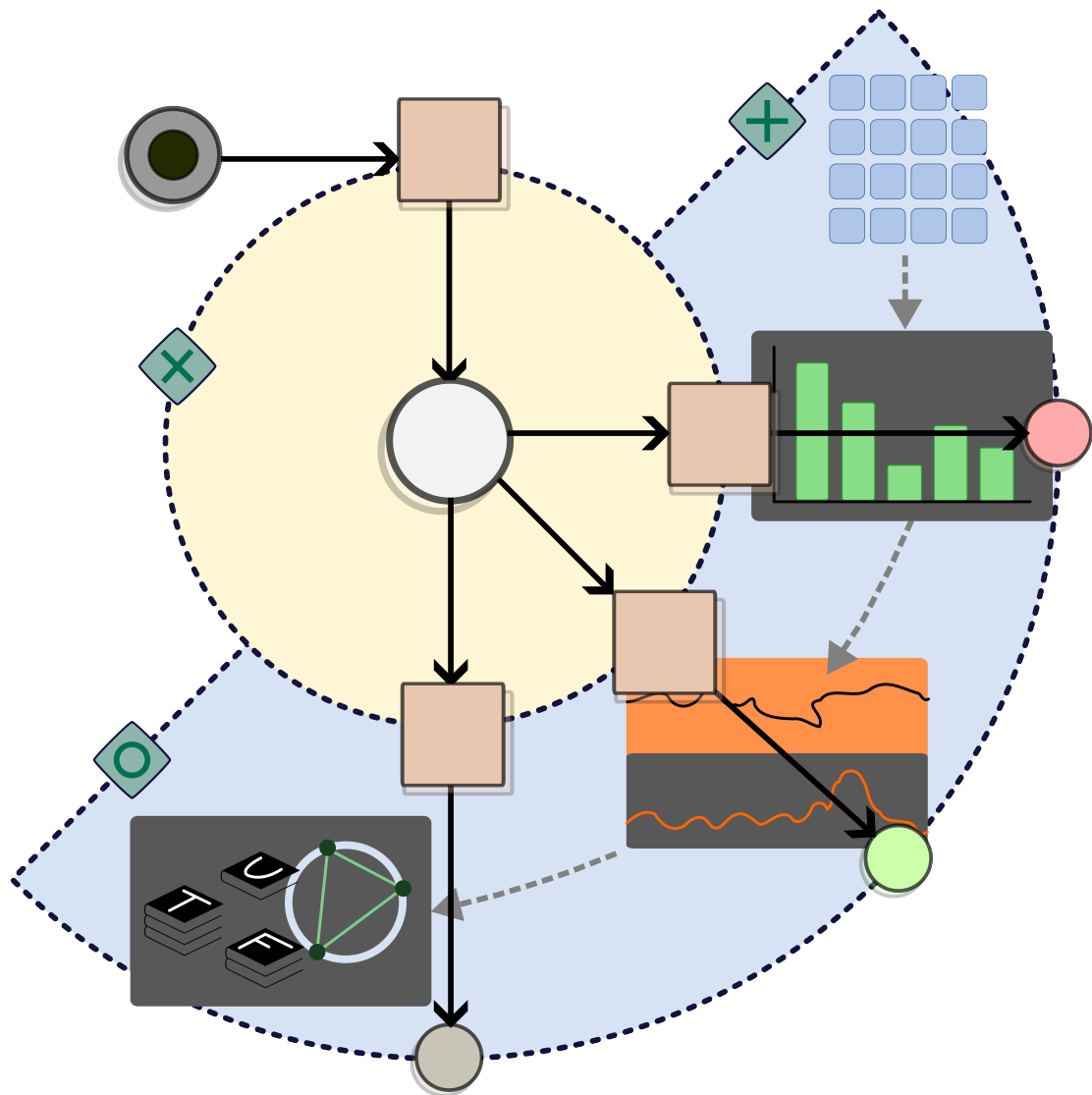


Process Mining with Exogenous Data

Rediscovering Decision Making in Business Processes



Process Mining with Exogenous Data

by

Adam Banham

BInfoTech(Hons)

Submitted in fulfilment of the requirement for the degree of
Doctor of Philosophy

School of Information Systems

Faculty of Science

Queensland University of Technology

2025

Keywords: Process Mining, Decision Mining, Time Series, Exogenous Data,
Multi-Perspective.

