has been selected during process executions. To accurately produce choice sequences, we perform model-trace alignments [4] between the event log and process model. For each trace, we find its corresponding model move sequence. Then for each interesting place p, we find its incoming and outgoing transitions. If we find two consecutive model moves where the first one belongs to the incoming transitions of p, and the second model move belongs to the outgoing transitions of p, we add a new element into the choice sequence of place p. In addition, we take the timestamp of the closest trace move which is before (or corresponded to) the first model move for the new element. We finally sort the choice sequence based on the timestamps. Figure 4 shows a complete example of this step.

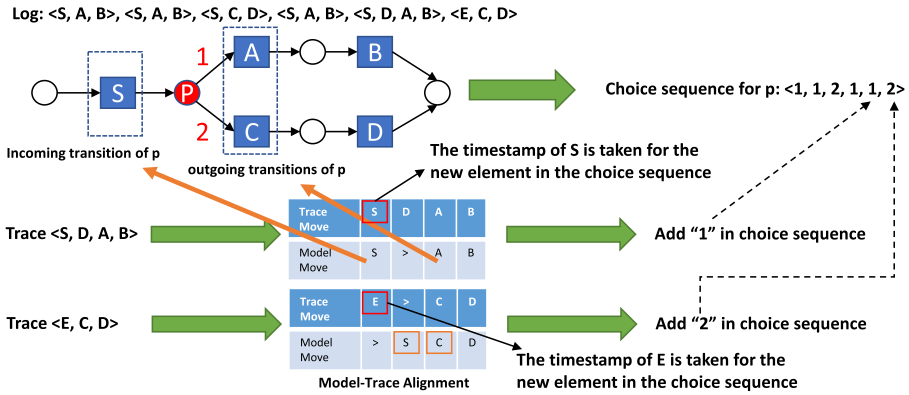


Fig. 4. Discovering choice sequences

3.3 Step 3: Detecting Change Points

After choice sequences are detected, the problem of finding branching frequency changes is changed into identifying a change point on sequences of categorical variables. We can then apply other change point detection methods to identify change points. In our implementation, we use an external library called ruptures [5]. Ruptures is a python package which contains a collection of change point detection algorithms for various dynamic systems [5]. It has also been used in detecting process drifts in [6,7].

Since each place has its own choice sequence, each time a change point is discovered, we can easily know what has been changed. For example, in Fig. 4, if we detect a change point on the choice sequence of place p, we know the process drift is related to the choice between activity A and C after S. As a result, by calculating the frequency of "1" and "2" in the choice sequence, a comprehensive report about the frequency changes between "A" and "C" after "S" can be provided.

4 Evaluation

The proposed method is implemented as a Python program based on the PM4PY [14] framework, and a prototype is publicly available[1].

[1] https://github.com/bearlu1996/FrequencyChange.

We evaluate our method on a real-life event log which is publicly available from the "4TU Data Center"[2]. The event log describes a ticketing management process of the help desk of an Italian software company.

The event log has also been used to evaluate the process drift characterization method in [8]. [8] reports two process drifts in the event log, one of which is related to a branching frequency change. Although we do not know the exact ground truth about process drifts in the event log, we can validate our results by comparing to the results provided by [8].

4.1 Step 1: Discovering Process Models and Selecting Interesting Places

In order to compare our results with [8], we directly use the process model discovered by [8] as the input model for our method. The model is discovered by the inductive miner, and infrequent behaviours are filtered out. The model is presented in Fig. 5. In this study, we select the place before "Wait" as the interesting place (marked in red) since we want to learn the change of frequencies between choosing "Wait", "Require upgrade" and "Resolve ticket" after "Take in charge ticket". Then the three outgoing arcs are automatically labeled as "1", "2" and "3". Each label represents an outgoing arc after "Take in charge ticket".

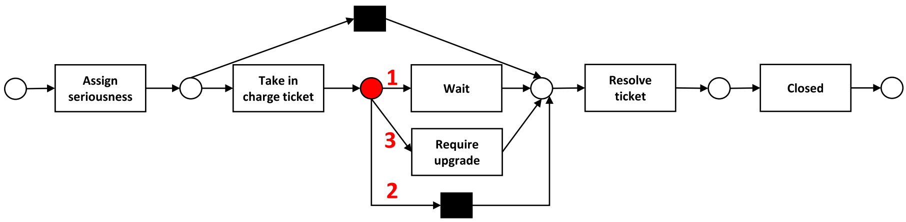


Fig. 5. Process model of the "Italian help desk" log from [8] (Color figure online)

4.2 Step 2: Discovering Choice Sequences

After the interesting place is set, its incoming and outgoing transitions are then identified. In our case, the incoming transition of the selected place is "Take in charge ticket", while the outgoing transitions are "Wait", "Require upgrade" and a hidden transition. We then perform model-trace alignments and generate a choice sequence for the selected place. Finally, we sort the choice sequence by the timestamp of each element.

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

4.3 Step 3: Detecting Change Points

Finally, the choice sequence is treated as input for Ruptures [5] to identify change points[3]. Two change points are detected in total, the first one is on Aug 31st, 2012, and the second one is on Apr 29th, 2013. A comprehensive result is presented in Fig. 6. Before Aug 31st, 2012, The frequency of choosing "Require upgrade" after "Take in charge ticket" is very low (0.2%). The frequency of skipping to "Resolve ticket" (78.2%) is much higher than "Wait" (21.6%). Between Aug 31st, 2012 and Apr 29th, 2013, the frequency of choosing "Require upgrade" is much higher than before (9.6%), and the frequency of "wait" and skipping to "Resolve ticket" are almost the same (around 45%). After the second drift on Apr 29th, 2013, the frequency of choosing "Require upgrade" almost remains unchanged, but the frequency of choosing "Wait" (64.3%) is much higher than the frequency for choosing skipping to "Resolve ticket" (25.5%).

The first drift point is detected by both our method and [8][4]. In addition, our method also reports one more frequency change point. Since the frequency change between "Wait" and skipping to "Resolve to ticket" is significant (the change is around 20%), we believe the second drift point we find is correct.

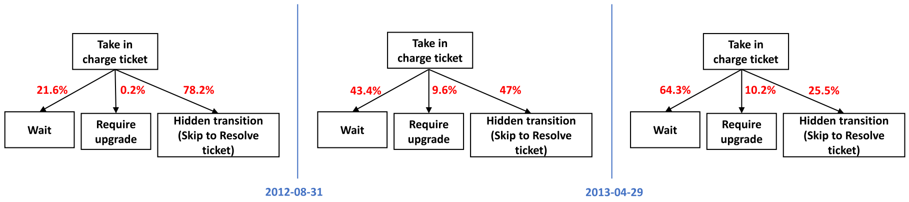


Fig. 6. Visualisation of the detected branching frequency changes

5 Conclusion

In this paper, we present a method to detect branching frequency changes in process models. The method can not only detect changes in processes, but can also provide comprehensive results to users. By retrieving information about branching frequency changes from processes, subsequent analysis can be performed for process improvement.

Future work may include different aspects. Firstly, we can integrate our method with a user interface which can visualize branching frequency changes in process models. Secondly, we aim at applying other change point detection methods to detect changes on choice sequences and empirically compare different methods.

[3] In this study, we use Pelt as the search method, CostRbf as the cost function. The penalty value is set to 5. For details, please refer to [5].

[4] In [8], the drift is detected on Sep 11th, 2012.

References

1. Maaradji, A., Dumas, M., La Rosa, M., Ostovar, A.: Fast and accurate business process drift detection. In: Motahari-Nezhad, H.R., Recker, J., Weidlich, M. (eds.) BPM 2015. LNCS, vol. 9253, pp. 406–422. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23063-4_27
2. Maaradji, A., Dumas, M., La Rosa, M., Ostovar, A.: Detecting sudden and gradual drifts in business processes from execution traces. IEEE TKDE **29**(10), 2140–2154 (2017)
3. Ostovar, A., Maaradji, A., La Rosa, M., ter Hofstede, A.H.M., van Dongen, B.F.V.: Detecting drift from event streams of unpredictable business processes. In: Comyn-Wattiau, I., Tanaka, K., Song, I.-Y., Yamamoto, S., Saeki, M. (eds.) ER 2016. LNCS, vol. 9974, pp. 330–346. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46397-1_26
4. van der Aalst, W.M.P., Adriansyah, A., van Dongen, B.F.: Replaying history on process models for conformance checking and performance analysis. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. **2**(2), 182–192 (2012)
5. Truong, C., Oudre, L., Vayatis, N.: Selective review of offline change point detection methods. Signal Process. **167** (2020)
6. Yeshchenko, A., Di Ciccio, C., Mendling, J., Polyvyanyy, A.: Visual drift detection for event sequence data of business processes. arXiv preprint (2020)
7. Yeshchenko, A., Di Ciccio, C., Mendling, J., Polyvyanyy, A.: Comprehensive process drift detection with visual analytics. In: Laender, A.H.F., Pernici, B., Lim, E.-P., de Oliveira, J.P.M. (eds.) ER 2019. LNCS, vol. 11788, pp. 119–135. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33223-5_11
8. Ostovar, A., Leemans, S.J., Rosa, M.L.: Robust drift characterization from event streams of business processes. ACM Trans. Knowl. Discov. Data (TKDD) **14**(3), 1–57 (2020)
9. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from event logs - a constructive approach. In: Colom, J.-M., Desel, J. (eds.) PETRI NETS 2013. LNCS, vol. 7927, pp. 311–329. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38697-8_17
10. Zheng, C., Wen, L., Wang, J.: Detecting process concept drifts from event logs. In: Panetto, H., Debruyne, C., Gaaloul, W., Papazoglou, M., Paschke, A., Ardagna, C.A., Meersman, R. (eds.) OTM 2017. LNCS, vol. 10573, pp. 524–542. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69462-7_33
11. Seeliger, A., Nolle, T., Mühlhäuser, M.: Detecting concept drift in processes using graph metrics on process graphs. In: Proceedings of the 9th Conference on Subject-Oriented Business Process Management, p. 6:1 (2017)
12. Stertz, F., Rinderle-Ma, S.: Process histories - detecting and representing concept drifts based on event streams. In: Panetto, H., Debruyne, C., Proper, H.A., Ardagna, C.A., Roman, D., Meersman, R. (eds.) OTM 2018. LNCS, vol. 11229, pp. 318–335. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02610-3_18
13. Aalst, W.: Data science in action. Process Mining, pp. 3–23. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4_1
14. 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. CoRR abs/1905.06169 (2019)

Detecting the "Split-Cases" Workaround in Event Logs

Yael Dubinsky(✉) and Pnina Soffer(✉)

University of Haifa, Mount Carmel, 3498838 Haifa, Israel
ddubinsk@campus.haifa.ac.il, spnina@is.haifa.ac.il

Abstract. Workarounds are frequently observed in business processes, where employees intentionally bypass or deviate from procedures and business rules embedded in process models. One of the common workaround behaviors is the "split cases" workaround. This workaround typically takes place in processes where threshold conditions over data items are used by decision rules, determining the process path to be followed. Being familiar with the process rules, employees may decide to split a case whose relevant data value exceeds the threshold into two or more cases whose data value is below the threshold. Doing this they avoid the path which should be followed otherwise – this path may seem undesirable to the employee (e.g., including lengthy approvals), but might be needed for avoiding risks. Detecting such workarounds is hence of importance.

Although the split case workaround is quite common, existing process mining and conformance checking techniques are not geared to detect it in an event log, since each of the cases is conformant by itself, and it is the relation among cases which might indicate this behavior. In this paper we propose an approach for detecting the split cases workaround in event log and report experimentation that has been performed using simulated as well as real life logs.

Keywords: Process mining · Workaround · Inter-case behavior

1 Introduction

The popularity of process-aware information systems (PAIS) in organizations has greatly expanded in the past years. These systems manage and execute operational processes, based on process models, enforcing the rules that are encoded in these models, so tasks are performed in an orderly manner without being skipped or postponed. In particular, the process model may involve decision rules on split nodes, in the form of conditions that are based on values of data-items involved in the process, to determine the next activity to be executed and the process path to be taken. Often, employees who use such systems as part of their jobs, perceive these conditions as restrictions which form obstacles for a proper execution of their tasks, and sometimes also hamper achievement of personal and organizational goals [1]. These employees often perceive that "the damage is greater than the benefit" in following processes enforced by such systems, and will often search for workarounds for avoiding the restrictions dictated by the process model.

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 47–61, 2021.
https://doi.org/10.1007/978-3-030-79186-5_4

One of the environments where workarounds are frequently observed are processes which include many approval activities, required by conditions over data values (e.g., when an amount exceeds a certain threshold). These required approvals typically slow the process and lead to what is often perceived as tedious tasks, sometimes leading to regulatory involvement and prevention of intended actions. A common type of workaround in such processes involves splitting one process instance (case) into several ones, so each instance separately is "eligible" to an easier, more preferable path than the one that would need to go through a long approval trail [2]. Taking this workaround, the process participant is aware of the detailed rules of the process and the conditions driving it, and aims to avoid certain activities or approvals which should be triggered by these conditions. In order to do so, the process participant deliberately chooses data values which, based on the conditions, lead to preferred paths. In this paper, we relate to this type of workaround as the split-cases workaround.

The split-cases workaround is fairly common in many contexts, and might sometimes have severe consequences. While often it is performed by employees with a good intention of quickly advancing processes in urgent cases, it may also be taken for performing fraud. For example, it had been revealed in 2018 that Nigerian military chiefs and companies were involved in a vast procurement fraud [3], which included splitting procurement contracts, and concealing benefits and incentives given to suppliers. Another common example is money laundering: splitting one financial transaction to several smaller transactions, in order to evade special taxes and fees. One may also split incomes in order to evade required taxes. Detecting split cases in such processes is essential, as it is a substantial problem and might reveal crucial law violations.

In this paper, the split cases workaround is addressed in an enterprise setting, concerning processes such as purchase or sales orders management. Splitting purchase or sales orders might be performed in order to avoid approvals which are required over certain amounts and sums. Splitting cases may also be performed to evade regulations which restrict the amounts and sums being ordered. Detecting split cases in these processes is essential, as the approvals and tasks that are bypassed are likely to be crucial for valid process execution, for minimizing business risks, and for complying with regulations and legislation.

Despite the importance of the detection of such behaviors, existing process mining techniques are not geared to detect them [4]. Up until now, process mining techniques focused mainly on examining intra-case behaviors – examination of each case in the event log separately. Recently, research concerning inter-case behaviors has emerged, focusing on methods of discovering instance-spanning constraints (ISC) from event logs [5, 6]. Unfortunately, the split cases behavior cannot be unveiled solely by ISC-based techniques, as there is no constraint to be violated in this behavior. Overall, most process mining-based techniques of conformance checking aim to identify and quantify behaviors which deviate from the expected execution as defined for each single process instance (according to the process model) [7]. However, when a split-cases workaround is performed, each of the resulting cases by itself may be fairly conformant; it is the specific connection between the instances that matters. Thus, the existing techniques are not geared to detect such occurrences. In order to detect this behavior, it is required to establish connection among cases, concerning the control flow, data values, the involved resources, and time periods.

In this paper, we aim to understand, define, and characterize the split cases behavior in business processes, and to suggest an approach for detecting it in an event log. We define and formalize the behavior relying on the Petri net with data notation [8], and present a set of expected characteristics of this behavior based on the formalization, demonstrated by a motivating example. The suggested detection approach is based on these definitions and characteristics. It is then evaluated using both simulated and real-life logs.

The rest of this paper is organized as follows. Section 2 presents a motivating example and a formalization of the split cases behavior. Sections 3 is devoted to the detection approach and its required preliminaries. Section 4 presents the evaluation of the approach, using both simulated and real-life logs. Section 5 discusses related work, and Sect. 6 concludes and discusses future research directions.

2 Problem Definition

2.1 Motivating Example

To illustrate the split-cases behavior, consider the following order handling process, described in Fig. 1.

A case is created by a sales representative when a customer requests a new order. Once the products and quantities are entered, the system calculates the total price of the order. If the total price exceeds $200K, an approval of the department manager is required, since the deal might be risky. After a manager's approval is given, an employee calculates the shipping fee according to the destination and quantity, and assigns the shipment. After the order is successfully shipped to the customer, payment for the order should be handled and processed, and eventually the order is closed.

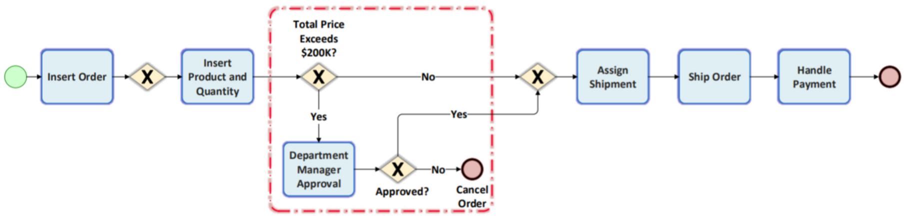


Fig. 1. Order handling process

In this process, facing a specific order whose priority is high so delays must be avoided, the sales representative may perform the split cases workaround in order to avoid the department manager's approval. The sales representative can calculate the total price of the products and quantities being ordered – which exceeds $200K. The sales representative is also aware of the condition which determines whether an order is required to go through the department manager's approval (see Fig. 1). Thus, to avoid the approval and the delay it may cause, the employee divides the ordered products to two orders, and creates two process cases instead of one, where the total price of each order in each case does not exceed $200K. The two cases are then executed separately, both bypassing the department manager's approval. Eventually, the customer receives the originally requested products and quantities, and the manager's approval activity has been avoided. In what follows, we provide a general formulation of the problem and its manifestation in an event log.

2.2 Problem Formalization

In order to formalize the split cases behavior in event logs, we rely the Petri net with data (DPN) [8] specification.

A Petri net with data DPN = (P, T, F, D, U, R, W, G) consists of:

- a Petri net (P, T, F), where P is a set of places, T a set of transitions, and F a set of flows;
- a set D of data items;
- a function U that defines the domain of values admissible for each data item d ∈ D;
- a read function $R \in T \rightarrow 2^D$ that labels each transition with the set of data items that it must read;
- a write function $W \in T \rightarrow 2^D$ that labels each transition with the set of data items that it must write;
- a guard function $G \in T \rightarrow \Phi(d)$ that associates a guard with each transition, where $\Phi(d)$ is a predicate in d.

As the split cases behavior may occur only under certain circumstances, we assume the following to hold for such behavior to be plausible:

(1) There exists a value function V which assigns a partial order relation to traces σ_i, σ_j, so it can be said that $V(\sigma_i) > V(\sigma_j)$. Note that this function is not necessarily explicit or even known, but it marks that an employee may favor one possible process path (and resulting trace) over another. Considerations may relate to time, effort, certainty of the outcomes, or even personal relations with other employees.

(2) There exists a data item d and at least two transitions t_1, t_2 (each marking or leading to a different process path), such that different guard conditions over d are set for t_1 t_2. Formally: $G(t_1) = \Phi_1(d), G(t_2) = \Phi_2(d), \Phi_1(d) \neq \Phi_2(d)$.

(3) Consider two transitions t_1, t_2, that satisfy (2). For two traces σ_1, σ_2, such that $t_1 \in \sigma_1, t_2 \in \sigma_2$, and $t_1, t_2 \notin \sigma_1 \cap \sigma_2$, $V(\sigma_1) \neq V(\sigma_2)$ holds. We then say that V *depends on d*. In other words, the value of d may determine whether a preferable path is taken or not.

The existence of the above assumptions in a process may motivate the creation of split cases. A typical scenario is of a case whose value of the decision data item d should lead to an undesirable path. To avoid this and improve the value function V, an employee might decide to create several split cases instead, each having a d value that warrants a preferable path, where the total sum of d over the split cases would be equal to the original d. In the log generated by these process executions, the original case would not be present, and it is hence termed a *concealed case*. Instead, the log would include the *split cases* as formalized in Definition 1.

Definition 1 (Concealed Case and Split Cases). Given an event log and a process model, a *concealed case* CC is an assumed case whose trace does not appear in the log. Instead, the event log includes traces of at least two *split cases* SC_i.

Let σ_{CC} be a trace fully aligned with the process model based on the data values of CC. Let d be a data item on which V depends. We denote the value of d for case C as $C.d$. Then $CC.d = \sum_j SC_j.d$ and $\mathrm{V}(\sigma_{CC}) < \mathrm{V}\left(\sigma_{SC_j}\right)$ for all j.

As our aim is to discover split cases in an event log, relying on the above discussion, we can infer several properties of the split cases behavior as reflected in a log:

- This behavior can be observed in processes where a data item d exists so some process property depends on it. For example, a process in which the control flow includes a decision rule that depends on the value of d. As split-case behavior takes place upon the creation of a case, it is reasonable to assume that d represents a property of the case object – the main entity managed by the process.
- A process participant is able to perceive a value function V for different possible traces in the process and may prefer certain traces over other traces.
- Split cases related to one concealed case are created by a single process participant and in time proximity to one another.
- Split cases are similar to one another in their data values (in correspondence to the properties of the concealed case).

Note that the assumptions and premises are formulated concerning a value function that depends on the control flow of the process. In practice, value functions may relate to additional aspects and the motivation for splitting cases may not be reflected in the control flow. Nevertheless, as such value functions may not yield detectable behavior, we focus on control-flow related ones. In the following section we build on the above premises as a basis for developing an approach for detecting split cases in an event log.

3 Approach

3.1 Preliminary Steps and Inputs

Based on the problem formulation and premises discussed in Sect. 2, the proposed detection approach requires some preliminary steps for assessing whether these premises are valid considering the data at hand and for identifying inputs that are required for the detection method. These include:

- Identification of a data item d on which decision rules (or guards) in the process depend. The value of d should be known upon case creation for the motivation to split cases to exist. We note that identification of data items that affect decision rules can be done using existing techniques, e.g., the one suggested in [9].

– Identification of threshold conditions over *d*: Typically, a decision rule, as specified in guard conditions, is in the form of a predicate in *d*, such that a *threshold* value over *d* determines the process path to be selected. Different values of *d* may lead to different activities, according to the threshold value. Threshold values as part of guard conditions can also be detected using decision mining [9] or other techniques for learning the threshold from the data. Furthermore, we expect the traces to be generated by values below and above the threshold to be differently ranked according to some value function V. Note that V is typically unknown when attempting to detect split cases, and domain knowledge might be needed for suggesting a plausible one (e.g., a smaller number of activities is more preferable than a larger one). Our assumption is that values above the identified threshold would be considered less desirable (so splitting the cases and reducing *d* in each case would lead to a preferable path).
– Uniting Attribute: as discussed in Sect. 2, we expect split cases to be similar in many of their data values, sharing attributes of the concealed case. However, different scenarios may be expected in this respect. For example, a customer order can be split so each new order will have part of the quantities ordered in the concealed case, resulting in split cases that have the same products, the same due dates, and the same customer. Alternatively, different products and due dates may be assigned to different split cases, yet all cases share the same customer. It follows that we expect at least one shared attribute for the split cases, that would indicate an originating concealed case. This "uniting" attribute can be expected beforehand based on domain knowledge, but can also be inferred from the data. Note that the resource who creates the cases is shared among the split ones by our premise that it is this resource who initiates the splitting. The uniting attribute should be common to the split cases in addition to the resource. In our proposed method we assume it to be an input specified by the analyst.

3.2 Detection Algorithm

Figure 2 provides an overview of the split cases detection algorithm[1]. The main idea of the algorithm is to form sets of cases, where each set consists of cases suspected as split cases. Based on the described premises, we suggest that a set of cases is suspected as split if all cases follow a preferable path (namely, hold a value of *d* which is below the threshold), hold the same value of *u* – the uniting attribute, and are created by the same resource in time proximity to one another (case creation is identified as the first event of a case in the log). After forming such sets, we aggregate the values of the splitted column *d* over cases in each set. This aggregation should represent the value of *d* for the concealed case. A suspected set of cases is classified as split cases and returned by the algorithm if the value of *d* crosses the threshold of the condition, and thus would lead the concealed case to an undesired path.

To this end, the algorithm uses a moving *timeframe*, so cases that are initiated within this relatively short timeframe can be considered in time proximity to each other (following our premises). The timeframe is moved along the log in iterations. Both the timeframe and the iteration size are given as inputs to the algorithm:

[1] Code is available at https://github.com/yael935/Split_Cases. The implemented algorithm addresses one uniting attribute, and can be easily generalized to multiple attributes.

– *Timeframe Size*: the size of the moving timeframe to be used by the algorithm. This variable is specified as X * (mean time between cases).
– *Iteration Size*: the amount of time the timeframe is moved in each iteration.

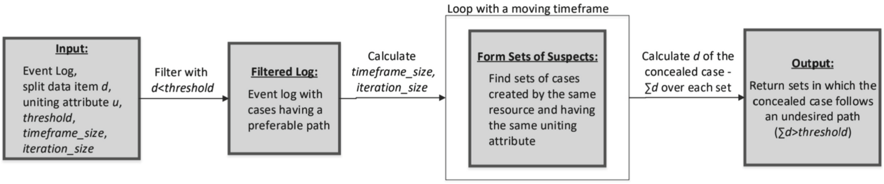


Fig. 2. An overview of the detection algorithm

Listing 1 presents the pseudo-code for the detection algorithm. The input includes an event log, a split data item, a condition threshold, a uniting data item, timeframe size, and iteration size. The output of the algorithm is a set of groups of cases, where each group contains cases suspected as split.

Listing 1. The split-case detection algorithm

```
Input: event log E, data item d, condition threshold value
threshold, uniting data item u, timeframe size timeframe_size,
iteration size iteration_size.
Output: a set of sets S2, where each set contains cases
suspected as splitted.
Begin
1. E1 ← E filtered on cases where d < threshold
2. E2 ← E1 filtered on first event of each case
3. S ← {}
4. For frame_start = minimal timestamp in E2;
   frame_start + timeframe_size < maximal timestamp in E2;
   4.1. E3' ← E2 filtered on timestamp between frame_start and
        frame_start + timeframe_size
   4.2. E3'' ← E3' sorted by resource
   4.3. For each resource r in E3'':
        4.3.1. For each value of u, u_i:
               4.3.1.1. RS ← {}
               4.3.1.2. RS ←{g | g contains cases from E3'', where
                        resource=r
                        and u= u_i}
# set of cases created in time proximity from one another, by the same resource, and hold the
same uniting attribute.
               4.3.1.3. S ← S + RS
   4.4.  frame_start = frame_start + iteration_size:

5. For each set g in S:
   5.1. aggregate_split_col ← ∑_{case in g} d
   5.2. If aggregate_split_col > threshold:
        5.2.1. S2 ← S2 + g
6. Return S2
End
```

4 Evaluation

The split cases detection method was primarily evaluated based on simulated event logs, including two sets of experiments. The main benefit of using simulated event logs is the ability to control the behavior of elements and data values in the log and to establish a ground truth of split cases. Based on this ground truth, the results could be evaluated in terms of precision, recall, and F1.

However, evaluation of the detection method solely based on simulated logs is insufficient and threatens the external validity. A simulated log is usually smaller than real-life event logs, and contains less unpredictable behaviors. The size and complexity of real-life logs is a threat to the scalability of the examined detection method. In order to overcome this threat and assess scalability and applicability of the approach, experimentation with a real-life log was also conducted.

The main limitation of experimentation with real-life event logs is the lack of knowledge of the ground truth, namely, the actual occurrences of the split cases behavior in the log. As the quantification of success rates of the detection problem is possible only when the ground truth is known, it is not possible to measure the success of detection in real-life logs the way it is done for simulated logs. The aim here is to assess the generalizability, scalability and applicability of the detection method. Thus, real-life logs are experimented as a proof of concept, while success is determined by whether the method is able to detect any occurrences of the split cases behavior. We also performed a qualitative analysis of the detected occurences in order to gain a better understandance of the detected behavior.

4.1 Experiments Using Simulated Event Logs

The simulated logs represent the order handling process used as the running example (Fig. 1). Nine simulated event logs were created for this experiment, each simulating a period of one year. Five logs represent normal process executions, without any irregular behaviors. Two other logs also represent normal process executions, but with a smaller frequency of split cases. The last two logs represent specific scenarios of process executions, which may be difficult for the detection method to handle.

One specific scenario that was simulated considers account executives: employees who dedicatedly handle the same customers. Considering the customer ID as a uniting attribute, this might mislead the detection method to identify sets of cases that have resource-customer matchings as split cases. The second specific scenario relates to busy work days, when the number of created cases reaches a peak. In an irregularly busy work day a large number of cases are created within a short time. The simulated log related to this behavior included 15 busy days along the year, while the rest 350 days behaved normally.

The detection method inspects a moving timeframe; cases which are created within the same timeframe might be considered as suspects for being split. The two described scenarios might result in an increased number of false positives, and thus decrease the precision of the detection. The time between creations of split cases in all simulated logs ranges from 1 to 20 min, distributed uniformly. Table 1 summarizes the statistics of the simulated logs.

Table 1. Simulated logs used in this experiment

Event Log File	Behavior	# of Cases	# of Events	# of Split Cases	Mean Time Between Cases (minutes)
1	Normal	10516	84112	500	52:13
2	Normal	10505	82039	500	50:49
3	Normal	10512	82261	500	48:44
4	Normal	10361	82593	500	51:37
5	Normal	10454	81788	500	50:16
6	Normal	10384	82137	300	51:38
7	Normal	10412	83411	100	52:21
8	Account Executive	10497	82645	500	50:08
9	Peak Days	10422	84008	500	49:35

The aim of the experiments using simulated logs was threefold. First, we wanted to test the performance of the approach in terms of precision, recall, and F1 [10]. Second, as the algorithm uses input parameters of timeframe size and iteration size, we wanted to assess the sensitivity of the detection to the values of these parameters. Furthermore, we expect that as the timeframe size is increased, the recall would increase until some stability would be reached; at the same time, precision may start to decrease for larger timeframe sizes. We hence expect the F1 measure to indicate an optimal value of this parameter. Last, as the algorithm involves iterations over the log with a moving (relatively short) timeframe, scalability to large logs might be an issue and result in long execution times. To test this, we measured the execution times of the algorithms for the different configurations of timeframe and iteration sizes.

For the first set of experiments we kept a constant ratio between the timeframe size and the iteration size, and tested different values of the timeframe size. These parameters were calculated as follows:

$$\textit{Timeframe size} = X * (\textit{Mean time between cases})$$
$$\textit{Iteration size} = Y * (\textit{Mean time between cases}), \quad Y = \frac{X}{4}$$

The algorithm was executed for all nine simulated logs. Our implementation was in Python via PyCharm IDE, and the experiments were conducted using a CPU of 12 threads, 6 cores, and clock speed of 4.3 GHz, and a 16 GB RAM. Resulting precision, recall, F1, and average execution times for different values of X are shown in Table 2. The table shows the average precision and recall values for the seven logs that simulate normal behavior, and separately the values for the logs that simulate the specific scenarios. F1 values and execution times are presented as average for all logs. Note that our experiments did not yield any difference in the results related to different numbers of split cases in the log (log 6 and 7), and hence we do not show these separately, but as part of the average results for the "normal behavior" logs.

As seen in Table 2, the recall values are similar for the "normal" behavior logs and the ones that simulate specific "difficult" scenarios. This result is expected, as the irregular behaviors (peak days and customer executives) may add false positives, but these do not affect recall values. Recall is low for low values of X and reaches values over 0.99 for $X \geq 0.4$, indicating that for this range of timeframe sizes the detection method is able to detect nearly all the split cases. As X stands for the ratio between the timeframe size and the mean time between cases, small values of X imply that the timeframe size is

Table 2. Precision, recall, F1, and execution times for different timeframe sizes with a constant ratio between timeframe size and iteration size

		Precision			Recall			Average F1	Average Execution Time (Minutes)
X	Y	Normal	Account Executives	Peak Day	Normal	Account Executives	Peak Day		
0.1	0.025	0.809	0.732	0.734	0.257	0.274	0.267	0.393	6:07
0.2	0.05	0.903	0.854	0.846	0.555	0.568	0.563	0.684	3:55
0.3	0.075	0.936	0.912	0.914	0.854	0.859	0.863	0.890	3:05
0.4	0.1	0.946	0.919	0.916	0.997	0.995	0.996	0.963	2:41
0.5	0.125	0.946	0.921	0.917	0.997	0.997	0.999	0.964	2:26
0.6	0.15	0.946	0.917	0.908	0.997	0.998	1.000	0.962	2:16
0.7	0.175	0.946	0.912	0.904	0.997	0.998	0.998	0.961	2:09
0.8	0.2	0.945	0.907	0.904	0.995	0.998	0.998	0.960	2:03
0.9	0.225	0.946	0.906	0.903	0.995	0.996	0.997	0.959	2:00

too small for capturing split cases that were not created immediately one by the other. The mean time between cases in the simulated logs is approximately 50 min. Values of 0.1–0.3 of X imply timeframe sizes of 5 to 15 min, which means that only split cases that were created up to 5–15 min from one another are found by the algorithm. As split cases in the simulated logs are created up to 20 min from one another, it is reasonable that many of them were not detected with smaller timeframes.

The average values of precision for normal behavior logs are over 0.8 for all values of X, and reach a stable value of 0.946 for X $\geq$ 0.4, implying that for this range of timeframe sizes the detection algorithm performs well in terms of precision as well and does not yield large numbers of false positive split cases. This result is understandable, as the iteration size increased proportionally to the timeframe size. With a fixed iteration size, the predicted results could have grown, and precision would have decreased.

However, for the logs that simulate specific irregular scenarios, lower precision values were achieved, and a different behavior of the precision measure was observed. As expected, precision values increased with the increase of the timeframe size until a maximum point (achieved for X = 0.5), after which increasing the timeframe size resulted in decreasing precision values. This implies that higher timeframe sizes resulted in false positive identification of split cases. We still note that even the precision values obtained for X = 0.9 (the largest in the experiment) were still over 0.9.

The average F1 values, which combine precision and recall, increase with the timeframe size until a maximum value of 0.964 is reached for X = 0.5. Additional increases of the timeframe size lead to a (slight) decrease in F1, reflecting the decreasing precision. It can hence be concluded that X = 0.5 reflects an optimal value of the timeframe size in this setting, yet for every timeframe where X $\geq$ 0.4 F1 is over 0.95.

The average execution times for this set of experiments are also presented in Table 2, showing that the execution time decreases as both X and Y increase. This is expected, as for higher values of X and Y less loops are executed. For the combinations of values that yielded good F1 values (above 0.95) the average execution times were between two and two and a half minutes.

The second set of experiments using simulated logs focused on the iteration size. Using a fixed timeframe size, selected as the optimal X value found in the first set of experiments (X = 0.5), we tested different iteration sizes, calculated as follows:

$$Iteration\ size = Y * (Mean\ time\ between\ cases)$$

Average precision, recall, and F1 values that were obtained, as well as the corresponding execution times are presented in Table 3.

Table 3. Precision, recall, F1, and execution times for different iteration sizes with a fixed timeframe size

	Precision			Recall			Average F1	Average Execution Time (Minutes)
Y	Normal	Account Executives	Peak Day	Normal	Account Executives	Peak Day		
0.1	0.948	0.907	0.903	0.997	0.995	0.994	0.956	03:34
0.3	0.948	0.914	0.916	0.934	0.941	0.938	0.932	01:12
0.4	0.944	0.922	0.923	0.812	0.821	0.816	0.869	00:54
0.5	0.935	0.928	0.931	0.674	0.669	0.672	0.780	00:44

In these experiments, expectedly the recall values for all logs decrease as the iteration size increases, as this means that the search along the log is less thorough. For precision, on the other hand, we observed different trends for the different logs. For the normally behaving logs, precision slightly decreased alongside the recall as the iteration size increased. For the two specific scenario logs, where the likelihood of false positive was higher as a starting point, precision increased with the increase of the iteration size. This can be explained by the reduction of false positive detection. Since the effect of changing the iteration size was stronger for recall than for precision, the F1 values decrease when the iteration size increases.

Similarly to the measurement of execution times for the first set of experiments, we measured execution times and checked their dependency on the iteration size. The results show that the execution time strongly decreases when the iteration size increases.

4.2 Experimenting with a Real-Life Event Log

The suggested approach was further evaluated by application to a real-life log of a purchase order handling process in a large company operating from the Netherlands, in the area of coating and paints [11], which served for the 2019 BPI challenge. The log contains four types of flows, each having different activities and business rules regarding the goods receipt messages and invoices. We focused on the '3-way matching, invoice after goods receipt' flow, which can be filtered from the whole log by the Item Category attribute. This category contains 15182 cases and 319,233 events, 38 different activities performed by 262 different users.

In order to identify the splitting decision rule (a threshold over a data item), we relied on the BPI challenge report [12], which indicated that cases whose Cumulative Net Worth is over 100,000€ take longer than cases below that sum. We filtered the log using DISCO and compared the mean case durations for cases where Cumulative Net Worth was less than 100,000€ (mean case duration of 71.7 days) with those where Cumulative Net Worth was over 100,000€ (mean case duration of 100.1 days). As there was a substantial difference in the mean case durations, we suggest that this might be the basis of a value function, which motivates process participants to split purchase orders: an employee might prefer creating several 'fast' purchase orders than one 'slow' order.

The uniting attribute for the log was decided based on domain knowledge provided by the process owners [13]. In the described process, each purchase order is linked to a vendor. Thus, the Vendor attribute was the uniting attribute used by the detection algorithm. The input parameters of *timeframe_size* and *iteration_size* were set to 0.5 * (mean time between cases) and 0.1 * (mean time between cases), respectively, as these were the optimal values found using the simulated logs. The mean time between cases in the event log was approximately 34 min.

Using the described inputs, 123 occurrences of the split cases behavior were observed in the log, where each occurrence is a set of cases which are suspected as split cases. A total of 513 cases were marked by the method, out of 15182 cases in the event log. The execution time for this log was 15:13 min. An example of a set of cases indicated by the method consists of three cases that were all created by user_091 in a period of 9 min, and are related to vendorID_0183. On average, the Cumulative Net Worth of these cases is 37,000€, thus for the concealed case this would have indeed exceeded 100,000€, which is the assumed threshold. Note that each line in the event log represents the purchase of one item type. Yet, each of the cases in the detected set holds the same item, which would be the ordered item in a concealed case.

To qualitatively assess whether the split cases behavior indeed occurs in this event log, we manually examined the detected cases. It appears that there are six cases created by user_091, related to vendorID_0183 and same item type, all created on January 2nd 2018, from 14:34 to 16:15. The six cases are presented in Table 4. Note that additional cases created by user_091 appear in the log, but not in time proximity to the ones in this set. This can indeed indicate that the detected cases are split. The three cases found by the method appear in the last three rows of Table 4. As the timeframe size we used was 17 min, these three cases were detected, while the others were not. Had we used a larger timeframe size, at least some of the other cases could be detected as well, but the risk of false positives would increase as well.

Table 4. Cases suspected as split in a real-life event log

Case ID	Resource	Complete Timestamp	Item	Purchasing Document	Vendor	Cumulative net worth (EUR)
4507000378_00010	user_091	1/2/2018 14:34	10	4507000378	vendorID_0183	37169
4507000417_00010	user_091	1/2/2018 15:51	10	4507000417	vendorID_0183	35832
4507000418_00010	user_091	1/2/2018 15:54	10	4507000418	vendorID_0183	38160
4507000421_00010	user_091	1/2/2018 16:06	10	4507000421	vendorID_0183	33058
4507000422_00010	user_091	1/2/2018 16:11	10	4507000422	vendorID_0183	33740
4507000425_00010	user_091	1/2/2018 16:15	10	4507000425	vendorID_0183	48054

In summary, experimenting with this event log has shown that our approach is capable of handling the scale and complexity of real-life event logs and detecting split-case behavior even when such is not known in advance to exist. As ground truth is not available for real-life logs, the detected sets of cases are merely suspected as split cases. When such would be detected in a real organization, additional investigations on site would be needed for validating this suspicion. Yet, the indicated cases form a starting point for such investigations.

5 Related Work

Up until now, the main focus of process mining techniques has been on examining intra-case behaviors – considering each case in the event log separately. Recently, research addressing inter-case behaviors has emerged, and approaches have been proposed in this direction. For example, a method of discovering instance-spanning constraints (ISC) from an event log was proposed by [6]. ISC addresses constraints which take into account more than one case, such as “a customer cannot submit more than 10 requests per week”. This constraint necessitates examination of all cases related to a specific customer in a certain week; detection of non-compliance to this constraint is not possible by examining each case separately. Examples of instance-spanning constraints have been collected [14], and methods of detecting these constraints automatically from event logs are proposed [6]. However, ISC discovery methods rely on the existence of explicit inter-case constraints in the process. The split cases workaround does not violate any explicit constraint, thus it cannot be unveiled using ISC-related methods.

Another inter-case behavior that has been studied is batch processing. A batch is a cluster of activity instances, which are executed in a synchronized manner, possibly according to pre-defined rules [15]. The waiting times of activities in a cluster may vary, as some activities might wait longer than others for execution. Several studies have suggested approaches for detecting batching behavior in event logs [16–18]. All suggested approaches consider attributes and relations among activities of multiple cases in the log and use them specifically for detecting batch behaviors.

Prediction of process properties is another field in which the usage of inter-case features has emerged [19]. Up until recently, prediction of properties such as the remaining time of a case was based solely on intra-case data [20]. However, in some processes and systems the predicted outcome may depend on the properties of all cases that are being executed concurrently. For example, in processes which include batching behavior, a remaining time prediction algorithm which addresses the batched cases and their time properties [21] would yield more accurate results. Although such prediction approaches are based on inter-case properties, explicit inter-case behaviors are not specified, but are rather assumed as contributing to prediction accuracy.

Considering workarounds in business processes, detection methods for various kinds of behaviors in an event log have been proposed. [22] examined six generic types of workarounds described in [2], and successfully detected four of them in event logs. The split cases workaround was one of the two kinds that could not be detected. This gap is addressed by our approach.

Another relevant stream of work relates to fraud detection in business processes via process mining. [23], for example, suggests a detection method of fraud in procurement processes using the heuristic miner [24]. Other process discovery algorithms, such as the fuzzy miner, are also used for fraud detection [25], where the examined fraud type involves deviations from the defined process model, unveiled by mining the process model and applying conformance checking techniques to the mined model and the event log. As discussed in this paper, the split cases behavior cannot be detected this way, as all the split cases may be conformant with the expected process model.

6 Conclusions

The split cases workaround is commonly observed in practice and might bear harmful consequences. While the importance of detecting such behavior is recognized, existing conformance checking techniques are not geared for accomplishing this task. We have proposed in this paper an approach that can accurately detect split cases in event logs. As a basis for the proposed approach, we have analyzed and formalized the motivation for this workaround and its manifestation in an event log. This analysis forms a main conceptual contribution of the paper.

We evaluated the approach both with simulated and with real life logs. The experiments performed with simulated logs assessed the dependency of the results on the input parameters as functions of the mean time between case creations in the log. We checked normal as well as specifically "difficult" scenarios and showed that the proposed method is capable of obtaining good results for both. We also checked the dependency of execution times on the input parameters. The experiment performed with a publicly available real-life event log demonstrated the applicability of the approach and its capability to detect suspected splitting of cases even in an unfamiliar log.

Future research will investigate more complicated scenarios and test the approach in a real organization, where suspected split cases can be further investigated.

Acknowledgement. The research was supported by the Israel Science Foundation under grant agreement 669/17.

References

1. Alter, S.: Theory of Workarounds. Commun. Assoc. Inf. Syst. **34**, 1041–1066 (2014)
2. Outmazgin, N.: Exploring workaround situations in business processes. In: La. Rosa, M., Soffer, P. (eds.) BPM 2012. LNBIP, vol. 132, pp. 426–437. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36285-9_45
3. Hepzibah, P.: $2.1 billion arms procurement fraud in Nigeria: its impact on national security, peace and sustainable economic development in Jonathan's administration 2011–2015. Int. J. Arts Sci. **9**(2), 225–248 (2016)
4. Outmazgin, N., Soffer, P.: Business process workarounds: what can and cannot be detected by process mining. In: Nurcan, S., et al. (eds.) BPMDS/EMMSAD -2013. LNBIP, vol. 147, pp. 48–62. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38484-4_5
5. Leitner, M., Mangler, J., Rinderle-Ma, S.: Definition and enactment of instance-spanning process constraints. In: Sean Wang, X., Cruz, I., Delis, A., Huang, G. (eds.) WISE 2012. LNCS, vol. 7651, pp. 652–658. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35063-4_49
6. Winter, K., Stertz, F., Rinderle-Ma, S.: Discovering instance and process spanning constraints from process execution logs. Inf. Syst. **89**, 101484 (2020)
7. Van Der Aalst, W.M.P.: Process mining: overview and opportunities. ACM Trans. Manag. Inf. Syst. (TMIS) **3**(2), 1–17 (2012)
8. De Leoni, M., Munoz-Gama, J., Carmona, J., van der Aalst, W.M.P.: Decomposing conformance checking on Petri nets with data. BPM center report BPM-14-06 (2014)

9. De Leoni, M., van der Aalst, W.M.P.: Data-aware process mining: discovering decisions in processes using alignments. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, pp. 1454–1461 (2013)
10. Powers, D.M.: Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061 (2020)
11. van Dongen, B.F.: Dataset BPI Challenge 2019. 4TU. Centre for Research Data. https://doi.org/10.4121/uuid:d06aff4b-79f0-45e6-8ec8-e19730c248f1
12. Kim, J., Ko, J., Lee, S.: Business Process Intelligence Challenge 2019: Process discovery and deviation analysis of purchase order handling process (2019)
13. BPI Challenge 2019. https://icpmconference.org/2019/icpm-2019/contests-challenges/bpi-challenge-2019
14. Rinderle-Ma, S., Gall, M., Fdhila, W., Mangler, J., Indiono, C.: Collecting examples for instance-spanning constraints. arXiv preprint arXiv:1603.01523 (2016)
15. Pufahl, L., Weske, M.: Batch activities in process modeling and execution. In: Basu, S., Pautasso, C., Zhang, L., Fu, X. (eds.) ICSOC 2013. LNCS, vol. 8274, pp. 283–297. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-45005-1_20
16. Martin, N., Pufahl, L., Mannhardt, F.: Detection of batch activities from event logs. Inf. Syst. **95**, 101642 (2021)
17. Wen, Y., Chen, Z., Liu, J., Chen, J.: Mining batch processing workflow models from event logs. Concurr. Comput. Pract. Exp. **25**(13), 1928–1942 (2013)
18. Martin, N., Solti, A., Mendling, J., Depaire, B., Caris, A.: Mining batch activation rules from event logs. IEEE Trans. Serv. Comput. (2019). https://doi.org/10.1109/TSC.2019.2912163
19. Senderovich, A., Di Francescomarino, C., Ghidini, C., Jorbina, K., Maggi, F.M.: Intra and inter-case features in predictive process monitoring: a tale of two dimensions. In: Carmona, J., Engels, G., Kumar, A. (eds.) BPM 2017. LNCS, vol. 10445, pp. 306–323. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65000-5_18
20. van der Aalst, W.M.P., Schonenberg, M.H., Song, M.: Time prediction based on process mining. Inf. Syst. **36**(2), 450–475 (2011)
21. Klijn, E.L., Fahland, D.: Identifying and Reducing Errors in Remaining Time Prediction due to Inter-Case Dynamics. In: 2020 2nd International Conference on Process Mining (ICPM), pp. 25–32. (2020).
22. Outmazgin, N., Soffer, P.: A process mining-based analysis of business process work-arounds. Softw. Syst. Model. **15**(2), 309–323 (2014). https://doi.org/10.1007/s10270-014-0420-6
23. Rahmawati, D., Yaqin, M.A., Sarno, R.: Fraud detection on event logs of goods and services procurement business process using Heuristics Miner algorithm. In: 2016 International Conference on Information & Communication Technology and Systems (ICTS), pp. 249–254 (2016)
24. Weijters, A.J.M.M., van Der Aalst, W.M., De Medeiros, A.A.: Process mining with the heuristics miner-algorithm. Technical report, Technische Universiteit Eindhoven, WP, vol. 166, pp. 1–34 (2006)
25. Sarno, R., Dewandono, R.D., Ahmad, T., Naufal, M.F., Sinaga, F.: Hybrid association rule learning and process mining for fraud detection. IAENG Int. J. Comput. Sci. **42**(2) (2015)

Towards a Content-Based Process Mining Approach in Personal Services

Felix Holz(✉), Birger Lantow, and Michael Fellmann

University of Rostock, Albert-Einstein-Straße 22, 18051 Rostock, Germany
{felix.holz,birger.lantow,michael.fellmann}@uni-rostock.de

Abstract. Personal Services are directed at enhancing a client's mind or body and often require flexibility for reaching service delivery goals. While often unstructured text data is created as part of this work e.g. due to documentation duties, existing information systems and process analysis methods do not leverage this process-related knowledge asset. As a technique for automatically generating process models, Process Mining mainly considers activities to visualize and analyze processes. Especially in Personal Services, the activities by itself provide too little information for a helpful model. The available unstructured text data enable refining process models by using the hidden process knowledge. Topic Modeling techniques can be used to extract the knowledge. This paper provides an approach for generating content-based process models in Personal Services. First, the text data is structured and summarized into labels by using the Labeled Latent Dirichlet Allocation (L-LDA). Thereupon, Process Mining techniques generate a process model showing the order of handled contents represented by the labels. This approach is presented and evaluated by a case study in a social care company. Problems have been identified in terms of perceived correctness and usability, while the automatic labeling process showed good results. In general, a good utility can be assumed if the identified problems are going to be resolved.

Keywords: Personal services · Topic modeling · L-LDA · Process mining · Case study

1 Introduction

Digitization is a defining subject that allows companies to increase effectiveness and efficiency through various means. For example, IT-driven process support enables the collection of key performance indicators, process streamlining, resource optimization, and the performance analysis of processes. In the context of personal services, such digitization efforts are not as advanced as they could be. Personal services are distinguished by the fact that service delivery is directed at the mind or body of people in order to achieve improvement [10]. Accordingly, the activities are characterized by a high degree of interaction with the so-called clients [11]. Especially in social services like family care, the improvement of a

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 62–77, 2021.
https://doi.org/10.1007/978-3-030-79186-5_5

client's situation can take a long period of time; the nature of the work also requires a high degree of flexibility from the employees in order to be able to adapt to changing circumstances [5]. With this comes the need for supporting techniques to accommodate this flexibility. Several attempts have been made to promote automatic process creation and control in personal services in the past. The known technique of process mining [18] for an automatic generation or adaptation of process models proved to be a practical means of structuring processes by using existing data from information systems without requiring much additional effort on the employees. However, since the number of different activities is rather small (e.g., "Create appointment", "Prepare session with client", "Talk with client"), the corresponding models also tend to be less meaningful [11]. In order to refine the activities, reports and documentation created during the casework with the clients can be considered. Successes or failures, changes in the client's situation or the activities' effectiveness are written down in them. Therefore, the documentation of client cases contain significantly more process knowledge than the activities. However, they are highly unstructured data like plain texts. Hence, several topic modeling techniques were applied in [12] to extract the knowledge or summarize the reports. As a result of this work, a fully automated technique, like Latent Dirichlet Allocation (LDA) [7], could discover topics without additional effort burdening the employees. Nevertheless, the topics were less accurate, hardly distinguishable, and poorly understandable. On the other hand, a dictionary-based approach requires more manual preparatory work on large scales, but the topics created and filled with representative words by domain experts outperformed the former ones. Therefore, a semi-automated technique considering the expert knowledge seems to be a promising approach for structuring the texts. Labelled LDA (L-LDA) according to Ramage et al. [15] takes domain expert's knowledge into account by labeling documents. While labels provide a more targeted way to describe the topics, the advantages of both manual and automatic topic modeling techniques are combined.

Accordingly, this paper is aimed at providing an approach for generating content-based process models. It utilizes L-LDA to summarize the content of the documentation into labels. These labels are converted into a process model using process mining techniques to represent a sequence of handled contents, instead of activities. It should be noted that process mining is only applied and not extended. While the approach is intended for Personal Services, it was evaluated through a case study in a real social welfare company. In Sect. 2, the necessary basics are explained, while Sect. 3 explains the approach for content-based process mining. The conducted case study is illustrated in Sect. 4 alongside an evaluation of the approach. The results and limitations of the approach are discussed in Sect. 5. Finally, a summary and outlook follow in Sect. 6.

2 Theoretical Background

In this section, the characteristics of personal services will be briefly presented. For topic modeling, the focus lies on the most basic automatic technique, Latent

Dirichlet Allocation (LDA). Based on this, the extension Labeled LDA (L-LDA) will be explained as well.

2.1 Personal Services

Services essentially differ from conventional goods in that they are more immaterial and involve the customer or consumer more closely in the service creation process. Service recipients are therefore referred to as co-producers, as a service cannot be created without the customer's involvement. Furthermore, the lack of material components means that services cannot be stored [20]. The interactivity between the service provider and the service recipient is particularly for personal services an essential part. The service provision focuses on changing the condition of a person. Thereby physical (e.g., by physiotherapy) or mental aspects, like the personality (e.g., by psychotherapy) of the client, are influenced with the aim of a positive development [10]. The nature of the interaction can be multifaceted because life situations for the clients may change; thus, much flexibility is required within the service delivery process as it is characterized by uncertainty, weak structure, and low predictability [5,12].

Individual clients are considered as cases, and the respective employees as caseworkers. These cases can span long periods of time to achieve a predefined goal [5]. According to [11], the number of clearly identifiable different activities is rather small and typically comprise more coarse-grained activities such as "Talk with the client". These activities can in turn be composed of numerous sub-tasks that occur in arbitrary order (e.g. ask a question, talk, take note, give advise). Due to this, generating process models based on the order of activities only does not lead to insightful and informative process models. To monitor and document the clients development and the degree of goal achievement over a longer period, the caseworkers produce reports (or documentation). These contain more detailed information about the activities carried out, their effectiveness, possible changes in situations, or successes and failures. Accordingly, more knowledge about the process can be found in the reports than in the activities.

2.2 Topic Modeling

In general, topic modeling is a way of systematizing underlying texts into their topic portions, mainly to make large amounts of information easier to structure and search through. Words are the defining entities for determining Topics from discrete data. While there are other use cases for topic modeling, the focus here is on processing plain text data. A simple basic form for a probabilistic topic model is *Latent Dirichlet Allocation* (LDA), according to Blei et al. [7, 8]. LDA is a generative probabilistic model characterized by random selection from a set of different distributions and discovers the hidden topic structure of a set of documents. The *generative aspect* assumes to generate a document by itself. A document is an entity consisting of words from a vocabulary. The corpus is composed of multiple documents and depicts the whole text data to be structured. Additionally, documents can contain several topics in different proportions. These topics are determined by words as well. Formally, a topic is defined "to be a distribution over a fixed vocabulary" [7].

While generating a document, LDA relies on conditional probabilities (e.g., $p(a|b)$). For deciding which word is written down in a document, the topic proportions θ for the document and the word distribution β for the respective topic are used. The former specifies which topic proportions should be present in the document, or how probable certain topics are for this document (e.g., 20% sports and 80% biology). The latter describes which words are most likely to describe this topic (e.g. "tennis" and "running" for "sports"). Both are multinomial probability distributions, from which the items are randomly selected. First a topic from θ is randomly chosen, and for this topic, a word from β will be derived. The higher the probability of a topic or a word, the higher is the chance for being randomly selected. The distribution themselves are randomly selected from Dirichlet distributions. They are determined by the mandatory input parameters α and η. Hence, the parameters must be given alongside the number of topics. Furthermore, LDA assumes a document as a "bag-of-words." Accordingly, the position of the words in the text does not matter at all. Likewise, it works regardless of the language used, as only the vocabulary and not the grammar is crucial.

In a real use case, the documents exist already; therefore, the words are observable. From that, the hidden (latent) variables of the topic proportions per document θ and the word distribution per topic β have to be determined. This describes the *inference* - the computation of the A-Posterior probability [7]. For computing the hidden variables, the generative process is rearranged so that their probabilities are conditioned by the observable variables. However, the computation involves a too large state-space; therefore, methods of approximation have to be used. Blei et al. [7] refers to sampling-based and variational-based methods. The most prominent representative of the sampling procedures lies in the Gibbs Sampling: A Markov chain is constructed from different random samples to provide an iterative approximate value [17]. The variational inference provides a deterministic alternative, which utilizes parameters and representations.

The *Labelled Latent Dirichlet Allocation* (L-LDA) by Ramage et al. [15] is an extension of the basic model which acts semi-supervised. In contrast to LDA, annotated data are necessary. The documents have to be tagged (or labeled) for "defining a one-to-one correspondence between LDA's latent topics and user tags." [15] Therefore, topics are represented by explicitly named labels in a more targeted way. Whereas LDA calculates the topic proportions of the documents and the word distributions of the topics randomly according to the texts' behavior without being influenced, L-LDA "learns" the distribution of highly probable words for each given label. While the topic assignments are already given with the document, the actual topic proportions (or probabilities) θ will be computed as well. LDA depicts the topics through their most probable words, L-LDA adds headings to the topics. Thus, the results may be more readable and interpretable [15]. The data can be more directed.

Formally, the label set Λ is added to the original LDA model as an additional observable variable. Thus, the topic proportions θ are not only dependent on the Dirichlet parameter α, but also on the label set Λ. By considering the new variable, the flow of both the generative process and the inference of L-LDA is somewhat similar to LDA.

3 Process of Content-Based Process Mining

With the goal of refining the collection and representation of automatically generated process models in the context of personal services, an approach is provided. It will be further described within this section. It is a procedure that combines process mining concepts with a topic modeling technique for structuring and preparing text data.

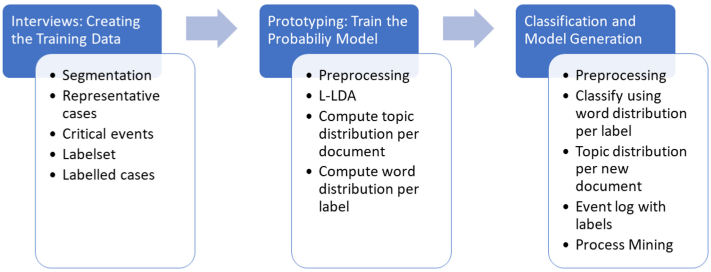


Fig. 1. Singular phases for creating a content based process model in personal services by utilizing topic modeling and process mining.

Figure 1 shows the concrete procedure divided into the individual phases that should be carried out for achieving the goal as mentioned above. Each phase produces interim results and uses results from previous steps as prerequisites. Different techniques were used to achieve them, which will be presented and discussed in the following sections.

As summarized in the figure, an initial workshop is followed by expert interviews, in which representative client case histories are discussed with the corresponding caseworkers and tagged with critical events. These critical events are transformed into labels, which act as representatives of the content or the documentations' topic. The labeled case histories are used as training data for the L-LDA algorithm for determining the topic distribution for each document and the word distribution for each label. The latter indicates how much the words can be assigned to the subject (more precise: how probable is the assignment of a word to a topic) and is used for automatically classify the remaining cases. Once all reports are labeled alongside the time dimension, the process mining step can start. As input for process mining, more specifically process discovery, an event log is used, containing all cases with their corresponding labels.

The output is a specific kind of process model, which describes an overall sequence of content handled during the cases instead of performed activities. The individual steps respectively phases will be described in the following subsections.

3.1 Interviews: Creating the Training Data

A workshop should be held at the beginning in order to narrow down the problem domain more precisely and to familiarize the employees of the cooperating company with the objective. The workshop is aimed to qualitatively identify additional context variables, like functional requirements on the resulting model or the notation at the end. However, relevant variables to be collected in this workshop are, firstly, *representative cases* and, secondly, *segments* into which the cases can be divided. The former will be used as a basis for the interviews and later as training data. Accordingly, the cases should contain as many different topics and tasks as possible. They should represent the "general" workflow of a social worker so that these findings can be transferred to other cases. The workshop's second important result lies in the segmentation of case histories into specific time windows applicable to every case. It is crucial for the preparation of process mining not to look at each documentation entry individually but join them together. This is especially beneficial for summarizing short reports. In topic modeling, documents with a larger amount of text promise more precise topic modeling results [15].

After the workshop, all representative cases should be individually discussed with the corresponding caseworker in an *interview*. The interviewees act as domain experts and designate critical events [9] that occurred during the case progression. They have a free choice for assigning one or more events to segments; however, they should cover special incidents that happened and that had to be dealt with, or concerned the client within the time window. These critical events are being formed into superordinate topics, respectively labels. The labels can cover the incidents, issues, and themes, which occurred during the work with the respective clients. Additionally, collecting case-independent critical events during the interviews is advantageous as well. As a result of the interviews, critical events are gathered and transferred into a label set, and the representative cases are labeled segment by segment. Each segment is therefore considered as a document forming the training data.

3.2 Prototyping: Train the Probability Model

The prototype's first sub-goal is to calculate the word distribution per topic β derived from the topic distributions per document θ. Therefore, the interview results must first be read in as training data. Before the actual inference can be applied, the text should be cleaned up.

Several *preprocessing* techniques can be used for this purpose since not every word in natural language usage has a semantically high meaning. For most text analysis and processing techniques, words with semantically high meaning act as better descriptors and more suitable representatives of topics or the text as a whole [6,19]. The most prominent preprocessing techniques are stop word reduction and stemming. Additionally, all letters in the document should be overwritten in lower case, and names, numbers, and special characters (like or ",") should be removed since they are not considered as meaningful text parts. According to Blei and McAuliffe [6], the removal of highly frequent and rare words, as well as the calculation of tf-idf-score for every word can also reduce unnecessary parts of the vocabulary.

After the step of preprocessing, the *L-LDA-algorithm* calculates the distributions θ and β by using the training data as corpus and the labeled case segments as documents. The input parameters for the Dirichlet distributions α and η must be given as well.

3.3 Classify the Documents and Generating the Models

In this step, additional cases are now to be classified that have not been manually labeled before. The topics of the remaining cases are determined based on the former calculated word distribution per topic β. For consistent wording, the documents to be classified should be preprocessed the same way the training data were. Ramage et al. [15] describe the *classification* step as "more complex" and "not straight-forward" because "it involves assessing all label assignments and returning the assignment that has the highest posterior probability." Therefore, the simplified L-LDA-Classifier (sL-LDA-C) according to Bai et al. [4] could be used. It describes a very simple method by counting the known topic's most probable words in the documents to be labeled; more complex alternatives can certainly lead to more accurate results but require more computational effort.

The completion of the classification step requires for the documents and corresponding classified labels to be transferred to an event log in an appropriate format so that the process mining tool can interpret this and transfer it to a process model. While the .xes-standard is mainly used to store event logs for process mining purposes, .csv-files are suitable for the common tools as well. In order to use process mining, there are mandatory specifications for generating a model, alongside some optional ones for displaying additional information (like roles). Table 1 shows essential data to be included, and which functionality they take. Furthermore, optional data can be given as well, for leveraging the model with additional information. Therefore, timestamps could be used for computing elapsed time, or, resources for displaying responsibilities of certain roles. Naturally, these information need to be given in the existing data. Placing the documents in the correct order is an essential step so that the miner works appropriately [18]. Also, preprocessing or filtering the event log is an option, e.g., for adding or reducing noise. Depending on the domain, the data, and the model's goal, a proper notation and mining algorithm must be chosen to display the required information.

Table 1. Mandatory data for process mining and the application after topic modeling.

Required for process mining	Generated with topic modeling	Remarks
Trace ID	Case ID	Every document of a case instance is seen as a contiguous entity. The event sequences of singular cases will be joined into a fitting model
Activity ID	Document ID	Every document of a case should have an unique identifier
Activity	Label	These will be linked with each other and presented in the model according to the found patterns

4 Case Study

The given approach was validated in a single in-depth case study in cooperation with a corresponding company in personal services. The "Wohnen ohne Barrieren GmbH" (WoB) is active in social welfare in the area of Rostock, Germany, and focuses itself on ambulatory assisted living via individual support of clients. In this chapter, remarks and interim results of the approach's phases will be presented alongside the final result. Finally, an evaluation of the approach is provided by judging the results due to expert interviews. The used data is produced by each caseworker individually and contains a reflection of the work with the client. Formal requirements regarding text length or writing style do not exist. The reports are loosely categorized into four superordinate topics. Furthermore, they are neither additionally annotated nor structured in data fields.

4.1 Execution of the Phases

In the first phases, interaction with the company's employees was the central aspect. Thus, in the **workshop** the participants estimated a window of two weeks as a segment in which reports can best be summarized. Furthermore, ten representative cases and their respective caseworkers were selected and then subjected to **interviews**. The cases' rough division into two weeks per segment forces the interviewees only to list the most important events. The collected critical events formed 14 topics, respectively labels. It is to be noted that certain events (and henceforth labels) occurred significantly more often than others. Whereas topics like "Residing situation," "Sickness," or "Work/Day structure" frequently appeared, other labels like "Violence/Delicts" or "Addiction" occurred seldom. As a result of the interviews, all labeled segments resulting in 169 multi-labeled documents, which can be used as training data for the L-LDA-algorithm.

The data were **preprocessed** through stop word reduction, utilizing a stop-word list from the natural language toolkit (nltk)[1], and stemming, using the snowball stemmer[2]. Furthermore, names were removed from the documentation by a name list merged from the nltk and a list of the client names[3] as well as special and singular characters. After the preprocessing step, 3873 unique words remain. This is still very much in relation to the number of documents. However, other possibilities for preprocessing have filtered out highly representative words for certain topics. So a need for specific techniques arises to cope with problems in the documentation like the usage of very natural language and individual writing styles. Since the texts are written in German, only certain preprocessing techniques can be applied to them. The reports are not reviewed as well; therefore, spelling mistakes are present. The timestamps have been adopted from the documentation, so that the one of the first report of a segment is used for the whole segment. For the **application of the L-LDA-algorithm** itself, there are some open-source implementations given, which differ from each other, e.g., regarding input formats. In this case study, the implementation from Shuyo [16][4] was used; it provides a code as simple as possible, and the input is read in as a whole text file so that the documents were transformed in this format. Alternatives like "tomotopy"[5] or the implementation from JoeZHJ[6] could be used as well. Every implementation uses the (collapsed) Gibbs sampling method [17] for approximate the A-Posterior Probability. To measure and estimate the effectiveness of preprocessing techniques, the implementation offers the computation of perplexity [16].

The chosen L-LDA-implementation requires two input variables α and η to define the Dirichlet distributions and a variable I, which determines the number of iterations used for Gibbs sampling. For this, the following default values were used: $\alpha = 0.1$, $\eta = 0.001$ and $I = 100$. Adjusting these values can significantly change the result. For this matter, underlying data and context must be considered in more detail.

L-LDA's application in the case study showed that - despite the preprocessing steps - many words with a general meaning are still listed as very likely. Some labels are underrepresented by the training data, this is noticeable because there were also many too general words here. However, an interesting observation is that topics perform better when assigned to a document alone. The documents are primarily multi-labelled; however, a segment is described by just a singular topic in some instances. For example, there were only four documents for the topic "Violence/Delicts" and nine documents for the topic "Offices/Authorities." Nevertheless, the former's words were rated better by the experts than the words

[1] https://www.nltk.org/.

[2] https://www.kite.com/python/docs/nltk.SnowballStemmer.

[3] Since the identical names occur frequently in the training data, the model would perceive these names as representative words for certain topics.

[4] https://github.com/shuyo/iir/blob/master/lda/llda.py.

[5] https://bab2min.github.io/tomotopy/v0.9.0/en/examples; https://github.com/bab2min/tomotopy.

[6] https://github.com/JoeZJH/Labeled-LDA-Python.

for the latter. One reason for this could be that "Offices/Authorities" was consistently listed in combination with other labels. On the other hand, it can be seen that especially the frequently occurring topics contain more words that can be considered representative.

The **classification** through sL-LDA-C by Bai et al. [4] was held relatively simple: the topic's 20% most probable words from the training set will be matched with the occurring words of the documents to be labeled. However, disadvantages lies in the missing coverage for multi-label classification and the disregard of unknown words. Therefore a threshold was set as "if more than 20 words occur in a new document, which are probable for a label, assign this label to the document." This fixed values (depending on the length of the vocabulary) provided better results than a threshold, which uses the percentages of words in the text. Since the documents' text sizes vary significantly, the latter approach would classify too many labels in short texts, and the former will classify more labels at long texts. Intuitively, the larger the text, the more topics it contains. For the classification, all data from the last two accounting periods were segmented and preprocessed the same way as the training data. It resulted in 2108 labeled documents.

4.2 Generating the Model

The model was created using the Process Mining Toolkit (ProM) [2]. It offers the generation of different types of models utilizing different variants of mining algorithms and plugins. Specifically, in this case study, a **dependency graph** was generated with the idHM plugin from Mannhardt et al. [13] using the classified case data. In this graph, start and end node are displayed alongside dependencies between labels on directed edges [1, p. 204]. The plugin offers thresholds for displaying only high enough dependency values and frequently occurring relations. For this purpose, the thresholds of 0.1 were chosen for both *dependency* and *frequency* to include more topics and depict relationships that occur less frequently. The model's goal was to be understandable for non-method experts and should therefore not contain too many connections [3,14]. It should also yield enough information and potential for assertions for the domain experts.

Figure 2 shows the generated model. There can be seen that the highest dependencies are between the same labels. So the data predominantly contains cases with mostly coherent topics. The connections of start and end nodes with the most labels do not imply a definite process or even control flow if only the high dependencies are considered; the dependencies between different labels are comparatively very low. The domain experts described half the statements of the model as correct. The depicted process is therefore only conditionally meaningful. Possible reasons for this will be discussed in Sect. 5. However, the relatively high dependency between "Relationship" and "residing situation" correctly represents the reality and implies a change in the way of living after new or broken relationships. The option to filter the log before was not used for not cutting potential relevant data away. Also, the plugin did not process the multi-labeled documents the anticipated way, so that the data had to be reduced to one label per document. Intuitively, multiple events on the same timestamp should be depicted as parallel, but they are interpreted as sequential. Furthermore, other notations, like BPMN, EPK, or Petri Nets, could not deliver more meaningful results. The respective models lack either in terms of expressiveness or understandability.

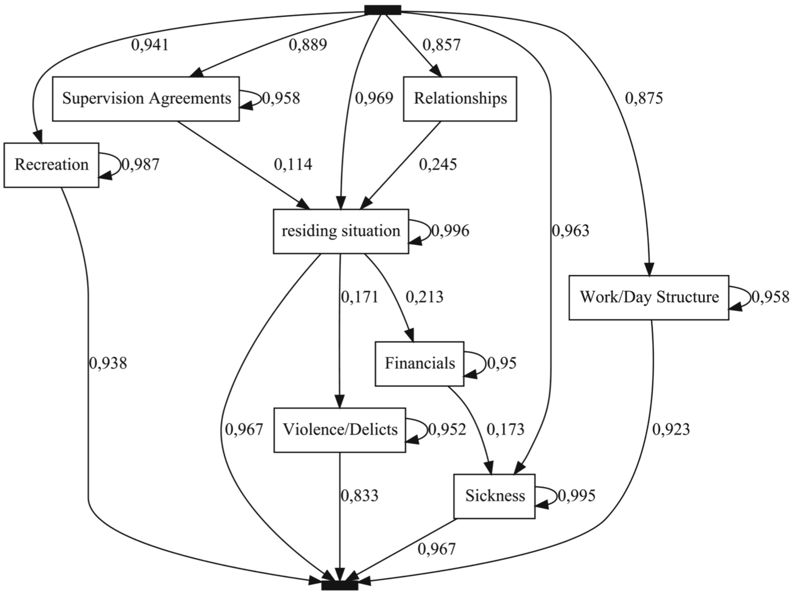


Fig. 2. Computed dependency graph with the iDHM-ProM-Plugin.

4.3 Evaluation

At the end of the case study, the approach was evaluated by interviewing three domain experts who already participated in the former interviews. The evaluation contains an assessment of the interim results and the final model. Therefore, (1) the labels were rated through fitness of the computed most probable words to the corresponding label, (2) the classification quality by judging automatically labeled cases, and (3) the model by simple quantitative and qualitative questions regarding understandability, correctness, and usefulness. The quantifiable results are shown in Table 2.

The first aspect was done by randomly sampling four topics whose 20 most probable words were assessed from the experts' point of view; as stated in the table, more words were rated as inappropriate than appropriate. The unfitting words were described in most cases as too general. Not every word distribution per label was rated; nevertheless, the observation is that labels with more training documents perform better.

However, the evaluation's core lies in the assessment of the classification quality. Each subject was presented with a case that was automatically labeled segment by segment. For each segment, the interviewee assessed whether the listed labels matched the contents of the respective documentation and whether other labels from the label set would apply that were not classified. This forms a test set with contingency values: true positives (tp) are correctly classified and assigned labels, false positives (fp) are labels that were incorrectly classified. On the other hand, false negatives (fn) are labels that should have been listed but were not, and true negatives (tn) are the labels that were actually not classified and would not have fit. These values can be used to calculate the evaluation metrics for determining how accurate the recognized labels are (Precision) and how well the topics were selected from the topic catalogue (Recall). The F1-Measure determines the harmonic mean of these values. Accuracy and Error can be computed as well, using the population (n) of all values in the segment [21]. As shown in the table, a moderately good result can be estimated from the performance measures since the values can take a score between 0 and 1.

Table 2. Quantitative scoring of the final result evaluation by the interviewees.

Measure	Rating	Elicitation
Words in labels		
Appropriate	30	Perceived assessment
Inappropriate	48	Perceived assessment
No statement	2	Perceived assessment
Classification		
Precision (P)	0,60	$tp/(tp+fp)$
Recall (R)	0,46	$tp/(tp+fn)$
F_1-score	0,52	$2*P*R/(P+R)$
Accuracy	0,75	$(tp+tn)/n$
Error	0,25	$(fp+fn)/n$
Model evaluation		
Understandability	5	Perceived assessment
Correctness	3	Perceived assessment
Usefulness	1,6	Perceived assessment

The model evaluation was conducted with quantifiable questions aiming at the three aspects of understandability, correctness, and usefulness and was raised and mapped onto a five-digit Likert scale. Since the number of interview partners was relatively low, the results are not accurate and can only depict tendentious statements. However, the interviewees described the model as entirely understandable and halfway correct. They disagreed on certain presented aspects, like the position of "Supervision Agreements" at the start of the process. Besides, the model's possible usage in a real case scenario is not imaginable at present, according to the domain experts. They stated, even if problems can be identified in this model, no appropriate action can be taken by the caseworker. The model is too general and too entangled for encouraging effective responses; no clear statements for everyday work can be derived from it, they said. Nevertheless, if the model were correct, they agreed to intensify the communication between other caregivers since the model shows "Supervision Agreements" at the top of the process.

5 Discussion

As seen in the case study's evaluation, the approach needs to be adjusted for leveraging the results and especially the usefulness. It is generally questionable whether the model in this context can be used as a driver or guide for caseworkers' actions; the analytic opportunities are also still relatively limited from the case worker's perspective. Even though an optimal result cannot be achieved due to the random nature of the topic modeling technique and thus classification, some limitations have emerged whose solution could significantly improve the result. Beginning with the training data, the vocabulary is considerably larger (3873 unique words) than the number of documents (169). For example, in the execution of the supervised LDA (sLDA) model [6], the training set was more extensive, and the number of documents clearly exceeded the vocabulary. To address this problem, either more interviews have to be conducted, or the text's preprocessing must be extended. The former is restricted due time and effort of the caseworkers, the latter through several techniques, which cuts down also representative words. Especially in social services, the data is poorly structured and inconsistent, which generally hinders automatic text analysis. The reports are written by different people and are not reviewed since they should be used for internal purposes and reflecting the work. Therefore, individual writing styles, word usages, and text sizes are pervasive throughout the data set. Incorrectly spelled words aggravate the performance of the text analysis because the algorithm considers them as unique words. Particularly minuscule text passages are problematic.

Another drawback lies in the missing support for multi-labeling. Bai et al. [4] do not offer a native way to classify multiple labels. Therefore, thresholds can be used, but the choice is highly dependent on how the underlying data is constituted. Furthermore, the used ProM plugins do not recognize simultaneous topics in a segment as parallel events; instead, they treat them as sequences. This distorts the model accordingly. Therefore the presented and evaluated model only covers single labeled cases, which is counteractive to the intuition for a segment having more than one topic most of the time. Additionally, the simple classification does not consider words unknown from the training set. Other classifiers may be aware of these but require more computational effort. Additionally, the resulting dependency graph can be altered or extended since patterns like parallelism or exclusivity are not displayed. Finding these patterns in the given data seems to necessitate a more suited or adapted mining algorithm. The approach could be more efficient and goal-oriented when the label set is created during the workshop and not after collecting critical events. This would save the conversion of the critical events and allow a more focused flow in the interviews by giving the subjects only a limited variety of labels to choose from for a segment.

6 Conclusion and Outlook

In this paper, an approach was provided and evaluated through a case study for generating content-based process models within the context of personal services. L-LDA was used as the topic modeling technique to condense the documentation into labels that represent the documentation's respective content. The labels were depicted in a model through process mining to show sequences of handled contents during client cases instead of activities. However, some limitations could be found; they affect the results throughout the approach, especially the model's perceived usefulness. To improve the combined method, it is essential to address the problems such as the size of the training data, more precise preprocessing techniques, better classification algorithms, and consideration of multi-labeled documents.

Although there were many obstacles, the combined procedure was able to generate results that tended to represent the data correctly. L-LDA has shown to be an enabler, especially at the studied company, where only four different superordinate topics are distinguished. The advantage of semi-supervision is to include expert knowledge; the effectiveness of other text analysis or topic modeling techniques would have to be investigated as part of the future work. While in current form, the generated models just partially provides relevant and usable information, the utility of the approach can be considerably improved in future iterations using different optimization options discussed previously.

Techniques such as clustering, named entity recognition, contingency analysis, or generation of association rules can also be considered as an extension of the method to achieve more accurate results and more usefulness for the domain experts. Other more refined topic modeling approaches or mining algorithms could be used as well. However, for applying the approach and changing its parts, decisions must be made, highly depending on the data's nature. The generalization range remains to be investigated as the approach was only conducted in a single entity within the field of social care enterprises. However, the method may be applied wherever unstructured text files are produced, i.e., mainly in personal services, but other areas should not be excluded. It should be noted that this technique is partially obsolete for files and texts that have already been manually labeled since the data do not need to be classified. Thus, process mining can be performed without L-LDA. Nevertheless, an underlying word distribution per topic can help caseworkers by giving recommendations for labels while preparing new reports. This combined approach shows the potential to create value via enrichted process documentations in organizations where large volumes of unstructured text data are produced during process execution (e.g. as part of documentation requirements).

References

1. Van der Aalst, W.: Data science in action. Process Mining, pp. 3–23. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4_1
2. Van der Aalst, W.M., van Dongen, B.F., Günther, C.W., Rozinat, A., Verbeek, E., Weijters, T.: ProM: the process mining toolkit. In: BPM (Demos), vol. 489, no. 31 (2009)
3. Aranda, J., Ernst, N., Horkoff, J., Easterbrook, S.: A framework for empirical evaluation of model comprehensibility. In: International Workshop on Modeling in Software Engineering (MISE 2007: ICSE Workshop 2007). IEEE (2007)
4. Bai, Y., Wang, J.: News classifications with labeled LDA. In: 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K), vol. 1. IEEE (2015)
5. Bieber, D., Geiger, M.: Personenbezogene Dienstleistungen in komplexen Dienstleistungssystemen – eine erste Annäherung. In: Bieber, D., Geiger, M. (eds.) Personenbezogene Dienstleistungen im Kontext komplexer Wertschöpfung, pp. 9–49. Springer, Wiesbaden (2014). https://doi.org/10.1007/978-3-531-19580-3_1
6. Blei, D., Mcauliffe, J.: Supervised topic models. In: Advances in Neural Information Processing Systems, vol. 20 (2007)
7. Blei, D.M.: Probabilistic topic models. Commun. ACM **55**(4), 77–84 (2012)
8. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
9. Flanagan, J.C.: The critical incident technique. Psychol. Bull. **51**(4), 327 (1954)
10. Halmos, P.: The personal service society. Br. J. Sociol. **18**, 13–28 (1967)
11. Lantow, B., Baudis, T., Lambusch, F.: Mining personal service processes. In: Abramowicz, W., Corchuelo, R. (eds.) BIS 2019. LNBIP, vol. 373, pp. 61–72. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36691-9_6
12. Lantow, B., Klaus, K.: Analysis of long-term personal service processes using dictionary-based text classification. In: Zimmermann, A., Howlett, R.J., Jain, L.C. (eds.) Human Centred Intelligent Systems. SIST, vol. 189, pp. 77–87. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-5784-2_7
13. Mannhardt, F., de Leoni, M., Reijers, H.A.: Heuristic mining revamped: an interactive, data-aware, and conformance-aware miner. In: BPM (Demos) (2017)
14. Moody, D.L.: Metrics for evaluating the quality of entity relationship models. In: Ling, T.-W., Ram, S., Li Lee, M. (eds.) ER 1998. LNCS, vol. 1507, pp. 211–225. Springer, Heidelberg (1998). https://doi.org/10.1007/978-3-540-49524-6_18
15. Ramage, D., Hall, D., Nallapati, R., Manning, C.D.: Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, vol. 1. Association for Computational Linguistics (2009)
16. Shuyo, N.: Python implementation of labeled LDA (Ramage+ EMNLP2009) (2013). https://shuyo.wordpress.com/2013/07/24/python-implementation-of-labeled-lda-ramage-emnlp2009/. Accessed 12 Mar 2021
17. Steyvers, M., Griffiths, T.: Probabilistic topic models. In: Latent Semantic Analysis: A Road to Meaning (2006)
18. 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
19. Vijayarani, S., Ilamathi, M.J., Nithya, M.: Preprocessing techniques for text mining-an overview. Int. J. Comput. Sci. Commun. Netw. **5**(1), 7–16 (2015)

20. Wolak, R., Kalafatis, S., Harris, P.: An investigation into four characteristics of services. J. Empir. Generalisations Mark. Sci. **3**(2), 22–43 (1998)
21. Yang, Y.: An evaluation of statistical approaches to text categorization. Inf. Retrieval **1**(1–2), 69–90 (1999). https://doi.org/10.1023/A:1009982220290

Event Stream and Predictive Monitoring (BPMDS 2021)

Generating Reliable Process Event Streams and Time Series Data Based on Neural Networks

Tobias Herbert[1(✉)], Juergen Mangler[2], and Stefanie Rinderle-Ma[2]

[1] Faculty of Computer Science, University of Vienna, Vienna, Austria
a11748469@unet.univie.ac.at

[2] Chair of Information Systems and Business Process Management,
Department of Informatics, Technical University of Munich, Garching, Germany
{juergen.mangler,stefanie.rinderle-ma}@tum.de

Abstract. Domains such as manufacturing and medicine crave for continuous monitoring and analysis of their processes, especially in combination with time series as produced by sensors. Time series data can be exploited to, for example, explain and predict concept drifts during runtime. Generally, a certain data volume is required in order to produce meaningful analysis results. However, reliable data sets are often missing, for example, if event streams and times series data are collected separately, in case of a new process, or if it is too expensive to obtain a sufficient data volume. Additional challenges arise with preparing time series data from multiple event sources, variations in data collection frequency, and concept drift. This paper proposes the GENLOG approach to generate reliable event and time series data that follows the distribution of the underlying input data set. GENLOG employs data resampling and enables the user to select different parts of the log data to orchestrate the training of a recurrent neural network for stream generation. The generated data is sampled back to its original sample rate and is embedded into the originating log data file. Overall, GENLOG can boost small data sets and consequently the application of online process mining.

Keywords: Time series generation · Recurrent neural network · Reliable dataset boosting · Synthetic log data · Deep learning

1 Introduction

The continuous monitoring and analysis of their processes is crucial for many application domains such as manufacturing [28] and medicine [12]. This holds particularly true if the processes are connected to physical devices such as machines or sensors that produce a plethora of data themselves, i.e., meta data that is actually not important for the genesis of work pieces and products, but is crucial to understand errors, optimize machines and processes, and to react to production problems more quickly. Often such machine and sensor data is

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 81–95, 2021.
https://doi.org/10.1007/978-3-030-79186-5_6

collected as time series. Consider the manufacturing example depicted in Fig. 1 where a CNC machine produces parts and the necessary tasks are logged into a *process event stream* during runtime. In addition, *metrics* such as load and torque are logged as time series for each motor. An analysis goal is to detect, explain, and predict anomalies in the process event stream (i.e., the process conducted by the CNC machine) based on the time series reflecting the parameters and using combined process mining and Machine Learning techniques[1].

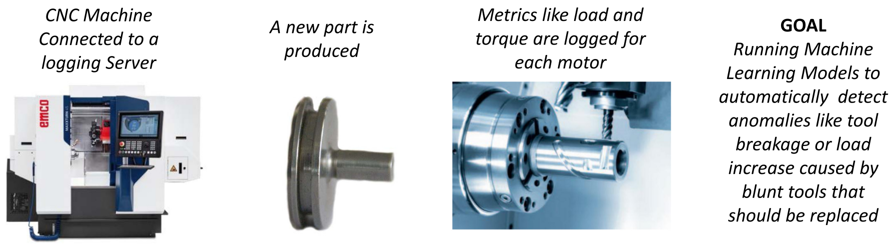


Fig. 1. Outages and machine deterioration inflict high costs in the manufacturing domain. The goal is to use machine learning models to detect tool deterioration and breakage to alert the machinist to avoid damages to the part, tools and the machine.

As promising as the application of powerful analysis techniques sounds, the precondition is to be able to collect *reliable* data sets as input. By reliable we refer to data sets of a sufficient volume and quality. The quality aspect is tied with the volume aspect as, for example, removing data of low quality can take a toll on the volume. If, in turn, measures are taken to boost data sets artificially, reliability of the result should imply that the structure of the original data is maintained, i.e., the resulting data set is as realistic as possible. In real-world scenarios, the following reasons might hamper the creation of reliable data sets:

– Slightly changing process models which often include manual repair steps, especially during early phases of production.
– Flaky time series data collection, as (a) machines utilize real-time computing capabilities and prioritize operational steps over data delivery steps, and (b) unstable networking and data aggregation, as individual sensors often produce more than hundred mebibytes per second.

For example the manufacturing domain, is in a tight spot. In theory it is expected to operate in a highly automated and connected manner, inline with widely advertised Industry 4.0 principles. In reality, it is still stuck with decade old machines, that are reliable, but black boxes when seen as sources of analyzable data, poor networking infrastructure, and lots of manual tweaking and repair steps. This results in gaps in most real-world manufacturing data sets.

[1] Previous work [28] combines, e.g., online process mining with dynamic time warping.

And this does not only hold for the manufacturing domain. Problems hampering the creation of reliable data sets for analysis purposes, might occur in other domains (e.g., medicine or logistics) for exactly the same reasons.

The main goal of this work is to support the creation of reliable data sets, i.e., process event streams and time series data, in case of existing low volume data. In Machine Learning, synthetic data has been already generated according to a given distribution and purposes such as privacy[2]. In this work, we propose GENLOG which employs Machine Learning techniques and is based on standard XES[3] log files. GENLOG aims at solving the following two problems:

– **Problem 1:** The ex-post generation of solid data sets with slight variations regarding both - process events and time series data.
– **Problem 2:** Filling in gaps in data sets caused by variations in data collection frequency and other data collection errors or deficiencies.

Current approaches for generating log data (in a process setting) often require many parameters that have to be estimated/simulated based on probabilities or probability distributions. Moreover, existing approaches often do not focus on or even include time series data. In order to overcome these limitations, the GENLOG approach uses Neural Networks to generate event and sensor streams, the latter reflected as time series data, based on existing log data. The generated data is then embedded into the originating file type and structure. In particular, no parameters are required to be set by the user.

Nonetheless, GENLOG aims at reflecting the distribution of the input log data in the generated event and data streams.

The ideas behind GENLOG are underpinned by the following considerations: Machine Learning algorithms are in general an asset for analyzing, monitoring, and optimizing business processes [30]. Whenever log data is available it is an opportunity to check for outliers and patterns [23], both for conformance and optimization purposes. Time series data can hold valuable insights about process execution, faults, failures, and changes in the process over time [33]. When a sufficient amount of data is available to train machine learning models, they can be used effectively in both online and offline scenarios such as near real time outlier detection as well as classification. However if there is only a small amount of data available it is not possible to train a model [16] that will generalize well to future data [22].

The implementation of GENLOG approach can be tried out at https://cpee.org/genlog/. The online tool allows to access all the data sets used in this paper, upload new ones, as well as access visual representations generated during the creation of new logs.

The remainder of this paper is structured as follows: Sect. 2 describes how log data sets are typically structured, and how the GENLOG pipeline works on these data sets. Section 3 outlines the prototypical implementation and Sect. 4

[2] https://research.aimultiple.com/synthetic-data-generation/.

[3] XES stands for eXtensible Event Stream and is the de facto process log standard, http://www.xes-standard.org/.

details the evaluation in a real-world scenario. Section 5 presents related work. In Sect. 6 the paper is concluded.

2 GENLOG Pipeline for Creating Reliable Process Event Streams and Time Series Data

Figure 1 uses the manufacturing of new part as an example to illustrate a real-world scenario in which GENLOG can be added to improve the monitoring abilities of the existing system to detect machine malfunction and deterioration which both reduces the need for human intervention and can prevent machine outages if a tool head is replaced before it breaks or the quality of the final parts is out of specification.

Figure 2 picks up the scenario presented in Fig. 1. It describes how GENLOG can be used to deal with the typical life-cycle of a part – i.e. design, production, and iterative improvements. Improvements might be necessary due to machining or process inefficiencies, e.g. to detect tool breakage or deterioration. All of these cases initially lead to a situation where not enough data is available:

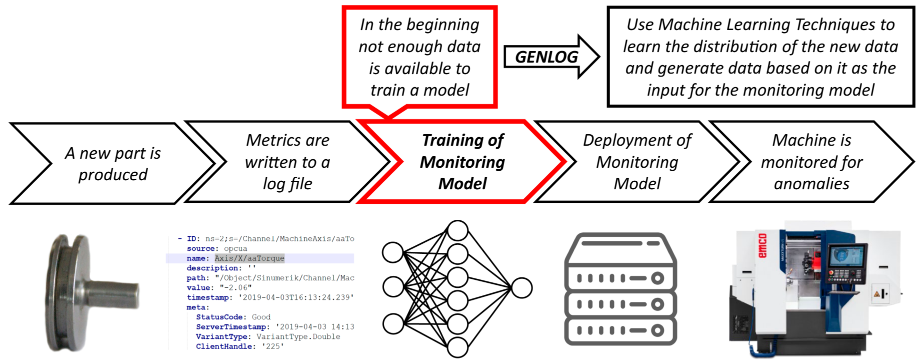


Fig. 2. Use GENLOG when a new part is produced and not enough training data is available as input for a Machine Learning Model

1. Process mining and Machine Learning techniques are employed in existing analysis tool chains. It is desired to use them as soon as possible when a new process is introduced, which is sometimes not possible due to the small size of the data set. In order to fill this “data gap”, the GENLOG approach takes the existing data to generate new data that follows the same distribution and is of the same file type, structure, and frequency as the original data. This way the data generated by GENLOG can be immediately used in all existing analysis processes that take the log files as input data.
2. Cleaning and pruning data for analysis purposes can also result in a data set that is too small to train a good model. Again the GENLOG approach can fill the gap by yielding a sufficient data set.

3. Finally, the data might not be balanced, i.e., the amount of data instances in one class differs from other classes. This can cause a bias in the resulting model [7] which is also common in other domains, such as the medical sector (e.g., most test subjects are healthy). Instead of balancing the classes by throwing away data from larger classes, data can be generated for the smaller classes. This increases the ability of the model to correctly adapt to new data.

GENLOG is added before the existing monitoring model is trained in order to provide a reliable data set, i.e., process event stream and connected time series. Thus from an early stage of production it is possible to train a model that can monitor the system and it will improve over time with more real data available for both direct training and as input to GENLOG.

Figure 3 shows the pipeline of the GENLOG approach. In this section, we start by sketching how the pipeline is designed in general, followed by details on the employed Machine Learning concepts. In Sect. 3, the description of the GENLOG pipeline is fleshed out in the context of its prototypical implementation.

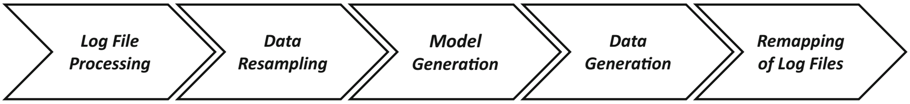


Fig. 3. GENLOG steps taking a log file as input, training a Machine Learning model to generate data and creating an output log file including the generated data

The GENLOG pipeline starts by processing the input log files. The result of this step is a collection of time series data, one for each metric for each log file. Examples for metrics are sensor data for temperature, acceleration or power related measurements. Next the data is resampled to a common sample rate for further processing. The data is then ready to be used as input for the training of the Neural Networks. At this stage, no human intervention was necessary and the only parameter was the list of metrics that should be extracted from the logs which can also be omitted if all metrics should be used. Now the user is presented with a dashboard that shows the metrics in terms of its grouping either by batches or temporal which are used to choose the data that will be the base for the data generation. The data is then generated using the prediction method of the trained models and using existing data as input. At this stage the data is still a time series and the final and crucial step is to map it back into its original file format. Thus it is embedded back into the original log file which results in a new log file with generated data that can be used by any process that uses log files as inputs and it could even be used in GENLOG to create new models which has been done and can be evaluated in the online tool.

Standard Neural Networks can be used in many applications. They do however not have a concept of time, but rather see their input as just a collection of inputs without a temporal relationship. Recurrent Neural Networks (RNNs) [19] use a hidden state vector to create a temporal context which influences the output of each node together with the weights of the node. This makes it a powerful concept for data generation and time series prediction, but they are still not able to keep important information over a longer period of time. During back propagation, recurrent neural networks suffer from the vanishing gradient problem [24] which causes the updates of the weights to become smaller the further the back propagation progresses which hinders learning. A Long Short-Term Memory (LSTM) [11] addresses this issue by keeping a global state and using gates to determine what information should be forgotten and what should be kept. It uses the hyperbolic tangent function to regularize the data and the sigmoid function to create forget, input and output gates that control which information are worth keeping and what should be discarded as shown in Fig. 4 [25]. Depending on the use case, it is possible to train many LSTM models even down to the individual instances. It is also possible to aggregate the data first and then train a model on, e.g., the median and standard deviation to reduce the effect of noise in the data. Besides the trained models, additional variance can be achieved by using the existing data as input for the models predict method such that the generated data is a combination of model and input data. By using domain knowledge it is possible to control the distribution of the data generation in the user interface by omitting invalid or problematic data. In the GENLOG approach, models are trained at fine granularity and the user selects the data that should be used to generate new log data. It can also be implemented as a continuous training by using new data to train new models and thus over time both replacing artificial data with real data and improving the quality of the generated data by having access to more real data for training. This approach allows to train machine learning models from a much earlier stage where only a small amount of data is available and over time increase the quality of the prediction. This is especially relevant to domains which deal with new scenarios regularly like the manufacturing domain where new parts are produced with no initial data available or the process model changes.

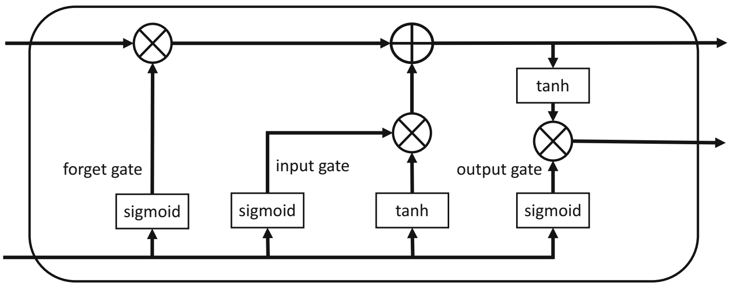


Fig. 4. Conceptual illustration of an LSTM

3 Prototypical Implementation

The GENLOG pipeline depicted in Fig. 3 and all the processing steps, model training, and data generation are implemented using python. The frontend makes heavy use of d3.js which is used to dynamically create the user interface based on the batches and metrics and is used to plot the data for validation and visual data analysis. The LSTM is implemented with the Tensorflow [34] backend which uses CUDA [26] for GPU accelerated training. The repository for the tool can be found here[4] and the tool can be used for evaluation here[5]. The core idea is to automate as much as possible and at the same time provide users with a high degree of influence over what data is used as the base for data generation, i.e., by removing or adding metrics and changing the mapping. This flexibility enables users to react especially to concept drift [2], i.e., changes in the log data that reflect process change and evolution.

3.1 Log File Processing

This step resembles an interface between the existing log file generation processes [21,27] and the test set data generation pipeline. The time series data for different metrics is extracted from the log files and stored in csv files as a timestamp and value pair. In a heterogeneous environment there can simply be multiple interfaces that take different input formats like XES and MXML [32] and output the time series csv files.

3.2 Data Resampling

All the data is now in the same format, but the frequencies differ both between the different measured metrics and within one file. The latter often arises because log data is written by the same device that is executing the process which can take resources away from log creation and towards the execution of the main process as well as data being transmitted through a network. The scale of the values matters in multivariate data as the metrics together form the features for model training. In a univariate model, feature scaling is not necessary as only one metric is used as the feature. To train a multivariate model it is advised to normalize the data because some loss metrics can introduce a bias otherwise [13] or converge more slowly as it is the case with gradient descent.

3.3 Model Generation

Preparing the data for training of the LSTM is done by creating a feature and a target vector. The target vector is created by shifting the values of the feature vector such that the target value is always the next value relative to the feature vector which is what the model is supposed to predict. The model is created by

[4] https://github.com/tohe91/GENLOG.
[5] https://cpee.org/genlog/.

defining the size of the LSTM layer and its activation function. The activation function used is the Rectified Linear Unit (ReLU) [10] which provides better gradient propagation [8] together with Adam optimization [14]. The size of the output layer is the number of features used and is one for a univariate time series. The loss function used is the Mean Squared Error [3]. A callback for early stopping [1] is added to increase performance by monitoring the loss function and stopping the training before max iterations are reached if the loss function did not change significantly in the last number of iterations.

3.4 Data Generation

To generate new data, both a trained model and an original data instance are chosen uniformly at random from a set that can be defined by the user through the user interface and the predict method of the model is called with a original data instance as the input. What makes this approach so powerful are the selective options that the user has. Often data can be grouped temporally by changes in the process and thus create data collections that correspond to a specific state of the process. By grouping the input data and the resulting trained models by the state of the process and other groupings like batch production or medical trials, the user can choose what should be the base for the newly generated data. This can be achieved by choosing the state of the process for each feature individually. In case there is no natural grouping apparent in the data, the user can chose a time window for each feature to represent the data basis for the data generation. This gives the user fine grained control over the generated data which can be beneficial if it is known that a specific feature was invalid or undesired for a given state of the process. Another approach the user can take is to specify if the generated data should be similar to the most recent data or if it should cover a broader time span. This can eliminate the need for manual pruning of data that is often required when using data analysis algorithms on raw log data. It can also be useful if manual pruning results in a data set that is too small for use in data analysis algorithms to produce meaningful results. By taking the pruned data and generating more data based on its distribution we create a more reliable data set.

3.5 Remapping to Log File Format

In order to go from validating the generated data to using it in existing workflows it is important to map the data back to its original log format. As we have extracted and resampled the data in the beginning of the pipeline we now have to reverse these steps by sampling it back and by embedding it into the original log file. To embed each metric, first the time series for that metric has be extracted and becomes the template for the generated data. The generated data is then resampled to match the template which can lastly be embedded back into the original log file. This makes it possible to maintain the integrity of the log file and selectively adding generated data to it. A detailed explanation will be given in the next section where the GENLOG approach is applied to the production of a metal part from a CNC turning machine on the shop floor.