This thesis presents algorithms, or specifications, revolving around data.

“A specification is often of the form ‘find the mimsiest borogove that is slithy’. Borogove is some noun, and mimsy and slithy are some adjectives, the meanings of which are not important for this discussion.”

— Johan Theodoor Jeuring, *Theories for Algorithm Calculation*

These algorithms consider the observed behaviour of processes, or systems, through modelling formalisms.

“There are several reasons for modelling a system, e.g. to create and evaluate a design of a new system, to compare alternative designs and to investigate possible improvements in a real system. Model building forces us to organize, evaluate and examine the validity of our thoughts. This way modelling reveals errors and possible improvements.”

— Willibrordus Martinus Pancratius van der Aalst, *Timed Coloured Petri Nets and their Applications to Logistics*

The outcomes from these algorithms are logical consequences of facts observable in the behaviour of a process.

“By a valid argument we mean an argument whose premises and conclusion are such that the truth of the former involves that of the latter: if the premises of a valid argument are all true, then its conclusion must also be true ... The conclusion of a valid argument is said to be a logical consequence of its premises.”

— L. T. F. Gamut, *Logic, Language, and Meaning*

Acknowledgements

An incomplete list of insurgents which have contributed to searching for a new model will make our existing options obsolete — Buckminster Fuller

To the usual suspects...

Principal Supervisor, Prof. Moe T. Wynn

As the powder peg to this journey, I often wonder what she thought when we first crossed paths in an undergraduate summer research program; it's best to leave that unanswered. Instead let us focus on the magnitude of opportunities and resources she has provided to enable me to discover process mining; a debt I will not be able to repay but will nonetheless attempt to do so.

External Supervisor, Prof. Sander J.J. Leemans

As the fire to powder peg, I have always been inspired by his fiery motivation for processes and process mining. Unfortunately, I often found myself too captivated by the fire and unable to do anything but walk on the toes of my brilliant supervisor. I am gracious for his patience as I learned to be a better collaborator and for his endless support.

Associate Supervisor, Dr. Robert Andrews

In comparison to the troublemakers above, Robert has been the careful architect ensuring the explosive end has the right impact and the largest possible effect. Robert took me on during several projects and helped me understand the nature of processes through constructing event logs. While, us troublemakers are not rock stars like Robert (and his band Jump The Shark), your mentorship has amplified my passion for research up to eleven.

To other brigands supporting our search...

Prof. Arthur H. M. ter Hofstede

I am ashamed I haven't had more time for tangents; when taking a meeting always allow enough time for at least one. These tangents into the world of academia based on your decades of research have, and continue, to be eye opening. Perhaps in the future, we will finally organise that podcast to collect all these tangents.

Prof. Felix Mannhardt

A good example empowers research. Unknowingly at the start of my candidature, I took the time to indulge on multi-perspective process mining as an example that would guide the course of my candidature. Felix, from the other side of the world, would later offer his support to help me understand how to make great examples through dense topics. He endured all my endless poking and prodding into the mechanisms of his work all while managing sick children, early hours, and long meetings throughout my candidature. Thank you for taking the time in the early mornings to discuss all matter of things, and I hope they were as enjoyable for you as they were for me.

Dr. Syed Rehan

Watching Dr. Evil boiling and brewing some of the most horrible looking ramen and instant noodles for lunch has been one of the most enjoyable times during my candidature. When our stomachs were filled or hunger evaded us, we turned to the world and discussed its darker underbelly, letting our curiosity boil and brew instead. Especially so, without any sunshine around to keep ourselves in line and avoid any dangerous crevasses. Syed has always been a strong supporter, perhaps too much; I still have not forgiven you for dragging me up to front of an undergraduate workshop (while taking the unit) to teach the workshop.

Dr. Adam Burke

Being somewhat doppelgängers, Dr. Burke and myself have often worked in parallel during our candidatures. However, Adam has been more of a shepherd to me, keeping away the behaviours that come from being trained by wolves. He strongly suggested the usage of “tests” and “common technical practices” like building your software implementation. Not all his teachings rubbed off on me, unfortunately. I still believe in human interfaces over fully embracing the light of the command line. I am not sure where I land, either as a wolf, a part of his flock, or a wolf hiding in plain sight within the flock. Nonetheless, I have valued all our discussions and they have been eye opening to me.

The unnamed reading group — Dr. Adam Burke, Dr. Jing (Roy) Yang, Dr. Mythreyi Velmurugan, to name a few

Our times exploring publications kept me on my toes and open to many research methodologies. Together we searched for the ideal way to read a paper, ultimately we landed on the following: understanding the authors' positions at the time of publication, discussing the venue and duration of the review, considering the written style chosen, whether there was a decent illustration included (or why illustrations were used), and if we had enough time left over we might delve into the content and future applications. Often we explored several tangents and beliefs of the group before completing these tasks, but these weekly meetings were a breath of fresh air and I am humbled by your continued participation in these discussions.

Professor Kevin B. Laupland and Dr Lucy Shinnars

I am indebted to these experts who through many long meetings helped me navigate medical data extracted from the ERP systems of hospitals. Together they provided insight across the multi-disciplinary boundaries between information systems, computer science, healthcare and shared their expertise as practitioners. Professor Kevin B. Laupland has continued to explore this challenging intersection of multi-disciplinary research in parallel to this candidature. I am grateful for his patience as we learnt to traverse this intersection academically.

To those that joined the lunch time table...

Dr Christoph Niesel while our overlap was short lived, you always brightened the discussion of the table. Dr Rebekah Eden tirelessly worked to ensure everyone joined the table for some portion of the lunch hour, even if they were dragged kicking and screaming. We didn't realise how blessed we were until she had left us behind. Dr Erwin Felt had the pleasure, hopefully, of sharing a "cubicle" wall with myself in our open plan desks. Your thought provoking discussions around the table, and your interesting talking points were a delight at lunch. Dr Alireza Nili, your softly spoken stories and wonderful laughter filled the table with worldly insight. Associate Professor Wasana Bandara, when you joined the table you always had a great success story to share. I am grateful for the work you put into running the doctoral consortium each year. Similarly, Dr Sareh Sadeghianasl and Dr Kanika Goel your presence at the table often led to long discussions.

Luckily, I have enjoyed an environment at the school of information systems where diversity is empowered, but does make it challenging to schedule a clear time for lunch with the many opportunities are available. While Professor Alistair Barros rarely made it to the table, he always took the time to grab everyone for a discussion over coffee or in the slow afternoons. My supervisors and Prof. Arthur ter Hofstede encouraged making the lunch table a potential opportunity for collaboration and led the charge for the common lunch hour, of which I have embraced. Too many tangents to record have been led by Arthur to encourage discussion and anyone coming to QUT should ensure they catch one or two. Thanks to the many visiting students from Germany who shared their experiences as well as the many visiting scholars who joined the doctoral consortiums and shared their expertise. I owe thanks to Professor Mahendrawathi ER, Dr Rebekah Eden, Prof Eduard Hoenkamp, Prof Colin Fidge, Associate Professor Marco Comuzzi , and Dr. Stephen Smith for their efforts at these doctoral consortiums and helping me improve my presentation skills. I believe there is another table on level seven, but unfortunately I had not mastered stairs yet. So I often overlooked the many amazing souls working above myself on level 6. Apologies for not adventuring enough and joining the higher table.