4 Evaluation in a Real World Scenario

The GENLOG pipeline (cf. Fig. 3) is applied in a production environment based on log data from the Competence Center for Digital Production (CDP)[6]. An EMCO "MaxxTurn 45 SMY" machines a part in the GV12 production. GV12 is a *"part of a valve from a gas motor. The complexity of this product is caused by the small tolerances, which necessitates the complex overall scenario"* [20]. The production process is executed and logged by the Cloud Process Execution Engine (CPEE) [21,27]. The CPEE also logs metrics of the machine like the torque and load of the motors for each axis as well as spindle speed, spindle load, and other metrics from the turning process together with the process execution steps and meta data [4]. The produced log file originates in XES format and is then stored as a YAML file for further usage (cf. Fig. 5).


```yaml
- ID: ns=2;s=/Channel/MachineAxis/aaLoad[u1,1]
  source: opcua
  name: Axis/X/aaLoad
  description: ''
  path: "/Object/Sinumerik/Channel/MachineAxis/aaLoad[u1,1]"
  value: '21.0205078125'
  timestamp: '2019-04-03T16:13:24.239'
  meta:
    StatusCode: Good
    ServerTimestamp: '2019-04-03 14:13:24.316427'
    VariantType: VariantType.Double
    ClientHandle: '234'
```


Fig. 5. YAML log file produced in the GV12 production

4.1 Log File Processing

The created log files (as shown in Fig. 5) include all the metrics and information about the state of the machine, the state of the process, and meta data from the servers and production environment. The only requirement for the GENLOG pipeline to run is to provide the log files and their internal structure if it differs from the default and enter the metrics that should be extracted. If no metrics are specified all of the available ones are selected by default, i.e., the user does not have set any parameters. The pipeline then handles the extraction, resampling, and model generation without supervision or further input.

[6] http://acdp.at.

4.2 Data Resampling

The data is extracted and stored as time series data for each metric. The GEN-LOG tool enables the user to use visual analysis to find outliers, view different aggregations of the data, and analyze how the metrics change over time and through changes in production by plotting the data. The production of the parts is organized in batches which creates a natural grouping that is used to organize the log data. In the background the data is resampled to a common sample rate which is at least as high as the highest rate between all the metrics to make sure no data is lost in the resampling process.

4.3 Model Generation

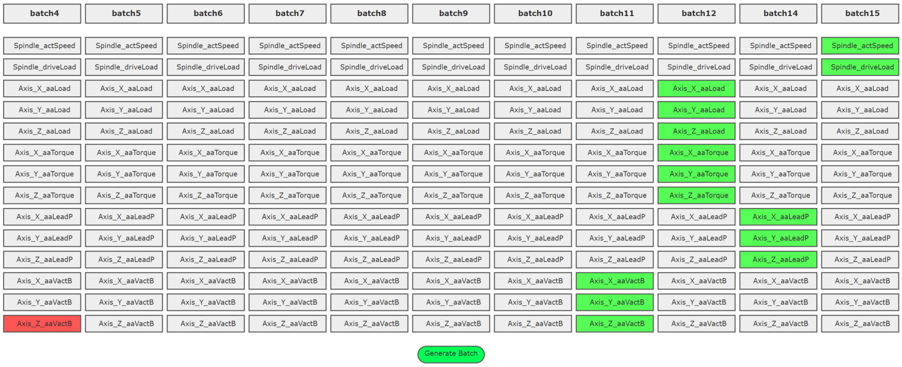


Fig. 6. User interface for data generation, **red**: model not trained yet - **gray**: model ready - **green**: model active for data generation (Color figure online)

The models are generated automatically as part of the pipeline and a fine grain model generation is used to preserve the characteristics of the individual production runs and no aggregation or generalization is done. Using an LSTM in this context is powerful as the different metrics have very different characteristics with some following a low polynomial function while others follow a high polynomial function with a high number of oscillations. This can be covered well by the LSTM because it predicts one data point at a time by using both recent information like an RNN and also older information which gives it its long term memory characteristics. Adding the early stopping functionality increases the total run time of the model creation significantly as some metrics result in no changes in the loss function earlier than others. For reference the creation of over 5800 LSTM models took just 5 h on a standard laptop (i7-7700HQ CPU, 16 GB RAM, NVIDIA GeForce GTX 1060).

4.4 Data Generation

At this point the user is presented with a dashboard (see Fig. 6) showing all the metrics for the individual batches. In the generic case this is where the temporal window could be chosen if there was no grouping in batches or process changes made beforehand. The user interface shows for which metric-batch combination the models have already been trained and are ready to use. The user can then pick which batches should be the data basis for each of the metrics. In case only the most recent data should be used as the basis the user can simply select the latest batch which activates all metrics for this batch and the generated data will represent this batch in all metrics. In case of a new process that only has very few data produced at this point, the selection can be based on the entire available data. In the scenario that multiple batches are chosen, the individual models and data instance, which represents one produced part, are chosen uniformly at random and are combined at random. As an example we chose batch14 and batch15 for a metric, there can be four possible combinations, model14-data14, model-14-data15, model15-data14, model15-data15, which produces a high degree of variance even if the original data set size was small because we can combine models and data input together.

4.5 Remapping to Log File Format

The previous step is sufficient to validate how well the newly generated data represents the same distribution as the original data. In order to become useful for data analysis, the data needs to be embedded into the original file format. In the remapping stage the generated data is embedded into the original log file. The embedding is done by first calculating the duration of the process from the original log file in milliseconds. Second the number of logged values for each metric are summed up, thus we have the duration of the process and the number of logged values for each metric which allows us to calculate a sample rate for each metric. Third the generated data of each metric is resampled with the calculated sample rate and therefore the generated data has the same amount of entries as the original log file for each metric. Fourth both value and timestamp for each metric are embedded into the log file. The timestamp was created by taking the first timestamp of the log file and then using the sample rate to increase each timestamp accordingly. As a result the generated log file has the same structure, length, variance and meta data as the originating log files.

4.6 Discussion of the Results

The results are encouraging as Fig. 7 shows the plots for generated data representing the original data quite well. Our technique is very suitable for processes that experience concept drift, meaning the process is subject to change over time, as the LSTM is conditioned to put emphasis on more recent data whilst considering the full data set. As a result the models continuously evolve with the process without the need to make a hard cut and starting over with an again

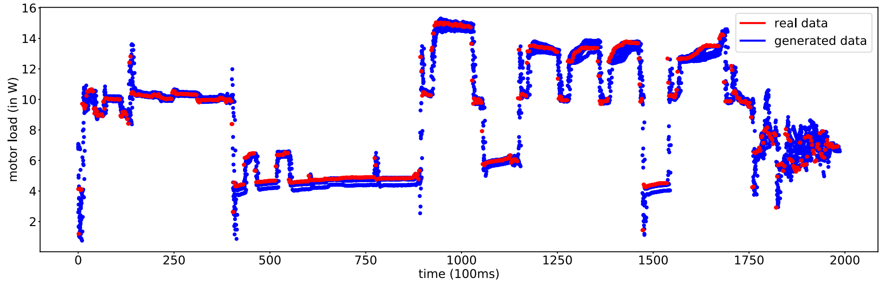


Fig. 7. The real data is shown in red while blue shows the distribution of 100 generated data samples for the load of the Y Axis Motor in the turning machine (Color figure online)

very small data set that only represents the altered process. It might be necessary to reset the data set in case the changes to the process are too severe, like changing the order of machining steps performed on the part, as this might cause unpredictable and undesirable artifacts in the generated data. Another limitation of this approach becomes apparent when a very small data set is used that does not include any significant variance. If the available data is very biased and not representing the real distribution well, the amount of variance GENLOG can introduce is also limited. Therefore we plan to introduce a list of statistical measures like variance and standard deviation and using dynamic time warping on the original data set and the resulting boosted data set to understand better how much variance can be introduced on different data sets. Figure 8 shows the minimum path for dynamic time warping between the generated data from two models and the two originating time series. The results show that using its own time series as input, the added variance on the output is small but exists because the models are not overfitted and therefore have not learned the data perfectly and when using the time series from another run, the added variance increases. This shows how cross using all models with all time series allows to boost a very small data set with realistic data.

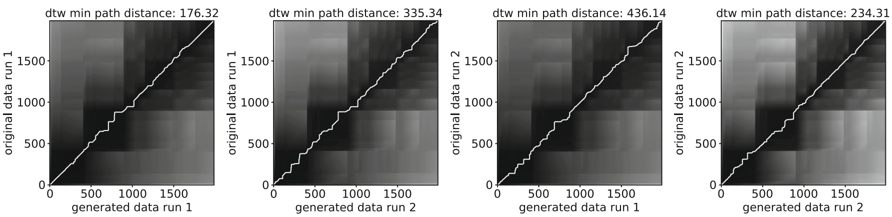


Fig. 8. Minimum path for dynamic time warping crossing two models with their originating time series data

5 Related Work

The process mining manifesto states that the provision of high quality event data is crucial (Guiding Principle 1 and Challenge 1) and that process mining should be combined with other analysis techniques (Challenge 9) [31]. GENLOG contributes to both claims. The generation of process log data can be approached from two directions, i.e., through simulation based on process models (e.g., using Colored Petri Nets [18] with CPN tools[7]) or through executing process models in "test mode", e.g., with mocked services [27]. These approaches, however, rely on process models with estimated values, e.g., probabilities at alternative branchings, and do not generate sensor data. An approach for integrating event streams from heterogeneous data sources is presented in [15], but no event streams are generated. Approaches such as [28] analyze and exploit event and sensor streams, but do not generate them either. Analyzing the capability of GENLOG for predicting the next activity in a process execution in comparison to existing approaches such as [5] using deep learning is part of future work.

LSTMs are not limited to time series data and can also be used for sequence labeling [17] and text classification [35] which can be used on the process and task level for conformance checking and anomaly detection. Gated Recurrent Units which are related to LSTMs and are also based on RNNs can be used to predict bottlenecks [6] in a production setting. So RNN based models might also be useful to address some of the issues in process mining that can be prone to both overfitting and underfitting [29] by creating models that use existing data instead of creating rules and using parameters.

In future research we intend to compare the results of the LSTM with different Generative Adversarial Networks (GAN) [9] variants within GENLOG to understand both theoretical and practical implications for both approaches for the generation of time series data. Grid search cross validation will be employed on all models to make it easier to adjust to new data sets without user intervention as a set of training parameters will be exhaustively checked and the best performing combination will be chosen as part of the pipeline.

6 Conclusion

A prototypical implementation of GENLOG is used offline to generate new event and time series data to be used as input for analysis algorithms. We applied GENLOG to a real-world data set from the manufacturing domain. The results showed that the models are not suffering from overfitting or underfitting as they introduce realistic variation both for their originating time series input as well as for the input of other runs.

Overall, the results show that it is feasible to use a pipeline such as GENLOG which does not rely extensively on pre-configured parameter sets, but instead solely utilizes historic log data, in order to generate new time series log data.

[7] http://cpntools.org/.

Acknowledgements. This work has been partly funded by the Austrian Research Promotion Agency (FFG) via the "Austrian Competence Center for Digital Production" (CDP) under the contract number 854187. This work has been supported by the Pilot Factory Industry 4.0, Seestadtstrasse 27, Vienna, Austria.

References

1. Blanchard, G., Hoffmann, M., Reiß, M.: Optimal adaptation for early stopping in statistical inverse problems. SIAM/ASA J. Uncertain. Quantif. **6**, 1043–1075 (2018)
2. Bose, R.P.J.C., van der Aalst, W.M.P., Zliobaite, I., Pechenizkiy, M.: Dealing with concept drifts in process mining. IEEE Trans. Neural Netw. Learn. Syst. **25**(1), 154–171 (2014)
3. Charkhi, A., Claeskens, G., Hansen, B.E.: Minimum mean squared error model averaging in likelihood models. Stat. Sin. **26**(2), 809–840 (2015)
4. Ehrendorfer, M., Fassmann, J., Mangler, J., Rinderle-Ma, S.: Conformance checking and classification of manufacturing log data. In: Business Informatics, pp. 569–577 (2019)
5. Evermann, J., Rehse, J., Fettke, P.: XES tensorflow - process prediction using the tensorflow deep-learning framework. In: Forum and Doctoral Consortium at International Conference on Advanced Information Systems Engineering, pp. 41–48 (2017)
6. Fang, W., Guo, Y., Liao, W., Huang, S., Yang, N., Liu, J.: A Parallel Gated Recurrent Units (P-GRUs) network for the shifting lateness bottleneck prediction in make-to-order production systems. Comput. Ind. Eng. **140**, 106246 (2020)
7. Fanny, Cenggoro, T.W.: Deep learning for imbalance data classification using class expert generative adversarial network. Procedia Comput. Sci. **135**, 60–67 (2018)
8. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: International Conference on Artificial Intelligence and Statistics, pp. 315–323 (2011)
9. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
10. Hara, K., Saito, D., Shouno, H.: Analysis of function of rectified linear unit used in deep learning. In: International Joint Conference on Neural Networks, pp. 1–8. IEEE (2015)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
12. Janiesch, C., et al.: The Internet of Things meets business process management: a manifesto. IEEE Syst. Man Cybern. Mag. **6**, 34–44 (2020)
13. Jayalakshmi, T., Santhakumaran, A.: Statistical normalization and back propagation for classification. Int. J. Comput. Theory Eng. **3**(1), 1793–8201 (2011)
14. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
15. Koenig, P., Mangler, J., Rinderle-Ma, S.: Compliance monitoring on process event streams from multiple sources. In: International Conference on Process Mining, pp. 113–120 (2019)
16. LeCun, Y., Bengio, Y., Hinton, G.E.: Deep learning. Nature **521**(7553), 436–444 (2015)
17. Ma, X., Hovy, E.H.: End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (2016)

18. Maggi, F.M., Westergaard, M.: Designing software for operational decision support through coloured petri nets. Enterp. IS **11**(5), 576–596 (2017)
19. Mandic, D.P.: Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability. Wiley, Hoboken (2001)
20. Mangler, J., Pauker, F., Rinderle-Ma, S., Ehrendorfer, M.: centurio.work - industry 4.0 integration assessment and evolution. In: Industry Forum at BPM, Volume 2428 of CEUR Workshop Proceedings, pp. 106–117 (2019)
21. Mangler, J., Stuermer, G., Schikuta, E.: Cloud process execution engine - evaluation of the core concepts. CoRR, abs/1003.3330 (2010)
22. Narayan, S., Tagliarini, G.: An analysis of underfitting in MLP networks. In: International Joint Conference on Neural Networks, vol. 2, pp. 984–988. IEEE (2005)
23. Oliner, A., Ganapathi, A., Xu, W.: Advances and challenges in log analysis. Commun. ACM **55**(2), 55–61 (2012)
24. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International Conference on Machine Learning, pp. 1310–1318 (2013)
25. Phi, M.: towardsdatascience.com: illustrated guide to LSTM's and GRU's: a step by step explanation (2018). https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21
26. Shi, S., Wang, Q., Xu, P., Chu, X.: Benchmarking state-of-the-art deep learning software tools. CoRR, abs/1608.07249 (2016)
27. Stertz, F., Rinderle-Ma, S., Hildebrandt, T., Mangler, J.: Testing processes with service invocation: advanced logging in CPEE. In: Drira, K., et al. (eds.) ICSOC 2016. LNCS, vol. 10380, pp. 189–193. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68136-8_22
28. Stertz, F., Rinderle-Ma, S., Mangler, J.: Analyzing process concept drifts based on sensor event streams during runtime. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) BPM 2020. LNCS, vol. 12168, pp. 202–219. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58666-9_12
29. van der Aalst, W., Rubin, V., Verbeek, H., Dongen, B., Kindler, E., Günther, C.: Process mining: a two-step approach to balance between underfitting and overfitting. Softw. Syst. Model. **9**(1), 87–111 (2010). https://doi.org/10.1007/s10270-008-0106-z
30. van der Aalst, W.: Process Mining: Data Science in Action, 2nd edn. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4
31. 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
32. van Dongen, B.F., van der Aalst, W.: A meta model for process mining data. EMOI-INTEROP **160**, 30 (2005)
33. Wei, W.: Time series analysis. In: The Oxford Handbook of Quantitative Methods in Psychology: Volume 2, Statistical Analysis. Oxford University Press (2013)
34. Zaccone, G.: Deep Learning with TensorFlow. Packt Publishing, Birmingham (2017)
35. Zhou, C., Sun, C., Liu, Z., Lau, F.C.M.: A C-LSTM neural network for text classification. CoRR, abs/1511.08630 (2015)

Inter-case Properties and Process Variant Considerations in Time Prediction: A Conceptual Framework

Avihai Grinvald, Pnina Soffer(✉), and Osnat Mokryn

Information Systems, University of Haifa, Haifa, Israel
spnina@is.haifa.ac.il

Abstract. Predictive monitoring in business processes has gained attention in recent years. It uses a predictive model, learned from event logs, to predict the variables of interest for a running process instance (case). An example of such a variable considered here is the remaining time to complete the running case. Prediction usually relies on the properties of individual cases. Recently, the effects of the case's environment, particularly cases that are executed in parallel to it, have been incorporated into prediction as inter-case properties. Furthermore, it has been recognized that, when different variants of the process exist, variant information should be considered by the predictive model. However, different prediction approaches use inter-case properties and variant information differently, and there is still no clear and agreed-upon manner in which these are considered for prediction. This paper proposes a conceptual framework that suggests categories of inter-case properties related to cases within a time window. Moreover, the framework considers the possible variant-awareness of these properties and suggests how variant information should be addressed in a predictive model. Reported experimentation supports our proposals.

Keywords: Remaining time prediction · Inter-case properties · Conceptual model · Process variant

1 Introduction

Business processes form the operational backbone of organizations. Processes are supported by business process management systems that generate event logs where the execution of processes is recorded. These event logs serve as a basis for various process mining techniques aimed at process model discovery, conformance checking, process analysis, recommendation, and predictive monitoring. Predictive monitoring uses a predictive model, learned from event logs, to predict variables of interest for a running process instance (case). The variables of interest for prediction may vary. Examples include predicting the next activity to be performed [10], predicting some business outcome variables (e.g., success or failure) [26], and predicting remaining time until completion of the running

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 96–111, 2021.
https://doi.org/10.1007/978-3-030-79186-5_7

case [28]. Predicting remaining time is highly important for operational planning and control, specifically for anticipating and avoiding delays. Furthermore, manual remaining time prediction is inherently difficult due to human cognitive biases [4]. Hence, it is the focus of our study.

The backbone features used for remaining time prediction are the individual cases' intrinsic properties, e.g., the trace, elapsed time, resources, and more. Several works have recently suggested additional information sources to enrich the feature-base and improve the prediction: contextual information from the environment [29] and information from cases that run parallel to the predicted case in the form of inter-case properties [19]. When predicting the remaining time of a running case, the significance of parallel cases is twofold. First, they compete with the case under consideration on resources, which might be a source for delays if resources are scarce or the system is overloaded. Second, unrecorded environmental events (e.g., a pandemic) may simultaneously affect all cases and be reflected in their duration and completion times. Observed delays at cases that are performed parallel to a running case may imply a similar delay in the predicted case.

An additional consideration that has been suggested is the existence of different variants of the process. For example, for first-aid treatment, [21] suggests that predictions for urgent patients should be performed separately from non-urgent ones. Consequently, parallel cases whose variant is not as the predicted case might not be relevant for predicting and even interfering, as *variants may differ* in completion times and business constraints.

Introducing contextual information from parallel cases is indeed material and can aid in the prediction. However, it is not clear how it should be incorporated into the global process of prediction. Moreover, this information has been considered in the context of arbitrarily selected sets of properties. Parallel cases can be considered, for example, in the context of resources, by computing the overall load created. Still, how should their control flow or data aspects be considered? And which set of cases should be considered a relevant context for predicting the remaining time of a running case of a given variant?

In this paper, we step towards unveiling this complexity by suggesting a conceptual framework for completion time prediction. The framework considers inter-case properties at different process perspectives, namely control flow, resource, time, and data while considering each perspective's variants' effect. We consider each property and assess whether it is variant aware or indifferent and incorporate it within the perspectives. To the best of our knowledge, this is the first attempt to suggest and evaluate inter-case properties for the four perspectives when considering process variant information. Our results demonstrate the importance of creating a global view, as manifested in our framework, for a prediction process that includes inter-case properties.

2 Preliminaries and Basic Concepts

This section briefly presents the preliminary and basic concepts required for our approach. Central to process mining are the concepts of event logs, cases, and events. Let ε be the universe of all events. An event log L is a K-sized set of event sequences (or traces), $L = \{\sigma_i, i \in 1 \ldots K\}$ with $\sigma_i = (e_i^1 \ldots e_i^{n_i})$ being the i_{th} trace of length n_i. Each trace is associated with a case - a unique execution of the process, marked with a case ID. At least three attributes characterize each event in ε: CaseId - referencing the case the event belongs to; Activity - the name of the activity related to the event; and the Timestamp of the event.

In addition, an event may be characterized by other attributes, e.g., resources. Let AN be a set of attribute names. For any event $e \in \varepsilon$ and attribute name $an \in AN$, $\#_{an}(e)$ is the value of the attribute an for event e. If event e does not have an attribute name an, then $\#_{an}(e) = \perp$ (null value). For convenience, the following standard attributes are assumed: $\#_{\text{activity}}(e)$ is the activity associated with event e, $\#_{\text{time}}(e)$ is the timestamp of event e, and $\#_{\text{resource}}(e)$ is the resource of event e. The CaseID is denoted by $\#_{id}(e)$. Given a trace σ_i in log L, e_i^j denotes the j_{th} event in the trace, σ_i relates to the case whose ID is i.

Our proposed prediction approach relies on each case's properties and inter-case properties, which are properties of a case that are not derived from data of the case and its events but rather from the extrinsic context and runtime data of other cases. The case to which prediction is applied is called the *base case*. The current event upon which prediction is made is termed the *anchor event*. The cases used for setting the base case's inter-case properties are the ones that are performed in parallel to it within a given time window.

Definition 1 (Time window). *Let $t_i^j = \#_{time}(e_i^j)$ be the time of e_i^j event in the base case c_i. The Time window $[f_{start}, f_{end}]$ over t_i^j with window size w is defined as $[t_i^j - w, t_i^j]$, where $t_i^j - w$ is the window start time, and t_i^j is the window end time.*

Note that this type of time window is defined as a backward (past) time window. Theoretically, other types of time windows could be defined (e.g., future time window $[t_i^j, t_i^j + w]$ or past + future time window $[t_i^j - w, t_i^j + w]$). However, these are not suitable for prediction. Our prediction approach refers to inter-case properties, considering cases that are active in parallel to the base case as peer cases during the defined time-window. Learning of inter-case properties is done only for peer cases, considering their partial traces prior to the anchor event.

Definition 2 (Peer cases). *Let e_i^j be an anchor event in a base case i with a time window $f = [t_i^j - w, t_i^j]$. A case $c_j, (i \neq j)$ is a Peer case with respect to event e_i^j if: $t_j^1 < f_{end}$ and $t_j^n > f_{start}$, where $n = |\#_{trace}(c_j)|$. The set of all the Peer cases with respect to event e_i^j is marked as $PC(e_i^j, w)$.*

In other words, a peer case starts before the end of the time window and ends after the beginning of the time window. For example, consider the cases whose timeline is shown in Fig. 1, with e_1^4 as an anchor event. Only cases that exist during the time window are in the set of peer cases $PC(e_1^4, w)$. Hence, cases σ_5 and σ_6 are not in this set because the former ends before the beginning of the time window, and the latter starts after the end of the time window.

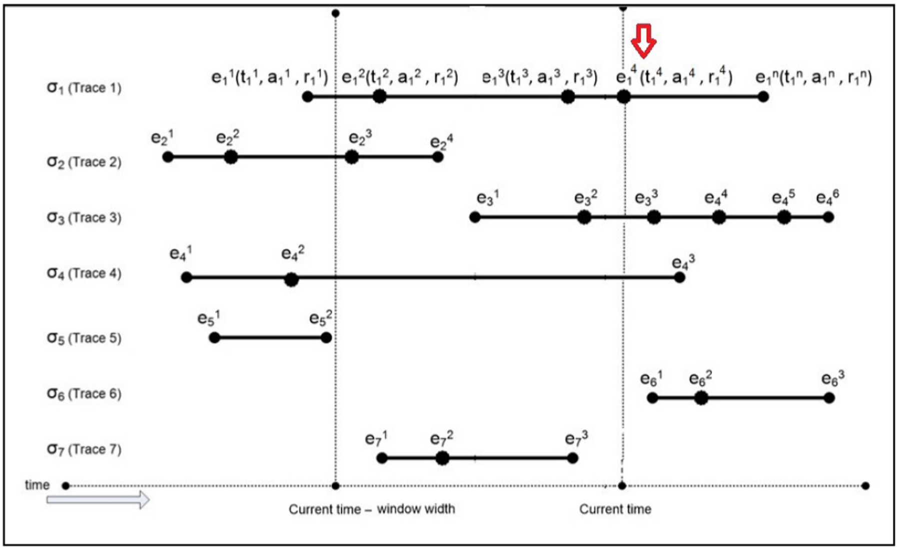


Fig. 1. Timeline schema of seven cases and their events. The anchor event is e_1^4, and the vertical dotted lines delimit the time window frame.

3 Inter-case Properties

This section presents and defines the conceptual framework, which considers intrinsic case properties and inter-case properties, using four process perspectives.

3.1 Perspectives of Inter-case Properties

The inter-case properties, which are derived from peer case properties, can relate to each of the four perspectives of a process [7], namely, control flow, resource, time, and data. Specific properties can be defined based on relevance assessment, which may rely on domain knowledge. We now provide example properties for each perspective, given a base case whose trace is $\sigma = (e_1 \dots e_n)$.

The control flow perspective relates to the activities ($\#_{\text{activity}}(e)$) in a trace and to their ordering.

Example property 1: Activity count – for each activity in σ, count the number of times it appears in traces of peer cases. This property reflects repetitions and looping behavior in peer cases that may exist similarly in the base case. It is represented as a set of features corresponding to each distinct activity in σ.

Example property 2: Frequent previous activity – for each activity in σ, return the most frequent immediate predecessor activity in traces of peer instances. This property reflects path decisions that may apply similarly in the base case. It results in a set of features corresponding to each distinct activity in σ.

The resource perspective relates to the resources ($\#_{\text{resource}}(e)$) that are associated with the events in σ and perform the related activities.

Example property 1: Activity count per resource – for each resource in the log, count the number of events associated with it in the time window f. This property reflects the general load on resources during the time window. It is represented as a set of features, corresponding to each resource.

Example property 2: Activity count per resource group – similar to Example 1, but relating to groups of resources that perform similar tasks. Resource grouping can be based on a role attribute if such is available in the data. Assuming that all resources of the same role can perform the same set of activities, the group's load can be more meaningful than each resource's. In case role information is not available, grouping can be done by clustering the resources with respect to their associated activities.

The time perspective relates to absolute time passing during the execution of the process and the delay or earliness of activity execution with respect to its expected time.

Example property 1: Average elapsed time of activities that correspond to the anchor event – let $a = \#_{\text{activity}}(e^A)$ be the activity corresponding to the anchor event. The property returns the average elapsed time of all the events associated with a in all peer cases. This property reflects the relative progress of the base case compared to peer cases with similar activities.

Example property 2: Average delay/earliness of the last event of peer cases – this property requires a preprocessing step, where, based on the entire log, the expected elapsed time of each activity $ET(a)$ is calculated. The property returns the average deviation (delay/earliness) in the elapsed time of the last event in the time window compared to the relevant $ET(a)$, for all peer cases (e.g., the elapsed time of the last recorded event of all peer cases is longer than expected). This information may reflect environmental effects which apply to all active cases.

We note that batching behavior [15] also reflects correlation along time.

The data perspective relates to additional data attributes that may be available for events in the log. Some attributes may hold relevant information for time prediction. For example, containers that hold hazardous chemicals require special care in a harbor. Thus a high number of peer cases whose content type is 'hazardous' may result in overall slower handling of all cases. This distinction, however, is highly domain-dependent and can be indicated by domain experts. Hence, no generic example properties are given. Nevertheless, we note that the data view properties typically count the number of cases that share some known attribute values.

3.2 Consideration of Process Variants

Often, not all cases are equally informative for a particular base case. Commonly, a process may have several variants, corresponding to different business cases with distinctive path selections, rules, or expected times. Consider, for example, a flight check-in process, in which business class passengers spend less time in queues than economy class ones. The prediction of completion time needs to consider only relevant peer cases, namely, cases that belong to the same process variant. A process variant is formally defined as follows [25]:

Definition 3 (Process variant, Variant indicator). *Given an event log L that can be partitioned into a finite set of subsets of traces* $\varphi_1, \varphi_2, ...\varphi_n$*, where* $\forall\varphi$ *and* $\forall\sigma_k \in \varphi, \exists d$*, such that* $\forall e_k^i,\ e_k^j \in \sigma_k$*,* $\#_d(e_k^i) = \#_d(e_k^j)$*, we call each subset* φ *a process variant, where d is the variant indicator.*

The variant indicator is an attribute that characterizes the process variants and distinguishes between them (e.g., by urgency, by type of customer, or by some combination of features). The variant indicator can typically be identified based on domain knowledge. Alternatively, machine learning methods can automatically identify it from the data [21].

We incorporate inter-case properties as part of the predictive model to capture correlations between cases and identify environmental (contextual) impacts. While variant information is essential for capturing strong correlations, contextual effects can be manifested across and stem from all the active cases. It follows that not all inter-case properties should be derived from the same set of cases. Properties that reflect context through case correlation need to rely on cases of the same variant as the base case, namely, be *variant aware* (VA). In contrast, properties where contextual information is manifested directly (e.g., system load) need to relate to all peer cases, namely, be *variant indifferent* (VI). We now discuss how variant information should be incorporated into inter-case properties for each of the four perspectives.

Control-flow perspective - captures properties related to the path taken by cases. We expect the same variant cases may follow similar paths, abiding by business rules that concern the variant indicator. For example, assume the variant indicator in an ordering process is the customer type. Different types of customers (e.g., business customers) entail different paths for handling their orders, and hence a strong correlation in the control flow is expected among customers of the same type. Although this is not necessarily always the case, inter-case properties of the control flow perspective are generally considered variant-aware.

Resource perspective - captures the current load and reflects contextual information for the base case. As process variants often share resources, the relevant information concerns all peer cases, regardless of their variant. The properties of this perspective are, therefore, considered variant indifferent. We note, however, that when resources are specifically designated to process variants, the resource-related properties may be variant-aware.

Time perspective - captures the timeline along which the process progresses. As process variants may have different governing rules that prioritize handling one variant over the other, we expect a high correlation of temporal behavior within variants and a lower correlation between cases belonging to different variants. Accordingly, we consider the time perspective properties as variant aware.

Data perspective - captures similarity and correlations based on data attribute values. As discussed earlier, the specific properties considered for this perspective may differ and are highly domain-dependent. Hence, it is not possible to determine whether these properties are variant aware or indifferent generally, and both configurations are possible.

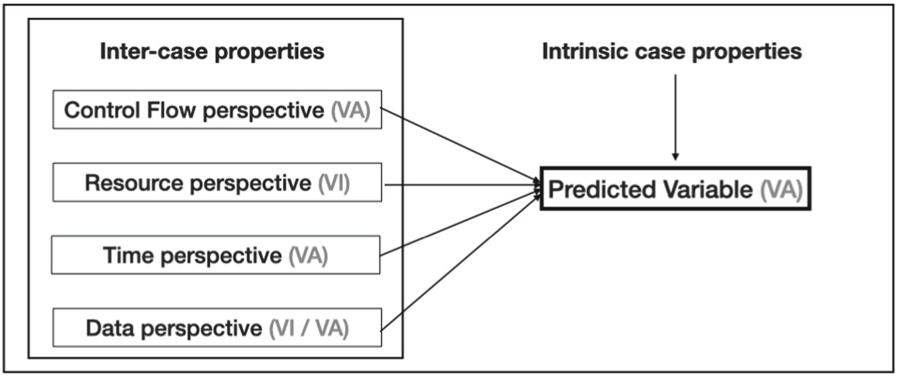


Fig. 2. The conceptual prediction framework with inter-case properties marked as variant-aware (VA) or variant-indifferent (VI).

In summary, our proposed framework is illustrated in Fig. 2. Each perspective is illustrated as either variant aware (VA) or variant indifferent (VI), as expected based on the discussion above.

Constructing the Feature Vector and Predicting Remaining Time for an Anchor Event in a Base Case: The feature vector includes intrinsic case properties and inter-case properties, based on the peer cases in the time window. As discussed, the inter-case properties include variant-aware and variant-indifferent properties. The variant-aware properties are calculated based on the peer cases of the same variant as the base case; the variant indifferent properties are calculated based on *all* the peer cases, indifferent to their variant. To incorporate both variant-aware and variant-indifferent properties, we consider the entire log, as opposed to other variant-aware prediction approaches (e.g., [8,21]), which split the log into separate logs per variant. Once the feature vector is created, any prediction algorithm can be applied, using the log portion that holds cases of the same variant as the base case. In other words, the prediction is variant-aware in general while allowing variant-indifference for the calculation of the variant-indifferent features.

4 Evaluation

Our goal here is to evaluate the added value of using our framework, namely, how the different perspectives and their combination contribute to the prediction compared to a baseline of intrinsic case properties. In addition, we explore

Table 1. Logs in the study

	TOS	BPIC2012	BPIC2015A	BPIC2015C	RTFM
Number of cases	13,087	78,972	1,199	1,409	150,370
Number of events	262,200	1,885,935	5,2217	59,681	561,470
Period	40 days	5 months	5 years	5 years	13 years
Mean case duration	5.5 days	8.6 days	96 days	62 days	48.8 weeks

prerequisites that need to be considered for working with the framework. For these purposes, any prediction algorithm can be selected as a basis. We implemented the framework as an extension of a Prom plugin [7][1], which provides a readily available and easily configurable prediction functionality. The implementation allows the user to select the variant indicator and to determine for each property whether it should be considered as variant aware or indifferent.

The Datasets Used in this Section: Several datasets are used in this research, some well known and a new one, termed Terminal Operating System (TOS). The TOS dataset is an extracted event log from a terminal operating system [23] in a global logistic hub seaport. The log records a container handling process, from the arrival of containers at the harbor till their embarking. The process has several variants, two of which form 80% of the log cases and are the focus of our study. One variant is for imported goods, where the containers are first unloaded from an incoming ship and handled in the port until collected in various means. The other is export, where containers are loaded to a ship before it leaves. Both variants share resources and activities. Each handling of a container is considered a trace in our log. The log also contains temporal information for all events in the containers' traces. Additional datasets are the well known Bpic 2015_1 and 2015_3 challenge logs (arbitrarily selected from the five logs of this challenge), the Bpic 2012 loan applications log, and the road traffic fine management (RTFM) log, where only cases that completed successfully (in fine payment) were used for prediction, to allow an unbiased prediction of time to completion. Table 1 provides statistics of the logs.

For the TOS dataset *Resource groups* were not indicated and needed to be calculated for the resource perspective properties, to reflect resources that share skills, capabilities, or roles and can often substitute or help each other. To find the resource groups, we found resources performing similar activities in a similar amount of time by clustering. We used K-means clustering with $k \in [6, 8, 10, 12]$ and found that $k = 10$ yields the best result (as evaluated by a domain expert). The resource groups are then used as features instead of resources.

[1] The code can be found at https://github.com/avihaigr/ConcurrentFeaturesPrediction.

Table 2. Remaining time prediction results as avg Mean Absolute Error (MAE) in hours over the TOS (Export), Bpic 2015_1, and Bpic 2015_3 datasets, for each perspective alone and for all perspectives joined with the best performing configurations. A baseline of the intrinsic-only view is given for comparison.

Perspective	Features	TOS MAE	Bpic 2015_1 MAE	Bpic 2015_3 MAE
Baseline	Baseline	50.3	34.11	18.75
Resource	Enhanced	34.9	33.89	18.55
Control flow	Enhanced	34.6	24.28	14.42
Time	Enhanced	37.0	33.24	18.67
Data	Enhanced	33.4	34.03	18.56
All	Best conf.	**33.0**	**24.23**	**14.41**

4.1 Framework Evaluation

We evaluated the prediction of each of the perspectives and the combined framework against a baseline feature set consisting of only intrinsic base-case properties available in the extended Prom plugin (e.g., elapsed time, previous activity, executing resource). The window size that was chosen for the TOS dataset and the Bpic 2012 was a day as the length of an average case is five days and nine days, respectively; the window size for the Bpic 2015 datasets was seven days, corresponding to an average case length of 96 days and 62 days correspondingly[2]. For the RTFM dataset, a window size of 7 days was used as well.

We performed the following experiments on all datasets:

- Using only the baseline intrinsic properties.
- Adding inter-case features of each perspective separately to the baseline ones.
- Adding inter-case features of all perspectives together (joined perspectives) to the baseline ones.

For the logs where variants were considered, we tested each added perspective as variant-aware and as variant-indifferent. For the joined perspectives, we selected the better-performing configuration for each perspective. All experiments were performed using 10-fold cross-validation and repeated three times. When preparing the feature vector for each case, we first calculated the inter-case properties using cases that preceded the base case (past cases). Then, the absolute timestamps were replaced by relative ones, showing elapsed time from the beginning of the case. By this, we eliminated possible time-related biases that may arise when using 10-fold cross-validation to all cases.

Table 2 shows the results of comparing the added value of each perspective for the TOS (Export variant), Bpic 2015_1, and Bpic 2015_3 in terms of Mean Absolute Error (MAE) in hours. For all three logs, the best prediction results are

[2] The Bpic datasets were also evaluated with a window size of a day, but the better results were obtained with a window size of seven days.

Table 3. TOS dataset for the exported containers case, perspectives are variant aware or indifferent. Framework corresponds to the joined perspectives when the best performing configuration (VI or VA) is taken for each perspective.

Perspective	VI	MAE (hrs)	% improv.	VA	MAE (hrs)	% improv.
Baseline	Baseline	50.4				
Resource	Enhanced	**34.9**	**30.75**	Enhanced	35.4	29.82
Control flow	Enhanced	35.2	30.06	Enhanced	**34.6**	**31.23**
Time	Enhanced	37.7	25.2	Enhanced	**37.0**	**26.49**
Data	Enhanced	**33.4**	**33.69**	Enhanced	34.1	32.35
Framework	Best confs.	33.0	34.56			

Table 4. Remaining time prediction results as avg MAE over the Bpic 2015_1 and Bpic 2015_3 datasets, perspectives are VA or VI. Framework stands for the joined perspectives when the best performing configuration (VI/VA) is taken for each perspective.

Perspective	Bpic 2015_1 VA	Bpic 2015_1 VI	Bpic 2015_3 VA	Bpic 2015_3 VI
Baseline	34.11		18.75	
Resource	34.08	**33.89**	18.70	**18.55**
Control flow	32.82	**24.28**	17.53	**14.42**
Time	34.16	**34.03**	18.78	**18.67**
Data	33.72	**33.24**	**18.51**	18.56
Framework	32.87	**24.23**	17.50	**14.41**

for the global view combining all perspectives. For both the Bpic 2015_1 and Bpic 2015_3 logs, the control flow perspective provides the highest contribution to the prediction result. For the TOS log, the contribution of the Data perspective is higher than that of the control flow.

The above results demonstrate the importance of the combined perspectives and the added value of a general view that considers all four perspectives. Moreover, different perspectives might be dominant for different datasets, and it is not trivial to foresee. Not knowing which view is dominant also strengthens the need for a combined global view that includes all perspectives.

4.2 Variants

As described in Sect. 3.2, an important aspect to be considered along with intercase properties for remaining time prediction is the process variant, as for some perspectives the behavior of different variants is substantially different, and overlooking this might be detrimental to the prediction task (see Fig. 2). Table 3 depicts the prediction results for the TOS export cases for different configurations of variant awareness vs. indifference of the inter-case properties. For each perspective, the configuration giving the best prediction is in bold. Affirming our design considerations, we see that the variant-aware feature set gives better results for both the control flow and time perspectives, and the variant-indifferent feature set gives better results for the resource and data perspectives. However, taking variants into consideration has its caveats, as we show next.

In the TOS dataset, where the variants were well known, our results show that variant awareness gave good improvement over the baseline. However, this information is not always present and cannot always be mined easily [3]. To investigate what happens when the variants are unknown and cannot be inferred easily, we guessed (erroneously) the variants in both Bpic 2015_1 and Bpic 2015_3 datasets. We note that the selected variant indicator was the type of permit requested in the process, which could be considered a logical choice, but was not confirmed in any other way [25]. We computed and compared the corresponding variant-aware and variant indifferent remaining time prediction.

Table 4 shows the prediction results in these cases for each perspective for both the variant-aware and variant-indifferent configurations, as well as for the global view. Clearly, the variant-indifferent configuration gives better results in the vast majority of cases. Why is that? As variant-aware features consider only part of the cases (those assumed in the same variant as the base case), less data is used for training the predictive model. As the variants were not inferred correctly, reducing the volume of data hampered prediction without adding information about the expected behavior of the cases.

Overall, the results obtained with the TOS dataset support our expectations concerning the variant awareness or indifference of the properties included in the feature-set for time prediction. Still, in processes where the variants are not distinct or cannot be easily inferred from the log, we expect the best prediction results with a variant-indifferent feature set.

4.3 When to Expect Contribution of Inter-case Properties

While the results obtained for the TOS export variant and the Bpic 2015 datasets show substantial improvement gained by adding the inter-case properties to the predictive model, this was not the case for all our experiments. As our results (Table 5) show, even different variants of the same process – TOS – exhibit different contributions of inter-case properties. While it is too early to draw general conclusions, we can make some observations and suggest possible explanations.

One possible explanation is deadlines and control over the completion of the process. The export variant of the TOS dataset has a firm deadline: all containers must be loaded on the ship by the time it has to leave, or the port is fined. In other words, the deadline is an extrinsic constraint with a fixed date and time for the completion of process cases [2], and this applies to all the cases assigned to the same ship as a batch. Furthermore, loading containers and getting the process to complete is at the hands of the harbor. In this environment, peer cases hold very relevant information for predicting a base case's completion time.

In contrast, for the import variant, completion time is not determined by a deadline. Instead, it depends on the customers, who should arrange the shipment of containers to their destinations, either by trucks, trains, or other ways. Consequently, completion time differs greatly among cases and is only partly affected by the harbor acts. Thus, the added value of peer case information is questionable. Indeed, the improvement gained by adding the inter-case properties adds up to 6%, far less than the 34% achieved for the export variant.

A similar observation can be made considering the RTFM process, where adding the inter-case properties improved the prediction result by 1.5% only. In this process, the actions that bring the process to completion (namely, paying the fine) are executed by the individuals who receive the fine and are not controlled by the process owners. Consequently, peer case information is not of much relevance for prediction.

Another explanation is the interaction among cases. For the RTFM, cases do not share a physical environment or resources and are executed individually without interacting. This observation is further supported over the Bpic 2012 dataset of a loan application process. The improvement gained by adding inter-case properties for this dataset is 0.3%—nearly no improvement at all. As this process is largely virtual, cases are performed individually, independently of each other, and without interactions. In general, for processes that rely on a few shared resources, if at all, the influence of peer cases is probably negligible. As there were no known variants for RTFM and Bpic 2012, the experiments for these datasets were performed with a variant-indifferent feature set.

Table 5 shows the results that affirm that inter-case properties in all these datasets – TOS import, Bpic 2012, and RTFM – do not improve substantially over the baseline intrinsic feature set.

Table 5. MAE of TOS dataset for the imported containers, Bpic 2012, and RTFM. Perspectives are variant aware or indifferent. Framework corresponds to the joined perspectives when the best performing configuration is taken for each perspective.

Perspective	Features	TOS imprt (hrs)	Bpic 2012 (hrs)	RTFM (days)
Baseline	Baseline	38.6	188.9	155.6
Resource	Enhanced	36.3	188.9	155.0
Control flow	Enhanced	36.3	188.6	154.8
Time	Enhanced	36.9	188.8	155.6
Data	Enhanced	36.3	188.9	153.6
Framework	Best conf.	36.2	188.3	153.3

5 Related Work

Prediction in business processes has gained attention in the past decade. The prediction may serve different purposes, most notably improving the process and making recommendations about its decision-making [6,13] and predictive monitoring (see survey in [17]), where the trained model is used for monitoring process execution at runtime, supporting early detection of possible risks and undesirable outcomes. Different predicted variables are targetted, including the next activity at a given moment (e.g., [10]), some outcome variable (see review in [26]), and remaining time until completion of the process (see review in [28]).

Prediction approaches also differ in the features considered for prediction. Most common are properties related to the control flow perspective – the partial trace until the point of prediction, often accompanied by the time dimension (e.g., [1,18]). A different focus is taken by [5], who addresses resource allocation and emphasizes the resource perspective. [7] combines features related to all perspectives in a generic framework, correlating process features, and predicting a selected one. The data perspective is often associated with the context of a case. Several works [8,11,13,22] use the data perspective for clustering cases by their context and then applying prediction to each cluster separately. Relevant context is discovered by [12] by classifying the cases based on their control flow and outcome. Capturing context aims to capture the effect of the environment on the predicted variable. Attempts in this direction include [16] and [29] who use external information about incidents [16] and about sentiments in the news [29] for time predictions of running cases.

The environment of a running case also includes the cases that are executed in parallel, reflected in inter-case properties which have been proposed. [9] characterizes predictive monitoring approaches and relates to the use of inter-case properties by several approaches as specific metrics. However, the choice of these properties varies in different approaches and is not made systematically. We review these along the perspectives of our framework. Most commonly used are properties related to the load of the system, which can be attributed to our resource perspective. [7] includes specific inter-case properties such as the number of cases and load on resources in a time-window; [27] consider the number of open cases in parallel to an event. [14] use three inter-case properties – the number of parallel events, the number of used resources, and the number of cases. Specifically considering resources as resulting in queues, [20] propose queue mining for predicting delays. The control flow perspective is addressed in combination with the time perspective by [19] and [21]. Considering the time perspective, [15] design and use specific inter-case properties related to batching behavior. Properties of the data perspective are addressed by three works. [24] defines features for the specific domain of invoice payments, including specific inter-case ones (e.g., the number of urgent invoices). [19] relate to process-variants that stem from case data (e.g., urgency level, also addressed similarly by [24]). None of the reviewed works addressed all four perspectives together.

Considering process variants, [19] and [21] split the log into separate logs for each variant, to which prediction is performed. This approach is extended in [21], where the selection of variant is automatically learned from the data and does not depend on domain knowledge. We note that other methods can automatically detect variants (e.g., [25]), independently of prediction. The main difference between the use of variant for prediction as suggested by [21] and the approach suggested in this paper is that we do not split the log. Rather, we keep the full log and allow a distinction between variant-aware properties, relying on cases of the same variant as the base case, and variant-indifferent ones, relying on all cases in the log.

In summary, compared to the reviewed literature, the main contributions of this paper are twofold. First, we contribute by providing a comprehensive framework for inter-case properties, from which relevant properties can be derived for future investigations. Second, we suggest the notions of variant awareness and variant indifference, which can be determined for each property separately.

6 Conclusion

Predicting the remaining time until completion is a central problem in process mining. Recently, remaining time prediction has gained increased attention, and the spectrum of properties considered for the prediction has expanded to include inter-case properties and contextual information. Here, we suggest a global framework for considering these additional properties of context and peers per the four perspectives of control-flow, time, resource, and data, while considering the effect of different case variants.

Over several datasets, we show that inter-case properties should be considered in all perspectives and that it is impossible to determine in advance which perspective would contribute most. We identify for each perspective whether only cases of similar variants should be considered (variant-awareness) or whether all cases hold relevant information (variant-indifference). However, we further show that incorrectly inferring them could degrade the prediction results when the variants are not known in advance.

Our experiments show that inter-case properties do not necessarily improve prediction in all datasets. Our observations point to two possible explanations. First, inter-case properties' relevance increases when process completion is determined by acts of the process owner rather than on exogenous events. Second, that shared resources and physical environment increase the likelihood of inter-case properties to contribute to the prediction.

In future work, we will follow these observations and devise criteria for classifying processes to enable anticipating whether inter-case properties would improve remaining time prediction.

References

1. van der Aalst, W.M.P., Pesic, M., Song, M.: Beyond process mining: from the past to present and future. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 38–52. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13094-6_5
2. van der Aalst, W.M., Rosemann, M., Dumas, M.: Deadline-based escalation in process-aware information systems. Decis. Support Syst. **43**(2), 492–511 (2007)
3. Bolt, A., van der Aalst, W.M.P., de Leoni, M.: Finding process variants in event logs. In: Panetto, H., et al. (eds.) OTM 2017. LNCS, vol. 10573, pp. 45–52. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69462-7_4

4. Buehler, R., Griffin, D., Ross, M.: Exploring the "planning fallacy": why people underestimate their task completion times. J. Pers. Soc. Psychol. **67**(3), 366–381 (1994)
5. Cabanillas, C., Di Ciccio, C., Mendling, J., Baumgrass, A.: Predictive task monitoring for business processes. In: Sadiq, S., Soffer, P., Völzer, H. (eds.) BPM 2014. LNCS, vol. 8659, pp. 424–432. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10172-9_31
6. Conforti, R., de Leoni, M., La Rosa, M., van der Aalst, W.M., ter Hofstede, A.H.: A recommendation system for predicting risks across multiple business process instances. Decis. Support Syst. **69**, 1–19 (2015)
7. De Leoni, M., van der Aalst, W.M., Dees, M.: A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. Inf. Syst. **56**, 235–257 (2016)
8. Di Francescomarino, C., Dumas, M., Maggi, F.M., Teinemaa, I.: Clustering-based predictive process monitoring. IEEE Trans. Serv. Comput. **12**(6), 896–909 (2017)
9. Di Francescomarino, C., Ghidini, C., Maggi, F.M., Milani, F.: Predictive process monitoring methods: which one suits me best? In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) BPM 2018. LNCS, vol. 11080, pp. 462–479. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_27
10. Evermann, J., Rehse, J.R., Fettke, P.: Predicting process behaviour using deep learning. Decis. Support Syst. **100**, 129–140 (2017)
11. Folino, F., Guarascio, M., Pontieri, L.: Discovering context-aware models for predicting business process performances. In: Meersman, R., et al. (eds.) OTM 2012. LNCS, vol. 7565, pp. 287–304. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33606-5_18
12. Ghattas, J., Peleg, M., Soffer, P., Denekamp, Y.: Learning the context of a clinical process. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 545–556. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12186-9_53
13. Ghattas, J., Soffer, P., Peleg, M.: Improving business process decision making based on past experience. Decis. Support Syst. **59**, 93–107 (2014)
14. Kasekamp, T.A.: Web application to support researchers in predictive process monitoring tasks. Master thesis, University of Tartu (2018)
15. Klijn, E.L., Fahland, D.: Identifying and reducing errors in remaining time prediction due to inter-case dynamics. In: 2nd International Conference on Process Mining (ICPM) (2020)
16. Mannhardt, F., Arnesen, P., Landmark, A.D.: Estimating the impact of incidents on process delay. In: 2019 International Conference on Process Mining (ICPM), pp. 49–56. IEEE (2019)
17. Márquez-Chamorro, A.E., Resinas, M., Ruiz-Cortes, A.: Predictive monitoring of business processes: a survey. IEEE Trans. Serv. Comput. **11**(6), 962–977 (2017)
18. Polato, M., Sperduti, A., Burattin, A., de Leoni, M.: Time and activity sequence prediction of business process instances. Computing **100**(9), 1005–1031 (2018). https://doi.org/10.1007/s00607-018-0593-x
19. Senderovich, A., Di Francescomarino, C., Ghidini, C., Jorbina, K., Maggi, F.M.: Intra and inter-case features in predictive process monitoring: a tale of two dimensions. In: Carmona, J., Engels, G., Kumar, A. (eds.) BPM 2017. LNCS, vol. 10445, pp. 306–323. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65000-5_18
20. Senderovich, A., Weidlich, M., Gal, A., Mandelbaum, A.: Queue mining for delay prediction in multi-class service processes. Inf. Syst. **53**, 278–295 (2015)

21. Senderovich, A., Di Francescomarino, C., Maggi, F.M.: From knowledge-driven to data-driven inter-case feature encoding in predictive process monitoring. Inf. Syst. **84**, 255–264 (2019)
22. Soffer, P., Ghattas, J., Peleg, M.: A goal-based approach for learning in business processes. In: Nurcan, S., Salinesi, C., Souveyet, C., Ralyté, J. (eds.) Intentional Perspectives on Information Systems Engineering, pp. 239–256. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12544-7_13
23. Steenken, D., Voß, S., Stahlbock, R.: Container terminal operation and operations research-a classification and literature review. OR Spectr. **26**(1), 3–49 (2004). https://doi.org/10.1007/s00291-003-0157-z
24. Tater, T., Dechu, S., Mani, S., Maurya, C.: Prediction of invoice payment status in account payable business process. In: Pahl, C., Vukovic, M., Yin, J., Yu, Q. (eds.) ICSOC 2018. LNCS, vol. 11236, pp. 165–180. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03596-9_11
25. Taymouri, F., La Rosa, M., Carmona, J.: Business process variant analysis based on mutual fingerprints of event logs. In: Dustdar, S., Yu, E., Salinesi, C., Rieu, D., Pant, V. (eds.) CAiSE 2020. LNCS, vol. 12127, pp. 299–318. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49435-3_19
26. Teinemaa, I., Dumas, M., La Rosa, M., Maggi, F.M.: Outcome-oriented predictive process monitoring: review and benchmark. ACM Trans. Knowl. Discov. Data (TKDD) **13**(2), 1–57 (2019)
27. Teinemaa, I., Dumas, M., Leontjeva, A., Maggi, F.M.: Temporal stability in predictive process monitoring. Data Min. Knowl. Discov. **32**(5), 1306–1338 (2018). https://doi.org/10.1007/s10618-018-0575-9
28. Verenich, I., Dumas, M., La Rosa, M., Maggi, F.M., Teinemaa, I.: Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. ACM Trans. Intell. Syst. Technol. (TIST) **10**(4), 1–34 (2019)
29. Yeshchenko, A., Durier, F., Revoredo, K., Mendling, J., Santoro, F.: Context-aware predictive process monitoring: the impact of news sentiment. In: Panetto, H., Debruyne, C., Proper, H.A., Ardagna, C.A., Roman, D., Meersman, R. (eds.) OTM 2018. LNCS, vol. 11229, pp. 586–603. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02610-3_33

Towards Reliable Business Process Simulation: A Framework to Integrate ERP Systems

Gyunam Park(✉) and Wil M. P. van der Aalst

Process and Data Science Group (PADS), Department of Computer Science, RWTH Aachen University, Aachen, Germany
{gnpark,wvdaalst}@pads.rwth-aachen.de

Abstract. A digital twin of an organization (DTO) is a digital replication of an organization used to analyze weaknesses in business processes and support operational decision-making by simulating different scenarios. As a key enabling technology of DTO, business process simulation provides techniques to design and implement simulation models that replicate real-life business processes. Existing approaches have been focusing on providing highly flexible design tools and data-driven evidence to improve the accuracy of simulation models. Provided with such tools and evidence, business analysts are required to reflect comprehensive aspects of reality with subjective judgments, including the design of ERP systems and the organizational interaction with the system. However, given the complexity of ERP systems, it is infeasible and error-prone to manually incorporate the business logic and data restrictions of the system into simulation models, impeding the faithfulness and reliability of the following analysis. In this work, we propose a framework to integrate ERP systems in business process simulation to overcome this limitation and ensure the reliability of the simulation results. The framework is implemented in *ProM* using the SAP ERP system and CPN Tools.

Keywords: Digital twin · Business process simulation · ERP systems · Business process improvement · Process mining

1 Introduction

A digital twin of an organization (DTO) is a digital representation of business processes and assets across an organization. By simulating this mirrored representation, business analysts can identify operational frictions in the organization with data-based analytics like process mining [1] and evaluate the efficiency of the decisions that are too expensive or dangerous to experiment with, e.g., assigning more resources to a task and increasing the capacity of machines.

Business process simulation is a key enabling technology of DTOs. A simulation model represents reality in a simplified manner and generates hypothetical instances of the business process, enabling the simulation of various scenarios

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 112–127, 2021.
https://doi.org/10.1007/978-3-030-79186-5_8

and "what-if" analysis. Many tools have been developed to support the design and implementation of simulation models, including Arena and CPN Tools [2].

The successful implementation of DTOs using business process simulation relies on how accurately the simulation model represents reality. Especially, given the increasing relevance of Enterprise Resource Planning (ERP) systems in organizations, it is essential to accurately model the business logic and data restriction of ERP systems (e.g., SAP, Salesforce, and Microsoft Dynamics).

Traditional simulation approaches left them to the subjective judgments by domain experts, focusing on providing easy-to-use design tools with high flexibility in the implementation. Despite the flexible design tools, it is infeasible and error-prone to design and implement simulation models reflecting the complex design of ERP systems, making the simulation results unconvincing and unreliable [3]. For instance, the SAP ERP system may produce over 2,821 different errors for violations of the business logic and data restrictions.

The violations mainly happen in two formats: *omission* and *commission* [3]. The former occurs when simulation models fail to reflect the behaviors required by the system. For instance, in the SAP ERP system, each material has a different warehouse management policy, thus having different control flows (i.e., sequences of tasks). Although different control flows should be modeled depending on the policy, simulation models often omit it, resulting in a huge difference in the simulated behaviors. On the other hand, the *commission* problem occurs when simulation models simulate the behaviors not allowed in the system. For instance, ERP systems are established upon data restrictions, limiting arbitrary creations, deletions, and updates of data. For instance, price changes (i.e., updates of price information) are not allowed for specific types of items, even though simulation models often ignore this data restriction.

In this work, we propose a framework to integrate ERP systems to business process simulation to 1) effectively incorporate the complex design of the system into simulations without modeling it in simulation models and 2) provide simulated event data without the omission and commission issues. Figure 1 conceptually describes how the proposed framework integrates ERP systems into business process simulation and how it realizes DTOs.

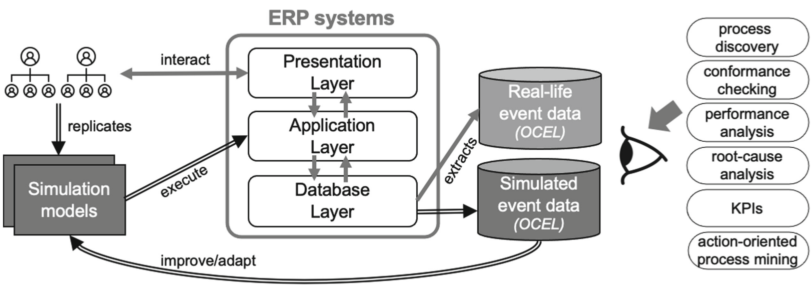


Fig. 1. A conceptual overview of integrating ERP systems in business process simulation

As shown in Fig. 1, ERP systems consist of three layers: representation, application, and database layers. In reality, human resources interact with the system through the representation layer. The commands by human resources are processed according to the business logic defined in the application layer, whose records are stored in the database layer. Event data, e.g., *Object-Centric Event Logs (OCEL)*[1], are extracted from the database layer and used for analysis, e.g., process discovery and root-cause analysis.

In the proposed framework, simulation models, designed by analyzing interactions in the presentation layer, produce commands replicating the behaviors of human resources. We execute the commands directly in the application layer of the ERP system, ensuring the non-existence of the omission and commission problems. The simulated event data are extracted from the database layer in the same manner that real-life event data are extracted. The simulated data may provide feedback to improve the design of the simulation model and adapt the model to changing circumstances.

Since the simulated and real-life data are extracted from the same database in the same manner, they can be analyzed using the same process mining tool. Besides, they support action-oriented process mining [4], which transforms insights from diagnostics to actions, by monitoring the undesired behaviors using real-life data and evaluating the effects of actions using simulated data (e.g., A/B testing between different actions).

To show the effectiveness of the proposed framework, we implement it using the SAP ERP system and CPN Tools. We have tested the implementation by simulating common scenarios in the order-to-cash process of the SAP ERP system and extracting process-centric insights from simulated event data.

The remainder is organized as follows. We discuss the related work in Sect. 2. Next, we present the framework for integrating ERP systems into business process simulation and its implementation in Sect. 3 and Sect. 4, respectively. Section 5 presents the validation of the framework and Sect. 6 concludes the paper.

2 Related Work

Simulation has been adopted for analyzing business processes since the seventies [5]. Nowadays, various simulation tools are available to design and implement simulation models. In [6], the simulation tools are classified into two types: *simulation language* and *simulation package*. The former is a programming language supporting the implementation of simulation models, including Simula and GPSS [7]. The latter is a tool providing graphical building blocks to enable the rapid creation of simulation models such as ARENA and CPN Tools [8].

[1] http://ocel-standard.org/.

Data-driven approaches have been proposed to provide conceptual guidance to design simulation models. Martin et al. [9] identify modeling tasks (e.g., modeling gateways, modeling activities, etc.) and present the relevant process mining techniques to support each of the modeling tasks. In [10], authors utilize the process model discovered using process mining techniques as the reference for the current state of the process and design simulation models with re-engineered business processes by manually identifying the possible improvement points from the reference model.

Furthermore, several techniques have been suggested to automatically discover simulation models from event data. Rozinat et al. [11] discover simulation models based on Colored Petri Nets (CPNs) in a semi-automated manner. Carmargo et al. [12] propose a method to optimize the accuracy of the simulation models discovered from an event log. It searches the space of all possible configurations in the simulation model to maximize the similarity between the discovered simulation model and the event log.

The existing approaches have focused on improving the accuracy by supporting business analysts to better reflect reality in simulation models using domain knowledge and data-driven insights. However, the question still remains: is the simulated behavior the one that can be supported (accepted) by the underlying ERP systems? Since the approaches do not explicitly involve the system in simulations, business analysts are left to ensure it with their subjective judgments. In this work, we tackle this question by proposing a framework to integrate ERP systems in the course of business process simulation.

3 A Framework for the Integration of ERP Systems

The proposed framework consists of three major components: *ERP system*, *simulation engine* and *transformation engine*. The ERP system creates, updates, and deletes objects in *object model* based on *executions*. The simulation engine simulates organizational behaviors in business processes and produces *commands* describing the behaviors. The transformation engine translates the commands into the executions according to which the system updates the object model. The behavior of the simulation may again depend on the updated object model. In the following, we explain the framework with formal definitions and examples.

3.1 ERP System

In this work, we define ERP systems in a narrow sense, focusing on its bookkeeping purpose (i.e., creating, updating, and deleting database tables based on transactions). To this end, we first abstract databases in the system as *object models*.

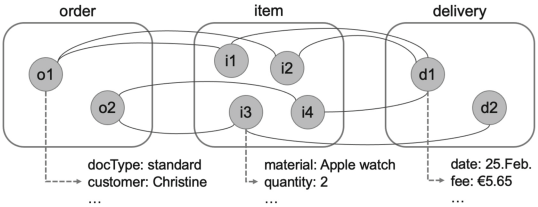


Fig. 2. We use object models to mimic the databases in ERP systems

As described in Fig. 2, an object model contains objects (e.g., *o1*, *o2*, and *i1*) with different types (e.g., *order*, *item*, and *delivery*). Also, the objects have relationships (e.g., *o1* is related to *i1*). Besides, each object involves attribute values (e.g., order has *docType* and *customer* information). We formally define object models as follows:

Definition 1 (Object Model). *Let $\mathbb{U}_o$ be the universe of object identifiers, $\mathbb{U}_{ot}$ the universe of object types, $\mathbb{U}_{attr}$ the universe of attribute names, and $\mathbb{U}_{val}$ is the universe of attribute values. An object model is a tuple $OM = (O, OT, OR, otyp, oval)$ where*

- *$O \subseteq \mathbb{U}_o$ is a set of object identifiers,*
- *$OT \subseteq \mathbb{U}_{ot}$ is a set of object types,*
- *$OR \subseteq O \times O$ is a set of object relationships,*
- *$otyp \in O \rightarrow OT$ assigns precisely one object type to each object identifier, and*
- *$oval : O \rightarrow (\mathbb{U}_{attr} \nrightarrow \mathbb{U}_{val})$ is the function associating an object to its attribute value assignments. We denote $oval(o)(attr) = \perp$ if $attr \notin dom(oval(o))$ for any $o \in O$.*

$\mathbb{U}_{om}$ denote the set of all possible object models.

For instance, $OM_1 = (O, OT, OR, otyp, oval)$ describes the object model described in Fig. 2, where $O = \{o_1, o_2, i_1, \dots\}$, $OT = \{order, item, delivery\}$, $OR = \{(o_1, i_1), (o_1, i_2), (i1, d1), \dots\}$, $otyp(o_1) = order$, $otyp(i_1) = item$, $oval(o_1)(docType) = standard$, $oval(o_1)(customer) = Christine$, etc.

Definition 2 (Transaction). *Let $\mathbb{U}_{pval} \subseteq \mathbb{U}_{attr} \nrightarrow \mathbb{U}_{val}$ be the universe of parameter value assignments. A transaction $tr \in \mathbb{U}_{om} \times \mathbb{U}_{pval} \rightarrow \mathbb{U}_{om}$ is a function that modifies object models based on parameter value assignments. $\mathbb{U}_{tr}$ denotes the set of all possible transactions.*

For instance, $po \in \mathbb{U}_{tr}$ is a transaction that places an order. Assume that $pval_1 \in \mathbb{U}_{pval}$ is a parameter value assignment such that $pval_1(customer) = Marco$, $pval_1(docType) = quoted$, $pval_1(item) = iPad$, $pval_1(quantity) = 1$, etc. $po(OM_1, pval_1) = (O', OT, OR', otyp', oval')$ such that $o_3, i_5 \in O'$, $(o_3, i_5) \in OR'$, $otyp'(o_3) = order$, $otyp'(i_5) = item$, $oval'(o_3)(customer) = Marco$, etc.

An execution specifies a transaction to be executed in the system along with the execution time, responsible resource, and parameter value assignment.

Definition 3 (Execution). *Let* $\mathbb{U}_{res}$ *be the universe of resources and* $\mathbb{U}_{time}$ *the universe of timestamps. An execution* $exec \in \mathbb{U}_{tr} \times \mathbb{U}_{res} \times \mathbb{U}_{time} \times \mathbb{U}_{pval}$ *is a tuple of a transaction, a resource, timestamp, and a parameter value assignment.* $\mathbb{U}_{exec}$ *denotes the set of all possible executions.*

For instance, $exec_1 = (po, Adams, \text{10:00 23.02.2021}, pval_1)$ describes the order placement by $Adams$ at 10:00 23.02.2021 with the parameter $pval_1$.

Definition 4 (ERP System). *An ERP system* $sys \in \mathbb{U}_{om} \times \mathbb{U}_{exec} \rightarrow \mathbb{U}_{om}$ *updates object models according to executions.*

For instance, $sys(OM_1, exec_1) = (O', OT, OR', otyp', oval')$ such that $o_3, i_5 \in O'$, $oval'(o_3)(timestamp) = \text{10:00 23.02.2021}$, $oval'(o_3)(resource) = Adams$, etc.

3.2 Simulation Engine

A simulation engine aims at producing commands describing which activity is done by whom at what time with what information. We formally define commands as follows:

Definition 5 (Commands). *Let* $\mathbb{U}_{act}$ *be the universe of activities. A command* $cmd \in \mathbb{U}_{act} \times \mathbb{U}_{res} \times \mathbb{U}_{time} \times \mathbb{U}_{pval}$ *is a tuple of an activity, resource, timestamp, and information value assignment.* $\mathbb{U}_{cmd}$ *is the set of all possible commands.*

For instance, $cmd_1 = (place\_order, Adams, \text{10:00 23.02.2021}, ival_1) \in \mathbb{U}_{cmd}$, where $ival_1(customer) = Marco$ and $ival_1(price) = €100$, clones the behavior of $Adams$ who places an order of €100 by $Marco$ at 10:00 23.02.2021.

Since simulating behaviors of human resources (i.e., generating commands) share commonalities to simulating events of business processes (e.g., placing order occurred at 10:00 23.02.2021 by $Adams$), we can deploy techniques for business process simulation to generate commands.

As presented in Sect. 2, various simulation tools are available for this purpose such as CPN Tools and ARENA. Remaining tool-independent, we focus on explaining essential components of simulation models to produce commands. In Sect. 4, we explain how these components are designed and implemented using CPN Tools.

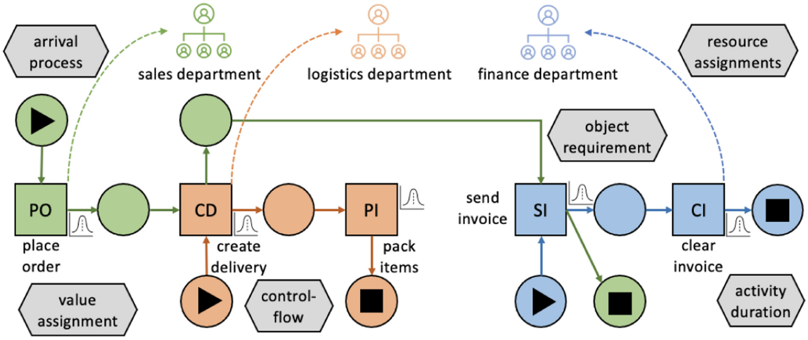


Fig. 3. Core components of business process simulation [6]

Figure 3 explains the core ingredients of business process simulation including *control-flows*, *resource assignments*, *arrival processes*, and *activity duration*, *object assignments* and *object value assignments*. Below is the explanation of each component:

- **Activities** represent the behaviors that human resources do to serve business processes (e.g., place order, send invoice, etc.).
- **Control-flows** determine the sequence of activities. For instance, the process model in Fig. 3 describes one sequence of activities, i.e., *send_ quotation*, *place_ order*, *create_ delivery*, *pack_ items*, *create_ invoice*, and *clear_ invoice* in order.
- **Object requirements** explain the required objects for the execution of activities. Each activity may involve multiple objects in its execution. For instance, the execution of *create_ delivery* involve an order and a delivery since the order information is required to create the delivery.
- **Value assignment** specifies how the execution of activities updates information of the involved objects. For instance, the execution of *place_ order* updates the document and customer information of the order.
- **Resource assignments** define who is eligible to perform activities in the business process. For instance, *place_ order* can be performed by the resources from a sales department.
- **Arrival processes** and **activity duration** define the inter-arrival time between arrivals and the duration required for the execution of activities, respectively.

3.3 Transformation Engine

The goal of the transformation engine is to convert the commands to the executable formats supported in the underlying ERP system (i.e., executions). To this end, we need two components: *transaction mapping* and *parameter mapping*. First, the transaction mapping translates the activity in commands to the transaction defined in ERP systems.

Definition 6 (Transaction Mapping). *A transaction mapping* $\mu_{tr} \in \mathbb{U}_{act} \rightarrow \mathbb{U}_{tr}$ *relates transactions to activities.*

Assume *place_order* in cmd_1 corresponds to $po \in \mathbb{U}_{tr}$ in an ERP system. A transaction mapping μ'_{tr} connects them, i.e., $\mu'_{tr}(place\_order) = po$.

Next, parameter mapping connects the parameters in commands to the system-defined parameters in ERP systems.

Definition 7 (Parameter Mapping). *A parameter mapping* $\mu_{pr} \in \mathbb{U}_{attr} \rightarrow \mathbb{U}_{attr}$ *relates system-defined parameters of an ERP system to parameters in commands.*

Assume $dom(ival_1) = \{docType, customer\}$ where $(place\_order, Adams, t_1, ival_1) \in \mathbb{U}_{cmd}$ and DOC_TYPE and PART_NUMB are the corresponding parameters defined in the system. A parameter mapping μ'_{pr} connects the parameters, i.e., $\mu'_{pr}(\text{docType}) = \text{DOC\_TYPE}$ and $\mu'_{pr}(\text{customer}) = \text{PART\_NUMB}$.

Given transaction and parameter mapping, a transformation engine transforms commands to executions by translating transactions and parameters. produces the executions. Note that we assume that the resource/timestamp in commands is compatible with the one in executions.

Definition 8 (Transformation Engine). *Let* μ_{tr} *be a transaction mapping and* μ_{pr} *a parameter mapping. A transformation engine maps executions onto commands, i.e., for any* $cmd = (act, res, time, ival) \in \mathbb{U}_{cmd}$*,* $trans(\mu_{tr}, \mu_{pr})(cmd) = (\mu_{tr}(act), res, time, pval)$ *such that* $\forall_{attr \in dom(ival)}\ pval(\mu_{pr}(attr)) = ival(attr)$*.*

4 Implementation

In this section, we implement the proposed framework using the SAP ERP system as the underlying ERP system. We design and implement simulation models using *CPN Tools*[2]. The transformation engine is implemented as a plug-in in *ProM*[3] and translates commands into executions in the SAP ERP system given a transaction and parameter mapping.

4.1 ERP System: *SAP ERP ECC 6.0*

The SAP ERP system is the most widely-adopted ERP system, supporting more than 400,000 businesses worldwide. In the implementation, we utilize *SAP ERP ECC 6.0*[4] supporting Global Bike Inc., an imaginary enterprise producing and distributing bicycle products where all relevant SAP solutions are represented.

[2] https://cpntools.org/.
[3] http://www.promtools.org.
[4] https://www.sap.com/.

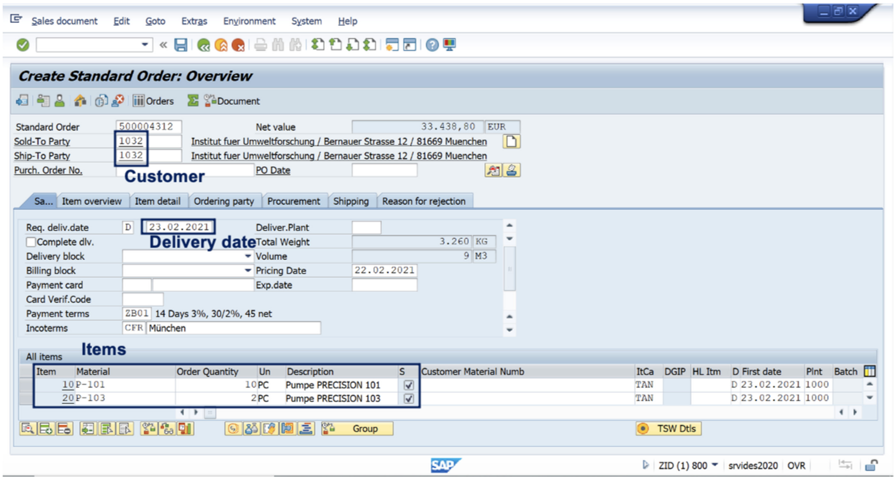


Fig. 4. A screenshot of the user interface in the SAP ERP ECC 6.0 (place an order)

Figure 4 shows the user interface in the representation layer where sales staff places orders. Given the inputs such as customer, delivery date, and items (i.e., parameters) by users (i.e., resources), the transactions in the application layer (e.g., BAPI_SALESORDER_CREATEFROMDAT2) are executed to update the database supported by Oracle Database Management System (i.e., object models).

4.2 Simulation Engine: *CPN Tools*

CPN Tools is a toolset providing support for editing, simulating, and analyzing Colored Petri Nets (CPNs). For the detailed information of *CPN Tools*, we refer readers to [8].

In the following, we explain how *CPN Tools* is used to implement the core ingredients of business process simulation introduced in Subsect. 3.2 with the example described in Fig. 3. Note that there exist various design choices resulting in different executable models in *CPN Tools*.

Figure 5 shows the overall composition consisting of multiple CPN pages, i.e., *overview page*, *environment page*, *process page*, and *activity pages*.

The overview page connects the environment page, process page, and resource pool. The environment page describes the **arrival process** implemented as a negative exponential distribution. A simulation instance is generated according to the arrival process and passed into the process page.

In the process page, relevant objects for the simulation instance are generated by transition "generate object". In our example, each simulation instance associates an order, a delivery, and an invoice. The transitions, including "PO", "CD", "PI", and "SI", represent **activities** in the process.

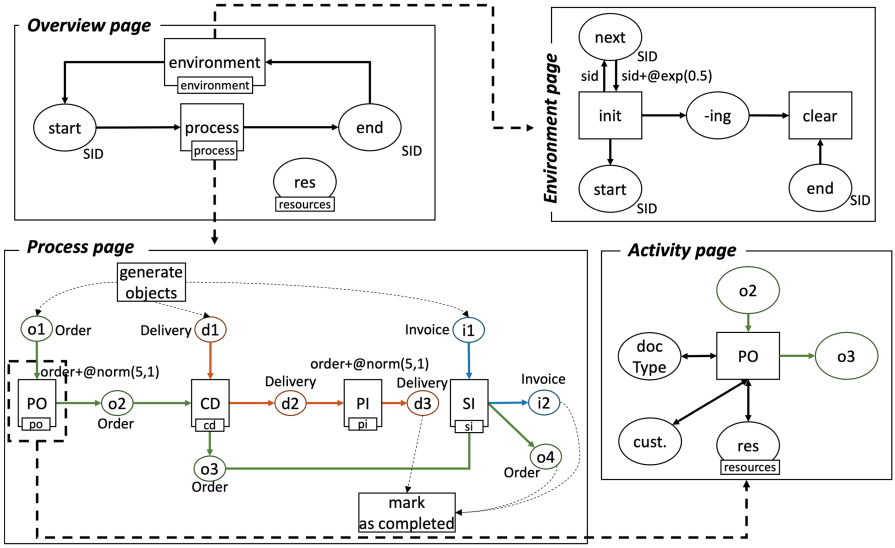


Fig. 5. A schematic overview of the CPN used to implement our simulation framework

The **object requirement** for the execution of an activity is indicated with the incoming arcs from the places representing object types to the corresponding transition. For instance, "create delivery" involves an order and a delivery (i.e., incoming arcs from the place for order type (i.e., o_1) and the place for delivery (i.e., d_1) to "create delivery").

Control-flows are represented using the semantics of CPNs, i.e., a transition is triggered by consuming tokens from its input places and producing tokens to its output places. In our example, *place_ order* is triggered first by consuming a token at o_1 and producing a token at o_2. Next, "create delivery" is triggered by consuming tokens from o_2 and d_1 and producing tokens at o_3 and d_2.

Each transition has a sub-page (i.e., activity page) where **resource assignments** and **value mappings** are modeled. First, in each execution of the transition, a resource is selected from the resource pool based on the role. Next, the relevant information for the execution of the activity (e.g., the customer and document type in *place_ order*) is passed by the tokens from the connected places.

Activity duration is implemented as the timed property of *CPN Tools*. The activity duration is inscribed on the transition. For instance, the duration of the *place_ order* activity is populated from a normal distribution.

We generate commands using the designed CPNs. Below is an example of the commands in XML-based *CMD* formats. In $(act, res, time, ival) \in \mathbb{U}_{cmd}$, Lines 4–6 correspond to *act*, *res*, and *time*, while Lines 7–12 specify *ival*.

Listing 1. An example of *CMD* format

```
<?xml version="1.0" encoding="UTF-8"?>
<commands>
    <command>
        <activity>place_order</activity>
        <resource>Adams</resource>
        <timestamp>2021-02-23 10:00:00</timestamp>
        <orderId>500004312</orderId>
        <customer>1032</customer>
        <docType>TA</docType>
        <salesOrg>1000</salesOrg>
        <materialList>P-101,P-103</materialList>
        <quantityList>6,5</quantityList>
    </command>
    ...
</commands>
```

4.3 Transformation Engine: ProM Plug-In

The transformation engine is implemented as a plug-in of *ProM*, an open-source framework for the implementation of process mining tools. Our new plug-in is available in a new package named *ERPSimulator* in the nightly build of *ProM*. The main input objects of the plug-in are transaction mapping, parameter mapping, and commands, whereas the outputs are SAP executions.

The transaction mapping is stored as an XML-based *AMAP* format, storing relations between activities and transactions. Below is an example of the transaction mapping for the commands generated by the simulation engine described in Fig. 5. Four activities in the simulation engine are assigned to corresponding transactions defined in the SAP ERP system.

Listing 2. An example of *AMAP* format

```
<?xml version="1.0" encoding="UTF-8"?>
<transactionMapping>
    <string key="place_order" value="BAPI_SALESORDER_CREATEFROMDAT2"/>
    <string key="create_delivery" value="BAPI_OUTB_DELIVERY_CREATE_SLS"/>
    <string key="pack_items" value="L_TO_CREATE_DN"/>
    <string key="create_invoice" value="BAPI_BILLINGDOC_CREATEMULTIPLE"/>
</transactionMapping>
```

The parameter mapping is stored as an XML-based *PMAP* format, storing relations between the parameter in commands and the system-defined parameter. Below is an example of the parameter mapping for the commands produced by the simulation engine described in Fig. 5. In Line 3–4, “docType” and “customer” are matched into DOC_TYPE and PARTN_NUMB in the SAP ERP system.

Listing 3. An example of *PMAP* format

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <parameterMapping>
3     <string key="docType" value="DOC_TYPE"/>
4     <string key="customer" value="PARTN_NUMB"/>
5     <string key="orderId" value="SALESDOCUMENTIN"/>
6     <string key="salesOrg" value="SALES_ORG"/>
7     ...
8 </parameterMapping>
```

Given the transaction and parameter mapping, the transformation engine translates commands in *CMD* format into SAP Remote Function Calls (RFCs) that can be directly executed in the system. The SAP RFC is an SAP interface protocol managing the communication between the SAP ERP system and any external system. For instance, the command in Listing 1 is translated to the RFC specifying BAPI_SALESORDER_CREATEFROMDAT2, as defined in Listing 2, with the parameters such as DOC_TYPE and PARTN_NUMB, as described in Listing 3.

5 Proof of Concept

In this section, we validate the feasibility of the proposed framework in generating simulated event data that contain no omission and commission problems (i.e., reliable) and have the same view as the real-life event data (i.e., realistic) without having to manually model the complex design of ERP systems in simulation models. To this end, we simulate common business challenges in the order-to-cash (O2C) process using the implementation presented in Sect. 4. The CPN files and commands are publicly available via https://github.com/gyunamister/ERPSimulator, as well as the user manual.

5.1 Experimental Design

The O2C process deals with customer orders. First, customers send inquiries and, in return, the company sends corresponding quotations. Sales staff converts the quotations into orders if the customers confirm them. Afterward, deliveries are prepared by picking up and packing the items of the orders. Next, invoices are sent to customers and the corresponding payments are collected.

In the following, we simulate common business challenges in the O2C process using the implementation presented in Sect. 4, i.e., 1) low conversion rate, 2) frequent price change, and 3) order cancellation. In each scenario, we evaluate if simulated data have omission and commission problems by measuring the number of executions with errors using the error handling module in the SAP system. Besides, we apply process mining techniques, such as process discovery and root-cause analysis, to simulated event data to show that they have the same view as real-life event data, containing insightful knowledge.

5.2 Scenario 1: Low Conversion Rate

The low conversion rate from quotations to orders is not desired because not only of the lost revenue but also of the waste of resources. We design and implement the simulation model where quotations are not converted to orders mostly due to the late response to the corresponding inquiry using *CPN Tools*. 288 commands are generated by the simulation model and transformed into 288 RFCs using the *ProM* plug-in. Among the 288 RFCs, 288 are successfully executed in the system. As a result, 286 objects of 8 different object types, including inquiry, quotation, order, etc., are created, updating more than 11 tables in the database.

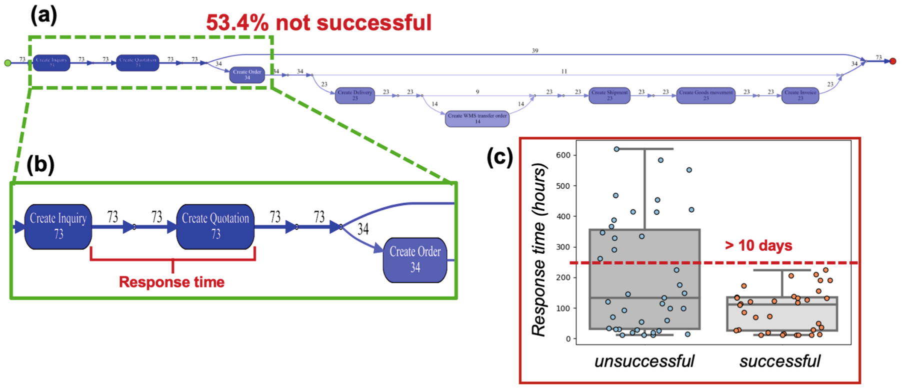


Fig. 6. (a) a discovered process model of the O2C process in BPMN notation, (b) a low conversion from quotations to orders (53.4%), (c) a box plot showing the correlation between the response time and (un)successful conversions

We analyze the behaviors in the business process using the Inductive Visual Miner in *ProM* [13]. Figure 6(a) describes the process model in BPMN notation. As shown in Fig. 6(b), only 34 out of 73 quotations from the company are converted to orders, showing the conversion rate of 46.6%. We define the response time as the time difference between the completion of "create inquiry" and "create quotation". Figure 6(c) shows the difference in the response time between the successful and unsuccessful confirmed orders. Especially, the quotations that are responded to later than 10 days are all rejected, showing the correlation between the response time and unsuccessful conversion.

5.3 Scenario 2: Manual Price Changes

Due to different reasons (e.g., outdated pricing policy in the system), manual changes in prices are carried out. We design this scenario in *CPN Tools* and produce 4,249 commands that are transformed into 4,249 RFCs by the *ProM* plug-in. All of the 4,249 RFCs are successfully executed in the system without errors, creating 4,093 objects and updating more than 15 tables in the database.

Figure 7(a) depicts the process model discovered using the process discovery technique. As shown in Fig. 7(b), for 113 orders out of 402 orders, the manual price change occurred. Figure 7(c) describes manual price changes per product (e.g., *P-100* and *P-101*). The outer pie indicates the total number of orders per product, while the red part in the inner pie represents the proportion of the changes for each product. For instance, *P-109* is included in 138 orders and 79% of them require manual changes.

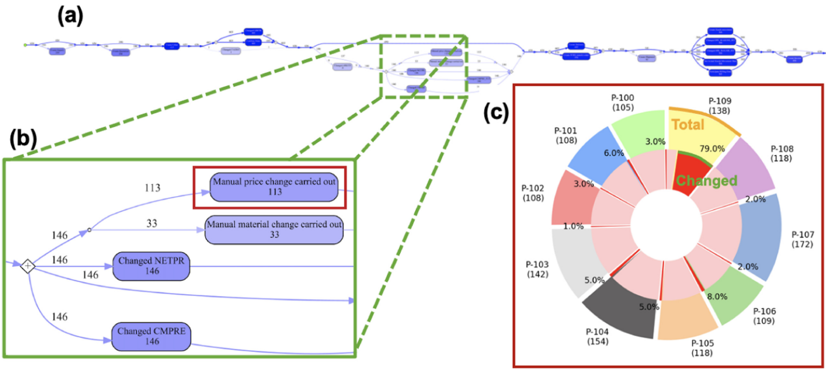


Fig. 7. (a) a discovered process model, (b) manual price changes required for 113 orders, (c) a pie chart describing the ratio of price changes to total orders per products

5.4 Scenario 3: Frequent Cancellation of Orders

For the frequent cancellation of orders, we first generate 4,540 commands using *CPN Tools* and transform them into 4,540 SAP RFCs using the *ProM* plug-in. We successfully execute the 4,540 RFCs without errors and, accordingly, 4,384 objects of 8 different object types are created.

Figure 8 shows the process model discovered with the process discovery technique. As shown in Fig. 8(b), 97 out of 562 orders are canceled in the process. We conduct further analysis on these canceled orders by analyzing the reasons for the cancellation. Figure 8(c) shows the pie chart explaining the proportion of different reasons. The most frequent reason is the delivery date set too late.

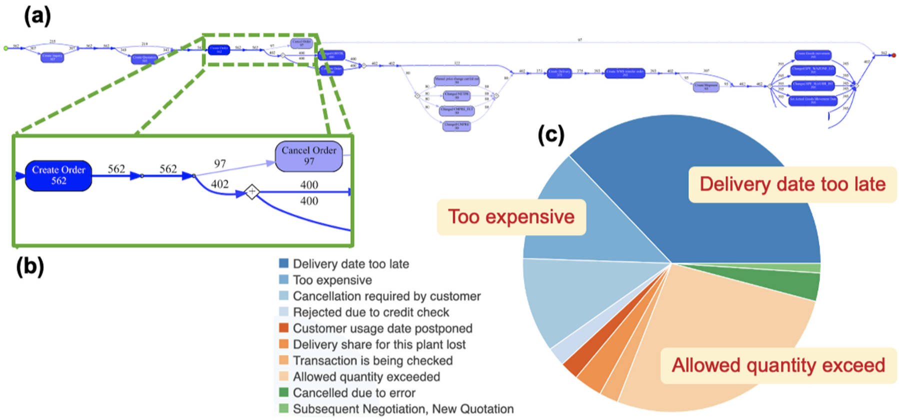


Fig. 8. (a) a discovered process model, (b) order cancellations (97 out of 462 orders), (c) a pie chart depicting the frequency of reasons for the order cancellation

The second most frequent reason is that the order exceeds the quantity limit of one of its items, followed by the high price of the order.

6 Conclusion

In this paper, we proposed the framework for integrating ERP systems into business process simulation to realize DTOs. The framework is composed of three components: the ERP system, simulation engine, and transformation engine. Commands are generated by the simulation engine and translated to system-executable formats by the transformation engine. The executions are applied to the system to update the object model in the system. The framework is implemented using the SAP ERP system as the underlying ERP system, CPN Tools as the simulation engine, and a ProM plug-in as the transformation engine.

By integrating ERP systems, we can effectively reflect the complex design of the system into simulation models. Moreover, the resulting simulated data have no omission and commission issues, ensuring the reliability of simulation results. Also, having the same data structure as the real-life event data, the simulated event data can be analyzed by existing analysis techniques. Furthermore, it supports action-oriented process mining by providing a digital twin where different actions can be implemented and tested.

As future work, we plan to improve the implementation to support the feedback loop between the simulated data and simulation engine. In addition, we plan to deploy the implementation to the techniques for action-oriented process mining to evaluate the efficiency of actions. In the proposed framework, we assume a one-to-one relationship between activities (i.e., human behaviors) and transactions. However, in real-life business processes, a single human behavior may involve multiple transactions and vice versa. In this work, we resolve the issue in the implementation by manually aligning the level of simulated human behaviors to the level of transactions. Future work should present a method to support the resolution of the many-to-many relationship.

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

References

1. van der Aalst, W.M.P.: Data science in action. In: Process Mining, pp. 3–23. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4_1
2. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-662-56509-4
3. Rashid, A., Tjahjono, B.: Achieving manufacturing excellence through the integration of enterprise systems and simulation. Prod. Plann. Control **27**(10), 837–852 (2016)

4. Park, G., van der Aalst, W.M.P.: A general framework for action-oriented process mining. In: Del Río Ortega, A., Leopold, H., Santoro, F.M. (eds.) BPM 2020. LNBIP, vol. 397, pp. 206–218. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-66498-5_16
5. Shannon, R., Johannes, J.D.: Systems simulation: the art and science. IEEE Trans. Syst. Man Cybern. **SMC-6**(10), 723–724 (1976)
6. van der Aalst, W.M.P.: Business process simulation survival guide. In: vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management 1. IHIS, pp. 337–370. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-642-45100-3_15
7. Dahl, O.J., Nygaard, K.: SIMULA: an ALGOL-based simulation language. Commun. ACM **9**(9), 671–678 (1966)
8. Jensen, K., Kristensen, L.M., Wells, L.: Coloured petri nets and CPN tools for modelling and validation of concurrent systems. Int. J. Softw. Tools Technol. Transfer **9**(3), 213–254 (2007)
9. Martin, N., Depaire, B., Caris, A.: The use of process mining in business process simulation model construction: structuring the field. Bus. Inf. Syst. Eng. **58**(1), 73–87 (2016)
10. Măruşter, L., van Beest, N.R.T.P.: Redesigning business processes: a methodology based on simulation and process mining techniques. Knowl. Inf. Syst. **21**(3), 267–297 (2009)
11. Rozinat, A., Mans, R., Song, M., van der Aalst, W.M.P.: Discovering simulation models. Inf. Syst. **34**(3), 305–327 (2009)
12. Camargo, M., Dumas, M., González-Rojas, O.: Automated discovery of business process simulation models from event logs. Decis. Support Syst. **134**, 113284 (2020)
13. Leemans, S., Fahland, D., van der Aalst, W.M.P.: Process and deviation exploration with inductive visual miner. In: Limonad, L., Weber, B. (eds.) BPM Demo Sessions, pp. 46–50 (2014)

Modeling Languages and Reference Models (BPMDS 2021)

Supporting Data-Aware Processes with MERODE

Monique Snoeck(✉), Johannes De Smedt, and Jochen De Weerdt

KU Leuven, Leuven, Belgium
{monique.snoeck,johannes.desmedt,jochen.deweerdt}@kuleuven.be

Abstract. Most data-aware process modelling approaches have been developed from a process perspective and lack a full-fledged data modelling approach. In addition, the evaluation of data-centric process approaches reveals that, even though their value is acknowledged, their usability is a point of concern. This paper presents a data-aware process modelling approach combining full-fledged domain modelling based on UML class diagrams and state charts with BPMN. The proof-of-concept has been implemented using the MERODE code generator, linking the resulting prototype application to a Camunda BPM engine, making use of RESTful web-services. The proof of concept is evaluated against 20 requirements for data-aware processes and demonstrates that the majority of these are already satisfied by this out-of-the-box prototyping approach.

Keywords: Conceptual modelling · Process modelling · Data-aware processes · Model-driven engineering

1 Introduction and Motivation

For many years, data modelling, process modelling, and decision modelling have evolved as largely separate worlds, focusing on the respective modelling languages and methods, having different communities, conferences, and publication outlets [1]. While this "separation of concerns" allows focusing on the particularities of each domain, such silo-based approach comes with drawbacks as well. Data and processes are two concerns that underly different architectural viewpoints of a same system and integration is thus required to ensure consistency and correctness [2]. Architectural descriptions should come with defined correspondences and ensuing correspondence rules to express, record, enforce and analyse consistency between models and views. From an enterprise engineering perspective, defining the essential business concepts and their relationships through domain modelling and defining how the business operates through process modelling, should go hand in hand. Both perspectives should be aware of and integrated with the other perspective.

In recent years the importance of data aspects has been acknowledged by the process modelling community, and several approaches have been proposed, see [3–5] for overviews. Most of this research was initiated by experts from the process modelling domain, focusing on how to make processes data-aware, e.g. through case-based

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 131–146, 2021.
https://doi.org/10.1007/978-3-030-79186-5_9

approaches [6], artefact-centric approaches [7, 8], object-centric approaches [9], developing connections to a database [10], or focusing on developing support for verifying process properties such as safety, liveness, etc., see for example [11, 12].

While research on data-ware processes provides progress towards an integrated approach, how data is addressed largely varies between approaches [3]. A full-fledged domain modelling approach focuses on defining business objects and their associations so as to provide an enterprise-wide definition of business concepts, as a common language shared by all business domains, and hence all business processes. A global perspective on the relationship between process modelling and domain modelling is still missing (e.g. in terms of an integrated meta-model), as well as a practical approach for modelers on how to tackle the balance between process modelling and domain modelling: what should come first, how are the models related to each other, and how do modelling decisions in one of the views affect the other view.

The goal of this paper is to investigate a data-aware process modelling approach that assumes the existence or joint development of a full-fledged domain model. In particular, the MERODE modelling method provides an approach to domain modelling [13] based on the Unified Modelling Language (UML), and formally grounded in process algebra [14]. While the MERODE-approach captures behavioural aspects through object lifecycle modelling and object interactions, it nevertheless also suggests the use of a business process layer to handle user and work-related aspects. Combining MERODE with process modelling results in data-aware process modelling, but -as opposed to most current approaches- the domain modelling is considered in its own right, rather than in function of process modelling. This paper contributes to the current state of the art by 1) providing a data-ware process modelling approach that relies on full-fledged domain modelling, 2) providing a concrete proof of concept for this suggested combination and 3) evaluating the resulting approach against the criteria presented in [4].

The remainder of this paper is structured as follows. Section 2 presents the state of the art on research that combines the process and data perspective. Section 3 describes the proposed approach based on the running example of [4]. Section 4 presents a detailed evaluation of the approach along the criteria defined in [4]. Section 5 presents a discussion and Sect. 6 concludes the paper.

2 Related Work

In 2019, a systematic literature review on data-aware process modelling covering the period up till 2016 was published [3]. This review identified 17 different approaches to data-centric process modelling, described in 38 primary studies. While 13 papers relate to the Artefact-Centric approach proposed in [7], many other approaches have been developed as well. The results of this literature review also show that nearly each of the identified approaches have defined their own particular data representation construct. While some could be unified under the denominator of "Object" or "Entity", there still remains quite a large variation, and chosen constructs may not map to standard conceptual data modelling practices such as entity-relationship modelling or conceptual UML class diagrams. For example, certain approaches work on unstructured data like documents [15], others use Petri Nets to represent data [16]. As the authors state "*a* *general understanding of the inherent relationships that exist between processes and data is still missing*" [3].

Running the same query again in Web of Science and Scopus for the period 2017–2020 yielded 9 unique papers, 5 of them addressing an aspect of the artefact-centric approach (e.g. [17, 18]) or a specific subtopic of data and process integration like consistency, instance migration, the use of ontologies or process adaptation (e.g. [19]). No fundamentally new approach has been proposed.

A major drawback of some data-aware process modelling approaches is that data is often considered on a per-process basis (e.g. by only modelling the data relevant for the process at hand, see language requirement 3 in [20], or [19]). In some approaches a global domain model is considered as a given, and data-awareness mainly resides in bridging the process model to an existing data model, e.g. by developing a data querying and manipulation language to allow for data-aware process execution such as DAPHNE [10]. In [21] the notion of Artefact acts as a collection of process variables to be associated to a process instance, and serves as interface between the process model and the classes in a pre-defined data model. While providing a practical solution to process execution, this does not constitute a fully data-aware process modelling approach, where process models are inherently aware of the enterprise-wide conceptual data model of the domain in which they operate [22].

Process-aware domain modelling on the other hand, seems a largely unexplored topic. In object-oriented (OO) conceptual modelling (as e.g. in OO-Method [23] and MERODE [13]) business objects can have a state chart imposing sequences on the invocation of an object's low-level methods that manipulate its data. Business process modelling is absent or not fully elaborated. Artefact-centric modelling (e.g. [7, 24]) equips business artefacts with a lifecycle, and considers that the business processes result from the composition of services, which are associated to the business artefacts and their lifecycles through associations. Both in the OO approach and in artefact-centric approaches, object lifecycles capture behavioural aspects on a per-object/artefact basis, but are not meant to address the user perspective and defining work organisation as business processes, which was one of the motivations behind the PHILharmonicFlows approach [9].

In terms of integrating the process and data perspective, a significant amount of research has been performed in consistency verification, e.g. [12, 25, 26]. While formal verification may provide useful support for modellers to verify their work, most of the approaches are formal, not intuitive nor practical from a business point of view [12]. Even the most practical approach does not come with a priori guidelines providing modellers intuitive insights in the relationship between constraints embodied by the conceptual data model and those included in the process model. The survey published in [5] reveals that even though the value of data-centric approaches is acknowledged, their usability remains a point of concern. Moreover, as previous research has demonstrated UML class diagrams and BPMN to be practitioners' favourite languages [27], it makes sense to look for a solution based on UML and BPMN.

3 Integrating Process and Domain Modelling

3.1 Architectural Layers

Combining data and process modelling boils down to a multi-modelling approach, where each model captures a specific viewpoint of the architecture [2]. Typical viewpoints are:

- VP1 - the data or business objects viewpoint, addressing the information that a business creates and maintains;
- VP2 - the business object behaviour viewpoint, addressing the relevant states in the life of a business object, from its creation to its final disposition and archiving;
- VP3 - the shared services viewpoint, describing how a service may provide access to information or perform changes to one or more business objects;
- VP4 - business process behaviour viewpoint addressing units of work and how these are combined to coarser-grained processes and governed by constraints such as task precedence;
- VP5 - business actor viewpoint, addressing the distribution of work across actors;

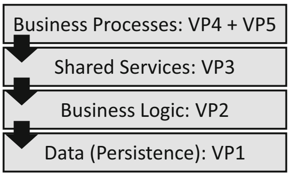


Fig. 1. Software layers

A good practice from a software architecture perspective, is to organize software into layers. Typically, layers address specific viewpoints, and layers implementing stable aspects of a system are positioned in the kernel of the software architecture, whereas elements with higher needs for flexible adaptation should be implemented in outer layers [28]. Business processes are typical examples of elements with a higher need for flexible adaptability, whereas the data layer tends to be more stable. Above-mentioned viewpoints would typically be arranged as shown in Fig. 1. Current data-aware process approaches do not address all these viewpoints explicitly. And while it may be useful to allow bypassing layers (e.g. for performance), it is a good practice to avoid direct access to a database and instead install intermediate services layers (VP3) to isolate the business process layer from the persistence layer [28]. Many current data-aware process approaches however, let business process activities directly access the data layer, thus skipping the shared service layer (VP3). Artefact-centric approaches do not have a separate process layer. In BALSA [7], the Business Artefacts and the Lifecycles address VP1 and VP2. VP4 is addressed by the Services that define units of work, and the Associations that may define constraints governing the services' access to artefacts thus defining (among others) precedence relationships between services. VP3 and VP5 are not addressed. In BAUML [21], the class diagram and state charts address VP1 and VP2. Activity Diagrams address the associations and OCL is used to define contracts for services. This allows addressing aspects of VP4. VP3 and VP5 are not addressed. In approaches that combine process modelling with access to data (e.g. [10]), the process model addresses VP4 and VP5 and the data model captures the data viewpoint (VP1). Artefact behaviour (VP2) is not captured. PHILharmonicFlows [9] combines a data model (VP1) with Object Life Cycles (VP2) that define micro-processes, and defines macro processes too (VP4). Authorisations address VP5. VP3 is not addressed.

3.2 Layers in the MERODE Approach

The MERODE method follows the principles of layers and identifies three major layers (Fig. 2): the Enterprise layer (EL) is the bottom layer, the Business Process layer (BPL) is the top layer and in between sits an Information System Services layer (ISL). The Enterprise layer (EL) itself contains two sublayers. Business Objects are stored in the domain layer (DL). Additional logic is defined in the Object Life Cycles (OLCs). Transitions in OLCs are triggered by events, in MERODE called "Business Events". An Event-Handling Layer (EHL) offers an interface to invoke events and routes these to the relevant Business Objects that will handle the event by means of a corresponding operation effecting the required state changes. In between the EL and BPL sits the Information System Services layer (ISL) offering shared input and output services to access the EL. Output services allow querying the attributes and states of business objects. Input services capture input data but do not directly invoke operations on business objects. Rather, they achieve the requested operations by triggering one or several business events via the EHL. The business events and their handling through an EHL allows combining the advantages of an event-driven architecture with the advantages of the layered architecture, while also managing the transitioning to consistent states [29].

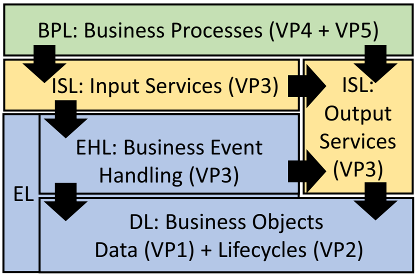


Fig. 2. MERODE layers: BPL (green), ISL (yellow), EL (blue) with two sublayers: EHL and DL. (Color figure online)

The use of Business Events and an intermediate Event Handling layer is an important distinctive characteristic of the MERODE approach. Whereas usually a business process task's operational logic is defined in terms of SQL operations [10] or micro-processes defining read and write accesses to objects' attributes [9] (Fig. 3 left), in MERODE, the connection between the business process layer and the domain layer (or database layer) happens through the intermediary of input and out services and business events (Fig. 3, right). Input services can be kept simple or can incorporate logic that is reusable across different variants of similar tasks. Where to put what logic in view of balancing flexibility against business logic enforcement is discussed in [13], chapter 10.

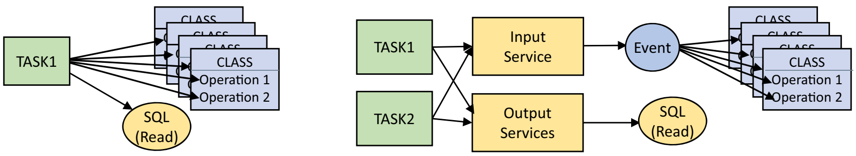


Fig. 3. Connecting the BPL to data objects: current approaches (left) vs. MERODE (right)

3.3 Example

We illustrate the proposed approach by means of the recruitment process from [4]. The example describes a process of people applying for a job, requiring reviews of their application forms before deciding to hire the candidate or not[1]. The following paragraphs describe the MERODE domain model used to generate the EL and ISL and how it can be connected to a BPL.

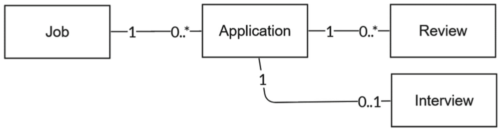


Fig. 4. UML class diagram (EDG)

The MERODE Domain Model (EDG, OET and FSMs)

In the EL, the domain model defines the business objects and their associations by means of a UML class diagram in which all associations express existence dependency, therefore also called "Existence Dependency Graph" (EDG). It is obtained by means of systematic association reification for all associations that do not express existence dependency, thus identifying important "relators" [30] as explicit business concepts. For the given case, the class diagram is shown in Fig. 4. Each class in the class diagram is also equipped with a State Chart (Finite State Machine, FSM).

MERODE defines business events as phenomena shared between the real-world and the information system, and operationalises these as call events (a subcategory of message events) that may trigger state changes in several business objects. The mapping of Business Events to Business Object types is captured through the Object-Event table (OET), where each cell indicates the type of state change that may be caused by the business event: C (creation), M (modification), or E (Ending). A marked cell thus means that the class of the corresponding column needs an operation to handle the event of the corresponding row.

The propagation rule defines a correspondence between the EDG and the OET: a master object will always be affected (at least indirectly) by the events affecting its dependents. This indirect participation is labelled 'A' (from Acquired), whereas the most dependent object affected by a business event is labelled as 'Owner' (O) of a business event. For example, 'decideToHire' is owned by Application, but will indirectly also affect the related Job as indicated in Fig. 5.

	Job	Application	Review	Interview
EVcrJob	O/C			
EVendJob	O/E			
EVcrApplication	A/M	O/C		
EVendApplication	A/M	O/E		
EVcrReview	A/M	A/M	O/C	
EVendReview	A/M	A/M	O/E	
EVcrInterview	A/M	A/M		O/C
EVendInterview	A/M	A/M		O/E
EVsetIneligible	A/M	O/E		
EVsetEligible	A/M	O/M		
EVsubmitApplication	A/M	O/M		
EVmodApplication	A/M	O/M		
EVupdateMotivation	A/M	A/M	O/M	
EVsubmitReview	A/M	A/M	O/M	
EVdecideToHire	A/M	O/M		
EVdecideNotToHire	A/M	O/M		

Fig. 5. OET

Object behaviour is defined by means of FSMs showing how the events will cause state transitions. Each object type has a default lifecycle consisting of creating an object (triggered by any of the */C business events), having an arbitrary number of modifications in a random order (triggered by its */M events). Transitions triggered by a */E business event bring the object to the final state. A more specific FSM can be defined when

[1] For the ease of reading, the description can also be downloaded here.

needed. Figure 6 show the FSMs for Review, Applications and Job. Interview has a default lifecycle. Because of the fact that a same business event may be reacted upon by several business objects, objects will synchronise and interact by means of joint participation to business events. The propagation rule allows a master to adjust its state upon activities and/or to restrict activities of its dependent object types. For example, the lifecycle of Application shows how events relating to reviews can only happen after an application has been considered eligible, and new reviews cannot be initiated once a final decision to hire or not to hire has been taken. In the lifecycle of Job, the decision to hire a candidate will cause a state change for the job ensuring that other candidates can no longer be hired.

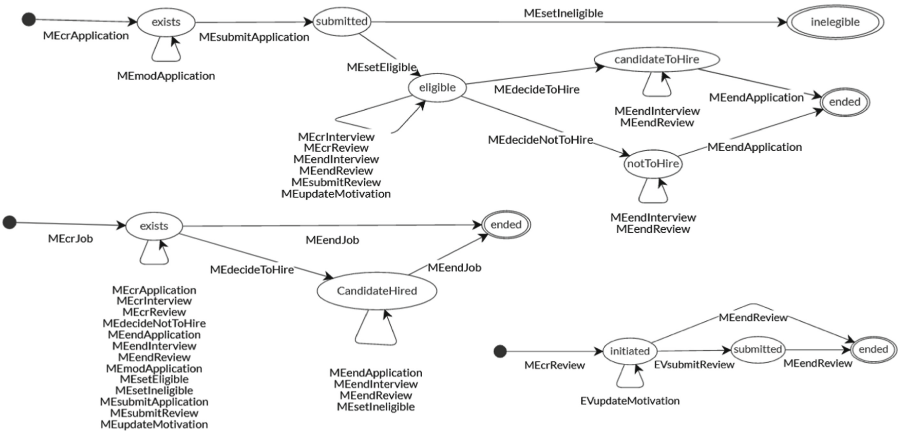


Fig. 6. Lifecycles of application, job and review

The Business Process Layer: Business Process Models
Activities in the business processes may invoke the output and input services to obtain information from the data layer and update information. While the EL captures behaviour on a per business object type basis, the BPL will capture other aspects of behaviour relating to users, task attribution and permissions. Assume the process for collecting reviews for a PhD candidate depicted in Fig. 7. The Faculty's HR consultant will start the process when an application arrives. The task "Check Eligibility" will use an output service to inspect the application file and then use an input service to trigger either the EVsetEligible or the EVsetIneligible business event for this application (see explanations in Sect. 3.4). A review by the International Office is only needed in case of international candidates. This aspect relates to work organisation, and the criteria to request a review by International Office may change over time. By managing these criteria in the BPL, maximal flexibility for adjusting the criteria for performing this task is ensured. The next task of the HR Consultant is to ask for reviews from three professors, to be looped until three professors have accepted. Each professor may accept or refuse the request. As opposed to the solution proposed in [4], we choose not to model acceptance and refusal of the tasks as part of the lifecycle of the review object type. A Review object

will only be created when the Reviewer actually writes a review. The reviews *requests* are thus distinguished from the actual reviews, the former being managed in the BPL and the latter being persisted and managed in the EL. The "Write Review" task may include updates if a professor decides to take time to think it over, and will be concluded by submitting the review. Thus, the task "Write Review", be it executed by International Office or by a professor, will trigger the business event "create review" when started, possibly trigger a number of "update motivation" events during its execution, and finally end by triggering a "submit" business event. The invocation of these events through the event-handling layer will trigger the necessary changes in the data layer while being subject to the constraints defined by the associations, multiplicities and FSMs in the EL. Connecting the BPL to the EL happens by means of a table mapping tasks to input and output services, as explained in [13], chapter 10. This technique suffices for simple one-to-one mappings. A more complex mapping would require integration with MERODE's extension for UI Design [31].

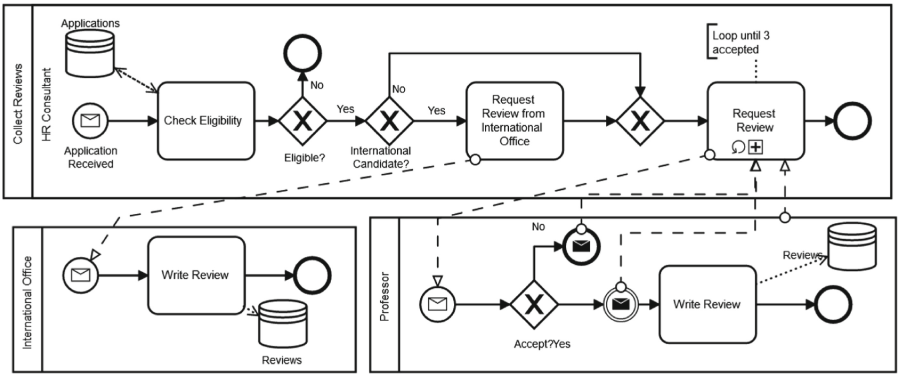


Fig. 7. Business Process for collecting reviews

3.4 Proof of Concept of Model Integration

MERODE allows generating Java applications as prototypes of the EL with default IS services in an ISL. Assume the steps of reviewing an application and taking a decistion. Figure 8 on the left shows the layered structure of such application. The generated Java Swing interface allows to "View" the details of an Application (❶) and from there to navigate to the details of its Reviews (❷). The User Interface (UI) accesses the objects making use of SQL. When a decision is taken to hire a candidate, a corresponding button will trigger the "EVDecideToHire" event (❸). The event-handler will check the permissibility of hiring the candidate against the status of that application (❹). If allowed, the state changes are performed by invoking the corresponding class's operation (❺). The result (error or success) is notified to the UI (❻).

To add a BPL layer, we used the Camunda BPM platform and the Camunda Modeler. Camunda[2] was chosen for being open source Java-based and providing a free demo

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

account. In the Camunda BPM platform, Tasklists manage users' interactions with their tasks; The Camunda Cockpit web application presents the users facilities to monitor the implemented process and its operations; Camunda Admin is used to manage the users and their access to the system. For example, groups can be created and different authorizations can be managed for distinct participants.

To connect the MERODE application to the Camunda BPM platform, the EL and the EHL are wrapped and exposed as REST web-services by using the corresponding code-generator's option [32]. The Java user interface is then replaced by Camunda Task Forms and Service Tasks. The forms take the structure of an HTML document and manipulate business objects through the RESTful web services [33]. For now, these are created manually, but they could be generated from UI models [31]. Figure 8, right shows the corresponding layered structure. The EHL ensures that sequences constraints as specified in the lifecycles are respected. The MERODE checking algorithms ensure that these lifecycles together define deadlock-free system behaviour [14].

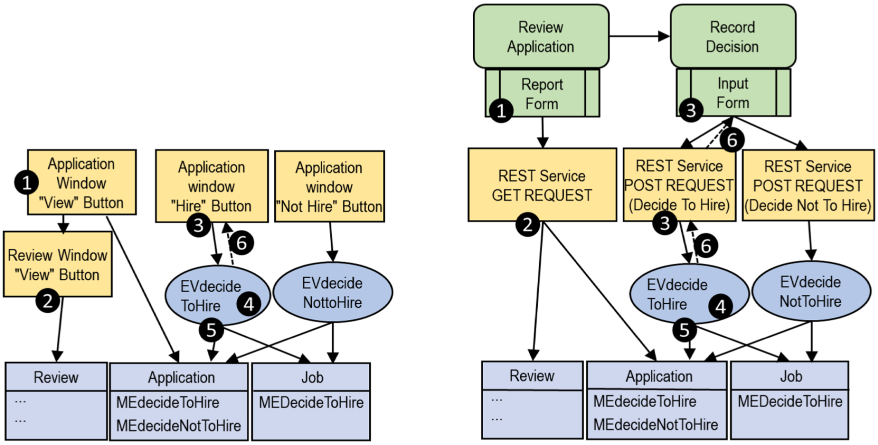


Fig. 8. Layered architecture of a generated Java prototype (left) and after integration with a BP Engine (right).

4 Evaluation

To evaluate to what extent the combination of MERODE and BPMN may support data-aware process modelling, we evaluate the prototype resulting from combining a generated MERODE-application interacting with a Camunda BPM engine through REST interfaces against the 20 requirements formulated by Künzle et al. [4]. These requirements are developed around four sets of important properties for object-aware processes. First, we elaborate on these four properties. Then, the different requirements are evaluated one by one.

4.1 General Properties

Properties Relating to Data. Data should be managed based on object types (including attributes) which are related to each other. The EDG-part of the MERODE model addresses these requirements. Furthermore, [4] identifies a hierarchy between objects, whereby an object that references another object is considered a "lower level" object and the referred to object the "higher-level" object, e.g. a job application being the higher-level object instance of a set of associated reviews. This corresponds exactly with the notions of master and dependent as specified in the MERODE method, where the Job object type would be the master of the Application object type, which in turn is the master of Review and Interview object types.

Properties Relating to Activities. The different types of activities that are identified in [4] can be addressed. Per default the triggering of a single business event, and therefore input tasks relating to a single instance, are supported, as well as viewing the details of individual objects or lists of objects and navigating to related objects. While not provided per default in the prototype, more complex queries and transactions triggering multiple events can be programmed (cfr. chapter 9, [13]).

Properties Relating to Processes. The modelling and execution of processes is based on two levels of granularity: *object behaviour* and *object interactions,* a requirement that is satisfied by the MERODE method. In addition, the ISL and BPL allow for defining coarser-grained levels of behaviour (complex transactions and processes).

Properties Relating to Users. The notion of a user is not part of a default prototype MERODE-application: per default any user has access to any operation. But the Camunda Admin can be used to manage the users and their access to the system.

Monitoring. The overall state of the process is made transparent by means of default output services allowing to view the state of individual objects. If needed, specific queries can be run on the database to provide for more specific reports. The Camunda Cockpit provides additional information.

4.2 Individual Requirements

In what follows, we go over the different categories in more depth and clarify the twenty different requirements for the evaluation of the prototype. A requirement is labelled ✓✓ when already fully satisfied by the proposed approach; ✓ when minor extensions or adjustments would be needed, ○ when complex adjustments or extensions would be required the basic ideas of which have already been described, and with a '✗' if not supported.

Data

R1 (data integration, ✓✓) describes the need for data objects that should comprise attributes and have connections to other objects [4]. This requirement is met by the MERODE EDG which presents the connected structure of the business object types.

R2 (Access to data, ○) pertains to authorisations. While the Camunda Admin allows managing this partly, data-based authorisation management would require setting an authorisation system in place. Full satisfaction of this requirement is possible, but it is not yet satisfied by the out-of-the-box approach.

R3 (cardinalities, ✓) requires the possibility to set cardinalities on relationships. The MERODE-approach allows setting a minimum constraint of 1, but for maximal constraints higher than 1, it uses the UML default of many (denoted as "*"). Setting a specific maximum number larger than one is possible but would require (straightforward) application specific coding. This requirement is thus largely satisfied.

R4 (mandatory information, ✓✓) requires the ability to distinguish between optional and mandatory attributes and to forbid proceeding further when mandatory attributes are missing. Per default, the generated code considers all attributes mandatory and will refuse the entering of incomplete data. Allowing for optional attributes is straightforward when hand coding or with minor adaptations of the code generator.

Activities

R5 (Form-based activities, ✓✓) defines form-based activities as "comprising a set of atomic actions. Each of them corresponds to either an input field for writing or a data field for reading the value of an object attribute". Making use of the REST interfaces and custom UIs, any type of form can be developed, or even generated automatically at runtime. Thus, R5 is satisfied through custom development.

R6 (black-box activities, ✓✓) activities enable complex computations or integration of advanced functionalities (e.g., sending e-mails or invoking web services). This requirement can be satisfied through custom coding and using the REST interfaces.

R7 (Variable granularity, ✗) requires the ability to distinguish between instance-specific, context-sensitive and batch activities so that users can to choose the most suitable action. The EL and ISL layers allow for providing these services, but to allow users choosing at run-time, CMMN should be used for the BPL rather than BPMN.

R8 (Mandatory and optional activities, ✓✓). Both at the level of FSMs, and at the level of the Business Processes, mandatory and optional events/activities can be defined. E.g. asking a review by International Office, may or may not be requested.

R9 (Control-flow within user forms, ✓✓) refers to adjusting the mandatory or optional character of an attribute on-the-fly while a user fills a form. Task Forms in Camunda allow for making certain attributes mandatory for the execution of an activity. The on-the-fly aspect of the requirement requires some custom-made logic.

Processes

R10 (Object behaviour, ✓✓) requires object type behaviour to be defined in terms of states and transitions. This requirement is obviously satisfied. Driving process execution based on states needs to be implemented at the business process layer, e.g. by means of rule-based events that react to conditions becoming true.

R11 (Object interactions, ✓✓) requires the possibility to process object instances concurrently while synchronising them when needed. In MERODE, creation dependencies are naturally enforced through the rules on existence dependency. A master object also has access to all information of its (direct and indirect) dependents, thus satisfying the need for aggregative information. Execution dependencies, e.g. when switching an object instance to a certain state depends on the state of another object instance, can be enforced by a master object managing execution sequences across all its dependents. Some execution dependencies may need to be managed by defining transactions that group events, or by defining a process that implements the required logic. For example, initiating a re-order when a product is out of stock would be implemented in the BPL, while the hiring of a candidate resulting in the automatic rejection of other candidates can be implemented as a transaction in the ISL.

R13 (Flexible process execution, ✗). When using BPMN to define the processes, flexibility of processes as described by Künzle et al. will not be possible. A possible solution could be using a case-based approach instead of BPMN.

R14 (Re-execution of activities, ✓✓) states that the re-execution of activities should be allowed, even if mandatory attributes are already set. The example that a person may change his/her application arbitrarily often until s/he explicitly agrees to submit it, is modelled by the self-loop ‘EVmodApplication’ in the Application FSM.

R15 (Explicit user decisions, ✗) requires allowing users to choose between execution paths. In the proposed approach, this would boil down to having gateways relying on user decisions rather than data to choose the next activity. Such user-based decisions could be captured by combining BPMN with DMN [34]. This is thus only partly satisfied by the proposed approach, unless DMN-support would be added.

User Integration

R16–R19 (○) deal with different forms of authorisations. Camunda offers a number of functionalities relating to the authorisations. A full-fledged authorisation system, combining the notions of user roles, their tasks and access to the required data is beyond the scope of the current proof-of-concept. The general design of such authorisation system has been described in [13], yet a practical implementation has not been made yet. These requirements are therefore considered as not yet satisfied.

Monitoring

R20 (Aggregated view, ✓✓) states that process monitoring should provide an aggregated view of all object instances involved in a process as well as their interdependencies. The database in the MERODE-application provides information about the objects, their dependencies and their states. The Camunda Cockpit provides information on tasks and users. Event logging is another source of information that may provide useful insights.

In summary, most requirements are at satisfied immediately or easily by the out-of-the box approach, though custom coding may be needed in addition to the default code generation. The addition of an authorisation layer (R2, R16–19) and support for different forms of flexibility (R7, R13, R15) need elaborating the approach further.

5 Discussion

Ideally, process modelling should be “data-aware” in the sense that an existing domain model is presumed to exist or to be developed jointly. Possibly process modelling may require revisiting the domain model. Similarly, domain modelling should be conscious of the business processes that need to be supported. As constraints set by a domain model will impact processes, the conceptual domain modeller should be aware of which processes are hindered or made possible in order to make the right decisions during his/her data modelling.

The main limitation of this research is that it limited itself to an out-of-the-box implementation using the default application generated by the MERODE-code generator and linking it to simple Camunda service and form tasks by means of the default REST web-services generated by the code generator. Nevertheless, this basic proof-of-concept combining MERODE with BPMN is able to satisfy a majority of the 20 requirements defined in [4]. This comes as no surprise given that the MERODE approach contains the main ingredients defined in the BALSA framework [7]. Augmenting the proposed architecture with DMN, and providing integration of MERODE with a case-based approach next to BPMN, could help to achieve the for now unsatisfied requirements on process flexibility. Investigating Camunda Admin's possibilities more deeply and implementing a data-based authorisation system requires further investigation and would be key to satisfy the authorisation-related requirements.

The review of data-centric approaches in [5] reveals that their usability is a source of concern. On the other hand, research also shows that UML-class diagrams and UML state charts are amongst the most-used modelling languages [27]. Combining these "modelling favourites" with BPMN could meet the usability concerns and stimulate the uptake of data-centric process management. Teaching of the MERODE + BPMN approach to students and to Enterprise Architects has already proven its ease of use. Enterprise Architects in particular value the innate data-centric process aspects embedded in the MERODE approach.

The whole process of generating and starting the web services, setting up the connection with Camunda, etc. requires several steps [33], but could ideally be done with less hazzle. The utopian goal would be to achieve this through code generation as well, to allow for process validation through the integrated prototyping of a collection of processes and the supporting information system with just a few clicks.

Finally, process verification has not been addressed in this paper. The process algebra formalisation of MERODE provides extensive consistency checking [35], but checking the consistency of the combined state charts against a business process model needs further investigation. An initial study has been published in [36], but requires further extension to achieve support for process verification as a complement to the above-mentioned validation through integrated prototyping.

6 Conclusion

While being preliminary, the proof-of-concept of MERODE and Camunda presented in this paper provides interesting opportunities to elaborate its functionalities. Considering other process implementation platforms and augmenting the proposed approach with DMN can provide additional pathways for future research. On the other hand, addressing the authorisation issues could prove a challenge. Besides addressing the unfulfilled requirements, development of a proof of concept with more complex models including completing the generated code by hand and using more elaborate BPMN models would allow to gain deeper insights into the merits of this combination. A formal evaluation of the approach could shed light on remaining issues, and how to make data-centric process management easier to use.

References

1. Hasić, F., De Smedt, J., Vanthienen, J.: Augmenting processes with decision intelligence: principles for integrated modelling. Decis. Support Syst. **107**, 1–12 (2018)
2. ISO/IEC/IEEE 42010:2011 systems and software engineering: architecture description. https://www.iso.org/standard/50508.html. Accessed 11 Jan 2021
3. Steinau, S., Marrella, A., Andrews, K., Leotta, F., Mecella, M., Reichert, M.: DALEC: a framework for the systematic evaluation of data-centric approaches to process management software. Softw. Syst. Model. **18**(4), 2679–2716 (2018). https://doi.org/10.1007/s10270-018-0695-0
4. Künzle, V., Weber, B., Reichert, M.: Object-aware business processes: fundamental requirements and their support in existing approaches. In: Krogstie, J. (ed.) Frameworks for Developing Efficient Information Systems: Models, Theory, and Practice, pp. 1–29. IGI Global, Hershey (2013)
5. Reijers, H.A., et al.: Evaluating data-centric process approaches: does the human factor factor in? Softw. Syst. Model. **16**(3), 649–662 (2016). https://doi.org/10.1007/s10270-015-0491-z
6. Haarmann, S., Holfter, A., Pufahl, L., Weske, M.: Formal framework for checking compliance of data-driven case management. J. Data Semant. (2021). https://doi.org/10.1007/s13740-021-00120-3
7. Hull, R.: Artifact-centric business process models: brief survey of research results and challenges. In: Meersman, R., Tari, Z. (eds.) OTM 2008. LNCS, vol. 5332, pp. 1152–1163. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88873-4_17
8. Calvanese, D., Montali, M., Estañol, M., Teniente, E.: Verifiable UML artifact-centric business process models. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pp. 1289–1298 (2014)
9. Künzle, V., Reichert, M.: PHILharmonicFlows: Towards a Framework for Objectaware Process Management. J. Softw. Maint. Evol. Res. Pract. **23**(4), 205–244 (2011)
10. Calvanese, D., Montali, M., Patrizi, F., Rivkin, A.: Modeling and in-database management of relational, data-aware processes. In: Giorgini, P., Weber, B. (eds.) CAiSE 2019. LNCS, vol. 11483, pp. 328–345. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21290-2_21
11. Artale, A., Calvanese, D., Montali, M., van der Aalst, W.M.P.: Enriching data models with behavioral constraints. In: Ontology Makes Sense, Volume of 316, no. Dynamics, vol. 365, pp. 257–277 (2019)
12. Estañol, M., Sancho, M.R., Teniente, E.: Ensuring the semantic correctness of a BAUML artifact-centric BPM. Inf. Softw. Technol. **93**, 147–162 (2018)
13. M. Snoeck, *Enterprise Information Systems Engineering*, vol. 141. 2014.
14. Snoeck, M., Dedene, G.: Existence dependency: The key to semantic integrity between structural and behavioral aspects of object types. IEEE Trans. Softw. Eng. **24**(4), 233–251 (1998)
15. Neumann, C.P., Lenz, R.: $\alpha-$ flow: a document-based approach to inter-institutional process support in healthcare. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) Business Process Management Workshops, pp. 569–580. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12186-9_55
16. van der Aalst, W.M.P., Stahl, C., Westergaard, M.: Strategies for modeling complex processes using colored petri nets. In: Jensen, K., van der Aalst, W.M.P., Balbo, G., Koutny, M., Wolf, K. (eds.) Transactions on Petri Nets and Other Models of Concurrency VII, pp. 6–55. Springer Berlin Heidelberg, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38143-0_2

17. Kiss, P.J., Klimkó, G.: A reverse data-centric process design methodology for public administration processes. In: Kő, A., Francesconi, E., Anderst-Kotsis, G., Tjoa, A.M., Khalil, I. (eds.) EGOVIS 2019. LNCS, vol. 11709, pp. 85–99. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-27523-5_7
18. Ouali, N.H., Tmar, M., Haddar, N., Tmar, M.: Models and run-time systems for data intensive workflow applications. In: 2017 18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), pp. 429–436 (2017)
19. Rietzke, E., Bergmann, R., Kuhn, N.: ODD-BP - an ontology- and data-driven business process model. In: CEUR Workshop Proceedings, vol. 2454, pp. 409–415 (2019)
20. Mertens, S., Gailly, F., Poels, G.: A Generic framework for flexible and data-aware business process engines. In: Proper, H.A., Stirna, J. (eds.) Advanced Information Systems Engineering Workshops: CAiSE 2019 International Workshops, Rome, Italy, June 3-7, 2019, Proceedings, pp. 201–213. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-20948-3_18
21. De Giacomo, G., Oriol, X., Estañol, M., Teniente, E.: Linking data and BPMN processes to achieve executable models. In: Dubois, E., Pohl, K. (eds.) Advanced Information Systems Engineering, pp. 612–628. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_38
22. Hasić, F., De Smedt, J., Vanden Broucke, S., Asensio, E.S.: Decision as a service (DaaS): a service-oriented architecture approach for decisions in processes. IEEE Trans. Serv. Comput. 1 (2020). https://doi.org/10.1109/TSC.2020.2965516
23. Pastor, O., Molina, J.C.: Model-Driven Architecture in Practice: A Software Production Environment Based on Conceptual Modeling. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71868-0
24. Estañol, M., Munoz-Gama, J., Carmona, J., Teniente, E.: Conformance checking in UML artifact-centric business process models. Softw. Syst. Model. **18**(4), 2531–2555 (2018). https://doi.org/10.1007/s10270-018-0681-6
25. Deutsch, A., Hull, R.: Automatic verification of database-centric systems **43**(3), 1–13 (2014)
26. Calvanese, D., Ghilardi, S., Gianola, A., Montali, M., Rivkin, A.: SMT-based verification of data-aware processes: a model-theoretic approach. Math. Struct. Comput. Sci. **30**(3), 271–313 (2020)
27. van der Linden, D., Hadar, I., Zamansky, A.: What practitioners really want: requirements for visual notations in conceptual modeling. Softw. Syst. Model. **18**(3), 1813–1831 (2018). https://doi.org/10.1007/s10270-018-0667-4
28. Richards, M.: Software Architecture Patterns. https://www.oreilly.com/content/software-architecture-patterns/. Accessed 13 Mar 2021
29. Snoeck, M., Lemahieu, W., Goethals, F., Dedene, G., Vandenbulcke, J.: Events as atomic contracts for component integration. Data Knowl. Eng. **51**(1), 81–107 (2004)
30. Guizzardi, G.: Ontological foundations for structural conceptual models. University of Twente (2005)
31. Ruiz, J., Sedrakyan, G., Snoeck, M.: Generating user interface from conceptual, presentation and user models with JMermaid in a learning approach. In: Proceedings of the XVI International Conference on Human Computer Interaction, pp. 25–32 (2015)
32. Scheynen, N.: Construction of web services using the MERODE approach. Faculteit Economie en Bedrijfswetenschappen, KU Leuven, Leuven (2016)
33. Mohout, I., Leyse, T.: Enriching business process simulation by integration with MERODE prototype applications. KU Leuven (2020)
34. Hasić, F., Serral, E., Snoeck, M.: Comparing BPMN to BPMN + DMN for IoT process modelling: a case-based inquiry. In: Proceedings of the 35th Annual ACM Symposium on Applied Computing, pp. 53–60 (2020)

35. Snoeck, M., Michiels, C., Dedene, G.: Consistency by construction: the case of MERODE. In: Jeusfeld, M.A., Pastor, Ó. (eds.) ER 2003. LNCS, vol. 2814, pp. 105–117. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39597-3_11
36. De Backer, M., Snoeck, M., Monsieur, G., Lemahieu, W., Dedene, G.: A scenario-based verification technique to assess the compatibility of collaborative business processes. Data Knowl. Eng. **68**(6), 531–551 (2009)

Evaluating Perceived Usefulness and Ease of Use of CMMN and DCR

Amin Jalali(✉)

Department of Computer and Systems Sciences, Stockholm University, Stockholm, Sweden
aj@dsv.su.se

Abstract. Case Management has been gradually evolving to support Knowledge-intensive business process management, which resulted in developing different modeling languages, e.g., Declare, Dynamic Condition Response (DCR), and Case Management Model and Notation (CMMN). A language will die if users do not accept and use it in practice - similar to extinct human languages. Thus, it is important to evaluate how users perceive languages to determine if there is a need for improvement. Although some studies have investigated how the process designers perceived Declare and DCR, there is a lack of research on how they perceive CMMN. Therefore, this study investigates how the process designers perceive the usefulness and ease of use of CMMN and DCR based on the Technology Acceptance Model. DCR is included to enable comparing the study result with previous ones. The study is performed by educating master level students with these languages over eight weeks by giving feedback on their assignments to reduce perceptions biases. The students' perceptions are collected through questionnaires before and after sending feedback on their final practice in the exam. Thus, the result shows the perception of participants can change slightly by receiving feedback, while the change is not significant due to being well trained. The reliability of responses is tested using Cronbach's alpha, and the result indicates that both languages have an acceptable level for both perceived usefulness and ease of use.

Keywords: Business process modeling · Knowledge-intensive · Case management · CMMN · DCR

1 Introduction

Case Management is a research paradigm that supports knowledge-intensive process (KiP) management [9]. The support is defined around the concept of the case, e.g., patient in the healthcare or customer in the insurance domain. In these processes, knowledge workers decide how a case shall proceed instead of predefined rigid rules. As a result, the support for KiPs differs from the workflow-based processes so that even the management lifecycle needs its variation, known as Collaborative knowledge work lifecycle [9].

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 147–162, 2021.
https://doi.org/10.1007/978-3-030-79186-5_10

The collaborative nature of work in KiPs requires more freedom by knowledge workers to decide how a case shall proceed. Therefore, traditional workflow-based models could not support case management, for they become too complex by capturing the high degree of flexibilities required by different cases. Thus, several case management modeling languagues are defined, e.g., Declare [27,28], Dynamic Condition Response (DCR) [16], and Case Management Model and Notation (CMMN) [25] which is based on Guard-Stage-Milestone (GSM) [17]. As these languages are new, it is hard to predict if they will be accepted by users, which opens up rooms for further investigation.

The user acceptance of process modeling languages, like other information systems, can be evaluated based on two variables, i.e., perceived usefulness (PU) and perceived ease of use (PEU) [8]. Currently, there are few studies on how users perceive Declare and DCR, e.g. [13,30,35]. However, there is a gap in how users perceive CMMN. Therefore, this study aims to evaluate the user acceptance of CMMN language. It also includes DCR to have a comparison reference to relate the result to other investigations. DCR is selected because it is expected to be more comprehensive than Declare, as it has fewer modeling elements [30]. The evaluation is performed based on the Technology Acceptance Model [8].

As these languages are new, it is hard, if not impossible, to find users who already know them. Thus, the users shall be trained, and the acceptance shall be measured based on their perception (self-assessment). As the self-assessment is a subjective score, it can differ during the learning process due to biased factors, i.e., over- or under-confidence [10]. These biased factors can be minimized by repeated experiences and feedback [10]. In this study, we aim to answer this research question:

– *How do trained process designers perceive the usefulness and the ease of use of DCR and CMMN languages for modeling knowledge-intensive processes?*

To answer this question, we trained students in the business process and case management course at Stockholm University. The students practiced DCR and CMMN for around eight weeks and received feedback on their assignments - to minimize the overconfidence and underconfidence biases. Finally, we collected perceptions from those who were interested in participating in this study before and after the feedback on their performance on the exam. Participation in this study was optional. The reliability of responses is tested using Cronbach's alpha, and the result shows that both languages have an acceptable level for both perceived usefulness and ease of use.

The remainder of this paper is structured as follows. Section 2 gives a brief background on related work. Section 3 describes the method that is used in this study. Section 4 reports the result, and Sect. 5 concludes the paper.

2 Background

This section summarizes related works and excerpts of DCR and CMMN notations, which are used later to present sample processes. Please note that we do not aim to give the full syntax of these languages, which can be found in related literature.

2.1 Users Perceptions in Business Process Management

A language will eventually die if people do not accept and use it in practice, which is also true for business process modeling languages. Thus, it is important to evaluate how users perceive languages to determine if there is a need for improvement. The evaluation can help us to improve process modeling languages. Here, we mention some related work that evaluated the user acceptance in the Business Process Management (BPM) area, in general, and in the case management area, in particular.

Users Perceptions in Business Process Modeling
Process models can easily become complex as they represent complex business processes in practice, so different approaches have been developed to enable process designers to deal with the complexity. La Rosa et al., categorize these approaches into two main category, i.e., concrete syntax modifications [22] and abstract syntax modifications [23]. They also identified different patterns that can be applied in each category.

The concrete syntax modifications refer to i) using highlights, ii) following layout guidelines, iii) following naming guidelines, or iv) applying different representations techniques, e.g., using icons for tasks or etc. [22]. The abstract syntax modifications refer to applying different abstraction techniques in business process modeling, e.g., using vertical, horizontal, or orthogonal modularization technqiues [23]. La Rosa et al. evaluated how users perceived usefulness and ease of use of the identified patterns by applying the technology acceptance model [8]. Their evaluation study showed that all identified patterns are perceived as relevant.

The users' perception evaluation is important because the artifact's actual usage is influenced by the potential users' perceptions - which can be measured in terms of usefulness and ease of use [8]. Therefore, researchers have used techniques like the technology acceptance model to evaluate different business process modeling techniques. For example, the technology acceptance model is used to evaluate i) how users perceived orthogonal modularization based on aspect-oriented business process modeling in [19,20], ii) how users perceived the vertical decomposition using BPMN in [32], and iii) how graphical highlights can increase the cognitive effectiveness of business process models in [21].

Users Perceptions in Case Management
Case management is fairy new paradigm in comparison with Business Process Management, and few languages have been developed to support managing cases. Declarative Service Flow Language (DecSerFlow) [33] (a.k.a., Declare) can be considered as one of the first attempts for such languages.

The understandability and maintainability of process models is a crucial requirement for any process modeling language. Thus, Fahland et al. identified and hypothesized a set of propositions for differences between imperative and declarative process modeling languages concerning understandability and maintainability in [11] and [12], respectively. Weber et al. [34] conducted a controlled experiment to investigate if process designers can deal with increased levels of constraints when using Declare. The participants were 41 students from two universities. The results show that the participants can deal with introduced constraints, which justified further development of declarative process modeling.

Pichler et al. investigated if the imperative or the declarative process modeling languages are better understood by running an experiment with 27 students [29]. Their study shows that the imperative process modeling languages are better understood by students. The technology acceptance model is used for the first time to assess how professionals perceived Declare, and DCR languages by Reijers et al. [30]. Ten professionals from the industry participated in their workshop, and the result shows that they found the languages easy to learn. This study also revealed the potential for defining hybrid approaches. Several hybrid process modeling is proposed, among which the understandability of DCR-HR is investigated through an explorative study [2].

Zugal et al. [35] investigated the effect of using hierarchies in expressiveness and understandability of declarative models. The study is based on nine participants in two universities. It shows that hierarchies enhance expressiveness. It also shows that the hierarchies can increase the models' understandability, but they should be applied with care. Haisjackl et al. [13] investigated how declarative models are understood through an explorative study. Their study shows that the subjects could understand a single constraint well, but it was challenging for them to handle a combination of constraints. This study also shows that some graphical notation in Declare, which are similar to imperative modeling languages, causes considerable trouble in understandability.

2.2 Dynamic Condition Response (DCR)

Hildebrandt and Mukkamala introduced DCR in 2010 as a declarative process modeling language [14]. The syntax of the language includes the definition of nodes (the group of an activity and its roles) and the relation that can be defined among them, i.e., response ($\bullet\!\!\rightarrow$), condition ($\rightarrow\!\!\bullet$), inclusion ($\rightarrow\!+$), and exclusion ($\rightarrow\!\%$) [14].

They also defined semantics for DCR, where several events can occur for a node. A node can be included or excluded from the process structure. A node can also be in the pending state, meaning that the process cannot successfully be finished until an event of the node occurs. In short, the response relation among nodes a and b ($a \bullet\!\!\rightarrow b$) means that the state of node b will be pending if an event of node a happens. More precisely, event b must eventually happen if event a happens. The condition relation among nodes a and b ($a \rightarrow\!\!\bullet\, b$) means that an event of b cannot be occured unless a occurs.

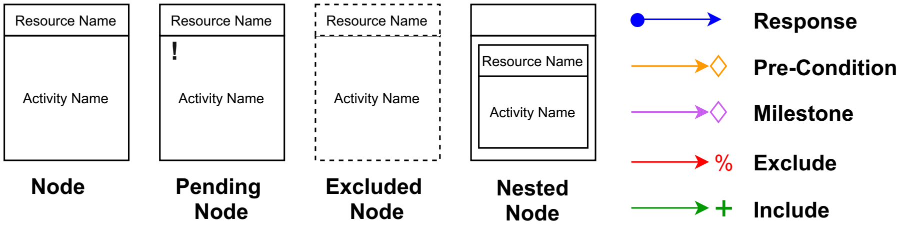


Fig. 1. An excerpt of DCR syntax

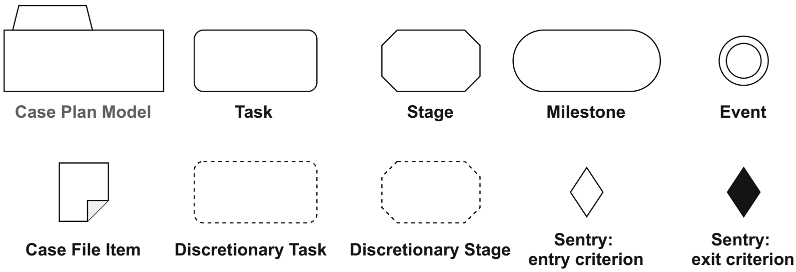


Fig. 2. CMMN legends - basic elements

The DCR's syntax and semantics have evolved over the years. For example, the syntax is enriched to represent the excluded nodes by dashed border line [16]. The extended syntax also represents a node's pending state by decorating it with an exclamation mark (!). In addition, milestone relation ($\rightarrow\!\diamond$) and nested nodes are also introduced to the language [15]. The milestone relation among two nodes a and b ($a \rightarrow\!\diamond\, b$) means that b can occur as long as a is not in the pending state.

DCR graph is the toolset that enables modeling and simulation of DCR models. Currently, it supports other relations, including Pre-Condition. The pre-condition relation among two nodes a and b means that $a \rightarrow\!\diamond\, b \wedge a \rightarrow\!\bullet\, b$. This relation is represented with the same graphical notation as the milestone but with a different color. Figure 1 shows an excerpt of DCR syntax.

2.3 Case Management Model and Notation (CMMN)

Case Management Model and Notation (CMMN) is a case modeling language which is defined by Object Management Group (OMG) [25]. This language is developed by extending the Guard-Stage-Milestone (GSM) language [17,18]. Figure 2 shows an excerpt of CMMN syntax.

The *case plan model*, represented by a folder, is the core part of a CMMN model. The *case plan model* captures the complete behavior of a case, and all other elements will be children of the *case plan model*. The *case file item* represents the data, *Task* represents activities that can happen, *Stage* represents a container that includes other elements (like subprocess in BPMN), *Milestone*

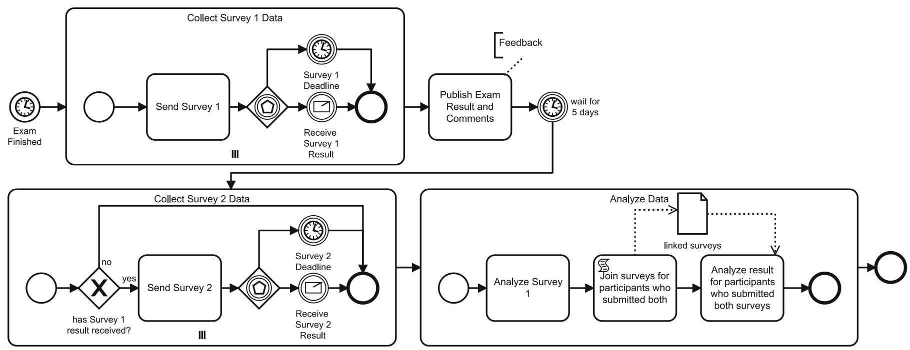


Fig. 3. Research steps

represents an achievable target, and *event* represents something that happens during the course of a case. Some elements like tasks can have sentries on their border. A *sentry* can represent *entry criterion* or *exit criterion*, which defines the condition or event - based on which the element can be enabled or terminated, respectively.

Tasks and Stages can also be represented by the dashed borderline, which is known as *discretionary task* or *discretionary stage*. Discretionary elements are not available to knowledge workers at runtime. However, they can add these elements to their case plan at runtime. These elements can be related to each other through lines, and the connection rules are defined in CMMN specification [25]. Elements can also be decorated using different decoration icons. For example, ! or # indicates that the task, stage, or milestone are mandatory or repetitive, respectively.

3 Research Method

Davis F.D. describes how the actual system's usage depends on the attitude of users, which can be predicted using two variables: perceived usefulness (PU) and perceived ease of use (PEU) of a system [7]. He also defined measurement scales that can be used to evaluate these variables [8]. This evaluation technique is widely used to evaluate different information systems including different business process modeling languagues, e.g. [19,20,22,23,30]. This study adopted the technology acceptance model to evaluate how users perceive DCR and CMMN in concern with these variables.

The user acceptance evaluation is a sort of subjective score because users need to respond based on self-assessment. Thus, the result can change during repeated experiences due to self-assessment biases. These biases are rooted in over- or under-confidence, which can be minimized by repeated experience and feedback [10]. Indeed, the answers will be more reliable when these biases are minimized.

To minimize the over- and under-confidence, we trained the students in the business process and case management course at Stockholm University for around eight weeks. In this period, students practiced DCR and CMMN languages in groups by designing process models, and they received feedback. They also had two sessions with an external expert to ask their questions for DCR language. Then, they participated in the exam where they needed to design a DCR and a CMMN model for a given case description individually. The data collection and data processing for this study starts after the exam, which is shown in Fig. 3. Participation in this study was voluntary.

In this study, we collect data from students through two surveys conducted before and after giving feedback on their exams. We call them Survey 1 and Survey 2, respectively. The surveys were identical, but students did not know about it beforehand. The questions are defined based on [7], where participants could response by choosing options in the range of *extremely unlikely* (1), *quite unlikely* (2), *slightly unlikely* (3), *neither likely nor unlikely* (4), *slightly likely* (5), *quite likely* (6), and *extremely likely* (7).

After the exam, survey 1 is sent out, and the responses are collected upon the announced deadline. Then, we published the grades and comments for the exam result. We gave students five days to go through the comments and discuss their questions with the teacher. Then, we sent out the second survey to those who participated in the first survey. We collected data that were submitted before the announced deadline. Finally, we started data analysis.

We analyzed the collected data from Survey 1, and we name it *Study 1*. The second survey responses are linked with the first one to analyze the result before and after the feedback. We analyzed the collected data from the linked data source containing both Survey 1 and Survey 2, and we name it *Study 2*. This study includes the result of students who participated in both Survey 1 and Survey 2, enabling us to track how opinions are changed after receiving the feedback. We tested if the changes in responses are significantly different through three nonparametric test of the null hypothesis techniques. To test the reliability of responses, we used Cronbach's alpha, which is widely used in related work, e.g. [5,6,24,26]. The Cronbach's alpha value above 0.7 is usually considered as reliable.

The details of the study will be given in the result section. Here, we also give the case description for which students designed a DCR and a CMMN model.

3.1 Case Description

Managing courses is a sort of knowledge-intensive process which relies on the skills and knowledge of academic staff as knowledge workers. This text describes the course management process at the department of computer and systems sciences (DSV) at Stockholm University.

The process starts when the course director registers each course coordinator when the course planning period begins. After the course coordinator is registered, (s)he can set up the courses. (S)he must publish the course contents and define the course schedule. It is important to check potential conflicts among mandatory sessions among different courses. Thus, the administrative personnel needs to check the conflicts after the course coordinator schedule the course. If they approve, the course coordinator needs to release the course, so students can see the schedule. Indeed, the course coordinator shall release the course after (s)he apply any changes so that students can see the changes.

To avoid having unreleased courses, the administrative personnel will notify course coordinators sometimes before the academic term starts. Note that the course coordinator can change the course content and schedule during the course several times, but the same process shall be followed. After the course is released, the course coordinator can run the course, which includes Registering groups, Publishing Recorded Lectures, opening submission box, and Registering Assignments Grades. These activities can be done several times during the course.

The course coordinator can also start preparing the exam after the course is released. It includes submitting exam questions and reporting the exam grades. The course coordinator needs to Upload Answer sheets after reporting the exam grades. The grades can vary between A to F. There is a special grade known as Fx. This grade means that the student has not passed the exam, but the submission was very close to the passing grade. In such a case, the course coordinator can Define and publish complementary Assignments for those students. The course coordinator needs to Correct Submissions and Report Grades in such a case. Note that this change will only be given to students once.

After the course and its examination is over, the course coordinator shall evaluate the course. The evaluation starts by defining and publishing the evaluation form. After the evaluation is done, the course coordinator shall write and Submit the Course Evaluation.

4 Result

Among 24 master level students who registered for the exam, 20 students participated in *Study 1* among which 13 students also participated in *Study 2*. In *Study 1*, 9 and 11 students were male and female, respectively; while in *Study 2*, 5 and 8 students were male and female, respectively. The average age for students in both studies was around 34, who participated in the "Master's Programme in Open eGovernment" - which is a distance program. It is usual to have students with an industrial background in this course.

Students were familiar with BPMN process modeling languages, but they did not declare any prior experience on a declarative process modeling language. Each student submitted one DCR and one CMMN model, so we collected different versions of their designs. Here, we elaborate on how the process can be designed using these languages through two sample models. We will elaborate on the study result afterward.

4.1 Sample DCR Model

The case is modeled differently by each student, as there is the possibility to decide on the level of flexibility that a KiP model shall support. Figure 4 shows a sample DCR model for the given case description.

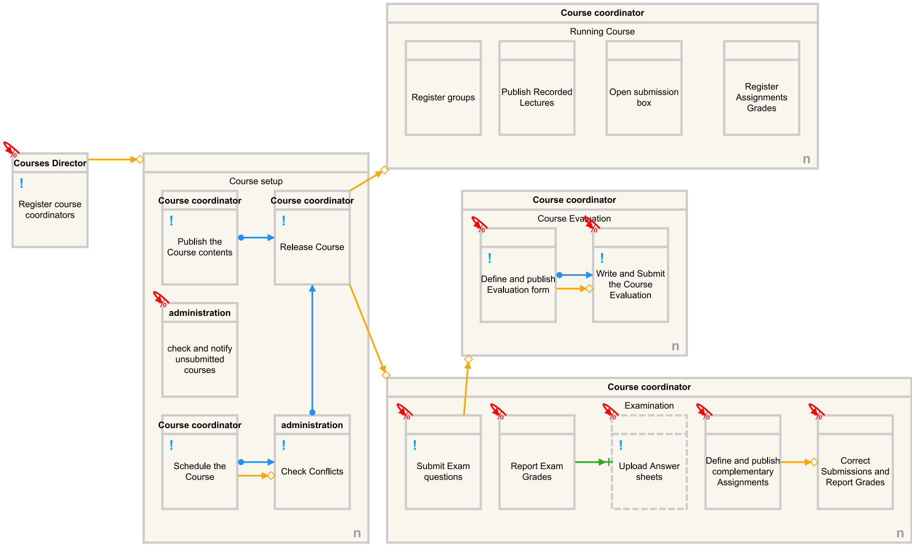


Fig. 4. Sample DCR model for the given case description

In this model, the start of the process is modeled to be rigid - meaning that the process has only one start point, i.e., *Register course coordinators*. The course director can perform this activity, and it is annotated as mandatory to apply more constraints for knowledge-workers to finish the process. This activity excludes itself after execution, meaning that it can only be performed once! This constraint is not specified in the case description, and it can be considered as an extra control that the designer applied. This constraint may cause a problem as the course director may want to assign another course coordinator later, which is not given in the case description. As there might be many similar situations in the real world, it might be better if the process designer avoids applying extra control on models when designing KiPs.

The execution of *register course coordinators* activity enables *course setup*, which includes five activities. The process model captures the case description. Here, the self-*Exclude* relation applies extra control, which can be avoided. After *Release course* activity, *Running course* and *Examination* are enabled. The *Running course* is quite flexible to give freedom to the course coordinator to choose how to do the activities. The *Examination* includes five activities - all decorated with a self-*Exclude* relation to limit the number of their execution. All exclusions are fine as they are aligned with the case description, except *Submit Exam questions*. Again, the process is designed too rigid as it is expected for this node.

After *Submit Exam questions* is performed, the *Course Evaluation* will be enabled. The self-*Exclude* relations are not needed again here, as they make the process rigid.

4.2 Sample CMMN Model

Figure 5 represents a sample CMMN model for the given case description. In this model, every top-level task, stage, and milestone has a sentry as an entry criterion, which means that they are not enabled when the model is created - as their condition is not fulfilled. Thus, the process starts when the *Start of course planning* timer event occurs AND the course director starts *Register course coordinators* task. Note that if two events are related to an entry criterion sentry, both of them shall be fulfilled, which is equivalent to AND join in workflow-based models. To show the OR relation, events shall be related to different entry criterion sentries, e.g., look at *Release Course* activity. To specify the resource that performs a task, one can relate the resource directly to the entry criterion. However, this can make the model very complex. Thus, *DCR representation for resources seems better in terms of model simplicity.*

There is also one difference in how DCR and CMMN enable flexibility in terms of repeatable tasks/activities. In DCR, an activity can be repeated unless explicitly limited. In contrast, *a task is not repeatable in CMMN by default.* It

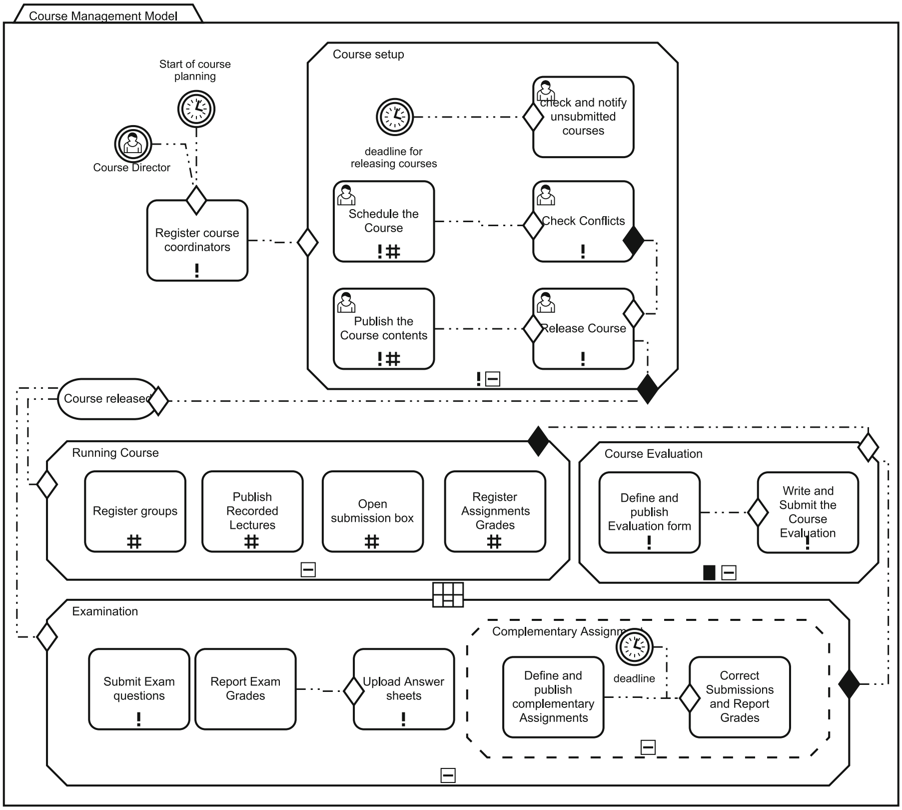


Fig. 5. A sample CMMN model for the given case description

needs to be decorated with *Repetition Rule* (#). So knowledge-worker can repeat the task.

When the course director performs the *Register course coordinators*, the *Course setup* stage would be enabled. After the stage is completed, the *Course released* milestone would be achieved. This milestone enables *Running Course* and *Examination* stages. Finally, the *Course Evaluation* can be done to finish the process.

It is possible to analyze all students' submissions to explore how the case is modeled differently in these languages. We skip this analysis as it is outside the scope of this paper.

4.3 Perception Analysis

Figure 6 shows how students perceived the utilized KiPs (i.e., DCR and CMMN) to be useful and easy to use through Box plots. The figure is not specific for each of these languages. The left- and right-side of Fig. 6 shows the result of *Study 1* and *Study 2* representing the perception of students before and after recieving the feedback, respectively. It worth reminding that some students from *Study 1* did not participated in *Study 2*, so the perception of students for *Study 1* who participated in *Study 2* is also demonstrated in the *Study 2* through dashed box plot.

In *Study 1*, represented by the left- sub-figure in Fig. 6), the median for both PU and PEU is around 5 (out of 7). In *Study 2*, represented by the right- sub-figure in Fig. 6), the median for PU before and after the feedback is around 5. The difference is visible in the first and second quartiles, where some students lowered their scores for PU after receiving feedback. The difference for PEU is even more, where the median is lowered by one after receiving the feedback. We will check if the difference is significant later.

Figure 7 shows the means and 95% confidence interval for all measures that are demonstrated in Fig. 6. The distribution of our data is not normal, so we cannot perform statistical tests like t-test to identify if the feedback had a significant difference in responses. Therefore, we used three nonparametric statistical significance tests, i.e., Mann-Whitney U, Wilcoxon Signed-Rank, and Mood's median tests. We measured p-values based on these tests for PU and PEU in Study 2 for the responses that we received before and after the feedback. Figure 8 shows the distribution of responses in these two cases in addition to the p-values. The null hypothesis (H_0) is that the distribution of responses before and after the feedback are the same for both PU and PEU. The p-values are greater than 0.05, so we cannot reject the null hypothesis. Thus, the feedback did not change the perceptions significantly.

Table 1 shows the Cronbach's alpha result that we calculated per variable per language, where all values are above 0.7, which is generally considered as the accepted threshold. The Cronbach's alpha for PU for both languages and PEU of DCR is quite high, i.e., above 0.9. However, Cronbach's alpha for PEU of CMMN is not as high as others.

Table 2 shows the Cronbach's alpha result that we calculated per variable per language before and after feedback, where all values are above 0.7. The Cronbach's alpha for all measures is quite high, i.e., above 0.9, except the Cronbach's alpha for PEU of CMMN before the feedback, which is 0.78. It is worth mentioning that Cronbach's alpha for PU and PEU of both languages before the feedback is similar to their Cronbach's alpha for the whole population reported in Table 1.

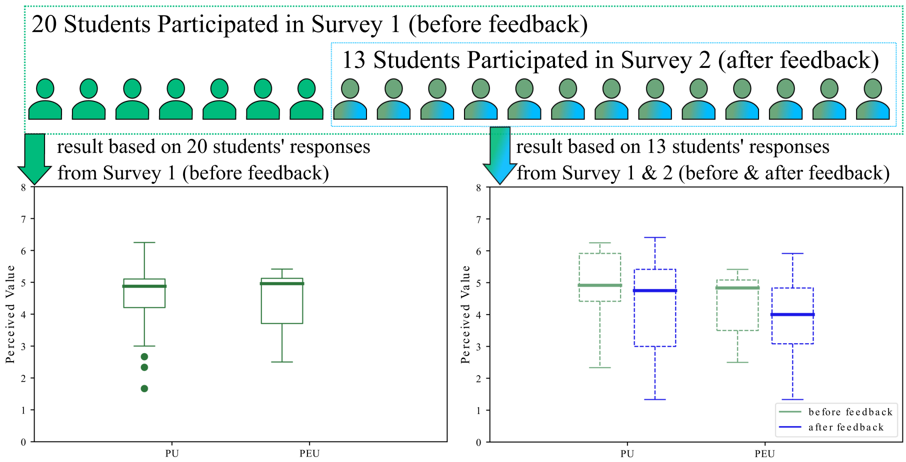


Fig. 6. Aggregated Perceived Usefulness (PU) and Ease of Use (PEU)

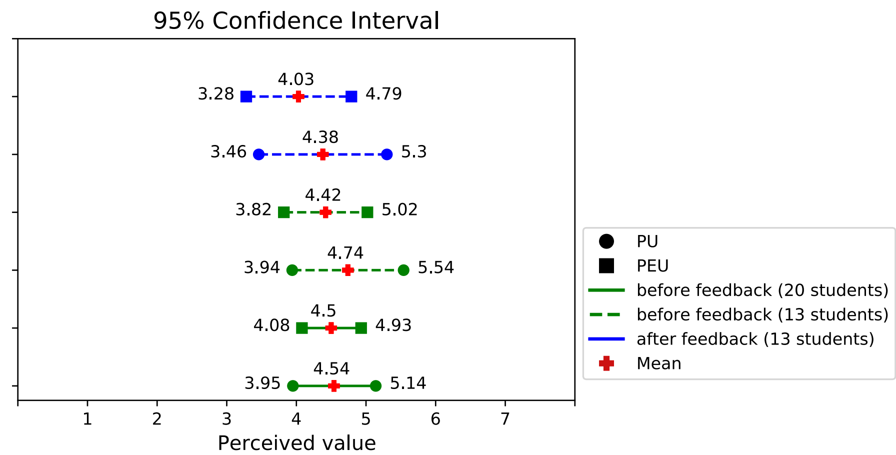


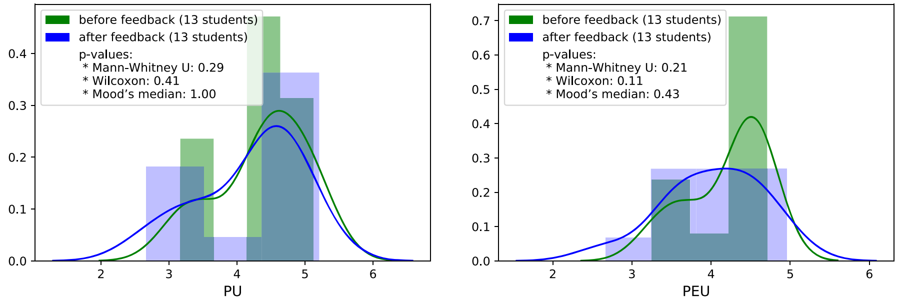
Fig. 7. 95% Confidence Interval for the means

Table 1. Cronbach alpha for study 1

	Perceived Usefulness (PU)	Perceived Ease of Use (PEU)
DCR	0.97	0.94
CMMN	0.97	**0.70**

Table 2. Cronbach alpha for study 2

	Perceived Usefulness (PU)		Perceived Ease of Use (PEU)	
	Before feedback	After feedback	Before feedback	After feedback
DCR	0.98	0.98	0.96	0.97
CMMN	0.96	0.96	**0.78**	**0.93**



(a) p-values for PU (b) p-values for PEU

Fig. 8. P-Values of PU and PEU for before and after feedback

4.4 Discussion on Biases and Threats to Validity

First, we have used students as our test subjects instead of real process designers as explained and motivated in this paper. Students are considered valid subjects in this area as these languages are new and are mostly unknown for practitioners outside. Thus, students can be used to evaluate how these languages can be perceived by process designers, which is also used in related work such as [1–4,13,29,31,34]. The use of students as subjects can weaken the causal relation for predicting if the artifact will be used in the future. The fact that students belong to the same class and trained under the same process can also be considered as a learning bias.

Second, it shall be mentioned that students were familiar with BPMN business process modeling language, which may potentially impact their PU and PEU of declarative languages. From the author's perspective, this impact is unknown, and it will be interesting to evaluate if prior knowledge on workflow-based modeling language can have a positive or negative impact!

Third, feedback can impact the subjects' opinions as they can be used as positive or negative treatment. However, the lack of feedback can result in under- or over- confidence biases. We tried our best to use neutral wording [10] to minimize this effect in this study.

5 Conclusion

In this paper, we reported our study result on how CMMN and DCR has been perceived in terms of usefulness and ease of use. The study is performed by applying the technology acceptance model, where we educated master level students with these languages over eight weeks by giving feedbacks to reduce perception biases. The students' perceptions are collected through two questionnaires as two sub-studies (studies 1 and 2). We collected students' opinions before and after sending feedback on their final practice in the exam in study 1 and study 2, respectively. In total, twenty students participated in Study 1 among which thirteen students also participated in Study 2.

The study result indicates that both languages have an acceptable level for both perceived usefulness and ease of use. The students' perceptions have changed a little before and after receiving the feedback. We performed three nonparametric statistical significance tests, and the result indicates that the difference is not significant. We also evaluated the reliability of responses using Cronbach's alpha. The result showed an acceptable level of reliability in students' responses.

As future work, it is interesting to investigate how prior knowledge on workflow-based business process modeling can influence users' perception when learning declarative modeling languages. It is also interesting to perform this study with more participants and different backgrounds.

Acknowledgement. We appreciate all support that is provided by Morten Marquard from dcrgraphs.net, without which it was difficult to train the students and perform this study.

References

1. Abbad Andaloussi, A., Buch-Lorentsen, J., López, H.A., Slaats, T., Weber, B.: Exploring the modeling of declarative processes using a hybrid approach. In: Laender, A.H.F., Pernici, B., Lim, E.-P., de Oliveira, J.P.M. (eds.) ER 2019. LNCS, vol. 11788, pp. 162–170. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33223-5_14
2. Abbad Andaloussi, A., Burattin, A., Slaats, T., Petersen, A.C.M., Hildebrandt, T.T., Weber, B.: Exploring the understandability of a hybrid process design artifact based on DCR graphs. In: Reinhartz-Berger, I., Zdravkovic, J., Gulden, J., Schmidt, R. (eds.) BPMDS/EMMSAD 2019. LNBIP, vol. 352, pp. 69–84. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20618-5_5

3. Abbad Andaloussi, A., Davis, C.J., Burattin, A., López, H.A., Slaats, T., Weber, B.: Understanding quality in declarative process modeling through the mental models of experts. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) BPM 2020. LNCS, vol. 12168, pp. 417–434. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58666-9_24
4. Abbad Andaloussi, A., Slaats, T., Burattin, A., Hildebrandt, T.T., Weber, B.: Evaluating the understandability of hybrid process model representations using eye tracking: first insights. In: Daniel, F., Sheng, Q.Z., Motahari, H. (eds.) BPM 2018. LNBIP, vol. 342, pp. 475–481. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11641-5_37
5. Arachchilage, N.A.G., Love, S.: A game design framework for avoiding phishing attacks. Comput. Hum. Behav. **29**(3), 706–714 (2013)
6. Chae, M., Kim, J., Kim, H., Ryu, H.: Information quality for mobile internet services: a theoretical model with empirical validation. Electron. Mark. **12**(1), 38–46 (2002)
7. Davis, F.D.: A technology acceptance model for empirically testing new end-user information systems: theory and results. Ph.D. thesis, Massachusetts Institute of Technology (1985)
8. Davis, F.D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Q. **13**, 319–340 (1989)
9. Di Ciccio, C., Marrella, A., Russo, A.: Knowledge-intensive processes: characteristics, requirements and analysis of contemporary approaches. J. Data Semantics **4**(1), 29–57 (2015)
10. Eberlein, M., Ludwig, S., Nafziger, J.: The effects of feedback on self-assessment. Bull. Econ. Res. **63**(2), 177–199 (2011)
11. Fahland, D., et al.: Declarative versus imperative process modeling languages: the issue of understandability. In: Halpin, T., et al. (eds.) BPMDS/EMMSAD -2009. LNBIP, vol. 29, pp. 353–366. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01862-6_29
12. Fahland, D., Mendling, J., Reijers, H.A., Weber, B., Weidlich, M., Zugal, S.: Declarative versus imperative process modeling languages: the issue of maintainability. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 477–488. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12186-9_45
13. Haisjackl, C., et al.: Understanding declare models: strategies, pitfalls, empirical results. Softw. Syst. Model. **15**(2), 325–352 (2016)
14. Hildebrandt, T., Mukkamala, R.R.: Distributed dynamic condition response structures. In: PLACES Workshop (2010)
15. Hildebrandt, T., Mukkamala, R.R., Slaats, T.: Nested dynamic condition response graphs. In: Arbab, F., Sirjani, M. (eds.) FSEN 2011. LNCS, vol. 7141, pp. 343–350. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29320-7_23
16. Hildebrandt, T.T., Mukkamala, R.R.: Declarative event-based workflow as distributed dynamic condition response graphs. arXiv preprint arXiv:1110.4161 (2011)
17. Hull, R., et al.: Business artifacts with guard-stage-milestone lifecycles: managing artifact interactions with conditions and events. In: Distributed Event-based System Conference, pp. 51–62 (2011)
18. Hull, R., et al.: Introducing the guard-stage-milestone approach for specifying business entity lifecycles. In: Bravetti, M., Bultan, T. (eds.) WS-FM 2010. LNCS, vol. 6551, pp. 1–24. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19589-1_1

19. Jalali, A.: Weaving of aspects in business process management. Complex Syst. Inform. Model. Q. **15**, 24–44 (2018)
20. Jalali, A., Maggi, F.M., Reijers, H.A.: A hybrid approach for aspect-oriented business process modeling. J. Softw. Evol. Process **30**(8), e1931 (2018)
21. Jošt, G., Huber, J., Heričko, M., Polančič, G.: Improving cognitive effectiveness of business process diagrams with opacity-driven graphical highlights. Decis. Support Syst. **103**, 58–69 (2017)
22. La Rosa, M., Ter Hofstede, A.H., Wohed, P., Reijers, H.A., Mendling, J., Van der Aalst, W.M.: Managing process model complexity via concrete syntax modifications. IEEE Trans. Industr. Inf. **7**(2), 255–265 (2011)
23. La Rosa, M., Wohed, P., Mendling, J., Ter Hofstede, A.H., Reijers, H.A., van der Aalst, W.M.: Managing process model complexity via abstract syntax modifications. IEEE Trans. Industr. Inf. **7**(4), 614–629 (2011)
24. Masrom, M.: Technology acceptance model and e-learning. Technology **21**(24), 81 (2007)
25. Object Management Group (OMG®). Case Management Model and Notation (CMMN™)
26. Palm, J.-M., Colombet, I., Sicotte, C., Degoulet, P.: Determinants of user satisfaction with a clinical information system. In: AMIA Annual Symposium Proceedings, vol. 2006, p. 614. American Medical Informatics Association (2006)
27. Pesic, M.: Constraint-based workflow management systems: shifting control to users (2008)
28. Pesic, M., Schonenberg, H., Van der Aalst, W.M.: Declare: full support for loosely-structured processes. In: EDOC, p. 287. IEEE (2007)
29. Pichler, P., Weber, B., Zugal, S., Pinggera, J., Mendling, J., Reijers, H.A.: Imperative versus declarative process modeling languages: an empirical investigation. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM 2011. LNBIP, vol. 99, pp. 383–394. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28108-2_37
30. Reijers, H.A., Slaats, T., Stahl, C.: Declarative modeling–an academic dream or the future for BPM? In: Daniel, F., Wang, J., Weber, B. (eds.) BPM 2013. LNCS, vol. 8094, pp. 307–322. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40176-3_26
31. Sanchez-Ferreres, J., et al.: Supporting the process of learning and teaching process models. IEEE Trans. Learn. Technol. **13**(3), 552–566 (2020)
32. Turetken, O., Dikici, A., Vanderfeesten, I., Rompen, T., Demirors, O.: The influence of using collapsed sub-processes and groups on the understandability of business process models. Bus. Inf. Syst. Eng. **62**(2), 121–141 (2019). https://doi.org/10.1007/s12599-019-00577-4
33. 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
34. Weber, B., Reijers, H.A., Zugal, S., Wild, W.: The declarative approach to business process execution: an empirical test. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 470–485. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02144-2_37
35. Zugal, S., Soffer, P., Haisjackl, C., Pinggera, J., Reichert, M., Weber, B.: Investigating expressiveness and understandability of hierarchy in declarative business process models. Softw. Syst. Model. **14**(3), 1081–1103 (2013). https://doi.org/10.1007/s10270-013-0356-2

Subject-Oriented Reference Model for Virtual Factory Operations Commissioning

Matthes Elstermann[1](✉), Stefanie Betz[2], Matthias Lederer[3], Werner Schmidt[4], and Lisa Bührer[1]

[1] Karlsruhe Institute of Technology, Karlsruhe, Germany
matthes.elstermann@kit.edu
[2] Hochschule Furtwangen, Furtwangen, Germany
[3] Technical University of Applied Sciences Amberg-Weiden, Amberg-Weiden, Germany
[4] Technical University Ingolstadt, Ingolstadt, Germany

Abstract. The commissioning of manufacturing plants is considered a complex and at the same time critical phase in the life cycle of a production system, because errors and delays lead to long downtimes and cost-intensive re-work at the start of production. Optimizations and rationalisation of this phase can be implemented by employing the idea of virtual commissioning. This paper summarises conceptual approaches to virtual commissioning and concertises them with the help of subject-oriented process models. As reference models, these depict the plant design process and demonstrate the advantage of virtual commissioning in terms of time, cost, and quality. The discussion along recognised quality criteria show that the reference model is suitable for providing orientation for the people involved in the plant engineering process and that the model can be used to support the creation of company-specific models.

Keywords: Subject orientation · Reference model · Factory operations commission

1 Introduction

The point at which a product is ready to be produced is a decisive factor in the economic success of a product [28]. A quick production deployment with no error rate is the goal repeatedly called for in science and practice. At the same time, an increasing product complexity as well as a large amount of variants and shorter product life cycles are reflected in more complicated requirements for the development of the producing machines and systems. Factory operations commissioning, in short commissioning or deployment, plays a key role in system development. It accounts for up to 25% of the entire project duration [27]. Due to the individual design of the systems and adjustments to customer requirements, the development is often unique and the commissioning project is accordingly

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 163–177, 2021.
https://doi.org/10.1007/978-3-030-79186-5_11

difficult to plan. Errors that were not recognised in previous work steps come together here. Long downtimes and delays are the result. The elimination of software errors takes up most of the commissioning time [27]. Accordingly, the increasing share of information technology in automated production systems defines new challenges for the commissioning but also possible solutions for its growing complexity. The virtual commissioning offers the possibility of testing the control technology at an early stage in the development process using a real-time system model. In this way it is possible to improve the software quality and avoid incalculable loss of time. By anticipating and paralleling important work steps, the overall throughput and commissioning time can be shortened. There are numerous phase and procedural models that depict the plant development process at a high abstraction level. However, presentation of the interdisciplinary development as well as parallel and iterative activities is problematic. Virtual commissioning of products is even more complex and needs a common language to argue about. Furthermore, there are many schematic drawings to illustrate the potential of a virtual commissioning, but no process model that depicts the sequence correctly in terms of time. Thus, a formal model is needed to be able to use completion tools. Therefore, the goal of the paper is to present a reference model for virtual commissioning. The reference model should be presented using a formal process modeling language to be able to depict and simulate the virtual commissioning process. For this reason the concept of subject orientation has been chosen for the model, as the formal, per-role/per-subject-instructions should allow better understandability and enable numeric quantification (see related work in the section on fundamentals).

The paper is structured as follows: In the next section we present fundamentals on virtual commissioning to sharpen the addressed research gap. This is followed by a background section on process models and subject orientation. Section 3 explains our reference model). Subsequently we discuss our work and close the paper with a summary and outlook.

2 Fundamentals

There are numerous terms in literature and practice that describe commissioning (e.g. [21,23,26]). What the definitions have in common is that commissioning initially means setting up a technical system. Commissioning in the context of manufacturing includes all activities that are required to start up and ensure the correct functioning of assemblies, machines, and complex systems that have been previously assembled and checked for proper assembly. Traditionally, the mechanical-oriented approaches dominated the design of mechanical engineering systems in the literature. However, with increasing digitalization the synergistic and equal interaction of mechanical engineering, electrical engineering and information technology is being promoted. The systems to be designed consist of a series of sensors, actuators, and information processing units [12,20]. A look into literature reveals a number of models, that consider commissioning and its role in product life cycle concepts (e.g. [5,6,13,16,23]). Commissioning is recognised in all lifecycle models as a critical process and, in particular, brings with it a high potential for rationalisation (cf. [6]). Commissioning is a complex and therefore, especially if it does not run smoothly, takes a long time, has negative effects on downtimes, capital commitment, delivery times, and reputation with customers, among other things. In the classic golden triangle of project management, [9] describe the three central advantages to be gained when optimising project setups: (i) time - lead time can be reduced, (ii) costs - productivity can be increased and (iii) quality - disturbances in the real commissioning (RC) can be reduced.

Testing of the control system prior to RC has been carried out since the introduction of programmable logic controllers. Traditionally, time-consuming and costly test environments have been used for this purpose. Virtual commissioning (VC), on the other hand, is the anticipation of control commissioning using a real-time capable plant simulation. It is recognised that the process of a VC is divided into three phases [5,23,24,29], in which the previous findings from the literature can be presented.

(i) First, the necessary data of the plant specification and its components must be available for (simulation)*model creation*. The model may reproduce the mechanical, hydraulic, pneumatic, and electrical components of the system. It is considered ideal to create the simulation model in parallel to the design activities and to achieve an interdisciplinary integration. CAD data, functional descriptions, and I/O lists, e.g., form the starting point for creating the simulation model (SM). Often, a huge amount of information from different areas has to be compiled and analysed.

(ii) After the model has been created, the *VC execution* follows. In this way, the control programme can be tested simultaneously with the production of the system. If real control technology is used to carry out the VC, the term hardware-in-the-loop (HIL) is used in the literature. In practice, HIL simulation is the most commonly applied approach. To test the interface behaviour of the control system, the real control system is coupled with the virtual simulation model. Fieldbus systems are often used for this purpose. If instead the control technology is described by a model, it is a software-in-the-loop simulation (SIL). In this simulation type, an emulation of the real control system is used [3,17,18]

(iii) The steps of *implementation and evaluation* are typically carried out iteratively. Based on the simulation results, disturbance hypotheses are developed. Then, the control programme is modified and tested again. When the simulation runs without errors the tested and optimised control programme is released for real commissioning (RC). Test protocols can be several hundred pages long and contain a concrete list of test steps that must be successfully carried out before release [3,14].

In a nutshell, the virtualization of these activities has the potential to tackle many operational problems of commissioning. It means to anticipate the actual behaviour during commissioning of the system by using a real-time simulation. Other than the traditional commissioning, the system integration and testing can already start early during system design and run in parallel with development and manufacture, feeding back the findings to these activities. Additionally, virtual commissioning allows for early integrating all relevant roles in the overall system development process. This can improve communication and mutual understanding and fosters a quick and flawless commission. However, it is important to note that virtual commissioning does not replace real commissioning, but precedes it and helps to improve its duration, costs and quality.

All explained concepts show promising approaches for virtual commissioning. However, in the recognised process hierarchy models (e.g. according to [19]), they are on level 1, i.e. they give guidance (e.g. phases) on the level of rough process steps. Instead of conceptual level 1, however, as described in the introduction, precise and specific process steps are necessary, especially for the complex and challenging virtual planning of production. In the following section, this paper therefore develops a reference model with specific activities and actors that, while not fitting well into a level/phase concept, best can be seen to be on the recognised level 3 of the process hierarchy model. It thus serves as a concretion of the Related Work in a recognised and standardised modelling notation (S-BPM).

2.1 Subject Orientation

With the subjects, in addition to the functions (predicates) and the data (objects), the modeling paradigm of Subject Orientation (integral to the methodology of Subject-Oriented Business Process Management (S-BPM) focuses on the active elements (actors) in a process and their communication [10]. Subject-oriented models therefore implicitly contain subject, predicate, and object, and thus are based on the standard sentence semantics of natural languages. Subjects play the role of "primus inter pares". They are abstract, process-specific representations of actors (people, systems) who actively drive the process [10].

The methodology is based on a process algebra for modeling parallel processes and takes up work from computer science from the 1980s. The addition of a graphic notation and the integration of aspects of object orientation resulted in the formal modeling language of the Parallel Activity Specification Scheme (PASS) [10,11].

In comparison to other process modeling languages, PASS requires only a few symbols and has clear formal semantics. This allows the graphical specifications to be seamlessly transferred into executable models [4,10].

Subject-oriented process descriptions comprise two types of diagrams:

(1) Subject Interaction Diagram (SID): This type of diagram depicts the actors involved in the process as subjects and the information exchanged between them as message types (see Fig. 1). The modeler uses this to structure the communication in the course of process execution.

(2) Subject Behavior Diagram (SBD): For each involved subject, a Subject Behaviour Diagram describes the order of activities that an instance of the subject should carry out at runtime. While those actions are strictly sequential within a subject, different subjects can act in parallel, synchronizing themselves by message exchange. Activities within and interactions between subjects are described using three states and the associated state transitions (see Fig. 2): (i) Do or functional state/transition afterwards specifies the result of the activity

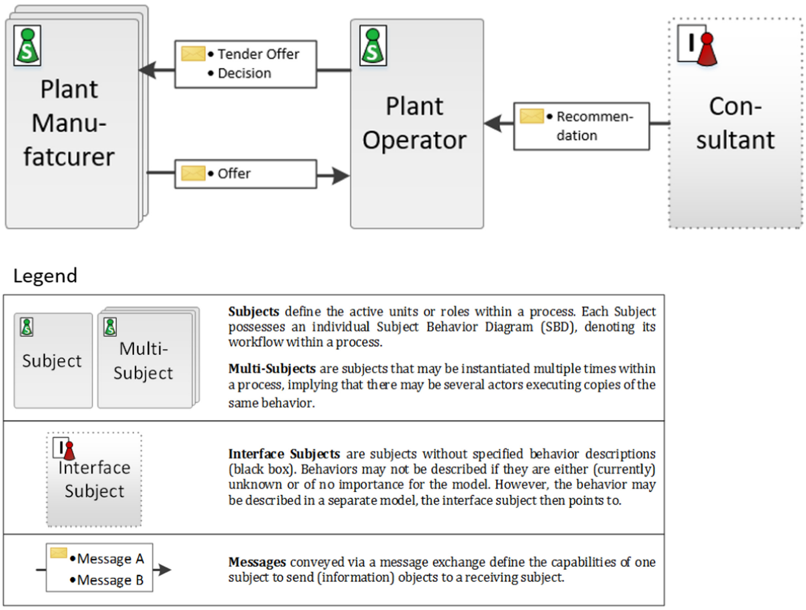


Fig. 1. SID example

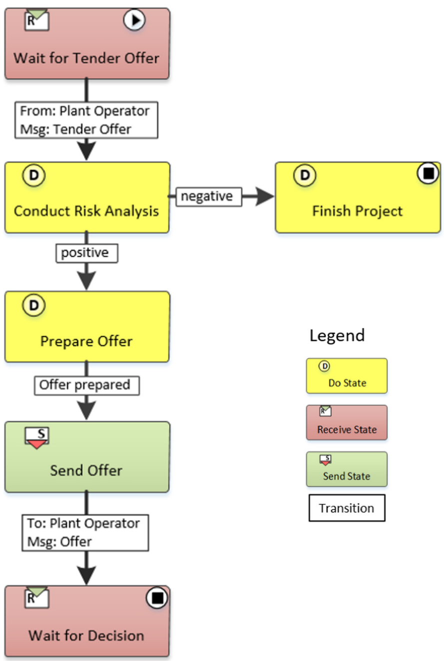


Fig. 2. SBD plant manufacturer

(ii) Send state/transition afterwards specifies the recipient and message type (iii) Receive state/transition afterwards specifies the sender and message type.

The simple example in Figs. 1 and 2 depicts the connection between the diagram types. The SID shows the interaction of three subjects in a process, where a plant operator obtains offers from multiple plant manufacturers. After having received the offers, he makes and communicates his decision back. The consultant just gives some recommendation. As its behaviour is of no interest here, it is modeled as an interface object for which no SBD is necessary.

The manufacturer's behaviour is visible in Fig. 2. It starts with a Receive state, indicating that activities are triggered by an incoming tender offer. The state is marked as the unique starting point by the little 'Play' button. A risk analysis for such an offer is subject of the following Do state, splitting the sequence. The arch for the negative result leads to an end in the simplified example here, marked by the 'Stop' button. In reality more steps like sending a rejection message would follow. In case of a positive result, the offer is prepared in a Do state and subsequently sent to the plant operator. The behaviour description ends with the Receive state for waiting for the operator's answer to the offer, i.e. the decision. Here also more steps would probably be modeled in reality, e.g., a split with either triggering the order management process or internal handling of a refusal.

The SBD of the plant operator would be described analogous.

2.2 Process Reference Model

A business process reference model often describes common practices in a specific domain. Beside communicating those and guiding process improvement and benchmarking, the main purpose is offering a starting template for customizing the more or less generic pattern of the reference model to the needs of a concrete organization, thus facilitating the creation of individual, enterprise-specific models. Designing a process reference model requires deep understanding of the domain the model is intended for, as well as a high expertise in the modeling concept and language used [15,22].

3 The VFOC Process Reference Model

As a starting point for our research work and to deeply dive into the processes under review, we first developed a subject-oriented reference model of the traditional/conservative commissioning process. Due to space limitations the resulting model is not shown here, but is available at [2].

3.1 Model Structure

Rather, this section presents and discusses the structure of the model for virtual factory operations commissioning (VFOC) we derived from the conservative model based on literature and good practice (see Sect. 2). Due to its overall size, even this model cannot be presented as whole in this work. However, the Subject Interaction Diagram (SID) will be discussed as well as two of the Subjects Behaviour Diagrams.

As Fig. 3 shows, the main roles involved in a virtual commissioning process are classical:

– The 'Design department': responsible for the actual development of machine or production system that is to be employed
– The 'Production and Construction' team: responsible for the actual creation or procurement of the parts physically necessary to build the production system
– The 'Initial Operations Ramp Up Team' responsible for the actual construction and operations commissioning on-site at the customer's location
– The 'Programmers': Software engineers responsible to create the control systems and programs for, assumedly, the at least partially automated machine/production system
– The 'Virtual Factory Operations Commissioning (VFOC) Team': The only role that is non-existent in a classical operations commissioning processes and that all deviations center around. People responsible for this subject are the ones creating and controlling all information and tools necessary for simulating a production system up to a virtual reality level of integration in order to reap the potential benefits of that method.

Theses essential roles a framed by two interface subjects:

– The general 'Project Planning Team' that initialises and is responsible for the overall endeavour.
– The 'Facility Operator' or customer that triggers the overall process and for whom the factory-to-be-commissioned is intended (and who will pay when everything is in order, but also might sue for compensation if the commissioned production system does not run as specified).

The message flow between the subjects shows the intensive communication between all participants. It particularly depicts how early and intensely the VFOC team is embedded in the whole process. The SID here also includes communication channels between subjects (bold lines with squares at the end). Compared to messages, they represent non-directed and less specific, partially informal, information exchange[1].

Of the five SBDs of the overall model also only two are shown here to give an impression of the inner workings of the model.

The SBD of the Design team (Fig. 4) describes the principle tasks of developing the to-be-deployed-and-commissioned production facility. The actual tasks are done in Do states that are the same than they would be in a set-up without a dedicated Virtual Factory Operations Commission (VFOC) subject.

[1] Communication channels were not explained in Sect. 2.1 as none of them occured in the simple SID there.

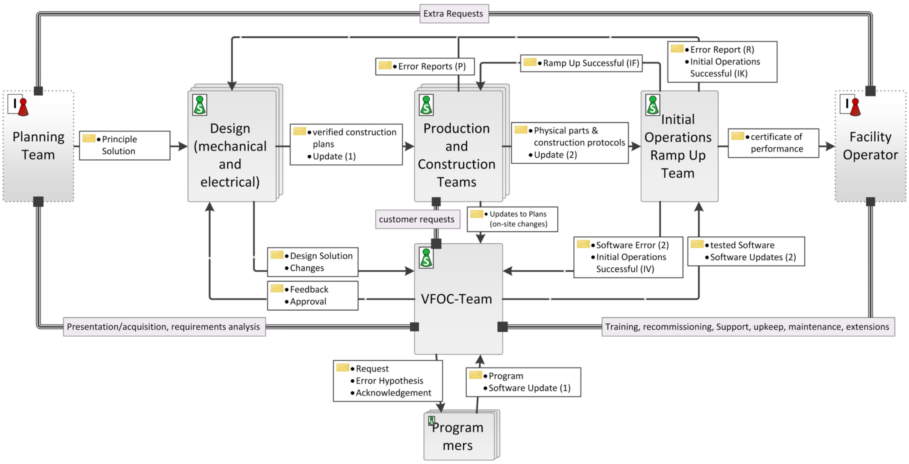


Fig. 3. SID of the referential VFOC process model

It could be argued that the incorporated first feedback loop is barely more than a slightly advanced factory planning with incorporated state-of-the-art simulation. And for their principle activities that would be true. The reference model proposes to separate both concerns, factory planning and the creation of information/simulation model that can be part of a full VFOC effort. The real difference for the planning comes in the second feedback loop that will occur once the actual construction and commissioning has started and has unearthed possible planning problems[2]. If, as intended by the model, the VFOC system has been updated with the current changes, the feedback loop to handle changes during the actual changes should be faster and more efficient as it is based on current up-to-date information and no extensive integration work needs to be done manually.

The principle process to organise the technical pre-condition for functional Virtual Factory Operations Commission (VFOC) efforts are described in the SBD of the VFOC subject (Fig. 5). Naturally, technical expertise is necessary to actually create and operate the VFOC system. These partially quite sophisticated tasks are described in the Do states of the SBD. The Send and Receive states reflect the SID and how the VFOC team is heavily involved in all aspects of the endeavour. The underlying state groups are the visual aid that emphasise the three principle tasks. The first is the actual creation and evaluation of the simulation model, basically the 'Initial Virtual Production Planning' that is not largely different from a normal planning effort with integrated simulation. However, the second block of 'virtual factory operations commissioning' depicts the concept that programmers responsible for the creation of machine control software should be able to use the virtual model as a testing ground. Furthermore,

[2] In an ideal world this will not happen. However, Murphy's colloquial law does allow the assumption that something WILL always go wrong.

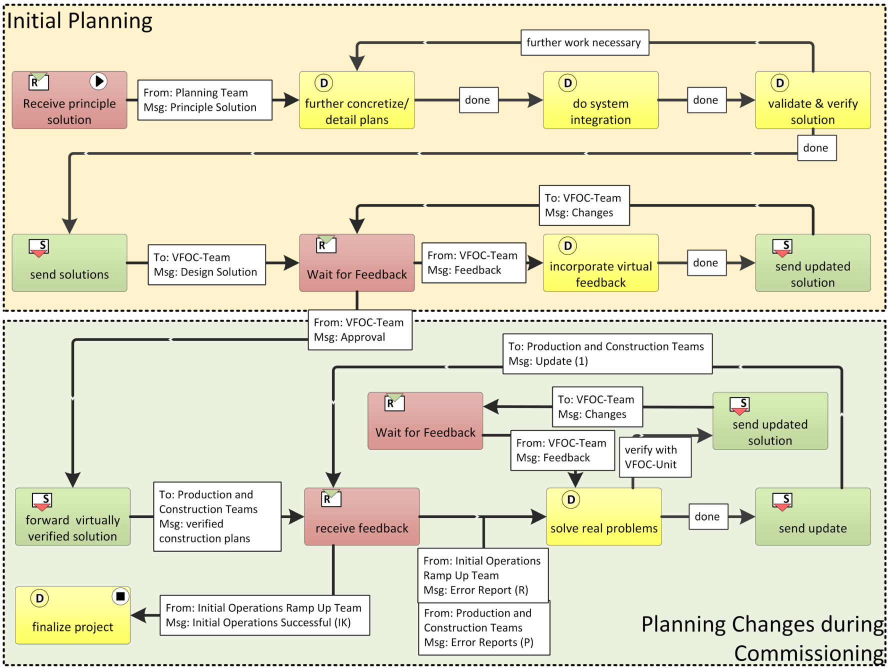


Fig. 4. Subject behaviour diagram for subject ‘Design’ from Fig. 3

this is not necessarily happening in sequence and only once. Rather, changes in the design made by the design team or deviations from the original plan made during construction of the to-be-controlled machines are constantly factored into the simulation system and included into the feedback for the programmers. By the time the factory can actually go into operations the virtual model should still be up to date and, ideally be usable for pre-training of future factory staff using, e.g., VR technology. Finally, if problems should still occur during the actual (non-virtual) commissioning procedures of the planned factory, necessary changes to the control software can also be tested and improved in the virtual environment before actually being deployed. Potentially this may avoid longer feedback cycles of transferring software to the site and waiting for the results of the real system[3].

[3] This is also the case if the virtual model serves as a digital twin of the future production system and updates will be input remotely. The software update still needs to be tested.

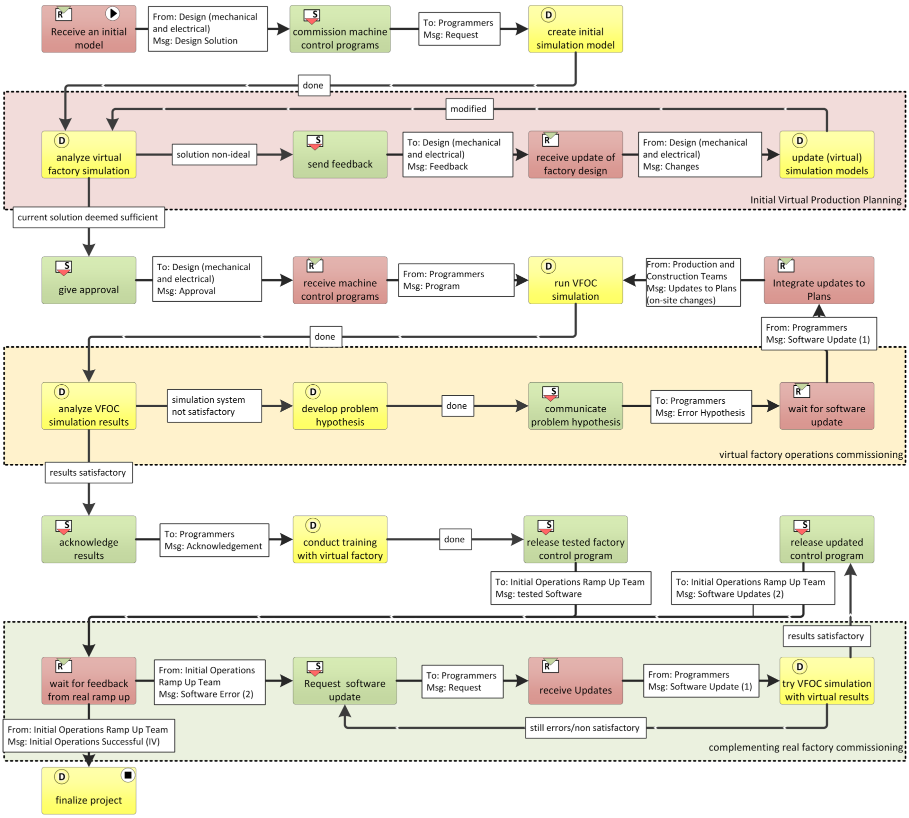


Fig. 5. Subject behaviour diagram for subject 'VFOC Team' from Fig. 3

3.2 Simulation

The main purpose of the model is, as stated, to provide a reference. At the same time, one of the questions driving our research was how it is possible to estimate and quantify a potential impact of such a virtual factory commissioning process. This is quite tricky. In an ideal world, where planning is done only perfectly once and software is free of bugs and will work out of the box there would be no need for such a virtual system and the involved efforts. However, it can be stated with certainty that this never will be case[4]. The assumption is having a virtual factory model involved in the planning will basically have two impacts. First it will increase the speed of feedback cycles and more easily and profoundly allow later planning changes. Secondly, and in correlation to the first, it will increase the chances that errors in planning, design, and/or software will be found early on. However, it cannot prevent them (completely). Effects will therefore only

[4] If claimed to be the case then only in the most trivial cases and/or with huge error buffers/margins in all planning efforts.

appear in the overall speed and the chances to have unwanted events occur during the final factory commission process.

To quantify effects on the process comparatively, both models, the reference model with VFOC as well as the conservative one will need to serve as the basis of simulation. While, as stated, the use of the PASS modeling language and the subject-oriented paradigm is potentially advantageous for the main purpose of the reference model, they also come with the benefit of being a formal and executable modeling language at the same time. This property allows for applying the Simple Simulation method (SiSi), which was developed particularly for the executable PASS models [7]. For running simulations only assumptions for duration of individual task and probability values for the branching need to be annotated. Figure 6 shows an example for such values.

We have fed the models with time and chance values based on the literature and reasonable assumptions and compared results. Discussion of chosen values and the results as well as their limits must be detailed in another work. Initial evaluations indicate that, with relative conservative values and estimations, the simulation scenarios allow the following quantification: the scenario with VFOC had an average lead-time reduction of 15% as well as a reduction of average on-site commissioning time of up to 75%. However, as stated, these are preliminary results and will be discussed in future works. For now, the models for traditional and virtual factory commissioning can be downloaded for review at [2].

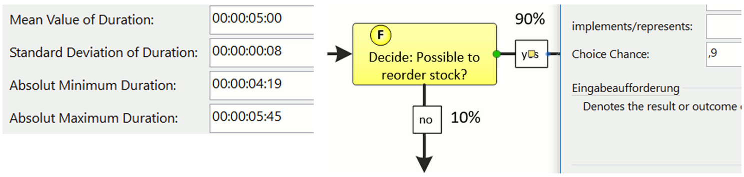


Fig. 6. Example for PASS Simple Simulation (SiSi) Inputs

4 Discussion

4.1 Process Reference Model Quality

In order to check the quality of the reference model for virtual deployment we apply the framework of Taylor and Bandara [25]. They identified syntactic, semantic, and pragmatic quality as dimensions and assigned user-perceived attributes to them.

Syntactic quality refers to the language of the model, including the implicit meta-model and the underlying concepts. It also comprises the strict application of the grammar, i.e. constructs and rules to combining them, their mapping to real world elements (ontological aspects) and their consistent use (e.g. by naming conventions) [25]. Consequently, according to Taylor and Bandara, major attributes to assess these aspects are that the language is clearly defined, used consistently, and easy to understand. A glossary for key terms and training for explaining the terminology should add to that [25].

As shown in Sect. 2.1, the subject-oriented PASS language, is based on complete active sentences in natural language with subject, predicate and object, and has a well understood meta-model. As a consequence, models are clearly defined and easy to comprehend both on (business) user and on IT developer level. Consistent use of the language can be partially guaranteed by modeling software, which enforces adhering to the formal syntax of the language. This e.g. includes preventing from sending or receiving messages in behaviour diagrams, which do not appear in the interaction diagram. Also, e.g., the consistency of labels on the respective transitions is given (name of sending/receiving subject, name of message). However, some labeling still requires conventions to keep them consistent (e.g. strictly identifying objects/messages as nouns or using verbs in present tense as predicates for do states). This is not unusual, as the freedom for labeling symbols as model entities is valid for modeling languages in general, and solved by a list of conventions. The PASS language structure and elements are subject to standardisation efforts of the respective research community. The current status of the results in form of the 'Standard PASS Ontology' [8], developed in the Web Ontology Language (OWL) and using the Abstract State Machine (ASM) concept [4,10] is available on GitHub [1].

Considering the discussion above, the presented subject-oriented process reference model largely covers main criteria of syntactic quality. Of course, this claim needs to be evaluated in more detail, e.g. in the field of virtual deployment practice.

Semantic quality is achieved if the reference model captures what is in its scope. That is, if it captures 'everything of relevance and nothing of irrelevance' [25]. Therefore, the quality attributes assigned to evaluate the semantic quality of a reference model are [25]: (i) relevance, (ii) completeness, (iii) correctness, and (iv) consistency. In the following, we apply the attributes to our developed reference model. The first attribute (i) relevance must be judged within the light of the scope of the model. The scope is to support virtual commissioning to improve quality and save money (see Sects. 1 and 2). As shown, virtual commissioning is able to improve many operational problems of commissioning. It offers the possibility of testing the control technology and by anticipating important work steps and paralleling work, the overall throughput can be shortened. The reference model is relevant as by using it time can be saved. Additionally, a reference model offers the possibility to tailor it to specific company needs and to enable mutual understanding. This possibility is provided by applying subject orientation. As explained, this paradigm enables flexibility to adapt the model and is based on the natural language, thus it is easy to understand. The next attribute (ii) completeness is always difficult to tell. Overall, a reference model should cover standard cases as well as exceptions and contain more information than a specific model [25]. We have derived the model from literature and good practice (see Sect. 2). Thus, we included standard cases as well as exceptions (see Sect. 3). Nevertheless, an additional expert evaluation would enhance the model quality further. Additionally, we validated the model in a simulation experiment. (iii) Correctness of the model has been proven insofar, as, when simulating it, time savings have been achieved. Additionally, unnecessary repetitions are avoided and as stated, the model is based on good practices. Lastly, (iv) consistency, that no model statement contradicts another, is again proven via simulation. To sum up, regarding semantic quality, we can conclude that a sufficient quality of the reference model is achieved.

As a final criterion area, to assess whether the reference model is of good quality, *Pragmatic issues* will be addressed. Good reference frameworks provide approaches for quick but also sustainable applicability, e.g. in business practice. First of all, the established model provides a concrete and extended documentation, in order to profit from the classical advantages of the virtual commissioning. The presented diagrams can serve as first case studies for the use of the model and should be further enriched (e.g. by implementation guides). The use of the subject orientation and not everyday notation language (like BPMN) may force model users to familiarise themselves with the model as well as the underlying notation. However, the model is based on a fixed notation standard, so that the reference is generally easily accessible - in particular, the consideration of subjects instead of BPMN as presented can also serve the purpose of virtual commissioning. The reference model is based on related reference models, literature and practitioner's interviews so that it is grounded in practitioners knowledge - likewise, feedback from stakeholders already flowed into iterations. Nevertheless, it is a model tailored to specific needs, specifically virtual commissioning. In sum, the quality criteria of the 'pragmatism' area tend to be met. To summarise, all three quality criteria are met to a sufficient extend, especially the syntactic and semantic quality. Nevertheless, the evaluation of the quality of the model is partially based on the subjective judgement. However, the evaluation is based on objective arguments whenever possible and, if not, reasonable arguments for our judgement are provided.

5 Summary

In this paper we presented a reference model for virtual commissioning. We developed our reference model based on existing models, practitioners interviews and literature and depicted and simulated it using PASS, a formal and easy to understand process modeling notation[5]. Using our reference model thus enables different stakeholders to improve communication and mutual understanding between them and it enables numeric quantification. We evaluated the quality of our model applying the framework of Taylor and Bandara [25] for analysing a process reference model. We evaluated the syntactic, semantic, and pragmatic quality of our model and came to the conclusion that we achieved good syntactic and semantic quality and partially a good pragmatic quality.

[5] The model can be reviewed by accessing it at [2].

Currently, we are conducting an in-depth case study to answer open questions of pragmatic quality like implementation and customising of the proposed reference model.

References

1. A standard for subject-oriented specification of systems. https://github.com/I2PM/Standard-Documents-for-Subject-Orientation. Accessed 16 Mar 2021
2. Subject-oriented reference models for traditional and virtual commissioning. https://github.com/MatthesElstermann/VirtualFactoryOperationsCommisionReferenceModel. Accessed 16 Mar 2021
3. Brökelmann, J.: Systematik der virtuellen Inbetriebnahme von automatisierten Produktionssystemen. University of Paderborn, Paderborn (2015)
4. Börger, E., Stärk, R.: Abstract State Machines: A Method for High-level System Design and Analysis. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-642-18216-7
5. Davis, B.: How to validate machines with virtual commissioning: Virtual commissioning begins with a vision of the desired machine behavior and sequence of operation. Plant Eng. **74**(7) (2020)
6. Dominka, S.: Hybride Inbetriebnahme von Produktionsanlagen. Von der virtuellen zur realen Inbetriebnahme. Sierke, Göttingen (2007)
7. Elstermann, M., Ovtcharova, J.: Sisi in the ALPS: a simple simulation and verification approach for pass. In: Proceedings of the 10th International Conference on Subject-Oriented Business Process Management. S-BPM One 2018. Association for Computing Machinery, New York (2018)
8. Elstermann, M., Wolski, A.: Matching execution and modeling semantics for subject-oriented process models. In: Freitag, M., Kinra, A., Kotzab, H., Kreowski, H.J., Thoben, K.D. (eds.) Subject-Oriented Business Process Management: The Digital Workplace - Nucleus for Transformation. No. 1278 in Communications in Computer and Information Science. Springer (2020)
9. Eversheim, W., Koerth, D., Gentzke, J., Walenta, F., Kettner, P., Esser, H.: Integrierte Produktentwicklung. Denkabläufe, Methodeneinsatz, Zusammenarbeit. Hanser, München (2009)
10. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Börger, E.: Subject-Oriented Business Process Management. Springer, Berlin (2012). https://doi.org/10.1007/978-3-642-32392-8
11. Fleischmann, A.: Distributed Systems - Software Design and Implementation. Springer, Berlin (1994). https://doi.org/10.1007/978-3-642-78612-9
12. Gausemeier, J., Schäfer, W., Anacker, H., Bauer, F., Dziwok, S.: Einsatz semantischer technologien im entwurf mechatronischer systeme. In: Gausemeier, J., Rammig, F.J., Schäfer, W., Trächtler, A. (eds.) 8. Paderborner Workshop Entwurf mechatronischer Systeme. Verlagsschriftenreihe des Heinz Nixdorf Instituts, Paderborn, vol. 294, pp. 7–35. Heinz Nixdorf Institut, 19–20 May 2011
13. Hand, A.: Digital twin makes virtual commissioning a reality. Autom. World **17**(1), 1 (2019)
14. Hoffmann, P.: On Virtual Commissioning of Manufacturing Systems - Proposal for a systematic VC simulation study methodology and a new simulation model building approach. University of Applied Sciences Hannover, Hannover (2016)

15. Küster, J.M., Koehler, J., Ryndina, K.: Improving business process models with reference models in business-driven development. In: Eder, J., Dustdar, S. (eds.) BPM 2006. LNCS, vol. 4103, pp. 35–44. Springer, Heidelberg (2006). https://doi.org/10.1007/11837862_5
16. Makris, S., Michalos, G., Chryssolouris, G.: Virtual commissioning of an assembly cell with cooperating robots. Adv. Decision Sci. (2012)
17. Menzel, T., Stoeppler, G., Douglas, S.: Hardware-in-the-loop simulation of machine tools and manufacturing systems. Comput. Control Eng. **14**, 10–15 (2005)
18. Mihoubi, B., Bouzouia, B., Tebani, K., Gaham, M.: Hardware in the loop simulation for product driven control of a cyber-physical manufacturing system. Prod. Eng. Res. Devel. **14**(3), 329–343 (2020). https://doi.org/10.1007/s11740-020-00957-w
19. zur Muehlen, M., Wisnosky, D., Kindrick, J.: Primitives: design guidelines and architecture for BPMN models. In: Proceedings of ACIS 2010. ACIS (2010)
20. Nikolakis, N., Alexopoulos, K., Xanthakis, E., Chryssolouris, G.: The digital twin implementation for linking the virtual representation of human-based production tasks to their physical counterpart in the factory floor. Int. J. Comput. Integr. Manuf. **32**(1), 1–12 (2019)
21. Poeschl, S., Wirth, F., Bauernhansl, T.: Strategic process planning for commissioning processes in mechanical engineering. Int. J. Prod. Res. **57**(21), 6727–6739 (2019)
22. La. Rosa, M., Gottschalk, F., Dumas, M., van der Aalst, W.M.P.: Linking domain models and process models for reference model configuration. In: ter Hofstede, A., Benatallah, B., Paik, H.-Y. (eds.) BPM 2007. LNCS, vol. 4928, pp. 417–430. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78238-4_43
23. Slansky, D.: Virtual commissioning evolves into model-driven digital twin. Autom. World **18**(8), 41 (2020)
24. Spitzweg, M.: Methode und Konzept für den Einsatz eines physikalischen Modells in der Entwicklung von Produktionsanlagen. Uhl, München (2009)
25. Taylor, C., Bandara, W.: Defining the quality of business process reference models. In: Lethbridge, N. (ed.) Proceedings of 14th Australasian Conference on Information Systems (2003)
26. Weber, K.: Inbetriebnahme verfahrenstechnischer Anlagen. Praxishandbuch mit Checklisten und Beispiele. Springer, Heidelberg (2015)
27. Weck, M., Assmann, S.: Abteilungsübergreifendes Projektieren komplexer Maschinen und Anlagen: Methodik und Systemkonzept zur Verbesserung der Zusammenarbeit. VDI-Z integrierte Produktion **137**(10), 54–60 (1995)
28. Wünsch, G.: Methoden für die virtuelle Inbetriebnahme automatisierter Produktionssysteme. Technical University of Munich, Munich (2008)
29. Zäh, M.F., Wünsch, G., Hensel, T., Lindworsky, A.: Nutzen der virtuellen inbetriebnahme: Ein experiment. Zeitschrift für wirtschaftlichen Fabrikbetrieb **101**(10), 595–599 (2006)

Enterprise Modeling (EMMSAD 2021)

Participatory Modeling from a Stakeholder Perspective: On the Influence of Collaboration and Revisions on Psychological Ownership and Perceived Model Quality

Anne Gutschmidt[1(✉)], Birger Lantow[1], Ben Hellmanzik[1,2], Ben Ramforth[1,2], Matteo Wiese[1,2], and Erko Martins[2]

[1] Computer Science Department, University of Rostock, Rostock, Germany
{anne.gutschmidt,birger.lantow}@uni-rostock.de
[2] Business and Social Psychology, Fachhochschule des Mittelstands - University of Applied Sciences, Schwerin, Germany
martins@fh-mittelstand.de

Abstract. Participatory enterprise modeling is about gathering domain experts and involving them directly in the creation of models, aided by modeling experts. It is meant to increase commitment to and quality of models. This paper presents an exploratory study focusing on the subjective view of the domain experts. We investigated the influence of direct collaboration versus individual modeling, and the influence of model revisions by modeling experts on psychological ownership and perceived model quality. Our results give hint that domain experts working individually with a modeling expert perceive model quality as higher than those working collaboratively whereas psychological ownership did not show any difference. Revisions caused changes in the subjects' assessments only of model quality.