My journey through the Queensland University of Technology as a undergraduate to a doctoral student would not possible without the wonderful souls who educated me during these studies. I am thankful that during my study these souls provided in-person lectures to students, which I ungracefully raided for all their worth.

Professor Paul Roe whose first year unit introduced the notions of computer science to me and is still embodied within my mindset. Professor Colin Fidge who I have crossed paths with several times (all of which were beyond insightful), and introduced myself to the world of programs. Associate Professor Artem Polyvyanyy who may of incited this whole ordeal by being the first to inform me that world is indeed made up of processes and the power of Petri nets. Dr Reihaneh Bidar who tutored me for Petri nets and too often called myself for answers in the workshops when no one else would answer. The aforementioned Dr. Evil who taught me all about the wonderful power of business process management. Alongside, I again owe Professor Sander Leemans for his lectures on business process model and notation. Dr Elliot Carr for introducing computational mathematics and mathematical denotation through amazing working lectures (pen on “paper” and the overhead projector on).

Dr Pamela Burrage who explored and envisaged a world of simulation using mathematics also through wonderful working lectures. Professor Niko Suen-derhauf & Distinguished Emeritus Professor Peter Corke for their amazing course on robotics. Dr Andrew Gibson and Dr Catarina Pinto Moreira who prepped my skills for business intelligence and deriving insights for business stratagems. Prof Yuefeng Li who gracefully endured somewhat empty lectures around data science tooling with just myself, of which I greedily consumed. Associate Professor Yue Xu who explored ETL, data transformation, and OLAP databases techniques in her also somewhat undiscovered lectures. Dr Mahsa Baktashmotlagh who introduced the wonderful world of algorithms that machine learning uses to analyse challenging questions. Associate Professor Wayne Kelly who during my honours explored programming paradigms and taught me how to handle computation parallelism all while COVID-19 was being introduced to the world. Professor Guy Gable who gracefully allowed me to interview you during my doctoral studies and provided an insightful set of pre-doctoral talks to empower future doctoral studies within information systems. I apologise for not naming every single wonderful soul that I had met during my studies, if your name is missing it is not intentional by any means.

These adventurous endeavours would not of been possible without the support of...
the Australian government through the research training program stipend scholarship which allowed me to focus on performing high-quality research without worrying about my next meal and keeping a roof over my head...
the Queensland University of Technology with a wonderful desk and an amazing view, supporting the candidature with all the equipment needed for software development, high-performance computing, and for the most part, a secure network (internet, data storage, and a safe working environment) to conduct this research...
the Centre of Data Science at Queensland University of Technology also extended their support by offering many interesting workshops and development opportunities. Their support also meant I could afford coffee and some delights to enjoy life on top of the scholarship...
Rheinisch-Westfälische Technische Hochschule Aachen for the opportunity for scholars to visit and selecting my application for a three month research trip to Europe at their amazing university...
Finally, professional editor, Karyn Gonano, provided copyediting and proof-reading services, according to the guidelines laid out in the university-endorsed national 'Guidelines for editing research theses'. I am thankful for the editing services of Karyn Gonano, who enables people to craft words that influence.

Abstract

Woe to the Wretched; Searching for Bliss in Processes

In the modern era, humanity collects endless amounts of data and performs hours of analysis to derive information for the optimisation of our behaviour, all to earn an extra piece of bliss for tomorrow. The definition of ‘bliss’ is irrelevant and whatever it is, will come from the efforts of the wretched, individuals or organisations, improving their behaviour to obtain more of it. For organisations, bliss may be another day in operation, happier customers, greater efficiency of internal processing of requests, or compliance with company policies. The wretched yearn for techniques that quantify the impact of their behaviour towards their goals. Process mining is a discipline of research that focuses on the study of processes, providing techniques for the wretched, and this thesis contributes to its advancement.