Keywords: Participatory enterprise modeling · Collaboration · Psychological ownership · Perceived model quality

1 Introduction

Enterprise modeling helps companies in systematically capturing knowledge about their organization, processes, responsibilities, product structures, IT systems etc. On the one hand, models may depict a current state, on the other hand, models may be used to visualize plans of a future state [1,2]. Models, such as business process models can be used for simulation and deployment. Nevertheless, in many cases their actual use lies in sense-making, communication and improvements within the organization [3].

Enterprise modeling may be carried out in different ways, especially with regard to knowledge elicitation. Participatory modeling means that the stakeholders are actively involved in the modeling process by gathering them, letting

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 181–195, 2021.
https://doi.org/10.1007/978-3-030-79186-5_12

them discuss and create the models jointly supported by method experts. The latter master modeling method and notation, while the stakeholders play the role of domain experts who concentrate on contributing the relevant content [4].

Participatory modeling is well-suited for companies that are consensus-oriented. Modeling projects may take more time with a participatory approach, but this is compensated by higher model quality and higher commitment to the models [2,4].

In this paper, we concentrate on the subjective perceptions of domain experts towards the model they have created, i.e. feelings of ownership towards the model and perceived model quality. We claim that the stakeholders should feel that the model is *theirs*. Psychological ownership, introduced by [5], has been shown to have a positive effect on affective commitment, the desire to maintain a relationship [6,7]. This is particularly important when a model depicts future plans the stakeholders should later help to implement. Model quality is important when a model is meant to be used for simulation or deployment. However, it is also important to take the stakeholders' perception of model quality into account. In consumer research, it was found that the quality of products and services has a positive influence on satisfaction, trust and loyalty [8,9]. Thus, we additionally claim that domain experts should perceive the quality of the model as good, so that they will be satisfied with and trust the outcome.

We will address two factors that might have an influence on psychological ownership and perceived model quality. First, a participatory setting involves the collaboration of several people, i.e. several domain experts and at least one method expert. A setting where a method expert models together with only one domain expert at a time might lead to different feelings of ownership. Secondly, method experts usually refine models and present them again in a subsequent meeting, either because there was not enough time in the preceding meeting to model everything that was said or because further sources have been consulted in the meantime. So, the original model has been changed by someone else.

To sum up, we examine possible differences in psychological ownership and perceived model quality that might be caused 1) by collaboration between domain experts in a participatory setting, and 2) by revisions made by modeling experts.

In the following section, we will give more explanations on participatory modeling, psychological ownership and model quality, leading to our research questions. In Sect. 3 we will describe our research method. We have conducted an experiment where groups of five persons either worked collaboratively or individually on a process model, always supported by a method expert. After a first meeting, the method expert refined the model and presented it to the participants in a second meeting. After each meeting, psychological ownership and perceived model quality was measured. A mixed analysis of variance was used to explore the data for differences concerning the two dependent variables. The results of the study will be presented in Sect. 4 and discussed in Sect. 5.

2 State of the Art and Theoretical Background

2.1 Participatory and Collaborative Enterprise Modeling

In enterprise modeling, there exist two main roles. On the one hand, the stakeholders, also called domain experts, usually belong to the company that requires the models. They hold the knowledge and ideas that should be captured and visualized. On the other hand, method experts, who may come from outside of the company, have the task to support the domain experts by putting the domain experts' knowledge and ideas into models. The method experts master modeling methods and notations so that the domain experts can concentrate on the content [2,4].

The process of creating the models can be implemented in different ways. The method experts may elicit the necessary information via interviews, by observation and by scanning documents and then, based on this information, they draw the models [2]. When method experts create the models in a team it is called collaborative modeling. The most important goal of collaborative modeling is to create complete models in a fast and accurate way [10]. However, involving stakeholders directly in the model creation process might bring some additional benefits. Luebbe and Weske conducted a study where a domain expert created a process model aided by a method expert. They used a special tangible toolkit that should make modeling easier for beginners [11]. They found that, compared to the traditional way of interviewing the domain expert and having the method expert create the model alone, the domain experts took more time thinking about the process, made more corrections, and had more insights into process modeling.

Participatory modeling goes a step further by involving several stakeholders at a time. It is particularly beneficial when consensus among the stakeholders is important [1]. Sandkuhl and Seigerroth characterize participatory modeling along two dimensions: the number of modelers and stakeholder involvement. With participatory modeling, a group of stakeholders and method experts collaboratively develop the models. The authors refer to all the remaining settings as conventional modeling, including bilateral modeling with one method expert and one domain expert as described above. Among the method experts, further role distributions may be applied. A facilitator who leads the discussion is always present. Optionally, there is a tool operator who handles the modeling tool and a secretary who documents the modeling process [2,4]. The advantage of a participatory setting is that there is an exchange of knowledge and ideas among the stakeholders. Furthermore, conflicting perspectives that may not have been discovered or resolved in individual consultations become explicit in participatory sessions [12]. As a consequence of this, participatory modeling is claimed to lead to greater commitment to the models, including the plans and ideas that may be connected to them. Additionally it is said to lead to greater model quality as information elicitation is more exhaustive [1,4]. However, empirical evidence is based on case studies whereas we want to investigate these causal relationships in an experiment.

2.2 Psychological Ownership

Psychological Ownership means that people feel they own something, e.g. an object, a person or an idea, although they may not legally possess the target of interest [5]. In organizational context, it was shown to lead to certain positive effects such as citizenship behavior and extra role behavior [7,13] and affective commitment [6,7]. The latter one means, that one feels the desire to maintain the relationship towards e.g. a company [14]. Pierce and colleagues name three so-called routes that lead to psychological ownership. One must control the target, have intimate knowledge about the target, and invest the self into the target in terms labor, time and other personal resources [5,15].

In this paper, we will compare participatory modeling (a team of domain experts models collaboratively, supported by a method expert) with individual modeling (where the method expert creates the model together with single stakeholders). In the participatory setting, the stakeholders follow the essential parts of the creation process. They are present when the others make their suggestions. In the second setting, everyone generates their individual model and they are later confronted with others' input which may be supporting, adding or contradicting to their preceding input. Yet, in the participatory setting, some people might not be able or willing to give as much input as in individual meetings. With regard to getting to know the target thoroughly, participants of a participatory modeling session may get new insights into the subject through their teammates. However, this is also possible when being confronted with a model built from several individual models that possibly contain different pieces of knowledge although there is no synchronous exchange. Thus, it is for us to explore whether these two different settings cause any difference in psychological ownership. As it is very difficult to hypothesize about the exact influence of collaboration on feelings of ownership, we formulate an open research question:

RQ1: Do these different settings (participatory/collaborative vs. conventional/individual) cause a different level of ownership feelings towards the model?

Another influence that has to be considered is the one of revisions by the modeling expert. The fact, that one may not have been present when further parts of the model have been created might induce a feeling of having less control over the model. Either additions are made from information the participants gave to the method expert, but which could not be modeled, e.g. due to a lack of time. Or, in case of individual models, parts of the refined model might even come from other domain experts. That is why we investigate the influence of revisions on psychological ownership. So the second research question is:

RQ2: Is the feeling of ownership towards the model different when the first model has been created and when being confronted with a refined model in a second meeting?

In addition, we will look for a possible interaction between the two factors revision and collaboration.

2.3 Model Quality as Subjective Target

The quality of models is crucial. Only when a model meets certain standards, and especially the expectations of the stakeholders, it will be of use. Many approaches that try to define model quality are based on a division into quality aspects inspired by semiotics theory [16]. Moody and colleagues [17,18] refer to that same theory when they use the terms syntactic quality, semantic quality and pragmatic quality. While the first one represents syntactic correctness with regard to a modeling language and its rules, semantic quality means that a model reflects reality correctly and as completely as necessary. Pragmatic quality is about the stakeholders' understanding the model [18]. Krogstie [3] presented an extended system of model quality by presenting the SEQUEL framework, adding empirical quality (comprehensibility of the model), physical quality (availability of models to the relevant actors), perceived semantic quality (correctness and completeness of the model from a stakeholder's subjective point of view), social quality (stakeholders' agreement on model interpretation) and deontic quality (meeting the goals of modeling). Krogstie and colleagues [19,20] also scrutinized these criteria to create guidelines for modeling processes.

We agree that researchers and modeling experts should strive for guidelines that tell us how to create models of high quality and that help us assess a model's quality. However, while the above-mentioned approaches take mostly a modeling expert's perspective – which is absolutely important and useful, we want to regard quality from the stakeholder's perspective. We claim that besides feeling that "the model is mine" it is also important that, after a model has been created, domain experts can say that "this model is of good quality". We assume that what domain experts will probably value the most is that the model contains all relevant information and is correct, that they think they and others can understand the model and that they have trust in the model's syntactic correctness.

A study by Heggset and colleagues [21] gave hint that there is a connection between syntactic quality and model comprehension. After models had been refined with regard to syntactic quality, the subjects would give more correct answers to questions about the models' content. In a study about text revisions, Caspi and Blau [22] found that after others had edited the subjects' texts, perceived quality increased whereas perceived ownership decreased. So, possibly, domain experts might value revisions of modeling experts and perceive them as an increase in model quality. The modeling expert might not only improve syntactic quality but also add content that was named by the domain experts but had not yet been included in the model. Yet, revisions might not necessarily be perceived as positive. When several single models must be merged this will mean that domain experts will in some cases be confronted with new content which might furthermore contradict to their views. Thus, the research question is:

RQ3: Is there a difference in how domain experts perceive a model's quality before and after it has been refined by a modeling expert?

In a modeling session, where a modeling expert works with only one domain expert, the domain expert has a chance to voice all thoughts and ideas without being interrupted or distracted by other persons. In fact, although team work as in participatory modeling has its benefits, e.g. knowledge of different sources is gathered, it has its drawbacks. Team members might be more reluctant because someone more dominant or with a higher position in the organizational hierarchy might be present [23], or they may be less motivated, e.g. because they do not see their individual contribution in the overall outcome [24]. Moreover, collaboration requires more coordination concerning communication resulting in productivity blocking [25]. So, on the one hand, domain experts might be more satisfied with the model when they could work intensely with the modeling expert in an individual session than when having to assert themselves in a group of people. On the other hand, the variety of suggestions of different people might generate the impression of a better model. So, the last research question is:

RQ4: Is there a difference between conventional/individual modeling and participatory/collaborative modeling with regard to perceived model quality?

We will also investigate a possible interaction between the two factors revision and collaboration concerning perceived model qualities as their different combinations might lead to different outcomes.

3 Research Method

3.1 Experimental Design

The first independent variable we considered is represented by collaboration. Either a group of five persons created a joint process model collaboratively, supported by a modeling expert, or separate models were created by five single persons with a modeling expert in individual sessions, and later merged by the method expert. The first setting represents participatory modeling whereas the second setting is to be regarded as a form of conventional modeling according to [1].

All participants attended two modeling sessions such that the second independent variable is represented by time of measurement using two points in time. Measurements were taken after each of the two meetings. In the first meeting, an initial model was created. In the second one, a model revised by a modeling expert was presented to the participants and corrections and additions could be suggested. The experimental design can also be seen in Fig. 1.

In the first meeting, we set the task to create a model that would describe the process of registering a final paper at the university, such as a bachelor thesis. After a short introduction to the Business Process Modeling Notation (BPMN) with a brief example, the participants had thirty minutes time for modeling the process of interest using BPMN. Due to the Covid 19 pandemic, we did not meet the participants in person but via video conference (Zoom). The modeling was done with the software Cawemo which runs within the browser and can be used collaboratively. Cawemo is a tool for process modeling with BPMN.

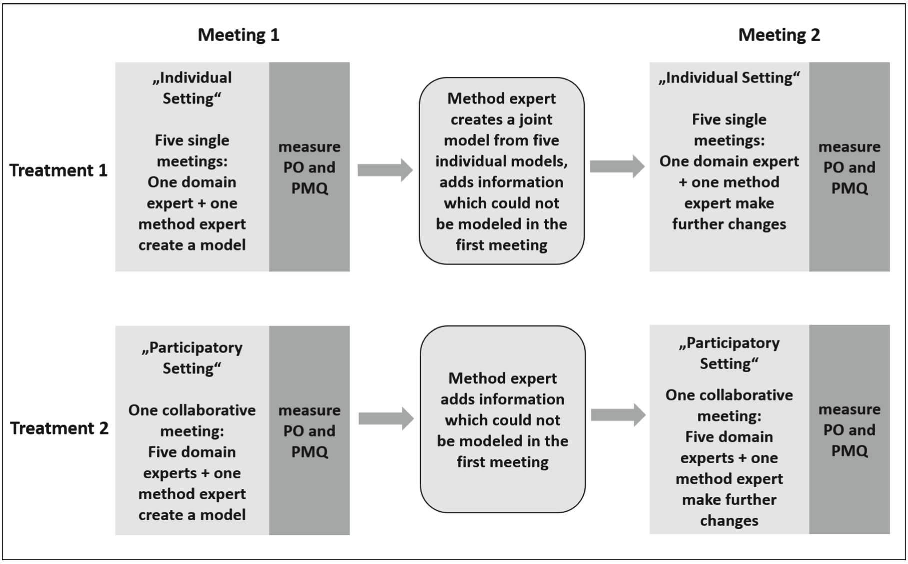


Fig. 1. Experimental design with the treatments individual vs. participatory setting, and the two meeting where each time psychological ownership (PO) and perceived model quality (PMQ) were measured.

For each trial, we gathered a team of five persons: two or three students (one participant in the overall sample had recently graduated, and started working), one employee familiar with the administrative part of the process, and one or two employees who supervise students in writing papers. They should represent the important stakeholder groups in the process. Addressing the independent variable of collaboration, we either gathered the whole group to create the model together, aided by a modeling expert, or the modeling expert met each member of the group individually to create five separate models. After the modeling, we asked the participants to fill out a questionnaire, assessing demographic data, modeling experience and the dependent variables psychological ownership and perceived model quality.

In the second meeting, a few days later, a revised model was presented to the participants. In case of the collaboration treatment, modeling experts would have added information that was named in the first meeting, but could not be modeled then due to a lack of time. In case of the individual modeling treatment, the modeling expert had to merge five models and presented the one resulting model again individually to the participants. The participants had 15 min time to suggest changes and additions. Afterwards, psychological ownership and perceived model quality were again assessed using a questionnaire.

3.2 Measuring the Dependent Variables

We measured psychological ownership as dependent variable using a mix of the scales by [26] (German version by [27]) and [22], adapted to the context of modeling. We used seven German items which had to be rated on a five-point Likert scale. Here, we present the English items: 1) This is MY model, 2) I sense that this model is OUR model, 3) I sense that this is MY model, 4) This is OUR model, 5) Most of the people that have worked on this model feel as though they own the model, 6) It is hard for me to think about this model as MINE (reversed), and 7) I feel that the model is mine, even if others contributed to its development.

For the second dependent variable, we were searching for a scale which allowed accessing the subjective perception of model quality. We chose a scale introduced by [22] which originally referred to self-written text, but appeared to be fitting to the modeling context nonetheless. The authors reported very good reliability with a Chronbach's alpha of 0.93 and 0.94, respectively, for two measurements. The scale comprises six items which we translated and adapted to the context of model quality. Furthermore, we used reversed formulations for two items to prevent a dominating tendency towards one side as suggested by [28]. The items ask for the model being 1) not of good quality (reverse), 2) comprehensive, 3) exhaustive, 4) well-drawn, 5) exact, and for 6) not presenting information clearly (reverse).

Both scales were included in the questionnaire the participants had to fill out after each meeting.

3.3 Evaluation Methods

For a mixed design like ours, comprising a combination of a between-subjects design (collaboration vs. individuals) and a within-subjects design (two measurements), comparisons of mean values with a mixed analysis of variance (ANOVA) is preferred. Beside the main effects of collaboration and point in time, the method allows the consideration of the interaction between the two factors. Its main preconditions include homogeneity of variance checked with the Levene test, and balanced treatment groups. Normal distribution of residuals is less important as long as the two former conditions are fulfilled [29].

We used factor analysis and reliability analysis in SPSS to check the scales' validity and reliability. Based on these results, we calculated scores for the corresponding constructs of psychological ownership and perceived model quality. The scores were used as input variables for the mixed ANOVA. For all significance tests, we considered an alpha of 0.05.

3.4 Sample

We recruited participants from three universities with different courses of study: business information systems, computer science, specialized areas of business administration, and psychology. On the whole, four groups of five persons each, 20 persons overall, took part in the study, among them 11 men and 9 women. The average age was 33.6 years ($\sigma = 11.7, min = 20, max = 60$). Based on a five-point Likert scale with 1 as minimum, experience with BPMN modeling was rated on average with 2.6 ($\sigma = 1.3$) and 1.6 ($\sigma = 0.5$) for the collaboration treatment groups, and 3.2 ($\sigma = 1.3$) and 1.0 ($\sigma = 0$) for the individual treatment groups. When the collaboration teams met in the first session, some of them knew each other already, some had met for the first time. The participants meeting the modeling expert individually, knew they were part of a group and that their models would be merged to one overall model.

4 Results

4.1 Factor and Reliability Analyses

Although the scale measuring psychological ownership (PO) is assumed to assess a one-dimensional construct, we already found in a prior study that there was a division between individual and collective psychological ownership [30]. In this study, the participants seemed to feel this distinction, too, because it clearly shows in their rating of the PO items containing MY and OUR. The exploratory factor analysis in SPSS (principal components, Varimax rotation) showed two factors where the items "I sense that this model is OUR model" and "This is OUR model" load on one factor and the remaining items on a second factor. All factor loadings were greater than 0.6 for both measurements. The reliability analyses showed satisfying values for Cronbach's alpha for all subscales and all measurements (first and second meeting). For the OUR scale, Cronbach's alpha was .872 and .912. For the MY scale, Cronbach's alpha was .889 and .925. As a consequence, we calculated two scores, one for individual PO and one for collective PO.

Caspi and Blau's quality measure is originally one-dimensional [22]. Two factor analyses for the two measurement data sets showed, however, a division of items into two factors. One item (The model is not of good quality) was separated addressing general quality whereas the other items refer to specific characteristics. The item "The model does not present information clearly" loaded on factor one in the first measurement and on factor two in the second measurement, apparently being ambiguous. Cronbach's alpha increases 1) from .777 to .853, and 2) from .709 to .759 when leaving out the item of general quality. It increases even more when removing the above-mentioned ambiguous item, to .885 and .851, respectively. We decided to use two variables: a variable consisting of only the item measuring general quality and a score calculated from the items addressing the characteristics comprehensive, exhaustive, well-drawn, and exact.

Table 1 shows the descriptive statistics of the resulting variables, separated for the two factors collaboration and point in time of measurement.

Table 1. Descriptive values of psychological ownership (PO) and of perceived Model Quality (PMQ), separated by collaboration (yes/no) and the two measurements M1 and M2.

	Collaboration	Individual PO		Collective PO		General PMQ		Specific PMQ	
		μ	σ	μ	σ	μ	σ	μ	σ
M1	Yes	2.2	1.2	4.5	0.8	3.7	1.1	3.3	0.9
	No	3.1	1.1	3.9	1.0	3.9	1.4	4.2	0.8
	Overall	2.6	1.2	4.2	1.0	3.8	1.2	3.7	0.9
M2	Yes	2.1	1.0	4.1	1.1	4.3	0.7	3.7	0.6
	No	2.9	1.3	4.0	0.8	4.1	1.4	4.7	0.4
	Overall	2.5	1.2	4.0	0.9	4.2	1.1	4.2	0.7

4.2 Analysis Regarding Psychological Ownership

According to the Shapiro Wilk test, not all variables showed normal distributions. The ANOVA is, however, said to be robust against the violation of this condition if the data show homogeneous variances and are balanced [29], which was the case for our data set.

For the individual PO, the mixed ANOVA resulted in non-significant effects for time of measurement (F = 0.4, df = 1, p = 0.5), for the interaction of the two factors (F = 0.1, df = 1, p = 0.7), and for the collaboration treatment (F = 3.4, df = 1, p = 0.08).

For the collective PO, the mixed ANOVA again showed only non-significant effects, i.e. for time of measurement (F = 0.6, df = 1, p = 0.4), for the interaction of the two factors (F = 1, df = 1, p = 0.3), and for the collaboration treatment (F = 0.8, df = 1, p = 0.4).

Thus, neither the different measurements nor collaboration nor the combination of the two factors showed a difference in PO.

4.3 Analysis Regarding Perceived Model Quality

As with PO, normal distribution was not always fulfilled for the perceived model quality variables, but the Levene test showed homogeneity of variance and the data set was balanced.

For the variable addressing general perceived model quality, there was a significant main effect for time of measurement (F = 7.2, df = 1, p = 0.015), but not for collaboration (F = 0, df = 1, p = 1) and not for the interaction of the factors (F = 1.8, df = 1, p = 0.2).

For the second variable summarizing specific perceived quality attributes, there was a significant main effect for time of measurement (F = 6.8, df = 1, p = 0.018), and for collaboration (F = 12.2, df = 1, p = 0.003), but not for the interaction of the factors (F = 0.2, df = 1, p = 0.7).

To sum up, there was a difference between the two measurements of both perceived general quality and the variable measuring specific perceived aspects of model quality. The latter one also showed a difference caused by collaboration vs. working individually.

5 Discussion

5.1 Summary and Interpretation

We have conducted a study on enterprise modeling that focused on the subjective view of the domain experts. We were interested in how they perceive quality of and ownership feelings towards a model they helped creating depending on different settings. We compared a participatory setting where domain experts modeled collaboratively supported by a method expert with a setting where method experts worked with domain experts in individual meetings. Furthermore, we examined how the domain experts' assessments changed after the model had been revised by method experts.

With regard to psychological ownership, we have not found any differences caused by collaboration between the domain experts (RQ1) nor by revisions made by modeling experts (RQ2). Considering the descriptive statistics, it becomes clear that the participants make a distinction between individual and collective psychological ownership. Apparently the feeling that the model is common property is more predominant. In fact, collective psychological ownership shows a high average value comparing it to the greatest possible value of 5. Unfortunately, we do not know to whom the participants were referring when they said the model is 'OUR' model. Were they referring to the other participants or to the modeling expert? In future studies this must be assessed in particular. The different actors might have their own influence on the participants' perceptions.

All participants spent the same amount of time to create the model. The more exclusive and intense exchange with the modeling expert in the individual setting did not seem to have had an influence on psychological ownership.

In the participatory setting where the participants worked as a team, it happened quickly that team members took on the task of drawing the model whereas in the individual setting, most of the participants concentrated just on the content and let the modeling expert draw the model. A former study showed that if modeling experts take on the task of drawing this will not undermine the emergence of psychological ownership [30] which seems to be the same in our study. So, letting others model will probably not decrease a feeling of being in control.

Caspi and Blau had shown that editing of one's text by other people decreased psychological ownership [22]. In our study, there is just a slight and non-significant decrease in three of the four groups with regard to psychological ownership. This effect might have been stronger if the issue to be modeled had involved different opinions and more profound differences in the knowledge of the participants. We seem to have chosen a process which was very clear to the participants. There were only few small parts that some participants had not been aware of before.

With respect to perceived model quality, we found the measuring instrument we used to split in two dimensions. We found one dimension to capture general quality and a second one to comprise the characteristics comprehensive, exhaustive, well-drawn, and exact. The question is what the participants' understanding of general quality includes and if there are further characteristics of a model that are important to domain experts. We will reconsider existing approaches, e.g. [19], as a basis.

Both quality variables showed a significant difference with regard to revisions by the modeling experts. Perceived quality tended to be higher in the second meeting, i.e. after the method expert's revision (RQ3). What the participants might have noticed is that the models were extended, syntactically revised and reformatted. It seems that this is perceived as an improvement. This is in accordance with previous findings by [22]. From this, we conclude that the participants value the support they get from the modeling expert as very high. With regard to the quality variable comprising specific characteristics, the intense exchange with the modeling expert in the individual setting seems to make a difference; i.e. the participants rated the quality higher than those participants in the participatory setting (RQ4). In the first setting, the participants could describe the process and the modeling experts would translate everything into model elements step by step. So, there is probably an additional value of bilateral modeling in more thorough explanation of what is modeled and how.

Based on our results, we draw two major conclusions. First, to support the emergence of psychological ownership, it is not necessary to organize modeling sessions with single stakeholders. Participatory settings seem to be equally suited to trigger feelings of collective ownership. If a model should depict a future state or goals the stakeholders have to agree on, single bilateral meetings would probably be even counterproductive. Second, modeling experts are of great importance to the domain experts. As stated earlier by other authors [2,4], it is advisable to have a facilitator and a tool operator support the domain experts. The domain experts trust in their modeling expertise which will probably increase the domain experts' confidence and trust in the outcome.

5.2 Limitations

The sample we considered in this study was small and can only give first insights. However, it has to be noted, that the implementation of this study was connected with a great effort in time both for the participants and for the researchers. The sample comprised persons who had never created a model before as well as some persons with modeling experience. A greater sample with more inexperienced participants would increase generalizability.

Our outcomes, especially concerning psychological ownership, have certainly been influenced by the task we set. With a task where more discussion and new ideas would be required, the participants, especially in the individual setting, might have assessed the situation and the model differently. Participants would probably have been confronted with more new model parts that might even be in conflict to their knowledge or opinions.

From the experiment, we experienced that in participatory modeling, more time must be given because discussion takes more time. This is a known characteristic of participatory modeling [1]. Nevertheless, our intention was to keep the conditions in the treatment groups as similar as possible. That is why we set the same time limit for all groups. Future studies should reconsider how to make the conditions more comparable.

Furthermore, we see a need to determine more strictly what the modeling expert should do. In the study, the participants were free to choose whether they wanted the modeling expert to draw or whether to draw themselves. To generate similar conditions, one common procedure should be chosen.

Finally, due to the Covid 19 pandemic we could not meet the participants in person. We cannot rule out that this could have had a general influence on the communication between all actors.

5.3 Implications for Future Research

Despite a small sample and some limitations in the experimental setting, the study presented in this paper offers an informative insight into how domain experts perceive a model under certain conditions. We were especially interested in the impact of a participatory setting on the assessment of model quality and feelings of ownership. Concerning ownership feeling, it seems that it is not necessary for method experts to sit down and work on a model with every domain expert in single meetings. The latter might even lead to more complications. How should the modeling expert proceed when some domain experts give conflicting information? One might use an incremental proceeding where every domain expert adds further information to the same model. But who decides on the right sequence of domain experts to meet when e.g. a given model might influence or restrict participants in their thinking? Thus, we claim a participatory setting as preferable since it also promotes collective psychological ownership.

Moreover, the study has underlined the importance of method experts. Their contributions seem to lead to more trust in the model's quality. Our results give hint that the method experts should take on the task of modeling and make sure that the participants can follow the model creation process.

Future work should involve the development of a scale for measuring model quality from the point of view of a domain expert. Concerning psychological ownership, it would be interesting to explore whether method experts are also considered as co-owners and if so what the consequences might be.

Finally, it must be noted that this study considered just the point of view of the domain experts. Based on the data we have already collected, we plan to compare the objective quality of a model with the perceived model quality.

References

1. Sandkuhl, K., Seigerroth, U.: Participative or conventional enterprise modelling? Multiple-case analysis on decision criteria. In: Rowe, F., et al. (eds.) 28th European Conference on Information Systems - Liberty, Equality, and Fraternity in a Digitizing World, ECIS 2020, Marrakech, Morocco, 15–17 June 2020 (2020)

2. Sandkuhl, K., Stirna, J., Persson, A., Wißotzki, M.: Enterprise Modeling. TEES. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43725-4
3. Krogstie, J.: Model-Based Development and Evolution of Information Systems: A Quality Approach. Springer, London (2012). https://doi.org/10.1007/978-1-4471-2936-3
4. Stirna, J., Persson, A.: Enterprise Modeling - Facilitating the Process and the People. Springer (2018). https://doi.org/10.1007/978-3-319-94857-7
5. Pierce, J.L., Kostova, T., Dirks, K.T.: The state of psychological ownership: integrating and extending a century of research. Rev. General Psychol. **7**(1), 84–107 (2003)
6. Mayhew, M.G., Ashkanasy, N.M., Bramble, T., Gardner, J.: A study of the antecedents and consequences of psychological ownership in organizational settings. J. Soc. Psychol. **147**(5), 477–500 (2007)
7. O'Driscoll, M.P., Pierce, J.L., Coghlan, A.M.: The psychology of ownership: work environment structure, organizational commitment, and citizenship behaviors. Group Organ. Manage. **31**(3), 388–416 (2006)
8. Lie, D., Sudirman, A., Butarbutar, M.: Analysis of mediation effect of consumer satisfaction on the effect of service quality, price and consumer trust on consumer loyalty. Int. J. Sci. Technol. Res. **8**(8), 412–428 (2019)
9. Anderson, E.W., Sullivan, M.W.: The antecedents and consequences of customer satisfaction for firms. Market. Sci. **12**(2), 125–143 (1993)
10. Barjis, J.: CPI modeling: collaborative, participative, interactive modeling. In: Proceedings of the 2011 Winter Simulation Conference (WSC), pp. 3094–3103 (2011)
11. Luebbe, A., Weske, M.: Tangible media in process modeling – a controlled experiment. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 283–298. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21640-4_22
12. Stirna, J., Persson, A., Sandkuhl, K.: Participative enterprise modeling: experiences and recommendations. In: Krogstie, J., Opdahl, A., Sindre, G. (eds.) CAiSE 2007. LNCS, vol. 4495, pp. 546–560. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72988-4_38
13. Avey, J.B., Avolio, B.J., Crossley, C.D., Luthans, F.: Psychological ownership: theoretical extensions, measurement and relation to work outcomes. J. Organ. Behav. Int. J. Ind. Occup. Organ. Psychol. Behav. **30**(2), 173–191 (2009)
14. Meyer, J.P., Herscovitch, L.: Commitment in the workplace: toward a general model. Hum. Resour. Manage. Rev. **11**(3), 299–326 (2001)
15. Pierce, J.L., Jussila, I.: Psychological Ownership and the Organizational Context: Theory, Research Evidence, and Application. Edward Elgar Publishing (2011)
16. Morris, C.W.: Foundations of the theory of signs. In: International Encyclopedia of Unified Science, pp. 1–59. Chicago University Press (1938)
17. Moody, D.L., Sindre, G., Brasethvik, T., Solvberg, A.: Evaluating the quality of information models: empirical testing of a conceptual model quality framework. In: 25th International Conference on Software Engineering, 2003. Proceedings, pp. 295–305 (2003)
18. Moody, D.L.: Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. Data Knowl. Eng. **55**(3), 243–276 (2005). Quality in conceptual modeling
19. Krogstie, J.: Quality of business process models. Quality in Business Process Modeling, pp. 53–102. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42512-2_2

20. Heggset, M., Krogstie, J., Wesenberg, H.: Understanding model quality concerns when using process models in an industrial company. In: Gaaloul, K., Schmidt, R., Nurcan, S., Guerreiro, S., Ma, Q. (eds.) CAISE 2015. LNBIP, vol. 214, pp. 395–409. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19237-6_25
21. Heggset, M., Krogstie, J., Wesenberg, H.: The influence of syntactic quality of enterprise process models on model comprehension. In: CAiSE Forum, pp. 89–96 (2015)
22. Caspi, A., Blau, I.: Collaboration and psychological ownership: how does the tension between the two influence perceived learning? Soc. Psychol. Educ. **14**(2), 283–298 (2011)
23. Torrance, E.P.: The behavior of small groups under the stress conditions of "survival". Am. Sociol. Rev. **19**(6), 751–755 (1954)
24. Karau, S.J., Williams, K.D.: Social loafing: a meta-analytic review and theoretical integration. J. Pers. Soc. Psychol. **65**(4), 681 (1993)
25. Diehl, M., Stroebe, W.: Productivity loss in brainstorming groups: toward the solution of a riddle. J. Pers. Soc. Psychol. **53**(3), 497 (1987)
26. Van Dyne, L., Pierce, J.L.: Psychological ownership and feelings of possession: three field studies predicting employee attitudes and organizational citizenship behavior. J. Organ. Behav. Int. J. Ind. Occup. Organ. Psychol. Behav. **25**(4), 439–459 (2004)
27. Martins, E.: Psychological Ownership in Organisationen: Explorative Untersuchungder Antezedenzen und des Entstehungsprozesses.Rainer Hampp Verlag (2010)
28. Bühner, M.: Einführung in die Test-und Fragebogenkonstruktion. 2nd edn. ps. Pearson Studium, München [u.a.] (2006)
29. Bühner, M., Ziegler, M.: Statistik für Psychologen und Sozialwissenschaftler. Pearson Studium - Psychologie, Pearson Studium (2009)
30. Gutschmidt, A., Sauer, V., Schönwälder, M., Szilagyi, T.: Researching participatory modeling sessions: an experimental study on the influence of evaluation potential and the opportunity to draw oneself. In: Pańkowska, M., Sandkuhl, K. (eds.) BIR 2019. LNBIP, vol. 365, pp. 44–58. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-31143-8_4

Reference Service Model for Federated Identity Management

Daniela Pöhn(✉) and Peter Hillmann(✉)

Universität der Bundeswehr München, 85577 Neubiberg, Germany
{daniela.poehn,peter.hillmann}@unibw.de

Abstract. With the pandemic of COVID-19, people around the world increasingly work from home. Each natural person typically has several digital identities with different associated information. During the last years, various identity and access management approaches have gained attraction, helping for example to access other organizations' services within trust boundaries. The resulting heterogeneity creates a high complexity to differentiate between these approaches and scenarios as participating entity; combining them is even harder. Last but not least, various actors have a different understanding or perspective of the terms, like "service", in this context. Our paper describes a reference service with standard components in generic federated identity management. This is utilized with modern Enterprise Architecture using the framework ArchiMate. The proposed universal federated identity management service model (FIMSM) is applied to describe various federated identity management scenarios in a generic service-oriented way. The presented reference design is approved in multiple aspects and is easily applicable in numerous scenarios.

Keywords: Modelling · Services · Security · Identity management · Federated identity management

1 Introduction

Due to the digitization, people are using an increasing number of identities in various fields. Every person has in average about 130 online accounts [11] for, e.g., work, e-government, online shops, and social networks. Each digital identity contains different user information, called attributes. Providers like Google allow users to securely reuse their accounts to authenticate at other services by utilizing the protocols Open Authorization (OAuth)/OpenID Connect (OIDC). The alternative Federated Identity Management (FIM) protocol Security Assertion Markup Language (SAML) is typically used in research and education. FIM has the advantage over traditional centralized identity and access management (I&AM) that the attributes are stored authoritatively at the home organization, i.e., Identity Provider (IDP). Thereby, the data is consistent and the user has only one digital identity to remember. The Service Provider (SP) asks the IDP

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 196–211, 2021.
https://doi.org/10.1007/978-3-030-79186-5_13

about the authentication respectively authorization. As a consequence, the SP securely reuses the information. If not all entities use the same protocol or a variant of it, additional effort is required. By shifting the focus from applications to the user, user-centric approaches were introduced, adding even more diversity to the landscape. According to Dhamija and Dusseault [6], users follow the path of least resistance, while cognitive scalability is as important as technical scalability. With the different ecosystems in mind, these goals seem difficult to reach. The diversity of identities is growing owing to several factors, like smartphones and Internet of Things (IoT) devices. The multitude of standards and solutions hardens the task of companies due to the lack of interoperability between them. This makes it increasingly difficult to get an overview of these complex structures. An outline and an understanding of the provided services is though needed to improve the current identity management systems. Therefore, a systematic approach is required.

To address these problems, a reference architecture for federated identity management service model (FIMSM) is designed with the supporting toolkit of Enterprise Architecture applying the framework ArchiMate as it is the most used modelling approach [4]. ArchiMate is an open enterprise architecture modelling language highlighting mainly the design and structure of a system. Furthermore, it supports the description, analysis, and visualization of architectures in a generic way. A reference architecture consists of reusable models and patterns, which are adaptable and customizable. It describes a particular recurring design problem that arises in specific contexts and presents a well-proven solution for the problem [4]. FIMSM offers overviews from different hierarchy levels and several detailed perspectives. The main advantage of a representation in the form of a model is that it is more precise and easier to understand than a textual description. In addition, the transfer is supported with model-driven architectures. Our reference architecture is conform to Information Technology Infrastructure Library (ITIL), The Open Group Architecture Framework (TOGAF) and Federated Mission Network as well as further frameworks. Organizations can refer to this template when adapting or constantly improving FIM.

This paper contributes the following improvements to the modelling and identity management landscape: With FIMSM, a reference architecture focusing on federated identity management is designed, which can be reused. With the model and patterns, the architecture describes elements, relations, functionality, and data flows required for FIM. It is the first concept for a universal identity management reference architecture covering all identity management variants. This overview is an important step towards this goal. With the reference model as background, it forms the basis to improve and enrich identity management in future work.

The paper is structured as follows: After the introduction, we discuss a motivating scenario with three entities and different protocols, which identifies several key components in Sect. 2. Related approaches are shown in Sect. 3. Based on the key components and findings, we present the FIMSM in Sect. 4. The discussion in Sect. 5 is three-folded with scenarios, requirements, and an model adaption. Section 6 serves as conclusions and outlook.

2 Motivating Scenario

The problem description arises from three entities, which want to cooperate: 1) IDP of a university, 2) IDP and SP of a commercial company, and 3) IDP and SP of a university hospital. All three entities use different identity management systems, which may result in several accounts for the end users.

1) University

Local I&AM: In order to provide several services, like email, file transfer, and video conference systems, to students, employees, and professors alike, universities have a local I&AM in place. This I&AM is typically based on directory services, e.g., lightweight directory access protocol (LDAP). For Single Sign-On (SSO), another service is running, tracking the different sessions.

FIM: As a result of cooperation, national and international research and education federations, like DFN-AAI and SWITCHaai, were established. These provide services to each other within certain trust boundaries. The university is part of one of these national federations. The participating IDPs and SPs communicate with a pre-decided protocol, i.e., SAML. In order to make use of the federation, the SAML implementation Shibboleth is applied. The federation operators provide additional services, e.g., distribution of aggregated SAML metadata, via a trusted third party (TTP). The aggregated SAML metadata consists of information about the members of the federation enabling cooperation. Collaborations outside of the federation are difficult though. Some services already accept different protocols; otherwise proxies may be needed.

2) Commercial company

Local I&AM: The company uses Active Directory (AD) as local I&AM.

FIM: In order to access different web services including a private cloud, the component Federation Services (AD FS) is deployed. Suppliers and project partners have access to certain shared folders within the federated private cloud via some kind of OIDC. For specific services, a second factor is required. As both partners do not use the same protocol, a translation service is needed. If users of the company want to access external services, a proxy has to be added, since AD FS only works with one SP.

Others: Other systems, like Pluggable Authentication Modules (PAM), are in control of access to servers. The company has factories, optimizing the manufacturing by adding IoT sensors. The I&AM does not only include identification and management of access to different types of devices, and, thereby, data. It also addresses interactions between devices, services offered, and how to identify the device as well as control the access to sensitive data. Since IoT devices often are installed, where everyone can tamper with the device, they should not have access to the whole network. This puts different constraints on the adoption of existing I&AM frameworks. IoT management is so far another identity management silo for the company. Therefore, interoperability is needed.

3) University hospital

Local I&AM: The hospital is associated with the university. At the same time, it runs different research labs, where external researchers can access medical devices from remote via a virtual private network. These medical devices still run on older operating systems, as updates need to undergo a certification process. Since the hospital only provides a local I&AM, external researchers receive local accounts. This local identity management system has implications for employees, which own another account from the university. Therefore, many end user hold several usernames and credentials. With all the different accounts, the end user want to get a better overview.

Based on this scenario, we obtain the following requirements, which a reference architecture should fulfil.

R1: Reusable architecture for I&AM with generic and universal terminology.
R2: Systematic overview and detailed perspectives on single aspects.
R3: Adaptability to different protocols and use cases.
R4: Dependencies between various providers with related interfaces including requirements regarding an appropriate service management.

We take these requirements into account when designing a reference architecture for FIM. Furthermore, the following research questions will be addressed.

Q1: How to describe I&AM scenarios with a scenario-independent approach?
Q2: Which elements are required to fulfil the requirements?
Q3: Is it possible to adapt the reference architecture to different areas?

3 Related Approaches

In this section, we give an overview of related approaches for I&AM and models of selected processes and components within. First, we outline related work in the area of I&AM (A). Then, we discuss modelling approaches for trust (B), followed by modelling approaches for identity management (C).

A) According to Yuan Cao and Lin Yang [13], three different identity management models exist: isolated, centralized, and federated identity management. The authors further classify I&AM into the paradigms network-centric, service-centric, and user-centric, which evolved into the models isolated, centralized, federated, and user-centric. Isolated identity management, i.e., local accounts, is typically no longer used, although exceptions exist. Centralized I&AM is the base for FIM, which mainly comprises of the protocols SAML, OAuth, and OIDC. SAML federations are rather static due to the federation structure and metadata exchange via a TTP. OAuth and OIDC are more dynamic, but not less complex. OIDC is the authentication layer on top of OAuth. Both protocols, OAuth and OIDC, are further improved within OpenID Foundation and Internet Engineering Task Force. Another model relates to user-centric approaches. While different privacy enhancing technologies never established long living approaches in I&AM, two main directions currently exist. User Managed Access decouples identity resolution from the maintenance of identity information. It is built upon OAuth, although the principle can be applied to other protocols as well. User Managed Access is currently extended for the usage with self-sovereign identity. The principles of self-sovereign identities are stated by Toth and Anderson-Priddy [20]. They are typically implemented by blockchain, a form of decentralized identifiers. DecentID by Friebe et al. [7], e.g., uses a public blockchain as trust anchor and utilizes external storage, i.e., distributed hash tables, for attributes due to scalability. The stated identity management models are adjusted for cloud identity management by Zwattendorfer et al. [22]. An overview of numerous research approaches within the identity management models is provided by Pöhn and Hommel [16]. These approaches are compared with derived requirements, showing that no flexible and universal solution exist.

B) The need for a holistic architecture view is mandatory to avoid duplication in realization and to obtain synergy effects between the different systems [1]. Especially the collaboration of several institutions in a federation becomes a challenge due to the growing IT complexity and architecture as a whole. The diversity of over 60 frameworks [12] offers a wide range of modelling description possibilities. Organizations are required to present the information systems underlying their business in a transparent manner. The establishment of Enterprise Architecture is a mandatory discipline for companies to overcome this challenge. One of the newer modelling languages is ArchiMate [19], published in the year 2004. The main advantage is the simplicity. Therefore, it fits to our intention to provide a detailed overview and the interfaces for a federated system. Additionally, a transformation to other modelling approaches is possible [9].

C) The stated identity models are described in a high-level or use case respectively implementation specific architectures, but no generic model for identity management is known to the authors. The developers of ArchiMate themselves offer only a rudimentary approach to identity management [18]. This does not explore the technical possibilities. Although Dabrowski and Pacyna [5] claim a reference architecture for identity management, no official meta-model is used. Pöhn and Hommel [15] describe FIM and Inter-FIM with an older meta-model, not taking different protocols and elements into account. The approaches of Yang et al. [21] and Katsikogiannis et al. [10] focus on the process of authentication and authorization. However, they do not take a holistic view from an architecture enterprise perspective. Elements for reference architectures are described in frameworks and standards, like ISO/IEC 24760. These are neither universal nor up-to-date. A closely related approach to the problem is presented in form of an Enterprise Architecture Pattern by Perroud and Inversini [14]. Their reference design of the pattern LetMeAccess provides different views on the centralized I&AM architecture, following no official meta-model. Amaral et al. [2] model the aspect of trust in general with ArchiMate, providing guidelines for this approach. Further similar models based on ontologies exists as pattern in ArchiMate for resource and capabilities by Azevedo et al. [3] as well as services by Griffo et al. [8]. Another approach by Petrovska et al. [17] describes security object relationships. The use case of oracle identity manager is not specific modelled according a systematic framework. Cioroaica et al. [23] provide a reference architecture for trust-based digital ecosystems. Their focus is on evaluation of trustworthiness on runtime. Even if it does not provide a direct input for FIM, we ensure the compatibility, especially regarding the interfaces.

4 Federated Identity Management Service Model

In this section, we utilize ArchiMate to achieve FIMSM, a generic scenario-independent service model for FIM. The universal FIMSM needs to fulfil the requirements, stated in Sect. 2. It thereby has to be reusable with generic and universal terminology (R1), while presenting a systematic overview (R2). The overview should provide interfaces between different entities (R4), throughout being adaptable (R3). These requirements are addressed in the following model. According to ArchiMate's approach, Fig. 1 shows the overview for the reference architecture FIMSM, using different layers to differentiating roles from business, applications, and technology. The reference architecture includes external layer, business service layer, business layer, application service layer, application layer, and the technical service layer. Consequently, a clear distinction is made between the provision of services and their implementation in the respective layer. We explain the universal approach according to top-down.

The **highest layer** describes the internal and external actors, i.e., the end users. On the top external/environment layer, a user wants to access a service. For authentication, he or she uses an I&AM system, which is accessed via a client application, e.g., a web browser. The user's IDP created an account with a digital identity for the user beforehand. Depending on the business of the IDP, it may perform an initial identity check using appropriate identification procedures, e.g., by verification of an ID card or bank account. Web services typically have no such or lower trusted verification methods. The roles are assigned to the account according to its position in their associated organization. Each user can take on different roles and assume different identities in relation to tasks. The user is either internal or external from a trusted organization. The latter describes typical FIM use cases.

The **business service layer** shows the internal and external usage by the end users. This layer helps to separate the internal structure and organization from the external observable behaviour expressed at the service layer. In the business service layer of FIMSM, i.e., the identity service provider layer, a corresponding service catalogue with trusted services is maintained. Such a list is usually not publicly available, but would be of great benefit. Access is provided by the process of access management, where the user can identify him- or herself by different methods. The most common authentication is via username and password. Increasingly, security is being enhanced by a second factor. Access management process triggers the business service of the authentication and authorization (AuthNZ) broker offering different methods. The core business is extended by another broker, which takes over the handling for trustworthy third parties. For this purpose, appropriate collaboration agreements are negotiated beforehand. This can range from formal contracts to black and white lists. In the policies, the corresponding technical specifications for the respective services are kept available to define the transfer criteria between providers and customers.

The **business layer** includes actors, roles, collaborations, and interfaces with regards of the provided service. These structural elements interact by processes, functions, interactions, events, and services. Combined they represent elements

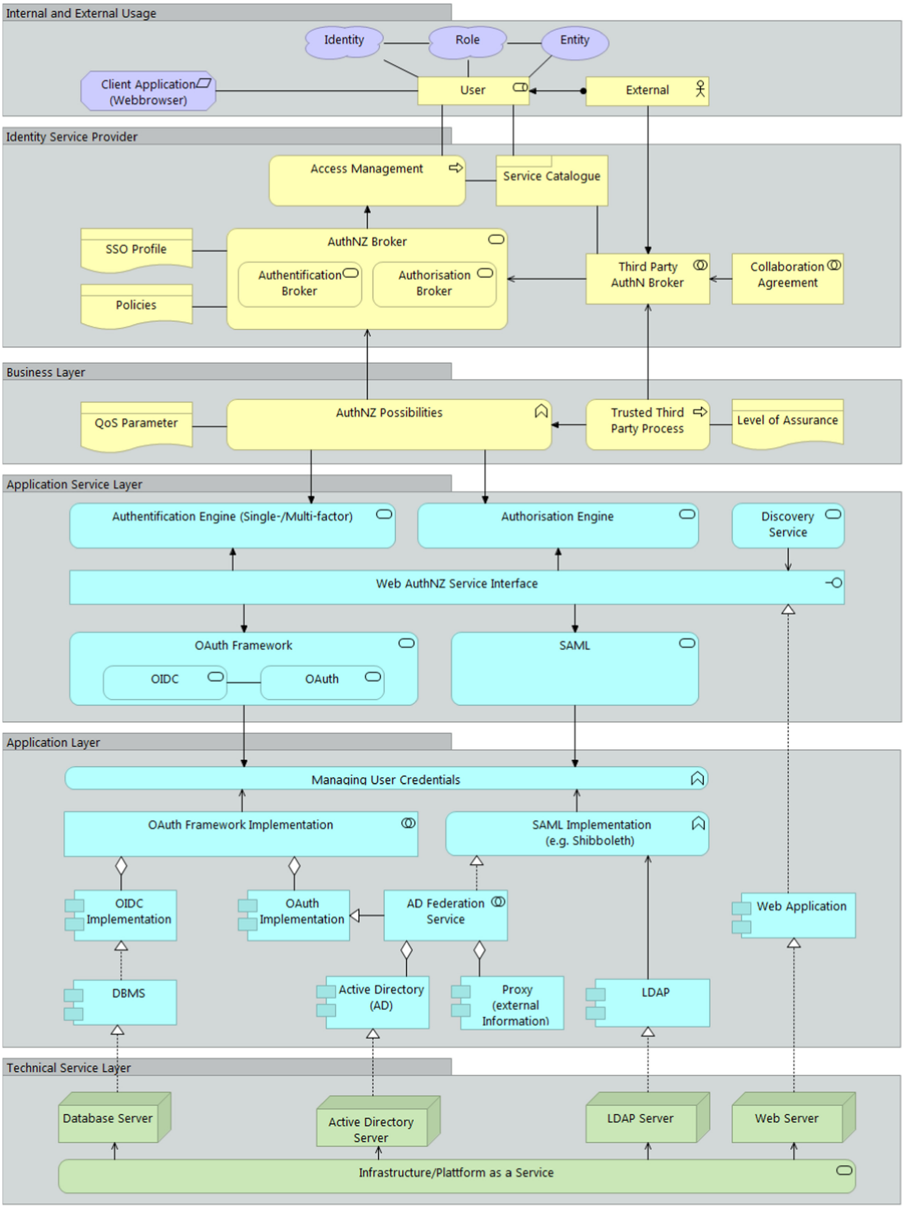


Fig. 1. Architecture for Federated Identity Management

with relevance from a business perspective. For the sake of clarity and the goal of presenting the overall context of the architectural overview, some elements are not included and will appear in the adjusted detailed models during realization. For FIMSM, the main service provided to end users and external entities is authentication and authorization, combined as AuthNZ possibilities. AuthNZ has quality of service (QoS) parameters, which the provider needs to fulfil. The service is only provided to externals, if both entities have a form of agreement or contract. The handling for external customers is embedded into the trusted third party process, which enables the trust establishment via an independent party. Participating entities may have requirements regarding security management, described by level of assurance. Level of assurance is the quantification of factors leading to confidence into an entity, relying on internal processes. This level of assurance can be stated within protocols.

The **application service layer** is built upon the application layer and shows accessible services and interfaces for other entities. The main service, especially for service providers, is the web service, where users need to authenticate. The web service therefore requests the authentication and authorization engines. In order to do this, the home organization of the user needs to be discovered, described by the discovery service. Often, a user request is redirected to or personal data of the user is requested from the IDP. The discovery service assists the web AuthNZ service. The web service itself uses and triggers different protocols, e.g., representational state transfer (REST) interface. Within FIM, this is mainly SAML and the OAuth framework with OAuth for authorization and OIDC for authentication. As several libraries and implementations exist for both protocol variants, many web service offer both.

The **application layer** gives an overview of the application layer concepts and their relationships. The concept describes different application components, software, and interfaces. In order to provide services to other entities and end users, several software components need to be implemented. The web service requires a web application, running on a web server, described within the technical service layer. The core part is surrounding I&AM itself, i.e., managing user credentials. Depending on the applied software, different protocols can be implemented. The user information itself is stored either in a database, AD, or LDAP. OIDC may be based on a database, but also uses other means of user storage. SAML typically utilizes LDAP. With AD in a federated use case, the extension AD FS needs to be implemented. AD FS provides SSO for different protocols, like OAuth and SAML. AD FS typically interacts with one SP. If a proxy is added, it can serve several SPs. An IDP configures roles and thereby permissions at the I&AM system. It is typical that services, independent of the location, have more fine-grained permissions locally. An entity is able to run all forms of FIM, but may decide on one specific software, implementation, and protocol. The most common variants are displayed. Further extensions can be added dependent on the use case.

The **technical service layer** consists the physical layer with actual hardware, including possible nodes, devices, infrastructure interfaces, communication paths, and networks. At the FIMSM technical service layer level, separate servers are provided for the various applications. This separation results in a better illustration and takes the idea of micro-service architectures, e.g., based on Docker, into account. In addition, specific security requirements can be implemented in a targeted manner. The different servers are made available as infrastructure or platform as a service scalable according to the cloud business service model.

5 Discussion

With the ever evolving extensions and revisions of FIM protocols, it is important to get the picture of the current status and gain a common understanding of the services offered. A reference architecture can help by providing a systematic way to model the domain services and associated interfaces. It though needs to fulfil several requirements to be of good use. The provided feedback from selected experts was embedded into the reference model. In FIMSM, we show the most common workflow. Several aspects of I&AM can be separated into business processes and the formal specification of it. In the following, we apply the reference architecture to the three organizations described in the scenario. This is followed by analyses of the requirements and research questions. Last but not least, we adapt the reference architecture for Kerberos. It is a protocol for client authentication, which therefore belongs to centralized I&AM for clients. Nevertheless, the principle of FIMSM can be adapted, showing that 1) it can be extended in future work for other identity management models as well as use cases and 2) it can help to identify combinations.

5.1 Application of the Reference Architecture on the Scenario

Since the goal of FIMSM is to be reusable for and adaptable to different use cases and protocols, the application of the reference architecture on the motivating scenario in Sect. 3 is shown. The university consists of a local I&AM based on LDAP, which enables SSO. For FIM, the SAML implementation Shibboleth is used. The architecture of the FIM implementation Shibboleth is typically shown as external systems, like LDAP, and one big module. Hence, FIMSM is more fine-grained. As the university is part of federations, the federation operators use TTPs to distribute federation metadata. Therefore, the following can be seen.

External Layer: The external layer consists of end users at the university, which have different identities and roles dependent on their function, like student, professor, or employee. These users utilize their browser to access services at the university and other entities.

Business Service Layer: In order to be able to access services, the user needs to authenticate. For this, a broker enables SSO.

Business Layer: The business layer of the university consists of the authentication and authorization functionality provided for internals and externals.

With the trusted third party process, the university enables user to make use of FIM. The university signed a contract with the federation operator. After being an official member of the federation, the university has to follow certain procedures to ensure assurance, described by level of assurance. This may be "Advanced" within DFN-AAI. Even though the entity is a university, it has internal quality of service parameters for the service.

Application Service Layer: The application service level shows the discovery service used by SP. As the university offers services, they run their own local discovery service. The discovery service displays possible IDPs based on the SAML metadata file pulled from the federation operators TTP. The second factor may be required, dependent on the service providers' needs.

Application Layer: The application layer of the university is also centred on user credential management. With Shibboleth as SAML implementation and LDAP along some web applications, the layer is rather lightweight.

Technical Service Layer: The university runs a LDAP server and web servers for identity management. Outside of the FIM scenario, the university has several other servers.

The identity management of the company is centred around AD with AD FS and OAuth/OIDC. The latter is used to grant access to the private cloud for internal as well as external users. Additionally, PAM and IoT identity management are enabled. In this paper, we concentrate on FIM. AD FS architecture is typically shown as AD FS servers upon AD with several services made available through a proxy. Therefore, the following model is more fine-grained.

External Layer: The external layer consists of end users, which are either associated with the company or business partners, having different identities with attributes, roles, and permissions.

Business Service Layer: The user wants to access services, e.g., the private cloud, and needs to authenticate before use. With AD FS, SSO should be implemented. The company holds agreements with partners for FIM.

Business Layer: As the company has both roles, IDP and SP, the TTP process and the level of assurance need to be regarded in both directions.

Application Service Layer: With both roles, the company provides web service as well as authentication and authorization engine. Instead of SAML, the company makes use of OIDC build on OAuth.

Application Layer: According to the protocols and software, the application layer is more complex. The company applies FIM protocols build upon AD with AD FS. Although OIDC often has an interface to a database, it can be added to other sources and implementations. Additionally, a web application, i.e., the private cloud, is shown.

Technical Service Layer: The technical service layer is a result from the layers above and consists of both, Active Directory Server and Web Server.

The hospital runs a local I&AM, which is not in the focus of the FIMSM. Nevertheless, basic concepts can be applied. In context with the scenario, the reference architecture also shows the need for further software and services. Therefore, it can be utilized to identify and model them in a controlled process.

External Layer: The users of the hospital are part of the external layer, having identities and roles, even though they are only used internally and locally.

Business Service Layer: Internally, authentication, authorization, and SSO are provided by a local I&AM. Without FIM, collaboration aspects cannot be specified.

Business Layer: The hospital internally offers authentication and authorization possibilities, but as FIM software is missing, the users cannot interact with other services. Hence, no trusted third party process is available.

Application Service Layer: For internal services, which use LDAP, authentication and authorization engines are available.

Application Layer: The LDAP server runs a LDAP implementation, like OpenLDAP. This implementation builds the basis for different FIM software.

Technical Service Layer: In consequence, this layer consists of a LDAP server.

5.2 Analysis of the FIMSM Based on the Problem Statement

Based on the motivating scenario, we obtained four requirements, which led to the research questions. In order to verify our results, we analyse the fulfilment of them.

R1: Reusable architecture for I&AM with generic and universal terminology. As shown in the previous section, the architecture can be reused for different protocols, since it provides a generic and universal terminology.

R2: Systematic overview and detailed perspectives on single aspects, like provided services. The application on the entities and services of the scenario was straight forward. Further division into entity roles could detail the layers.

R3: Adaptability to different protocols and use cases. The reference architecture is adaptable to SAML, OAuth, and OIDC with different implementations and constellations. Even though the focus is on FIM, it can be applied to centralized I&AM, identifying missing services. In future work, we plan to adapt the approach for it.

R4: Dependencies between different providers with related interfaces including requirements regarding an appropriate service management. With the TTP process including level of assurance and collaboration agreement, a business process interface is modelled. The other dependency is a technical one with the discovery service, embedded into the web service. Since quality of service is relevant, additional interfaces, e.g., regarding federated security management can be modelled and added in the future.

The analysis shows that the requirements are fulfilled. Additionally, future work is identified, helping to build a truly universal service model for I&AM. Regarding the stated research questions, the following is recognized.

Q1: How to describe I&AM scenarios with a scenario-independent approach? Different FIM scenarios can be described with FIMSM. Local I&AM modelled on the approach shows deficits for FIM use cases. The systematic can be applied to other user cases, e.g., Kerberos, shown in the next section.

Q2: Which elements are required to fulfil the requirements? The architecture includes different protocol variants, their implementations and service aspects as well as process steps for federated use case in a generic, implementation-independent form. The layers help to identify and distinguish between applications, underlying infrastructure as well as services at different levels.

Q3: Is it possible to adapt the reference architecture to different areas? With regards to the hospital example, the reference architecture can be applied and adapted to other areas. We apply the architecture to Kerberos without changes in the systematic in the following section.

5.3 Adaptation for Kerberos

Active Directory is built upon two main protocols: LDAP and Kerberos. Additionally, it needs Domain Name System (DNS). Kerberos is a computer network authentication protocol working with tickets to allow nodes communication over a non-secure network. It gives nodes the possibility to prove their identity to one another over this insecure network in a secure manner. The design is focused on a client-server model providing mutual authentication for both, the user and the server. Kerberos uses symmetric key cryptography and, therefore, requires a trusted party or public-key cryptography for the key exchange. The messages are hence protected against eavesdropping and replay attacks. Windows 2000 and later versions apply Kerberos as their default authentication method. It is therefore often found in companies, which have Windows clients. Joining a client to the Windows domain means enabling Kerberos as default protocol for authentication. The domain is a trust boundary similar to a federation on local level. Consequently, it is also a drawback, as user accounts and services need to have a trust relationship to the Kerberos token server. While Kerberos is utilized for computers, it can be used for, e.g., intranet web applications. Many UNIX like operating systems include software for Kerberos authentication of users or services. The reference architecture for FIM is applied for Kerberos. The adaptation is described from top to bottom per layer, see Fig. 2.

External Layer: The layer consists of users with a client application for authentication. Identities, roles, and permissions are left out for simplicity.

Business Service Layer: Authentication is provided to a client by tickets, making SSO available. Some sort of policies need to be followed. The key exchange is important for mutual trust.

Business Layer: Authentication engine describes the authentication process of Kerberos, which has quality of service parameters. The authentication is basically provided by the technical service layer, described below. As authentication is only possible within trust boundaries, trust needs to be established.

Application Service Layer: The main service provided to both, user and server, is authentication, i.e., Kerberos tickets. In order to do this, the end points need to be discovered.

Application Layer: Kerberos is typically integrated into software, like AD, which utilizes the protocols LDAP, DNS, and Kerberos. This can be combined with FIM, shown in the section beforehand.

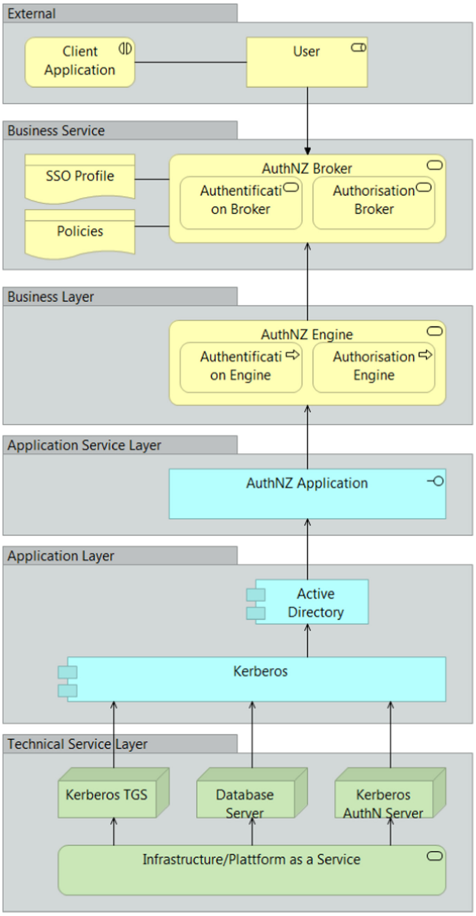


Fig. 2. Adapted architecture for Kerberos

Technical Service Layer: AD internally uses several different server applications running on Windows servers. Besides LDAP and DNS server, this is mainly the Key Distribution Center (KDC), consisting of Kerberos ticket-granting service (TGS) and Kerberos Authentication Server (KAS). The authentication server forwards the username to KDC, which issues a ticket-granting ticket, stamped and encrypted by TGS.

6 Conclusion and Outlook

Internet services and, thereby, identity management are more relevant than ever before. Different protocols were developed, while further extensions and revisions are in the making. The multitude of standards and solutions challenges users as well as system administrators to get an outline. The lack of interoperability further calls for a reference architecture to improve the current identity management landscape. In order to gain an overview and, therefore, an understanding of the services, we designed a reference service with the supporting toolkit of Enterprise Architecture using the framework ArchiMate.

Based on a motivating scenario, we identified four core requirements and three research questions. These are not fulfilled by current approaches targeting identity management and modelling. Therefore, we introduced FIMSM. It consists of several layers, describing a multitude of aspects related to the term service. The reference architecture includes external layer, business service layer, business layer, application service layer, application layer, and the technical service layer. It distinguishes services from applications, providing a better overview. Our reference model FIMSM is compliant with and extends the I&AM approach of ArchiMate itself. It integrates into the best practice solutions of ITIL, establishes the connection to TOGAF, and is conform to the Federated Mission Network, while using the standardized nomenclature. It thus also forms the link to the UML profile I&AM and the ontologies for service offer, negotiation, and contract for federations. In order to discuss FIMSM, we applied the motivating scenario and analysed the reference architecture based on requirements and research questions. In comparison to current architectures, our model is more fine-grained and at the same time generic. It is the first universal model and conform with frameworks. With the application of FIMSM to centralized I&AM, i.e., LDAP respectively Kerberos, we see that it can be adapted.

In future work, we plan to detail FIMSM for FIM with submodels and interfaces to business processes. This will help to establish federated processes in accordance to the components. We further aim to extend FIMSM to other identity management systems, e.g., for IoT and PAM, and models, e.g., user-centric and centralized identity management. This will assist to improve and extend the FIMSM with the goal to build a universal model for generic identity management. Last but not least, we intent to model further security related processes, mechanisms, and services.

References

1. Al Hadidi, F., Baghdadi, Y.: Ontology for enterprise interactions: extended and virtual enterprises. In: Baghdadi, Y., Harfouche, A. (eds.) ICT for a Better Life and a Better World. LNISO, vol. 30, pp. 365–379. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-10737-6_24
2. Amaral, G., Sales, T.P., Guizzardi, G., Almeida, J.P.A., Porello, D.: Modeling trust in enterprise architecture: a pattern language for archimate. In: Grabis, J., Bork, D. (eds.) PoEM 2020. LNBIP, vol. 400, pp. 73–89. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-63479-7_6
3. Azevedo, C.L., et al.: Modeling resources and capabilities in enterprise architecture: a well-founded ontology-based proposal for ArchiMate. Inf. Syst. **54**, 235–262 (2015)
4. Buschmann, F., et al.: Pattern-Oriented Software Architecture: On Patterns and Pattern Languages, vol. 5. Wiley, Hoboken (2007)

5. Dabrowski, M., Pacyna, P.: Modular reference framework architecture for identity management. In: IEEE International Conference on Communication Systems, Guangzhou, China, pp. 743–749 (2008)
6. Dhamija, R., Dusseault, L.: The seven flaws of identity management. IEEE Secur. Priv. **6**(2), 24–29 (2008)
7. Friebe, S., et al., DecentID: decentralized and privacy-preserving identity storage system using smart contracts. In: IEEE International Conference on Trust, Security and Privacy in Computing and Communications/ International Conference on Big Data Science and Engineering, New York, USA, pp. 37–42 (2018)
8. Griffo, C., et al.: From an ontology of service contracts to contract modeling in enterprise architecture. In: IEEE 21st International Enterprise Distributed Object Computing Conference, Quebec, Canada, pp. 40–49 (2017)
9. Heiland, E. et al.: Enterprise architecture model transformation engine. In: International Conference on Operations Research and Enterprise Systems, Online, pp. 137–144 (2021)
10. Katsikogiannis, G., et al.: An identity and access management approach for SOA. In: IEEE International Symposium on Signal Processing and Information Technology, Limassol, Cyprus, pp. 126–131 (2016)
11. Lord, N.: Uncovering Password Habits: Are Users' Password Security Habits Improving? Digital Guardian (2020). https://digitalguardian.com/blog/uncovering-password-habits-are-users-password-security-habits-improving-infographic. Accessed 22 May 2021
12. Oberhaus, R.: Rethinking enterprise architecture frameworks for the digital age. Int. J. Adv. Softw. **12**(1 & 2), 125–151 (2019)
13. Cao, Y., Yang, L.: A survey of Identity Management technology. In: IEEE International Conference on Information Theory and Information Security, Beijing, China, pp. 287–293 (2010)
14. Perroud, T., Inversini, R.: Enterprise Architecture Patterns. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37561-3
15. Pöhn, D., Hommel, W.: Management architecture for dynamic federated identity management. In: CS & IT Conference Proceedings, vol. 6, no. 6, pp. 211–226 (2016)
16. Pöhn, D., Hommel, W.: An overview of limitations and approaches in identity management. In: International Conference on Availability, Reliability and Security. Association for Computing Machinery, New York, USA, Article 90, pp. 1–10 (2020)
17. Petrovska, J., et al.: SOA approach - identity and access management for the risk management platform. In: Mediterranean Conference on Embedded Computing, Budva, Montenegro, pp. 1–4 (2019)
18. Schoonderbeek, J.: Modelling Identity in Enterprise Architecture, ArchiMate (2018). https://www.archimatetool.com/blog/2018/12/07/long-read-modelling-identity-in-enterprise-architecture-archimate/. Accessed 22 May 2021
19. The Open Group: ArchiMate 3.1 Specification (2017). https://pubs.opengroup.org/architecture/archimate3-doc/. Accessed 22 May 2021
20. Toth, K.C., Anderson-Priddy, A.: Self-sovereign digital identity: a paradigm shift for identity. IEEE Secur. Priv. **17**(3), 17–27 (2019)
21. Yang, Y., et al.: An identity and access management architecture in cloud. In: Seventh International Symposium on Computational Intelligence and Design, Hangzhou, China, pp. 200–203 (2014)

22. Zwattendorfer, B., et al.: An overview of cloud identity management-models. In: International Conference on Web Information Systems and Technologies, vol. 1, pp. 82–92 (2014)
23. Cioroaica, E., et al.: Reference architecture for trust-based digital ecosystems. In: IEEE International Conference on Software Architecture Companion, Salvador, Brazil, pp. 266–273 (2020)

Traceability from the Business Value Model to the Enterprise Architecture: A Case Study

Wilco Engelsman[1,3(✉)], Jaap Gordijn[2,4], Timber Haaker[1], Marten van Sinderen[3], and Roel Wieringa[2,3]

[1] Saxion, University of Applied Sciences, Enschede, Netherlands
w.engelsman@saxion.nl
[2] The Value Engineers, Soest, Netherlands
[3] University of Twente, Enschede, Netherlands
[4] Vrije Universiteit Amsterdam, Amsterdam, Netherlands

Abstract. Information and communication technology (ICT) is a significant part of business value propositions. Netflix and Spotify would not have been possible without internet technology. Therefore, it is not sufficient to consider the ICT of a business as a cost center only. Rather it drives profit, and hence should be considered in concert with the business value model of a company. In previous research we have defined guidelines to transform a business value model into an enterprise architecture.

In this paper, we validate the set of guidelines in a real-world case study, in which we created the business value model of a ecosystem and used our guidelines to redesign its enterprise architecture (EA). We quantified the business value model with revenues and expenses for the company. We validated the resulting models and their traceability with management and the enterprise architect of the company. The result is a further improvement and simplification of our guidelines.

In this paper we present the case study, the models and the resulting guidelines. We end the paper with a discussion of further research needed to bring these research results to practice.

Keywords: e^3*value* · ArchiMate · Traceability · Business value model · Enterprise architecture

1 Introduction

Commercial services and physical products rely heavily on ICT. For example, Netflix and Spotify would not have been possible without the large scale deployment of content servers and networks. Physical products often have digital twins, which complement the product with additional features, allowing for simulation, training, etc. Since ICT is an intrinsic part of the value proposition of an organization, it can not be considered as a cost-only factor. ICT should be part of value proposition design.

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 212–227, 2021.
https://doi.org/10.1007/978-3-030-79186-5_14

Additionally, many products and services are offered as bundles in complex *business ecosystems*, where each enterprise focuses on its core competence and jointly they satisfy a complex customer need. Following Moore [15], we define an ecosystem as a "collection of companies that work cooperatively and competitively to satisfy customer needs."

In order to be financially sustainable, an ecosystem requires a *business value model* (henceforth called "business model"), which we define as a conceptual model that represents the creation, distribution, and consumption of economic valuable objects in a *network* of participants, namely the ecosystem [7]. Valuable objects are services and products that satisfy customer needs, as well as payment for these; also called the reciprocal value transfers.

With ICT-intensive services and products, the design of the provisioning *Enterprise Architecture* (EA) is part of business design. An EA is a high-level conceptual model of an enterprise designed to put the business strategy of an organization into operation [22]. Ideally, in case of ICT intensive products and services the EA puts the business model into operation and hence contributes directly to the profit of enterprise. For this, we need an approach to design the EA of ICT-intensive products and service *in concert* with the business model of the eco-system. Currently, there is no such approach.

As we take a network view of business models, we use e^3*value* as the business model notation [7,8]. In accordance with our networked view, EAs too should be extended to an ecosystem of enterprises [20]. We use ArchiMate [18] as the EA language, where we focus on its capability to model business services and collaborations.

An e^3*value* model focuses on actors in a value network and the economic feasibility of the value adding activities in an organization. An ArchiMate model operationalizes this in terms of business services and collaborations, business processes and applications needed to realize this. The models contain different, but also partly overlapping information. A transformation from an e^3*value* model to an ArchiMate model can therefore not be automatic and the guidelines defined in this paper must be complemented with design choices. The contribution of this paper is that we provide a real-world validation of transformation guidelines we developed earlier, and present further improvements to the guidelines to make them more precise. We also elicit an evaluation from practitioners of how to use the conjunction of e^3*value* design and EA design for investment analysis. This paper is structured as follows. In Sect. 2 we describe our research methodology and research questions. Section 3 lists our redesigned guidelines and Sect. 4 describes our case study. Section 5 describes related work. We discuss our results in Sect. 6 and conclude with an indication of future research in Sect. 7.

2 Methodology and Research Questions

We have developed guidelines for business-model-driven EA design in three iterations of the design cycle [21].

- **Conceptual design:** We analyzed the metamodels of e^3*value* and ArchiMate to define an initial version of the guidelines (version 1). We tested it on a small real-world example: an EA for the Cirque du Soleil [4].

– **Lab validation and redesign:** We tested the guidelines in an experiment where we compared the EAs designed by practicing architects from a business model in a laboratory assignment, with the EA that results from our application of the guidelines [5]. Althouh the assignment took place in the lab, the cases for which the architects designed an e^3*value* model and an EA were from the real world: the companies where they were employed. Analysis of the experiment led to a redesign of the guidelines (version 2).
– **Real-world validation and redesign:** We applied the guidelines to a real-world case to redesign the business layer of the enterprise architecture of an enterprise. This is the case study reported in this paper. This experience led to a further improvement of the guidelines (version 3).

Although version 3 of the guidelines are the result of applying version 2 on the case study, for readability we present version 3 in Sect. 3 before presenting the case study in Sect. 4.

The case study is technical action research, as it has two goals: to learn more about the guidelines and refine them, and if we can construct a correct business layer architecture from a e^3*value* business model diagram. [21]. We validated the resulting enterprise architecture, and the e^3*value* model that we designed, with management and the enterprise architect of the company. This allowed us to answer the following research questions:

Q1. Do the guidelines produce a correct enterprise architecture of the business model?
Q2. Is the resulting traceability relation useful to make investment and divestment decisions?

To preserve confidentiality, we refer to this company as company X and we changed some details in the models that we will present.

3 Redesigned Guidelines

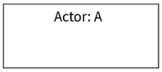
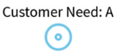
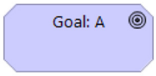
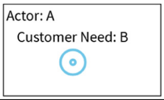
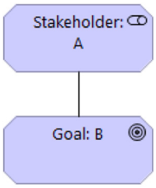
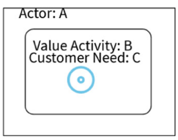
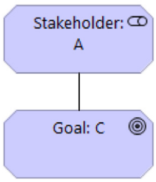
Figure 1 provides the legends of e^3*value* and ArchiMate that we use in this paper[1]. Tables 1, 2 and 3 present some of our revised guidelines. Table 1 gives one of the guidelines to transform an e^3*value* model into an ArchiMate motivational model[2]. Basically, the idea is that:

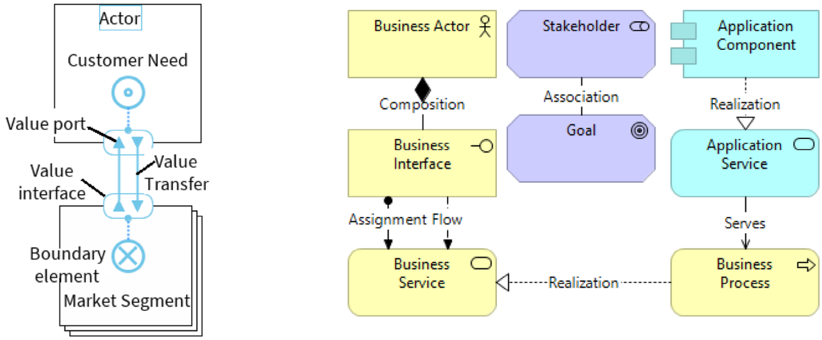
– An e^3*value* actor is mapped to an ArchiMate stakeholder. The reason is that an e^3*value* actor is an actor who has something to gain or lose.
– A need of an e^3*value* actor is mapped to an ArchiMate stakeholder goal. The reason is that a need in e^3*value* is a lack of something valuable that the actors wants to acquire. In other words, it is a goal.

[1] See the e^3*value* user guide at https://e3value-user-manual.thevalueengineers.nl/ and the ArchiMate documentation at https://pubs.opengroup.org/architecture/archimate3-doc/.

[2] A version of the paper with the complete table is available at https://www.thevalueengineers.nl/pdf/EMMSAD-2021-long.pdf.

Table 1. e^3*value* mapping to the motivation layer of ArchiMate.

No	Guideline	Additional advice
G1	An e^3*value* actor or a market segment can always be modeled as a stakeholder in the ArchiMate motivation layer with the same name. By definition an actor is always a stakeholder. This is not true for the other way around. 	Additional detail can be added to the stakeholder using the composition, aggregation or specialization relation in ArchiMate. It is not always necessary to model every actor as a stakeholder. This is a choice the enterprise architect has to make 
G2	A customer need must be modeled as a goal from the ArchiMate motivation layer with the same name as from e^3*value* 	Construct a complete and correct goal model if needed. 
G3	When the e^3*value* actor contains a customer need, this combination must be modeled using stakeholder, goal and association relationship from the motivation layer of ArchiMate. 	This is a combination of G1 and G2. 
G4	When the e^3*value* actor contains a value activity and a customer need, this combination must be modeled using stakeholder, goal and association relationship from the motivation layer of ArchiMate. 	This is a combination of G1 and G2. The value activity is not translated into the motivation layer. 



(a) Legend for e^3*value* . (b) Legend for ArchiMate.

Fig. 1. Legends for e^3*value* and ArchiMate

Tables 2 and 3 list the guidelines for designing an ArchiMate business layer model from an e^3*value* value model. Compared to version 2 [5], we improved the mapping of ports and collaborations, merged a few rules and added rules for and- and or-gates. The tables show the guidelines for designing an EA of a focal company, which is embedded in a network of companies. To design the EA of more than one company, the guidelines have to be applied to each company.

- An e^3*value* actor is mapped to an ArchiMate business actor.
  A e^3*value* actor is an entity that is responsible for its survival and well-being, e.g. a profit-and-loss responsible company or a consumer [7]. An ArchiMate business actor is a business entity that is capable of performing behavior [18]. This implies that all e^3*value* actors are ArchiMate actors but not the other way around. By definition an ArchiMate actor is always a stakeholder, as he has something to gain or lose. e^3*value* actors can therefore be depicted on both the stakeholder and business actor concepts.
- An e^3*value* value activity of an actor becomes an ArchiMate business service of that actor.
  An e^3*value* activity is a task performed by an actor that potentially results in a benefit for the actor [7]. An ArchiMate business service is explicitly defined behavior that a business role, business actor, or business collaboration exposes to its environment [18]. We view every e^3*value* activity as a potential business service exposed by the focal company. A value activity is connected to its environment through a value port, similarly as a business service is connected to its environment. This way, a value activity cannot be mapped to a business process, since a business process is internal to the organization.
- An e^3*value* value port of an actor becomes an ArchiMate business interface of a that actor. This is a change of version 2.
  An e^3*value* value port is a willingness to provide or request something of value (a value object) [7]. ArchiMate business interfaces are channels through which a business service is made available. An value port has no

direct counterpart in ArchiMate. However a value port can be composed into an ArchiMate business interface. This adds a channel. Value ports from multiple value interfaces from an OR dependency graph can be mapped to a single business interface. One or more value ports of a single business interface or an AND dependency graph can also be represented as a single business interface. When there are different channels to a value port, multiple business interfaces are mapped to value ports from a single value interface.

- An e^3*value* value interface is a collection of two or more ports that defines a commercial transaction. This is not translated into an ArchiMate model component because transactionality of commercial transactions is not represented in ArchiMate.
- An e^3*value* value transfer can be mapped to an ArchiMate flow relation and to an ArchiMate serving relation with a direction that depends on the direction of the e^3*value* dependency path.

  An e^3*value* transfer is a willingness to transfer a value object from provider to requester [7]. A flow relationship is a transfer of information, goods or money between elements [18]. In addition, value transfers are part of a dependency path that starts from one or more needs. This determines which elements delivers a servive to which other element (the serving relation) [18].
- A value transfer is associated with a *value object* in e^3*value*.

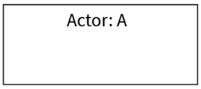
  A value object is a money- or non-money object that is of economic value for at least one other actor in the value network. There are multiple ways to depict a value object in ArchiMate. A value object can be tangible or intangible. A tangible value object may be represented as a *business object* in ArchiMate. The *value* concept can be used for intangible value objects. But money is neither a business object nor value concept. Because of this, a value object will not be translated into ArchiMate for now.

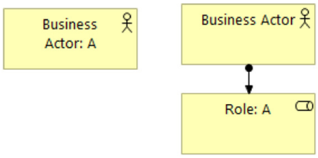
4 Case Study

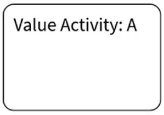
4.1 The Company

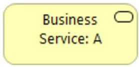
Company X is responsible for building startups based on an acquisition of intellectual property. The main goal of organization is to increase the share value of the startup and finally sell the startup to other investors. Company X has three major value-adding activities: Scouting new technology, supporting startups during their growth, and selling matured startups. Support provided by X ranges from HRM, legal, financial administration, providing management, etc. We cannot provide exact details and the business model below differs somewhat from the actual business model. The company had an existing EA in place.

Table 2. e^3*value* mapping to the Business layer.

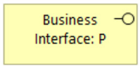
<table>
<tr><th>No</th><th>Guideline</th><th>Additional advice</th></tr>
<tr><td>G5</td><td>An e^3value actor or market segment is mapped to an ArchiMate business actor with the same name, or to an ArchiMate actor that assumes a role with that name.


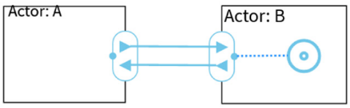
</td><td>In ArchiMate we can identify additional business actors. For example, we may identify actors internal to an organization and we may decompose an actor.


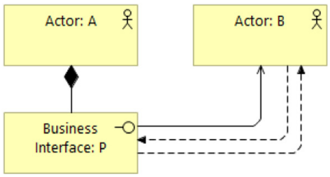
</td></tr>
<tr><td>G6</td><td>An e^3value value activity is mapped to an ArchiMate business service with the same name.


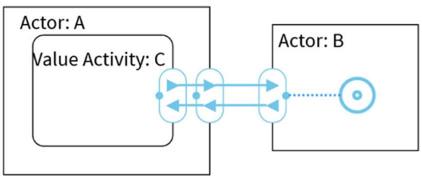
</td><td>Services can be internal or external to the organization itself. Additional detail and service composition might be required.


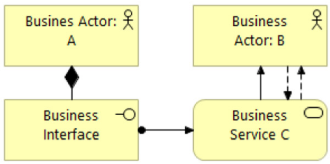
</td></tr>
<tr><td>G7</td><td>e^3value value ports are mapped to ArchiMate business interfaces


</td><td>one or more value ports from one or more value interfaces of a same value activity can be mapped to the same ArchiMate business interface.


</td></tr>
<tr><td>G8</td><td>An e^3value value exchange is mapped to ArchiMate flows. In addition, the exchange can be mapped to an ArchiMate serving relation in the direction of supplier to customer. Ports of the focal company will be mapped to one or more ArchiMate business interfaces.


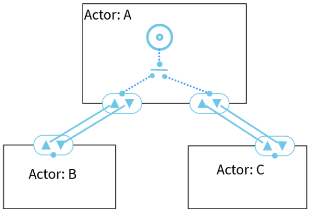
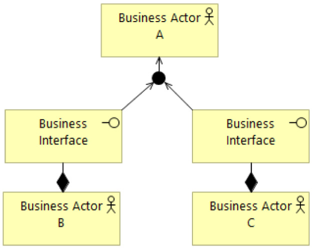
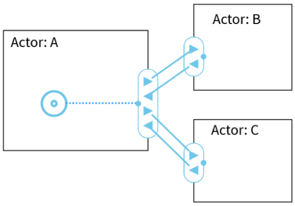
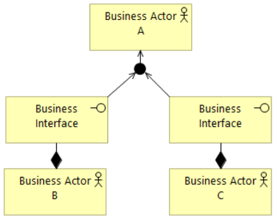
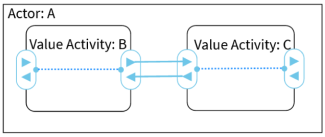
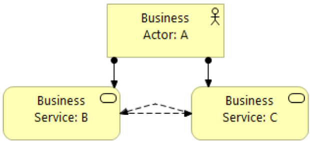
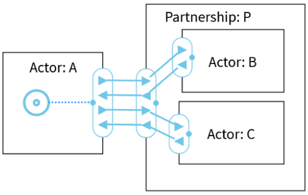
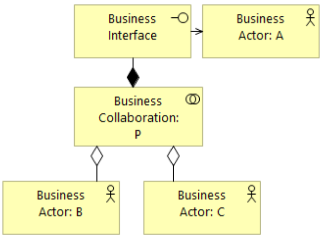
</td><td>If B contains a boundary element instead of a need, the direction of the serving relation would be reversed.


</td></tr>
<tr><td>G9</td><td>An e^3value activity connected through a value exchange to a need of an actor is mapped to an ArchiMate business service serving the actor.


</td><td>If B contains a boundary element instead of a need, the direction of the serving relation would be reversed.


</td></tr>
</table>

Table 3. e^3*value* mapping to the Business layer

<table>
<tr><th>No</th><th>Guideline</th><th>Additional advice</th></tr>
<tr><td>G10</td><td>An AND/OR gate in e^3value maps to an and/or junction in ArchiMate.

</td><td>For the or junction the connector in ArchiMate is a hollow circle. Flows can be added as needed as in G8.

</td></tr>
<tr><td>G11</td><td>Two e^3value actors connect to a single value interface to address a customer need are mapped to an and-junction in ArchiMate in the same way as in G11.

</td><td>
</td></tr>
<tr><td>G12</td><td>An e^3value value exchange between two value activities inside an actor maps to mutual flows between two ArchiMate business services of that actor.

</td><td>The flow relation denotes the transfer of money, information or goods. If the direction of the dependency path is known, this can be represented by a serving relation in ArchiMate.

</td></tr>
<tr><td>G13</td><td>A composite actor in e^3value is mapped to a business collaboration in ArchiMate.

</td><td>The business collaboration will offer services from both actors as a bundle. Use a composition or aggregation relation between the parent and child services.

</td></tr>
</table>

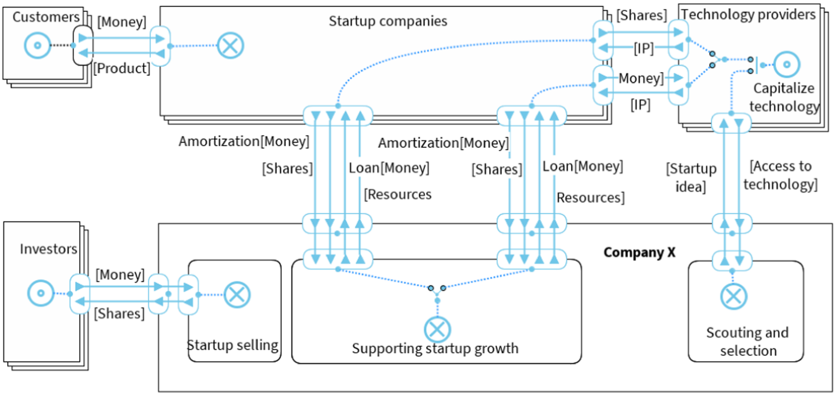


Fig. 2. e^3*value* model of company X.

4.2 Application of the Guidelines

Figure 2 shows the business model of X. An e^3*value* model represents commercial transactions in a time period called the *contract period*.

In the model of Fig. 2, company X performs three value-adding activities, Scouting and selection, Supporting startup growth, and Startup selling. Technology providers have the need to capitalize technology. To satisfy this need, they exchange access to technology for a startup idea with company X and they transfer IP in the technology to a startup. IP can be transferred in exchange for shares or in exchange for money. In both cases, X lends money to the startup, makes resources available, receives startup shares and receives the amortization of the loan. However, only a fraction of the shares will be transferred if the technology provider receives part of the shares. This will be represented in the quantified model, discussed later.

Since this model contains no time ordering, it provides no information about when the loan is given, when the amortization takes place, or when the startup is sold. The model in Fig. 2 shows *all* of the commercial transactions and value activities that X is involved in during the contract period. These are the activities to be supported by an EA.

Figure 3 shows a high level layered EA model of X. The top part is constructed by our guidelines. We have also included a simple goal model. The diagrams have been annotated with the names of the guidelines applied. The part of the EA below the line had already been designed by the enterprise architect of X. Our guidelines produced the part of the enterprise architecture above the line. This part differs from the enterprise architect's model; most of it was absent. In particular, the business services were not complete and the external actors were absent.

4.3 Quantification

A quantification of an e^3*value* model is called a *market scenario* [7]. For example, Fig. 4 shows a quantified version of the model in Fig. 2. It quantifies the size of

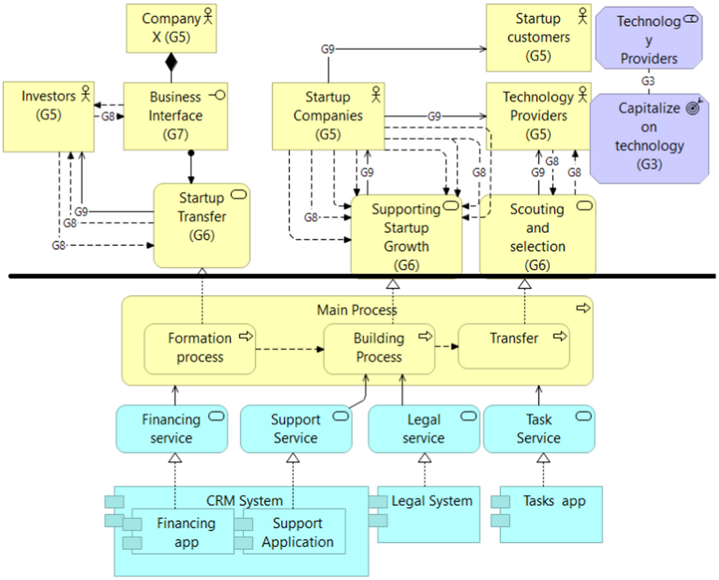


Fig. 3. Layered EA model of X.

market segments, the average number of needs per actor in a market segment, expenses of value activities, the value of money flows, the distribution of choices, and the value of any other variable that we introduced. Here, we introduce the initial share value of a startup as a variable. We use "f" as a generic currency symbol (read: "Florin"). The numbers in Fig. 4 are arbitrary and do not reflect company X.

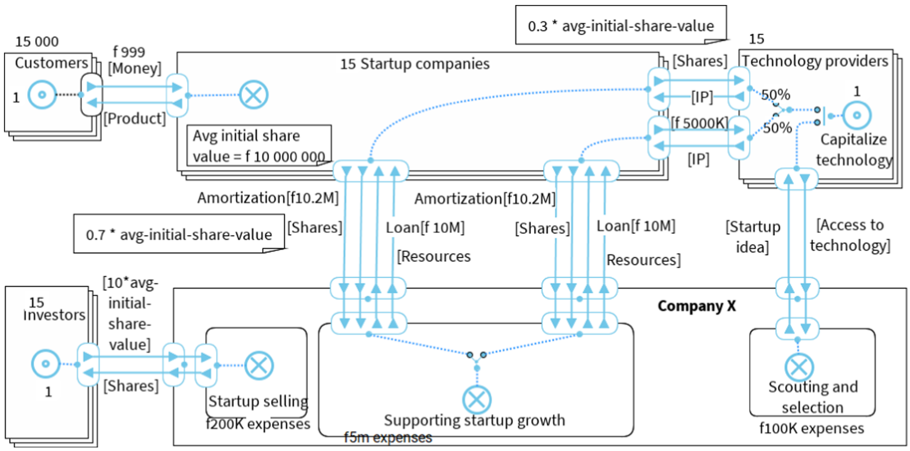
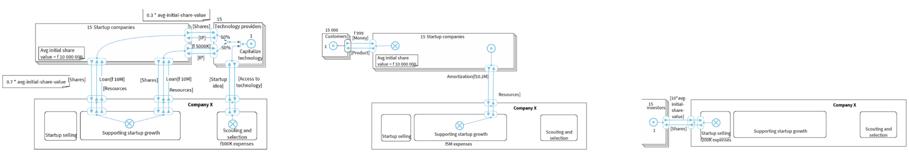


Fig. 4. A market scenario for company X. "f"is a generic currency symbol.



(a) Period 0: investment (b) Period 1: growth (c) Period 2: sale

Fig. 5. Sketch of a time series for company X. The three models are extracted from the market scenario in Fig. 4.

Each actor has revenues and expenses, and adding these up, the e^3*value* tool computes that the company has a net revenue of f 747 M in this scenario. By making many different scenarios, we can assess how sensitive the business model is to differences in market assumptions.

A *time series* is a sequence of scenarios with an interest rate used for net present value (NPV) computations. Figure 5 shows what a time series looks like.[3] In the first period, X makes an investment in startups. In the second period, these startups do business and in the third period, they are all sold to investors. Using the quantifications of Fig. 4 and a fictional interest rate of 2%, our tool computes an NPV in period 0 of f 571 M. By varying the quantifications, an investment risk analysis can be done.

4.4 Expert Evaluations

In order to validate and elaborate on the correctness and utility of the realized traceability we organized a validation and requirements elicitation session with management and the enterprise architect of X.

To answer Q1, we made some mistakes in our initial business model, which we corrected. The mistakes were not in the value activities or the actors in the value network, but how actual value was created. For example in our initial business model it is possible that not all shares are sold to X. In reality all shares are owned by X from startup creation. Also,the actual costs structure is completely different from our quantification. However, these mistakes have no impact on the resulting EA model.

The EA of Fig. 3 correctly represents the IT environment and business services of X and the upper part, designed by us, maps properly to the lower part, designed by the enterprise architect. This was also validated with the enterprise architect.

Our upper-part extension of the EA embodies an improved traceability from the EA to the e^3*value* business model, and the discussion revolved around answering Q2: Is this traceability useful for investment and divestment decisions?

This discussion turned into a requirements engineering session for tool support. Management of X wish to determine what the effect of changes in the business model on the EA is. Their strategic goal is to scale up to more startups than they have now (a quantitative goal) and they need to decide on the best investment in IT to support this goal. The value activities of X will not change but the number of technology providers and startups they interact with will change. Traceability between an e^3*value* model and an EA is a nice-to-have; the traceability will be considerably more useful to X if different quantified time series can be related to IT investments. X was particularly interested in being able to use NPV to evaluate different IT architectures from the business model in a top down manner and to be able to perform scenario generation and evaluation.

[3] Included to give a rough impression only. A version of the paper with a readable time series is available at https://www.thevalueengineers.nl/pdf/EMMSAD-2021-long.pdf.

5 Related Work

In previous work we have created transformation guidelines between e^3*value* and ArchiMate [4,5]. This work extends on our previous work by applying the guidelines in a real-world scenario, an initial evaluation of utility by practitioners and a redesign of the guidelines based on this exercise.

Derzi et al. [3] propose traceability between UML deployment diagrams and e^3*value* . They annotate UML deployment diagrams with costs and benefits and create traceability between deployment diagrams and e^3*value* to be able analyze the profitability of an organization with the proposed IT. We have a similar goal but at a higher holistic level, because we use ArchiMate instead of the UML. Cost estimation usually takes place at the enterprise architecture level and not at the detailed UML level.

Meertens et al. [10,13] propose a mapping from the Business Model Canvas (BMC) [16] to ArchiMate. This means that they miss the business model network view that we think is essential, and they do not have the capability to quantify the business model and simulate market scenarios, as e^3*value* has.

Related to this is the work of Iacob et al. [9] who propose a method for IT portfolio evaluation using ArchiMate. Aldea et al. also propose a way to link EA to the business strategy of the organization [1]. This work focusses on tracing business goals to an enterprise architecture. In our guidelines we relate a business model to the motivational extension of ArchiMate, which includes business goals. Also relevant is the work of Fritscher and Pigneur [6]. They link EA with the BMC on a very coarse-grained level. They do not realize actual traceability to different concepts of different languages nor do they provide guidelines or building blocks.

Petrikina et al. [17] describe a preliminary investigation about linking business models with EA at the meta-model level. The authors propose to link the business model to the products and services and create a new meta-model. However, this work is preliminary and they have not identified any transformation guidelines.

Aldea et al. propose adaptations of ArchiMate to incorporate value modeling [2]. Recently the Open Group also proposed to incorporate business modeling concepts in ArchiMate [19]. The result is a version of ArchiMate that has even more symbols and concepts than it has now. And it does not solve the problem of traceability between business models and enterprise architectures, because guidelines are absent.

Our approach differs from others because we use a networked approach to business models, allow quantification of business models, and define and test traceability guidelines to transform business models into an enterprise architecture. As argued elsewhere, business models as well as enterprise architectures should be networked [12,20]. And we do not want to integrate business modeling and enterprise architecture design in one language.

De Kinderen, Gaaloul and Proper propose to link ArchiMate to e^3*value* using an intermediary language. They do not propose a direct mapping [11] and they do not provide guidelines.

Our work is different, because we focus on realizing traceabilty to the value network with guidelines. Using these guidelines we will create traceabilty to the value network from ArchiMate. We will not use intermediary languages to realize this. We see this work as an improvement of existing work.

6 Discussion

6.1 Traceability

Our application of the guidelines showed that we can produce the upper part of an ArchiMate EA that is a sound basis for designing a complete EA aligned to an e^3*value* model. Such an alignment is needed in order to relate IT expenses to value-adding business activities. All value-adding business services are included in the EA model, and they are related to interfaces with other companies. However, expert feedback told us that to be of use in investment decisions, this traceability relation should allow quantification.

6.2 A Business Model-Driven Method for EA Design

Designing a business model and an EA requires many decisions and we found it expedient to use the following steps:

1. Construct an e^3*value* model for the value network of the focal organization.
2. Construct an ArchiMate motivation model.
   - Create a goal model for the organization using the guidelines of Table 1.
   - Elaborate this goal model using ArchiMate guidelines and relations (composition, aggregation, specialization).
3. Construct an initial ArchiMate business layer model.
   - Construct a high level business service architecture (Tables 2 and 3).
   - Identify sub-services where needed using standard ArchiMate modeling guidelines and operations (composition and aggregation).
4. Design the business processes and application architecture and link them to the service architecture.

We consider this as a lesson learned from this project and we will use this method for our next case study and to teach to students.

6.3 Validity

Internal validity is the extent to which the outcome of an experiment has been produced by the treatment. In this action experiment, the outcome is an EA and the treatment is our set of guidelines. Our description in this paper shows that the outcome is indeed produced by this treatment.

The *utility* of this outcome is still an open question. The traceability that we established is a nice-to-have, but to be useful in investment and divestment decisions, we need to provide tool support to relate IT expenses to revenue in different investment scenarios.

Another open issue is the *external validity* of this treatment. Can other people use these guidelines and come up with similar results? Are these guidelines sufficient for all companies? Are the resulting EAs useful for other companies too? To answer these questions we need to do more case studies and experiments, in which we ask other people to use these guidelines for other companies.

A higher-level external validity question is whether guidelines like these can be used with other business modeling and EA languages. Achieving that level of generality is not our goal. Since our guidelines are derived from an analysis of the metamodels of e^3*value* and ArchiMate and refined in experiments and case studies using these languages, we do not expect generalizability beyond these languages.

7 Conclusion and Future Work

We conclude that establishing traceability between an EA and an e^3*value* business model is possible in practice and is potentially useful if we can quantify this traceability relationship. In future research, we will define a relationship between IT investments and company revenue and test this in a new case study. There is some previous research that we can build on [3,14].

References

1. Aldea, A., Iacob, M.E., Quartel, D.: From business strategy to enterprise architecture and back. In: 2018 IEEE 22nd International Enterprise Distributed Object Computing Workshop (EDOCW), pp. 145–152. IEEE (2018)
2. Aldea, A., Iacob, M.E., van Hillegersberg, J., Quartel, D., Franken, H.: Modelling value with ArchiMate. In: Persson, A., Stirna, J. (eds.) CAiSE 2015. LNBIP, vol. 215, pp. 375–388. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19243-7_35
3. Derzsi, Z., Gordijn, J., Kok, K., Akkermans, H., Tan, Y.-H.: Assessing feasibility of IT-enabled networked value constellations: a case study in the electricity sector. In: Krogstie, J., Opdahl, A., Sindre, G. (eds.) CAiSE 2007. LNCS, vol. 4495, pp. 66–80. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72988-4_6
4. Engelsman, W., Wieringa, R.J., van Sinderen, M., Gordijn, J., Haaker, T.: Realizing traceability from the business model to enterprise architecture. In: Guizzardi, G., Gailly, F., Suzana Pitangueira Maciel, R. (eds.) ER 2019. LNCS, vol. 11787, pp. 37–46. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34146-6_4
5. Engelsman, W., Wieringa, R., van Sinderen, M., Gordijn, J., Haaker, T.: Transforming e3value models in ArchiMate diagrams. In: 2020 IEEE 24th International Enterprise Distributed Object Computing Conference, pp. 11–20 (2020). Springer
6. Fritscher, B., Pigneur, Y.: Business it alignment between business model and enterprise architecture with a strategic perspective. Int. J. Inf. Syst. Model. Design **6**(1), 1–23 (2015)
7. Gordijn, J., Wieringa, R.J.: e^3value User guide. The Value Engineers (2021). https://e3value-user-manual.thevalueengineers.nl/
8. Gordijn, J., Akkermans, J.M.: Value Webs: Understanding E-business Innovation. The Value Engineers (2018). www.thevalueengineers.nl

9. Iacob, M.-E., Quartel, D. and Jonkers, H.: Capturing business strategy and value in enterprise architecture to support portfolio valuation. In: 2012 IEEE 16th International Enterprise Distributed Object Computing Conference, pp. 11–20. IEEE (2012)
10. Iacob, M.-E., Meertens, L.O., Jonkers, H., Quartel, D.A., Nieuwenhuis, L.J., Van Sinderen, M.J.: From enterprise architecture to business models and back. Softw. Syst. Model. **13**(3), 1059–1083 (2014)
11. de Kinderen, S., Gaaloul, K., Proper, H.A.: Bridging value modelling to ArchiMate via transaction modelling. Softw. Syst. Model. **13**(3), 1043–1057 (2014)
12. Mayer-Schönberger, V., Ramge, T.: Reinventing Capitalism in the Age of Big Data. Basic Books (2018)
13. Meertens, L.O., Iacob, M.E., Nieuwenhuis, L.J., van Sinderen, M.J., Jonkers, H., Quartel, D.: Mapping the business model canvas to ArchiMate. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing, pp. 1694–1701. ACM (2012)
14. Miguens, J., da Silva, M.M., Guerreiro, S.: A viewpoint for integrating costs in enterprise architecture. In: Panetto, H., Debruyne, C., Proper, H.A., Ardagna, C.A., Roman, D., Meersman, R. (eds.) OTM 2018. LNCS, vol. 11229, pp. 481–497. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02610-3_27
15. Moore, J.F.: The Death of Competition: Leadership and Strategy in the Age of Business Ecosystems. Harper (1996)
16. Osterwalder, A., Pigneur, Y.: Business model generation: a handbook for Visionaries, Game Changers, and Challengers. Wiley (2010)
17. Petrikina, J., Drews, P., Schirmer, I., Zimmermann, K.: Integrating business models and enterprise architecture. In: 2014 IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations, pp. 47–56. IEEE (2014)
18. The Open Group: ArchiMate 3.0.1 Specification. Van Haren Publishing (2017)
19. Walters, E.: Modeling the business model canvas with the ArchiMate specification (2020). https://publications.opengroup.org/w195
20. Wieringa, R., Engelsman, W., Gordijn, J., Ionita, D.: A business ecosystem architecture modeling framework. In: 2019 IEEE 21st Conference on Business Informatics (CBI), vol. 1, pp. 147–156. IEEE (2019)
21. 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
22. Zachman, J.A.: The concise definition of the Zachman framework (2017). https://www.zachman.com/about-the-zachman-framework

Design Thinking and Enterprise Modeling: An Investigation of Eight Enterprise Architecture Management Projects

Hasan Koç[1(✉)], Kurt Sandkuhl[2], and Janis Stirna[3]

[1] Berlin International University, Salzufer 6, 12524 Berlin, Germany
koc@berlin-international.de
[2] Rostock University, Albert-Einstein-Straße 22, 18051 Rostock, Germany
kurt.sandkuhl@uni-rostock.de
[3] Stockholm University, Postbox 7003, 164 07 Kista, Sweden
js@dsv.su.se

Abstract. Organizations adopt new tools and technologies even more during the Covid19 pandemic. Enterprise Architecture Management (EAM) as a holistic management discipline supports systematic adoption processes by integration business and IT concerns. This integrated view requires the participation of different stakeholders to EAM projects. To minimize the risks resulting from this multi-perspective aspect, the principles of Design Thinking methods should be applied in the early phases of EAM. For Enterprise Modeling (4EM) is an Enterprise Modeling method that exhibits Design Thinking characteristics. By examining eight EAM projects from qualitative and quantitative perspectives, this paper tackles the question whether the 4EM as a Design Thinking Method contributes to the goals of the EAM projects.

Keywords: Enterprise Architecture Management · Goal modeling · Design thinking · Enterprise modeling

1 Introduction

Organizations are seeking responses to the business challenges on a daily basis. Due to ever-changing landscapes, high number of transformation projects, pressure from the competitors and the need to make sense of data, flexibility becomes more and more important. For example, many organizations have shifted to remote-working models almost overnight in order to operate businesses during the Covid19 pandemic. Employees across all functions have learned how to complete tasks remotely, using digital communication and collaboration tools. This required new tools, technologies and processes to be adopted at an unprecedented pace.

Enterprise Architecture Management (EAM) is a well-established discipline in Information Systems. Connecting different aspects of an enterprise and thus showing the interdependencies by means of graphical notations, EAM supports systematic adoption processes of tools and technologies by integrating business and IT concerns. Offering

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 228–242, 2021.
https://doi.org/10.1007/978-3-030-79186-5_15

a holistic view on the structure of companies, EAM helps firms to align their organizational structures and Information Technology (IT) artefacts. As Enterprise Architectures traditionally consist of different dimensions, e.g., Business Architecture, Technology Architecture and Data Architecture, taking into account different stakeholder views, such as, goals, requirements and most importantly backgrounds, is a crucial aspect to align Business and IT. To minimize the risks resulting from this multi-perspective aspect, one way could be the application of Design Thinking methods especially in the early phases of the EAM projects. Design Thinking is an approach which is based on the solution of wicked problems by taking a user-oriented stance and defined as a mindset, process and a toolbox [1]. Studies indicate an increasing number of publications as well as adoption of Design Thinking methods at well-known firms such as IBM, Hewlett-Packard, and Samsung [2].

Enterprise Modeling (EM) is an area of conceptual modeling methods and tools that seeks solving organizational design problem from a holistic, multi-perspective view. One of its core principles is participatory stakeholder involvement and goal-oriented development. "For Enterprise Modeling" (4EM) [3] is an EM method that exhibits Design Thinking characteristics (cf. Sect. 3). 4EM recommends the analysis of organizational problems from several perspectives, such as vision, data, business processes, organizational structure by means of participative techniques [4]. The central question this paper tackles is, whether the 4EM as a Design Thinking method can contribute to the goals of the EAM projects. More specifically, the research questions are:

- RQ1: Can 4EM's Goal Modeling component be used as a Design Thinking method? What are the commonalities and differences?
- RQ2: Does 4EM's Goal Modeling component enhance the understanding of EAM project goals?
- RQ3: Does 4EM's Goal Modeling component help to specify the added-value of the project artefacts (tools, models etc.)?
- RQ4: Is there an association between the preparedness of the design team and the perceived usefulness of the method?
- RQ5: What are recommendations for the method application in the EAM-context?

By Goal Modeling component we refer to the 4EM language for expressing Goals Model and the guidelines for modeling. This paper has three contributions. (1) By comparing the characteristics of both approaches and demonstrating them in an industrial case, it shows that 4EM can be applied as a Design Thinking method, (2) by analyzing questionnaire data gathered from seven enterprises over the last three years, it shows the contribution of the 4EM's Goal Modeling component as a Design Thinking method in the EAM context, (3) resulting from the total of eight projects, it provides the 4EM adopters with best practices in the context of EAM.

The remainder of the paper is organized as follows. Section 2 introduces Design Thinking, EAM and 4EM. Section 3 discuss the grounds on which 4EM is defined as a Design Thinking method (cf. RQ 1) and demonstrates them in an industrial case study. Section 4 presents the research method and the survey design. Section 5 details the findings and answers RQ2, RQ3 and RQ4. Section 6 discusses the recommendations for improving 4EM (RQ5), Sect. 7 discusses limitations and future endeavors.

2 Background

2.1 Design Thinking

To fulfill the need to innovate and generate new value elements, Design Thinking gained attention both in practice and academia. Different definitions exist for the term Design Thinking (cf. [5] for an overview). Razzouk and Shute define Design Thinking as "an analytic and creative process that engages a person in opportunities to experiment, create and prototype models, gather feedback, and redesign" [6]. For Thoring and Müller Design thinking is "a specific method to solve wicked problems and to generate innovative solutions, based on a user-centered approach with multi-disciplinary teams" [7]. This methodical view is enriched in [1], which defines Design Thinking as a mindset, process and toolbox.

Due to the applicability of design methods for promoting innovation and their applicability across many areas, it enjoys a great interest both from academia and practice [8]. In Strategic Management, it is understood as an approach addressing shortcomings of analytical strategic development methods in a dynamic and fast-paced business environment [9]. In Information Systems, Design Thinking is applied to improve the early phases of an IT project to specify requirements and develop the overall concept. More specifically, it is discussed that Design Thinking enhances the understanding of the needs, problems and goals in a project [10]. Design Thinking has become an integral part in business settings. Micheli et al. point out to the growing number of implementations of Design Thinking in major organizations, including SAP, P&G, Intuit, Bank of America and Samsung [2].

2.2 Enterprise Architecture Management (EAM) and Architecture Modeling

Ahlemann et al. [11] define EAM as "a management practice that establishes, maintains and uses a coherent set of guidelines, architecture principles and governance regimes that provide direction and practical help in the design and development of an enterprise's architecture to achieve its vision and strategy." EAM is an integration of different management disciplines to create an integrated view of an enterprise. A key focus area of EAM is business and IT alignment, which requires creating a strategic dialogue between business planning and IT strategy planning. This is a process of analyzing enterprise goals together with the as-is state of technology in order to set objectives for the to-be enterprise architecture and for the transformation process. EAM is usually guided by an EA framework defining which domains, perspectives or views are to be elaborated, cf., for instance, TOGAF [12], NAF [13]. Examples of such views are vision, function, services, processes, components, capabilities, resources, products, performance.

EAM projects begin with scoping, analyzing the needs and eliciting goals for the enterprise and the project. The required benefits of this are twofold, (i) knowledge is explicated, and (ii) the stakeholder understanding of the issues relevant to the project improves. In search for methods to guiding this work, EM supports eliciting, capturing, documenting, and analyzing organizational knowledge from various perspectives chiefly among which is business goals. Other perspectives that are typically addressed by EM are business processes, concepts, services, products, rules, actors and resources. EM is also based on close cooperation with stakeholders and, several EM methods recommend a participatory way of working. Hence, EM can be seen as practicable instrument for supporting the early phases of EAM projects.

2.3 For Enterprise Modeling (4EM) and Goals Modeling

4EM [3] is a representative of the Scandinavian strand of EM methods. It shares many underlying principles of the so-called multi-perspective approaches that recommend the analysis of organizational problems from several perspectives and that the analysis process integrates the perspectives to enhance traceability and to offer a more complete view of the enterprise. 4EM meets the criteria of being an EM method as it includes a modeling language with graphical representation, a prescribed modeling process, suggested elicitation approaches, and modeling tools.

The 4EM language consists of six sub-model types, each of them focusing on a specific aspect or perspective of the enterprise - goals, business rules, concepts, business processes, actors and resources, as well as information system technical components. Due to the lack of space and for the purpose of this paper we will only present the Goals Model (GM). More about the rest of 4EM sub-models is available in [3].

The GM focuses on describing the goals of the enterprise. It describes what the enterprise and its employees want to achieve, or to avoid, and when. The Goals Model usually clarifies questions such as:

- Where should the organization be moving and what are the goals of the organization?
- What are the importance, criticality, and priorities of these goals?
- How are goals related to each other?
- Which problems hinder the achievement of goals?

The modeling constructs of the GM are goal, opportunity, problem (threat and weakness), cause, and constraint. They are linked by binary relationships of the types “supports” and “hinders.” Goals are refined by goal operationalization relationships of types AND, OR, as well as AND/OR.

The process of 4EM follows the steps described in [3] and [4]. The approach to modeling is mainly participatory. Models are developed in sessions involving a group of domain experts and one or two facilitators, sometimes referred to as method experts. In the modeling sessions, models are often documented on large plastic sheets using paper cards. An alternative way of conducting modeling sessions that has been used increasingly often during the Covid19 pandemic is to support remote participation of modelers with board-based collaborative work tools such as Miro. The modeling sessions are never arranged ad hoc. They are preceded by preparatory activities of planning, setting the goals for the session, selecting the suitable domain experts to participate etc. An important part of preparing for the modeling sessions is interviewing the domain experts expected to participate in the session. The purpose of the interviews is not to gather information to be included in models but rather to prepare the group process, to select the focus, and to prepare the driving questions for modeling sessions. Another goal is to prepare the domain experts for what will happen during the modeling session and why. The interviews also allow to clarify how the participant’s particular competency contributes to the goals of the session and of the project and what is expected from them during the session.

3 4EM as a Design Thinking Method

In the literature, characteristics, principles and processes of Design Thinking have been analyzed extensively. Based on the findings and background information from the previous section, this section discusses how the 4EM Goal Modeling component can be used as a Design Thinking method. Section 3.6 illustrates the application of 4EM's GM component in an industrial case and discusses the Design Thinking principles therein.

3.1 Participatory Aspects

Design Thinking emerged first time in 1960s in the context of participatory design, which was characterized by quick software prototype development cycles, incorporating customer feedback into the prototyping process [9]. According to Luchs Design Thinking can be construed as "*a systematic and collaborative approach for identifying and creatively solving problems*" [14]. A core principle of the 4EM method is the stakeholder participation – stakeholders are active participants of the modeling effort usually carried out as workshops supported by a modeling facilitator. In this way the stakeholders are responsible for the knowledge and decisions that are represented in the model and the modeling facilitator is responsible for the modelling method and syntactic quality of the model. This separation of responsibilities increases the participant's acceptance of the solutions and commitment to decisions. During method application in EAM projects, this played a central role, as modelling activities concerned changes to organizational units (definition of the responsible for the tools), business processes (maintenance of architecture models), and information systems (implementation of a new architecture modelling tool) to name a few.

3.2 Interdisciplinary Aspects

The participation of stakeholder groups should include a broad group from all departments, units, and organizations that are relevant to solving the problem at hand [2]. Involving stakeholders from different domains is an important aspect of tackling wicked problems. 4EM addresses the necessity of including "members from different areas or departments of the enterprise who are involved in reaching the project's objectives or have an interest in the value the project intends to create" [3]. In the EAM project context we informed the project managers on the importance of including different roles from various departments in the modelling sessions. As a result, sessions had participants with various roles, e.g., Requirements Engineer, Process Manager, Business Analyst, Software Developer, Business/Data/IT-Architect. Moreover, a pattern we observed in the public sector was the involvement of "the customers of the customers". Organizations which should benefit from "Enterprise Architecture as a Service" participated to the design activities as well.

3.3 Human-Centeredness

Design Thinking places human needs in the center. Brenner, Uebernickel and Abrell state that the Design Thinking principle “innovation is made by humans for humans” is the guiding rule for all other principles [1]. Schallmo, Williams and Lang [5] takes human needs as a starting point and perceive Design Thinking as a way to identify and implement the requirements and needs in customer-oriented solutions. Thus, human-centeredness is not reduced to participation, collaboration and interaction, but includes also the need for hearing all ideas, solving the conflicts and arranging physical spaces where the sessions may take place [15]. The Goal Modeling component of 4EM aims to create a better understanding for the concerns, limitations and priorities of business and IT professionals. To do so, the method offers guidelines and notational elements on how to resolve inconsistencies and conflicts between the goals [3]. In the EAM projects, the teams were always informed that the focus is on designing their needs and thus active engagement was an absolute prerequisite for the modeling sessions. The possibility of using physical spaces was not given mostly because the organizations were lacking such special spaces.

3.4 Personal Characteristics of a Design Thinker

4EM method as a whole including its Goal Modeling component, provides guidelines for stakeholder analysis, e.g., for selecting participants from the enterprise for the modelling sessions. Among the personal characteristics of the members of a multidisciplinary team, 4EM values enthusiastic, open-minded, and cooperative people [3]. These personal attributes have a high resonance with the characteristics of “Design Thinkers”, which are “optimism, empathy, integrative thinking, experimentation and cooperation capability” [16]. In the EAM projects, these attributes were communicated with the project managers and moderators, yet there was neither an instrument to measure the personal characteristics of the participants nor it was the intention of the designers.

3.5 Iteration and Experimentation

Modelling goals with the 4EM method is a brainstorming activity. As different stakeholder views are considered, the initial models tend to be difficult to understand and somewhat unstructured. Furthermore, new goals and/or problems can be discovered, analyzed with the team, and added to the Goals Model. This is an iterative way of problem solving and designing. The Goals Model also offer means and guidelines how to capture and document as well refine and assess various design alternatives. This is in line with the principles of Design Thinking [5, 7, 10, 17]. As Luchs explains, “Design Thinking is not intended to be a linear process, (…) [it] is best understood as an iterative approach to problem solving, rather than as a sequence of steps” [17].

3.6 Industrial Case of 4EM Goals Modeling for Design Thinking

The case company is a holding organization of a city in Northern Germany for all city-owned utility providers, public transportation, waste management companies, and public-private partnerships. After a political shift in the city, the agenda of the new local government includes the development of smart city services as one of the main priorities. The management of the case company was asked to design a portfolio of services directed to the citizens and exploiting synergies between the companies in the holding.

Preparation of the design workshop involved the management of the holding organization only and consisted of an interview and a preparation meeting. One purpose of the preparatory interview was to understand the decision-making processes and general policies of the holding organization and the companies owned by the holding. A second purpose was to investigate the digitalization strategy of the holding. Before the interview, guidelines consisting of a list of questions and aspects to explore were prepared. The interview took 45 min. and was conducted by one researcher; notes were taken. The preparation meeting aimed at defining the organizational setting (rooms, required equipment, rough schedule) and to decide on the participants for the workshop.

In the design workshop in early March 2020 participated 11 persons including the CEO of the holding, the managing directors or responsible executive functions of the different companies and their CIOs or CTOs. The workshop had three phases: goal/problem modeling with 4EM to agree on basic characteristics of "common city services" design, an idea collection and prioritization phase for such services and the initial enterprise architecture design for the implementation of common city services based on components of the different companies. The latter had the purpose of evaluating effects of common services on the individual companies and prepare decision making, in case of several competing options.

All 11 participants were informed in beforehand about the purpose of the workshop and importance of their participation. The workshop was documented in photo documentation of collected ideas and clusters, written documentation of the initial enterprise architecture model. Notes were taken to capture additional information regarding ideas and the EA. For brevity reasons, we will not go into detail of all workshop phases but focus on the results and Design Thinking aspects from Sect. 3:

- Participatory aspect: all three workshop phases encouraged contributions to the task at hand by all participants and engaged them in joint development of goal/problem cluster and relationships, common service identification and EA design.
- Interdisciplinary aspect: most of the common services could not have been designed without a combination of different company functions and application domains.
- Human-centeredness: the focus of the common services to be designed were the citizens. Discussions about customer journey and user experience design during the second part of the workshop reflect this focus.
- Personal characteristics: the selection of the participants for the workshop had to happen without deep knowledge about the personal characteristics. However, all participants turned out to be open for new ideas and experienced in collaboration.
- Iteration and experimentation: the results of all phases of the workshop did not materialize in a single modeling activity, but included revisions and refinements.

The core results included a goal/problem model developed on a plastic wall; goals in yellow, problems in blue and clusters in red (see Fig. 1, left) and an initial architecture model developed by positioning cards on a printer A0 poster and the focus on business architecture and information architecture (see Fig. 1, right).

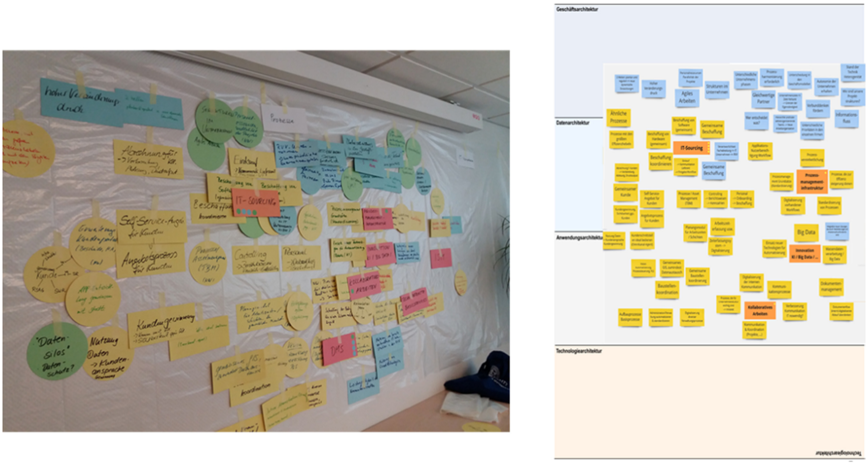


Fig. 1. Goal/problem model (left) and initial EA model (right)

4 Method

4.1 Research Design

The research design is based on a mixed method approach adopting a survey technique, which is explained in the following sections.

4.2 Design of the Questionnaire

Data for the questionnaire has been collected from seven organizations over a period of three years, where an online questionnaire was sent to design teams after the artefact creation, i.e. the Goals Model.

In line with [18] a number of measures were taken to increase the return rate of surveys. The session participants were informed with a notice that an online questionnaire will be arriving within several days after the sessions. The introduction of the questionnaire conveyed the importance of the research and the need for the person's response. An email was sent when necessary to remind the participants to complete the questionnaire before closing it.

Two response formats are selected for the questions in the survey; numerical response format with a 5-point Likert scale and open-ended questions. Table 1 summarizes the survey questions and the response formats. Reducing the risk of ambiguity of the questions and enhance their comprehension, a pretesting was performed with three domain experts in Enterprise Architecture Management. The questionnaire was designed on Google Forms.

Table 1. Questions and response formats

Survey Question (SQ)	Response format	Response rate
SQ 1. The length of the workshops for the method application was adequate	Numerical	100%
SQ 2. Receiving information about the 4EM method prior to the workshops was helpful	Numerical	81.6%
SQ 3. Analysis of the goals helped me to understand the objectives of the EAM Project	Numerical	90.3%
SQ 4. After the workshops, I have a better understanding of how the project artefact can help me	Numerical	87.1%
SQ 5. I liked/disliked following aspects in the analysis of the goals	Open ended	32.25%
SQ 6. The trainer was well-prepared	Numerical	51.2%
SQ 7. The trainer had competence in 4EM	Numerical	48.4%
SQ 8. The trainer was highly motivated	Numerical	48.4%

To minimize the bias in research, the guidelines by [19] are followed in the questionnaire. In particular, to adhere to the principles of wording, language of the questionnaire included the terms that are simplified, e.g., "Trainer" (instead of "Modeling Facilitator"), "Analysis of the goals" (instead of "Goal Modeling") and "Project Artefact" (instead of "Enterprise Architecture/EAM Tool"). To eliminate the bias towards social desirability in questions 5, 6, 7, and 8 the questionnaires were only sent out after the artefact was deemed as completed. The survey included an introduction disclosing the identity of the researcher, addressing that the anonymity of the respondent is ensured and no personal data are being collected. Furthermore, no data about the demographics of the participants and their positions were collected to avoid the risk to identify the identity of the participants.

4.3 Data Preprocessing

The questionnaire was distributed by contacting the workshop participants via email. Only in one instance, the questionnaire had to be printed out as the organization did not have an access to Google Forms. In total 31 answers from 7 different organizations, i.e. public, private, SMEs and large enterprises, were recorded. For the online questionnaires, the answers were directly stored on Google Drive and then exported as a Microsoft Excel file to conduct further analysis in SPSS.

First, the data set was checked against outliers and duplicates. The values were in the expected ranges. There was only one duplicate, most probably caused by a double submission. The exact same answer to the open ended and Likert-type questions helped to identify the duplicate data point and to exclude it from further analysis. Second, to ensure the reliability, an analysis in SPSS with Cronbach's alpha coefficient was performed with a result of .872. with N = 7 and 17 excluded items based on a listwise deletion. As noted in [20], for a rather small number of items (fewer than 10) in the scale, this value suggests that questionnaire is internally consistent.

As shown in Table 1, the questions regarding the trainer performance have rather low response rates. To remedy the missing data, the four-step process proposed by [21]. The missing data is not ignorable (Step 1) and the extent of missing data is substantial enough to warrant action (Step 2). In Step 3, the randomness of the missing data process should be examined, i.e., whether the data is missing at random (MAR), not missing at random (NMAR) or missing completely at random (MCAR). One approach to determine whether the missing data can be classified as MCAR is the Little's MCAR Test [22]. The null hypothesis for Little's test states that the data are MCAR. In our underlying data set, Little's MCAR test resulted in Chi-Square = 21.738, DF = 17, Sig. = .195, which was not enough the reject the null hypothesis that the data is missing completely at random. In Step 4, an imputation method should be selected. As no prior value taken from a similar sample was available, we either had to calculate replacement values from the valid data. Here, the "regression imputation approach" is followed, which is used to predict the missing values of a variable based on its relationship to other variables in the dataset [21]. The responses to the first question did not include missing values due to which its respective variable was used as a predictor on the missing values of the remaining variables by adopting the Expectation Maximization (EM) algorithm [23]. Section 5 analyzes the answers based on the preprocessed data set including answers from thirty participants.

5 Data Analysis

5.1 Goal Modeling Component and the Understanding of EAM Project Goals

4EM's Goals Model was used in various situations to create a common basis for the EAM project goals. One common case in the modeling workshops was the tendency to express rather abstract goals, e.g., "Realizing Business-IT Alignment". In order to further refine such goals, we engaged the participants to discuss and to make the goals of the project more traceable. The design team also defined questions that are tied to the documented goals. To name an example, the goal "Realizing Business-IT Alignment" was related to the questions "which requirements of the business units can be implemented with the current application landscape?" or "which process are potentially affected after signing with a technology vendor and who are the process owners?".

To have a better understanding whether the goal modeling component in this context was helpful, the survey included the question "SQ 2 Analysis of the goals helped me to understand the objectives of the EAM Project". The responses show a strong tendency towards a positive contribution, which are illustrated in Fig. 2 (left part). The lowest scoring response was "neither agree nor disagree", which was recorded only in one response. One participant noted in the open-ended question SQ 5 "The workshops helped me to understand the broad spectrum of the objectives [in the EAM projects]." Another participant stated "[(s)he liked the] connection between goals and questions that need to be answered to reach the goals." Furthermore, it helped to discuss with the others concerning the specific and abstract [project] goals." In line with this feedback, another participant noted that "[4EM's Goal Modeling] is an appropriate method for identifying and discussing the project goals, but a goal prioritization was missing". We discuss the second aspect in Sect. 6.

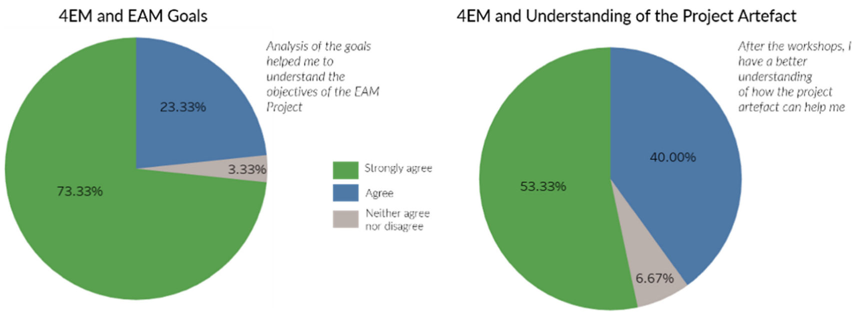


Fig. 2. 4EM and EAM Goals (left) and Added-value of the project artefacts (right)

5.2 Goal Modeling and the Added-Value of the Project Artefacts

Various artefacts can be produced or implemented in the EAM projects. Such artefacts usually consist of the maintenance process for the architectural information, metrics to measure the data quality or creation of patterns for modeling the architecture (e.g., modeling data exchange between two application interfaces). One central prerequisite to produce such artefacts is the implementation of an architecture management tool, which by itself is a project artefact. Thus, we wanted to find out whether the application of 4EM's Goal Modeling component was helpful to understand the added-value of the project artefact, i.e., the architecture management software tool to be introduced.

The survey question "SQ 4 After the workshops, I have a better understanding of how the project artefact can help me" gives insights to the research question at hand. The results shown in Fig. 2 (right part) indicate a positive attitude towards the method application. A cumulative total of over 90% agree that the application of the Goal Modeling component helped to understand the project artefact, whereas nearly 7% neither agrees or disagrees. Goal Modelling helped the team to first focus on the objectives and then relate them to the specific questions that the architecture management tool needs to answer. In the last step, best practices in architecture modeling from other projects were introduced. Inevitably, this led to the emerging of the new modeling patterns and at the same time contributed to the understanding of the architecture management tool. One respondent noted in the open-ended question SQ 5 that "best practice examples of the trainer were very useful".

5.3 Preparedness of the Design Team vs. Perceived Usefulness of the Method

RQ4 concerns the association between the preparation of the design team for the workshops (SQ2) and perceived usefulness of the method (all other SQs). Here, as the assumptions of the Pearson correlation are violated, Spearman's rank correlation coefficient was used. Figure 3 shows the correlation matrix.

Preparing the design team before the workshop, which includes informing them about the 4EM and goal modelling, is significantly positively correlated with the enhanced understanding of the project artefact (.420**). However, no significant correlation between the preparedness of the team and understanding of the project goals was observed. In fact, one respondent stated that "(…) the modelled goals were the 'usual suspects'". This might be due to the fact that goals in an EAM project are somewhat known by the architects long before the initiative starts, whereas the real challenge is their realization by means of specific project artefacts with the other teams. We found some evidence as one another respondent stated "I became aware of the abundance of the goals and the need for a [goal] hierarchy". From this perspective, we think providing the design team with the main principles of 4EM may help different roles to empathize with the architects, particularly in terms of their EAM initiatives.

		The length of the workshop for the method application was adequate	Receiving information about the 4EM method prior to the workshop was helpful	Analysis of the goals helped me to understand the objectives of the EAM Project	After the workshops, I have a better understanding of how the project artefact can help me	The trainer was well-prepared	The trainer had competence in 4EM	The trainer was highly motivated
The length of the workshop for the method application was adequate	Correlation Coefficient	1.000	.476**	.164	.447**	.229	.495**	.363*
	Sig. (1-tailed)	.	.004	.193	.007	.112	.003	.024
	N	30	30	30	30	30	30	30
Receiving information about the 4EM method prior to the workshop was helpful	Correlation Coefficient	.476**	1.000	.238	.420*	.363*	.496**	.370*
	Sig. (1-tailed)	.004	.	.103	.010	.024	.003	.022
	N	30	30	30	30	30	30	30
Analysis of the goals helped me to understand the objectives of the EAM Project	Correlation Coefficient	.164	.238	1.000	.434**	.292	.283	-.022
	Sig. (1-tailed)	.193	.103	.	.008	.058	.065	.454
	N	30	30	30	30	30	30	30
After the workshops, I have a better understanding of how the project artefact can help me	Correlation Coefficient	.447**	.420*	.434**	1.000	.339*	.406*	.269
	Sig. (1-tailed)	.007	.010	.008	.	.033	.013	.076
	N	30	30	30	30	30	30	30
The trainer was well-prepared	Correlation Coefficient	.229	.363*	.292	.339*	1.000	.557**	.473**
	Sig. (1-tailed)	.112	.024	.058	.033	.	.001	.004
	N	30	30	30	30	30	30	30
The trainer had competence in 4EM	Correlation Coefficient	.495**	.496**	.283	.406*	.557**	1.000	.523**
	Sig. (1-tailed)	.003	.003	.065	.013	.001	.	.002
	N	30	30	30	30	30	30	30
The trainer was highly motivated	Correlation Coefficient	.363*	.370*	-.022	.269	.473**	.523**	1.000
	Sig. (1-tailed)	.024	.022	.454	.076	.004	.002	.
	N	30	30	30	30	30	30	30

Fig. 3. Correlations matrix including seven variables (**Correlation is significant at the 0.01 level, *Correlation is significant at the 0.05 level.)

Being informed about 4EM and goal modeling is significantly positively correlated with a number of other variables, including a positive attitude towards the "capabilities of the method expert" (e.g., adequacy of the workshop length, competence of the trainer) as well as towards understanding the added-value of the project artefact. Concerning the capabilities of the method expert, one respondent appreciated the "excellent theoretical background delivered by the trainer" and another respondent liked the active involvement of all participants. One respondent noted that "although the method is useful, length of the workshop was short and the number of participants was low for a 'real' goal analysis".

Understanding the added-value of the project artefact is significantly positively correlated with understanding the objectives of the EAM project (.434**). Consequently, without claiming causality, one can expect that a specific project artefact may find more acceptance in a context where the goals are explicitly communicated and modeled.

6 Recommendations for Method Application

Based on the use cases and workshops, this section briefly discusses the recommendations for the method application in the EAM context.

Encourage the Adoption of Goal Models as a Promotion Tool. EAM is a holistic management discipline, any successful EAM project uses the data available in different departments of an organization. EAM project teams used goal models to promote their initiatives and to attract the relevant stakeholders. By integrating them to the project as early as possible, the models raised the chances of a potential contribution. Also, the goal models were presented organization-wide and in some cases even to the customers as a "progress report".

Prioritize Goals. As our observations stem explicitly from the EAM projects, we experienced that goal models were treated to a certain extent as "project roadmaps". The goals were either mapped to the project milestones or categorized into project phases, which made a goal prioritization necessary. For this, we developed a rather pragmatic approach, which in particular is in line with the "human-centeredness" principle of Design Thinking. After the finalization of the goal model, the design team gathers and votes for the goals one by one individually. For this, the team could select one from the three possibilities; a goal can be highly important (3), important (2) and less important (1). Then, for each goal the mode values are calculated. Goals with the higher mode values might be the candidates for a higher prioritization. Depending on the skewness of the vote distribution, average or median can be also used as a supporting statistic, in case there are too many goals that seem to be highly important.

Develop a Question Catalogue. When it comes to the project goals, EAM projects in a way pursue similar objectives (cf. Sect. 5.1). One example is "Recognizing and eliminating the redundancies", i.e., there was no team which did not mention this goal in their projects. To further specify the meaning behind the more abstract goals, we used a simple approach, which concerns expressing at least one question connected to the respective goal. In the abovementioned goal, an exemplary question might be "which (business) functions are supported/implemented by more than one department/application?". There have been two advantages of this approach. First and foremost, the design team could connect to the respective goal easily because it was more specific thanks to the question. Second, besides the goal model, the team had a question catalogue at the end of the project, which was used to create user-specific reports and dashboards with the selected EAM tool.

Use the Flexibility to Extend the Syntax. Resulting from the observations above, it is recommended to allow for the introduction of additional shapes and the use of different colors for presenting goals in a model. For example, goals with higher priority can be represented with a single dot on the right corner. Similarly, letters in circles that are attached to the goals may point out to the architecture dimension that are concerned with the respective goal (see Fig. 1, right part). Furthermore, different colors can be used to convey information about the goal fulfillment. In this context, the flexibility offered by 4EM should be adopted to encourage the creativity of the design team.

7 Summary, Limitations and Future Work

This study demonstrates how goal modeling using 4EM's GM can contribute as a Design Thinking method to set up EAM projects.

Certain limitations apply to our findings, which stem form an industrial use case and a survey performed with seven organizations. In a Design Thinking context, teams need to be heterogenous in various dimensions to be successful [4]. This heterogeneity is related to "a good mix of personalities from different countries and cultures". The method application in the EAM projects did not pay attention to this aspect.

Another Design Thinking principle considers the personal characteristics (see Sect. 3.4). No instruments have been used to measure the characteristics, as it would extend the boundaries of this study. On the other hand, missing information about the personal characteristics of the Design Thinker makes it difficult to establish the representativeness of the sample and to generalize the findings, because those participating to the sessions and responding to the survey may not at all represent the population they are supposed to [19].

This study is not cross-sectional as the survey data were gathered in three years period. On the other hand, it is not a "longitudinal study" in a sense that we studied the behavior before and after a change. The three years period has given us enough time to collect more data in different settings. Possible bias stemming from the time-related aspects is another limitation of this work. Furthermore, we observed a slight decrease in Cronbach's alpha after the regression imputation from .872 -> .774 with an inter-item correlation of .382, which is a good indicator of the questionnaire's the internal consistency [19].

A higher response rate for SQ 5. could have allowed us to gather more qualitative data. Future work aims to reduce this gap by applying the method in larger industrial use cases and to help EM practitioners by extending the best practices for the use of goal modeling in EAM projects.

References

1. Brenner, W., Uebernickel, F., Abrell, T.: Design thinking as mindset, process, and toolbox. In: Brenner, W., Uebernickel, F. (eds.) Design Thinking for Innovation, pp. 3–21. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-26100-3_1
2. Micheli, P., Wilner, S.J.S., Bhatti, S.H., et al.: Doing design thinking: conceptual review, synthesis, and research agenda. J. Prod. Innov. Manag. **36**, 124–148 (2019). https://doi.org/10.1111/jpim.12466
3. Sandkuhl, K., Stirna, J., Persson, A., Wißotzki, M.: Enterprise Modeling: Tackling Business Challenges with the 4EM Method. The Enterprise Engineering Series, 1st edn. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43725-4

4. Stirna, J., Persson, A.: Enterprise Modeling: Facilitating the Process and the People. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94857-7
5. Schallmo, D., Williams, C., Lang, K.: An integrated design thinking approach –literature review, basic principles and roadmap for design thinking. In: Bitran, I. (eds.) Proceedings of the ISPIM Innovation Conference 2018: Innovation, the name of the game conference, The International Society for Professional Innovation Management Ltd., UK (2018)
6. Razzouk, R., Shute, V.: What is design thinking and why is it important? Rev. Educ. Res. **82**, 330–348 (2012). https://doi.org/10.3102/0034654312457429
7. Thoring, K., Mueller, R.: Understanding design thinking: a process model based on method engineering. Paper Presented at 13th International Conference on Engineering and Product Design Education, E&PDE 2011, London, United Kingdom (2011)
8. Chasanidou, D., Gasparini, A.A., Lee, E.: Design thinking methods and tools for innovation. In: Marcus, A. (ed.) DUXU 2015. LNCS, vol. 9186, pp. 12–23. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-20886-2_2
9. Diderich, C.: Design Thinking for Strategy: Innovating Towards Competitive Advantage. Management for Professionals. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-25875-7
10. Dolata, M., Schwabe, G.: Design thinking in IS research projects. In: Brenner, W., Uebernickel, F. (eds.) Design Thinking for Innovation, pp. 67–83. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-26100-3_5
11. Ahlemann, F., Stettiner, E., Messerschmidt, M., Legner, C. (eds.): Strategic Enterprise Architecture Management: Challenges, Best Practices, and Future Developments. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24223-6. ISBN 978-3-642-24223-6
12. The Open Group: The TOGAF® Standard, Version 9.2 (2018)
13. NATO: NATO Architecture Framework v.4 (2018). https://www.nato.int/nato_static_fl2014/assets/pdf/pdf_2018_08/20180801_180801-ac322-d_2018_0002_naf_final.pdf
14. Luchs, M., Griffin, A., Swan, S. (eds.): Design Thinking: New Product Development Essentials from the PDMA. Wiley, Hoboken (2016)
15. IDEO LLC: Design Thinking for Educators Toolkit, 1st edn. (2012)
16. Plattner, H., Meinel, C., Weinberg, U.: Design thinking: Innovation lernen - Ideenwelten öffnen, Nachdr. mi-Wirtschaftsbuch, FinanzBuch, München (2009)
17. Luchs, M.G.: A brief introduction to design thinking. In: Luchs, M., Griffin, A., Swan, S. (eds.) Design Thinking: New Product Development Essentials from the PDMA, pp. 1–12. John Wiley & Sons Inc, Hoboken, New Jersey (2016)
18. Salant, P., Dillman, D.A.: How to Conduct Your Own Survey. Wiley, New York (1994)
19. Sekaran, U., Bougie, R.: Research Methods for Business: A Skill-Building Approach, 7th edn. Wiley, Chichester (2016)
20. Pallant, J.: SPSS Survival Manual: A Step by Step Guide to Data Analysis Using IBM SPSS, 7th edn. Routledge, London (2020)
21. Hair, J.F., Black, W.C., Babin, B.J., et al.: Multivariate Data Analysis, Eighth edition. Cengage, Andover (2019)
22. Little, R.J.A., Rubin, D.B.: Statistical Analysis with Missing Data, 3rd edn. Wiley Series in Probability and Statistics. Wiley, Hoboken (2019)
23. Enders, C.K.: Applied Missing Data Analysis. Methodology in the Social Sciences. Guilford, New York (2010)

Handling Models and Modeling Methods (EMMSAD 2021)

Formalizing Conceptual Modeling Methods with MetaMorph

Victoria Döller(✉) and Dimitris Karagiannis

Research Group Knowledge Engineering, University of Vienna, Vienna, Austria
{victoria.doeller,dimitris.karagiannis}@univie.ac.at

Abstract. Models evolved from mere pictures supporting human understanding to sophisticated knowledge structures processable by machines. This entails an inevitable need for computer-understandable models and languages and causes formalization to be a crucial part in the lifecycle of a modeling method. An appropriate formalism must be a means for providing a unique, unambiguous but implementation-independent way of specifying arbitrary modeling languages and for this purpose must be generic and open to capture any domain and any functionality. In this paper we give a pervasive description of the formalism MetaMorph based on predicate logic – an approach fulfilling these requirements. This is done with an extensive proof-of-concept case illustrating the application of the formalism concept by concept. For the case study we use the modeling language ProVis from the domain of stochastic education. The language ProVis comprises only few objects and relation types but with high interconnection and therefore appears as a interesting specimen for formalization and showing the feasibility of the demonstrated approach.

Keywords: Conceptual modeling · Agile metamodeling · Modeling method · Formal language · Predicate logic

1 Formalization in Conceptual Modeling

Conceptual modeling methods are an established tool for representing complex information in a clear and focused way. Once meant as a means for understanding and communication between humans [19], the field of employment of models was considerably broadened meanwhile. Models are nowadays processable, exploitable knowledge structures [3] and provide diverse functionality such as simulation, transformation, or reasoning, going beyond a mere visual presentation of information [1]. This amplifies the value of models and implies an imminent need for technical support of model processing and hence, an inevitable requirement for formalized models and modeling languages.

Besides the established standardized languages such as the Unified Modeling Language (UML) and the Business Process Model and Notation (BPMN) language modeling has proven beneficial in a variety of domains such as mechanical engineering or enterprise modeling. Formalization is especially important in the

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 245–261, 2021.
https://doi.org/10.1007/978-3-030-79186-5_16

light of the emergent practice of agile development of these domain-specific modeling languages (DSML) [10, 14]. Domain-specific languages rely on a fast evolvement and are frequently based on standardized modeling languages. Therefore, it is essential to establish a generic formal foundation of conceptual modeling to enable the evolvement, linking, and extension of existing languages. Only with a common practice of how to specify modeling languages completely and unambiguously they become comparable, reusable, and modularizable.

A process model for DSML development is proposed in the Agile Modeling Method Engineering (AMME) lifecylce [14], see Fig. 1. The need for formal, computer-processable modeling languages is reflected in the AMME lifecylce by stating formalization as an integral phase in the development of a modeling method. This phase targets the refinement and precise specification of a modeling language using an appropriate formalism to support implementations across various metamodeling platforms. A common way of formalizing modeling methods allows for a platform-independent yet computer-processable specification of these languages. Consequently, we open the opportunity to develop platform-specific translators to transform the single-source of platform-independent specification to platform-specific code. This is one step into the direction of automating the AMME lifecycle.

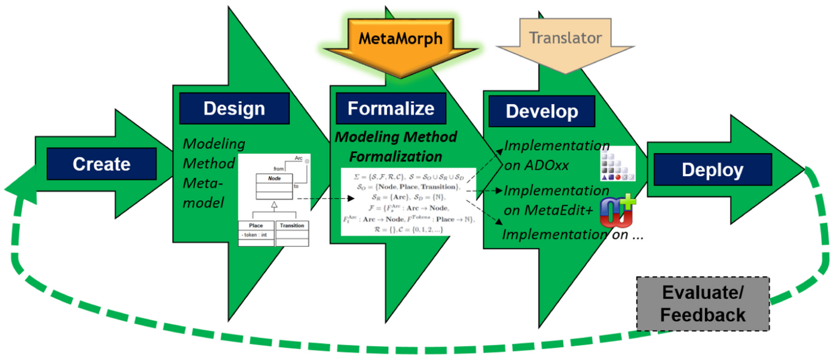


Fig. 1. The AMME lifecycle of modeling methods [14]

A formalism bridging the gap from method design to method implementation in the AMME lifecycle must be as generic as the framework itself. This means, the formalism must not be restrictive in its domain of application or the supported functionality. It serves as a tool in the process of agile method development. Yet there is no established formalism suitable for this task.

To distinguish a formalism from a formal syntax such as provided by the Meta-Object Facility (MOF) standard, we outline the difference between both. A formalism is able to capture the underlying structure of a conceptual modeling method whereas a formal syntax system offers only a unique way of specification. This can be compared to the concept of a graph and the diverse methods for recording a graph, e.g., in a graphical manner, as adjacent matrix, etc. The underlying concept of a graph is the actual notion carrying the semantics of the structure. The different ways of specification do not provide this semantics in itself and give no means to investigate the underlying formal structure.

Surveying scientific literature for a suitable structural theory for conceptual modeling reveals a consensus: modeling languages comprise all characteristics of formal languages [4,20,21]. Formal languages as defined in mathematical logic canonically reflect the prominent characteristics of modeling languages, i.e., the linguistic character providing a vocabulary and the instantiation relation between the meta-layer and the model-layer.

Examining existing approaches for formalization we recognize that none of them meets all the requirements postulated above. A variety of approaches address the formalization of a single modeling language or a concrete functionality contradicting the requirement for generality. Languages like the MOF or the Object Constraint Language (OCL) offer a syntax to specify modeling languages but lack an underlying structural theory. Approaches such as KM3 [13] or FDMM [9] conceive models as graphs or sets, respectively. These underlying mathematical theories lack the linguistic character of a modeling language. Several ontologically driven formalisms such as [2,11] adopt logic as basis but focus on ontology- and domain-engineering, thereby treating the modeling language itself as an a-posteriori artefact. The few approaches focusing on the formalization of modeling languages with formal-language theory still leave a gap between the concept of a model as used in practice and the way it is formalized using these formalisms, i.e., as sets of statements about the model [12,18].

To accomplish all the requirements, the formalization approach MetaMorph was developed and presented in [5], and refined in a contribution currently under review [6] (invited to a special edition in the International Journal on Software and Systems Modeling (SoSyM)). This formalism comprises the core meta^{2}concepts as outlined in [17] and [20]. With MetaMorph we are able to describe conceptual modeling languages as a subclass of formal languages and approach them with established tools and methods from mathematical logic.

The research objective of the paper at hand is to give a concise introduction into MetaMorph, to demonstrate its applicability to full-fledged modeling languages rather than to demonstrative, downscaled example cases, and to summarize some lessons learned. This serves on the one hand as proof of adequacy and on the other hand as guidance for others formalizing their own modeling language.

The rest of this paper is structured as follows: In Sect. 2 we give a definition of formal modeling languages and models comprising a discussion of included meta^{2}concepts. In Sect. 3 the modeling method ProVis from the area of stochastic education is formalized step by step, illustrating all aspects of the definition. We summarize in Sect. 4 the lessons learned from all case studies conducted so far. Finally, in Sect. 5 we give an outlook to the future development and research agenda concerning the formalism MetaMorph.

2 Conceptualization of the MetaMorph Formalism

To establish a definition of formal modeling languages we first have to examine what concepts have to be included in this definition. We therefore use a survey conducted by Kern et al. [17] investigating six established metamodeling platforms for the incorporated concepts in the corresponding meta^{2}models. The approach at hand consolidates all concepts detected in at least half of the platforms: object types, relation types (binary), attributes (multi-value), inheritance (for object types, single), and a constraint language. These concepts mainly coincide with the building blocks mentioned in Olivé's book on conceptual modeling of information systems [20] and are therefore endorsed both from the viewpoint of technical realization as well as from the theoretical viewpoint. Concepts for future integration are inheritance of relations, n-ary relations, model types [17], derived types, and powertypes [20].

In the following we replicate the definition for a formal modeling language as well as for a model formulated in a concrete modeling language. This definition is based on [5] and was refined in [6]. We use typed (also called sorted) first-order predicate logic. The mathematical background can be found in textbooks on logic or mathematics for computer science, e.g. [8].

Definition 1. *A (formal) modeling language* $\mathcal{L}$ *consists of a typed signature* $\Sigma = \{\mathcal{S}, \mathcal{F}, \mathcal{R}, \mathcal{C}\}$ *and a set* $\boldsymbol{C}$ *of sentences in* $\mathcal{L}$ *for the constraints, where:*

- $\mathcal{S}$ *is a set of types, which can be further divided into three disjoint subsets* $\mathcal{S}_O$*,* $\mathcal{S}_R$*, and* $\mathcal{S}_D$ *for object types, relation types and data types;*
  - *the type set* $\mathcal{S}_O$ *is strictly partially ordered with order relation* $<_O \subset \mathcal{S}_O \times \mathcal{S}_O$ *to indicate the inheritance relation between the corresponding object types;*
  - *the type set* $\mathcal{S}_D$ *can contain simple types* T *for value domains of single value attributes, or product types* $T' = T_1 \times T_2 \times \cdots \times T_n$ *for value domains of n-ary multi-value attributes* $(n > 1)$*, where the i-th value is of type* $T_i \in \mathcal{S}_D \cup \mathcal{S}_O \cup \mathcal{S}_R$*;*
- $\mathcal{F}$ *is a set of typed function symbols such that:*
  - *for each relation type* $\boldsymbol{R}$ *in* $\mathcal{S}_R$ *there exist two function symbols* $F_s^{\boldsymbol{R}}$ *and* $F_t^{\boldsymbol{R}}$ *with domain type* $\boldsymbol{R} \in \mathcal{S}_R$ *and codomain type* $\boldsymbol{O}_s, \boldsymbol{O}_t \in \mathcal{S}_O$ *assigning the source and target object types to a relation;*
  - *for each single-value attribute* $\boldsymbol{A}$ *of an object or relation type* $\boldsymbol{T}$ *there exists a function symbol* $F^{\boldsymbol{A}}$ *with domain type* $\boldsymbol{T}$ *and codomain type an element in* $\mathcal{S}_D \cup \mathcal{S}_O \cup \mathcal{S}_R$ *assigning the simple data type or referenced object type or relation type to the attribute;*
  - *for each multi-value attribute* $\boldsymbol{A}$ *of an object or relation type* $\boldsymbol{T}$ *there exists a function symbol* $F^{\boldsymbol{A}}$ *with domain type* $\boldsymbol{T}$ *and codomain type an product type in* $\mathcal{S}_D$*;*
- $\mathcal{R}$ *is a set of typed relation symbols containing* $<_O$*;*
- $\mathcal{C}$ *is a set of typed constants to specify the possible values* c_i *of a simple type* $\boldsymbol{T} \in \mathcal{S}_D$ *of the attributes;*

– *the set $\boldsymbol{C}$ is a set of sentences in $\mathcal{L}$ constraining the possible models, also called the postulates of the language.*

Notice that relation types are defined as independent modeling elements with two function symbols to specify their source and target object instead of defining them as tuples of object element. This is motivated by the relevance assigned to relations as information carryers in models. Furthermore, this allows for attributes on relations as well as multiple relations between the same two object elements.

When defining attributes, existing language specifications often refer to basic data types and enumerations such as in programming. This attempt is generalized in the definition at hand by introducing attribute-value types. They are meant to name the whole range of possible values of an attribute. The concrete values possibly adopted by an attribute are defined as constants assigned to the corresponding attribute value type. An example for this is the value type $\mathbb{N}$ with constants $1, 2, 3, \ldots$ or the value type *gender* and constants *male, female, other*.

Constraints are an integral part of MetaMorph. This means, we do not need any additional syntax such as, e.g., OCL to formalize the restrictions but have the expressive power of predicate logic at our disposal.

When interpreting modeling languages as formal languages, we get a canonical definition for models as structures of a concrete language.

Definition 2. *A model $\mathcal{M}$ of a language $\mathcal{L}$ with typed signature $\Sigma = \{\mathcal{S}, \mathcal{F}, \mathcal{R}, \mathcal{C}\}$ is an $\mathcal{L}$-structure conforming to the language constraints $\boldsymbol{C}$, i.e. $\mathcal{M}$ consists of*

– *a universe $\mathcal{U}$ of typed elements respecting the type hierarchy, that is*
  - *for each $\boldsymbol{T}$ in $\mathcal{S}$ there exists a set $\mathcal{U}_{\boldsymbol{T}} \subset \mathcal{U}$ and $\mathcal{U} = \bigcup_{\boldsymbol{T} \in \mathcal{S}} \mathcal{U}_{\boldsymbol{T}}$;*
  - *all sets $\mathcal{U}_{\boldsymbol{T}}$ for $T \in \mathcal{S}_{\boldsymbol{O}} \cup \mathcal{S}_{\boldsymbol{R}}$ have to be pairwise disjoint except for sets $\mathcal{U}_{\boldsymbol{O}_1}$ and $\mathcal{U}_{\boldsymbol{O}_2}$ with $\boldsymbol{O}_1, \boldsymbol{O}_2 \in \mathcal{S}_{\boldsymbol{O}}$ where $\boldsymbol{O}_1 <_{\boldsymbol{O}} \boldsymbol{O}_2$. In this case $\mathcal{U}_{\boldsymbol{O}_1}$ must be a subset of $\mathcal{U}_{\boldsymbol{O}_2}$, i.e. $\mathcal{U}_{\boldsymbol{O}_1} \subseteq \mathcal{U}_{\boldsymbol{O}_2}$;*

– *an interpretation of the function symbols in $\mathcal{L}$, i.e. for each function symbol $F \in \mathcal{F}$ with domain type $\boldsymbol{T}_1 \times \ldots \times \boldsymbol{T}_n$ and codomain type $\boldsymbol{T}$ a function $f : \mathcal{U}_{\boldsymbol{T}_1} \times \ldots \times \mathcal{U}_{\boldsymbol{T}_n} \to \mathcal{U}_{\boldsymbol{T}}$;*

– *an interpretation of the relation symbols in $\mathcal{L}$, i.e. for each relation symbol $R \in \mathcal{R}$ with domain type $\boldsymbol{T}_1 \times \ldots \times \boldsymbol{T}_m$ a relation $r \subset \mathcal{U}_{\boldsymbol{T}_1} \times \ldots \times \mathcal{U}_{\boldsymbol{T}_m}$;*

– *for each simple type $\boldsymbol{T} \in \mathcal{S}_{\boldsymbol{D}}$ and constant $C \in \mathcal{C}$ of type $\boldsymbol{T}$ an interpretation $c \in \mathcal{U}_{\boldsymbol{T}}$;*

– *for each constraint ϕ in $\boldsymbol{C}$ the model $\mathcal{M}$ satisfies ϕ, i.e. $\mathcal{M} \models \phi$.*

Note that the specification of a model using the definition above is a supplementary way of representation to the usual graphical way of describing a model. An advantage of this representation is the completeness of information. This means that also attribute values are concretely specified, a detail which is often not visible in a graphical model.

To illustrate the definitions and show the feasibility of MetaMorph we illustrate its application on an extensive case study.

3 Formalizing the ProVis Modeling Language—A Proof of Concept

In this case study we will demonstrate the procedure of formalizing a modeling language concept by concept. As demonstration language we use ProVis – Probability Visualized, a language for digitizing the visualization methods tree diagrams and unit squares for stochastic education. For a detailed description of the language and the mathematical background see [7]. The metamodel of ProVis is shown in Fig. 2 using the notation of CoChaCo [15]. This language contains only a few object and relation types but each of them carries a lot of information in interdependencies and functionality. Therefore, ProVis provides an interesting specimen for formalization.

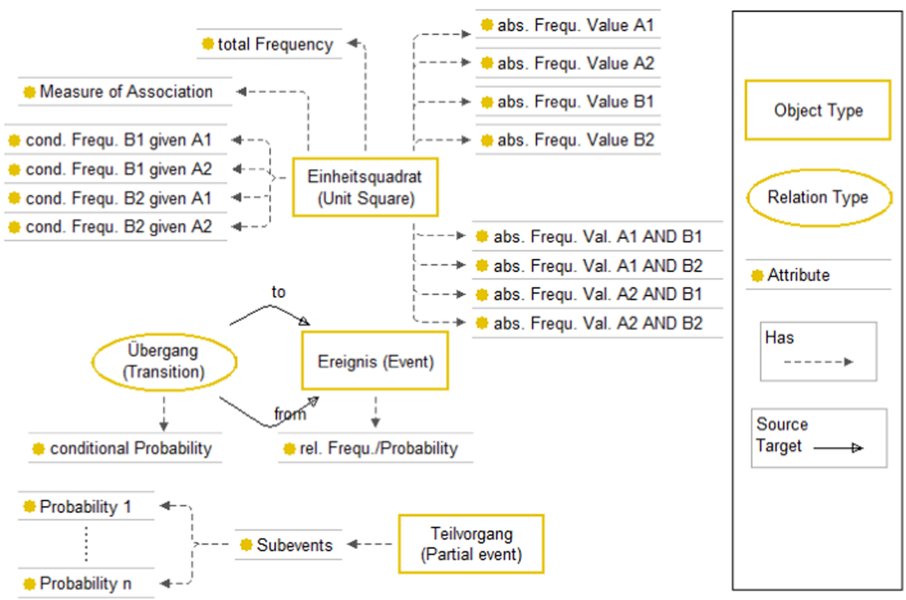


Fig. 2. The metamodel of ProVis

In the full-fledged language both visualization methods – tree diagrams and unit squares – offer to work with absolute frequencies as well as with probabilities. In the case study we restrict unit squares to absolute frequencies and tree diagrams to probabilities/relative frequencies. We do that to reduce the number of attributes, but do not loose the expressivity of the formalization.

In Fig. 3 we see examples of a unit square (a) and a tree diagram (b). A unit square is an object on its own whereas a tree diagram is a construct constituted from several events (rectangles) and transitions (arrows) between them. Figure 3 presents the number of master's and doctoral graduations at the University of

Vienna in 2019 depending on the gender[1]. This example reveals that although nearly 70% of master graduates are female (1762 female : 823 male) the rate of females finishing a doctoral program is with around 45% much lower (189 female : 229 male). If we assume, that this proportions are similar over the last few years, we can conclude that females finish their academic career after a master's degree more often than males.

3.1 Object Types

From the metamodel, we see that there are three object types **UniSquare (US)**, **Event (E)**, and **PartialEvent (PE)**. These types constitute the types in $\mathcal{S}_O$

$$\mathcal{S}_O = \{\textbf{UnitSquare}, \textbf{Event}, \textbf{PartialEvent}\}.$$

There is no inheritance between these types: $<_O = \{\}$.

3.2 Relation Types

The metamodel contains only one relation type **Transition (Tr)**

$$\mathcal{S}_R = \{\textbf{Transition}\}.$$

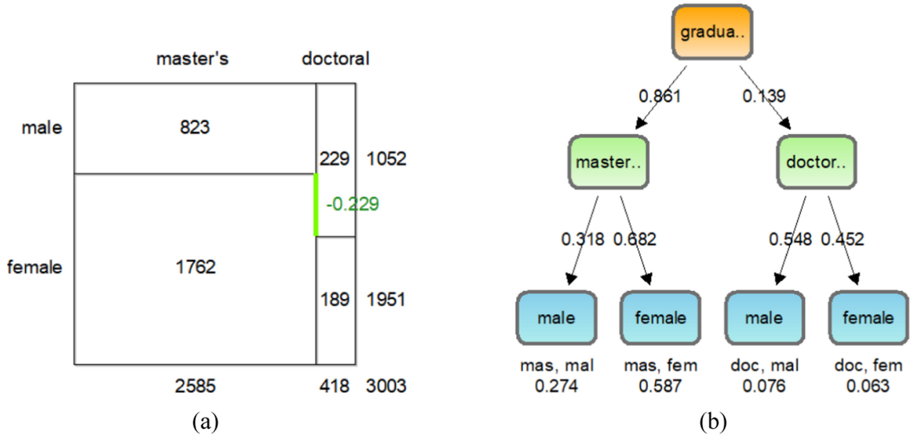


Fig. 3. Number of graduations (master's and doctoral degree) at the University of Vienna in 2019

According to Definition 1 we have to specify two function symbols F_s^{Tr} and F_t^{Tr} indicating source and target types to **Transition**. From the metamodel we see that this relation type connects elements of type **Event** with other elements of type **Event**

$$F_s^{\text{Tr}} : \textbf{Transition} \rightarrow \textbf{Event}, F_t^{\text{Tr}} : \textbf{Transition} \rightarrow \textbf{Event}.$$

[1] www.univie.ac.at/en/about-us/at-a-glance/facts-folders/.

3.3 Attributes

In the following, we will demonstrate how to define attributes and their value domains. We start with single-value attributes and then show how to use product types to define multi-value attributes.

Single-Value Attributes. A unit square is a graphical representation of a 2×2 contingency table with two characteristics A and B each of which has two values A_1, A_2, and B_1, B_2, respectively. In Fig. 3, characteristic A is the type of degree and characteristic B is the gender. A concrete combination of characteristic values is the number of male graduates with a master's degree (A_1B_1). The unit square is built up by the four frequencies, one for each possible combination of attribute values: $A_1B_1, A_1B_2, A_2B_1, A_2B_2$. Each of them points to an amount, i.e., a natural number including 0. To capture the attribute-value domain and possible attribute values in the language, we have to introduce a value type in $\mathcal{S}_D$ called $\mathbb{N}_0$. The possible values of this type, i.e., constants of type $\mathbb{N}_0$, are $0, 1, 2, 3, \ldots$ with their usual semantics.

$$\mathbb{N}_0 \in \mathcal{S}_D,\ C \supset \{0, 1, 2, ...\} \text{ of type } \mathbb{N}_0.$$

With this, we can define the four attributes via function symbols mapping the element of type Unit Square to the attribute value (attributes *abs. Frequ. Value* A_i *AND* B_j in the metamodel):

$$F^{abs}_{A_1B_1} : \mathbf{US} \rightarrow \mathbb{N}_0, F^{abs}_{A_2B_1} : \mathbf{US} \rightarrow \mathbb{N}_0,$$
$$F^{abs}_{A_1B_2} : \mathbf{US} \rightarrow \mathbb{N}_0, F^{abs}_{A_2B_2} : \mathbf{US} \rightarrow \mathbb{N}_0.$$

Also, the absolute numbers of a single value, e.g., the number of all female graduates, as well as the size of the whole cohort are of interest (attributes *abs. Frequ. Value* $A_i \ldots$ in the metamodel). We get five more attributes:

$$F^{abs}_{A_1} : \mathbf{US} \rightarrow \mathbb{N}_0, F^{abs}_{A_2} : \mathbf{US} \rightarrow \mathbb{N}_0,$$
$$F^{abs}_{B_1} : \mathbf{US} \rightarrow \mathbb{N}_0, F^{abs}_{B_2} : \mathbf{US} \rightarrow \mathbb{N}_0,$$
$$F^{abs}_{total} : \mathbf{US} \rightarrow \mathbb{N}_0.$$

For the construction of the visual appearance of a unit square it is essential to know the conditional frequency of B_1 given A_1 ($B_1|A_1$ in short), etc. (e.g., the percentage of females under all doctoral graduates). This evaluates to a value between 0 and 1. We specify a new attribute value type **Percentage** and add all real numbers in $[0; 1]$ to the constants:

$$\mathbf{Percentage} \in \mathcal{S}_D,\ C \supset [0; 1] \text{ of type } \mathbf{Percentage}.$$

Whit this type we are able to define the required attributes:

$$F^{cond}_{B_1|A_1} : \mathbf{US} \to \mathbf{Percentage}, F^{cond}_{B_2|A_1} : \mathbf{US} \to \mathbf{Percentage},$$
$$F^{cond}_{B_1|A_2} : \mathbf{US} \to \mathbf{Percentage}, F^{cond}_{B_2|A_2} : \mathbf{US} \to \mathbf{Percentage}.$$

Another important factor is the measure of association (attribute *Measure of Association* in the metamodel). This value gives a measure for the dependency of the two characteristics A and B and is calculated by subtracting the conditional frequency of $B_1|A_2$ from $B_1|A_1$. Again, we need a new attribute-value type. We call it $\mathbb{R}$ and give it the usual semantics known from math classes:

$$\mathbb{R} \in \mathcal{S}_D, \mathcal{C} \supset (-\infty; \infty) \text{ of type } \mathbb{R},$$
$$F^{ass} : \mathbf{US} \to \mathbb{R}.$$

Trees are built up from their single events, depicted by coloured rectangles and transitions visualized as arrows between them. The types **Event** and **Transition** both contain only one attribute representing their relative frequency/probability and conditional probability, respectively (in the metamodel attributes *conditional Probability* and *rel. Frequ./Probability*). The corresponding function symbols look as follows:

$$F^{prob}_{E} : \mathbf{E} \to \mathbf{Percentage}, F^{prob}_{Tr} : \mathbf{Tr} \to \mathbf{Percentage}.$$

We do not include free text attributes in the formalized language. This is done because the automatic processing of a model requires a machine to understand the semantics of each value. This is only the case when we neatly define the possible values in the attribute-value type. Natural language descriptions do not qualify for semantic processing. Nevertheless, to cover usual practises of method engineers the future research agenda of the proposed formalism MetaMorph also includes free text variables.

Multi-value Attributes. The last object type for which we have to define the attributes is the type **PartialEvent**. A partial event is a single stage event, for example the first throw when you roll a dice three times. It can contain several subevents, each of which has a probability (attribute *Subevents* in the metamodel). The partial event of rolling a dice includes following six subevents: *rolling a 1*, *rolling a 2*, ..., *rolling a 6*. Each of these subevents has probability 1/6. This is an example of a multi-value attribute, as a tuple of several values is mapped to the modeling element. For this, we introduce new product types $(\mathbf{Percentage})^i$ for an i-valued attribute. A partial event may contain arbitrary many subevents, so we have to build the union of all of these product types and introduce a new attribute value type **Subevents** for it:

$$\mathbf{Subevents} = \bigcup_i \mathbf{Percentage}^i.$$

This means that the corresponding attribute points to an (arbitrary dimensioned) array of probabilities, i.e., the mentioned subevent of rolling a single dice corresponds to the 6-dimensional tuple $(\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$. The attribute results in a function symbol

$$F_{PE}^{sub} : \mathbf{PartialEvent} \rightarrow \mathbf{Subevents}.$$

3.4 Additional Symbols

Often we need additional symbols to be able to formulate constraints on our modeling languages. In this case study, such symbols are the usual addition, subtraction, division (all function symbols), and order relation on natural numbers:

$$+_{\mathbb{N}_0} : \mathbb{N}_0 \times \mathbb{N}_0 \rightarrow \mathbb{N}_0 \text{ in } \mathcal{F}, \qquad -_{\mathbb{N}_0} : \mathbb{N}_0 \times \mathbb{N}_0 \rightarrow \mathbb{R} \text{ in } \mathcal{F},$$

$$/_{\mathbb{N}_0} : \mathbb{N}_0 \times \mathbb{N}_0 \rightarrow \mathbb{R} \text{ in } \mathcal{F}, \qquad <_{\mathbb{N}_0} \subset \mathbb{N}_0 \times \mathbb{N}_0 \text{ in } \mathcal{R}.$$

Furthermore, we want a construct for paths on trees, i.e., directed sequences of connected events:

$$Path \subset \mathbf{Event} \times \mathbf{Event}.$$

The usual behaviour of paths will be specified via constraints in the next section.

3.5 Constraints

Modeling languages carry constraints to ensure that only semantically meaningful models are created. These constraints are mostly not depicted in graphical metamodels and have to be added to a language specification by other means. In MetaMorph, they are an integral part of language definition and can be defined in logical sentences.

We split constraints in two categories: restrictive ones that confine the creation, connection, and attribution of models, and constructive ones that impose some dependencies between elements and values. The latter one can be understood as automatically derived values, or instances.

Constructive Constraints: On tree diagrams we have three constraints defining how the *Path* relation is deduced from connected elements. To be exact it is the transitive closure of the relation type **Transition**. A fourth constraint is given by the so called path rule, or multiplication rule. This rule states how the probability of compound events can be calculated from the preceding transition and event in the tree diagram. To ease the differentiation between language definition and language instantiation, we use capital letters for the symbols of the language signature and lowercase letters for interpretations of the symbols on model level.

$$\forall t \in \mathbf{Transition}, x, y \in \mathbf{Event}(f_s^{\mathrm{Tr}}(t) = x \wedge f_t^{\mathrm{Tr}}(t) = y \implies Path(x, y)) \quad (1)$$

$$\forall x, y, z \in \mathbf{Event}(Path(x, y) \wedge Path(y, z) \implies Path(x, z)) \quad (2)$$

$$\forall x \in \mathbf{E}, t \in \mathbf{Tr}(f_t^{\mathrm{Tr}}(t) = x \implies f_E^{prob}(x) = f_{Tr}^{prob}(t) \cdot f_E^{prob}(f_s^{\mathrm{Tr}}(t))) \quad (3)$$

On unit squares we obtain a couple of calculation rules. The absolute frequency of characteristic values A_1, A_2, B_1, B_2 and the total frequency of the whole cohort are fully dependent on the frequencies of value combinations.

$$\forall x \in \mathbf{US}(f^{abs}_{A_1}(x) = f^{abs}_{A_1B_1}(x) + f^{abs}_{A_1B_2}(x)) \tag{4}$$

$$\forall x \in \mathbf{US}(f^{abs}_{A_2}(x) = f^{abs}_{A_2B_1}(x) + f^{abs}_{A_2B_2}(x)) \tag{5}$$

$$\forall x \in \mathbf{US}(f^{abs}_{B_1}(x) = f^{abs}_{A_1B_1}(x) + f^{abs}_{A_2B_1}(x)) \tag{6}$$

$$\forall x \in \mathbf{US}(f^{abs}_{B_2}(x) = f^{abs}_{A_1B_2}(x) + f^{abs}_{A_2B_2}(x)) \tag{7}$$

$$\forall x \in \mathbf{US}(f^{abs}_{total}(x) = f^{abs}_{A_1B_1}(x) + f^{abs}_{A_1B_2}(x) + f^{abs}_{A_2B_1}(x) + f^{abs}_{A_2B_2}(x)) \tag{8}$$

Four more constructive constraints are given by the conditional frequency, which also can be automatically calculated.

$$\forall x \in \mathbf{US}(f^{cond}_{B_1|A_1}(x) = f^{abs}_{A_1B_1}(x)/f^{abs}_{A_1}(x)) \tag{9}$$

$$\forall x \in \mathbf{US}(f^{cond}_{B_2|A_1}(x) = f^{abs}_{A_1B_2}(x)/f^{abs}_{A_1}(x)) \tag{10}$$

$$\forall x \in \mathbf{US}(f^{cond}_{B_1|A_2}(x) = f^{abs}_{A_2B_1}(x)/f^{abs}_{A_2}(x)) \tag{11}$$

$$\forall x \in \mathbf{US}(f^{cond}_{B_2|A_2}(x) = f^{abs}_{A_2B_2}(x)/f^{abs}_{A_2}(x)) \tag{12}$$

Finally the measure of association can be calculated.

$$\forall x \in \mathbf{US}(f^{ass}(x) = f^{cond}_{B_1|A_1}(x) - f^{cond}_{B_1|A_2}(x)) \tag{13}$$

Restrictive Constraints: The most prominent restrictive constraints are cardinality constraints. Tree diagrams allow at most one incoming relation to an element of type **Event**. This is specified in Eq. (14). Similar to that, constraints of higher cardinality can be defined. Furthermore, a tree cannot contain circles. This is specified using the Path relation (15).

$$\forall x, y \in \mathbf{Transition}(f_t^{\mathrm{Tr}}(x) = f_t^{\mathrm{Tr}}(y) \implies x = y) \tag{14}$$

$$\forall x \in \mathbf{Event}(\neg Path(x, x)) \tag{15}$$

To make sure that the *Path* relation exclusively reflects the concatenated relation elements of type **Transition**, we need a further constraint. For brevity we use the abbreviation xty for relation t of type $\mathbf{T}$, x being $\mathtt{F}_s^{\mathbf{T}}(r)$ and y being $\mathtt{F}_t^{\mathbf{T}}(r)$)

$$\forall x, y \in \mathbf{E}\ \exists z \in \mathbf{E}, t \in \mathbf{Tr}\ (Path(x, y) \implies xty \vee (xtz \wedge Path(z, y))) \tag{16}$$

3.6 Formalizing Graphical Models

Female Graduates at the University of Vienna: We formalize the example presented in Fig. 3. First of all we define the universes of elements of all types.

There is one element *grads* of type **Unit Square** and seven elements of type **Event** constituting the tree diagram. For reasons of uniqueness, we renamed the events in the last row of the tree diagram.

$$\mathcal{U}_{US} = \{grads\}, \mathcal{U}_{PE} = \{\},$$

$$\mathcal{U}_{E} = \{graduates, masters, doctors, mMasters, fMasters, mDoctors, fDoctors\}.$$

The tree furthermore contains six relation elements of type **Transition**:

$$\mathcal{U}_{Tr} = \{t_1, t_2, t_3, t_4, t_5, t_6\}.$$

For the six relation elements of type **Transition** we have to concretize the source and target values:

$$f_s^{\mathrm{Tr}}(t_1) = graduates, f_t^{\mathrm{Tr}}(t_1) = masters, f_s^{\mathrm{Tr}}(t_2) = graduates, f_t^{\mathrm{Tr}}(t_2) = doctors,$$
$$f_s^{\mathrm{Tr}}(t_3) = masters, f_t^{\mathrm{Tr}}(t_3) = mMasters, f_s^{\mathrm{Tr}}(t_4) = masters, f_t^{\mathrm{Tr}}(t_3) = fMasters,$$
$$f_s^{\mathrm{Tr}}(t_5) = doctors, f_t^{\mathrm{Tr}}(t_5) = mDoctors, f_s^{\mathrm{Tr}}(t_6) = doctors, f_t^{\mathrm{Tr}}(t_6) = fDoctors.$$

The base attribute values needed in the tree diagram in Fig. 3 right are the following:

$$f_E^{prob}(graduates) = 1, f_{Tr}^{prob}(t_1) = 0.861, f_{Tr}^{prob}(t_2) = 0.139,$$

$$f_{Tr}^{prob}(t_3) = 0.318, f_{Tr}^{prob}(t_4) = 0.682, f_{Tr}^{prob}(t_5) = 0.548, f_{Tr}^{prob}(t_6) = 0.452$$

With constraint (3) from Sect. 3.5 we derive:

$$f_E^{prob}(masters) = 1 \cdot 0.861 = 0.861, f_E^{prob}(doctors) = 1 \cdot 0.139 = 0.139,$$

$$f_E^{prob}(mMasters) = 0.861 \cdot 0.318 \approx 0.274, f_E^{prob}(fMasters) = 0.861 \cdot 0.682 \approx 0.587,$$

$$f_E^{prob}(mDoctors) = 0.139 \cdot 0.548 \approx 0.076, f_E^{prob}(fDoctors) = 0.139 \cdot 0.452 \approx 0.063.$$

The four base attribute values of the given unit square look as follows (in this example A_1=# master's degrees, A_2=# doctoral degrees, B_1=# male graduates, B_2=# female graduates):

$$f_{A_1B_1}^{abs}(grads) = 823, f_{A_2B_1}^{abs}(grads) = 229,$$

$$f_{A_1B_2}^{abs}(grads) = 1762, f_{A_2B_2}^{abs}(grads) = 189.$$

With the constructive constraints from Sect. 3.5 we derive following values automatically:

$$f_{A_1}^{abs}(grads) = 823 + 1762 = 2585, f_{A_2}^{abs}(grads) = 229 + 189 = 418,$$

$$f_{B_1}^{abs}(grads) = 823 + 229 = 1052, f_{B_2}^{abs}(grads) = 1762 + 189 = 1951,$$

$$f_{total}^{abs}(grads) = 823 + 229 + 1762 + 189 = 3003.$$

For the conditional probabilities $B_i|A_j$ we calculate with constraint (9)–(13):

$$f^{cond}_{B_1|A_1}(grads) = 823/2585 = 0.318, f^{cond}_{B_1|A_2}(grads) = 229/418 = 0.548,$$
$$f^{cond}_{B_2|A_1}(grads) = 1762/2585 = 0.682, f^{cond}_{B_2|A_2}(grads) = 159/418 = 0.452,$$
$$f^{ass}(grads) = 0.318 - 0.548 = -0.23.$$

This concludes the full formalization of the example from Fig. 3.

Distribution of Blood Groups in Austria: We also want to demonstrate the formalization of partial events. This is done using the example of the distribution of blood groups and rhesus factor in Austria[2] shown in Fig. 4. There, we see two examples of partial events representing the probabilities of having blood group 0, A, B, or AB and the probabilities of being of positive-, or negative rhesus factor.

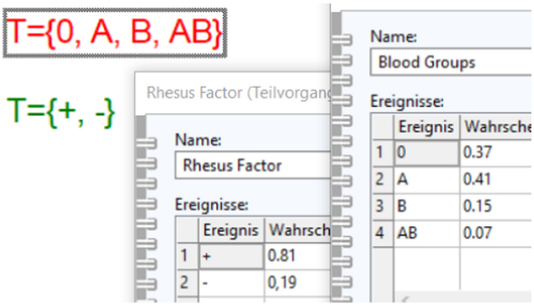


Fig. 4. Partial events concerning blood groups and rhesus factor

The universe of partial events therefore is

$$\mathcal{U}_{PE} = \{bloodGroups, rhesusFactor\}.$$

The multi-value attribute F^{sub}_{PE} : **PartialEvent** $\longrightarrow$ **Subevents** = $\bigcup_i$ **Percentage**^{i} maps the two elements to the two tuples:

$$f^{sub}_{PE}(bloodGroups) = (0.37, 0.41, 0.15, 0.07),$$

$$f^{sub}_{PE}(rhesusFactor) = (0.81, 0.19)$$

From these details and the fact, that the two partial events are independent from each other, we can construct a tree diagram depicted in Fig. 5. With the postulated constraints of the path rule 3, we can furthermore calculate the probabilities of the compound event, having blood group A with positive rhesus factor etc., revealing that the most frequent blood combination in Austria is A+ (33.2%) whereas the rarest one is AB– (1.3%).

2 www.gesundheit.gv.at/labor/laborwerte/blutgruppenserologie-transfusion/blutgruppenuntersuchung1-kh.

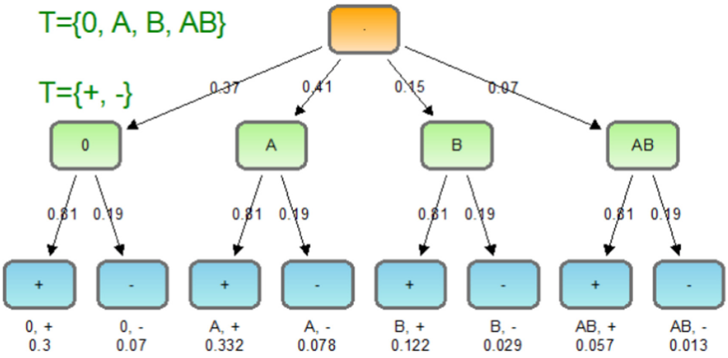


Fig. 5. Distribution of blood groups and rhesus factor in Austria

4 Lessons Learned

From the extensive case study presented in this paper and from previous formalization projects we want to record three recurring observations.

The most salient observation is that the expressiveness and complexity of a language is captured in the concepts commonly considered as subsidiary to object types, namely relation types, attributes and constraints, rather than in the object types themselves. Although object types are the most prominent element in a metamodel, the relation types and the "hidden" constructs of attributes and constraints do a better job of capturing processable semantics of models. This is reflected in the higher effort of formalizing these concepts according to Definition 1 and is underpinned by a number of case studies on expressive modeling languages with few object types but intricate constraints, e.g. Petri Nets [5] and ProVis, or complex multi-value attributes, e.g. ProVis and UML class diagrams [6].

Furthermore we observe that constraints can be deployed for two opposing purposes. The first one is a restrictive one, i.e., limiting the set of valid model constructs. The second one is a constructive one, i.e., enforcing the existence of additional modeling elements or determining attribute values. Examples for the first type are cardinality constraints, or the restrictions for tree-like structures without circles. Examples for the second type are rules for attribute values like the numerous dependencies of frequencies in unit squares or the execution time of whole processes depending on the single tasks. In this sense the constructive usage can be seen as a generating functionality of a language.

The final observation concerns the interrelationship of the graphical and the formalized representation of a model. Of course the spatial appearance is a crucial point for human comprehension in conceptual modeling but mostly omitted in the formalized representation of a model. Nevertheless, spatial arrangement and formalization can have a pertinent mutual impact. In the case study on ProVis, the graphical manifestation of a unit square is highly dependent on the attributes of conditional probability determined by the numerous constructive constraints.

In contrast to that, in UML sequence diagrams the vertical ordering of messages from a lifeline determine anteriority and posteriority of messages which definitely has to be captured in a formalized model.

5 Conclusion and Outlook

The paper at hand gives a pervasive description of our formalism MetaMorph proposed in [5] and [6] and illustrates its adequacy by applying it to an extensive case study. For the case study, we chose the modeling method ProVis – *Probability visualized* – from the domain of stochastic education. The distinguishing feature of this language is the low amount of object and relation types but high interconnectivity of concepts and their attributes. This makes it a highly interesting specimen for formalization. Conducting the case study, the formalization of all language concepts is shown step by step. Furthermore, we summarize some lessons learned from all case studies conducted so far.

The proposed formalism MetaMorph aims at closing the gap in the AMME lifecycle of modeling methods and building a bridge between the informal method design and the platform-specific method implementation. By providing a generic and complete formalism as the one proposed, we enable the development of platform-specific translators. With these translators, platform-independent formal specifications can be transformed to platform-specific code. This is one step into the direction of automatizing the AMME lifecycle of modeling-method engineering.

The future agenda of evolving MetaMorph includes additional concepts of established meta^{2}models, such as n-ary relations, or powertypes. Furthermore, the potential of logic- and formal-language theory shall be exploited for contributing concepts and methods beneficial for conceptual modeling. As an additional dimension, also mechanisms and algorithms on models shall be captured in the approach, to enable a full formalization of entire modeling methods as proposed in [16].

References

1. Bork, D., Buchmann, R.A., Karagiannis, D., Lee, M., Miron, E.T.: An open platform for modeling method conceptualization: the OMiLAB digital ecosystem. Commun. Assoc. Inf. Syst. **44**(1), 673–679 (2019)
2. Brachman, R.J., Schmolze, J.G.: An overview of the KL-ONE knowledge representation system. In: Mylopolous, J., Brodie, M. (eds.) Readings in Artificial Intelligence and Databases, pp. 207–230. Morgan Kaufmann, San Francisco (1989)
3. Buchmann, R.A., Ghiran, A.M., Döller, V., Karagiannis, D.: Conceptual modeling education as a "design problem". Complex Syst. Inform. Model. Q. (21), 21–33 (2019)
4. Delcambre, L.M.L., Liddle, S.W., Pastor, O., Storey, V.C.: A reference framework for conceptual modeling. In: Trujillo, J.C., et al. (eds.) ER 2018. LNCS, vol. 11157, pp. 27–42. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00847-5_4

5. Döller, V.: M2FOL: a formal modeling language for metamodels. In: Grabis, J., Bork, D. (eds.) PoEM 2020. LNBIP, vol. 400, pp. 109–123. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-63479-7_8
6. Döller, V.: Formalizing the four-layer metamodeling stack - potential and benefits. Preprint http://arxiv.org/abs/2105.01038 (2021)
7. Döller, V., Götz, S.: Tree diagrams and unit squares 4.0 - digitizing stochastic classes with the didactic modeling tool ProVis. In: Karagiannis, D., et al. (eds.) Domain-Specific Conceptual Modeling: Concepts, Methods and Tools, vol. II. Springer (2021, in press)
8. Enderton, H.B.: A Mathematical Introduction to Logic, 2nd edn. Harcourt/Academic Press, San Diego (2001)
9. Fill, H.G., Redmond, T., Karagiannis, D.: FDMM: a formalism for describing ADOxx meta models and models. In: ICEIS 2012 - Proceedings of the 14th International Conference on Enterprise Information Systems, vol. 3, pp. 133–144 (2012)
10. Frank, U.: Domain-specific modeling languages: Requirements analysis and design guidelines. In: Reinhartz-Berger, I., Sturm, A., Clark, T., Cohen, S., Bettin, J. (eds.) Domain Engineering: Product Lines, Languages, and Conceptual Models, pp. 133–157. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36654-3_6
11. Guizzardi, G.: On ontology, ontologies, conceptualizations, modeling languages, and (meta)models. In: Vasilecas, O., Eder, J., Caplinskas, A. (eds.) Selected Papers from the Seventh International Baltic Conference, DB&IS, pp. 18–39. IOS Press (2007)
12. Jackson, E., Sztipanovits, J.: Towards a formal foundation for domain specific modeling languages. In: Proceedings of the 6th ACM & IEEE International Conference on Embedded Software, EMSOFT 2006, pp. 53–62. ACM, New York (2006)
13. Jouault, F., Bézivin, J.: KM3: a DSL for metamodel specification. In: Gorrieri, R., Wehrheim, H. (eds.) FMOODS 2006. LNCS, vol. 4037, pp. 171–185. Springer, Heidelberg (2006). https://doi.org/10.1007/11768869_14
14. Karagiannis, D.: Conceptual modelling methods: the AMME agile engineering approach. In: Silaghi, G.C., Buchmann, R.A., Boja, C. (eds.) IE 2016. LNBIP, vol. 273, pp. 3–19. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73459-0_1
15. Karagiannis, D., Burzynski, P., Utz, W., Buchmann, R.A.: A metamodeling approach to support the engineering of modeling method requirements. In: 27th IEEE International Requirements Engineering Conference, pp. 199–210 (2019)
16. Karagiannis, D., Kühn, H.: Metamodelling platforms. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.) EC-Web 2002. LNCS, vol. 2455, p. 182. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45705-4_19
17. Kern, H., Hummel, A., Kühne, S.: Towards a comparative analysis of meta-metamodels. In: Proceedings of the Compilation of the Co-Located Workshops on DSM 2011, TMC 2011, AGERE! 2011, AOOPES 2011, NEAT 2011, & VMIL 2011, pp. 7–12. ACM (2011)
18. Koubarakis, M., et al.: A retrospective on Telos as a metamodeling language for requirements engineering. Requirements Eng. **26**, 1–23 (2021). https://doi.org/10.1007/s00766-020-00329-x
19. Mylopoulos, J.: Conceptual modelling and Telos. In: Conceptual Modelling, Databases, and CASE: An Integrated View of Information System Development, pp. 49–68 (1992)

20. Olivé, A.: Conceptual Modeling of Information Systems. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-39390-0
21. Thalheim, B.: The theory of conceptual models, the theory of conceptual modelling and foundations of conceptual modelling. In: Embley, D., Thalheim, B. (eds.) Handbook of Conceptual Modeling, pp. 543–577. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-15865-0_17

Situation-Specific Business Model Development Methods for Mobile App Developers

Sebastian Gottschalk(✉), Enes Yigitbas, Alexander Nowosad, and Gregor Engels

Software Innovation Lab, Paderborn University, Paderborn, Germany
{sebastian.gottschalk,enes.yigitbas,gregor.engels}@uni-paderborn.de, anowosad@mail.uni-paderborn.de

Abstract. The development of effective business models is an essential task in highly competitive markets like mobile ecosystems. Existing development methods for these business models do not specifically bring into focus that the development process profoundly depends on the situation (e.g., market size, regulations) of the mobile app developer. Here, a mismatch between method and situation can lead to poor resource management and longer development cycles. In software engineering, software projects use situational method engineering to configure a development method out of a method repository based on the project situation. Analogously, we support creating situation-specific business model development methods with a method base and new user roles. Here, the method engineer obtains the domain expert's knowledge and stores it in the method base as elements, building blocks, and patterns. We derive the expert knowledge from a grey literature review on mobile development processes. After this, the method engineer constructs the development method based on the described situation of the business developer. We provide an open-source tool and evaluate it by constructing a local event platform's business model development method.

Keywords: Business model development · Situational Method Engineering · Mobile app · Business model development tools

1 Introduction

The development of effective business models, defined by Osterwalder et al. as "the rationale of how the organization creates, delivers, and captures value" [23], is an essential task for a company to stay competitive. This is one of the results of the GE Innovation Barometer 2018 [14], a study with over 2000 business

This work was partially supported by the German Research Foundation (DFG) within the CRC "On-The-Fly Computing" (CRC 901, Project Number: 160364472SFB901) and the German Federal Ministry of Education and Research (BMBF) through Software Campus grant (Project Number: 01IS17046).

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 262–276, 2021.
https://doi.org/10.1007/978-3-030-79186-5_17

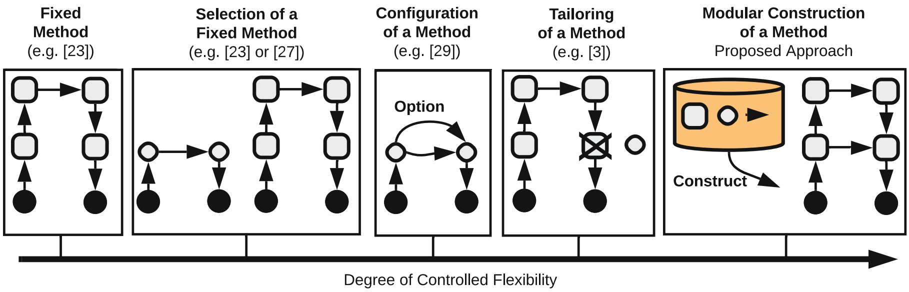


Fig. 1. Categorization of *Business Model Development Methods* (adapted from [20])

executives. In this study, 64% of these executives have the "difficulty to define an effective business model to support new ideas and make them profitable" [14]. An important reason for this is that customers expect solutions for perceived needs rather than just products [30]. These perceived needs correspond to the potential effect that the business model can be often more important than the latest technology of the product [6]. Attractive markets for companies are mobile ecosystems. As highlighted by the AppAnnie's State in Mobile 2021 study [1], these ecosystems provided 218 billion app downloads that led to 142 billion dollar revenue just in 2020. Nevertheless, app developers compete with their app against millions of other apps over the users' usage time. Therefore, effective business models are essential for staying successful in these markets.

The process of business model development is a creative task that often requires the collaboration of different stakeholders within a company [7,8]. The process needs a deep understanding of the market, the existing competitors, and potential customers [30]. From this analysis, different possible business models have to be created in this process, and the corresponding assumptions have to be validated within the market [26]. This validation, for example, can be done by conducting experiments with the potential customers of the product [22]. To support the business model development, different domain experts propose various methods to develop such business models in the form of development processes (e.g., [9,22]) and method repositories (e.g., [3,27]). However, as shown in Fig. 1, most approaches (e.g. [22,26]) provide *Fixed Methods* that do not focus on the mobile app developer's situation. This can lead to poor resource management and longer development cycles. Some approaches, see Fig. 1, try to cover this lack of *Controlled Flexibility* by providing different *Configurations of a Method* [28] or basic *Tailoring of a Method* [3]. Nevertheless, these one-size-fits-all methods are not capable of covering all relevant contextual factors. Examples for the factors could be the own company (e.g., low vs. high financial resources), the market (e.g., mass vs. niche market), the competitors (e.g., high vs. low number of competitors), or the potential customers (e.g., adults vs. children).

This *Modular Construction of a Method* to the situation, in turn, is possible with Situational Method Engineering (SME). SME is a subfield of method engineering and creates new methods out of existing methods based on the situational context where the method is applied [21]. To consider the situation in the business model development methods, we apply SME to business modeling with the roles of a domain expert, a method engineer, and a business developer. Here, the method engineer uses the method knowledge of the domain expert to create a method base. This method base consists of method elements (e.g., stakeholders, tasks) that are composed into method building blocks (e.g., conduct customer interview as a task with the customer as a stakeholder) and structured according to method patterns (e.g., conduction of customer interviews before the development of prototype). The method engineer can then use this method base to construct development methods according to the business developer's described situation. To apply our approach for mobile app developers, we construct a method base by conducting a grey literature review [12] of articles for developing business models of mobile apps. We extract the explicit and implicit steps of the development process and map them to their method elements, method building blocks, and method patterns. For using this method base in practice, we develop an open-source tool for the creation of situation-specific business model development methods. Moreover, we evaluate the whole approach by a case study of constructing a method of an event platform app.

2 Background and Related Work

2.1 Business Model Development

Business model development (BMD) is a continuous and challenging task, which often uses creativity and collaboration between different stakeholders [8]. It consists of several phases (i.e., initiation, ideation, integration, implementation in the BMI Magic Triangle [10]) where different possible business models have to be created, and the corresponding assumptions have to be validated within the market [26]. This validation, in turn, can be done by conducting experiments with the potential customers of the product/service [22]. Here, a method repository with different experiment sequences based on the type of business is introduced to provide flexibility in the method construction [3]. Moreover, the flexibility can be supported by alternative choices for process steps inside the method [28]. Nevertheless, those approaches focus on high abstraction levels and one-size-fits-all methods that cannot cover all relevant contextual factors. Artifacts and tools can support those approaches. One artifact is the Business Model Canvas [23], which divides the business model into nine building blocks where each building block consists of different elements. Moreover, manual tools like pattern databases [13] or software-based tools [29] can be used.

These software-based tools are often called Business Model Development Tools (BMDT) and provide different guidance levels to develop new and improve existing business models [29]. Here, earlier examples of these tools in the literature focus on the visualization of the business model [11] or simple financial assessments [16]. An analysis of business modeling tools in practice [29] shows that those tools focus on the design of business modeling but not on the actual decision support. Nevertheless, a shift from simple design support of business modeling to real decision support by these tools needs to be done [24]. BMDTs already introduce possible parts of decision support in research. Virtual Business Model Innovation [7] introduced a concept of a BMDT with the four phases of analysis, design, implementation, and management. The Framework for Analysis of Business Model Management does a similar division into phases [31]. To provide decision support in different phases, the Business Model Developer [4] supports the configuration of the business model together with validation in terms of static analysis. Moreover, a domain-specific modeling method for configuring the business models and generating business plans is introduced [32]. Nevertheless, the proposed tools focus on a high abstraction level to propose a generic solution for all phases or lower abstraction of single phases or specific application domains. Moreover, none of them provides the flexibility of a method repository to construct the used method.

2.2 Situational Method Engineering

A method is a description of how to perform a procedure systematically [5]. For that, a method can be decomposed into different parts. Here, a method fragment is a reusable atomic block of a method that can have a process (called work unit), a product (called work product), or a producer focus [5]. This focus can further refined (e.g., product into activity) due to the application of the whole method. A method component consists of inputs and outputs of work products together with a process to transform the input into the output [21]. We will use the naming method elements for a method fragment and method building block for a method component to stick to the business model terminology. Moreover, we use the term of method pattern to note sequences of method building blocks. Method Engineering, in turn, is a research discipline to design, construct, and adopt these methods for the development of information systems [5]. If the method is developed for a specific project and depends on the project's situational context, it is called Situational Method Engineering (SME) [21].

SME has its origin in creating software development methods and typically consists of the two roles of a method engineer and a project manager. Here, the method engineer analyzes various methods and stores them in a method base. After that, the method engineer identifies the project's situational factors and constructs a suitable method of the method base. This method, in turn, is then enacted by the project manager to manage the project. While most of the existing approaches focus on developing software products, some also include business-related parts to their methods base. Here, a study [25] investigates how the agility of different methods can be used in method engineering. A case study [2] identities different situational factors for the business, the customer (e.g., number of customers), market characteristics, product characteristics, and stakeholder involvement for phases in product management. An SME approach of IoT development methods [15] also includes business-related (e.g., regulations) and customer-related (e.g., domain experience) situational factors together with business-related (e.g., IoT Canvas) work products. Nevertheless, those approaches cover the business aspect as one side aspect of the product development process. They do not consider the BMD as a separate continuous process with its characteristics like hypothesis-driven experimentation [3].

3 Research Approach

This paper introduces situation-specific business model development methods. We apply the approach for mobile app developers by providing a method base on practical knowledge outside the research world. To gather such knowledge systematically and repeatably, grey literature reviews (GLRs) have been introduced as a promising approach in the last years [12]. Here, we follow the guidelines according to Garousi et al. [12], who structure the GLR in the three phases of *(1) Planning the Review*, *(2) Conducting the Review*, and *(3) Reporting the Review*. While this section considers just the most essential aspects, the complete review with sources and results is provided in our technical report [19].

In *(1) Planning the Reviews*, the need for a GLR needs to be motivated together with the explicit formulation of the research question the study aims to answer. Out of the research question, the search string and related inclusion and exclusion criteria are determined. To motivate the need for the GLR, we have used the checklist of Garousi et al. [12]. Here, we concluded the need for a GLR based on the subject's complexity and the lack of practical experience in the formal literature. For filling the method base, we have defined the following research question: *What are the main business model development steps that need to be done by a mobile app developer?* To answer this question and by testing different search terms with the terminology in business model development, we have defined the following search string: *app AND (business model OR idea) AND (test OR validate OR develop).* We include articles in English where the URL is accessible and directly connected to the research question. We exclude articles that provide no business model development process, not relate to mobile apps, are posted in forums, or are presented in the non-textual form.

In *(2) Conducting the Review*, the review needs to be conducted by considering the search process, the source selection, the quality assessment, the data extraction, and the data synthesis. In the search process, we applied the search string to the Google search engine on January 19th, 2021, and anonymized our browser data for a maximum of objective results. We exported the first 50 results of the search results and manually scanned the inclusion and exclusion criteria that result in 38 articles. For the quality assessment, the essential criteria were the understandability of the processes. Moreover, the method base will contain links to the articles so that mobile app developers can convince themselves of the quality. After analyzing the first results, we found the division between atomic blocks (i.e., method elements), combined steps (i.e., method building blocks), and their different orderings (i.e., method patterns), which we choosed as initial attributes. We were iteratively refining our initial attribute method elements into the atomic parts of tasks, types, stakeholders, situational factors, artifacts, and tools. Based on these atomic elements, we extracted all method elements in the data extraction. Finally, in the data synthesis, which is shown in the technical report, we combined these elements into building blocks and patterns.

In *(3) Reporting the Review*, we documented the review results. We do this by publishing the highlights of our results in this research paper and providing access to the whole method base in our technical report [19].

4 Creation of Business Model Development Methods

In this section, we describe the creation of situation-specific business model development methods. As shown in Fig. 2, our approach adds the two roles of the *Domain Expert*, who provides the method knowledge, and the *Method Engineer*, who structures the method knowledge, in addition to the *Business Developer*, who describes his situation. Moreover, we have the two stages of *(1) Creation of Method Base* and *(2) Construction of Development Method*. After giving an overview of both stages, we will explain them in more detail and introduce a tool-support for the method development.

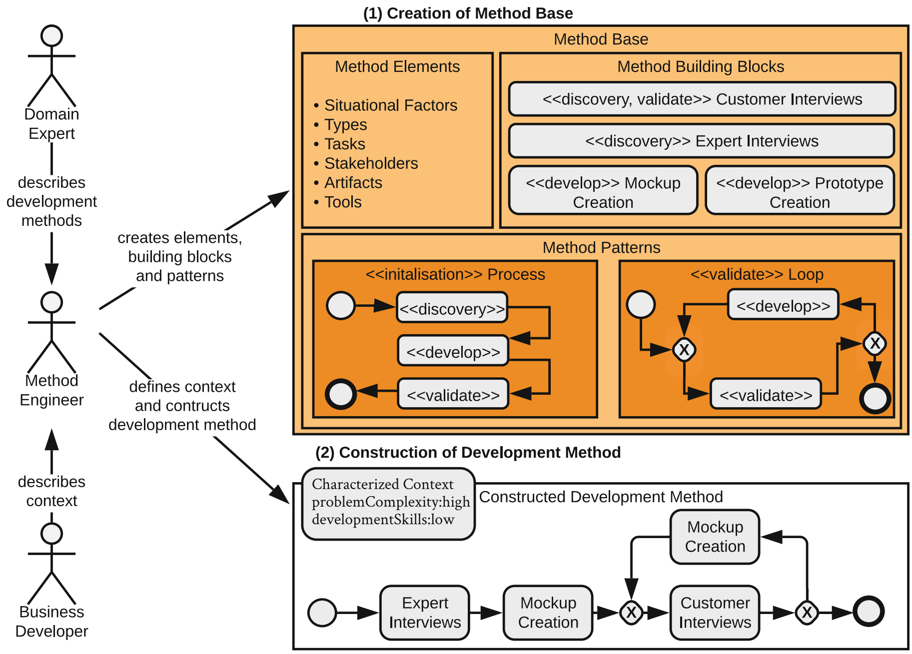


Fig. 2. Exemplary construction of a development method for the *Business Developer* by the *Method Engineer* based on the method knowledge of the *Domain Expert*

In the *(1) Creation of Method Base*, we have the *Method Base* consisting of *Method Elements*, *Method Building Blocks*, and *Method Patterns*. *Method Elements* are atomic parts of the method consisting of the possible *Situational Factors*, different *Types* of methods, performed *Tasks*, involved *Stakeholders* created *Artifacts* and used *Tools*. Multiple elements are combined to *Method Building Blocks* and structured to methods using *Method Patterns*. The content of the *Method Base* is created by the *Method Engineer* based on the knowledge of the *Domain Expert*. For example, the *Method Base* in Fig. 2 consists of the building blocks of *Expert Interviews* and *Mockup Creation*.

In the *(2) Construction of Development Method*, the *Constructed Development Method* is created out of the *Method Base*. For this, the *Business Developer* describes the current context. The *Method Engineer* uses this information to define the context according to the *Situational Factors* and constructs the method out of the combined *Method Patterns*. Moreover, the *Method Patterns* are filled with the *Method Building Blocks*. Here, the *Constructed Development Method* is characterized by a high *problemComplexity* and low *developmentSkills*. Therefore, *Expert Interviews* and *Mockup Creation* are selected from the *Method Base*.

4.1 Creation of Method Base

The first stage of our approach is the creation of the *Method Base*. This is done by the *Method Engineer* who is formalizing the *Method Elements*, *Method Building Blocks*, and *Method Patterns* out of the knowledge of the *Domain Expert*. In our case, we use the results of the grey literature review as domain knowledge. The usage of the grey literature review, in turn, allows us to accumulate knowledge from the practice of different experts in the field of business model development. While in this paper, we cover just the most important findings of our review, the full source of 234 method elements, 57 method building blocks, and 28 method patterns is available in our report [19]. Moreover, the whole *Method Base* can be used in the tool. An excerpt of this *Method Base* is shown in Fig. 3 and consists of *Method Elements*, *Method Building Blocks*, and *Method Patterns*.

The *Method Elements* are atomic parts of the methods that can be divided into tasks, types, stakeholders, situational factors, artifacts, and tools. *Tasks* are the main activities that need to be performed during the business model development. Here, various tasks on different granularity levels (e.g., *Conduct Interviews*, *Create Prototype*) are presented in the literature. *Types* are used to structure the building blocks and patterns. They can be divided into functional types, structure types, and method types. *Functional Types* are used to add additional logic to the method patterns. Here, the type *initialisation* can be used to represent the start patterns of the approach, and the type *generic* can be used to specify patterns that can be inserted into any other pattern, regardless of the type. Next, *Structure Types* are used to create a structure between different patterns. Finally, *Method Types* are used to classify different types of building blocks. *Stakeholders* are the persons who are involved in the building blocks. Here, we can divide between stakeholders of the own *Company* (e.g., *Business Developer*), existing *Partners* (e.g., *Development Agency*), and potential *Users* (e.g., *Customer*). *Situational Factors* are used to classify in which context a building block or a pattern can be applied. Here, we found the different categories of the own *Company* (e.g., *developmentSkills*), the identified *Problem* (e.g., *problemComplexity*), the potential *Customer* (e.g., *customerGroup*), the targeted *Market* (e.g., *marketSize*), the developed *Product* (e.g., *productComplexity*), and the state of the *Phase* (e.g., *productValitidy*) for structuring the factors. Moreover, the values of the factors can be nominal variables (e.g., *mass*

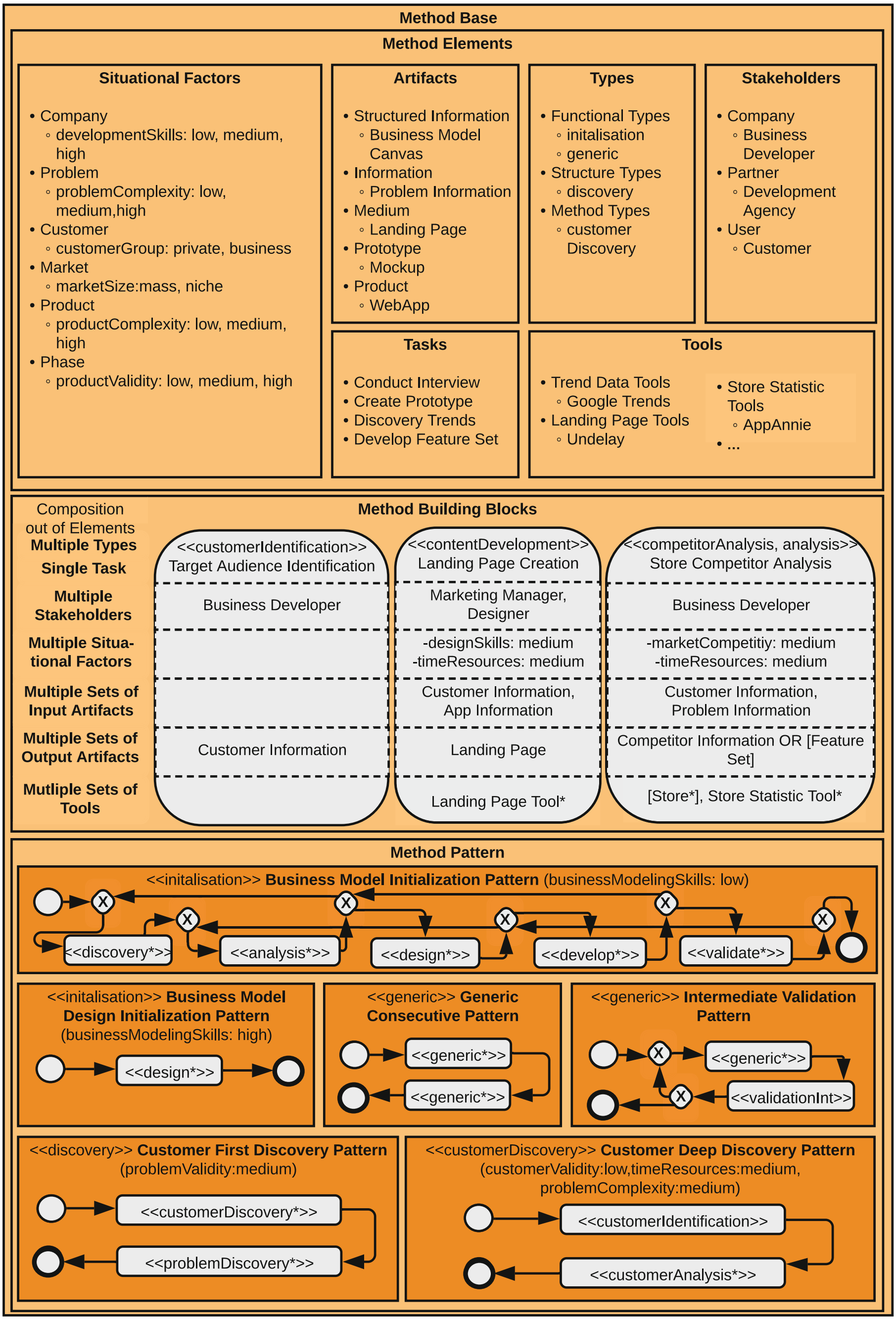


Fig. 3. Excerpt of the *Method Base* created by the Grey Literature Review

or niche for *marketSize*) or ordinal scales (e.g., *low* < *medium* < *high* for *developmentSkills*). *Artifacts* are working products that are created and modified during the business model development process. Here, we found the categories of *Structured Information* (e.g., *Business Model Canvas*), unstructured *Information* (e.g., *Problem Information*), used *Mediums* (e.g., *Landing Page*), created *Prototypes* (e.g., *Mockups*), and developed *Products* (e.g., *WebApp*). *Tools* are used to support the performing of activities. For this, there can be tools for supporting the tasks or creating and modifying artifacts.

The *Method Building Blocks* are combined out of the different elements. Here, a single *Task* is included in the method block together with multiple *Types* and multiple *Situational Factors*. Moreover, for the *Artifacts* we allow the modeling of *Input Artifacts* and *Output Artifacts* so that *Method Building Block* can be interpreted as the transformator of the *Artifact*. For the *Stakeholders*, *Artifacts* and *Tools*, we allow the definition of multiple sets, where the concrete set needs to be selected during the construction of the method (e.g., set with single elements of *Expert* or *Customer* as *Stakeholder* for *Conduct Interview* as a *Task*). Here it is also possible to define just a category of the element so that any element of this category can be selected during the construction (e.g., category of *User* as *Stakeholder* for *Conduct Interview*). This, in turn, allows us to reduce the manual modeling effort in the *Method Base* while keeping the method space flexible in a targeted manner. Last, we provide a list function, which means that multiple elements can be used within the building block (e.g., multiple *Feature Sets* out of a *Competitor Analysis*). With *Target Audience Identification*, *Landing Page Creation* and *Store Competitor Analysis*, three example building blocks are shown in Fig. 3. While the *Target Audience Identification* is completely fixed, the *Landing Page Creation* allows the choice of a tool at the construction (marked with ∗). Moreover, the *Store Competitor Analysis* allows the output artifacts of competitor information for multiple artifacts of a *Feature Set* (marked with []).

The *Method Patterns* are used to assemble the building blocks in a structured way. For this, each pattern has a type, a name, additional situational factors, and a process part in the form of a BPMN model. Here, the *initialisation* type refers to patterns that can be used at the beginning of the construction. In our example, this is the construction of a process with all extracted phases (e.g., *Business Model Initialization Pattern*) for beginners in business model development (e.g., *businessModelingSkills* are *low*) or single phases (e.g., *Business Model Design Initialization Pattern*) for experts in business model development (e.g., *businessModelingSkills* are *high*). Moreover, the *generic* type can be made to classify patterns that can be used in every other pattern to increase the number of steps (e.g., *Generic Consecutive Pattern*) or provide additional validation to a single step (e.g., *Intermediate Validation Pattern*). Finally, the placeholders in the pattern can be filled with method building blocks (marked with << *type* >>) or a choice of building blocks and patterns (marked with << *type*∗ >>).

4.2 Construction of Development Method

The second stage, as shown in Fig. 2, is the construction of the development method. This is done by the *Method Engineer* who is defining the context and constructing the method out of the description of the *Business Developer*.

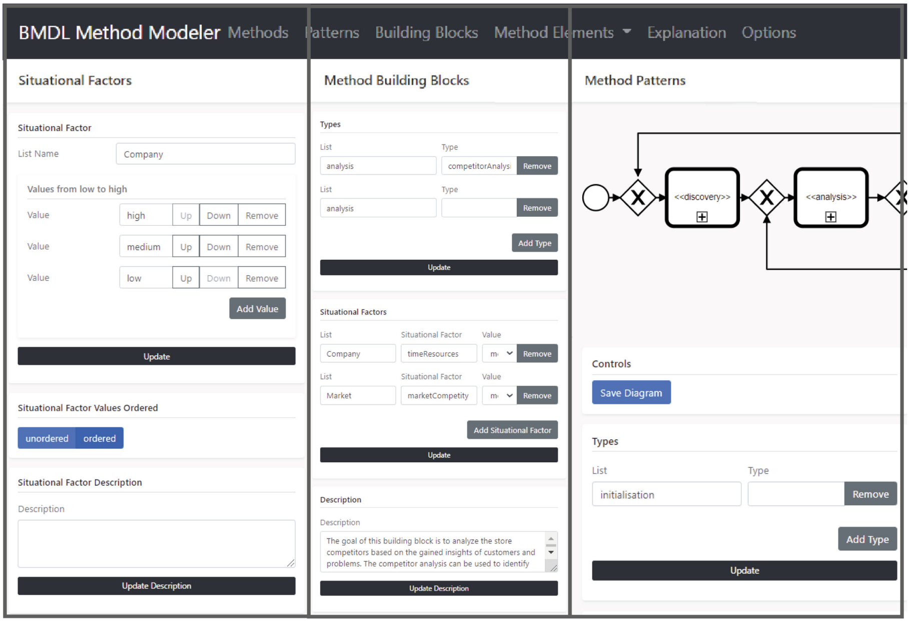
For the definition of the *Context*, a subset of the *Situational Factors* of the *Method Elements* has to be selected with corresponding values of the situation. These provided values are used to give recommendations for *Method Building Blocks* and *Method Patterns* by ordering both according to the distance between the provided factors of the context and the needed factors of the buildings blocks and patterns. Thus, the weighted distance of all factors selected for the context and provided by the building blocks and pattern is calculated. While nominal variables for factors need concrete matching (e.g., *mass* vs. *niche* for *marketType* has distance 1), ordinal scales are weighted according to the scale (e.g., provided *developmentSkills: medium* and required *developmentSkills: high* has distance 1/2). Moreover, we automatically matching values in the including direction (e.g., provided *developmentSkills: medium* and needed *developmentSkills:low* has distance 0) to cover building blocks and patterns outperform the context.

For the construction of the *Development Method*, a structured selection of building blocks and patterns based on the weighted distance is made. In the beginning, all patterns with the *initialisation* type out of the *Method Base* are recommended by the approach. Next, the placeholders inside this pattern need to be filled with *Method Building Blocks* of the required or the generic type (marked as $<< type >>$ and $<< generic >>$). During this filling, also abstract building blocks have to be instantiated with concrete elements from the *Method Base*. Moreover, it is possible to fill the placeholders with a mixture of *Method Building Blocks* and *Method Patterns* (marked as $<< type* >>$ and $<< generic* >>$). This process is repeated until all placeholders in the method are filled out with building blocks or patterns. To allow a continuous expansion of the method, it is also possible to extend the first pattern by another pattern of type *initalisation*. After the complete method has been constructed, the last step is to check the quality of the method. To ensure the quality of the method, the approach checks if every abstract building block is filled out, and for every input artifact, there is an output artifact created before in the method.

4.3 Tool-Support for Method Development

To use our approach in practice, we have developed a corresponding tool-support. The open-source tool BMDL Method Modeler, which is based on our BMDL Feature Modeler [18], can be accessed online[1] or downloaded[2]. The BMDL Feature Modeler, in turn, has been developed to derive business models out of an existing business domain knowledge. In the past, we have also extended the tool to model the knowledge of different experts [17]. Based on Angular and

[1] Use the Tool: https://sebastiangtts.github.io/bmdl-method-modeler/.
[2] Download the Tool: https://github.com/SebastianGTTS/bmdl-method-modeler.



a) Create Method Elements b) Create Method Building Blocks c) Create Method Patterns

Fig. 4. Creation of the *Method Base*

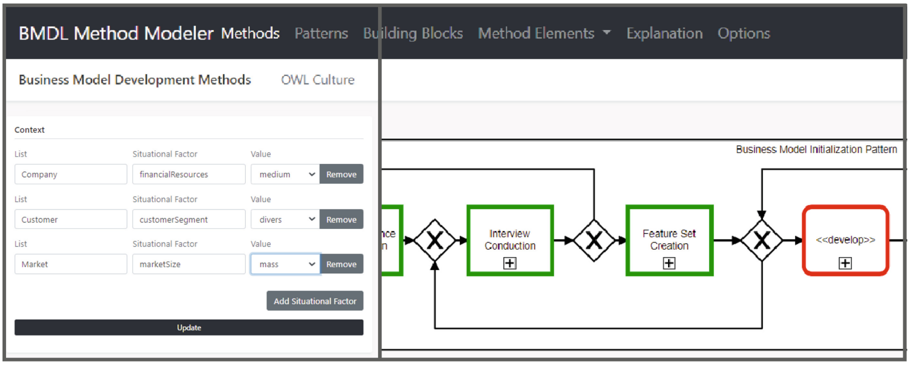
the BPMN.io Framework by Camunda, we have created a method development tool for the *Method Engineer*. The tool can be divided into the *(1) Creation of Method Base* and the *(2) Construction of Development Method*.

The *(1) Creation of Method Base* is shown in Fig. 4. Here, the *Method Engineer* can model the *Situational Factors* (see Fig. 4(a)), the *Stakeholders*, the *Types*, the *Artifacts* and the *Tools* as *Method Elements*. After that, he can create *Method Building Blocks* by choosing corresponding elements for a *Task* (see Fig. 4(b)). Moreover, he can define *Method Patterns* with corresponding placeholders for *Types* and recommended *Situational Factors* (see Fig. 4(c)).

The *(2) Construction of Development Method* is shown in Fig. 5. Here, the *Method Engineer* can characterize the context of the mobile app developer (see Fig. 5(a)). Based on this situation, he receives recommendations for *Method Patterns* and *Method Building Blocks*. During the construction of the method, he received live feedback on quality problems (see missing building block for << *develop* >> in Fig. 5(b)).

5 Case Study on Local Event Platform

In this section, we evaluate our approach by conducting a case study on OWL Live. OWL Live is a local event platform created in the OWL culture portal's



a) Characterize Context b) Create Development Method

Fig. 5. Construction of the *Development Method*

research project[3]. This research project aims to establish a local area event platform that the project partners should sustainably operate. The value of the platform is to aggregate event information from different sources based on machine learning algorithms. OWL Live is a two-sided market between event providers and event visitors that both have to be considered during the business model development. Therefore, an effective business model development method that fits the project partners' situation is essential for the platform.

To create a business model development method, we interview the responsible project manager to gather the context and additional project information. Next, we constructed a development method together with the project manager by using the BMDL Method Modeler. While the situational factors (e.g., *customerSegment: uniform*, *marketSizeMass:mass*, *customerValidity:high*) could be easily used in the tool, the additional information (e.g., different customer groups due to platform business, the promised conduction of a closed and open beta phase) are harder to model by using the different patterns and building blocks. After the construction, we discussed the results with the project manager. Here, the major outcome was that the approach supports structuring the business model development and allows the discovery of new essential method parts (e.g., analysis of store trends, inbound marketing during the development).

An overview of the development method, which is fully displayed inside the tool, can be seen in Fig. 6. As the project has already defined and validated a first target group, they can use the *Target Audience Identification* without a deeper analysis. To get into the problem, we suggest running a *Market Problem Observation* to find additional pain points, and a *Store Trend Discovery*, to identify existing trends in similar apps. Moreover, we suggest analyzing the problem by a *Customer Interview Conduction* for event providers and *Social Media Survey Conduction* for event visitors. After that, a *Market Potential Analysis* should be the foundation for sustainable operation, and the *Store Competitor Analysis*

[3] Project Website: https://www.sicp.de/en/projekte/owlkultur-plattform.

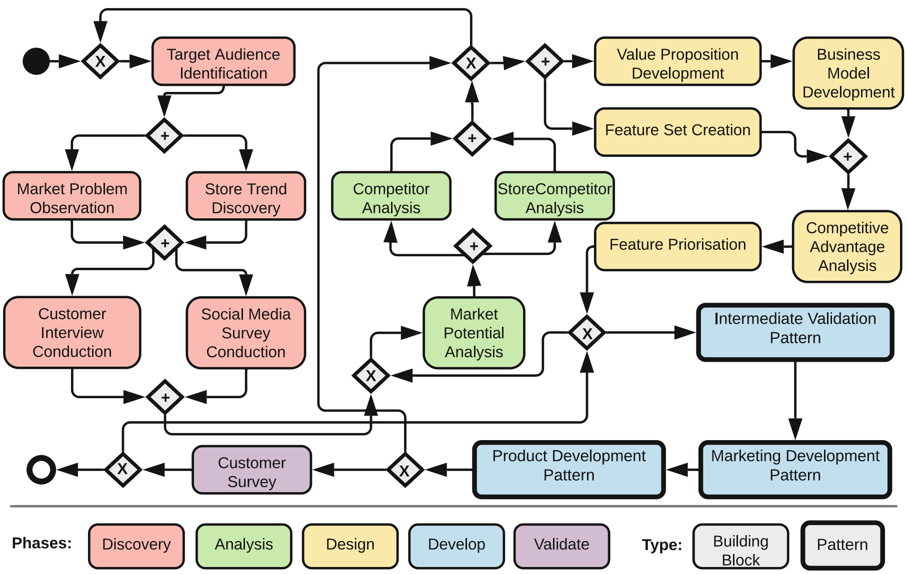


Fig. 6. Business model development method of OWL Live

and *Competitor Analysis* should allow getting an overview of other platforms. After that, the *Feature Set*, the *Value Proposition Development* and the *Business Model Development* should be conducted. Because of the different beta phases, it is important to do a *Competitive Advantage Analysis* and a subsequent *Feature Prioritization*. After that, the closed beta of the developed MVP with fewer users should be validated with group interviews in the first *Intermediate Validation Pattern*. After that, the open beta will be validated with customer surveys in the second *Marketing Development Pattern*. Moreover, during this open beta, the conduction of inbound marketing for getting new users is recommended. After that, the product will be developed with the *Product Development Pattern* and ongoing validated with a *Customer Survey*.

6 Conclusion

The development of effective business models is an essential task in mobile ecosystems. Here, existing development methods for these business models do not bring into focus that the development process profoundly depends on the mobile app developer's situation. To solve this problem, we have introduced an approach for creating situation-specific business model development methods. For that, our approach consists of the two stages of creating a method base and the construction of a development method. We have implemented our approach as an open-source tool and evaluated it by conducting a case study to create a business model development method for a local event platform.

Our future work is threefold and deals with enhancing the method base, enacting the development method, and the light-weight structuring of method building blocks. First, by conducting our GLR, we comparatively quickly reach saturation of new knowledge. Therefore, we want to extend our approach so that multiple domain experts can enhance the method base with their domain knowledge (e.g., gaming apps) and improve the GLR with scientific literature. Second, by analyzing the business model development domain, we saw that the used methods could change over time. Therefore, we want to support the enactment of the development method with an internal execution engine. This integration of construction and enaction of the method should allow a runtime adaptation of the method due to changing situations. Third, by analyzing our pattern structuring process, we investigate that it provides high guidance in structuring but also increases the modeling effort. Therefore, we want to examine a light-weight structuring of method building blocks using kanban boards. This structuring, in turn, should reduce the setup costs and complexity for smaller businesses.

References

1. App Annie Inc: The State of Mobile 2021. https://www.appannie.com/en/go/state-of-mobile-2021/
2. Bekkers, W., van de Weerd, I., Brinkkemper, S., Mahieu, A.: The influence of situational factors in software product management: an empirical study. In: International Workshop on Software Product Management, pp. 41–48. IEEE (2008)
3. Bland, D.J., Osterwalder, A.: Testing Business Ideas. Wiley, Hoboken (2020)
4. Boßelmann, S., Margaria, T.: Guided business modeling and analysis for business professionals. In: Pfannstiel, M.A., Rasche, C. (eds.) Service Business Model Innovation in Healthcare and Hospital Management, pp. 195–211. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-46412-1_11
5. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. Inf. Softw. Technol. **38**(4), 275–280 (1996)
6. Chesbrough, H.: Business model innovation: it's not just about technology anymore. Strat. Leadersh. **35**(6), 12–17 (2007)
7. Ebel, P., Bretschneider, U., Leimeister, J.M.: Leveraging virtual business model innovation: a framework for designing business model development tools. Inf. Syst. J. **26**(5), 519–550 (2016)
8. Eppler, M.J., Hoffmann, F., Bresciani, S.: New business models through collaborative idea generation. Int. J. Innov. Manage. **15**(06), 1323–1341 (2011)
9. Fagerholm, F., Sanchez Guinea, A., Mäenpää, H., Münch, J.: The RIGHT model for continuous experimentation. J. Syst. Softw. **123**, 292–305 (2017)
10. Frankenberger, K., Weiblen, T., Csik, M., Gassmann, O.: The 4I-framework of business model innovation: a structured view on process phases and challenges. Int. J. Product Dev. **18**(3/4), 249 (2013)
11. Fritscher, B., Pigneur, Y.: Supporting business model modelling: a compromise between creativity and constraints. In: England, D., Palanque, P., Vanderdonckt, J., Wild, P.J. (eds.) TAMODIA 2009. LNCS, vol. 5963, pp. 28–43. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11797-8_3
12. Garousi, V., Felderer, M., Mäntylä, M.V.: Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. Inf. Softw. Technol. **106**, 101–121 (2019)

13. Gassmann, O., Frankenberger, K., Csik, M.: The Business Model Navigator: 55 Models that Will Revolutionise your Business. Pearson, Harlow (2014)
14. General Electric Inc.: GE Global Innovation Barometer (2018). https://www.ge.com/reports/innovation-barometer-2018/
15. Giray, G., Tekinerdogan, B.: Situational method engineering for constructing internet of things development methods. In: Shishkov, B. (ed.) BMSD 2018. LNBIP, vol. 319, pp. 221–239. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94214-8_14
16. Gordijn, J., Akkermans, H.: Designing and evaluating e-business models. IEEE Intell. Syst. **16**(4), 11–17 (2001)
17. Gottschalk, S., Kirchhoff, J., Engels, G.: Extending business model development tools with consolidated expert knowledge (accepted). In: Business Modeling and Software Design. Springer (2021)
18. Gottschalk, S., Rittmeier, F., Engels, G.: Hypothesis-driven adaptation of business models based on product line engineering. In: International Conference on Business Informatics (CBI). IEEE (2020)
19. Gottschalk, S., Yigitbas, E., Nowosad, A., Engels, G.: Situation-specific business model development methods for mobile app developers: technical report. https://cs.uni-paderborn.de/fileadmin/informatik/fg/dbis/SituationSpecificBMDM.pdf
20. Grieger, M.: Model-Driven Software Modernization: Concept-Based Engineering of Situation-Specific Methods (2016)
21. Henderson-Sellers, B., Ralyté, J., Ågerfalk, P.J., Rossi, M.: Situational Method Engineering. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-41467-1
22. McGrath, R.G.: Business models: a discovery driven approach. Long Range Plann. **43**, 247–261 (2010)
23. Osterwalder, A., Pigneur, Y.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. Wiley, Hoboken (2010)
24. Osterwalder, A., Pigneur, Y.: Designing business models and similar strategic objects: the contribution of IS. J. Assoc. Inf. Syst. **14**(5), 237–244 (2013)
25. Qumer, A., Henderson-Sellers, B.: An evaluation of the degree of agility in six agile methods and its applicability for method engineering. Inf. Softw. Technol. **50**(4), 280–295 (2008)
26. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Business, USA (2014)
27. Shin, J., Park, Y.: On the creation and evaluation of e-business model variants: the case of auction. Ind. Market. Manage. **38**(3), 324–337 (2009)
28. Simmert, B., Ebel, P.A., Peters, C., Bittner, E.A.C., Leimeister, J.M.: Conquering the challenge of continuous business model improvement. Bus. Inf. Syst. Eng. **61**(4), 451–468 (2018). https://doi.org/10.1007/s12599-018-0556-y
29. Szopinski, D., Schoormann, T., John, T., Knackstedt, R., Kundisch, D.: Software tools for business model innovation: current state and future challenges. Electr. Markets **30**(3), 469–494 (2019). https://doi.org/10.1007/s12525-018-0326-1
30. Teece, D.J.: Business models, business strategy and innovation. Long Range Plann. **43**(2–3), 172–194 (2010)
31. Terrenghi, N., Schwarz, J., Legner, C., Eisert, U.: Business model management: current practices, required activities and IT support. In: Proceedings of the Conference on Wirtschaftsinformatik, pp. 972–986 (2017)
32. Wieland, M., Fill, H.G.: A domain-specific modeling method for supporting the generation of business plans. Modellierung **2020**, 45–60 (2020)

Searching for Class Models

Maxim Bragilovski(✉), Yifat Makias, Moran Shamshila, Roni Stern, and Arnon Sturm

Ben-Gurion University of the Negev, Beer Sheva, Israel
maximbr@post.bgu.ac.il, {sternron,sturm}@bgu.ac.il

Abstract. Models in model-based development play a major role and serve as the main design artifacts, in particular class models. As there are difficulties in developing high-quality models, different repositories of models are established to address that challenge, so developers would have a reference model. Following the existence of such repositories, there is a need for tools that can retrieve similar high-quality models. To search for models in these repositories, we propose a greedy algorithm that matches the developer's intention by considering semantic similarity, structure similarity, and type similarity. The initial evaluation indicates that the algorithm achieved high performance in finding the relevant class model fragments. Though additional examination is required, the sought algorithm can be easily adapted to other modeling languages for searching models and their encapsulated knowledge.

Keywords: Model driven development · Model repository · Model search

1 Introduction

Model-Based Development is continuously evolving [1]. In such a development, models play a major role and serve as the main design artifacts that developers aim at achieving. Due to the difficulties in developing high quality models developers aim at starting from an established point in which an existing model can be used for the planned application/system [2]. To facilitate that approach, repositories of models start emerging including for example [8,21]. There are many challenges in establishing such repositories [7]. These include technical issues in terms of model representation, scalability, and heterogeneity and semantic issues of how to analyze and query such repositories. In this paper, we aim at addressing the challenge of querying model repositories, and in particular class models. Indeed, existing repositories provide searching capabilities, yet, these mainly refer to meta data (e.g., size and type) or to textual and keyword similarity. However, similarity between model artifacts requires consideration of both or at least semantic similarity and structural similarity [26]. In recent work the graph structure is also taken into account when searching model repositories [11]. Nevertheless, the capability provided neglect the semantic similarity. One of the

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 277–292, 2021.
https://doi.org/10.1007/978-3-030-79186-5_18

main benefits of measuring semantics similarity between software artifacts (e.g., code, State Chart, Class Diagram...) is the ability to reuse them so to save time during the software development process [10,14]. In this paper, we aim at addressing this gap by proposing an algorithm that searches for class models that match the developers intention. The algorithm is based on a greedy search approach where in each iteration the algorithm tries to find the most similar class to the next class of the query using similarity function that considers type similarity, structure similarity and label similarity. We also evaluated the performance of the algorithm by various information retrieval metrics and found the result promising.

The paper is organized as follows. Section 2 analyze related studies for searching and measuring similarities between models. Section 3 introduces the sought algorithm and Sect. 4 presents its evaluation. Finally, Sect. 5 concludes and sets plans for future research.

2 Related Work

Several approaches have addressed the challenge we introduced. In this section we review related studies with respect to the following:

- **Type of similarity** refers to the way similarity is being calculated.
- **Query form** refers to the considered query form (e.g., text, model, type).
- **Experimental settings and datasets** refer to the way the suggested solutions were evaluated.
- **Examined metrics** refers to the way similarity and related algorithms are being considered.
- **Performance** refers to the values of the mentioned metrics.

Robles et al. suggest a new approach for representing knowledge of UML Class Diagrams (UCD) based on Information Retrieval (IR) techniques and calculating their semantic similarity using ontology [22] graphs. The type of similarity is calculated by a formula that considers semantic similarity between the concepts of the diagrams and the topological similarity between the types of elements in the diagrams. The queries are represented as UCD. To evaluate the proposed method, they used fifteen queries of different levels of complexity and performed the search on a corpus that contains sixty-five UCDs. To demonstrate the importance of using ontology similarity they performed their experiment twice, with and without ontology that is similar to the Boolean model in the IR area. The F-score indicated that using ontology leads to significantly better results of 0.83 compared to 0.36 on average.

Similar to [22], Al-Rhman et al. suggest different type of similarity equations, for example: lexical name similarity, attribute similarity, and operation similarity [4]. Al-Rhman et al. show how weighing each of the elements in the equation affects the total similarity between UCDs. They further suggest a greedy-based matching algorithm that finds the similarity of classes of two models in polynomial time [3]. They proposed several different metrics to measure similarity between UCDs. They examined which metric is the most accurate for UCDs in the same domain or from different domains. The first case contains five diagrams grouped up into ten pairs in the same domain. The results indicate that the lexical name similarity that measures the semantic similarity between classes gave the best results in terms of precision, recall, and accuracy. The second case contains four UCD formed into six pairs. In this case, the neighboring similarity that measures the similarity of two classes is superior in terms of precision, recall, and accuracy. Nikpforova et al. [16] suggest an algorithm that converts each class diagram to a vector and then refers to the similarity between them based on distance calculation. The experiment consists of three small similar UCDs.

Even though these methods show promising results, they rely on the fact that similar UCDs have to be in the same context. To overcome this problem, several works have been proposed to incorporate structural similarity. For UML Sequence Diagram, for example, Salami et al. [24] proposed a genetic algorithm that compares two sequence diagrams based on adjacency matrices. They calculate the Mean Average Precision (MAP) of 10 queries to determine how well their method performs for retrieving similar UML Sequence diagrams. The MAP was approximately 0.75 in all ten different repositories that they searched in.

Yuan et al. [26] categorized UML Class Diagrams into two aspects: intra-structure and inter-structure. Intra-structure represents the inner structure of classes like attributes and operations, whereas inter-structure represents the relationships between classes like association and composition. Based on this categorization, they converted UCDs into UML Class Graphs (UCG) and measured the similarity of the inter and intra-structure of the graphs. The results were promising based on the fact that the similarity of pairs of graphs was similar to the expert's results. Despite that, they only applied the approach to two domains.

López et al. [11] propose a search engine called MAR for retrieving any EMF model [25] focusing on converting a structure of models into an inverted index [5]. The queries take the form of a model that describes the desired results. They convert each model and query to a bag of paths and use the Okapi BM25 measure [20] for ranking the most relevant models. The similarity function considers the number of times a path appears in both query and model graph and whether those paths are unique.

Another option to consider is the similarity flooding algorithm [13] that aims at finding pairs of elements that are similar in two data schemes or two data instances. Yet, in this work we are interested in paths considering also the labels of the vertices and edges. Nevertheless, such algorithm, can be used as another node similarity function.

In Table 1 we summarized the analysis of the methods we mentioned, along the facets we specified before. The methods measure the similarity between queries and models and assume that similar UCDs have much in common. This assumption may lead to poor performance in cases where users are not fully knowledgeable in the domain that they search for. In this study, we aim at addressing this problem by searching for similar sub-diagrams in UCD and measure the similarity between queries and sub-diagrams of the diagrams. We consider the semantic similarity between labels and attributes of the classes, the type similarity of classes and relationships (for example 'Class' and 'Interface'), and consider the structural path between UCDs while other approaches consider only parts of these similarities.

Table 1. Related work summary

Work	Query form	Models	Type of similarity	Evaluation data	Search method
[22]	UCD	UCD	Semantic + type	65 UCDs + 15 queries	Inverted index
[4]	UCD	UCD	Semantic + type	9 UCDs	Different metrics
[3]	UCD	UCD	Semantic + type	9 UCDs	Greedy-based matching algorithm
[16]	UCD	UCD	Semantic + type	3 UCDs	Similarity function
[24]	USD	USD	Type + structural	60 USD + 10 queries	Genetic algorithm
[26]	UCD	UCD	Structural	60 UCDs + 10 queries	Similarity function
[11]	Any	Correlated to the model	Structural	17975 Ecore meta models[a]	Inverted index

[a] http://mar-search.org/experiments/models20
USD = UML Sequence Diagram; UCD = UML Class Diagram;

3 Greedy Search in UCD

The problem we are aiming to address is the search for relevant UCD fragments within UCDs given a specific query in the form of class diagram. For that purpose we devised a greedy search algorithm in UCDs that addresses the concerns of semantics. The algorithm adopts the similarity considerations appears in [18] and refers to label matching (exact match among labels), structure matching (in terms of links), semantic matching (in terms of labels semantic similarity and links' semantics), and type matching. In the following we first set the ground for the algorithm and then elaborate on its design and execution.

3.1 Settings

Definition 1. *A **UML Class Diagram (UCD)** is a diagram (D) that consists of classes (C), attributes (A), and relationships (R). D = (C, A, R).*

- *C is a set of classes.*
- *A is set of attributes* $A = \{A_1, A_2, ..., A_k\}$ *where* A_i *is a set of attributes of class* c_i. *Each* c_i *has a* $A_i = \{a_1, a_2, ..., a_j\}$.
- *R is a set of relationships. Each D has a* $R = \{r_1, r_2, ..., r_k\}$. *Where* $r_i = \{(c_i, l, c_j) | c_i, c_j \in C, l \in ET\}$. $ET = \{Association, Composite, Generalization, Aggregation, and\ InterfaceRealization\}$.

Definition 2. sim_w *refers to the similarity of two words. This can be calculated using for example WordNet [15], Sematch [27], or google w2v [17].*

Definition 3. *sim_l refers the similarity of two classes labels.*

$$sim_l(list_1, list_2) = \frac{\Sigma_{w_1 \epsilon list_1}(\arg max_{w_2 \epsilon list_2}(sim_w(w_1, w_2)))}{\arg\max(length(list_1), length(list_2))} \quad (1)$$

the lists are ordered set of words (of the labels of the classes).

Definition 4. *sim_{att} refers to the similarity of two sets of attributes.*

$$sim_{att}(att_1, att_2) = \sum_{list_1 \epsilon att_1} \frac{\arg max_{list_2 \in att_2}(sim_l(list_1, list_2))}{\arg max(length(att_1), length(att_2))} \quad (2)$$

Definition 5. *sim_t refers to a predefined similarity among classes types. Table 2 presents the similarity among UCD class types. The similarity matrix shown in Table 2 is based on similarities suggested in [22].*

Table 2. Class type similarity.

	Class	Abstract	Interface	Association class
Class	1	0.75	0.4	0.8
Abstract	0.75	1	0.75	0.7
Interface	0.4	0.7	1	0.2
Association class	0.6	0.7	0.0	1

Definition 6. *sim_c refers to the similarity of two classes.*

$$sim_c(c_1, c_2) = \frac{\mathrm{sim}_l(c_1, c_2) * 0.75 \ + \mathrm{sim}_{att}(att_{c_1}, att_{c_2}) * 0.25 + sim_t(c_1,\ c_2)}{2} \quad (3)$$

The search process should consider both the semantic similarity of the elements as well as their types. Thus, we weight these equally. For the semantic similarity of classes we consider the class label of higher importance with respect to its attributes as these are of secondary relevance and besides they may contain "noise" like generic attributes (e.g., id and status). Therefore we weighted sim_l higher then sim_{att}.

Definition 7. *sim_r refers to a predefined similarity among relationship types. Table 3 presents the similarity among UCD relationships. The similarity matrix shown in Table 3 is based on similarities suggested in [22].*

Table 3. Relationship type similarity

	ASS	CO	AG	GE	INR
ASS	1	0.89	0.89	0.55	0.2
CO	0.89	1	0.89	0.55	0.23
AG	0.89	0.89	1	0.55	0.23
GE	0.51	0.51	0.51	1	0.4
INR	0	0	0	0.21	1

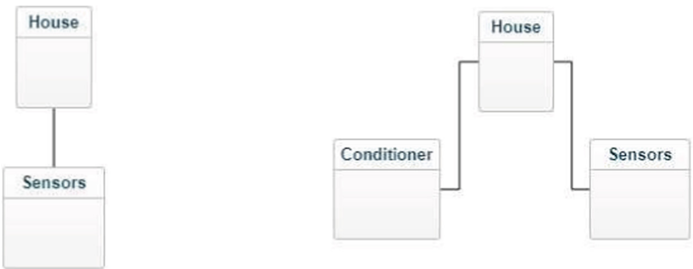
ASS = Association; CO = Composite; AG = Aggregation; GE = Generalization; INR = Interface Realization

Definition 8. *class-relationship similarity is calculated as follow:*

$$sim_{ne}(c_1, r_1, c_2, r_2) = \frac{2 * \mathrm{sim}_c(c_1, c_2) \ + sim_r(r_1,\ r_2)}{3} \tag{4}$$

This definition captures the similarity of a class along with it incoming relationship.

Definition 9. *A query Q is represented as a UCD.* ***Single path*** *query has a single path of classes and relationships (see Fig. 1a).* ***Multi paths*** *query has multiple paths (see Fig. 1b) without cycles.*



(a) Single path

(b) Multi paths

Fig. 1. Single path and Multi paths queries

Definition 10. *Similarity of paths of the query with those of the diagram is calculated as follow (classRelationshipSim is a set of sim_{ne}):*

$$Path - Sim\left(D^1, D^2, classRelationshipSim\right) =$$

$$sim_c\left(c_1^{D^1}, c_1^{D^2}\right) + \sum_{i=1}^{|classRelationshipSim|} classRelationshipSim_i$$

In the following we elaborate the algorithm for single path and multi-path queries. We differentiate between two types of class similarity. The first relies solely on the class information, we call this class-based similarity. The second also refers to the relationship connecting the class from the previous one, we call this class-relationship based similarity.

- The function of **class-based similarity** returns the most similar class in the UCD to a class within the query as calculated by Definition 6.
  Input: $D = (C, A, R)$, query diagram, similarity function. The similarity function is the one appears in Definition 2.
  Output: a class from D.
- The function of **class-relationship based similarity** is responsible for retrieving up to K most similar neighbors' classes that exceed a predefined threshold as determined by Definition 8. If there is no single class that its similarity exceed the threshold, the function returns all the neighbor's classes.
  Input: $D = (C, A, R)$, query class, parent class, similarity function, threshold, K.
  Output: list of classes and their similarities from D.

3.2 The Greedy Search Algorithm

To start the execution of the Greedy Search (GS) algorithm, the number of classes (K) the algorithm handles in each round (that refers to the next step of the query) and the threshold for the similarity need to be determined. The algorithm uses the class-relationship based similarity function to find the appropriate list of classes for its execution in each iteration. We also have to determine the UCD(s) in which the search should take place and $similarClasses$ where the search should start (this task can be achieved by the class-based similarity function). For the query, we have the $queryDiagram$ that need to be found in the set of UCDs. Additional required parameters include the $queryID$ that refers to the id of the class in a $queryDiagram$ and is initialized with '-1' that represents the first class of the query where we want to start search from. $isVisited$ is a list that contains all the classes that GS algorithm already iterates over in the UCD to allow GS algorithm handles diagrams with cycles. The output of the algorithm is a set of sub-diagrams from the UCD that are the most similar to the $queryDiagram$. Algorithm 1 describes the entire procedure.

The GS algorithm steps are as following: (line 1) Initialize results with an empty set; (lines 2–4) Stop GS if there are no classes in the $similarClasses$ or there are no classes with a given ID in the $queryDiagram$; (line 5) iterate over all neighbors of the current class; (lines 6–8) skip already visited classes; (line 9) For each class in $similarClasses$ we add it to $isVisited$ list; (line 10) Get up to K most $similarClasses$ by the class-relationship based function described above; (lines 11–12) If there are no classes in $similarClasses$ or the similarity of the classes in the $similarClasses$ is below the threshold the algorithm will run recursively for the selected $similarClasses$ and $queryID$, without moving on the next class; (lines 12–14) If there are classes in $similarClasses$ and the similarity of the classes in the *similarClasses* are above the threshold - the algorithm will run recursively for the selected *similarClasses* (up to K classes) and *queryID*, each time moving on to the next class – in other words, in order to allow the return of results that are transitive, the class in the query is promoted only when the similarity obtained for the class is higher than the threshold; (line 15) Merge the results that the algorithm retrieved with the results within the function arguments. The algorithm continues until it passes all the classes and relationships of the query. At the end of the algorithm, sub diagrams from the UCD that answer the query will be obtained. The sub diagrams will be sorted by their similarity ranking, as determined by Definition 10 normalized by the number of classes. The ranking follows the weighted similarity of the relationships and classes and is represented by a number between 0 and 1.

Algorithm 1 GS in UCD

Input $similarClasses, UCD(s), queryDiagram, K, threshold,$
$queryID, isVisited$
Output Set of sub diagrams with their similarities

1: $results \leftarrow \{\theta\}$
2: **if** $similarClasses \equiv \{\theta\} \mid\mid v_{queryID} \notin V_{queryDiagram}$ **then**
3: $\quad return\ results$
4: **end if**
5: **for** $class$ in $similarClasses$ **do**
6: $\quad$ **if** $class \in isVisited$ **then**
7: $\quad\quad continue$
8: $\quad$ **end if**
9: $\quad isVisited \leftarrow isVisited \cup class$
10:
$\quad similarClasses \leftarrow class - relationship\_based\_function($
$UCD(s), queryDiagram[queryID], class, K, threshold)$
11: $\quad$ **if** $similarClasses \equiv \{\theta\} \mid\mid arg\ max_{sim \in similarClasses}(sim) \leq threshold$ **then**
$results' \leftarrow GS(similarClasses, UCD(s), queryDiagram, K,$
$th, queryID, isVisited)$
12: $\quad$ **else**
13: $\quad\quad results' \leftarrow GS(similarClasses, UCD(s), queryDiagram, K,$
$threshold, queryID++, isVisited)$
14: $\quad$ **end if**
15: $\quad results \leftarrow results' \cup results$
16: **end for**
17: $return\ results$

3.3 GS Execution Example

In the following we demonstrate the execution of GS algorithm on a UCD of a machine architecture shown in Fig. 2 with the query shown in Fig. 3a. We set the threshold to 0.65 and the number of lookup classes (K) to 2.

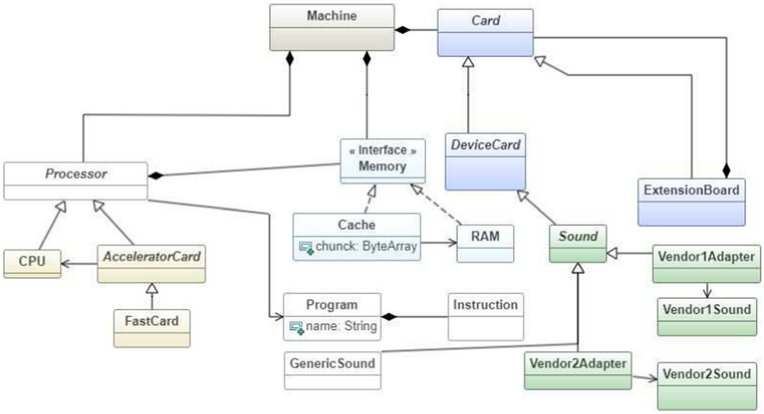
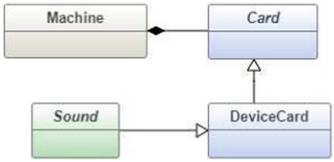


Fig. 2. The machine architecture UCD

(a) Query diagram



(b) Expected output

Fig. 3. Query and expected output

1. In the first stage, the class-based similarity function for finding the most similar class in the UCD is used to find the first class of the query diagram. The following result demonstrates the output for the class "Machine" in the query diagram:

```
[-4]: ['Cache'] :: ['Computer']
similarity: 0.624
[-5]: ['RAM'] :: ['Computer']
similarity: 0.634
[-6]: ['Machine'] :: ['Computer']
similarity: 0.809
```

   (for all other classes, the similarity was below 0.3)
2. Following the highest similarity of 0.809, the GS algorithm starts from the class "Machine". In this stage, the algorithm keeps trying to find the K (K = 2 in our case) most similar adjacent classes to the selected class in the UCD to the next class of the query diagram, that is, "Sounds". The results are the following:

```
Start finding 2 most similar classes in the UCD to the class of
a query:['Sounds']
The neighbors of vertex: -6 are: [-9, -2, -11]
[-9]: ['Memory'] :: ['Sounds']
similarity: 0.541
[-2]: ['Processor'] :: ['Sounds']
similarity: 0.583
[-11]: ['Card'] :: ['Sounds']
similarity: 0.624

Here, as no class exceed the threshold we continue with all
classes: [(0.541, -9), (0.583, -2), (0.624, -11)]
```

3. This stage splits into three paths of which the algorithm iterates over neighbors of the relevant classes ['−2'], ['−9'] and ['−11']. Here, we only show the results after iterating over class ['−11']:

```
Start finding 2 most similar classes in the UCD to the class of
a query: ['Sounds']
The neighbors of vertex: -11 are: [-10, -18]
[-18]: ['Extension', 'Board'] :: ['Sounds']
similarity: 0.574
[-10]: ['Device', 'Card'] :: ['Sounds']
similarity: 0.497
[-18]: ['Extension', 'Board'] :: ['Sounds']
similarity: 0.574
The most similar node is: [(0.497, -10), (0.574, -18)]
```

   Here again, since all the classes sim_{ne} value (Definition 8) is below the threshold, all the classes advance recursively to the next step without advancing to the next query class.
4. The GS algorithm terminates only when one of the following scenarios occur:
   (a) The GS algorithm finished iterating over the classes in the *queryDiagram* therefore, GS algorithm found one sub-diagram among all the potential sub-diagrams:

```
Start finding 2 most similar classes in the UCD to the class of
a query: ['Sounds']
The neighbors of vertex: -10 are: [-11, -12]
[-11]: ['Card'] :: ['Sounds']
similarity: 0.497
[-12]: ['Sound'] :: ['Sounds']
similarity: 0.670
The most similar node is: [(0.670, -12)]
```

   (b) There are no more classes to iterate over in the UCD. This happens due two reasons: (1) There are no neighbors; (2) The neighbors already visited by the algorithm. Therefore, all the sub-diagrams that end with class '−18' are not relevant answers:

```
Start finding 2 most similar classes in the UCD to the class of
a query: ['Sounds']
The neighbors of vertex: -18 are: []
The most similar node is: []
```

5. Finally, the results paths are calculated for their similarity based on Definition 10 and are sorted accordingly.

```
[('->Machine->Memory->Cache->RAM', 0.606),
('->Machine->Card->Device Card->Sound', 0.650),
('->Machine->Memory->Processor->Program', 0.666)]
```

   Indeed, the expected outcome that appears in Fig. 3b is included within the result set of the GS algorithm. The final similarity of the expected path is calculated by Definition 10 as follow: $\frac{0.809+0.624+0.497+0.670}{4} = 0.650$.

4 Evaluation

To evaluate the GS algorithm, we conducted an experiment to check the algorithm performance in various settings.

4.1 Settings

We considered nine domains from which we had UCDs: (1) Bank Management; (2) Library; (3) Machine; (4) Pizza store; (5) Restaurant (6) University; (7) Climate; (8) Tank; and (9) Store. Diagrams (1)–(6) were adjusted from GenMyModel[1], (7) and (8) were adjusted from [19], (9) was adjusted from [12]. The domains were chosen in order to check whether the algorithm can distinguish between domains that have similar semantic like Pizza store (5) and Store (9), and similar structure like Pizza store (5) and Library (3). Each UCD contains in average 9.44 classes and 9.55 relationships. Each class has in average 1.37 attributes and 2.04 neighbors. The label that might be a class name or an attributes name has 1.12 words in average.

We were also interested in exploring how the query complexity affects the algorithm performance. For that reason we defined three complexity categories based on two parameters:

1. **Path:** the number of paths from the first class.
2. **N-hop:** supporting hop relationship in the map according to the expected result. That is, the number of skips on the diagram relationships required to get the desired result.

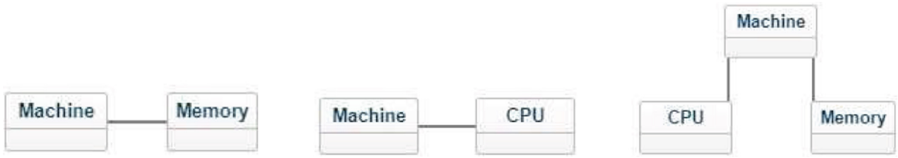
The categories were the following:

1. **Simple** the query has one path and n-hop equals to 0
2. **Medium** the query has one path and n-hop is more then 0
3. **Complex** the query has more then 1 path

[1] https://www.genmymodel.com/.

We set 20 queries that were classified into three categories of complexity: simple - 7 queries, medium - 7 queries, and complex - 6 queries.

Examples of a simple, medium, and complex queries are shown in Fig. 4. The names of the classes of the queries are matched to the class names in Fig. 2 to simplify the demonstration of how the parameters influence the difficulty of the categories. Figure 4a is a simple query because the expected result is $Machine{-}{>}Memory$ therefore, the *Path* and *N-hope* results in zero. Figure 4b is a medium query because the expected output is $Machine{-}{>}Processor{-}{>}CPU$ and as a result *N-hope*'s value is 1 and *Path*'s value remains 0. Finally Fig. 4c is a complex query due to the number of *Paths* in the expected output resulting in two, $Machine{-}{>}Memory$ *and* $Machine{-}{>}Processor{-}{>}CPU$.



(a) Simple (b) Medium (c) Complex

Fig. 4. Three complexity query level

The experiment material and results can be found in an online appendix [6]. To evaluate the performance of the GS algorithm we used the following metrics:

- **Domain Mean Reciprocal rank (Domain MRR)** indicates whether the right model was retrieved:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{5}$$

  where Q is the set of the queries, and $rank_i$ is the rank position of the first relevant domain result to the i-th query.
- **Diagram Similarity (D-sim)** that measures the similarity between a query diagram and a sub-diagram from the set of UCDs as defined in Definition 10. The D-sim takes into account the highest D-sim score of each query and ignores cases in which no answer was found.
- **Recall@k** that determines whether the expected result appears within the top k results.
- The **number of failures** in which the search did not succeed.

We executed the GS algorithm with K = 2 and searched for a threshold that maximized the **Domain MRR**. Figure 5 shows how the **Domain MRR** metric is affected by the threshold. Following the examination, it seems that a value of 0.65 leads to the best results. Therefore, we set the threshold to 0.65 and measured the metrics for each complexity category.

4.2 Results

Table 4 presents the results of the experiment. The numbers in each cell indicate the average result after running each of the relevant set of queries. Examining the Recall@1, Recall@3, Recall@5, Domain MRR and similarity metrics. The results indicate that the algorithm's performance decreases as the query complexity increases.

Table 4. Results

Complexity	Recall@1	Recall@3	Recall@5	Domain MRR	D-sim	Search failure	TOT. Answers
Easy	0.571	**0.857**	0.857	0.801	0.783	1	7
Medium	0.571	**0.714**	0.714	0.785	0.675	2	7
Complex	0.333	0.333	0.416	**0.509**	0.813	7	12
Mean	0.490	0.634	0.662	**0.698**	0.757	3.33	–
Std	0.113	0.221	0.183	0.134	0.059	2.62	–

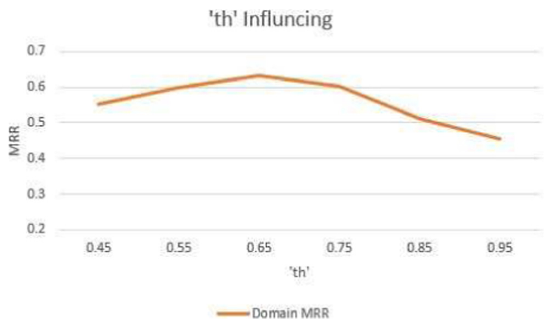


Fig. 5. 'th' influencing on Domain MRR

4.3 Discussion

In this section, we discuss the results with respect to the parameters of the GS algorithm and the input characteristics.

In the experiment we set the number of neighbours (K) to 2. It might be beneficial to increase K in correlation with the actual neighbours of each class (In our case, it was 2.04 in average). This might increase the accuracy of the search and allow the exploration of additional paths.

The threshold parameter also affects the results as it determines the relevant classes from which the algorithm moves forward. It might be that the threshold should be changed based on the domain and the sim_w function. The semantic of two classes (sim_l) depends on two parameters: similarity between two words (sim_w) and the length of a class's label in terms of how many words the label contains. For sim_l and sim_{att} we noticed that it is affected by the number of words. In the case of long labels the similarity decreases.

4.4 Threats to Validity

The initial evaluation we performed should be taken with caution and should consider the following threats to validity:

- **Small UCD:** The diagrams in the experiment are of limited size (9.44 classes and 9.55 relationships in average). We should explore the algorithm with much larger diagrams.
- **Self authoring:** We as the authors of the paper, developed the queries, so some biases might exist. Nevertheless, our aim in this evaluation was to challenge the algorithm, so the queries we devised aimed to accomplish that.
- **Simple queries:** Both the domains and the queries are quite small. There is a need to incorporate more complex domains and queries.
- **Lack of tuning:** Based on informal experience, we set the algorithm parameters as constants. Yet, there is a need to perform sensitivity analysis to learn about the impact of these parameters.
- **Comparing to other works:** Indeed, the results should be compared to other alternatives. Yet, such alternatives need to be adjusted for our UCDs repository. For example, converting UCDs into corresponding ontology graph and then run an ontology matching algorithm.

5 Summary

In this paper, we propose a greedy algorithm for searching UCDs. The algorithm takes into structure, semantic, and type similarity compering to other works shown in Table 1. The initial evaluation we preformed shows a promising results in terms of accuracy.

Nevertheless, in the future, we intend to improve the sim_l function using text mining techniques like RNN [23] and LSTM [9] for sentence similarity task. Moreover, we want to test how well the algorithm performs with other domains and examine alternatives for calculating the similarity of the classes and relationships. This includes the tuning of the algorithm's parameters, either a-priory or during its execution.

Although we evaluate the algorithm on UCDs, we believe it can be used for searching other models. Thus, we also plan to test our conjecture regarding this matter.

References

1. Abrahão, S., et al.: User experience for model-driven engineering: challenges and future directions. In: 2017 ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS), pp. 229–236 (2017)
2. Agt-Rickauer, H., Kutsche, R.-D., Sack, H.: DoMoRe? A recommender system for domain modeling. In: The 6th International Conference on Model-Driven Engineering and Software Development, pp. 71–82 (2018)

3. Al-Khiaty, M.A.-R., Ahmed, M.: Similarity assessment of UML class diagrams using a greedy algorithm. In: 2014 International Computer Science and Engineering Conference (ICSEC), pp. 228–233. IEEE (2014)
4. Al-Khiaty, M.A.-R., Ahmed, M.: UML class diagrams: similarity aspects and matching. Lect. Notes Softw. Eng. **4**(1), 41 (2016)
5. Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A., Raghavan, S.: Searching the web. ACM Trans. Internet Technol. (TOIT) **1**(1), 2–43 (2001)
6. Bargilovski, M., Makias, Y., Shamshila, M., Stern, R., Sturm, A.: Searching Models, March 2021. https://doi.org/10.17632/6685g76r9y.1
7. Basciani, F., Di Rocco, J., Di Ruscio, D., Iovino, L., Pierantonio, A.: Model repositories: will they become reality? In: The 3rd International Workshop on Model-Driven Engineering, pp. 37–42 (2015)
8. Hebig, R., Quang, T.H., Chaudron, M.R.V., Robles, G., Fernandez, M.A.: The quest for open source projects that use UML: Mining GitHub. In: The ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems, MODELS 2016, pp. 173–183 (2016)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
10. Lau, C.: Reusing code in object-oriented program development. US Patent 6,182,274, 30 January 2001
11. López, J.A.H., Cuadrado, J.S.: MAR: a structure-based search engine for models. In: The 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, pp. 57–67 (2020)
12. Martínez, F.J.L., Álvarez, A.T.: A precise approach for the analysis of the UML models consistency. In: Akoka, J., et al. (eds.) ER 2005. LNCS, vol. 3770, pp. 74–84. Springer, Heidelberg (2005). https://doi.org/10.1007/11568346_9
13. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity flooding: a versatile graph matching algorithm and its application to schema matching. In: Proceedings of the 18th International Conference on Data Engineering, pp. 117–128. IEEE (2002)
14. Mili, H., Mili, F., Mili, A.: Reusing software: issues and research directions. IEEE Trans. Softw. Eng. **21**(6), 528–562 (1995)
15. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
16. Nikiforova, O., Gusarovs, K., Kozacenko, L., Ahilcenoka, D., Ungurs, D.: An approach to compare UML class diagrams based on semantical features of their elements. In: The Tenth International Conference on Software Engineering Advances, pp. 147–152 (2015)
17. Řehůřek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: The LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, pp. 45–50. ELRA, May 2010
18. Reinhartz-Berger, I.: Towards automatization of domain modeling. Data Knowl. Eng. **69**(5), 491–515 (2010)
19. Reinhartz-Berger, I., Sturm, A.: Utilizing domain models for application design and validation. Inf. Softw. Technol. **51**(8), 1275–1289 (2009)
20. Robertson, S., Zaragoza, H.: The Probabilistic Relevance Framework: BM25 and Beyond. Now Publishers Inc., Hanover (2009)
21. Robles, G., Ho-Quang, T., Hebig, R., Chaudron, M.R.V., Fernandez, M.A.: An extensive dataset of UML models in GitHub. In: Proceedings of the 14th International Conference on Mining Software Repositories, MSR 2017, pp. 519–522. IEEE Press (2017)

22. Robles, K., Fraga, A., Morato, J., Llorens, J.: Towards an ontology-based retrieval of UML class diagrams. Inf. Softw. Technol. **54**(1), 72–86 (2012)
23. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. Technical report, California University San Diego La Jolla Inst for Cognitive Science (1985)
24. Salami, H.O., Ahmed, M.: Retrieving sequence diagrams using genetic algorithm. In: 2014 11th International Joint Conference on Computer Science and Software Engineering (JCSSE), pp. 324–330. IEEE (2014)
25. Steinberg, D., Budinsky, F., Merks, E., Paternostro, M.: EMF: Eclipse Modeling Framework. Pearson Education, London (2008)
26. Yuan, Z., Yan, L., Ma, Z.: Structural similarity measure between UML class diagrams based on UCG. Requirements Eng. **25**, 213–229 (2020). https://doi.org/10.1007/s00766-019-00317-w
27. Zhu, G., Iglesias, C.A.: Sematch: Semantic similarity framework for knowledge graphs. Knowl.-Based Syst. **130**, 30–32 (2017)

From Mental Models to Machine Learning Models via Conceptual Models

Wolfgang Maass[1], Veda C. Storey[2(✉)], and Roman Lukyanenko[3]

[1] Saarland University and German Research Center for Artificial Intelligence (GFKI), 66123 Saarbrücken, Germany
wolfgang.maass@iss.uni-saarland.de

[2] Georgia State University, Altanta, GA, USA
vstorey@gsu.edu

[3] HEC Montreal, Montreal, QC, Canada
roman.lukyanenko@hec.ca

Abstract. Although much research continues to be carried out on modeling of information systems, there has been a lack of work that relates the activities of modeling to human mental models. With the increased emphasis on machine learning systems, model development remains an important issue. In this research, we propose a framework for progressing from human mental models to machine learning models and implementation via the use of conceptual models. The framework is illustrated by an application to a citizen science project. Recommendations for the use of the framework are proposed.

Keywords: Machine learning · Mental models · Conceptual models · Data scientist · Citizen science

1 Introduction

Machine learning has continued to progress as a valued way to support decision making due to its ability to process large amounts of data, by extracting complex rules from those data. The models themselves are developed by data scientists who must possess both an understanding of the application domain and the mathematical models and algorithms needed to build the machine learning based systems. For data scientists, having a mental model of the application domains is crucial in order to avoid biases and mistakes in the results. This topic has emerged as the "black box" of artificial intelligence (AI) or explainable AI.

Conceptual models are used to capture and represent the parts of the real world that need to be included in an information system [1, 2]. Research on conceptual modeling has evolved over the past four decades from modeling database management systems to providing a mechanism for understanding the real world and abstracting concepts from the real world that are important for inclusion in an information system [3].

The objectives of research are to: recognize the important role that mental models play in the development of machine learning based information systems; analyze mental

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 293–300, 2021.
https://doi.org/10.1007/978-3-030-79186-5_19

models as input to conceptual models; and propose how translating mental models into conceptual modeling can support machine learning. To carry out the research, we examine relevant literature in each of these areas and integrate them. The paper contributes a framework that captures end-to-end progression from the needed mental models of a data scientist to the creation of effective machine learning models. The framework is applied to an example to illustrate its effective use and recommendations made for its further application and development.

This paper proceeds as follows. Section 2 defines and reviews related concepts which are integrated into a framework in Sect. 3. Section 4 applies the framework and discusses its implications. Section 5 summarizes and concludes the paper.

2 Related Research

2.1 Mental Models

Mental models are mental representation of reality, the relationships between its various parts and a person's intuitive perception about his or her own acts in the world and their consequences [4–6]. Although they have been identified as useful for research in information systems, they remain an under-studied area. Instead, much work on systems analysis and design starts with the notion of extracting user requirements and then representing them in a conceptual model before they are translated into a form that is useful for implementation. In machine learning applications, this means transforming them into a format that can be used in machine learning algorithms and processes.

The theory on mental models is based on three assumptions [20]: (1) mental models represent what is *common* to a distinct set of possibilities; (2) mental models are *iconic,* that is, the structure of a model selectively conceives the structure of what it represents; and (3) mental models of descriptions represent what is *true* at the expense of what is *false*. Mental models are used for human reasoning based on deductive inference [23] and probabilistic inferences. Conception and use of mentals models suffer from biases, illusions, emotions and limitation of cognitive resources but help humans to draw conclusions by mixing deduction, induction and abduction [5].

Because of the complexity of mental models and human reasoning, conceptual modeling is a difficult process that tries to extract representations from individual mental models and negotiate a common understanding between members of a conceptual modeling team [22]. Thus, conceptual modeling is grounded in complex cognitive processes that start with the creation of individual mental models [27], which we can then translate into a conceptual model.

2.2 Conceptual Modeling

Conceptual modeling is often referred to as modeling: "some aspect of the physical and social world around us for the purposes of understanding and communication" (p. 289) [4]. Conceptual models attempt to represent user requirements of an application domain, with the purpose of creating a shared understanding among designers and users of an information system within given boundaries or application domains. Conceptual models help to structure reality by abstracting the relevant aspects of an application domain, while ignoring those that are not relevant. They can structure concepts into hierarchies, or simply identify and label associations among concepts in the real world.

A conceptual model formally represents requirements and goals. It is influenced by the perspective of the cognitive agents whose mental representations it captures and, in this sense, is a *social artifact* that is intended to capture the shared conceptualization of a group [7]. Conceptual modeling is well-recognized as being complex, but important.

Conceptual modeling has been influenced by various disciplines including software engineering, requirements engineering, psychology, and philosophy. Its modeling activities and methods have been applied to a wide range of domains and problems [10]. Jaakkola and Thalheim [25] highlight the importance of modeling on the development of artificial intelligence and machine learning tasks, with research emerging that identifies conceptual modeling as a way to support machine learning [8–11]. Conceptual models enable humans to gain an "intuitive, easy to understand, meaningful, direct and natural mental representation of a domain" [7]. In contrast, machine learning uses data as a way to identify regularities and patterns in data taken from a domain [14–16].

2.3 Machine Learning

Machine learning enables computers to learn from experience by applying statistical methods. While early machine learning systems used only low-level data, deep learning models attempt to learn concept hierarchies with concepts learned from simpler concepts, anchored in raw data [24]. Machine learning models are designed by four parameter sets: (1) data, (2) model architecture, (3) hyperparameters and (4) objective functions. Conceptual knowledge is only indirectly used for selecting and pre-processing data and selecting and designing model architectures. Therefore, embedding conceptual knowledge into machine learning is considered a "black art." It is further complicated by the general challenges associated with modeling a real-world application.

The development of any information system requires understanding and representing the real world, which is the role of conceptual modeling. Modeling is especially important for capturing and representing the complexity found in the development of artificial intelligence and machine learning tasks. Thus, incorporating conceptual modeling into machine learning activities should improve machine learning due to the emphasis of conceptual modeling on accurately modeling the real world. Even if data scientists who use the models, do not need to understand necessarily how to create conceptual models, they can use the conceptual models as a communication vehicle.

3 Framework for Mental Models to Machine Learning Models

The origin of data used in machine learning is often independent of the purpose of a machine learning task. For instance, sensor data in medicine or industry used as input data is framed by technical specifications and pictures on social media follow user motivations. Thus, the logic behind data is usually not transparent to data scientists leading to misunderstandings and biases. In essence, data scientists simply do not have the knowledge or time to deeply understand data and analytical tasks. Therefore, they take the data as-is and see "what the data is telling." Conceptual models are an important layer of shared human knowledge that qualifies data, provides logical structures used for interpretation of input and output data, and, thus, enables data scientists to deeply understand their tasks from a point of view that abstracts from data.

To assist in creating machine learning models that reflect the real world, as understood by a data scientist, Fig. 1 (adapted from [21]) provides a framework that captures the potential interactions among mental models, conceptual models, and machine learning (ML) models. Mental models are formed within a domain by a data scientist. Data scientists need to understand the problem being solved and the domain in which it occurs, as well as the potential machine learning models and methods that could be applied. Data scientists often acquire domain knowledge through interaction with domain experts, which they form into their own mental models. Conceptual models represent a shared conceptualization about an application using representation constructs, methods, and rules. Traditionally, conceptual models are transformed into logical models for implementation in a database. For machine learning-based systems, the relationships among the data, the conceptual models, and the machine learning models need to ensure that the machine learning models are appropriate for the application.

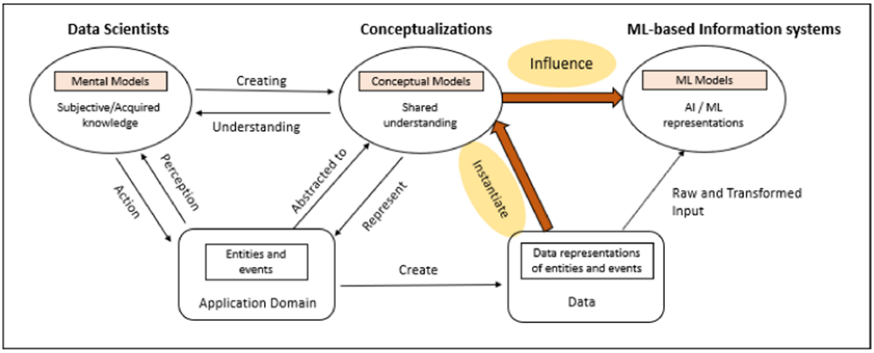


Fig. 1. Framework progressing mental models to machine learning models (adapted from [21])

The translation of conceptual models into database systems is based on models, such as the entity-relationship (ER) model. Current machine learning models cannot apply ER models, but use tabular data, 2D images or graphs as input [28]. The structure of the input data is often provided by a data scientist with only short textual descriptions of data features. For instance, the popular HR Analytics[1] dataset on Kaggle.com provides 13 input features including the following descriptions: "*enrolled_university: Type of University course enrolled if any*" and "*experience: Candidate total experience in years.*" It is unclear whether it considers only US universities or whether courses in mathematics at high school count as experience. Conceptual models should clarify.

For tabular data, the modeling decision involves the selection of data features. For instance, data features can represent facts, such as age, time series, categorical data or unstructured text. Two-D image data is structured as a matrix whose values are binary

[1] https://www.kaggle.com/arashnic/hr-analytics-job-change-of-data-scientists.

(black/white), grey or RGB. For graph data, data scientists make modeling decisions on which feature is represented as nodes and what is captured, by edges. In social network analysis, humans can be thought of as nodes and geographical proximity, the edges, for social network analysis. There are all decisions that must be made by data scientists based on their expertise. Although conceptual modeling abstracts from data and focusses on entities and relationships, machine learning emphasizes the importance of data for finding the most appropriate data features and data transformations.

4 Application of the Framework: Citizen Science Case

To illustrate the applicability and the value of our framework we present an application of machine learning in the context of citizen science. Citizen science refers to participation of the members of the general public (citizens) in scientific research [12–14].

Citizen science is emerging as a major societal movement and research approach, based on the support of regular citizens for data gathering and analysis. In biodiversity research, for example, it was estimated that, as of 2015, more than two million people were engaged in citizen science projects contributing up to $2.5 billion of in-kind value [15]. Citizen science has led to numerous discoveries, including of new exoplanets, biological species, novel celestial bodies, historical or geological phenomena [16].

As human society continues to face existential challenges, citizen science is increasingly viewed as an approach which can support addressing these challenges [17, 18]. This includes tackling humanity's "evil quintet" of climate change, overexploitation, invasive species, land use change, and pollution [15]. As Light and Miskelly [19] assert "[t]he urgency of environmental issues draws our attention to the management of finite resources, and the potential of digital tools to help us work with them effectively."

To demonstrate the application of our framework, we consider a case of citizen science, based on one of the author's own experience of developing a real citizen science project. The objective of this project is to map biodiversity of a region in North America with the sightings of wildlife by ordinary people; that is, citizens. The project has been online since 2010, and resulted in a large data set of observations, making it a prime target for the application of machine learning.

Machine learning can find additional patterns in the data provided by the citizens; for example, for effective environmental and conservation policies. One potential application of this data set is to predict the likelihood of animal encounters next to human infrastructure. For example, the likelihood of a particular kind of animal (e.g., a polar bear), appearing in the vicinity of a waste treatment facility. Such predictions can help to better plan infrastructure to reduce human encroachment into animal habitat and to minimize threats to animals due to dangerous infrastructure.

Mental Model. A data scientist must first form a mental model of the domain. Here, there are two focal domains: the domain of plants and animals; and the domain of human infrastructure. Each mental model is comprised of theories (e.g., of animal behavior and interaction with human artifacts), assumptions (e.g., some animals can learn with sufficient reinforcement), and conceptual structures (e.g., properties and kinds of infrastructure) held by the data scientists. These mental models are commonly incomplete or inaccurate. This is due to a natural lack of deep application domain knowledge by data scientists, who are trained in data management and machine learning techniques. If machine learning solutions are developed directly based on these "naive" mental models, the result could be biased or suboptimal.

Conceptual Models. As our Framework suggests, conceptual models can be used to remedy the lack of deep domain knowledge on the part of data scientists. Conceptual models make the mental models of data scientists explicit and hence verifiable and transparent. This allows both the data scientists and other stakeholders (e.g., domain experts such as biologists or infrastructure planners) to scrutinize the conceptual models (and hence, indirectly, the mental models of data scientists) and find gaps and biases.

In our citizen science application, assume that a data scientist uses own mental model to analyze the domain and determine whether a particular kind of infrastructure is dangerous or safe for different kinds of animals. Doing so, may result in the identification of common types of dangerous infrastructures for animals. For example, artificial dams may prevent fish from spawning, high velocity boats are known to damage whales and dolphins, and garbage treatment facilities may attract polar bears, which may stray then off their normal hunting grounds. Another common example is highways which are dangerous for most land mammals. Such mental models may help the data scientist obtain the requisite training data for the machine learning applications (by augmenting the sightings provided by the citizens with the information on the location of highways).

Creating a conceptual model would externalize the mental models by the data scientist, and subject them to external scrutiny. This might reveal important gaps in the domain knowledge of the data scientist. For example, contrary to a common misperception, birds are not safe from electricity and are commonly electrocuted on high power voltage lines. The absence of information on high-power voltage lines could easily be spotted by examining the entities in a conceptual model by the domain experts, who are, presumably, aware of this danger to birds. Upon making this observation known to data scientists, data scientists can update their mental model, and then, seek more representative and comprehensive training data to build the machine learning solutions.

Machine Learning Models. A data scientist can now train the machine learning algorithms using, for example, data that includes sightings of birds near high-voltage power lines. The result is a more accurate and unbiased machine learning solution, capable of better predicting the likelihood of encounters of animals with a dangerous human infrastructure.

5 Discussion

To develop machine learning solutions, data scientists must rely on their own mental models of the domains to identify relevant sources for the development (e.g., training, validation) of machine learning models, and to perform appropriate actions upon these data (e.g., data transformations). Generally, data scientists are non-domain experts, so their mental models of the requisite domains may not always be accurate, complete, or bias free. To rectify this problem, we proposed to use conceptual models – information technology artifacts especially tailored to representing mental models. Traditionally, conceptual models have been used to capture information systems requirements to guide database design and process engineering. However, the benefits from using conceptual modeling, although mainly applicable to selected contexts, are quite general.

By using conceptual models, data scientists can externalize their own mental models. Domain experts, and others, can examine the conceptual models, and indirectly, the mental models of data scientists. In the citizen science example, using conceptual models when forming solutions, enables experts to identify deficient mental models of data scientists, and build more representative and accurate machine learning models.

6 Conclusion

Machine learning applications continue to be widely developed and applied. One of the greatest challenges is modeling the application domain for which the machine learning applications will be used. This research proposes a framework for progressing from the mental models that data scientists create to representing them as conceptual models that supporting machine learning. The elements of the framework have been applied to a citizen science application to clarify the type of modeling applications for which the framework could be useful. Future research is needed to apply the framework to different applications and to assess each of the individual components.

References

1. Recker, J., Lukyanenko, R., Sabegh, M.A., Samuel, B.M., Castellanos, A.: From representation to mediation: a new agenda for conceptual modeling research in a digital world. MIS Q. **45**, 269–300 (2021)
2. Wand, Y., Weber, R.: Research commentary: Information systems and conceptual modeling-a research agenda. Inf. Syst. Res. **13**, 363–376 (2002)
3. Storey, V.C., Trujillo, J.C., Liddle, S.W.: Research on conceptual modeling: themes, topics, and introduction to the special issue. Data Knowl. Eng. **98**, 1–7 (2015)
4. Gentner, D., Stevens, A.L.: Mental Models. Psychology Press, New York (2014)
5. Johnson-Laird, P.N., Wason, P.C.: Thinking: Readings in Cognitive Science. Cambridge University Press, Cambridge (1977)
6. Jones, N.A., Ross, H., Lynam, T., Perez, P., Leitch, A.: Mental models: an interdisciplinary synthesis of theory and methods. Ecol. Soc. **16**, 46–46 (2011)
7. Guarino, N., Guizzardi, G., Mylopoulos, J.: On the philosophical foundations of conceptual models. Inf. Model. Knowl. Bases **31**, 1 (2020)
8. Fettke, P.: Conceptual modelling and artificial intelligence: overview and research challenges from the perspective of predictive business process management. Presented at the Modellierung (Companion) (2020)
9. Lukyanenko, R., Castellanos, A., Parsons, J., Chiarini Tremblay, M., Storey, V.C.: Using conceptual modeling to support machine learning. In: Cappiello, C., Ruiz, M. (eds.) Information Systems Engineering in Responsible Information Systems. LNBIP, vol. 350, pp. 170–181. Springer, Cham (2019)
10. Reimer, U., Bork, D., Fettke, P., Tropmann-Frick, M.: Preface of the first workshop models in AI. Presented at the Modellierung (Companion) (2020).
11. Bork, D., Garmendia, A., Wimmer, M.: Towards a Multi-Objective Modularization Approach for Entity-Relationship Models. ER Forum, Demo and Posters (2020)

12. Bonney, R., et al.: Next steps for citizen science. Science **343**, 1436–1437 (2014)
13. Levy, M., Germonprez, M.: The potential for citizen science in information systems research. Comm. Assoc. Inf. Syst. **40**, 2 (2017)
14. Show, H.: Rise of the citizen scientist. Nature **524**, 265 (2015)
15. Theobald, E.J., et al.: Global change and local solutions: tapping the unrealized potential of citizen science for biodiversity research. Biol. Cons. **181**, 236–244 (2015)
16. Lukyanenko, R., Wiggins, A., Rosser, H.K.: Citizen science: an information quality research frontier. Inf. Syst. Front. **22**(4), 961–983 (2019). https://doi.org/10.1007/s10796-019-09915-z
17. Burgess, H., et al.: The science of citizen science: exploring barriers to use as a primary research tool. Biol. Cons. **208**, 1–8 (2017)
18. McKinley, D.C., et al.: Citizen science can improve conservation science, natural resource management, and environmental protection. Biol. Conserv. **208**, 15–28 (2016)
19. Light, A., Miskelly, C.: Design for Sharing. Northumbria University/The Sustainable Society Network, Newcastle upon Tyne (2014)
20. Johnson-Laird, P.N.: Mental models and human reasoning. Proc. Natl. Acad. Sci. **107**(43), 18243–18250 (2010)
21. Maass, W., Storey, V.C.: Pairing Conceptual Modeling with Machine Learning. Data and Knowledge Engineering (2021). Forthcoming
22. Maass, W., Storey, V.C., Kowatsch, T.: Effects of external conceptual models and verbal explanations on shared understanding in small groups. In: Jeusfeld, M., Delcambre, L., Ling, T. (eds.) ER 2011. LNCS, vol. 6998, pp. 92–103. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24606-7_8
23. Johnson-Laird, P.N.: Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness. Harvard Univ Press, Cambridge, MA (1983)
24. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
25. Jaakkola, H., Thalheim, B.: Sixty years–and more–of data modelling. Inf. Model. Knowl. Bases XXXII **333**, 56 (2021)
26. Mylopoulos, J., Chung, L., Nixon, B.: Representing and using nonfunctional requirements: a process-oriented approach. IEEE Trans. Softw. Eng. **18**(6), 483–497 (1992)
27. Pastor, O., Conceptual modeling of life: beyond the homo sapiens. In: Comyn-Wattiau, I., Tanaka, K., Song, I.Y., Yamamoto, S., Saeki, M. (eds.) Conceptual Modeling. ER 2016. LNCS, vol. 9974. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46397-1_2
28. Zhou, J., et al.: Graph neural networks: a review of methods and applications. AI Open, **1**, 57–81 (2020)

Threat and Evidence Modeling (EMMSAD 2021)

Towards Measuring Test Coverage of Attack Simulations

Nicklas Hersén[1], Simon Hacks[1](✉), and Konrad Fögen[2]

[1] Division of Network and Systems Engineering, KTH Royal Institute of Technology, Stockholm, Sweden
{nhersen,shacks}@kth.se

[2] Research Group Software Construction, RWTH Aachen University, Aachen, Germany
foegen@swc.rwth-aachen.de

Abstract. Designing secure and reliable systems is a difficult task. Threat modeling is a process that supports the secure design of systems by easing the understanding of the system's complexity, as well as identifying and modeling potential threats. These threat models can serve as input for attack simulations, which are used to analyze the behavior of attackers within the system. To ensure the correct functionality of these attack simulations, automated tests are designed that check if an attacker can reach a certain point in the threat model. Currently, there is no way for developers to estimate the degree to which their tests cover the attack simulations and, thus, they cannot the determine the quality of their tests. To resolve this shortcoming, we analyze structural testing methods from the software engineering domain and transfer them to the threat modeling domain by following an Action Design Research approach. Further, we develop a first prototype, which is able to assess the test coverage in an automated way. This will enable threat modeler to determine the quality of their tests and, simultaneously, increase the quality of the threat models.

Keywords: Threat modeling · Attack simulations · Testing · Test coverage

1 Introduction

Designing secure and reliable systems is a difficult task and attacker constantly find new ways to break into and compromise digital systems and infrastructures. Threat modeling is a process that involves understanding the complexity of a system, as well as identifying and modeling potential threats to the system or individual components [26]. The process involves modeling all potential weaknesses, threats, and vulnerabilities of a system. Often these are presented by means of graphs, where each of these is modeled as a single node and the transitions between them are expressed as edges [18]. Given a specific threat model of a system, attack simulations can be used to analyze specific attack scenarios on the described infrastructure [4,9]. The initial capabilities of an adversary are

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 303–317, 2021.
https://doi.org/10.1007/978-3-030-79186-5_20

described as a set of initially compromised exploits or vulnerabilities, which in turn lead to other vulnerabilities the attacker can exploit in order to gain access to specific resources.

The meta attack language (MAL) [10] is a meta-language used to develop domain-specific languages (DSL) for threat modeling and defines which information about a system is required. Moreover, it specifies the generic attack logic. Then, a compiler can be used to generate a Java library from a MAL-based DSL, which can be used to programmatically create threat models. These models themselves are then used to perform attack simulations. More advanced simulations and analysis of threat models can be performed by using the generated jar file in applications such as securiCAD [4].

To ensure the quality of the created MAL-based DSLs, the language developers design tests that check the desired behavior of the language. Basically, they define a threat model and assess if the attack simulations on this model behave as expected. Currently, there is no way for developers to estimate how much of their created threat models are covered by existing tests. They can only manually assert whether specific attack steps are covered by the simulation or not. As the complexity of a threat model increases, it becomes harder to keep track of what parts have been tested resulting in many unnecessary overlapping tests. Furthermore, it is hard to determine how conclusive a test suite is without any type of coverage metric for a given threat model. Moreover, to the best of our knowledge, there exist no proposals how any test coverage for threat models could be measured nor is there a mapping from established coverage metrics.

Thus, we aim to develop a method to measure test coverage for threat models. In a first step, we develop a method to assess the coverage for a certain set of attack simulation to achieve this overall goal. Therefore, we consider classical coverage measurements from the software engineering domain and transfer them to attack simulations. Further, we implement an extension for the MAL that allows to automate the assessment of these test coverage.

The rest of the work is structured as follows. Next, we illustrate how we applied action design research to develop the coverage metrics presented in the section thereafter. This is followed by a introduction to the MAL and a description of the prototype to measure the coverage for attack simulations based on MAL. Finally, we give the related work before we conclude.

2 Method

In our research, we opt for an Action Design Research (ADR) [24] approach since we want to ensure a continuous feedback flow from the developer of MAL-based languages to the research team. ADR is characterized by four stages. Next, we illustrate how these stages are reflected in our research.

Stage 1: Problem Formulation. Our research is driven by the need for a structured evaluation technique that helps to ensure the quality of MAL-based languages. To the best of our knowledge, there exists no proposal for such measurement nor are the classical means of coverage assessment directly applicable, as the coverage is related to model instances and not a given source code. Therefore, we identify the common properties of attack graphs that are used for the attack simulations and research on common testing strategies within the software engineering domain. We opt for the software engineering domain because the identified properties of the attack graphs remind of logical flows in computer programs. Further, there exists a broad range of established tools for measuring coverage in this domain and we assume that we can easily adapt them to the properties of attack graphs.

Stage 2: Building, Intervention, and Evaluation. In stage 2, we follow an IT-dominant approach. Using the terminology of ADR, one author act as researcher who is developing the test coverage metrics and a tool to measure them, while two authors provide their input as practitioners in the context of MAL-based language development and testing of software. Together, this team develops two kinds of artefacts: the coverage metrics and a tool to measure them. The end-users are presented by a set of MAL-based language developers who will use the tool in future to assess the coverage of their implemented test cases. Consequently, the coverage metrics are first evaluated with the authors representing the practitioners and, second, with the developers.

Stage 3: Reflection and Learning. While in stage 2 a specific solution for a certain problem is developed, stage 3 focuses on the generalization of the solution so that it covers a class of problems. To achieve this generalization, we invert the proposed proceeding by developing first a general solution for measuring test coverage in attack graphs. Then, we implement a specific solution, which measures the test coverage in concrete attack graphs in relation to MAL.

Stage 4: Formalization of Learning. The outcomes of stage 3 are formalized by a set of four different coverage metrics that can be reused in all contexts of attack simulation and their testing.

3 Test Coverage for Attack Simulations

In this section, we define first the fundamental properties of threat models and attack simulations. Then, we relate the concept of coverage metrics to these properties and, finally, discuss different metrics we developed for attack simulations.

3.1 Fundamental Properties of Threat Models and Coverage

To transfer the general concept of coverage to the domain of attack simulations, we need first to understand the fundamental properties of attack graphs and threat models, which serve as input for the attack simulations.

Threat modeling involves understanding the complexity of a system and identifying all possible security vulnerabilities regardless of whether they can be exploited [18]. This involves identifying sensitive parts of the system, such as possible attack entry point and sensitive resources. This process provides the developers with a high-level perspective of the system, which may be used to establish new security requirements, mitigate potential threats, or serve as a general guideline during the development process [21].

Attack graphs are a type of threat modeling used to describe vulnerabilities and potential exploits of a system. Formally, we understand an attack graph as a graph $G = (A, D, E, \mathfrak{A})$, with A a set of nodes representing attack steps, D a set of nodes representing defenses, $\mathfrak{A}$ a set of assets, and E a set of edges linking either attack steps to each other or a defense to an attack step: $e \in E \subseteq \{(a, b) : a \in A \bigcup D, b \in A\}$. Attack steps represent attacks, exploits, or vulnerabilities that can be leveraged in order to reach another system state [25]. An attack step $a \in A$ has a function $\tau : A \rightarrow \{\texttt{AND}, \texttt{OR}\}$ denoted by $\tau(a)$. `OR`-type attack steps are only considered compromised if either of its parents are compromised, or if the step was initially compromised at the beginning of the simulation. In contrast, `AND`-type steps considered compromised if all parent attack steps have been compromised, or if its initial state was compromised.

A defense $d \in D$ has a function $\alpha : D \rightarrow \{\texttt{TRUE}, \texttt{FALSE}\}$ denoted by $\alpha(d)$, which codifies if the attacker is hindered to assume related attack steps if `TRUE`[1]. Lastly, the single nodes are grouped that they are related to certain assets that represent the fundamental components in the system under inspection [4]. These components can be any kind of hardware or software, depending on the desired abstraction level. Thus, we define an asset $\mathfrak{a} = (A', D', E') \mid \mathfrak{a} \in \mathfrak{A}, A' \subseteq A, D' \subseteq D, E' \subseteq E$ as a sub-graph representing a certain attack graph related to a cohesive component. All assets are distinct to each other, $\mathfrak{a}_i \bigcup \mathfrak{a}_j = \emptyset \mid \forall\ i \neq j$, so they share neither attack steps, defenses, nor edges. However, edges between the attack steps and defenses of two different assets are still possible that are not part of any asset, but of the overall attack graph G.

An attack simulation is a function $\phi : \{G, A_s, D_a \mid A_s \subseteq A, D_a \subseteq D\} \rightarrow \{A_c, E_c \mid A_c \subseteq A, E_c \subseteq E\}$ denoted by $\phi(G, A, D)$. This function maps an attack graph, G, with a set of attack steps, A_s, that represent the attack surface of the attacker, and a set of defenses, D_a, that are in place, to a set of attack steps, A_c, and edges, E_c, that are compromised by the attacker.

With the fundamental properties of attack graphs and threat models being introduced, we now adapt the concept of coverage from the domain of software testing to provide information about what parts of the threat models are covered by an attack simulation.

Many structural software testing methods can be adapted to work with threat models and the generated simulation data. Common structural coverage criteria are *node coverage, edge coverage* and various types of *path coverage* [1]. In general, coverage is defined as the percentage of covered requirements or entities [17]:

$$\text{Coverage} = 100\% \cdot \frac{\textit{number of covered entities}}{\textit{total number of entities}} \tag{1}$$

[1] We are aware that defenses might be effective by a probability. However, testing non-deterministic behavior is challenging and, thus, we make the assumption that a defense is either effective or not.

Software test data selection and generation is a big research topic and a lot of methods for producing valid and reliable test data have been proposed [5]. A poorly designed test suite may provide complete coverage of some software according to some model, but still fail to catch any errors. If code for dealing with edge cases is missing, structural coverage metrics will not account for these errors in the first place.

Similarly, the coverage results from attack simulations describe the coverage of the abstract threat model given some preconditions (A_s, D_a). If the threat model does not adequately describe a specific system or infrastructure the results of attack simulations on that model might not correspond to the capabilities of an adversary attacking the actual system.

Testing the threat modeling could be considered as a system level test. The security researcher attempts to measure the capabilities of an attacker on an abstract model of a system or infrastructure. System level tests, and thus testing threat modeling, share a lot of similarities with code-based testing, the primary difference is where the structures representing different coverage metrics come from [20]. The coverage metrics essentially use common structure analysis techniques of the object under inspection, such as branch and node coverage with different coverage metrics [1].

3.2 Exemplary Model

Following, we illustrate an example that we use to demonstrate the calculation of the different coverage metrics for a specific threat model. The simulated scenario covers a situation where an adversary attempts to gain access to a target host PC protected by a password set by a specific user. The attack graph, $G^{ex} = (A^{ex}, D^{ex}, E^{ex}, \mathfrak{A}^{ex})$, contains three assets, $\mathfrak{A}^{ex} = \{Host, Password, User\}$, with connected attack steps, $A^{ex} = \{authenticate, connect, ...\}$, and a defense, $D^{ex} = \{encrypted\}$, as shown in Fig. 1. For the means of visualization, $\tau(a) = \texttt{OR}$ is visualized using "|", $\tau(a) = \texttt{AND}$ using "&", and a defense as "#".

Initially the attacker is only able to connect to the target machine and is not able to influence the user in any way, only the connect and the authenticate step of the target machine are initially compromised $\phi(G^{ex}, \{connect, authenticate\}, D_a^{ex})$. Two scenarios are simulated, one where the password is encrypted, $D_a^{ex} = \{encrypted\}$, and one where the password is not encrypted, $D_a^{ex} = \emptyset$.

In the following, we introduce new coverage metrics to calculate the coverage specific to attack steps, edges, assets, and defenses.

3.3 Attack Step Coverage

The simplest structural coverage is the node coverage which is often realized as statement coverage in source code testing [1]. A statement considers to be covered if there is a test case in which the statement is executed. Transferring this concept to attack simulations, a single attack step, $a \in A$, equals a node. Thus, we consider a certain attack step, a, as covered if there is an attack simulation,

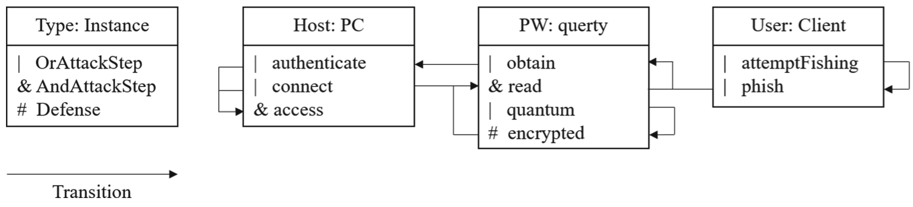


Fig. 1. Graphical representation of the threat model used for the example coverage calculations.

Table 1. Attack step coverage.

	$D_a^{ex} = \{encr.\}$	$D_a^{ex} = \emptyset$ $\bigcup$	
$\lvert A_c \rvert$	3	5	5
$\lvert A \rvert$	8	8	8
Cov.	37.5%	62.5%	62.5%

Table 2. Edge coverage.

	$D_a^{ex} = \{encr.\}$	$D_a^{ex} = \emptyset$ $\bigcup$	
$\lvert E_c \rvert$	3	6	6
$\lvert E \rvert$	9	9	9
Cov.	33.3%	66.7%	66.7%

$\phi(G, A, D)$, in which the attack step becomes compromised: $a \in A_c$. Then, let A_c^u be the united set of all compromised attack steps along all attack simulations.

Consequently, the attack step coverage metric is computed as the ratio between the number of compromised attack steps, $|A_c^u|$, and the total number of attack steps, $|A|$. The attack step coverage percentage is computed as the ratio between them:

$$P_S = \frac{|A_c^u|}{|A|} \tag{2}$$

Table 1 illustrates an exemplary calculation of attack step coverage according to the two attack simulations described in Sect. 3.2. If the encryption is enabled, the attacker is able to additionally compromise *access*, but does not proceed further as *read* is protected. If the encryption is disabled, the attacker is able to compromise all attack steps of $Host$ and two of $Password$ but none of $User$, as there is no edge from $Password$ to $User$ (just vice versa). Accordingly, the attack steps of $User$ are not compromised in any of our attack simulations.

3.4 Edge Coverage

After node coverage, authors usually define edge coverage as the next logical step that is most often implemented as branch coverage [1]. Here, we consider the edges between two attack steps, E, as equivalent to the edges in classical edge coverage. Accordingly, the edge coverage metric is computed as the ratio between all compromised edges $|E_c|$ and the number of edges $|E|$ in the graph:

$$P_E = \frac{|E_c|}{|E|} \tag{3}$$

Table 2 illustrates the calculation of edge coverage with respect to the two attack simulations described in Sect. 3.2.

Table 3. Asset coverage.

	$D_a^{ex} = \{encr.\}$	$D_a^{ex} = \emptyset$	$\bigcup$
$\mathfrak{A}_{partial}$	1	2	2
$\mathfrak{A}_{full}$	1	1	1
$\|A\|$	3	3	3
$P_{partial}$	33%	67%	67%
P_{full}	33%	33%	33%

3.5 Asset Coverage

Next, we propose a metric for asset coverage, which is a metric for computing the percentage of compromised assets during an attack simulation. There exists no equivalently structural test coverage in the domain of software engineering [1]. The construct of class coverage for testing object-oriented software would be equivalent, but such a coverage would not provide any useful information. However, in attack simulations it is of relevance to know if an asset is covered. Unfortunately, there is not one solution to determine when an asset is interpreted as compromised since the requirements may vary depending on the situation. One approach would be to allow the developers to annotate the requirements to each asset. A simpler approach that we follow, is to use two separate definitions when we consider an asset to be compromised:

Full Attack Step Coverage. An asset, $\mathfrak{a} = (A', D', E')$, is considered to be compromised if all associated attack steps, A', have been compromised: $A' \subseteq A_c^u$.

Partial Attack Step Coverage. An asset, $\mathfrak{a} = (A', D', E')$, is considered to be compromised if at least one associated attack step, $a' \in A'$, is compromised: $A' \bigcup A_c^u \neq \emptyset$.

Let $\mathfrak{A}_P$ represent the set of all partially compromised assets, C_A the set of all actually compromised assets and $\mathfrak{A}_F$ the set of fully compromised assets. The percentage of covered assets is computed as the ratio between one of the compromised asset sets and the set of all assets $\mathfrak{A}$:

$$P_X = \frac{|\mathfrak{A}_X|}{|\mathfrak{A}|} \mid X = \{full, partial\} \tag{4}$$

Table 3 illustrates an exemplary calculation of partial and full asset coverage according to the model described in Sect. 3.2. In the first attack simulation, the *Host* gets fully compromised, while the attacker is not able to proceed to *Password*, due to the defense active. In the second attack simulation, succeeds to *Password*, but is not able to compromise it fully, due to *quantum* not reachable.

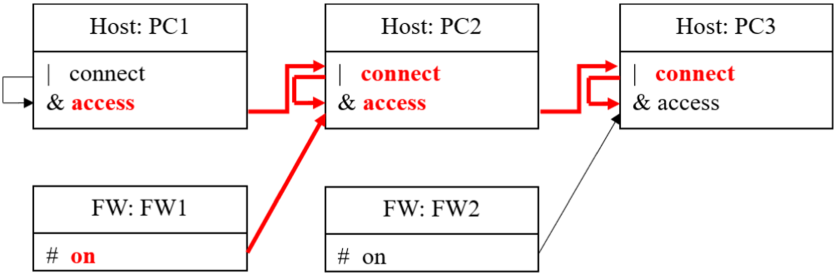


Fig. 2. Example of test covering for defense on FW1 active. Red color rep-resents that the attack step or edge is compromised. (Color figure online)

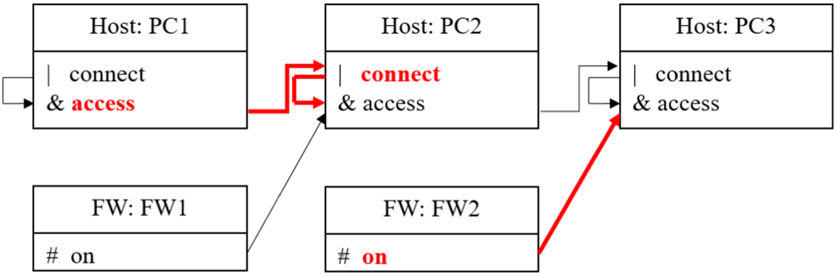


Fig. 3. Example of test covering for defense on FW2 active. Red color rep-resents that the attack step or edge is compromised. (Color figure online)

3.6 Defense Coverage

Defenses are a special because their value is set at the beginning of the simulation. Enabling and disabling defenses may result in different outcomes in multiple executions of an attack simulation where the preconditions of the attacker do not change. Thus, each defense can be understood as an input parameter to the attack simulation and, consequently, interface-based input domain modeling [1] is the equivalent in the software engineering domain. For each input parameter a coverage criterion needs to be defined, usually by applying grouping the different inputs to classes. As the defenses are either in place or not, we consider just these two states. It may of interest for a security researcher to run a simulation with different defenses enabled to investigate how the opportunities of the attacker change according to the presence of certain defenses. Simply running simulations such that each defense has been enabled and disabled in at least one of the simulations may produce misleading results. Figure 2 and 3 presents two simulations such that all values of all defenses in the model have been covered.

From these results it may seem like the attacker is unable to compromise the access step of the rightmost host independent of the defenses state. In this model it is trivial to see that there exists a defense state such that the attacker is able to compromise the rightmost attack step.

Not running separate simulations for each defense state in the threat model under fixed preconditions may produce misleading results. This solution quickly becomes unfeasible since the growth to the defense state space is exponential relative to the number of defenses in the model ($2^{|D|}$). Furthermore, running a simulation for different defense states only produces a coverage metric relevant for the current simulation.

For improved clarity, the notation of a simulation group is introduced which represents a set of simulations under some fixed precondition for different defense states (D_a):

$$SG_{cov} = \frac{|D_a|}{2^{|D|}} = \frac{2^{log_2|D_a|}}{2^{|D|}} = 2^{log_2|D_a|-|D|} \tag{5}$$

For models with multiple simulation groups the defense coverage metric is defined as the ratio between the sum of all simulated defense states in each simulation group, and the total number of defense states. Let $S = \{S_1, \ldots, S_t\}$ represent the set of all simulated simulation groups, where S_k is the set of uniquely simulated defense states in simulation group k. The model defense coverage is described as follows:

$$\frac{1}{t} \cdot \sum_{k=1}^{|S|} \frac{|S_k|}{2^{|D|}} = \frac{1}{2^{|D|} \cdot t} \cdot \sum_{k=1}^{t} |S_k| = \frac{1}{2^{|D|} \cdot t} \cdot 2^{log_2\left(\sum_{k=1}^{t} |S_k|\right)} \tag{6}$$

Then, the ratio of covered defense states for all simulation groups can be computed as follows:

$$Cov_{\mathcal{M}} = \frac{2^{log_2\left(\sum_{k=1}^{t} |S_k|\right)-|D|}}{t} \tag{7}$$

There is one defense in the example threat model presented in Sect. 3.2, resulting in a total of two defense states. Two simulations with the same initially compromised attack steps are executed with different defense configurations forming one simulation group. The resulting simulation group coverage is given by Eq. 5, which results in the following defense simulation group coverage ($|D| = 1, |C| = 2$):

$$SG_{Cov} = 2^{log_2|C|-|D|} = 2^0 = 1 \tag{8}$$

The defense model coverage is given by Eq. 7. Since both simulations in the example above are in the same simulation group, the model defense coverage metric is the same as the defense simulation group metric. The difference between the model and simulation group defense coverage metric is that the model version takes the different simulated groups into consideration when computing the coverage.

4 Test Coverage for MAL-Based Languages

4.1 The Meta Attack Language

Next, we give a short presentation of the MAL. For a detailed overview of the MAL, we refer readers to the original paper [10]. First, a MAL-based DSL contains the main elements that are encountered on the domain under study, those are called `assets` in MAL. The assets contain `attack steps`, which represent the actual attacks/threats that can happen on them.

An attack step can be connected with one or more following attack steps so that an attack path is created. Those attack paths are then used to create attack graphs which are used for the attack simulation. Attack steps can be either of the type OR or of the type AND, respectively indicating that performing any individual parental attack step is required (OR) or all parental attack steps (AND) for the current step to be performed. Additionally, each attack step can be related with specific types of risks. Those risks can be any combination of confidentiality (C), integrity (I), and availability (A) and are specified in brackets after the attack step name. Furthermore, defenses are entities that do not allow connected attack steps to be performed if they have the value TRUE which represents them to be enabled. Finally, probability distributions can be assigned to the attack steps in order to represent the effort needed to complete the related attack step or the probability of the attack step to be possible.

Assets should also have relations between them in order for a model to be constructed, those relations are called `associations` in MAL. Inheritance between `assets` is also possible and each child asset inherits all the attack steps of the parent asset. Additionally, the assets can be organized into categories for purely organization reasons.

In Listing 1, a short example of a MAL-based DSL is presented. In this example, four modeled assets can be seen together with the connections of attack steps from one asset to another. In the `Host` asset, the *connect* attack step is an OR attack step while *access* is an AND attack step. Then, the `->` symbol denotes the connected next attack step. For example, if an attacker performs *phish* on the `User`, it is possible to reach *obtain* on the associated `Password` and as a result finally perform *authenticate* on the associated `Host`. In the last lines of the example the `associations` between the assets are defined.

4.2 Implementation of a Prototype

Hitherto, we have developed four different test coverage metrics for attack simulations and introduced MAL on which we want to measure these coverages. Thus, we gather the coverage data by an extension to JUnit 5 on a system-level for specific threat models, which is available on GitHub[2]. Data gathered by the extension is a measurement of the capabilities of a simulated attacker on a specific model. The extension of JUnit was necessary, since the existing coverage measurements do not map one to on to the proposed measurements for attack graphs. This is caused by the fact that the attack graphs are translated into different classes. Thus, a statement coverage of these classes does not express anything e.g., on the attack step coverage, because attack steps are expressed as values in a list.

Since the extension does not have access to the original MAL language specification there is no metric of how much of the DSL is covered by existing tests.

[2] https://github.com/nicklashersen/malcompiler.

```
category System {
  asset Network {
    | access
      -> hosts.connect
  }

  asset Host {
    | connect
      -> access
    | authenticate
      -> access
    | guessPwd
      -> guessedPwd
    | guessedPwd [Exp(0.02)]
      -> authenticate
    & access {C,I,A}
  }

  asset User {
    | attemptPhishing
      -> phish
    | phish [Exp(0.1)]
      -> passwords.obtain
  }

  asset Password {
    | obtain {C}
      -> host.authenticate
  }
}

associations {
  Network [networks] *
    <-- NetworkAccess -->
    * [hosts] Host
  Host [host] 1
    <-- Credentials -->
    * [passwords] Password
  User [user] 1
    <-- Credentials -->
    * [passwords] Password
}
```

Listing 1: Exemplary MAL Code

However, model specific coverage can be translated to coverage of the DSL, and thus be used to verify the correct operation of specific instances.

By default, JUnit handles extensions on a per-class basis which requires developers to explicitly declare which extensions should be enabled for a specific class. This makes the coverage extensions an opt-in feature where the developers are able to specify which classes that should be tracked by our extension.

The coverage extension supports two modes of operation: global (default) and class-local. Developers can specify class-local exportable coverage targets in order to process certain classes differently or output the results in a different format. Class-local targets must specify a valid exportable target with the `org.junit.jupiter.api.extension.RegisterExtension` annotation declared in the body of affected test class. The global (default) mode of operation handles all classes with a registered coverage extension that are not class local. Developers can specify the default exportable target by changing the `_globalTarget` field in the `CoverageExtension` class.

A prerequisite for generating coverage data is that the extension has access to model and simulation data used during an attack simulation. Since attack simulations essentially are JUnit test instances, this data is generated during the *Test* stage of the JUnit test lifecycle mentioned [6]. At the end of the *Test* stage, all currently registered assets are assumed to be part of the simulated model and are passed to associated exportable target.

The coverage extension registers two test-lifetime cycle callback methods through the JUnit 5 extension API for handling setup and generation of coverage data. Depending on the mode of operation, the extension register itself as a closeable resource with the root- or class-level extension context store. This ensure that the extension does not go out of scope before the coverage data is exported independent of the lifetime of the test-class object instance.

The coverage extension supports custom exportable targets by extending the `ExportableTarget` class. Exportable targets are responsible for computing and formatting coverage data generated by each JUnit test. The `ExportableTarget` class does not provide any functions for computing coverage metrics. As a result, there is no common coverage metric included by default for each deriving class. Methods for accessing attack step and defense objects associated with a specific asset are provided, but the responsibility for deciding what information to include is delegated to the deriving classes.

The `ConsoleTarget` is an extension of the `ExportableTarget` class based on the coverage metrics discussed in Sect. 3. Coverage data is collected after each attack simulation and stored for further processing during the export phase. A `ModelKey` is computed for each model used in the simulations associating test-methods with specific threat-models. Each test-method gets assigned to a simulation group based on the initially compromised attack steps. Since each test method can only be in one defense state, the defense coverage metric is only printed for simulation groups and model coverage metrics.

All coverage data is processed and presented when the export method is called. The coverage metrics for simulation groups and models is represents the compounded result of all child simulations. If an asset, attack step or edge is compromised in at least one child simulation, the object will be considered compromised in the parent simulation group or model metric.

Alternatively, the results can be exported to a JSON file. To ease the understanding of the output, we have developed a dedicated viewer for these JSON files. The viewer is available on Github[3].

5 Related Work

Next, we provide an overview of previously conducted research related to the subject of study. First, we elaborate on model-driven security, which stems from research on extending the UML language for secure systems development. One of the earliest extensions of UML with focus on secure system development was proposed by Jürjens [11]. His proposal focused on extending the semantics of existing UML types for better expressing and integrating security related information such as security requirements, concepts, and threats for components in a UML diagram [11]. Another security focused UML extension was presented by Lodderstedt et al. called secureUML. Rather than providing general threat modeling functionality similar to UMLsec the secureUML extension provides tools for modeling role-based access control and authorization constraints.

Another common threat modeling process is to represent potential exploits and vulnerabilities of a system as an attack graph. The concept of attack graphs is generally considered to be introduced by Schneier in the form of attack-trees [23] and later formalized by Sjouke and Martijn [16]. A limitation with attack trees is that they fail to capture the interaction between attacks and defensive measures

[3] https://github.com/nicklashersen/mal-coverage-viewer.

present in the modeled infrastructure. Kordy et al. expanded the concept of attack trees by introducing defenses (attack-defense trees) to better capture the evolution of security mechanisms and vulnerabilities in a system [14].

Using simulations to provide insight over system operation and for evaluating system requirements is a common practice in many fields, like network security. Cohen [3] suggested a network security model consisting of a network model, a cause-effect model, characteristic function, and pseudo-random number generators for simulating attacks, defenses, and consequences. However due to the simplicity of the cyber-attacks or defense representations based on the cause-effect model, any practical application of the model may prove difficult [2].

Quantifying coverage of potential attacks and threats for a specific model is a difficult task. Domain experts are often required to make a judgment based on prior experience on what type of tests are necessary to provide adequate coverage over relevant attacks and threats for specific systems. Many threat modeling tools do not help with the task of determining when sufficient coverage has been reached, but rather focus on providing estimates of an attacker's capabilities through different metrics. For example, SecureITree [15] models a system through the use of attack-trees and estimates the risk of a specific attack occurring as the product between the incident probability and impact. There are other techniques that work in a similar manner but are based on other metrics, such as the attack and damage cost [22]. These metrics do not provide an estimate on whether sufficiently many attackers have been simulated, but they may provide insight over potential vulnerabilities of the models.

Automated security testing tools tend to provide more easily quantifiable coverage metrics. In a proposal for testing of security policies for firewalls, the security policies were represented as directed acyclic graphs (DAGs). Test cases represented each complete event sequence which could be derived from the DAG [27]. System testing and code-based coverage analysis employ similar testing methodologies. The primary difference is where components used to describe the system (graphs, clauses, etc.) come from [20].

The Meta Attack Language (MAL) is a language for creating domain specific cyber threat modeling systems and attack simulations [10]. While there has been a lot of research on the field of threat modeling, there seems to be little research related to threat modeling and attack simulations in the field of domain specific languages (DSL) [19]. There have been attempts to extend the cause-effect model by employing advanced modeling and simulation concepts for specific domains to support threat modeling and attack simulations [2].

While the MAL is a relatively new language there already exist a number of MAL based DSLs for different domains, like vehicleLang [13], which allows modeling cyber-attacks on modern vehicles. Other languages are coreLang [12], which contains the most common IT entities and attack steps, or powerLang [8] that is designed for modeling organizations in the power domain. Another approach is the automated creation of MAL languages by translating existing concepts from the ArchiMate language to MAL [7].

6 Conclusion

The goal of this work was to develop a method to measure test coverage for attack simulations and to provide a prototype that allows the automated measurement for MAL-based languages. We proposed four coverage metrics based on existing coverage criteria and implemented them. These metrics provide different estimates on how much and what parts of a threat model are covered by a set of tests. This is valuable feedback for the language developers because it supports the decision if further testing is needed.

Since our work is to the best of our knowledge, a new approach to secure the quality of threat models and attack simulations, there are a lot of potential improvements and new features that could be added. Firstly, there exist many different and more advanced coverage metrics like path coverage, modified condition/decision coverage (MC/DC), or active clause coverage (ACC) [1], which could serve as valuable coverage metrics. Secondly, a common metric interface could be designed to separate the coverage metric code from the exportable targets. This could be used to facilitate the reuse of existing coverage metrics during the development of new coverage metrics. Further, we consider defenses either as active or not, while in reality defenses are probably effective. Thus, further research is needed to consider this in our framework, too.

A shortcoming of our work is that the prototype only computes the coverage of assets used in the attack simulations. If an asset defined in the language is never used, the asset will not be included in the coverage computation.

Acknowledgement. This project has received funding from the European Union's H2020 research and innovation programme under the Grant Agreement No. 832907.

References

1. Ammann, P., Offutt, J.: Introduction to Software Testing, 2nd edn. Cambridge University Press, New York (2016)
2. Chi, S.-D., Park, J.S., Jung, K.-C., Lee, J.-S.: Network security modeling and cyber attack simulation methodology. In: Varadharajan, V., Mu, Y. (eds.) ACISP 2001. LNCS, vol. 2119, pp. 320–333. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-47719-5_26
3. Cohen, F.: Simulating cyber attacks, defences and consequences. Comput. Secur. **18**(6), 479–518 (1999)
4. Ekstedt, M., Johnson, P., Lagerström, R., Gorton, D., Nydrén, J., Shahzad, K.: SecuriCAD by Foreseeti: a CAD tool for enterprise cyber security management. In: 19th International EDOC Workshop, pp. 152–155. IEEE (2015)
5. Goodenough, J.B., Gerhart, S.L.: Toward a theory of test data selection. IEEE Trans. Softw. Eng. **2**, 156–173 (1975)
6. Gulati, S., Sharma, R.: JUnit 5 Extension Model, pp. 121–137. Apress, Berkeley (2017)
7. Hacks, S., Hacks, A., Katsikeas, S., Klaer, B., Lagerström, R.: Creating meta attack language instances using ArchiMate: applied to electric power and energy system cases. In: 23rd International EDOC, pp. 88–97 (2019)

8. Hacks, S., Katsikeas, S., Ling, E., Lagerström, R., Ekstedt, M.: powerlang: a probabilistic attack simulation language for the power domain. Energy Inform. **3**(1), 30 (2020). https://doi.org/10.1186/s42162-020-00134-4
9. Holm, H., Shahzad, K., Buschle, M., Ekstedt, M.: P^2CySeMoL: predictive, probabilistic cyber security modeling language. IEEE Trans. Dependable Secure Comput. **12**(6), 626–639 (2015)
10. Johnson, P., Lagerström, R., Ekstedt, M.: A meta language for threat modeling and attack simulations. In: 13th ARES Conference, pp. 1–8 (2018)
11. 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
12. 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
13. Katsikeas, S., Johnson, P., Hacks, S., Lagerström, R.: Probabilistic modeling and simulation of vehicular cyber attacks: an application of the meta attack language. In: 5th ICISSP (2019)
14. Kordy, B., Mauw, S., Radomirović, S., Schweitzer, P.: Foundations of attack–defense trees. In: Degano, P., Etalle, S., Guttman, J. (eds.) FAST 2010. LNCS, vol. 6561, pp. 80–95. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19751-2_6
15. Limited, A.T.: Hostile risk decisions and capabilities-based analysis (2020). https://www.amenaza.com/downloads/docs/Hostile
16. Mauw, S., Oostdijk, M.: Foundations of attack trees. In: Won, D.H., Kim, S. (eds.) ICISC 2005. LNCS, vol. 3935, pp. 186–198. Springer, Heidelberg (2006). https://doi.org/10.1007/11734727_17
17. Miranda, B., Bertolino, A.: Testing relative to usage scope: Revisiting software coverage criteria. ACM Trans. Softw. Eng. Methodol. **29**(3), 1–24 (2020)
18. Myagmar, S., Lee, A.J., Yurcik, W.: Threat modeling as a basis for security requirements. In: SREIS, vol. 2005, pp. 1–8. Citeseer (2005)
19. do Nascimento, L.M., Viana, D.L., Neto, P., Martins, D., Garcia, V.C., Meira, S.: A systematic mapping study on domain-specific languages. In: ICSEA 2012, pp. 179–187 (2012)
20. Offutt, J., Liu, S., Abdurazik, A., Ammann, P.: Generating test data from state-based specifications. Softw. Test. Verif. Reliab. **13**(1), 25–53 (2003)
21. Oladimeji, E.A., Supakkul, S., Chung, L.: Security threat modeling and analysis: a goal-oriented approach. In: SEA 2006, pp. 13–15. Citeseer (2006)
22. Saini, V., Duan, Q., Paruchuri, V.: Threat modeling using attack trees. J. Comput. Sci. Coll. **23**(4), 124–131 (2008)
23. Schneier, B.: Attack trees. Dr. Dobb's J. **24**(12), 21–29 (1999)
24. Sein, M.K., Henfridsson, O., Purao, S., Rossi, M., Lindgren, R.: Action design research. MIS Q. **35**(1), 37–56 (2011)
25. Sheyner, O., Haines, J., Jha, S., Lippmann, R., Wing, J.M.: Automated generation and analysis of attack graphs. In: Proceedings 2002 IEEE Symposium on Security and Privacy, pp. 273–284. IEEE (2002)
26. Shostack, A.: Threat Modeling: Designing for Security. Wiley, Indianapolis (2014)
27. Tuglular, T., Kaya, Ö., Müftüoglu, C.A., Belli, F.: Directed acyclic graph modeling of security policies for firewall testing. In: International Conference on Secure Software Integration and Reliability Improvement, pp. 393–398. IEEE (2009)

Murder, She Modeled: Modeling to Support Crimino-Forensic Processes

Dirk van der Linden[1(✉)], Hava Dayan[2], Anna Zamansky[3], and Irit Hadar[3]

[1] Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, UK
dirk.vanderlinden@northumbria.ac.uk
[2] Department of Criminology, University of Haifa, Haifa, Israel
hdayan@univ.haifa.ac.il
[3] Department of Information Systems, University of Haifa, Haifa, Israel
{annazam,hadari}@is.haifa.ac.il

Abstract. Digital technologies are increasingly used by law enforcement in forensic processes. This opens the door to new opportunities for the use of conceptual modeling techniques to support pro-active collection and modeling of crime scene evidence—especially for the more challenging case of soft evidence. As opposed to digital evidence and other 'hard' evidence (such as finger prints and DNA samples), 'soft' evidence refers to heterogeneous meta-data on the crime scene (such as modus operandi, victim's background, etc.), which challenges the automation and support of both collection and modeling processes for such evidence. Using homicide crime scene staging (HCSS) as a case study, we explore the challenges and opportunities that an approach applying conceptual modeling techniques offers to support crimino-forensic processes in the collection and systematic modeling of soft evidence.

Keywords: Conceptual modeling · User stories · Misuse cases · Crime scene staging · Homicide

1 Introduction

The increasing use of digital technologies by law enforcement provides new opportunities for popularizing the application of systems analysis and modeling techniques to help structure criminological analysis. Such techniques have been recently applied, e.g., to automate and promote pro-activity of collection of digital evidence at crime scenes [20]. Digital evidence, i.e., information stored on digital devices involved in a crime scene, is considered 'hard' evidence, along with e.g., finger prints and DNA samples. Less attention has been paid to the application of conceptual modeling techniques to supporting collection of 'soft' evidence. Soft evidence refers to all types of meta-data on the crime scene (such as modus operandi, victim's background, etc.). It plays a crucial role in the

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 318–331, 2021.
https://doi.org/10.1007/978-3-030-79186-5_21

initial classification of cause of death in potential homicide cases. Classifying a death as non-criminal results in the termination of all forensic processes, and stops mandatory autopsy of the body and forensic inspection of the death scene. Thus mistaken classification at this stage leads to the loss of crucial forensic evidence [3].

One important reason for misclassifications of cause of death is criminal staging of the murder scene by the perpetrator (i.e., murders staged either as a natural death or as a non-natural yet non-criminal death [6]). Usually homicide crime scene staging (HCSS) is an ad hoc or preplanned action aiming at wilfully misdirecting the criminal investigation by manipulating physical or verbal evidence [15], which bears substantial implications for the ability to identify homicide [21]. This phenomenon remains one of the least studied aspects of crime scene analysis [21], and its estimated frequency is up to 8% of all homicide scenes.

To determine whether HCSS has ocurred, a plethora of soft evidence needs to be gathered in order to reconstruct what has happened before, during, and after the homicide. Conceptual modeling techniques may be very valuable here as they allow for a simpler, more structured overview of all the relevant information, conveying relationships and structure between details that would otherwise be lost in masses of textual documents.

Suspected staged crime scenes, where soft evidence has been tampered with, pose a useful case study for the application of conceptual modeling techniques to model how the actions of those involved in the crime scene affect its soft evidence. Applying conceptual modeling techniques to support this is timely because the incidence of such killings are on the rise, and more importantly, because convictions in such cases rely mostly, if not exclusively, on circumstantial soft evidence. Circumstantial convictions pose a major challenge to judicial procedures, as they require ample comprehensive evidence that might be overlooked and missed if law enforcement misinterprets the scene at the time of initial arrival. Without sufficient circumstantial soft evidence, HCSS cases where the killers do not confess to the killing they committed, render themselves practically impossible for criminal conviction—leading to the acquittal of the killer and injustice to the victim's family and society. Circumstantial convictions in cases of HCSS require ample collection of multiple sorts of soft evidence, such as the defendant's behavior before/during/after the death, their behavior at the death scene, nature of prior relations between defendant and victim, the victim's financial beneficiary, lifestyle, social relations, and so on.

Objectives—The aim of this research-in-progress paper is to explore the potential benefits and challenges of applying conceptual modeling techniques for supporting soft evidence collection and analysis from a criminological point of view. In particular, we discuss capturing how staging behavior affects the collection of soft evidence in crime scenes. We focus on a case study of a homicide where the crime scene was staged, and show how capturing behavior of the criminal and law enforcement provides a clearer picture of which actions challenged the collection of soft evidence. In doing so, we show the potential of *patterns of staging behavior* that can be extracted through the use of conceptual modeling techniques, which can be an invaluable resource for the development of evidence-based criminological tools to support evidence collection and classification of death decision making upon initial arrival at a homicide crime scene.

Contribution—We present a preliminary user story and (mis)use case driven analysis of staging patterns from real crime scene data developed iteratively with a criminologist, showing how actions can be mapped to user stories and into balanced (mis)use case diagrams which show criminal and law enforcement activities as adversarial parties, and the benefit this brings from a criminologists' point of view.

In the remainder of this research-in-progress paper, Sect. 2 provides background regarding homicide crime scene staging and the challenges for capturing evidence of staging; Sect. 3 demonstrates how we used conceptual modeling techniques with a criminologist to capture HCSS actions and model how they affect the criminal investigation; and Sect. 4 concludes by discussing the importance of building a support system for criminologists and law enforcement which can use such models to support homicide investigation.

2 Homicide Crime Scene Staging

2.1 Background

Research identifies three main types of homicide crime scene staging (HCSS) as being: (1) primary staging, which is either ad hoc or pre-planned and aims to misdirect the criminal investigation by way of manipulating physical or verbal evidence; (2) secondary staging that manipulates the crime scene devoid of the intention to misdirect the investigation; and (3) tertiary scene alterations aimed at protecting the dignity of the deceased victim [10]. The type of staging that is most difficult to uncover by law enforcement is primary staging, particularly those that are pre-planned. In such cases there are hardly any direct witnesses as killers tend to plan, kill and stage while in isolated scenes, manipulate physical evidence in the scene in order to mislead criminal investigation, and hardly admit or confess to the killing and scene staging they committed.

According to Douglas and Munn [11], staging materializes when someone purposefully alters the crime scene prior to the arrival of the police in order to either redirect the investigation away from the most logical suspect, or to protect the victim or the victim's family. Staging is attained by altering physical evidence at the scene after and at times before the homicide, in order to present a misleading appearance of the crime scene. It imitates a "legitimate death" scenario or a disappearance of the victim [4,21]. Staging is thus a primary means of purposefully obstructing the criminal justice process by concealing the true nature of a crime [5,15].

Analyzing the profiles of both offenders and victims, HCSS researchers have found clear evidence of gender relevance to their identity as victims or perpetrators. First and foremost, a relationship or acquaintance with the victim is associated with homicide staging cases [10]. More to the point, the most common victim-offender relationship involving HCSS is an intimate partner relationship, and most staged homicidal scenes involve the killing of an intimate partner [21]. These HCSS characteristics align with the fact that in most HCSS cases the perpetrator knows the victim, and the perpetrator was the last person to have seen the victim alive. Added to this typicality are other gendered HCSS features: stagers are more often men rather than women [15,16], and women comprise the majority of victims of crimes where HCSS took place [21].

Current literature is divided as to the prevalence of HCSS, with some estimating it as high as approximately 8% of all homicidal crime scenes [15,16,22]. Most often these crime scenes are indictable at first glance as the murder scene stager has altered or eliminated all direct evidence. Since staging is a primary means of purposefully obstructing the criminal justice process by concealing the true nature of the crime [15], the criminological nature of such crimes poses a major challenge to the police, investigators, prosecution and judicial process. Undetected at all, or detected at a later stage of police investigation, often leads to detrimental total loss of crucial and indispensable forensic evidence needed to secure criminal procedure and conviction [3,6,7].

As the case study in Sect. 3.1 will demonstrate, 'gut' feelings at the death scene, based on soft evidence, are crucial. Indeed, homocide detectives often use 'gut' feeling to assess whether a crime scene may be staged. Geberth, a former commander in the NYPD's Bronx Homocide Division noted:

> "In my experience investigating suspicious deaths I have often times had a 'gut' feeling that something was amiss. Actually, that 'gut' feeling is your subconscious reaction to the presentation, which should alert you to the possibility that, things are not always what they appear to be, consistent with equivocal death investigations" [14].

However, these 'gut' feelings, or intuitions, are highly dependent on vast experience of the investigator. A more systematic, rationale-based means for supporting crime scene investigations based on known staging behavior could assist inexperience and veteran investigators alike to ensure that important evidence is not overlooked.

Several attempts toward this direction can be found in the literature. For example, Omeleze and Venter [18] proposed a "digital forensic application requirements specification (DFARS) process" as a first step toward developing an effective application of this sort. Pasquale et al. [20] worked on *digital* forensics, emphasizing its difference from 'offline' forensics, and proposing an adaptive security approach, whose adaptation depends on suspicious events and investigation findings. Some effort has also be invested in building supporting databases. Schlesinger et al. [22] studied a large sample (N = 946) of homocide crime scenes within the United States, finding that 8.4% of the cases were staged, identifying a number of different types of staging and their characteristics. Focusing specifically on staging, Hazelwood and Napier [15] performed a survey with 20 law enforcement officers, through which they identified motives for staging, and the acts that occur before it is performed. More recently, Ferguson and Wayne [13] examined common features in homicide crime scene staging, finding that simulation of self-injury was most frequent. Ferguson and McKinley [12] further explored staging, arguing that many homicides likely remain misclassified.

2.2 The Challenge of Applying Conceptual Modeling Techniques in HCSS

The major challenge in applying conceptual modeling techniques for collecting soft evidence in staged homicide crime scenes is that extant approaches to support evidence collection processes for e.g., digital evidence (cf. [20]) presuppose the existence of a domain model of a crime scene. This is a significant challenge for crimes in the physical world, as modeling real crime scenes (as opposed to e.g., the electronic devices related to a crime scene) pose several knowledge representation problems. Similar to the frame problem well known in the philosophy of Artificial Intelligence [8], it is impossible to make decisions on the basis of only what is relevant to an ongoing situation, without first having explicitly considered many more factors that are not relevant. In other words: when law enforcement arrives at a crime scene it is impossible to only look at what is (ultimately) relevant to the crime at hand. Moreover, many current approaches proposed in literature to support crime scene analysis through reasoning support focus either on cyber crime [2,9,19], or, if looking at 'real-world' crime, provide initial frameworks focused on hard evidence [1,17]. Some research, though, has noted that "more attention should be paid to human and social factors" [24].

The challenge of identifying what soft evidence is relevant is all the more prominent in HCSS, as the physical evidence in the scenes are purposefully altered by the criminal in order to mislead investigators as to what has actually taken place. Moreover, in HCSS cases, the soft evidence that can be collected and interpreted is heterogeneous and multi-faceted. For example, soft evidence of various and differing realms might be relevant, such as the prior medical condition of the deceased, their social networks, any monetary or financial agreements they may have had, the behavior of the criminal and their victim prior to the murder, the behavior of the criminal at the death scene, the characteristics of the death itself, and so on.

Finally, HCSS cases are in an ever-changing state. If we consider the homicidal crime scene as the final state of a 'system', with our goal being understanding the behavior of the system's 'users' (e.g., criminals and law enforcement)–we need to consider the system's transition through different states. Thus, when applying conceptual modeling techniques in order to gain insights about crime scene and the behavior of its relevant 'users', we need to account for its various transition states, such as actions taken:

1. before the homicidal crime is committed;
2. when the homicidal crime is being committed;
3. when the staging of the scene takes place;
4. when the police officers arrive to investigate the crime scene at hand; and
5. when the criminal returns to the crime scene for further staging.

3 Conceptual Modeling for Staging Detection

This section presents a case study of homicide crime scene staging by analyzing staging actions taken in an actual homicide case, and modeling them with conceptual modeling techniques. The case study draws from an ongoing project tasked with developing a decision-support system for law enforcement officers arriving at a suspected homicide crime scene, using a knowledge base of previous cases of homicide staging to support evidence collection in situ.

3.1 Case Study: TPH 3002/00 the People v. Siboni

In this criminal case, a circumstantial murder conviction was ruled, even though the murderer never admitted to their killing, made significant effort to conceal the true criminal nature of the death scene, and hard evidence in the form of forensic tests was immaterial as tests rendered inconclusive.

The criminal, Siboni, was married to the victim, his wife Mendy, for fifteen years. One day, in September 1999, upon deciding it was time for him to move out to live with his younger lover, he shot his wife in their bedroom. Mendy was 'found' dead by Siboni after being shot in her head once. Upon 'finding' his wife, Siboni called the emergency number, asking them to instruct him as to how to revive his wife. Upon the emergency services arrival at the couple's house, the rescue team was let in by a neighbor. Siboni was still in the bedroom, soaking his wife's head bleeding into a towel. A physician at the rescue team confirmed Mendy's death. Police investigators at the murder scene documented Siboni's claim that his wife shot herself by accident while checking her gun. His version for the events was that he took out the gun he legally owned from the bathroom, left it on their bed, and went downstairs. When he got back to their bedroom, he found Mendy holding the gun towards her eyes, stating that she was looking for 'dirt' in the barrel. Realizing how dangerous this was, he reached out for the gun, trying to get it out of her hand, when it discharged, and hit her in her forehead.

Siboni made sure to take their young children to kindergarten before Mendy's murder. Additionally, he notified the maid that she was not needed for her cleaning chores that day, since Mendy, as he described it, was not feeling well. In his 'efforts' to revive Mendy after the shooting, Siboni was able to move the gun away from her and explain his fingerprints on it as part of his attempt to prevent her from pulling the trigger. He was therefore the only person with her and the last to have seen her alive. Upon the arrival of the rescue teams he had told them the emergency services as well as the neighbors gathered in the house that Mendy shot herself. At the time, he also described their relationship to the police as ideal, despite being on the verge of divorce as the couple made a divorce settlement he hid from the police, and despite having had a lover for over a year. Moreover, later it was revealed that Siboni was supposed to move out from their house that day but had no alternative accommodation arrangements yet.

Siboni's actions led the police to be skeptical of his version of the way Mendy's death unfolded. Yet, he was only arrested on suspicion of her murder a year and a half later, when the collection of soft circumstantial evidence seemed to amount into substantial incriminating evidence required for the handling of the case by criminal prosecution. Particularly incriminating soft circumstantial evidence were wiretapped and recorded phone conversations with his lover a year and a half after Mendy's death, having had an intense sexual night for which the lover praised Siboni's sexual performance, while incidentally referring to Mendy's death.

Law enforcement's skepticism, and Siboni's eventual murder conviction, was based on circumstantial soft evidence surrounding his behavioral patterns prior, during and after his wife's death. Within these soft circumstantial evidence were for example suspicious behavior such as his discarding of the divorce agreement written and signed by both, hardly arriving to console Mendy's family at her burial and commemoration gatherings, seeming to Mendy's family not impacted and even emotionally joyful, or at least indifferent to Mendy's death. Moreover, he refused to grant the police permission to send Mendy's body for an autopsy, and was eventually found as being on the verge of a divorce and having no alternative place to live on the day Mendy 'accidentally' shot herself. Moreover, he had an extra-marital affair for which he was willing to leave Mendy which he initially concealed from police investigators, had financial hardships in light of the upcoming divorce and partition of joint assets, and additional soft evidence on Mendy's skilful knowledge of gun handling and shooting,

Siboni's evolving and ever changing account of the events that led to Mendy's death, the improbability of an alternative explanation of Mendy having shot herself, and the gathered soft evidence tarnished Siboni's credibility in the eyes of the court. This collection of soft circumstantial evidence was of utmost importance in Siboni's murder conviction, not only because the scene was tampered and staged, but more so because all forensic tests which law enforcement conducted (for example, to assess the probability of an accidental shooting in the angle described on Siboni, as well as shooting residues on Siboni and Mendy's palms) rendered inconclusive.

3.2 User Stories in Homicide Staging

Homicide crime scene staging, in the context of systems analysis, can be seen as user behavior, where criminals engage in systematic activity (staging) with a clear goal (to avoid detection by other stakeholders, i.e., detectives). For law enforcement to detect this kind of behavior, we need to systematically capture the patterns of behavior, meaning what *actions* a criminal took, and what *goal* they hoped to achieve in doing so. To do this in a systematic, yet intuitive way, we capture these behaviors as user stories to explicitly structure how actions taken by criminals and law enforcement threaten and mitigate the uncovering of crime scene staging.

Below in Box 1 we extract criminal user stories from Sect. 3.1's case study, writing the actions and intended goals taken by the criminal in *People v. Siboni*. These should all be read as originating from the role of the *criminal* (e.g., as a typical user story, being prefaced with 'As [a criminal],). Moreover, we divide the actions into the three chronological transition states of a crime scene as discussed in Sect. 2.2. We additionally highlight in red actions that the criminal *failed* to take, which importantly led to law enforcement's suspicion, continued investigation, and eventual criminal indictment.

Box 1. Criminal User Stories for *People v. Siboni*

(1) (Before the) Homicidal crime is being committed:

- I want to [murder my wife], so that [I can move in with my lover]
- I want to [be alone], so that [there are no witnesses to my murder].
  - I want to [take my kids to kindergarten], so that [there are no potential witnesses to the murder]
  - I want to [give my maid the day off], so that [there are no potential witnesses to the murder]

(2) Staging of the scene takes place:

- I want to [call authorities after murder and ask for instructions how to revive my victim], so that [I avoid initial suspicion]
- I want to [attempt to 'revive' my victim], so that [I can explain my otherwise incriminating fingerprints on the murder weapon]

(3) When police officers arrive to investigate the crime scene at hand:

- I want to [tell the rescue team and other people prsent that my victim shot herself accidentally], so that [there is a non-criminal explanation of the death]
- I want to [conceal suspicious facts], so that [I avoid initial suspicion]
  - I want to [conceal the fact of my relationship troubles to the police], so that [I avoid initial suspicion]
  - I want to [conceal the fact of having had a lover for over a year], so that [I avoid initial suspicion]
  - I want to [conceal the fact I was meant to move out that day with no alternative accommodation], so that [I avoid initial suspicion]
- I want to [object to an autopsy], so that [hard evidence cannot be collected]
- I want to [avoid suspicion by my environment], so that [I get away with my crime] **(FAILED)**
  - I want to [console my victim's family], so that [I avoid suspicion] **(FAILED)**
  - I want to [be seen as mourning my victim], so that [I avoid suspicion] **(FAILED)**
  - I want to [keep my account of the events consistent], so that [I avoid suspicion] **(FAILED)**

Next, in Box 2 we capture user stories from the role of *law enforcement*, necessarily only from the third chronological state of when police officers arrived to investigate the (potential) crime scene at hand. As Box 2 shows, while the criminal's actions mitigated law enforcement from achieving several key goals in gathering evidence, their failure to avoid suspicion by their environment led to law enforcement's skepticism and further collection of soft evidence. Here, we highlight in red which actions taken by law enforcement were failed, or were (at least temporarily) mitigated by the criminal's actions.

Box 2. Law Enforcement User Stories for *People v. Siboni*

(3) When police officers arrive to investigate the crime scene at hand:
- I want to [interview potential witnesses], so that [I can determine whether the cause of death is of criminal nature] **(MITIGATED)**
- I want to [investigate the potential murder weapon], so that [I can determine whether the cause of death is of criminal nature] **(MITIGATED)**
- I want to [investigate whether the victim's injury makes sense with the accident scenario], so that [I can determine whether the cause of death is of criminal nature]
  - I want to [conduct forensic tests on the shooting angle], so that [I can determine whether there was staging] **(FAILED)**
  - I want to [conduct forensic tests on the shooting residues], so that [I can determine whether there was staging] **(FAILED)**
- I want to [investigate the relation between the victim and the suspect], so that [I can determine whether the suspect had motive]
  - I want to [interview the suspect about their relationship with the victim], so that [I can determine whether the suspect had motive] **(MITIGATED)**
  - I want to [interview the suspect's environment about their relationship with the victim], so that [I can determine whether the suspect had a motive]
- I want to [wiretap the suspect's phone], so that [I can collect incriminating soft evidence]

These user stories only capture a small part of the wealth of context and behavioral data relevant to crime scenes. However, they help in more clearly showing (eventual patterns of) criminal actions that mitigate law enforcement actions—in this case mitigating critical law enforcement resources like witnesses, analysis of the murder weapon, and initial interview of the suspect. However, the criminal failed in taking actions to avoid arousing suspicion by their environment, which allowed law enforcement to build a case base and (on a longer time-frame) collect enough evidence to secure a prosecution.

3.3 (Mis)Use Cases in Homicide Staging

The key value of systematically extracting and explaining the actions taken by criminals and law enforcement in these cases is not what actions they took and why, but how criminal actions *mitigated* law enforcement actions. In other words: how criminal misuse cases threaten law enforcement use cases. To exemplify this, we have modeled the above case as a Use Case Diagram shown in Fig. 1, with criminal actions as misuse cases, and law enforcement actions as use cases.

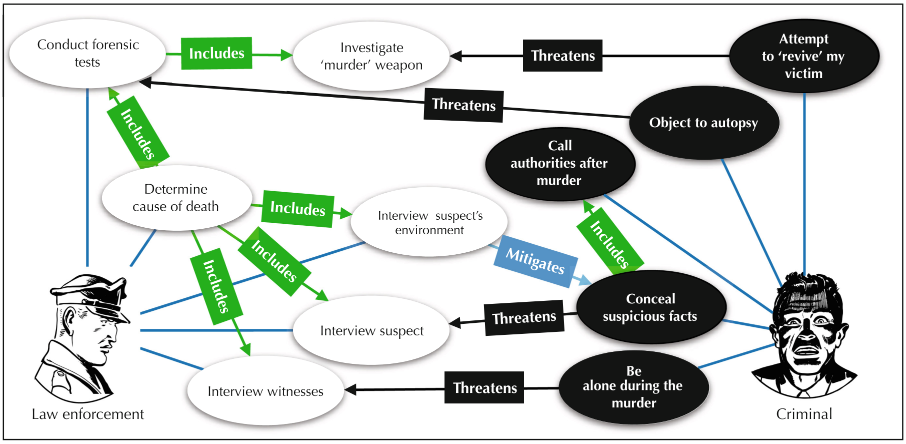


Fig. 1. Use/misuse case diagram of the interplay between criminal and law enforcement actions in the modeled People v. Siboni case.

Figure 1 visually shows that the key action by law enforcement here was the interviewing of the suspect's environment, as it led to a breakdown in their ability to conceal suspicious facts. Note that there is no use case for 'avoiding suspicion by my environment', as this was a critical set of actions *not* taken by the criminal, which may have allowed them to threaten law enforcement's 'interview suspect's environment'. Abstracting this to model a more general model, we see in Fig. 2 a general 'balance' between the criminal attempting to stage a crime scene, and law enforcement's uncovering what truthfully happened.

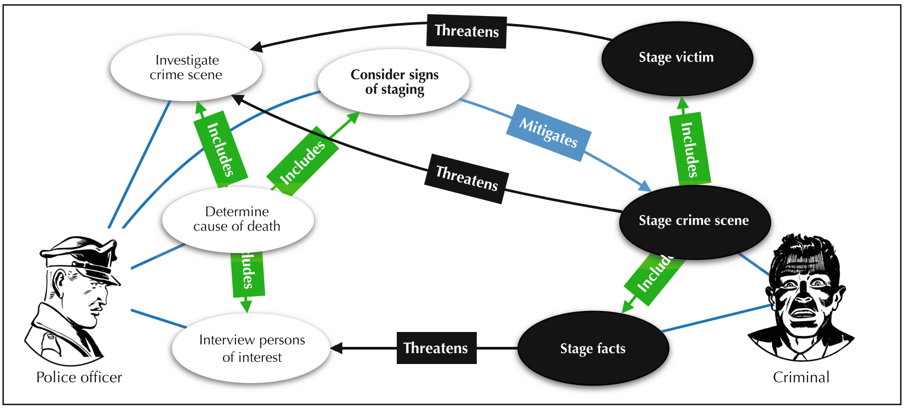


Fig. 2. Abstract use/misuse case diagram of actions in crime scene staging.

What is *critical* here for law enforcement, is the use case of 'consider signs of staging'–to continuously and systematically assess signs of stating in order to mitigate crime scene staging actions taken by criminals. This requires further work in extracting patterns of staging (and which law enforcement actions they mitigated or frustrated), and building support systems for law enforcement while investigating crime scenes.

4 Discussion and Future Work

In this research-in-progress paper we highlight crimino-forensic processes related to soft evidence collection as an exciting new opportunity for interdisciplinary collaboration between information systems and criminology researchers. The need for conceptual modeling techniques arises due to the nature of information that needs to be collected at a crime scene. This information is heterogeneous, unstructured and sometimes extremely ethically and legally sensitive. This leads to the idea explored here of thinking of a crime scene as a complex system with different stakeholders that use and manipulate it. Based on a real HCSS case from our ongoing project, we have shown here how structure can be introduced into existing data on HCSS by using conceptual modeling techniques like user stories and (mis)use cases models.

Such UML models with an accessibly friendly secondary notation for non-technical notation [23] are a good first step in visualizing and thinking about the way in which actions taken by suspects and law enforcement threaten and mitigate each other. They visually bring together these actions and relations in a single diagram, which aids in the reconstruction of what happened before, during, and after the crime, moreso than considering only textual descriptions of soft evidence. We opted for this approach, using these straightforward and accessible requirements engineering notations to fit the needs of domain experts–criminologists who want ways to help structure and reason over tons of soft evidence, while having little time or appetite to become professional modelers themselves. Thus, while arguably the presented reasoning could be captured with formal reasoning notations, or process modeling notations to capture their full procedural scope, this would do little to satisfy the direct domain expert needs: straightforward and usable notations that help in structuring and presenting knowledge.

This paper has so far presented only a single exemplary case study modeled collaboratively with a criminologist, with a limited number of actions, while a use case like 'conceal suspicious facts' will have many different concrete behaviors associated with it. For example, in *People v. Cooper*, the murderer attempted to conceal the true nature of his murder by constructing a possible motive for suicide by spreading lies that his wife was depressed for months prior to the murder, and subsequently staged the crime scene to look like a suicide by littering it with sleeping pills. A far larger number of behavioral patterns thus likely exist, which should be modeled in collaboration with criminologists in order to build a database capable of supporting law enforcement across different countries and cultures.

As reasoned above, given that the domain experts have little appetite to become professional conceptual modelers themselves, and there is no scope to 'model for the sake of modeling', where information systems research can thus further support criminologists working on HCSS is not only aiding in structurally capturing and documenting patterns relevant to HCSS, but to be able to move such patterns into an evolving and constantly updated knowledge base, based on which evidence-based decision support can be facilitated, aiding law enforcement in detecting similar patterns, as visualized in Fig. 3. Such expert systems could hide entirely the formal underlying mechanisms and knowledge representation, instead simply providing intuitive means of entering and representing soft evidence.

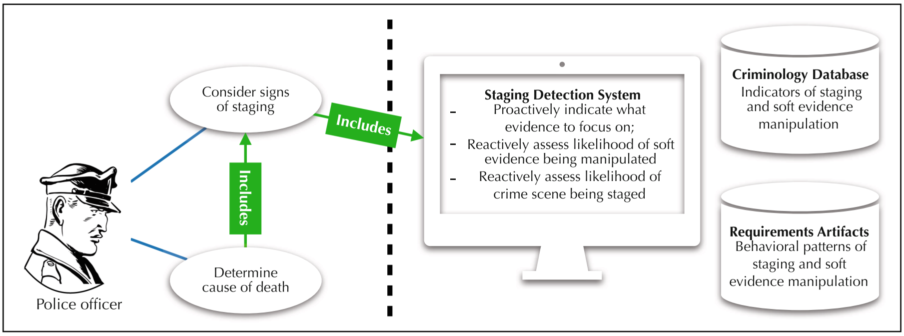


Fig. 3. Components of a system that could support law enforcement in detecting crime scene staging.

To support the development of such databases and support systems, a significant combined research effort from information systems engineers and criminologists is required, including:

- Modeling of past cases to capture detailed behavioral patterns that show how soft evidence was manipulated and with what goal;
- Modeling how behavioral patterns of criminals relate to that of law enforcement, in particular how they threaten and mitigate each other;
- Linking these behaviors in a wider context of criminology data to ground them in e.g., specific criminological or socio-cultural contexts;
- Building decision-support systems incorporating the modeled data to both pro-actively point towards potentially suspicious behavioral patterns for a given context (e.g., in cases of femicide combinations such as the victim's partner reporting a suicide while the victim's family saw no indicators to support this), and to provide general indications of what 'not to miss' during initial crime scene investigation.

5 Conclusion

This research-in-progress paper presented just the tip of the iceberg in exploring how conceptual modeling techniques can be helpful for modeling and structuring various aspects of crimino-forensic processes. It is our hope that this paper will serve as a discussion starter on the topic of modeling for criminology and get more researchers involved in our ongoing efforts with criminologists to document and model behavioral patterns relevant to HCSS. One such discussion we foresee is to what extent these research efforts could be misused by criminals to aid them in staging crime scenes more effectively, complicating law enforcement's analysis efforts even further. As a consequence, building a data-driven system to support law enforcement should incorporate critical discussions and reflections on the cyber security of such systems, prompting yet more research opportunities for (security) requirements engineers.

Acknowledgements. This research was partially supported by the Israel Science Foundation (grant N. 550/19).

References

1. Adderley, R., Townsley, M., Bond, J.: Use of data mining techniques to model crime scene investigator performance. In: Ellis, R., Allen, T. (eds.) SGAI 2006, pp. 19–32. Springer, London (2006). https://doi.org/10.1007/978-1-84628-666-7_2
2. Baryamureeba, V., Tushabe, F.: The enhanced digital investigation process model. In: Proceedings of the Fourth Digital Forensic Research Workshop, pp. 1–9 (2004)
3. Bitton, Y., Dayan, H.: 'The perfect murder': an exploratory study of staged murder scenes and concealed femicide. Br. J. Criminol. **59**(5), 1054–1075 (2019)
4. Chisum, W.J., Turvey, B.E.: Crime Reconstruction. Academic Press, Cambridge (2011)
5. Cobin, S.B.: Staged hunting mishaps and criminal homicide. Center for Homicide Research (2009)
6. Dayan, H.: Homicidal staged crime scenes: Israel as a case study (forthcoming)
7. Dayan, H.: Socio-cultural aspects of homicidal crime staged scenes: Implications for detection (forthcoming)
8. Dennett, D.C.: Cognitive wheels: the frame problem of AI. Lang. Thought **3**, 217 (2005)
9. Donalds, C., Osei-Bryson, K.M.: Toward a cybercrime classification ontology: a knowledge-based approach. Comput. Hum. Behav. **92**, 403–418 (2019)
10. Douglas, J., Douglas, L.: The detection of staging, undoing and personation at the crime scene. In: Crime Classification Manual, pp. 31–44 (2006)
11. Douglas, J.E., Munn, C.: The detection of staging and personation at the crime scene. In: Crime Classification Manual, pp. 249–258 (1992)
12. Ferguson, C., McKinley, A.: Detection avoidance and mis/unclassified, unsolved homicides in Australia. J. Crim. Psychol. (2019)
13. Ferguson, C., Petherick, W.: Getting away with murder: an examination of detected homicides staged as suicides. Homicide Stud. **20**(1), 3–24 (2016)
14. Geberth, V.J.: The staged crime scene. Law Order **44**(2), 89–91 (1996)

15. Hazelwood, R.R., Napier, M.R.: Crime scene staging and its detection. Int. J. Offender Therapy Comp. Criminol. **48**(6), 744–759 (2004)
16. Keppel, R.D., Weis, J.G.: The rarity of "unusal" dispositions of victim bodies: staging and posing. J. Forensic Sci. **49**(6), JFS2004033-5 (2004)
17. Ling, C.P., Noor, N.M.M., Mohd, F., et al.: Knowledge representation model for crime analysis. Procedia Comput. Sci. **116**, 484–491 (2017)
18. Omeleze, S., Venter, H.S.: Digital forensic application requirements specification process. Aust. J. Forensic Sci. **51**(4), 371–394 (2019)
19. Park, H., Cho, S.H., Kwon, H.-C.: Cyber forensics ontology for cyber criminal investigation. In: Sorell, M. (ed.) e-Forensics 2009. LNICST, vol. 8, pp. 160–165. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02312-5_18
20. Pasquale, L., Yu, Y., Salehie, M., Cavallaro, L., Tun, T.T., Nuseibeh, B.: Requirements-driven adaptive digital forensics. In: 2013 21st IEEE International Requirements Engineering Conference (RE), pp. 340–341. IEEE (2013)
21. Pettler, L.G.: Crime Scene Staging Dynamics in Homicide Cases. CRC Press, Boca Raton (2015)
22. Schlesinger, L.B., Gardenier, A., Jarvis, J., Sheehan-Cook, J.: Crime scene staging in homicide. J. Police Crim. Psychol. **29**(1), 44–51 (2014)
23. Schrepfer, M., Wolf, J., Mendling, J., Reijers, H.A.: The impact of secondary notation on process model understanding. In: Persson, A., Stirna, J. (eds.) PoEM 2009. LNBIP, vol. 39, pp. 161–175. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-05352-8_13
24. Wang, S., Wang, X., Ye, P., Yuan, Y., Liu, S., Wang, F.Y.: Parallel crime scene analysis based on ACP approach. IEEE Trans. Comput. Soc. Syst. **5**(1), 244–255 (2018)

Model-Driven Engineering and Applications (EMMSAD 2021)

Model-Driven Engineering: A State of Affairs and Research Agenda

Charlotte Verbruggen(✉) and Monique Snoeck

Research Center for Information Systems Engineering, KU Leuven, Naamsestraat 69, 3000 Leuven, Belgium
{charlotte.verbruggen,monique.snoeck}@kuleuven.be

Abstract. The Model-Driven Architecture has been launched in 2001 by the OMG. Since then, model-driven engineering has been embraced by the research community but less than hoped for by practitioners. To ensure the relevance of a research agenda, we need a good understanding of practitioners' problems, in particular with modelling. We therefore performed a literature review on the state of practice in the use of modelling languages for software engineering in the last 5 years according to Kitchenham's guidelines. This paper serves as orientation within the research field and as a basis for further research. It contributes to literature by focusing on papers discussing practical use of modelling languages and the benefits and problems perceived by practitioners. The main finding presented in this paper is that while practitioners experience benefits of modelling for analysis and design, requirements engineering, quality management, implementation and deployment, they still struggle with external tool integration/model transformation and export, cognitive fit, visual expressiveness, high effort required in acquiring skills, automated analysis and high effort required in using tools. Other findings are that modelling is mostly used for documentation and requirements elicitation, the most used modelling language is UML.

Keywords: Model-driven engineering · Modelling in practice · Systematic literature review · UML · BPMN

1 Introduction

The Model-Driven Architecture has been launched in 2001 by the OMG. In Model-Driven Engineering, software systems are modelled platform-independent [1]. Platform dependent models can then be obtained through transformations. Since then, modelling has been embraced by the research community but less than hoped for by practitioners. In [2], the authors discuss the need to 'pull' from the needs of practitioners instead of 'pushing' solutions from research to the industry. This is an important motivation for researchers to conduct surveys on practitioners in order to identify research paths that will solve problems the industry faces. Several surveys of practitioners' problems have been performed in the past and they all report different results. The differences may (for example) be due to the papers focusing on different aspects, having different

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 335–349, 2021.
https://doi.org/10.1007/978-3-030-79186-5_22

demographics, having been performed at a different time or because of analyzing the results through a different lens. For example, [3] is a survey that discusses the evolution of modelling in software design over a decade, whereas [4] focusses on the use of UML in open source projects. These differences will be discussed throughout this literature review. While this conveniently allows a researcher to cherry-pick a survey that best demonstrates a specific gap, the main drawback is that knowledge about the state of practice remains scattered. In order to address practitioners' problems in an adequate way, a good understanding of their problems is needed, and in order to set priorities, it is interesting to know what problems surface consistently across surveys.

The goal of this research is performing a meta-review in order to obtain a general overview of the needs of practitioners. Therefore, the research questions addressed will be of a general nature in order to capture all relevant information from the last five years. The timespan is set to five years since the industry is rapidly expanding and therefore the use of modelling languages can be drastically different now compared to five or more years ago.

In the next section, we first discuss the related research. Then, the methodology will be discussed. In Sect. 4, the results of the literature review are reported. In Sect. 5, the findings from the results are discussed.

2 Related Research

In order to find existing meta-reviews related to modelling in practice, we executed a general query on Web of Science and Scopus. The query returned 15 papers closely related to this topic. From these 15 papers, two papers have the same focus as our meta-review and are discussed below. The other 13 papers have a too narrow focus e.g.: aspects of a language or a specific domain such as cyber-physical systems, or focus on academic research instead of practitioners.

In [5], the authors performed a literature review investigating "the mismatch between the research field of modelling *language quality* evaluation and the actual MDE practice in industry". They identified seven challenges that the industry faced and that were not addressed by the research. While language quality and the identified challenges are pertinent, we aim for a broader scope.

In [6], the authors performed a systematic literature review on the "applications of ontologies in requirements engineering" selecting papers from 2007 to 2013, thus addressing a different time-frame than we aim for. Some of their conclusions are that OWL was used by the most studies as an ontology-related language to support requirements engineering and that the three main benefits of ontologies in the RE process are that they "reduce ambiguity, inconsistency or/and incompleteness", they "aid requirements management" and they improve "domain knowledge representation for guiding requirements elicitation".

3 Methodology

This paper follows the guidelines for a systematic literature review by Kitchenham [7]. First, the research questions are defined. Then, we discuss the search strategy for finding relevant papers. In Sects. 3.3 and 3.4 we explain the inclusion and exclusion criteria and the selection procedure respectively. In the final section, the quality criteria are explained and discussed.

3.1 Defining Research Questions

The first step in a systematic literature review is to identify the research questions. Architecting systems requires dealing with a variety of concerns (also named viewpoints) and many tasks related to phases of a development lifecycle and/or goals of the requirements engineering process. This raises the following question:

RQ1 - For which aspects (type of activities) of software development do practitioners use modelling languages?

Two often named modelling languages are UML and BPMN. As a general-purpose language, UML can be used for a wide range of goals. The use of BPMN is constrained to the domain of Business Process Management. As the popularity of these languages is growing, an interesting question is to what extent these two languages are dominating the field. This leads to the second research question:

RQ2 - Which other modelling languages are in use next to BPMN and UML and how often are they used?

The first two research questions discuss how modelling languages are currently used. The final two research questions identify the negative and positive experiences that practitioners have of modelling languages.

RQ 3 - Which problems/difficulties/requirements do practitioners experience with model-driven engineering?

RQ 4 - What are the benefits of model-driven engineering according to practitioners?

3.2 Search Strategy for Finding Relevant Papers

Prior to identifying the search query, we selected a set of 5 papers as a golden standard. These are surveys that would ideally be included in the literature review. The goal of this golden standard is to check whether the query finds all these surveys: [2, 3, 8–10]. The query was executed on Web of Science & Scopus. We choose these databases because they cover a wide selection of publishers including Springer, Elsevier, IEEE and ACM. The initial query executed on both databases was:

topic = [(conceptual modelling OR UML OR BPMN) AND (practice* OR use OR practitioner* OR professional*) AND (review OR survey OR summary OR summarize*)] AND LANGUAGE = English.

This resulted in 766 papers on Web of Science and 320 papers on Scopus. In order to filter more thoroughly, we restricted the document type to “Review”. This resulted in 66 papers on Web of Science and 92 papers on Scopus. There were still many results that were related to different fields. Therefore, the results were additionally filtered on the categories that seemed irrelevant. Checking the filtering confirmed that only irrelevant papers were left out through this additional filtering.

The initial query was executed again with exclusion of papers with publication date before 2015 or in the irrelevant categories. This resulted in 221 papers on Web of Science and 102 papers on Scopus. This selection included all 4 of the 5 golden standard papers published after 2014. Removing the duplicate papers that appear in both databases led to a final query result of 277 papers.

3.3 Inclusion and Exclusion Criteria

The inclusion and exclusion criteria determine which papers will be discussed in the review. A paper is included if (1) it is an empirical study with practitioners or a literature review of empirical studies with practitioners, (2) it addresses at least one of the research questions and (3) its topic is information systems modelling, not statistical or mathematical modelling or simulation. A paper is excluded if (1) it is an empirical study with only students or academics as participants or (2) it is a theoretical paper.

3.4 Study Selection Procedure

The study selection procedure consists of three iterations. In the first iteration, two researchers assessed each paper against the inclusion and exclusion criteria using the title and abstract. When there was doubt or disagreement, the papers were included in the next iteration. 42 papers were selected in this iteration. In the second iteration, one researcher read the full papers and reassessed the inclusion and exclusion criteria. Final decision was taken jointly by the two researchers. This second iteration resulted in a set of 20 papers. In the final iteration, we analyzed the authors of the papers and the data used in the papers, and discarded two more papers to eliminate duplicate reporting on the same data.

The final selection of 18 papers is [2–4, 10–24]. Since the focus of this literature review is the experience of practitioners, the majority of the selected papers are surveys. Table 1 provides an overview of the demographic information of all the surveys included in this literature review. Papers [11–13] and [18] are not surveys but they are included because they focus on the perspective of practitioners. Paper [11] is an analysis of modelling tools and how they comply with a set of requirements. Paper [12] is a literature review that focusses on primary studies. Paper [13] is an analysis of architectural languages and how they comply with a set of requirements and paper [18] is a literature review of empirical studies on BPMN.

3.5 Quality Criteria

The quality criteria were not used for the selection of the papers, but are useful to indicate the overall quality of the papers. For each type of paper, we defined a set of criteria. For systematic literature reviews, the quality criteria are (1) the paper has a clear search strategy based on the guidelines by Kitchenham [7] and (2) the paper has a clear selection strategy based on the guidelines by Kitchenham [7]. For systematic mapping studies, the quality criterion is (3) the paper reports the protocol used. Furthermore, surveys should comply with (4) the survey design is reported in the paper, (5) detailed survey results are included in the paper and (6) the demographics of the survey participants are discussed. Finally, all types of papers should comply with (7) a discussion of the threats to validity is included. Most papers fulfilled the quality criteria. One survey [16] did not fulfill criterion 6. The two literature reviews [12] and [18] and one survey [21] did not fulfill criteria 7.

Table 1. Summary of demographic information of the surveys.

Paper	Year of survey	Nr of re-spondents	Europe	Americas	Asia/ Pacific	Middle East	Africa	Largest group re-ported
[10]	March '18-June '18	109	34 countries; USA is the top-popular country, followed by India, France, UK, and Turkey.					USA
[2]	Oct '16-March '17	108	33%	20%	8%	5%	2%	
[14]	June-Dec 2016	115	28 countries					
[15]	Oct 2013	113	unknown					
[3]	unknown	228	10,3 %	70,5 %	19,2 %			USA/Canada 70%
[4]	July 2016	485	91 countries.					
[16]	unknown	50	unknown					
[17]	unknown	52	5,8%	15,40 %	55,80 %	0%	23,1 %	Asia/Pacific 55.80%
[19]	Jan 2016 to June 2016.	66	Worldwide					USA 37% Unknown 39%
[20]	Apr.-May 2015	627	66%	14%	19%		1%	Europe 66%
[21]	unknown	222		Brazil				
[22]	Sept 2014 - April 2016	96	67%	8%	4%			Germany 40%
[23]	Feb - April 2013	178	86,5 %	8,90 %	3,40 %	0,60 %	0,60 %	Europe 86,5% (Italy 61,8%)
[24]	unknown	17	unknown					

4 Results

The selected papers use a variety of methods of data collection. Therefore, summarizing this data is not straight forward. In order to make meaningful summaries of the data for each research question, we often discuss the papers in separate groups depending on how the results are reported. For each research question, the summary approach is clearly indicated.

4.1 RQ1 - For Which Aspects (Type of Activities) of Software Development Do Practitioners Use Modelling Languages?

Two surveys [3, 22] asked their participants to answer using a 5-point Likert scale. In order to be able to combine the results of these papers with the results of the other surveys that do not use a Likert scale, the 5-points Likert scale data needed to be transformed to binary data. The general approach used in this paper is to identify the questions of the survey that correspond to the research question and sum the percentages for the points of the Likert Scale that indicate an agreement. For example, in the survey of paper [22] a Likert scale is used to investigate to what extent participants use models for a set of suggested activities in software development (never & rarely – sometimes – often & always). In this case, only one column clearly indicates agreement with the suggested activities (often & always). Regarding the activity 'discuss with colleagues', 79% of participants indicated 'often & always' and therefore consider this as an activity for which they use models. These percentages can now be compared to the results of the surveys not using Likert scale data [2, 4, 10, 13, 15, 18, 20].

The surveys addressing the first research question use a variety of frameworks to base their survey questions on. A first framework are the software architecture viewpoints by Rozanski and Woods [25]: the Context, Functional, Information, Concurrency, Development, Deployment and Operational viewpoints. Paper [10] reports that the Information and Functional viewpoints are modelled by the highest number of participants (99% and 96% respectively). Other viewpoints that were modelled by many participants are the Deployment, Concurrency and Development viewpoints (75%, 66% and 64% respectively). The Operational viewpoint was only modelled by 29% of the participants. Paper [13] presents an evaluation of 113 architectural languages. This paper confirms that the Logical and Information viewpoints are supported by the majority of the architectural languages (91% and 78% respectively). The concurrency viewpoint is supported by 45% of the languages, but the Development, Deployment and Operational viewpoint are supported by a relatively few architectural languages (26%, 15% and 10% respectively). Two additional viewpoints discussed in this paper are the behavioral viewpoint which is supported by 46% of the architectural languages and the Physical viewpoint which is supported by 15% of the architectural languages. The viewpoints that are supported by the most languages according to paper [13] are also the viewpoints that are the most modeled by practitioners according to paper [10].

Two papers designed their survey according to phases in the development lifecycle. While not referring explicitly to the Rational Unified Process (RUP) framework, we used it to unify the results of two papers. The RUP framework [26] contains 6 engineering disciplines (Business modelling, Requirements Engineering, Analysis and Design, Implementation, Testing, Deployment) and three supporting disciplines (Configuration and change management, Project management, Environment). Paper [4] is a survey of open source software developers. In 68% of the projects that use UML for design, the UML models were implemented completely or with minor changes. Paper [20] is a survey in the embedded systems industry. The disciplines where modelling was used the most are Analysis and Design (89.5%), Implementation (74.4%) and Requirements Engineering (64.1%).

Table 2. Software development activities grouped by the dimension of the RE framework.

RE dimension	paper	What are models used for?	% using models frequently for this activity
HIGH frequency of use			
negotiation	[22]	discuss with colleagues	0,790
documentation	[20]	Documentation generation	0,768
	[22]	visualize an idea or concept	0,750
	[4]	Documentation (e.g.: reverse engineered)	0,712
elicitation	[4]	Design/architecture for (existing/new) systems parts	0,705
	[22]	help me think, sketch a thought	0,700
	[20]	Understanding a problem	0,670
-	[20]	Code generation	0,762
	[15]	Simulation	0,681
	[15]	Code generation	0,664
MEDIUM frequency of use			
negotiation	[22]	communicate with clients	0,440
	[20]	Communication	0,405
documentation	[20]	Documenting designs	0,578
	[15]	Information/documentation	0,531
	[3]	transcribing a design into digital format	0,517
	[22]	document a system or code	0,450
elicitation	[22]	capture domain knowledge	0,570
	[3]	developing a design	0,551
	[22]	design systems or code	0,550
	[22]	capture technical requirements	0,430
	[22]	capture client requirements	0,470
	[3]	brainstorming possible designs	0,448
LOW frequency of use			
negotiation	[22]	negotiate consensus	0,260
	[22]	define contract (model is part of contract)	0,180
documentation	[22]	reconstruct knowledge from source code etc.	0,200
-	[15]	Test case generation	0,398
	[20]	Test-case generation	0,384
	[15]	Structural consistency checks	0,381
	[20]	Model-to-Model (M2M) transformation	0,373
	[22]	generate prototype code	0,350
	[3]	generating code (code editable)	0,344
	[15]	Traceability	0,336
	[15]	Behavioral consistency checks	0,336
	[3]	prototyping a design	0,322
	[3]	generating all code	0,310
	[15]	Timing analysis	0,283
	[22]	generate production code	0,280
	[22]	create a DSL	0,260

(*continued*)

Table 2. (*continued*)

	[15]	Safety compliance checks	0,230
	[15]	Formal verification	0,221
	[22]	look up product details	0,200
	[4]	Verification	0,178
	[20]	Model simulation	0,151
	[15]	Reliability analysis	0,142
	[4]	Refactoring	0,141
	[4]	Code generation	0,129
	[4]	Models are test data	0,060

Finally, some surveys asked questions that were not based on a framework. To be able to compare the answers from this group, we grouped the topics from surveys [3, 4, 15, 20, 22] according to the "Dimensions of RE" framework [27] as shown in Table 2. Table 2 lists the topics with high (≥70%), medium (<70% and ≥40%) and low (<40%) frequency of use separately. Papers [15] and [20] focus on software engineering for embedded systems and report that the main purposes are code generation and documentation. The activities for which more than 70% of participants indicate that models are frequently used, lead to the conclusion that the main purposes are documentation and elicitation of requirements and being used for negotiation to a lesser extent. The survey in [2] is discussed separately since it is the only survey where participants chose on average just one purpose for their models. Therefore, we cannot compare the percentages reported in paper [2] to the other surveys. The activities in paper [2] were coded into six purposes. We mapped the first five onto the RE dimensions. The sixth coded purpose was Requirements Engineering. Similar to the other surveys, the majority of participants chose an elicitation activity: 42.6% participants selected an elicitation activity, 28.7% of participants selected a documentation activity, 26.9% selected a negotiation activity and 11.1% of participants selected an activity that the authors classified under the general category "Requirements Engineering".

Paper [18] is a systematic literature review on BPMN that concludes that documentation is the main activity that BPMN is used for. A final observation is that the purposes that fall outside the three "Dimensions of RE" score low (Code generation, creating a DSL, prototyping design, reconstruct knowledge from source code…).

4.2 RQ2 - Which Other Modelling Languages Are in Use Next to BPMN and UML and How Often Are They Used?

For this research question, we only considered general papers that do not focus on a specific language: surveys [2, 3, 14–16, 20, 22]. In 6 of the 7 papers, UML is reported as the most used modelling language, with at least 41.6% of the participants in each of the surveys using it. Paper [3] shows slightly different results. Here, ERD is the most popular modelling language with 40% of participants using it very often, followed by structured design models which were used very often by 38% of the participants. The use of UML is split into three categories in this survey. 34% of the participants use UML

2.* very often, 33% of the participants use any version of UML very often and 27% of the participants use UML 1.* very often. Paper [3] is the only paper with American authors and it reports that 70% of the participants are from the USA or Canada as shown in Table 1.

In papers [10, 14, 20, 23], participants were also asked about the UML diagram types they use the most. Across all papers alike, class diagrams are among the top 2 most used diagram types.

According to papers [3, 14, 15, 20, 22] a secondary group of languages are Domain Specific Languages. The percentage of use varies from 8% [15] to 36% [14].

BPMN is mentioned by 2 papers that report contradicting results. Paper [2] reports 34,6% of the 108 participants using BPMN and paper [21] reports only 12 of the 96 participants (12,5%) using BPMN. A notable difference is that in paper [22] 40% of the respondents are from Germany while the population surveyed in paper [2] is more diverse. Another difference is that paper [22] specifically targets Industrial Software Development. In five surveys [2, 14, 15, 20, 22], 5% to 21.2% of the participants use SysML and in two surveys [14, 20], 16.9% to 19% of the participants use UML profiles.

4.3 RQ3 - Which Problems/Difficulties/Requirements Do Practitioners Have with Model-Driven Engineering?

Out of the 18 selected papers, 14 discuss this topic. Ten of these are surveys and asked about the requirements that practitioners have for modelling languages, shortcomings in model-driven engineering, disadvantages of modelling languages and tools and inhibitors for successful adoption of model-driven engineering. Six surveys [2, 3, 15, 19–21] asked their participants to answer using a 5-point Likert scale. The approach described in RQ1 is applied. For example, in the survey of paper [20], regarding the problem 'difficulties with model-level debugging', 6% of participants indicated 'strongly agree' and 68% of the participants indicated 'agree'. Therefore, we summarize this as 74% of participants considering 'difficulties with model-level debugging' a problem of MDE environments or tools.

These percentages can now be compared to the results of the surveys not using Likert scale data [14, 16, 17, 24]. For these 10 surveys, the problems that were identified by more than 60% of the participants were consolidated. The next step is to code the different responses in order to summarize them. This was done by mapping the responses onto one of two complementing frameworks: the Physics of Notations framework [28] and Lago's framework [29]. The Physics of Notations framework refers to characteristics of a Notation in terms of its ease of understanding, while Lago's framework provides an overview of practical requirements for architectural languages based on a survey with practitioners. The remaining problems that could not be mapped to one of these two frameworks were sorted in categories under 'Other' in Table 3.

The problems can be summarized in Table 3. The 5 problems that were identified by three or four separate surveys are highlighted in the table.

Table 3. Practitioners' issues with model-driven engineering.

Issues	[2]	[14]	[15]	[3]	[16]	[17]	[19]	[20]	[21]	[24]	**To-tal**
Lago			**3**	**5**	**1**			**2**	**3**		**14**
(Lago) automated analysis					1			2			3
(Lago) extensibility and customization				3							3
(Lago) large-view management				1					1		2
(Lago) programming framework			1								1
(Lago) support for collaboration			1						1		2
(Lago) support for software architecture-centric design									1		1
(Lago) support for versioning			1								1
(Lago) support to specify nonfunctional properties/(Lago) automated analysis				1							1
Other		**1**	**9**	**7**			**8**	**2**	**1**		**28**
-				1			1				2
cost			1								1
cultural/social/management inhibitors							4				4
customer support by tool manufacturer								1			1
external tool integration/model transformation & export			5	1					1		7
high effort in acquiring MBSE skills		1	1	1			3				6
high effort in using tool			2	2				1			5
synchronization between model and code				2							2
PoN	**9**			**1**		**9**		**1**		**3**	**23**
(PoN) cognitive fit	1					1		1		2	5
(PoN) cognitive fit/(PoN) cognitive integration										1	1
(PoN) cognitive integration	1					1					2
(PoN) complexity management	1					1					2
(PoN) dual coding	1					1					2
(PoN) graphical economy	1					1					2
(PoN) perceptual discriminability	1					1					2
(PoN) semantic transparency	1					1					2
(PoN) semiotic clarity	1					1					2
(PoN) visual expressiveness	1			1		1					3

Finally, there are four papers addressing this research question that don't include a survey [11–13, 18]. In paper [11] the authors provide a list of requirements for tools that practitioners find important. Paper [12] is a systematic literature review that list quality criteria used in primary studies and their frequency of appearance. Paper [13] lists the requirements for modelling languages that are part of Lago's framework. Paper [18] is a systematic literature review on BPMN that reports two common disadvantages of BPMN: "Less than 20% of the BPMN vocabulary is common used" and "Utilization in specific domain can be difficult". The problems mentioned by these four papers can be mapped to the above mentioned problems, whereby each problem is mentioned at most once or twice.

Table 4. Areas of benefits of model-driven engineering according to practitioners

Area of benefits	[14]	[15]	[3]	[4]	[19]	[20]	[22]	Total
RUP discipline - Analysis and Design	1	2	4	5	5	1	1	19
Quality management		7			2	3	1	13
RUP discipline - Implementation		4	4	1		3		12
RUP discipline - Requirements Engineering		2	1		4	1	1	9
RUP discipline - Deployment		2	1			2	1	6
General	1		2				2	5
RUP supporting discipline - Project management						1	2	3
Cost savings		1				1		2
RUP discipline - Testing						2		2
RUP discipline - Business modelling							1	1
Accessibility/support	1							1
RUP discipline - Implementation & Testing							1	1

4.4 RQ4 - What Are the Benefits of Model-Driven Engineering According to Practitioners?

Nine of the 18 selected papers discuss this topic, out of which 8 are surveys: [3, 4, 14–16, 19, 20, 22]. Again, the data collection approaches are quite varied. From each survey, we selected the benefits that were confirmed by at least 60% of the participants. For surveys using a Likert scale, the same approach was used as in RQ3. We collected all these benefits and mapped them onto a discipline of the Rational Unified Process framework presented. Some of the benefits could not be placed in a discipline. They were put in one of the following categories: Quality management, General benefits, Accessibility/support and Cost savings. Most of the benefits fell under these five categories:

- RUP discipline – Analysis and Design (e.g.: making complex design decisions, brainstorming & collaboration, understanding & explaining a system's behavior etc.)
- RUP discipline – Requirements Engineering (requirements analysis, traceability of requirements, etc.)
- Quality management (integrity, reliability, etc.)
- RUP discipline – Implementation (code generation, reusability, shorter development time, etc.)
- RUP discipline – Deployment (maintainability, modifications, etc.)

Paper [18] is the only paper without a survey. Three of the six benefits of BPMN that were identified in this paper can be categorized under 'accessibility/support'. The others fall under 'RUP discipline – Analysis and Design', 'RUP discipline – Implementation' and ‘RUP discipline – Business modelling’. Table 4 presents the results in descending order of number of mentioned benefits.

5 Discussion

In general, there is a lack of unified terminology used in the papers to describe the issues that practitioners face when they use modelling languages. Many papers also do not provide enough details about the demographics of the survey’s sample, or do not provide the details about the questionnaire used. All these elements make it hard to compare and summarize results across different surveys.

For RQ1, we noticed that the modelling purpose ’Documentation (e.g.: reverse engineered)’ scores high in paper [4], but the similar purpose ‘reconstruct knowledge from source code etc.’ in paper [22] scores low. This could be due to differences in the populations of the surveys. Paper [4] is a survey with 485 respondents across 91 different countries, and paper [22] is a survey with 96 participants of which 40% are from Germany. Similarly, For RQ2, the reported results in the papers varied. Also these differences in results could be due to differences in the population or a different focus/domain of expertise of the respondents.

When contrasting the results of RQ1 and RQ4, we see that in RQ1 we concluded that modelling is mostly used for the ‘Analysis and Design’, ‘Implementation’ and ‘Requirements Engineering’ discipline of the RUP framework [26]. These disciplines were also considered among the five main benefits of modelling in RQ4.

When contrasting the results of RQ1 and RQ3, the results for RQ1 reveal that besides the information and functional viewpoint, modelling is frequently used for the other viewpoints as well, and -considering the lifecycle phases- not only for the early phases of analysis and design and requirements engineering, but also for implementation. Given the high frequency of goals in the range of communication, the fact that of all 65 reported problems in Table 4, 23 (35%) are PoN -related problems, is a significant issue.

When contrasting the results of RQ3 and RQ4, we see that while some of the main benefits of modelling are collaboration and understanding and explaining the system’s behavior according to the results of RQ4, RQ3 indicates that there are still many practitioners that experience understandability issues related to the PoN framework. This contradiction could be due to the varying levels of experience and fields of expertise of the participants of the surveys. It is therefore important to keep improving the understandability of modelling languages to account for the diversity in users.

Due to the diversity in scope, perspective and reporting in the papers selected for this review, it is not possible to draw conclusions about the evolution of Model-Driven Engineering in the last five years. When contrasting the results of this literature review with the related research, we find that in paper [5], which was published in 2015, the authors identify the complexity of modelling tools as one of the seven main issues of the model-driven engineering field. In particular, they discuss the lack of usability of the tools and the problems with tool interoperability. Based on the results of RQ3 these are still big issues that practitioners face. A second problem identified in paper [5] that practitioners still have is the lack of automated analysis of the quality of the models. This problem also surfaced in the results of RQ3. One of the results reported in paper [6] is that OWL is the most used ontology related language, with very few papers discussing UML. In our results for RQ2, we see that UML is the most used modelling language in practice. The main benefits of ontologies in paper [6] are related to improvements in quality and requirements management. This corresponds to the benefits found in RQ4.

The results presented in this paper are also subject to some validity threats. The first threat to the validity of this paper is the limitation of the search. While we followed Kitchenham's guidelines and used two databases containing millions of papers, it is nevertheless possible that some papers were missed by the query. We nevertheless consider the current set sufficiently large to allow for an acceptable level of confidence in the results. Assuming that all surveys have been issued to disjoint populations, in total a population of around 2500 practitioners is reported on in this meta-survey.

A second threat to validity is the publication lag. The survey that was conducted most recently is from 2018, but often the surveys are older or the period wherein the survey was conducted is not reported. While we were aiming for reviewing the period of 2015–2020, in practice, the data itself relates to the period of 2013–2018.

A Final threat to validity is that the surveys were based on different frameworks and often incomplete in the reporting of the demographic information and in the explanation of their questions and answer options. While we attempted to be as careful as possible when matching results, we nevertheless were sometimes required to make certain interpretations in order to summarize the results.

6 Conclusion

This meta-review of survey and review papers published in the last 5 years provides an overview of how practitioners use modelling languages and what they perceive to be the benefits and issues of Model-Driven Engineering. We found that modelling is very highly used in the functional and information viewpoint, during the design phase of the software development lifecycle and for documentation and elicitation of requirements. The most frequently used modelling language is UML and the most frequently used UML diagram types are class diagrams. While practitioners experience benefits of modelling for analysis & design, requirements engineering, quality management, implementation and deployment, they still struggle with a number of significant issues. In particular, the most common reported problems are external tool integration/model transformation & export, cognitive fit, visual expressiveness, high effort required in acquiring skills, automated analysis and high effort required in using tools.

The issues identified in this review may serve for the identification of a research agenda. In particular, a first point would be to identify possible improvements for usability of the tools. A second closely related issue to address is the interaction, integration or interoperability of different tools. Finally, a third issue we want to investigate further is how the understandability of models can be enhanced, either by improving current languages or by providing additional help for non-proficient users.

References

1. Soley, R.: Model Driven Architecture Model Driven Architecture Preface: OMG's Accomplishments, p. 308 (2000)

2. van der Linden, D., Hadar, I., Zamansky, A.: What practitioners really want: requirements for visual notations in conceptual modeling. Softw. Syst. Model. **18**(3), 1813–1831 (2018). https://doi.org/10.1007/s10270-018-0667-4
3. Badreddin, O., Khandoker, R., Forward, A., Masmali, O., Lethbridge, T.C.: A decade of software design and modeling: a survey to uncover trends of the practice. In: Proceedings - 21st ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MODELS 2018, pp. 245–256 (2018). https://doi.org/10.1145/3239372.3239389
4. Ho-Quang, T., Hebig, R., Robles, G., Chaudron, M.R.V., Fernandez, M.A.: Practices and perceptions of UML use in open source projects. In: Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track, ICSE-SEIP 2017, pp. 203–212 (2017). https://doi.org/10.1109/ICSE-SEIP.2017.28
5. Giraldo, F.D., España, S., Giraldo, W.J., Pastor, O.: Modelling language quality evaluation in model-driven information systems engineering: a roadmap. In: International Conference on Research Challenges in Information Science, pp. 64–69 (2015)
6. Dermeval, D., et al.: Applications of ontologies in requirements engineering: a systematic review of the literature. Requirements Eng. **21**(4), 405–437 (2015). https://doi.org/10.1007/s00766-015-0222-6
7. Kitchenham, S., Charters, B.: Guidelines for performing systematic literature reviews in software engineering. Technical Report Ver. 2.3 EBSE, vol. EBSE-2007-, no. School of Computer Science and Mathematics, p. 65 (2007). https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf
8. Petre, M.: "No shit" or "Oh, shit!": responses to observations on the use of UML in professional practice. Softw. Syst. Model. **13**(4), 1225–1235 (2014). https://doi.org/10.1007/s10270-014-0430-4
9. Baltes, S., Diehl, S.: Sketches and diagrams in practice. In: Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering, pp. 530–541 (2014). https://doi.org/10.1145/2635868.2635891
10. Ozkaya, M., Erata, F.: A survey on the practical use of UML for different software architecture viewpoints. Inf. Softw. Technol. **121**, 106275 (2020). https://doi.org/10.1016/j.infsof.2020.106275
11. Ozkaya, M.: Are the UML modelling tools powerful enough for practitioners? A literature review. IET Softw. 13(5), 338–354. Institution of Engineering and Technology (2019). https://doi.org/10.1049/iet-sen.2018.5409
12. Awadid, A., Nurcan, S., Ayachi Ghannouchi, S.: On leveraging the fruits of research efforts in the arena of business process modeling formalisms: a map-driven approach for decision making. Softw. Syst. Model. **18**(3), 1905–1930 (2018). https://doi.org/10.1007/s10270-018-0689-y
13. Ozkaya, M.: The analysis of architectural languages for the needs of practitioners. Softw. Pract. Experience **48**(5), 985–1018 (2018). https://doi.org/10.1002/spe.2561
14. Ozkaya, M.: Do the informal & formal software modeling notations satisfy practitioners for software architecture modeling? Inf. Softw. Technol. **95**, 15–33 (2018). https://doi.org/10.1016/j.infsof.2017.10.008
15. Liebel, G., Marko, N., Tichy, M., Leitner, A., Hansson, J.: Model-based engineering in the embedded systems domain: an industrial survey on the state-of-practice. Softw. Syst. Model. **17**(1), 91–113 (2016). https://doi.org/10.1007/s10270-016-0523-3
16. Ozkaya, M.: What is software architecture to practitioners: a survey. In: MODELSWARD 2016 - Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development, pp. 677–686 (2016). https://doi.org/10.5220/0005826006770686
17. Saleh, F., El-Attar, M.: A scientific evaluation of the misuse case diagrams visual syntax. Inf. Softw. Technol. **66**, 73–96 (2015). https://doi.org/10.1016/j.infsof.2015.05.002

18. Kocbek, M., Jošt, G., Heričko, M., Polančič, G.: Business process model and notation: the current state of affairs. Comput. Sci. Inf. Syst. **12**(2), 509–539 (2015). https://doi.org/10.2298/CSIS140610006K
19. Huldt, T., Stenius, I.: State-of-practice survey of model-based systems engineering. Syst. Eng. **22**(2), 134–145 (2019). https://doi.org/10.1002/sys.21466
20. Akdur, D., Garousi, V., Demirörs, O.: A survey on modeling and model-driven engineering practices in the embedded software industry. J. Syst. Architect. **91**, 62–82 (2018). https://doi.org/10.1016/j.sysarc.2018.09.007
21. Farias, K., Gonçales, L., Bischoff, V., da Silval, B., Guimarães, E., Nogle, J.: On the UML use in the brazilian industry: a state of the practice survey. In: Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE, pp. 372–375 (2018). https://doi.org/10.18293/SEKE2018-183
22. Störrle, H.: How are conceptual models used in industrial software development? A descriptive survey. In: ACM International Conference Proceeding Series, vol. Part F128635, pp. 160–169 (2017). https://doi.org/10.1145/3084226.3084256
23. Fernández-Sáez, A.M., Caivano, D., Genero, M., Chaudron, M.R.V.: On the use of UML documentation in software maintenance: results from a survey in industry. In: MODELS, pp. 292–301 (2015)
24. Monsalve, C., April, A., Abran, A.: Business Process Modeling with Levels of Abstraction (2015)
25. Rozanski, N., Woods, E.: Software Systems Architecture. https://www.viewpoints-and-perspectives.info/home/viewpoints/. Accessed 16 Mar 2021
26. Kruchten, P.: The Rational Unified Process: An Introduction, 3rd edn. Addison-Wesley, Boston (2000)
27. Pohl, K.: The Requirements Engineering Framework. Springer, Berlin Heidelberg (2010)
28. Moody, D.: The physics of notations toward a scientific basis for constructing visual notations in software engineering. IEEE Trans. Softw. Eng. **35**(6), 756 (2009). https://doi.org/10.1109/TSE.2009.67
29. Lago, P., Malavolta, I., Muccini, H., Pelliccione, P., Tang, A.: The road ahead for architectural languages. IEEE Softw. **32**(1), 98–105 (2015)

FloWare: An Approach for IoT Support and Application Development

Flavio Corradini, Arianna Fedeli, Fabrizio Fornari(✉), Andrea Polini, and Barbara Re

Computer Science Division, University of Camerino, Camerino, Italy
{flavio.corradini,arianna.fedeli,fabrizio.fornari,
andrea.polini,barbara.re}@unicam.it

Abstract. With the advancement of computing technology, we are witnessing the dawn of a new era of the Internet of Things (IoT) paradigm in which objects equipped with sensors, actuators and processing capabilities communicate with each other to serve a given goal. The IoT's intrinsic nature, which uses heterogeneous devices, resources and different communication protocols, complicates IoT applications' design, development, and validation. Reducing the complexity of building IoT applications is one of the current challenges in this area.

To address this challenge, we focus on a model-driven approach to support IoT systems' management and the development of IoT applications. In particular, we propose the FloWare approach and its toolchain, which combine Software Product Line and Flow-Based Programming paradigms to manage the complexity in the various stages of the IoT application development process. An automatic transformation procedure generates the final IoT application, an executable Node-RED flow, starting from a configuration of the designed Feature Models.

Keywords: IoT · Model-driven · Feature model · Flow-based programming · Node-RED

1 Introduction

The Internet of Things (IoT) is a paradigm that gained ground in the scenario of modern wireless telecommunications [3]. The basic idea of this concept is the pervasive presence, all around us, of a variety of things or objects – such as Radio-Frequency IDentification (RFID) tags, sensors, actuators – which, through unique addressing schemes, can interact with each other and cooperate with their neighbours to reach common goals. The development of IoT applications is complex and demanding activities that require the investment of resources and carry a significant risk of failure [14].

One of the main issue recognised in the state of the art [18] is the lack of a software engineering methodology to support the entire IoT applications development life-cycle. This gap results in difficulties in design and developing applications that can be addressed through a Model-Driven Engineering

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 350–365, 2021.
https://doi.org/10.1007/978-3-030-79186-5_23

(MDE) approach [26]. The MDE methodology allows to automatically generate easily maintainable and better-quality software based on modelled system requirements. This methodology can foster software productivity in IoT scenarios supporting better solutions, reducing development time and costs, and increasing its quality [18]. In MDE methodology, different models can be designed to describe IoT system requirements and functionalities. Models gaining attention in IoT scenarios have been consolidated in the academy and industry under the Software Product Line (SPL) paradigm. The SPL paradigm refers to software engineering methods for creating a collection of similar software systems from a common model [20]. One of the most applied SPL models in IoT scenarios is the Feature Model [10], which is used to model all possible points of variability and commonality of IoT domains [5]. Feature Models can be used to derive documents, other models, or chunks of code that the developer can arrange for defining a software application.

Referring to the code and programming of IoT applications, the increasing complexity of application development in the IoT domain is well handled by applying the Flow-Based Programming (FBP) paradigm [27]. The FBP is a type of data-flow programming mainly focused on data and their manipulation, differing from other techniques such as workflow-based in which the focus is on representing activities and their execution order [15]. The FBP suits well in the context of the IoT, where the pillar is the reception, manipulation and sending of data as it sees the application as a set of "black-box" software processes; each process has a specific behaviour and exchange data through predefined connections. Thanks to visual tools based on this programming paradigm, it is possible to easily connect different processes to form different applications without changing their internal content [23].

To reduce the complexity and facilitate all the various stages of IoT applications development, we propose *FloWare*, an MDE approach that combines Software Product Line and Flow-Based Programming paradigms. Our approach supports the IoT applications development from the design to software development, including different phases, actors, steps, and artefacts that allow realising model-to-code transformation. We also propose a toolchain and an open-source core component, named *FloWare Core*[1], to support the approach. Finally we describe how FloWare can be applied to a realistic scenario concerning the management of IoT devices and the development of IoT applications in a Smart Campus.

The paper is organised as follows. Section 2 gives an insight on related works concerning the modelling and development of IoT applications based on Software Product Line and Flow-Based Programming paradigms. Section 3 describes the *FloWare* approach for modelling and developing IoT applications. Section 4 presents an introduction to our supporting toolchain. Section 5 reports on the application of *FloWare* in practice. Section 6 concludes the paper by touching upon directions for future work.

[1] FloWare Core: http://pros.unicam.it/floware/.

2 Related Work

Nowadays, the combination of Flow-Based Programming and Software Product Line paradigms is not a consolidated trend concerning IoT applications' development, but it presents a great potential [22].

For what concerns the use of FBP in different IoT domains, examples from [6,7,11,21,24] shows its adaptability in small-medium cases like Smart City, Smart Home, Smart Building, Smart Laboratory. In [12,13,25] are presented applications on large scenarios like Smart Logistic, Smart Hospitals and Smart Military Environments. The literature mentioned above highlights the use of FBP to facilitate the interconnection between heterogeneous devices using different technologies. In particular, we found a common usage of Node-RED[2], an FBP tool that gained research and industry attention over the last years. Node-RED allows the interconnection of devices and services to develop IoT applications. Thanks to the availability of a visual editor, Node-RED enables the easy creation of IoT applications through the interconnection of asynchronous components isolated from the application context.

Although many research works focus on IoT application development, according to the literature [23] emerges a strong need for standardisation. Currently, no standard approach describes the process of developing IoT applications based on this paradigm. The developer does not have a guiding model to follow during IoT applications; this slows the entire application development process and exposes the developer to introduce errors. Furthermore, the developer must deal with the functional specifications of the devices (e.g., the way the device communicates, the application protocol they use, the port numbers) to correctly manage the devices communications and produced data. It directs away from the developer attention from the actual development of the application. Possible support to help a developer designing and developing an IoT application can come from using a Model-Driven Engineering (MDE) methodology. The use of MDE methodology in the software development process is gaining more attention thanks to a high level of abstraction using models as a base for creating the software [9].

The use of Feature Models of the Software Product Line paradigm presents positive evidence for the adaptation in the context of IoT [10]. In [29] and [17], Feature Models are used to model complete heterogeneous systems like Wireless Sensor Networks and Body Area Networks. In [2,19], different authors design the same Smart Home scenario using different levels of abstraction and focusing on different functional requirements. In [16] and [1], Feature Models represent the design configuration of a single IoT device or a set of devices and their communication protocols without linking them to a specific IoT application domain. In [8], the scope is to manage the evolution of a family of middleware for smart environments using cardinality-based Feature Models to express the structural variability of all the involved devices, applications, network protocols. In [28], Feature Models are used to support the representation of a Body-Area

[2] Node-RED: https://nodered.org/.

Network application to analyse functional and non-functional requirements in IoT-oriented healthcare applications. In [4], the authors propose a model-based reconfiguration engine, that uses Feature Models, for dynamically reconfiguring IoT system architectures. The paper presents a Smart Home example scenario to demonstrate how an autonomic reconfiguration can be achieved using Feature Models at run-time.

Although previous works focus on the use of Feature Models for the IoT, most of them apply the models mainly to describe the reference domain of their IoT system. In our approach, we use Feature Models to design the IoT domain and support and guide the developers/users in the software development, using the IoT system configurations derived from the models to produce code that automatically allows the devices' interconnection and data visualisation.

It is for filling the gaps that emerged from the literature that we present in the following sections our *FloWare* approach, which combines Feature Models and Flow-Based Programming to provide an MDE methodology for facilitating the development of IoT applications starting from the modelling of IoT systems and devices.

3 The FloWare Approach

In this section, we present our *FloWare* approach, which supports the modelling and development of IoT applications. We describe the various *phases*, *actors*, *steps* and *artefacts* that characterise our approach, as depicted in Fig. 1.

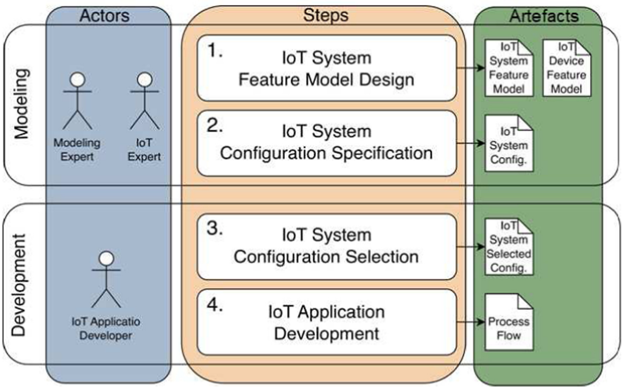


Fig. 1. FloWare approach

Phases. The approach is divided into two main phases: the *Modelling* and the *Development* phases. The modelling phase involves the design and configuration of Feature Models to represent the entire IoT domain. The development phase involves the automatic transformation of models into code to develop the actual IoT application.

Actors. As described in [18], IoT application development is a multidisciplinary process that intersects heterogeneous knowledge from the different involved actors. These actors have different roles in the development process. In our approach, in the modelling phase, we require the involvement of human actors such as a *Modelling Expert* (*ME*), an expert capable of designing and representing specific domains using modelling languages and tools, and an *IoT Expert* (*IoTE*), an expert of the Internet of Things domain responsible for the management of IoT devices deployed (or to deploy) in the IoT system. Instead, the development phase requires an *IoT application developer* to exploit the potential provided by the approach to develop IoT applications.

In the following, the various steps and artefacts (respectively indicated in bold and italic) involved in the approach are described in details.

IoT System Feature Model Design. The ME and the IoTE are in charge of defining two Feature Models to represent the entire scenario (Step 1 of Fig. 1): the *IoT System Feature Model* and the *IoT Device Feature Model*. The required Feature Models are designed based on the ME and IoTE expertise, or they can use already deployed models saved in a common repository. The IoT system Feature Model provides an overview of an IoT system's generic domain and all functionalities it could present; the second provides an overview of IoT devices' characteristics. Figure 2 reports examples of the two models that can be defined to represent a simple IoT system and device. The experts can start the modelling activities from a model presenting a similar skeleton and then modify features and relations according to the modelled IoT system and devices' needs.

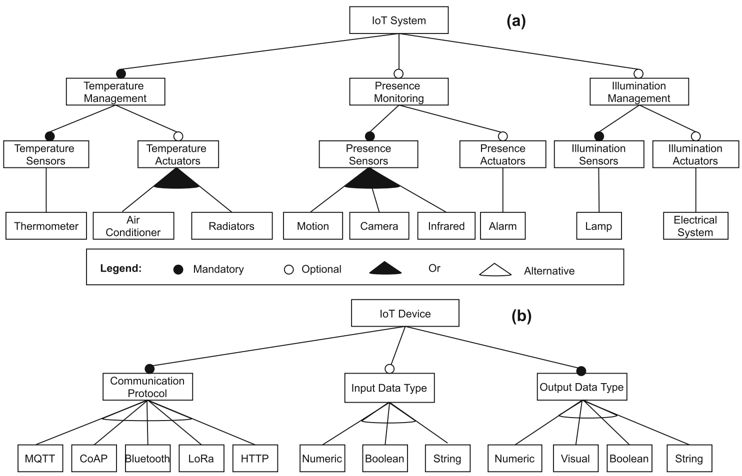


Fig. 2. (a): IoT system feature model; (b): IoT device feature model

For the IoT system Feature Model, presented in Fig. 2(a), the tree's root, named IoT System, will be substituted with a more appropriate term to represent the scenario to be specified (e.g., Smart University). Then, different features linked to the root can be represented to describe the various functionalities that the considered IoT System can have. Some of these functionalities are linked by a mandatory link, so they must be selected in the configuration phase; others are optional and can be inserted into the configuration according to appropriate needs. For each functionality inserted, it is necessary to represent all the devices involved in the scenario, dividing them between sensors and actuators. For the IoT devices Feature Model shown in Fig. 2(b), one of the primary information to be inserted regards the communication protocols used to interact with the devices. Other information necessary to allow the correct manipulation of the devices' data refers to the data types that the device can handle. Regarding the data that the device sends or receives, it is mandatory to define if they are numeric values, Boolean, etc., to allow the system to process them correctly.

IoT System Configuration Specification. The designed Feature Models will be used as a starting point to specify the *IoT System configuration* in a scenario. The configurations based on the Feature Models can be related to deployed IoT systems or under-deployment systems. IoT configurations represent different instances of the same model and are usually different based on several factors depending on how the IoT system is deployed. For example, referring to Fig. 2(a), the temperature management functionality could be realised in one location using temperature sensors and an air conditioning system, while in another site through the use of radiators. Concurrently, other IoT systems may need to implement features such as presence monitoring rather than illumination management to meet the use cases needs. This differentiation produces different configurations starting from the same model.

An example of a valid IoT System configuration is the one reported in Fig. 3(a), where the IoT System is composed of two functionalities, one related to the temperature management, as a mandatory functionality to represent the configuration, and the presence monitoring, with their corresponding devices.

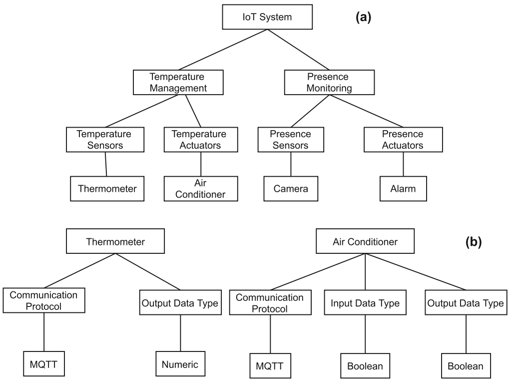


Fig. 3. (a): IoT system configuration; (b): different IoT devices configurations

The features chosen in the IoT system configuration include the use of devices divided between sensors and actuators, which need to be configured. Following the two models in Fig. 3(b), we can see how the two devices use the same protocols to communicate but provide different types of data to be processed. In particular, if the device is an actuator, it also includes the input data type that specifies the data type to receive as input.

In addition, the ME and the IoTE can specify where IoT devices are deployed (building name, room name, geolocation, etc.) together with other specific information according to each device's chosen protocol. Devices related information can include: an identifier, stored together with its location (e.g., GPS coordinates and a known nomenclature for defining room names), an MQTT broker address, and a topic (UTF-8 string) used to access the device data. The defined IoT System Configuration with all specified information will then be made available to the IoT Application developer that desires to develop an IoT application for one or more IoT systems.

IoT System Configuration Selection. It allows an IoT Application Developer to choose which IoT system configuration to interact with. Starting from a configuration like the one reported in Fig. 3(a), the IoT application developer can choose which functionality of the deployed system to access together with the respective IoT devices. This choice will be guided by the developer requirements for the specific application and the information stored, which allows distinguishing between devices of the same type (e.g., the position or the type of device could be a selection factor). For example, referring to the configuration in Fig. 3(a), an IoT developer can build an application using only the Temperature Management functionalities and access the available thermometers' information and command the air conditioner instead of selecting the entire configuration. The *IoT System selected configuration* obtained will be used to generate an IoT application.

IoT Application Development. After the IoT developer specified which functionality of an IoT system and which devices to interact with, the environment for developing the IoT application is configured accordingly. For instance, if the developer specified the need to interact with a temperature sensor that uses an MQTT protocol to communicate, the environment is automatically configured to include the MQTT functionality. In addition, an IoT application is automatically generated in the form of a *process flow* that communicates with the device and shows the obtained values through a dashboard.

4 The Supporting Toolchain

To support the usage of our *FloWare* approach, we defined a toolchain and developed an open-source component named *FloWare Core*. The FloWare Core component is an open-source JavaScript software that can be installed both on the cloud and on less powerful machines such as gateways (for example, raspberry etc.). Thanks to a graphical and intuitive interface, it is possible to easily configure all the IoT systems and devices and to develop IoT applications based on them. Specific information regarding FloWare Core, how to access its source code and set it up, is reported on http://pros.unicam.it/floware/. In the following, we provide a technical description of the toolchain components and we also describe how to use them.

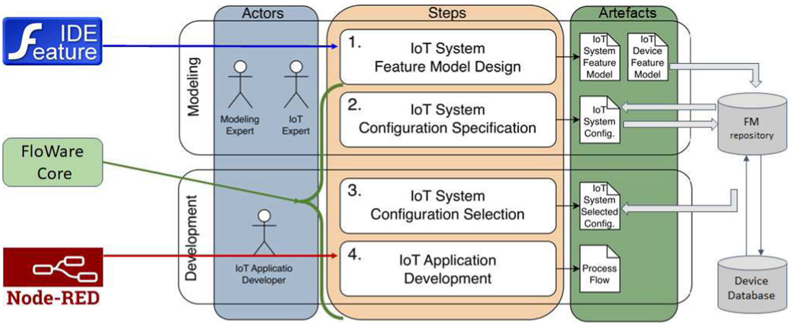


Fig. 4. FloWare approach with the supporting toolchain

As explained in Sect. 3, initially, the IoTE and ME will design the two models. The first model represents the reference IoT system, and the other represents all the IoT devices' functionalities (communication protocol, type of data to be processed, etc.). For the design of the Feature Models, as shown in *Step 1* of Fig. 4, we suggest using Eclipse FeatureIDE[3], which allows to generate a Feature Model and verify its validity. Once the Feature Models have been designed, they will be saved in a Feature Model repository as XML files, as shown in Fig. 4. These models will be used as input for FloWare Core to support the following phases of the approach.

As reported in Fig. 5, the FloWare Core's architecture is formed by several components; each performs a specific task to support the configuration and the automatic generation of the IoT application. In the following, we provide a detailed description of each component.

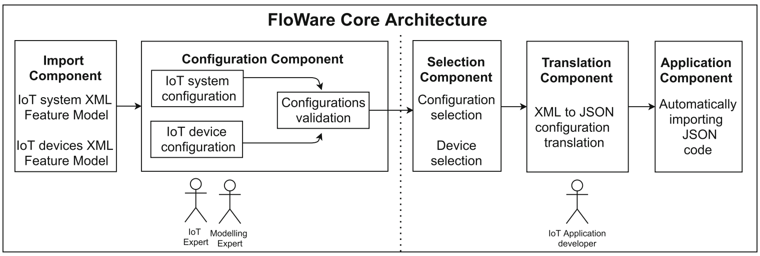


Fig. 5. The FloWare Core architecture. The left side reports components and actors involved in the first phase of the approach. The right side reports components and actors involved in the second phase.

The *Import Component* manages the import of the XML files representing the designed Feature Models. These models serve as a basis to support the configuration of the IoT system and the devices involved. FloWare Core comes with

[3] FeatureIDE: http://www.featureide.com/.

some Feature Model templates that can be used as a starting point for modelling the IoT system and the IoT devices avoiding the complete redesign phase. Once the XML Feature Models are imported, the Configuration Component provides a graphical interface to help the experts in the system configuration and the addition of information regarding the devices involved.

A prototype of the *Configuration Component*, represented in Fig. 6, reports an extract of an IoT system's possible configuration. From this configuration section, the experts can select all the functionalities and devices that will be deployed on the IoT system. Then, for each chosen device, they are requested to fill a form concerning the specific information of each device involved. A user's configuration is valid if it ensures compliance with the Feature Model designed during the first step of the approach and elaborated by the tool. Once the systems and devices' configurations are completed, they will be saved in the repository, while the devices information will be saved in a device database. Both entities communicate to keep track of the systems and related devices involved in the scenario. In particular, for what concerns the devices, all the information are saved following the WoT Standard[4].

Once terminated the devices and systems' modelling phase, the IoT Application developer can use the FloWare Core to support IoT applications' development.

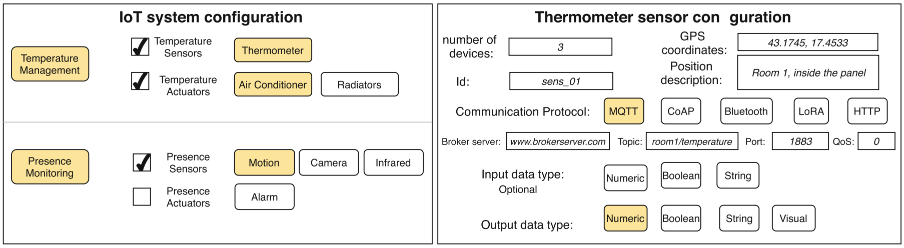


Fig. 6. Prototype of a configuration of an IoT system and a device configuration prototypes using FloWare Core

In the *Selection Component*, the developer selects one of the previously configured systems and specifies which functionalities of the system want to represent in the application. From all the devices involved in the selected scenario, the developer chooses the ones to include in the IoT application. In this way, we leave the developer free to realise the IoT application according to specific needs. For example, the developer may need to represent only a particular room in a building; thus, he/she have the possibility to freely select only the appropriate devices from the entire configuration to represent the desired scenario.

[4] WoT Standard: https://www.w3.org/TR/wot-thing-description.

Then, the *Translation Component* take as input the functionalities and devices selected in XML format and automatically encode them in a JSON file that represents the IoT application. The JSON file is generated to be processable by Node-RED, a development tool incorporated into FloWare Core, to provide a single working and processing solution. Node-RED allows the composition of IoT applications using components; each performs a specific operation and retrieves a result. It also offers the possibility to use some dashboard components to visualise the data obtained from the devices in real-time. In our approach, an IoT application is represented as a Node-RED process flow written in the JSON file. In the generated file, each JSON field represents a different Node-RED component that performs a specific function.

```
<configuration>
<feature automatic="selected"  name="Temperature"/>
<feature automatic="selected"  name="Communication protocol"/>
<feature automatic="undefined" name="MQTT"/>
<feature automatic="undefined" server="www.brokerserver.com"/>
<feature automatic="undefined" port="1883"/>
<feature automatic="undefined" qos="0"/>
<feature automatic="undefined" topic="room1/temperature"/>
<feature automatic="undefined" name="Output Data type"/>
<feature automatic="undefined" name="Numeric"/>
</configuration>
```

```
{ id: 42a56979.773094,
broker: www.brokerserver.com,
port: 1883,
name: Temperature,
qos: 0,
topic: room1/temperature,
type: mqtt input }
```

Fig. 7. Automatic translation from an XML device configuration to JSON component

Figure 7 shows the correspondence between the information collected in the XML configuration and the translation into a JSON field. The conversion in example allows filling each field of the JSON file with the information necessary to start the MQTT communication with the device. There are also generated specific JSON fields able to process the data types received, such as the Numeric type in the example, and send them to other fields that show them in a graphical form. The JSON file also reports the connections between each field, intending to generate a process flow that brings the information from the beginning component (e.g., the component of the receiving device) to the final one (e.g., the data display component).

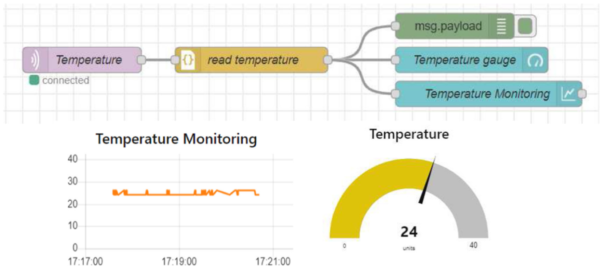


Fig. 8. Temperature monitoring application with dashboard

For each involved device, we obtain a process flow generated automatically by the FloWare Core component, like the one reported in Fig. 8. The generated flow is automatically imported by the *Application Component* into Node-RED, giving the user the possibility to add other functionalities to the application or immediately execute it to inspect the device's data in real-time in a dashboard. In this way, we obtain an IoT application with all the device specifications automatically imported from the previously defined XML configurations. The developer is not obliged to know and insert specific information regarding each device, but the FloWare Core automatically retrieve them from the Feature Model repository and the device database. Then, the developer can expand the generated application by inserting additional Node-RED components.

5 Case Study

In this section, we present a realistic case study to show how our *FloWare* approach can support the management of IoT systems in a Smart Campus, from the models' domain design to the development of a temperature management IoT application.

A university would like to standardise the different IoT systems deployed over the years through the various departments to have a clear vision of the practices adopted. At the same time, the university wants to reason about improvements in terms of new IoT systems that could be deployed in other departments and the development of IoT applications that could take advantage of all the sensors and actuators that have been deployed. Applying *FloWare* in such a scenario, could provide support in the construction of IoT systems and the decoupling of IoT system configuration from IoT application development for the developers that manage these systems. In this way, aim at reducing complexity for the developer, who can work with already configured systems and devices. We can assume that all the university departments are equipped with IoT devices to achieve basic intelligent systems (i.e., temperature and illumination control systems). At the same time, they may include different IoT systems based on the department's specific necessity. In the following, we report some examples of what specific departments need.

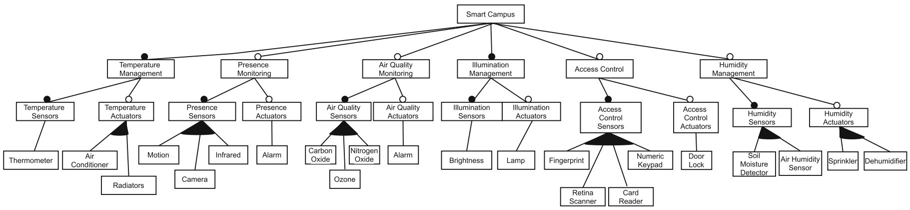


Fig. 9. Feature Model representing the entire Smart Campus functionalities

Inside the *Chemistry department*, for safety reasons, it is mandatory to have a system for air-quality monitoring that can immediately report the levels of CO, CO_2, SO_2 and NO_2 that may derive from experiments conducted in the laboratories. In the *Biology department*, there is the necessity of monitoring experiments involving plants. In particular, there is the necessity to have a system able to control and act on the soil's humidity level and optimally manage the various plants' level of light and temperature. In the *Computer Science department*, on the other hand, there is the need to control accesses to the different server rooms located in it. These rooms can only be opened by authorised staff who can use various access systems, including fingerprint or retina scanner, identification badges.

Using our approach, it is possible to model the entire IoT Smart Campus scenario, as reported in Fig. 9. In particular, are highlighted all the different functionalities and devices that may be made available in the departments. It is important to note that temperature and illumination management are mandatory fields in each configuration as defined at the beginning of the case study; the others are optional. In the same way, IoT devices have to be modelled. The experts can design a model like the one previously reported in Fig. 2(b) or use the FloWare Core's templates.

After designing the Feature Models, the experts use the graphical interface provided by our FloWare Core component to configure all the various devices to use in each department. We defined, using our FloWare Core component, some configurations to represent specific departments' needs in a Feature Models form, as shown in Fig. 10.

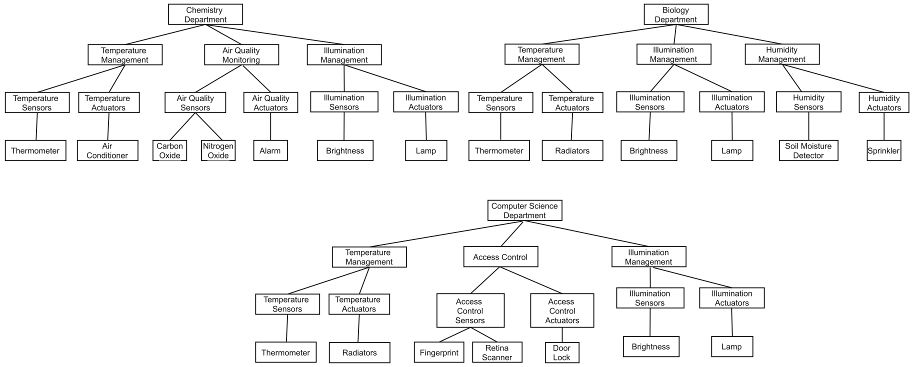


Fig. 10. Departments configuration starting from the IoT system Feature Model

At this point, all the departments' configurations are saved in an Feature Model repository. Starting from these final configurations, we decided to develop an IoT application to test our *FloWare* approach and the FloWare Core component. Our component was installed in a Raspberry Pi to be used as a gateway; thus, it collects the data it receives from the devices and shows them in real-time. Using the FloWare Core's graphical interface, we selected which department, systems, and devices to use in the development of the IoT application, as shown in Fig. 11.

We limited our application to the temperature functionality present in the Chemistry Department' configuration. Then, we selected the devices that we want to automatically include in the application development for that department. In this way, we did not necessarily have to be aware of every device's specifications from which to get information. We only needed to know the functionalities of a device exposed (e.g. temperature measurement) and its location (e.g. place description or GPS coordinates) to identify and choose it uniquely.

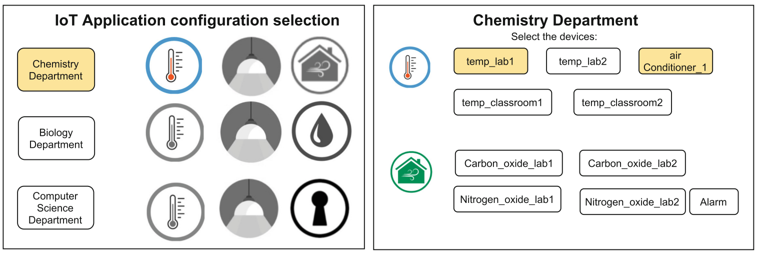


Fig. 11. A prototype of departments and respective devices selection using FloWare Core component

According to the selection, for each chosen device, the FloWare Core component generated an application like the one reported in Fig. 8. In this way, the application can automatically read and show the devices' data incoming in real-time and, if needed, send data to interact with them.

In addition, as shown in Fig. 12, we manually expanded the application to automatise temperature management functionality by adding other components presents in the Node-RED palette, such as a switch conditioner state to automatise the activation of an air-conditioner based on the department's actual temperature. Furthermore, we added an HTTP node to perform an HTTP request to send the data obtained from the devices to a third-party application for external uses.

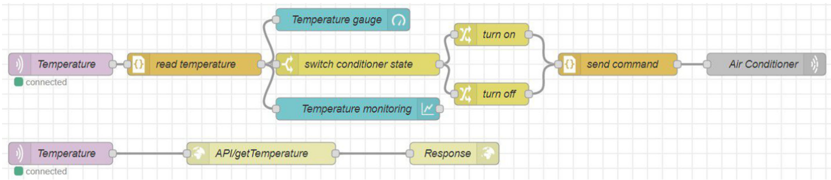


Fig. 12. A complete temperature management IoT application on Node-RED

6 Conclusion

To resolve the complexity of developing IoT applications, we presented the *FloWare* approach, which combines and exploits the potential of the Software Product Lines paradigm, in the form of Feature Models, with the Flow-Based Programming paradigm of the Node-RED tool. Our approach provides a way to facilitate IoT applications development, providing an approach that covers the IoT system's modelling up to its implementation. In particular, we argued that moving the focus on modelling and configuring different IoT systems from experts can help the IoT developers build IoT applications. With our solution, the developers remain unaware of each device's technical specifications because it will inherit them from the experts' previous configurations.

We also presented a supporting toolchain for the *FloWare* approach, focusing on our FloWare Core component. We illustrated its application to a sample case study; the case study involved a Smart Campus. From our studies and tests, the conjunction between the two paradigms seems a suitable choice which requires further studies to consolidate and extend the approach.

As future works, we want to involve other IoT and modelling experts to design and develop heterogeneous IoT systems in practice to validate and test the entire approach on a complete real scenario. Referring to the FloWare Core component, we intend to expand the set of templates including more complex ones to simplify and speed up the development of IoT applications. Those templates may support the developer in setting up interactions with external services for data storage and data analytic.

References

1. Abbas, A., Siddiqui, I.F., Lee, S.U.J., Bashir, A.K.: Binary pattern for nested cardinality constraints for software product line of IoT-based feature models. IEEE Access **5**, 3971–3980 (2017)
2. Alférez, M., Moreira, A., Amaral, V., Araújo, J.: Model-driven requirements specification for software product lines. In: Model-Driven Domain Analysis and Software Development: Architectures and Functions, pp. 369–386. IGI Global (2011)
3. Atzori, L., Iera, A., Morabito, G.: The Internet of Things: a survey. Comput. Networks **54**(15), 2787–2805 (2010)
4. Cetina, C., Giner, P., Fons, J., Pelechano, V.: Autonomic computing through reuse of variability models at runtime: the case of smart homes. Computer **42**(10), 37–43 (2009)
5. Cognini, R., Corradini, F., Gnesi, S., Polini, A., Re, B.: Business process flexibility - a systematic literature review with a software systems perspective. Inf. Syst. Front. **20**(2), 343–371 (2016). https://doi.org/10.1007/s10796-016-9678-2
6. Cognini, R., Corradini, F., Polini, A., Re, B.: Extending feature models to express variability in business process models. In: Persson, A., Stirna, J. (eds.) CAiSE 2015. LNBIP, vol. 215, pp. 245–256. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19243-7_24
7. Cognini, R., Corradini, F., Polini, A., Re, B.: Business process feature model: an approach to deal with variability of business processes. In: Domain-Specific Conceptual Modeling, pp. 171–194. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39417-6_8
8. Gámez, N., Fuentes, L.: Architectural evolution of FamiWare using cardinality-based feature models. Inf. Softw. Technol. **55**(3), 563–580 (2013)
9. Gascueña, J.M., Navarro, E., Fernández-Caballero, A.: Model-driven engineering techniques for the development of multi-agent systems. Eng. Appl. Artif. Intell. **25**(1), 159–173 (2012)
10. Geraldi, R.T., Reinehr, S.S., Malucelli, A.: Software product line applied to the Internet of Things: a systematic literature review. Inf. Softw. Technol. **124**, 106293 (2020)
11. Havard, N., McGrath, S., Flanagan, C., MacNamee, C.: Smart building based on Internet of Things technology. In: International Conference on Sensing Technology, pp. 278–281 (2018)
12. Jain, R., Tata, S.: Cloud to edge: distributed deployment of process-aware IoT applications. In: International Conference on Edge Computing, pp. 182–189. IEEE Computer Society (2017)
13. Jalaian, B., Gregory, T., Suri, N., Russell, S., Sadler, L., Lee, M.: Evaluating LoRaWAN-based IoT devices for the tactical military environment. In: World Forum on Internet of Things, pp. 124–128. IEEE (2018)
14. Lee, I., Lee, K.: The Internet of Things (IoT): applications, investments, and challenges for enterprises. Bus. Horiz. **58**, 431–440 (2015)
15. Morrison, J.P.: Flow-Based Programming: A New Approach to Application Development, 2nd edn. CreateSpace, Scotts Valley (2010)
16. do Nascimento, N.M., Alencar, P.S.C., Lucena, C., Cowan, D.D.: An IoT analytics embodied agent model based on context-aware machine learning. In: IEEE International Conference on Big Data, pp. 5170–5175. IEEE (2018)
17. Ortiz, Ó., García, A.B., Capilla, R., Bosch, J., Hinchey, M.: Runtime variability for dynamic reconfiguration in wireless sensor network product lines. Int. Softw. Prod. Line Conf. **2**, 143–150 (2012)

18. Patel, P., Cassou, D.: Enabling high-level application development for the Internet of Things. J. Syst. Softw. **103**, 62–84 (2015)
19. Pereira, J.A., Maciel, L., Noronha, T.F., Figueiredo, E.: Heuristic and exact algorithms for product configuration in software product lines. In: International Systems and Software Product Line Conference, p. 247. ACM (2018)
20. Pohl, K., Bockle, G.V.D.L.F.: Software Product Line Engineering: Foundations, Principles and Techniques. Springer, Heidelberg (2005). https://doi.org/10.1007/3-540-28901-1
21. Poongothai, M., Subramanian, P.M., Rajeswari, A.: Design and implementation of IoT based smart laboratory. In: International Conference on Industrial Engineering and Applications, pp. 169–173. IEEE (2018)
22. Prehofer, C., Chiarabini, L.: From Internet of Things mashups to model-based development. In: 39th Annual Computer Software and Applications Conference, pp. 499–504. IEEE Computer Society (2015)
23. Ray, P.P.: A survey on visual programming languages in Internet of Things. Sci. Program. **2017**, 1231430:1–1231430:6 (2017)
24. Sicari, S., Rizzardi, A., Coen-Porisini, A.: How to evaluate an internet of things system: models, case studies, and real developments. Softw. Pract. Exp. **49**(11), 1663–1685 (2019)
25. Sicari, S., Rizzardi, A., Coen-Porisini, A.: Smart transport and logistics: a node-red implementation. Internet Technol. Lett. **2**(2), 34 (2019)
26. Sosa-Reyna, C.M., Tello-Leal, E., Alabazares, D.L.: Methodology for the model-driven development of service oriented IoT applications. J. Syst. Archit. **90**, 15–22 (2018)
27. Szydlo, T., Brzoza-Woch, R., Sendorek, J., Windak, M., Gniady, C.: Flow-based programming for IoT leveraging fog computing. In: International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, pp. 74–79. IEEE Computer Society (2017)
28. Venckauskas, A., Stuikys, V., Jusas, N., Burbaite, R.: Model-driven approach for body area network application development. Sensors **16**(5), 670 (2016)
29. Venckauskas, A., Stuikys, V., Toldinas, J., Jusas, N.: A model-driven framework to develop personalized health monitoring. Symmetry **8**, 65 (2016)

A Model-Based Application for the Effective and Efficient Management of Data Associated with Retina-Macula Pathology

José Fabián Reyes Román(✉), Alejandro Marco Palomares, Alberto García Simón, and Oscar Pastor

PROS Research Center, Universitat Politècnica de València, Valencia, Spain
jreyes@pros.upv.es, opastor@dsic.upv.es

Abstract. The macula is the central area of the retina responsible for the best visual acuity and direct central vision, necessary for activities of daily living. An example of this is Age-related Macular Degeneration (AMD), the most common cause of legal blindness in people over 50 in Western countries. This paper aims to provide an Information System (IS) based on Conceptual Models to ensure proper management and analysis of data associated with Macular Diseases. This domain of application lacks an adequate conceptual characterization and it shows a high degree of heterogeneity in the data; in order to carry out this work, we have had the support of experts from the Ophthalmology Services (OS) of three hospitals in the Valencian Community. The main contributions of this work are the following: i) Study and analysis of the domain (Macular Disease), ii) facilitate a Conceptual Model of the Macular Pathology (CMMP) that will serve as ontological support for the development of the solution, and iii) IS that will allow the management and exploitation of the data. This work demonstrates the benefits of the use of conceptual modeling techniques in the clinical environment, which has a direct impact on the improvement of patient care services (prevention and treatment).

Keywords: Conceptual modeling · Information systems · Macula · AMD · CMMP

1 Introduction

Information Systems play a key role in health as they facilitate professionals to provide efficient and effective services to users. Besides, Information Systems improves both the quality of health care and decision making of clinicians, which has a direct impact on treatment cost by generating significant savings [12].

That is why experts in health sciences and Software Engineering (SE) join forces with the aim of facilitating innovative solutions to the challenges that arise in this domain of great importance for society. The Age-Related Macular

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 366–379, 2021.
https://doi.org/10.1007/978-3-030-79186-5_24

Degeneration (AMD) is an example of such a significant challenge due to the high probability that people over 50 years of age have of suffering from this anomaly. The macula, which is a pigmented area that is located inside the retina, plays an essential role in the process of vision because it allows us to have a detailed view of the things around us. The deterioration of this area causes a dependency problem for the person who suffers from it, a reduction in the quality of live, and it is associated with depression. Visual loss is not the only symptom that this disease produces but includes a wide range of symptoms: blurred central vision, alteration of straight lines, shapes, and sizes of things (metamorphosis, macropsia or micropsia), among others.

Historically, no effective treatment for this disease was available, but thanks to the scientific advances in this area, a significant reduction in the risk of progression of this disease has been achieved thanks to the use of vitamins, zinc, and other drugs. However, the use of these drugs is very costly and there is no statistic evidence that their use will reduce the risk of AMD progression in a systematic way.

Furthermore, this domain presents a series of problems from which we can highlight the following: i) the complexity of the domain, based on the context of clinical data, a changing environment, etc., ii) The difficulty in the management of the data, which prevents from obtaining correct data integration and harmonization.

The professionals that work in the Ophthalmology domain have recognized these problems, and they are actively trying to reduce the high costs that are generated in their daily work through the use of new Information System (IS) that allows them to manage (both correctly and precisely) the treatment and evolution of patients who suffer from this type of disease. In addition, the achievement of this objective contributes to the reduction of the great current economic losses thanks to the support and help that an IS provides in decision-making activities. The amount of available software to deal with this problem is limited, and the cost of their license is very high, difficulting the obtention of tools that allow managing all the information generated by specialists in this work environment.

The goal of this work is to present our work in a real-world use case in collaboration with the Ophthalmology Services (OS) of the Conselleria de Sanitat[1]. We developed an IS for the analysis and management of the data of patients treated for macular pathology. In order to provide an IS that improves the management of clinical data, this solution helps in the understanding of an increasingly prevalent pathology that has an increasing pharmaceutical cost and that requires more and more healthcare resources. To achieve this goal, the following specific objectives have been determined:

1. To study and analyze the macular domain together with clinical experts in order to define a Conceptual Model of the Macular Pathology (CMMP). This conceptual model is intended to provide a common ontological framework to discuss with experts in order to increase our understanding of the domain. It will also guide the development of the IS to improve data management.

[1] http://www.san.gva.es/web_estatica/index_va.html.

2. To design and develop an IS that is based on the defined CMMP. This IS will improve the integration and management of the existing data in the macular domain. It will also help clinical experts to improve their decision making.

Our proposal seeks to address this problem by providing a software solution that allows health specialists to deal with the management of treatments for patients with macular pathologies in a efficient and safe way. To achieve this, we had to gather all the knowledge of the domain of the Retina-Macula through many meetings with experts in Ophthalmology. As a result, the Conceptual Model of Macular Pathology (CMMP), which can be seen in more detail in Sect. 3, was generated. This conceptual model allows us to:

1. *Get a shared understanding of the domain. Moreover, the CMMP acts like the single point of truth to integrate and harmonize all existing data, allowing to generate knowledge through the possible application of AI technologies, machine learning, among others.*
2. *Facilitate an IS by applying a model-driven development approach. In this way, an IS can automate tasks and allows monitoring of patients where information is generated to determine which drugs are effective and under what circumstances, preventing medication from being wasted and result in more efficient use of resources.*

In this sense, the conceptual model helps us to achieve a greater understanding and compression of the domain. The paper is divided into the following sections: Sect. 2 presents the state of the art (conceptual modeling applied to the clinical context). Section 3 precisely describes the CMMP. Section 4 contains the design and implementation of the IS based on the CMMP. Finally, Sect. 5 presents the conclusions and outlines future work.

2 Background and Related Work

Within the framework of this work, the development of an IS for the management and treatment of clinical data related to macular diseases is presented. For this reason, this section presents the context in which the research is focused on, treating pathologies such as AMD or age-related macular degeneration (see in more detail in [6]).

Additionally, other research works related to the application of conceptual modeling techniques in different health contexts (i.e., *clinical* and *genomic data*) are also presented.

Conceptual Modeling in Life Sciences

The advantages of using Conceptual Modeling (CM) techniques have been demonstrated several times in clinical practice. Consequently, software developers use CM techniques to add value when defining their processes [1].

Currently, there are a lot of research works whose aim is to provide with sound solutions to the existing problems regarding health. In this section, we present some of these works in different health domains. Although from different areas, they all have the common goal of building a IS to improve the diagnose and treatment of a disease, like breast cancer, or neuroblastoma.

In the context of IS, CM is a common task used to obtain a (general) description of how a system is organized and operated [7]. The application of CM techniques allows the domain of the problem in question to be defined more clearly, defining the entities involved and the relationships between them [4].

The application of CM techniques in the bioinformatics environment with the aim of improving the treatment of clinical and genomic data has given rise to a wide variety of research works seeking to demonstrate its usefulness and benefit for this community. From the perspective of genomics, understanding the behavior of the human genome is a very difficult and complex task since there are a large number of elements that participate in the development of life. Through different works, the effectiveness of the application of the CM on this type of domains has been demonstrated.

Below are some of the most relevant works related to this research line:

- An interesting proposal on CM-oriented approaches was presented by Paton and Bornberg-Bauer (2000, 2002) modelling the genome from such a CM perspective. They tried to describe it from different perspectives: the description of the genome of the eukaryotic cell, the interaction between proteins and the transcriptome, among others [10,11].
- Ram et al. (2004) shows the application of CM principles in the field of proteins, in which large amounts of data with a fairly complex structure are consulted. The result of this work is to facilitate the development of user-friendly tools to search and compare proteins by their structure. This exercise shows that it is easier to search the structure of a protein in 3D through a solution where conceptual modeling is applied [14].
- Pastor et al. (2020) presents the application of CM techniques for the improvement in the management of genomic data, this work provides a solution that helps researchers to organize, store and process information focusing on the data that are relevant and minimizing the impact of information overload in clinical and research contexts [8,9].

The reported works constitute a relevant contribution regarding the application of CM techniques in health. Likewise, CM techniques have been used in other lines of research. Below, we report a list of interesting work where the use of CM techniques has been applied in other domains.

- García et al. (2020), together with the Valencian Institute of Agrarian Research (IVIA, http://www.ivia.gva.es/es), create a software platform to perform a comparative analysis exercise with hundreds of full citrus genome sequences [5]. This work is proof of how a conceptual model-based solution increases the value of the generated artifact. Another relevant result of this work is the Conceptual Schema of the Citrus Genome (CSCG), the ontological basis from which a solid knowledge is obtained in the genomic citrus domain.

– Reyes, in his PhD thesis (2018) [15], generated the Conceptual Schema o the Human Genome (CSHG), which facilitates the understanding of the human genome and its internals from a holistic perspective. His work faced the challenge of how to deal with the complexity of genomic data (heterogeneity, dispersion, isolation), and characterized relevant concepts of the genome.
– Burriel, in her PhD thesis (2017) [3], developed an Information System to manage breast cancer-related clinical and genomic data using conceptual models. Breast cancer is a disease with a huge impact in society, whose data and information are complex and heterogeneous.
– Arevshatyan et al. (2019) defined a conceptual schema to understand the Neuroblastoma disease [2]. Once again, the use of CM techniques became crucial to efficiently manage and analyze this domain's data.

Ophthalmology specialists face a great number of problems when managing the information of the treatments and follow-ups of their patients. This is due to the fact that there is a high dispersion of information, due to the use of files and documents in different formats (e.g., *.doc, *.xls, *.csv, etc.), that is, there is no structure well-defined. As seen above, the definition of a CM would help to bring together all the existing knowledge of the "retina-macula" domain, thus allowing specialists to address the problem they face with greater clarity and precision. This conceptual model, seen in more detail in Sect. 3, will give way to the development of an IS that initially serves as a support to the Ophthalmology Services of several hospitals in the Valencian Community (Hospital Vega Baja -Orihuela-, Hospital Lluís Alnayís -Xátiva-, and Hospital Virgen de los Lírios -Alcoy-). These medical services are faced with the need for powerful software that allows them to manage all the information related to their patients and their respective treatments. However, the scarcity of these types of tools makes the management, monitoring, and exploitation of data associated with these diseases more difficult.

Currently, there is only one program worldwide, called FRB or Project "Fight Retinal Blindness!"[2], whose license per hospital costs around EUR 30,000, this product has limited access and is not easily accessible. This program allows us to follow the progress of the patient (individual treatment for each affected eye) and generate graphs that help to obtain a better diagnosis.

3 Conceptual Model of the Macular Pathology (CMMP)

The use of CM in the life sciences domain (clinical, genomics, etc.) facilitates the understanding of all the participating concepts in this complex and continuously growing context. This work presents the design and development of an IS based on a conceptual model, which aims to be implanted in the ophthalmology services of different hospitals in the Valencian community. The result of this research work

[2] http://www.savesightinstitute.org.au/research-units/save-sight-registries/fight-retinal-blindness/.

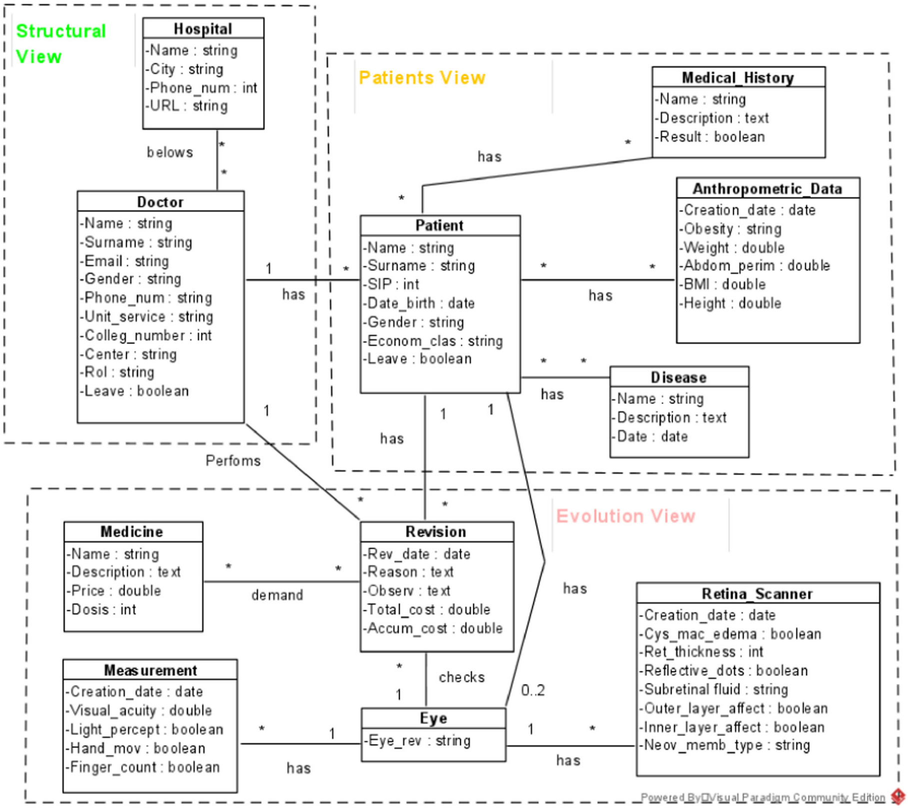


Fig. 1. Conceptual model of the macular pathology (CMMP).

is a Conceptual Model of the Macular Pathology (CMMP), which is represented by Fig. 1.

The CMMP version 1 is composed of three different views, grouping the entities which are related in some way. These views are structural, patients and evolution.

- **Structural View:** Stores the information associated with the doctors and hospitals in the system. This part is more focused on the management and control of human resources and system infrastructure.
- **Patients View:** Contains the patient information during the start of treatment, this view is composed of the following classes: "Patient", "Disease", "Anthropometric_Data" and "Medical_Anteced".
- **Evolution View:** It allows monitoring of the patient's evolution thanks to the definition of all the information related to their reviews and tests (previously carried out).

Before starting the modeling tasks, a series of brainstorming and discussion meetings were held with the group of experts in ophthalmology, in order to consider all the elements participating in this domain.

During the definition of the CMMP, the entire medical process of the Ophthalmology Service of the hospital centers has been considered. This conceptual model contemplates the process from when a patient arrives and the first consultation is made, until his last steps in the medical follow-up-treatment. This includes both the clinical management carried out by professionals in ophthalmology (for instance, hospitals, doctors, drugs, etc.), as well as the treatment of patient information.

In ophthalmology departments, in an initial phase, there is a first consultation with the patient. If the doctor considers it appropriate to start the treatment, they proceed to take the patient's data. This personal information contains the following attributes: name, date of birth, address, among others, and they are represented in the conceptual model through the class "Patient".

It is also convenient to know if the patient has a medical history, for example, if he has suffered some type of heart attack, if he is or has been a smoker, if he has diabetes, cholesterol, obesity, etc. Or if you have any type of disease that can affect the eye: MAE, EMD, or OVCR. All this information can be seen represented in the model through the classes "Anteced _Medical" and "Disease", respectively.

In addition, it is relevant to keep track of the patient's anthropometric data, such as weight, abdominal girth, height, and BMI, since they can significantly influence its evolution. Through the class "Anthropometric_data" we can manage a more efficient history, thus enhancing a better diagnosis.

Once all the information related to the patient has been collected, a series of reviews are carried out where the evolution of the patient is observed. The "Revision" class allows knowing the date on which the revisions took place, the reason for them, the observations made by the specialist, and the total cost. The total cost will allow quantifying the expenses in the event that treatments have been scheduled for the patient. This value is a derived attribute obtained from the sum of the prices of each of the drugs used in the treatments.

After having defined the conceptual model that gathers the knowledge of the Retina-Macula domain, we can talk about the design and implementation of the resulting SI, seen in Sect. 4, after having considered the CMMP during its construction.

4 Solution Design

This section presents the solution generated from the conceptual model defined in the previous section. This conceptual model has served as ontological support in the creation of the software solution, which has been developed with innovative technologies that have great support within the SE community. All this with the purpose of providing an IS that responds to the needs of clinical experts

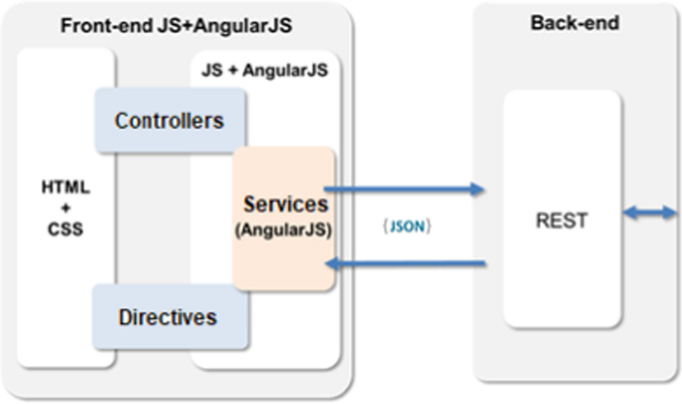


Fig. 2. Solution architecture.

in Ophthalmology Services (SO), and that also has an appropriate degree of maturity for its application in medical practice.

The first one is in charge of providing the services and the functionality of the application through the publication of a REST API, while the second one corresponds to the user interface, which allows communication with the server (backend) and consumes the services published by this API (see the solution architecture in Fig. 2 [6].

The design and development of the G-MAC information system was carried out from three (3) different perspectives, which are defined below:

– *Structural Design (data layer design):* this task includes the creation of the application data model, extracted from the CMMP. This requires a database server configuration process (DBMS), in this case, the MySQL6 relational database manager, and the creation of entities or models that allow storing all the information required by the IS. Previously, it has been necessary to define a series of models, extracted from the generated CMMP.
– *Functional Design:* this task focuses on the construction of an API (Application Programming Interface) in order to provide a service that allows two software to be connected together, facilitating the exchange of messages and information in a specific format. This API is part of the SI backend, and allows access to available resources, such as information in the database. The API designed is essential in the project since it allows to fulfill all the functionalities required by the stakeholders in the IS.
– *Interface Design:* this task focuses on the design of the application's user interface. For this, the "Angular" framework and its "Angular-CLI" command line interpreter (https://angular.io/cli) have been used, which facilitate a faster and more agile workflow. To achieve the functionality desired by the stakeholders, it has been necessary to define 25 components, which have been grouped into five (5) directories according to their function: i) Login: Contains the implementation of the login or access control to the tool. It is made up of a

single component, ii) Models: Stores the definition of each of the application's models or entities. A model represents an entity in the system and collects its information (attributes and types) for better representation. They represent the format of the data obtained from a request to the API, iii) Pages: Contains most of the views (pages) of the application and is made up of a total of nineteen (19) modules, iv) Services: Contains all the services and guards used in the components. A service is basically a data provider, which, by making an HTTP request to the API, obtains a set of data in JSON format, and v) Shared: Contains a series of common views when navigating the application's user interface.

It is important to highlight that the design and development of the software solution for this work is based on an MVC architecture[3] pattern. To see in detail the rest of the components of this architecture (e.g., *database layer*) can be consulted in ([6], Chapter 4, pp. 50–57). This proposal allows to increase the quality and integrity of the solution architecture. This work is part of an ongoing research project (TRL **4**, see Fig. 3) with examples of the web interface), on which the integration of all the improvements detected by clinical experts is maintained.

5 Validation

Wieringa defines in [16] the validation phase as the evaluation of the use of an artifact in a context in order to predict the impact of the artifact in the real world, used by stakeholders. In this section, we validate our proposed solution together with the experts in retina-macula from the Valencian system of health. Our validation process included three exercises (E):

- **(E1) Prototype Validation**: Conducting several meetings with software developers and domain experts.
- **(E2) Expert Opinion**: Reviewing the artifact under laboratory conditions together with domain experts to obtain their feedback.
- **(E3) Technical Action Research (TAR)**: to bridge the gap between the idealization of the initial designs and the limitations and threats of the real world. The TAR is being carried out by the time we wrote this paper.

In *prototype validation* (E1), two types of meetings were performed. In the first type of meetings, we met with software developers from the PROS Research Center[4] in order to validate the architecture and the developing process of the generated artifact. Also, these meetings allowed us to detect future improvements at the technological level. In the second type of meetings, we met with domain experts in order to validate the that the gathered requirements were implemented adequately.

[3] https://www.w3schools.in/mvc-architecture/.
[4] http://www.pros.webs.upv.es/.

Fig. 3. G-MAC user interface: A) User login for the SI, and B) Evolution of a patient's visual acuity.

In *expert opinion* (E2), we validated the functionality of the generated artifact. We used the User Interview technique, which allowed us to gather domain experts' feed in an agile way [13]. The expert opinion exercise was divided into two steps: i) *user observation* and ii) *user interrogation*.

1. In user observation, we asked users to perform a set of predefined tasks using G-MAC. These tasks were defined based on user requirements and the feedback obtained from the feedback gathered in the previous exercise (i.e., prototype validation). The experts that performed the tasks included medical staff from three hospitals: Vega Baja (Orihuela), Lluís Alnayís (Xátiva), and Virgen de los Lirios (Alcoy), under the coordination of Dr. Rubén Cabrera Beyrouti. Being the first validation of the functionality of G-MAC, this exercise was conducted in lab context so that we had greater control over this process.

The defined tasks included:

- *Register a patient*: The user has to register a new patient, show the list of existing patients, and search the new patient by two criteria: "SIP number" and "full name".
- *Register a set of revisions*: The user has to create a set of new revisions for the previously added patient. Also, the user has to list the revisions of this patient and consults the most recent one.
- *Consult the evolution of a patient*: The user has to consult the evolution of i) visual acuity and ii) the treatment cost of the patient that was registered in the first task.

2. In user interrogation, we interviewed users to know their opinion regarding the use of the application after performing the tasks defined above. The gathered feedback can be summarized in four (4) relevant points:
   - *The tools improves their daily work*: They stated that the use of the prototype allowed them to manage the information associated with the patients of the Ophthalmology Service efficiently. Also, it allowed them to achieve greater knowledge regarding the evolution of patients, and to provide an improved and more accurate service to patients.
   - *The tools is easy to use*: The tool is easy to use and intuitive. They were satisfied with the workflow of the tool and the information displayed. In line with that, we observed that users did not struggle to complete the defined tasks.
   - *It achieves stakeholders' goals*: The stakeholders expressed that they were satisfied with use of the prototype. They indicated that the data that is integrated, managed, and displayed will improve the management of the costs that are associated to the AMD treatment. In addition, they expect to be able to identify treatment patterns to increase the effectiveness of drug treatments so that patients recovery is improved.
   - *They identified potential improvements to make the more useful tool*: They provided us with minor visual improvements, which shows their interest in using the tool. The three improvements that they identified are the following:
     1. The "Result" section of the "Medical Background" management can be eliminated since this requirement is complete with the "name" and "description" sections.
     2. In the case of "Anthropometric Data", the system must allow the data to be stored, even if it is incomplete. This is because many times in the initial revision this data is not obtained, and the possibility of adding it to the system in the future should be considered.
     3. Provide an interface where you can view all the information associated with the revised eyes in two parts (i.e., one per right eye and one per left eye).

In general, stakeholders' reported a positive use of the prototype. The reason is that the provided tool allowed them to achieve their goals and improve AMD treatment in an easy and intuitive way. Also, they mentioned that adopting the tool takes a small amount of time. This is crucial in their working context (i.e., hospitals). In conclusion, it is an undeniable fact that integrating the data related to AMD treatment and automating patient monitoring produces greater satisfaction and benefits than performing it manually. The validation reported encouraging findings but the results should be understood under the conditions of the evaluation. Our next step in this line is to carry out more empirical evaluations (**E3**) that reinforce the results obtained.

6 Conclusions

The goal of this work has been to facilitate an IS that is based on conceptual models, which improved the analysis and management of the data of patients treated for retina-macula pathologies in the OS of the Conselleria de Sanitat. Two more specific goals were established to achieve this: i) the study and analysis of the domain, which allowed us to generate the CMMP, and ii) the development of a "G-MAC" prototype (based on the defined conceptual model), which allowed a efficient and effective management of the data of patients with macular diseases.

During the development of this work, it has been possible to improve the understanding of this complex domain (through the defined CMMP), and we acquired a high degree of knowledge from the activities carried out by the OS experts, which has allowed the integration of all the needs/demands of the clinical staff in the prototype generated. At a technical level, it is important to highlight the use of cutting-edge technologies for the design and development of the tool, in addition to the incorporation of advanced technologies for security management, which is a fundamental aspect in the context of health.

As the first iteration of this work, a conceptual model of the domain (CMMP) and a prototype in its first version (G-MAC) have been generated, this solution is being continuously validated (both at a technical and functional level) by clinical experts. Future work includes:

- the *evolution and extension of the CMMP*, which would focus on the detection of new knowledge (e.g., new tests, methodologies, among others.) in the domain for its integration into the conceptual model with the in order to provide a better diagnosis or treatment to the patient;
- the *Generation of the second version of the software solution provided*, for this, all the feedback received after completing the prototype validation phase will be considered. In addition, to integrate the changes added in the CMMP (as an ontological basis), and finally;
- the *Transfer of the software product*, once its usefulness in healthcare practice has been demonstrated (In the three hospitals mentioned above), it would be installed in the rest of the Health departments of the Consellería de Sanitat. In the future, each hospital is expected to have its own user license.

Acknowledgements. The authors would like to thank the members of the PROS Research Center Genome group for the fruitful discussions regarding the application of CM in the medical field. In addition, we would like to thank Dr. Rubén Cabrera Beyrouti as an expert in Retina-Macula for their contribution to this research. This work was supported by the Spanish Ministry of Science and Innovation through Project DataME (ref: TIN2016-80811-P), the Preparatory Action - UPV-FISABIO (A36-G-MAC, 2019), and the Generalitat Valenciana through project GISPRO (PROMETEO/2018/176).

References

1. Aguilera, D., Gómez, C., Olivé, A.: Enforcement of conceptual schema quality issues in current integrated development environments. In: Salinesi, C., Norrie, M.C., Pastor, Ó. (eds.) CAiSE 2013. LNCS, vol. 7908, pp. 626–640. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38709-8_40
2. Arevshatyan, S., Reyes Román, J.F., Burriel, V., Cañete, A., Castel, V., Pastor, O.: Integration and analysis of clinical and genomic data of neuroblastoma applying conceptual modeling (2019)
3. Burriel, V.: Tesis doctoral: Diseño y desarrollo de un sistema de información para la gestión de información sobre cáncer de mama sobre cáncer de mama, Universitat Politècnica de València (2017). https://doi.org/10.4995/Thesis/10251/86158
4. Delcambre, L.M.L., Liddle, S.W., Pastor, O., Storey, V.C.: A reference framework for conceptual modeling. In: Trujillo, J.C., et al. (eds.) ER 2018. LNCS, vol. 11157, pp. 27–42. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00847-5_4
5. García, A., Reyes Román, J.F., Casamayor, J.C., Pastor, O.: Towards an effective and efficient management of genome data: An information systems engineering perspective (2019)
6. Palomares, M.A.: Desarrollo Dirigido por Modelos para el Análisis y Gestión de los Datos asociados a la Patología Macular. Master's thesis, Universitat Politècnica de València (2020). https://hdl.handle.net/10251/151886
7. Olivé, A.: Conceptual Modeling of Information Systems. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-39390-0
8. Pastor, O., León Palacio, A., Reyes Román, J.F., Casamayor, J.C.: Using conceptual modeling to improve genome data management. Brief. Bioinform. **22**, 45–54 (2020). https://doi.org/10.1093/bib/bbaa100
9. Pastor, O., León Palacio, A., Reyes Román, J.F., Casamayor, J.C.: Towards the understanding of the human genome: a holistic conceptual modeling approach. IEEE Access **8**, 197111 (2020). https://doi.org/10.1109/ACCESS.2020.3034793
10. Paton, N.: Conceptual modelling of genomic information. Bioinformatics **16**, 548–557 (2000). https://doi.org/10.1093/bioinformatics/16.6.548
11. Paton, N., Bornberg-Bauer, E.: Conceptual data modelling for bioinformatics. Brief. Bioinform. **3**(2), 166–180 (2002)
12. Plazzotta, F., Luna, D., Bernaldo, F.G.: Sistemas de información en salud: Integrando datos clínicos en diferentes escenarios y usuarios **32**, 343–351 (2015). http://www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S1726-46342015000200020&nrm=iso
13. Preece, J., Rogers, Y., Sharp, H.: Interaction Design: Beyond Human-Computer Interaction, 4th edn. John Wiley & Sons Inc., New York (2015)

14. Ram, S., Wei, W.: Modeling the semantics of 3D protein structures. In: Atzeni, P., Chu, W., Lu, H., Zhou, S., Ling, T.-W. (eds.) ER 2004. LNCS, vol. 3288, pp. 696–708. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30464-7_52
15. Reyes Román, J.F.: Diseño y Desarrollo de un Sistema de Información Genómica basado en un Modelo Conceptual Holístico del Genoma Humano (doctoral dissertation), Universitat Politècnica de València (2018). https://doi.org/10.4995/Thesis/10251/99565
16. Wieringa, R.J.: Empirical research methods for technology validation: scaling up to practice. J. Syst. Softw. **95**, 19–31 (2014). https://doi.org/10.1016/j.jss.2013.11.1097

API Management Maturity of Low-Code Development Platforms

Michiel Overeem[1(✉)], Slinger Jansen[2,3], and Max Mathijssen[2]

[1] AFAS Software, Inspiratielaan 1, Leusden, The Netherlands
michiel.overeem@afas.nl

[2] Utrecht University, Princetonplein 5, Utrecht, The Netherlands
{slinger.jansen,m.mathijssen}@uu.nl

[3] School of Engineering Science, LUT University, Lappeenranta, Finland

Abstract. Low-code development platforms are environments that enable *citizen developers* without software engineering knowledge to create software products. These software products range from small business applications to large business platforms, around which software ecosystems increasingly form. In these software ecosystems, different organizations want to extend the created software products with services and software, with the goal of creating active enterprise networks that create value collaboratively. Well designed and maintained application programming interfaces are crucial for these organizations.

In this paper we evaluate the application programming interface management maturity of four low-code development platforms. We show that these platform providers are not yet concerned with helping their customers build software ecosystems around the software platforms that citizen developers create. Furthermore, we identify the *software engineering research challenges* that these platform providers face. For instance, low-code development platforms should create abstractions that let *citizen developers* design, develop, and manage application programming interfaces. If low-code development platform providers follow our advice and act on it, they will become able to provide customers with complete ecosystem-enabled platforms instead of providing only simple throwaway business applications.

Keywords: Low-code · Model-driven development · API management · Citizen software development

1 Introduction

Increasingly, traditional Software Producing Organizations, i.e., organizations whose main activities include the production of software such as software vendors and open source organizations, are discovering platforms as a vehicle to

This research was supported by the NWO AMUSE project (628.006.001): a collaboration between Vrije Universiteit Amsterdam, Utrecht University, and AFAS Software in the Netherlands. See amuse-project.org for more information.

A. Augusto et al. (Eds.): BPMDS 2021/EMMSAD 2021, LNBIP 421, pp. 380–394, 2021.
https://doi.org/10.1007/978-3-030-79186-5_25

increase the value of their software for their customers through collaboration with third parties. This transformation from product towards a platform is called 'platformisation' [14]. Platforms are a vehicle for software ecosystems and are defined as a set of organizations collaboratively serving a market for software and services [9]. These ecosystems form around software platforms, which in turn are managed by software platform orchestrators.

We find that not all software products can easily transform into software platforms. One particular category of software products are products that are created using no-code/low-code development platforms (LCDPs). LCDPs apply model-driven development to raise the abstraction level of software development, increasing productivity and decreasing complexity as a result [3]. They target a wide variety of users, from professional software developers to non-technical business experts [13]. The latter category is commonly referred to as the 'citizen developer': people without software development education who nonetheless build software applications. Customers of these LCDPs utilize these platforms to increase their agility, it enables them to develop business applications more efficiently without suffering from the lack of professionally trained software developers. While LCDPs are traditionally used to create agile business software applications, they are increasingly becoming part of the core IT landscape [17]. As long as the LCDPs prove their value, customer companies will utilize them in more and more diverse projects. The customer companies will be interested in evolving their application into a platform to enable complementors, which are organizations that build application that extend the core system, to create more value [9].

Evolving into an ecosystem is done by offering parts of the developed application through Application Programming Interfaces (APIs). Research shows that concerns such as stability, security, and scalability in ecosystems are related to API management capabilities [1]. API management is the activity that enables organizations to design, publish, and deploy their APIs for (external) developers to consume. This is one of the enabling practices for the creation of an ecosystem. Traditionally the activities within API management are executed by technical staff members. The strength of LCDPs, however, is that activities that used to be executed by highly technical staff, such as software engineers, are now executed by citizen developers. To support these API management activities LCDPs have to provide the means to the citizen developers to integrate applications developed on the LCDP with other applications.

We believe that LCDP providers must mature their API management capabilities to remain relevant for citizen developers. After all, software ecosystems are increasingly seen as the way to remain strategically relevant in the software industry [8]. In this paper, we distinguish between two software ecosystems that form around LCDPs. First, the LCDP ecosystem is the ecosystem that forms around the LCDP and the provider of the platform. The second is the LCDP Application Ecosystem, which forms around the application that is created by one of the customers of the LCDP. Our contribution is an evaluation of the maturity of API management capabilities support among LCDPs, along with a set of challenges that we identify. We evaluate the capabilities of four LCDPs through descriptive case studies. In these case studies we measure the maturity of their API management capabilities through a Focus Area Maturity Model (FAMM). A FAMM is a model that groups capabilities and practices in focus areas and aligns them along maturity levels, which can be used to evaluate organizational practices around particular focus areas [2,8].

Section 2 describes our research method and gives a short description of the four LCDP providers investigated in our evaluating case studies. The API management FAMM (API-m-FAMM) is described in Sect. 3. In Sect. 4 we evaluate the LCDPs to uncover their level of support for a citizen developer in API management activities. The results of the case studies are analyzed in Sect. 5. We contribute a set of engineering research challenges for software engineering researchers in Sect. 6. Threats to validity are discussed in Sect. 7. In Sect. 8 we observe, through the four case studies, that some LCDPs are slowly maturing their API management capabilities while others do not act on these opportunities. Furthermore, we observe that the API-m-FAMM, while mostly aimed at organizations with a proprietary API infrastructure, is also applicable to LCDPs. Finally, we conclude that these platform providers need to increase the level of abstraction of the technical complexity of managing APIs, without losing the strategic strengths of it.

2 Research Method

Our research focuses on the question: *How mature are the API management capabilities that LCDPs offer?* We use an established framework for evaluation of API management maturity, to evaluate how well applications, created with four LCDPs, enable API capabilities for the LCDP customer.

In our research we apply a FAMM to measure the maturity of API management in LCDPs. The API-m-FAMM [11] captures API management in 81 practices, grouped in 20 capabilities, which are in turn assigned to six focus areas. The model is intended for organizations that develop their own proprietary API infrastructure. However, in our case studies we evaluate the LCDPs, to uncover how they support the API management activities of their customers. We apply a maturity model to measure the current state and provide a roadmap to advance this state, similar to [5]. Please note that more details are provided on the API-m-FAMM in the next Section.

The evaluations are performed in four descriptive case studies, which were conducted with the ACM SIG Empirical Research standard in mind [16]. To be able to extrapolate our findings we selected platforms that represent the current state of LCDPs. Our selection was made based on the Quadrant for Enterprise LCDPs of the advisory company Gartner [19]. This report surveys 18 platforms and categorizes them into leaders, visionaries, challengers, and niche players. We selected two leaders and two visionaries that were willing to cooperate in our evaluation. From Gartner's report we can conclude that leaders and visionaries will show the state of the art in Enterprise LCDPs and represent the most advanced platforms.

The case studies were conducted with the following steps. First, public sources such as product documentation and company blogs were studied to create a first impression of the API management capabilities of the LCDP. Next, an interview with a company expert (either a product manager, architect, or chief technology officer) was conducted. During the interview the API-m-FAMM [11] was described to the interviewee, including all of the practices and their terminology. Together with the interviewee the API-m-FAMM was used to asses the LCDP. Finally, the interviewees discussed their LCDP with respect to API management, and their opinion on creating platforms on top of the LCDP. Subsequently the interviews were processed and analyzed. Based on this analysis and together with the company documentation the evaluation of the LCDP using the API-m-FAMM was completed. Afterwards the evaluation was shared with the interviewee to correct mistakes and oversights. Finally, our general findings were discussed with the interviewees, to establish how they perceive the role of APIs in their generated applications and whether they equally acknowledge the trend of 'ecosystemification'.

We shortly describe the case study organizations here. *Mendix* is a worldwide operating low-code platform provider, founded in the early 2000s. The company employs 1,000 employees, serving thousands of customer companies and an ecosystem of almost 150,000 developers. *OutSystems*, as the second biggest and oldest provider, has been active for almost 20 years. With well over 1,000 employees worldwide, they serve a large range of companies. The LCDP originated as a rapid application development platform. *Betty Blocks*, founded almost 10 years ago, employs around 200 people. Operating worldwide, they serve customers in all business domains. This LCDP has a strong focus on the citizen developer in enterprises, as reflected in their vision: 'anyone should be able to build an application.' *Pega* is the oldest (40 years) and biggest (6,000 employees) company of the four. Its LCDP evolved from a business process modelling suite.

3 Introduction of the API-m-FAMM

The goal of the API-m-FAMM[1] is to support organizations that expose their APIs to third-party developers in their API management activities. Using the API-m-FAMM, organizations may evaluate, improve upon and assess the degree of maturity their API management processes have.

A focus area maturity model [18] consists of focus areas, and an area consists of capabilities, which are defined as *the ability to achieve a certain goal related to API management, through the execution of two or more interrelated practices.* A practice in turn is defined as *an activity that has the express goal to improve, encourage and manage the usage of APIs.* The API management maturity model is created following the steps described by [18] and [2]. The scope, design, and populate phase are based on a systematic literature review [10]. The model was further refined through two rounds of interviews with experts: eleven interviews

[1] A detailed description and the source data are published [11]. The model is also available on the https://MaturityModels.org web site.

in the first round and three interviews in the second round. Finally the model was used to asses five different software products.

The API-m-FAMM consists of six focus areas that we briefly summarize here:

- **Lifecycle Management**: An API undergoes several stages over the course of its lifetime [12]. Version management is particularly challenging: complementors in the ecosystem benefit from stable APIs, but at the same time demand new functionality to further their own product.
- **Security**: APIs provide access to valuable and protected data and assets. Therefore, mature APIs implement the latest security standards, such as the *OAuth 2.0* authorization protocol, and protection against threats such as Denial of Service attacks.
- **Performance**: APIs deliver data and services to complementors in the ecosystem. This increases the demand on APIs to perform well under load: the application itself as well as the complementors are negatively effected by a decrease in performance.
- **Observability**: An organization benefits from insight into the API's usage. Through various monitoring techniques, the organization is able to collect metrics which can shed light on the API's health and performance, as well as its usage by complementors. A performant and healthy API is crucial, because an interrupted service of the APIs will also most likely interrupt the complementors application.
- **Community**: It is desirable for organizations to foster, engage, and support the community that exists around the API. This entails offering developers the ability to register for API access and offering them access to test environments, code samples, and documentation.
- **Commercial**: Exposing and consuming APIs can have a commercial aspect tied to it [4]. On one hand, APIs can require a subscription fee from the complementors, on the other hand complementors might demand Service Level Agreements from the provider.

These focus areas are composed of 20 capabilities, which in turn comprise 81 practices. Within their corresponding capabilities, which may be regarded as sub-topics, practices are ranked based on the perceived complexity of their implementation. In order to verify whether an organization has implemented a practice, a set of conditions for implementation has been defined for each practice. By examining the fulfillment of the aforementioned implementation conditions, it may be determined whether an organization has implemented a practice. When this is done for each practice a capability consists of, an organization's *maturity level* for that capability may be determined.

We provide a description of the practice *Implement Multiple API Versioning Strategy* here, to clarify how the practices are evaluated. The description of this practice is *"The organization has a versioning strategy in place which entails the process of versioning from one API to a newer version. In order to do so, the organization must be able to maintain multiple versions of (one of) their API(s) for a period of time. Possible strategies include URI/URL Versioning (possibly* *in combination with adherence to the Semantic Versioning specification), Query Parameter versioning, (Custom) Header versioning, Accept Header versioning or Content Negotiation."* Each practice has an *Implemented when* text, that describes one or more conditions to evaluate whether a practice has been implemented or not. In this case the condition is self-explanatory: "The organization utilizes one of the following versioning strategies: URI/URL Versioning, Query Parameter versioning, (Custom) Header versioning, Accept Header versioning or Content Negotiation.". For the LCDPs, we discussed whether it is possible to maintain different versions of the API, or whether an API always co-evolves with the model and does not have any kind of evolution mechanisms implemented to enable different API versions.

4 Case Studies

This Section describes the four evaluations of the LCDPs that were done with the API-m-FAMM. This assessment was done based on the available platform documentation and the interview. The interviewees were able to point out mistakes in the evaluation, comments were incorporated accordingly. First we describe the LCDPs in general, then we discuss the six focus areas and how the LCDPs support these.

Mendix - The road map of *Mendix* shows a focus towards enabling citizen developers to create increasingly complex applications, which is motivated by two developments. First of all, applications developed on the LCDP are growing and becoming increasingly complex. Second, an increase in demand from citizen developers to build integrated applications independent from professional developers is observed. Strong API management capabilities enable customers to split their large applications into smaller integrated applications. The envisioned central API catalog will bring together applications within an enterprise, enabling citizen developers to develop integrated solutions.

OutSystems - The focus of *OutSystems* is ensuring that the co-development between the citizen developer and the professional developer is made as efficient as possible. Their vision is 'fast and agile development of enterprise applications'. API management is not hidden behind abstractions, but rather placed in the hands of professional developers. The gap between technical API management and the citizen developers is not actively bridged, instead the co-development between citizen developers and professional developers is promoted.

Betty Blocks - This LCDP is focused on consuming APIs, instead of publishing them. *Betty Blocks* states that their LCDP is not used to develop core systems, but to develop supporting applications. Applications mostly complement existing systems. The runtime of the LCDP consists of a web-based server and browser-based client application. Developers do not have to explicitly design APIs, as every application feature is an API by default through this architecture.

Pega - Ease of change and rapid application development by collaborating departments in enterprises are the focus of *Pega*. The developer tool of the LCDP supports multiple personas, both the citizen developer as well as the professional developer. *Pega* supports a myriad of integration options, among REST or SOAP APIs it also supports integration through database connection or e-mail. Despite plentiful of options to integrate with other applications, *Pega* does not observe a large portion of their customers using these capabilities to integrate with complementors outside of the organization.

Through the API-m-FAMM evaluation we measure the state of API management support that the LCDPs offer. Considering that the API-m-FAMM is targeted towards organizations that expose their APIs to third-party developers, the evaluation of LCDPs differs from this original intention. A LCDP is both an application (run-time) platform and a development platform. API management practices can be implemented in different ways in a LCDP:

- A practice can be **statically implemented** by the LCDP, meaning that the customer cannot influence it (an example is *Load balancing*).
- **Variable** implemented practices are those practices that can be influenced by citizen developers or professional developers, such as *Multiple API Versions Strategy*.
- Some practices can only be implemented by using products from **third-party** vendors. Example is the *Adopt Subscription-Based Monetization Model* practice.
- The LCDP is a development environment and in that capacity the LCDP can be used to implement a number of API management practices. Examples of these **build-your-own** practices are *Community Forum* and *Broadcast API Status*.

The last two categories, third-party and build-your-own, result in more work for the customers of the LCDP. They become responsible for developing and maintaining these specific practices, while statically and variable implemented practices do not have these liabilities. Therefore, we evaluate the four LCDPs by scoring the practices in two categories: *supported* (by the LCDP) and *custom* (developed) practices. The first category consists of all statically and variable implemented practices, third-party and build-your-own practices are grouped in the second category. Table 1 shows the results per API-m-FAMM capability[2]. Every score consists of two numbers: first the number of practices that are supported by the LCDP, then the number of practices that need to be custom implemented.

Focus Area: Lifecycle Management - Generally speaking, the LCDPs support both the consumption and publication of modern APIs. Standard protocols such as SOAP and REST are supported by all LCDPs. *Mendix*, *OutSystems*, and *Pega* also support the API protocol OData. The decision of *Betty Blocks* to create APIs automatically shows a strong opinion on APIs. Some practices are

[2] The detailed evaluations are available through http://dx.doi.org/10.17632/wdtg5ytdpf.1.

Table 1. Evaluation of the API management maturity of the four LCDPs according to the API-m-FAMM: *Mendix* (M), *OutSystems* (O), *Betty Blocks* (B) and *Pega* (P). For every API-m-FAMM capability we show the total number of practices, and the LCDP evaluation. The two numbers per LCDP stand for practices *supported* by the LCDP and practices that need to be *custom* developed respectively.

	Focus area	M	O	B	P
1	**Lifecycle management**	**7/5**	**8/4**	**6/4**	**8/4**
1.1	Version Management (*4 practices*)	2/2	3/1	2/1	3/1
1.2	Decoupling API & Application (*4 practices*)	4/0	4/0	3/0	4/0
1.3	Update notification (*4 practices*)	1/3	1/3	1/3	1/3
2	**Security**	**12/4**	**12/4**	**10/4**	**12/4**
2.1	Authentication (*3 practices*)	3/0	3/0	2/0	3/0
2.2	Authorization (*4 practices*)	4/0	4/0	4/0	4/0
2.3	Threat detection & Protection (*6 practices*)	3/3	3/3	2/3	3/3
2.4	Encryption (*3 practices*)	2/1	2/1	2/1	2/1
3	**Performance**	**6/5**	**6/5**	**8/2**	**7/4**
3.1	Resource Management (*4 practices*)	3/1	3/1	3/1	3/1
3.2	Traffic Management (*7 practices*)	3/4	3/4	5/1	4/3
4	**Observability**	**5/7**	**5/7**	**5/7**	**5/7**
4.1	Monitoring (*3 practices*)	0/3	0/3	0/3	0/3
4.2	Logging (*4 practices*)	4/0	4/0	4/0	4/0
4.3	Analytics (*5 practices*)	1/4	1/4	1/4	1/4
5	**Community**	**10/8**	**9/9**	**10/8**	**8/10**
5.1	Developer onboarding (*4 practices*)	4/0	4/0	4/0	4/0
5.2	Support (*3 practices*)	1/2	1/2	1/2	0/3
5.3	Documentation (*3 practices*)	2/1	2/1	2/1	2/1
5.4	Community engagement (*5 practices*)	0/5	0/5	0/5	0/5
5.5	Portfolio management (*3 practices*)	3/0	2/1	3/0	2/1
6	**Commercial**	**2/10**	**2/10**	**2/10**	**2/10**
6.1	Service-Level agreements (*4 practices*)	2/2	2/2	2/2	2/2
6.2	Monetization strategy (*4 practices*)	0/4	0/4	0/4	0/4
6.3	Account management (*4 practices*)	0/4	0/4	0/4	0/4
	Total	**42/39**	**42/39**	**41/35**	**42/39**

not implemented, because their choice for GraphQL based APIs enforces APIs without versions. Their LCDP customers cannot implement a versioning strategy, considering that API consumers always use the latest version, and customers need to take care of backwards compatibility. The other capabilities *Decoupling API & Application* and *Update Notification* show great resemblance between the four LCDPs. In the first capability all practices are supported by the LCDPs, while customers are expected to *custom* implement most practices in the second capability.

Focus Area: Security - In the area Security there is almost no differentiation between the LCDPs. All of the LCDPs follow modern security standards such as *Implement Transport Layer Encryption*, *Implement Authentication Protocol*, and *Implement Access Protocol*. The platforms support their customers in most practices.

Only the capability *Threat Detection & Protection* has a number of advanced practices that need to be implemented by the LCDP customers: *Security Breach Protocol*, *Conduct Security Review*, and *Implement Zero Trust Network Access*. While the providers have implemented these practices for their own hosted services, customers are responsible for their own protocols and are thus required to implement these practices as well.

Focus Area: Performance - Again the LCDPs are similar in their support of the practices in this area. In the *Resource Management* capability we observe that the LCDPs implement most of the practices. *Load Balancing*, *Scaling*, and *Failover* are all supported by the providers. Advanced practices in *Traffic Management* are mostly left to be *custom* implemented by the LCDP customers. The LCDP customers are required to configure third-party applications that implement practices such as *Manage Quota* and *Prioritize Traffic*. Implementing these practices requires expertise of professional developers and thus extra investment from the customers.

Focus Area: Observability - In this area there is no difference between the LCDPs. The practices in the capability *Logging* are supported by all four platforms. Monitoring the health, performance and resource consumption of APIs is left to the customers. *Custom Analysis Reports*, *Status Broadcasting*, and *Alerts* are also left to be custom implemented.

Focus Area: Community - Practices from the area *Community* that focus on technical capabilities, such as *Software Development Kit Support* and *API Catalog* are supported by the LCDPs. All of the LCDPs implement the API specification language OpenAPI, which supports practices such as *Use Standard for Reference Documentation* and *Provide SDK Support*. Less technical practices, such as *Social Media Presence* and *Communication Channel* are left to the customers to implement. Some of these practices, such as *Community Form*, can be built on top of the LCDP.

Focus Area: Commercial - The area *Commercial* is underdeveloped in all four LCDPs. Developers are not able to monetize the APIs developed on top of the LCDP, neither are they able to construct custom Service Level Agreements towards their API consumers. In order for customers to implement these practices they are required to integrate with third-party API management solutions.

5 Analysis of the Results

The four LCDPs under study show a great resemblance when evaluating their API management maturity. The fact that they are all either leaders or visionaries in the Quadrant for Enterprise LCDPs makes this no surprise, considering that they are ranked similarly. However, what is surprising is the general lack of support for advanced API management practices.

Mendix supports 42 practices, leaving 39 practices to be implemented by their consumers, making it an almost 50-50 split. The roadmap, as discussed during the interview, shows a focus on supporting their customers in the API management activities. This support is aimed at making it easier for citizen developers to build and publish APIs of higher quality. This roadmap has a focus on the practices in the *Community* practice. The area *Commercial* is not on the roadmap, making it harder for customers to monetize their APIs.

OutSystems supports mostly the same practices as *Mendix*, but made it clear during the interview that their ambitions differ. They have a strong focus on creating a platform that can enable the development of core enterprise systems by teams consisting of both citizen and professional developers. In this vision, there is no need to support all practices, because the provider recognizes that their customers already have several enterprise API platform solutions in place. Therefore, although there is a mature platform, many of the API management practices are left to their customer to implement themselves.

Within *Betty Blocks* (supporting 41 practices), publishing APIs is possible, but the capabilities are not mature enough to build a platform. In agreement with their vision, the LCDP can be used to complement other applications, but is less usable to create core systems. This is caused by two main reasons. First of all, their opinionated implementation of API versioning through GraphQL limits customers in how they want to expose APIs to their complementors. Second, *Betty Blocks* focuses less on the implementation of practices with third-party vendors. This makes it hard to implement practices such as *Prioritize Traffic*.

Pega (supporting 42 practices) is focused on letting their consumers build richer applications with their platform. These richer applications require integration with other applications. However, the focus of *Pega* is on in-house company projects that integrate within the company, or are complemented by selected organizations and partners. Companies are not supported in building an open platform that attracts complementors: capabilities for advanced community engagement or monetization strategies are not supported.

Overall the focus of the LCDPs appears to be on building enterprise applications, and less on platforms or even ecosystems. All LCDPs show that they have developed mature platforms, with support for modern standards in security and resource management. Through their implemented practices they enable their customers to develop and publish modern APIs that can be consumed by complementors. However, looking at the areas *Community* and *Commercial*, which contain less technical practices, we observe a gap. Many of the more advanced capabilities that customers can use to build platforms and attract complementors, such as *Monitoring*, *Analytics*, *Community Engagement*, and *Monetization*

Strategy, are left to their customers to implement. The LCDPs support around 50% of the practices, leaving the other 50% to be implemented by their customers. In the evaluation of the API-m-FAMM with non-LCDP ecosystems we encountered five products that implemented respectively 42%, 59%, 42%, 77%, and 79% of these practices. Three out of five of these products are more mature than support offered by the LCDPs, meaning that customers would have to implement a number of practices themselves to built comparable products with one of the LCDPs.

Not all providers agree with our belief that they should support API management activities to enable their customers to create platforms. By not implementing these practices, and leaving them to be custom implemented, their customers have to invest more effort in building a platform on top of their LCDP. The providers miss the opportunity to support better API management for citizen developers. Instead they obligate citizen developers to seek help from professional developers to complete these tasks. This creates a dependency from citizen developers on these professional developers, and misses the opportunity to put more power in the hands of the citizen developers, democratizing software development even further. As claimed in the Gartner report [19] LCDPs improve the productivity and reduce the time to market, and because of the shortage of developers, democratizing development would offer a possible solution. However, the current state of these LCDPs does not enable citizen developers to create platforms, without requiring the support of professional developers. Raising the abstraction of API management practices could and should be the next step for the LCDPs.

6 Engineering Research Challenges for LCDPs

The previous section discussed the current state of API management support among LCDPs, measured with the API-m-FAMM. Even though we provide LCDP providers with engineering and product planning direction through this evaluation with the API-m-FAMM, there are still several research challenges that hamper further progress in this domain. These challenges are based on our observations made during the case studies and on the authors' experience with software ecosystems and LCDPs. They are based on the capabilities that show the highest number of practices that require a *custom* implementation, and common remarks extracted from the interviews. We outline these engineering research challenges here and provide several solution directions.

Life Cycle Management - Citizen developers will be constructing new application extensions and releasing them to customers, probably without regard for software and data complementors who use a previous version of the application. Citizen developers need to be made more aware of the effects of data model and interface changes for the software ecosystem surrounding the application. While typically these problems would be solved through abstraction, it is practically impossible for citizen developers to remain ignorant of the effects of software evolution on interfaces with third parties in the ecosystem. This can be accomplished through, often complementary, practices such as versioning policies, backwards compatibility, publishing road maps, and change notifications [4,6,12].

The LCDPs support impact analysis within the platform, knowing the relations between different components. However, novel solutions to support analysis of impact on applications outside of the LCDP are necessary to support the citizen developer.

Performance - Considering that the created applications will be approached through different channels than the traditional user interface, novel architectures are required that can handle large volumes of traffic through other channels, such as APIs. Architecture styles such as Microservices [7] offer a possible solution to these scalability challenges. Of course, an important requirement is the abstraction that citizen developers should be offered.

Observability - LCDPs should be able to handle an increase in users, while still providing the citizen developer with control over who uses the API, how much the API is used, and how the API is used. API gateways [4] traditionally provide these controls to professional developers and operational staff, but now need to be supported by the LCDP itself and usable by non-technical users. The citizen developers need to have access to API usage metrics and statistics to ensure that they too can identify misuse and monitor traffic from the citizen developer's partners [20].

Community - As the community around a product starts growing, complementors need to be supported as much as possible. Such capabilities are for example enabling citizen developers to generate API access credentials for complementors, infrastructures for communicating with complementors, as well as providing application stores around a generated product. All studied LCDPs leave the development of these community practices to the citizen developers.

The abstractions provided by LCDPs should give citizen developers enough control over API documentation and usage, while automatically adding technical documentation such as SDKs and source code examples. The studied LCDPs offered this through the use of standardized specification languages, such as the OpenAPI specification.

In the past, research has been conducted on the generation of APIs [15]. However, a number of related practices, such as *Provide FAQ and Code Samples* and *Provide Start-up Documentation*, are only offered through consumer built solutions. These practices are challenging to support due to an ever evolving generated object model. Without support from the LCDP the developer is responsible for evolving the manual written documentation together with the model of the API. This will lead to mistakes that hurt the community.

While the term 'citizen developer' indicates that it has been the goal to open up software engineering to people without formal software engineering education, we can hardly claim that this has been successfully accomplished for API management. The complexity of modern software solutions and the inherent simplification required to create LCDPs are constantly in direct conflict with each other. The platformisation trend lays this bare and shows that new models and perspectives are required to truly make software engineering accessible to any citizen developer. We see it as future work to design new abstractions that make LCDP solutions simpler and more powerful in supporting API management practices.

7 Threats to Validity

In this paper we present four descriptive case studies that we conducted based on interviews and documentation. Through these case studies we evaluate the current state of API management support offered by LCDPs.

Our conclusions are threatened by concerns regarding the generalizability of these four LCPDs when compared to the LCDP industry as a whole with respect to API management maturity. We cannot deny that there could be an LCDP that we did not study that supports more, or even all, API management practices. However, given that we studied four major platforms that are recognized as such in the Quadrant for Enterprise LCDPs report [19] confirms that we have studied a representable group. While the API maturity evaluation might not be generalizable to other LCDPs, these four platforms are recognized as the most innovative in the industry. Given that there might be a provider that has a more mature support of API management practices only confirms that these leaders and visionaries are missing out on opportunities to further support their customers. Providers with less mature support of API management practices make our call to action only more pressing.

Another threat to validity of this research are the evaluations of the LCDPs based on the API-m-FAMM. Wrong or imprecise evaluations based on documentation and interpretation could distort the conclusions. The fact that the interviewees reviewed and corrected the evaluation mitigates this risk. We believe that the general evaluation of the LCDPs with respect to API management, combined with the vision of the LCDP provider gives a truthful representation of the current state of API management maturity. Our findings and conclusions are based on the global state of API management support of the LCDPs, and do not depend on specific practice support.

In our research we focused on the LCDPs and their API management capabilities. Although we discussed a number of organizations that built an internal platform on top of the LCDP, we did not discuss specific example platforms. We did not specifically search for an example, but rather focused on the general state of API management maturity. Future work should study existing platforms built on LCDPs to further understand what opportunities LCDPs have.

8 Conclusion

Our case studies, as presented in Sect. 4, evaluate four LCDPs using the maturity model API-m-FAMM. Our research was guided by the research question: *How mature are the API management capabilities that LCDPs offer?* We conclude that these LCDPs support around 50% of the practices described in the API-m-FAMM. The other practices are left to be implemented by the customers of the LCDPs. We conclude that only *Mendix* places API management firmly on its road map. Both *Betty Blocks* and *Pega* do not observe a demand for API management capabilities among their customers, and neither are they promoting these capabilities. *OutSystems* recognized the demand, but has not yet focused on providing more of these capabilities to their customers. Instead they defer much of the work to either third-party vendors or the LCDP customers. By not supporting these practices we believe that LCDP providers miss out on the opportunity to further democratize software development. They instead require citizen developers to solicitate the support of professional developers to develop platforms that are open for other companies to extend.

We draw the following conclusions from this work. First, we suspect that LCDP providers will soon be challenged in providing capabilities that enable citizen developers to transform their applications into platforms. Our research shows that LCDP providers are currently unable to support such capabilities for citizen developers and require technical staff to implement such architectures and mechanisms through either third-party solutions or custom solutions built on top of the LCDP. Second, we conclude that as LCDPs are becoming more powerful, they can use the API-m-FAMM to evaluate and update their road maps. Finally, we identify five engineering challenges that, if solved, will create a next generation of citizen developers who can independently create complete software platforms and software ecosystems, and subsequently manage them without the requirement for highly specialized technical knowledge.

References

1. Andreo, S., Bosch, J.: API management challenges in ecosystems. In: Hyrynsalmi, S., Suoranta, M., Nguyen-Duc, A., Tyrväinen, P., Abrahamsson, P. (eds.) ICSOB 2019. LNBIP, vol. 370, pp. 86–93. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33742-1_8
2. de Bruin, T., Rosemann, M., Freeze, R., Kulkarni, U.: Understanding the main phases of developing a maturity assessment model. In: ACIS 2005 Proceedings - 16th Australasian Conference on Information Systems (2005)
3. Cabot, J.: Positioning of the low-code movement within the field of model-driven engineering. In: Proceedings - 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MODELS-C 2020 - Companion Proceedings, pp. 535–537 (2020). https://doi.org/10.1145/3417990.3420210
4. De, B.: API Management. Apress, Berkeley (2017). https://doi.org/10.1007/978-1-4842-1305-6_2
5. de Feijter, R., Overbeek, S., van Vliet, R., Jagroep, E., Brinkkemper, S.: DevOps competences and maturity for software producing organizations. In: Gulden, J., Reinhartz-Berger, I., Schmidt, R., Guerreiro, S., Guédria, W., Bera, P. (eds.) BPMDS/EMMSAD -2018. LNBIP, vol. 318, pp. 244–259. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91704-7_16
6. Hora, A., Robbes, R., Valente, M.T., Anquetil, N., Etien, A., Ducasse, S.: How do developers react to API evolution? A large-scale empirical study. Softw. Qual. J. **26**(1), 161–191 (2016). https://doi.org/10.1007/s11219-016-9344-4

7. Jamshidi, P., Pahl, C., Mendonca, N.C., Lewis, J., Tilkov, S.: Microservices: the journey so far and challenges ahead. IEEE Softw. **35**(3), 24–35 (2018). https://doi.org/10.1109/MS.2018.2141039
8. Jansen, S.: A focus area maturity model for software ecosystem governance. Inf. Softw. Technol. **118**, 106219 (2020). https://doi.org/10.1016/j.infsof.2019.106219
9. Jansen, S., Brinkkemper, S., Cusumano, M.A.: Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry. Edward Elgar Publishing (2013). https://doi.org/10.4337/9781781955635
10. Mathijssen, M., Overeem, M., Jansen, S.: Identification of practices and capabilities in API management: a systematic literature review. Technical report, Utrecht University (2020). http://arxiv.org/abs/2006.10481
11. Mathijssen, M., Overeem, M., Jansen, S.: source data for the focus area maturity model for API management. Technical report, Utrecht University (2020). https://arxiv.org/abs/2007.10611v3
12. Medjaoui, M., Wilde, E., Mitra, R., Amundsen, M.: Continuous API Management: Making the Right Decisions in an Evolving Landscape. O'Reilly Media, Sebastopol (2018)
13. Overeem, M., Jansen, S., Fortuin, S.: Generative versus interpretive model-driven development: moving past 'it depends'. In: Pires, L.F., Hammoudi, S., Selic, B. (eds.) MODELSWARD 2017. CCIS, vol. 880, pp. 222–246. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94764-8_10
14. Poell, T., Nieborg, D., van Dijck, J.: Platformisation. Internet Policy Rev. **8**(4), 1–13 (2019). https://doi.org/10.14763/2019.4.1425
15. Polák, M., Holubová, I.: REST API management and evolution using MDA. In: C3S2E 2015: Proceedings of the Eighth International C* Conference on Computer Science & Software Engineering, pp. 102–109 (2015). https://doi.org/10.1145/2790798.2790820
16. Ralph, P., et al.: Empirical Standards for Software Engineering Research (2021). https://arxiv.org/abs/2010.03525
17. Sanchis, R., García-Perales, O., Fraile, F., Poler, R.: Low-code as enabler of digital transformation in manufacturing industry. Appl. Sci. (Switzerland) **10**(1) (2020). https://doi.org/10.3390/app10010012
18. van Steenbergen, M., Bos, R., Brinkkemper, S., van de Weerd, I., Bekkers, W.: The design of focus area maturity models. In: Winter, R., Zhao, J.L., Aier, S. (eds.) DESRIST 2010. LNCS, vol. 6105, pp. 317–332. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13335-0_22
19. Vincent, P., Iijima, K., Driver, M., Wong, J., Natis, Y.: Magic Quadrant for enterprise low-code application platforms. Technical report, September, Gartner (2019)
20. Weir, L.: Enterprise API Management: Design and Deliver Valuable Business APIs. Packt Publishing Ltd., Birmingham (2019)

Author Index

Betz, Stefanie 163
Bragilovski, Maxim 277
Bührer, Lisa 163

Chen, Qifan 39
Corradini, Flavio 350

Dayan, Hava 318
De Smedt, Johannes 131
De Weerdt, Jochen 131
Döller, Victoria 245
Dubinsky, Yael 47

Elstermann, Matthes 163
Engelberg, Gal 3
Engels, Gregor 262
Engelsman, Wilco 212

Fedeli, Arianna 350
Fellmann, Michael 62
Fögen, Konrad 303
Fornari, Fabrizio 350

García Simón, Alberto 366
Gómez-López, María Teresa 19
Gordijn, Jaap 212
Gottschalk, Sebastian 262
Grinvald, Avihai 96
Gutschmidt, Anne 181

Haaker, Timber 212
Hacks, Simon 303
Hadad, Moshe 3
Hadar, Irit 318
Hellmanzik, Ben 181
Herbert, Tobias 81
Hersén, Nicklas 303
Hillmann, Peter 196
Holz, Felix 62

Jalali, Amin 147
Jansen, Slinger 380

Karagiannis, Dimitris 245
Koç, Hasan 228

Lantow, Birger 62, 181
Lederer, Matthias 163
Lu, Yang 39
Lukyanenko, Roman 293

Maass, Wolfgang 293
Makias, Yifat 277
Mangler, Juergen 81
Marco Palomares, Alejandro 366
Martins, Erko 181
Mathijssen, Max 380
Mokryn, Osnat 96

Nowosad, Alexander 262

Ortega, F. Javier 19
Overeem, Michiel 380

Park, Gyunam 112
Pastor, Oscar 366
Pöhn, Daniela 196
Polini, Andrea 350
Poon, Simon 39

Ramforth, Ben 181
Ramos-Gutiérrez, Belén 19
Re, Barbara 350
Reyes Román, José Fabián 366
Rinderle-Ma, Stefanie 81

Sandkuhl, Kurt 228
Schmidt, Werner 163
Shamshila, Moran 277
Snoeck, Monique 131, 335
Soffer, Pnina 3, 47, 96
Stern, Roni 277
Stirna, Janis 228
Storey, Veda C. 293
Sturm, Arnon 277

van der Aalst, Wil M. P. 112
van der Linden, Dirk 318
van Sinderen, Marten 212
Varela-Vaca, Ángel Jesús 19
Verbruggen, Charlotte 335

Wieringa, Roel 212
Wiese, Matteo 181
Wynn, Moe Thandar 19

Yigitbas, Enes 262

Zamansky, Anna 318