A *process* is triggered by some event occurring, leading to the execution of activities and achieving an outcome. Consider the journey of a patient being admitted to a hospital, where tests, procedures, and patient movements are recorded as actions. The actions and data related to completing actions are internal to the process execution, or *endogenous* data. Studying endogenous data can reveal dependencies between actions, like to be discharged from a hospital, a patient needs to be admitted to a hospital. However, the discrete nature of endogenous data struggles to capture the context surrounding processes or external data streams, which are temporal in nature. For instance, once a patient is admitted to a ward, sensors or constant nurse examinations are conducted to measure the patient’s vital signs. These measurements may externally influence actions in processes, like irregular heartbeats informing that an electrocardiogram should be the next action performed. Observations surrounding the enacted actions of processes, or *exogenous* data, are important to consider when investigating processes. To date, the investigation of external, extrinsic, extraneous, or *exogenous* influences on processes has received little focus in literature. Therefore, this thesis serves to address this gap by providing techniques, tools, and visualisations to study exogenous influences on processes.

This thesis uses historical recordings of a process to understand why behaviour occurs and to clarify how choices within processes lead to such behaviour. These recordings are compared to a formal process model description using process modelling formalisms. Pairing a model with an execution of a process enables the consideration of states, likely reached throughout the execution. Investigating the traversal of these states can reveal that an execution settled a choice, i.e. several actions were possible, but executing one settles the choice between them. This thesis assumes actors executing processes follow the constraints of processes, but when

faced with a choice, their decision making used to settle a choice may be influenced by trends, seasonal dependencies, spikes or troughs in exogenous data, or the lack of available exogenous data.

Firstly, this thesis contributes a framework which provides a means for connecting the historical recordings of a process with exogenous data sources and focuses on the temporal nature of exogenous data. Secondly, a visualisation technique over a process model and ranking procedure is contributed which identifies exogenous influences on the outcomes of processes. Thirdly, a process mining technique is contributed to quantify the likelihood of an action occurring based on exogenous data, allowing the influences of exogenous data on the behaviour of processes to be quantified. Lastly, an approach to quantifying process models enhanced with decision making is contributed to ensure that such process models can be compared and quantified based on desirable properties rather than ad hoc measurements. Together these contributions allow the wretched to explore how the decision making in processes can be influenced by exogenous data and how to quantify the impact on behaviour in processes.



Contents

Title Page	
Keywords	i
Research Aspirations	ii
Acknowledgements	iii
Abstract	viii
Contents	x
List of Figures	xiv
List of Tables	xix
1 An Introduction to Processes	1
1.1 Describing and Recording Processes	1
1.2 Process Mining: A Discipline for Studying Processes	4
1.3 Decision Making in Processes	6
1.4 Exogenous Data	7
1.5 Problem Statement	9
1.6 Research Questions	11
1.7 Contributions	12
1.7.1 Approach	14
1.8 Publications	14
1.9 Structure of Thesis	15
2 Management, Workflows and Processes; A Snapshot of Literature	16
2.1 The Science of Studying Processes	16
2.2 Process Mining: A Formal Study of Processes	21
2.2.1 Contextual Data for Process Mining	24
2.2.2 Multi-Perspective Process Mining	26

2.2.3	Process Mining in Healthcare	27
2.2.4	Process Mining with Internet of Things	28
2.3	Process Discovery	29
2.3.1	Dependency Graphs	29
2.3.2	Petri Nets	30
2.3.3	Process Trees	32
2.3.4	DECLARE	34
2.4	Conformance Checking	35
2.4.1	Traversing Models using Alignments	35
2.4.2	Data-aware Conformance Checking	37
2.5	Process Enhancement	39
2.5.1	Decision Mining	40
2.5.2	Stochastic Discovery	41
2.6	Research Gaps and Summary	43
3	A Framework for Process Mining with Exogenous Data	49
3.1	Introduction	49
3.2	Approach	51
3.3	Preliminaries	51
3.4	The xPM Framework	53
3.4.1	Exemplar Scenario	54
3.4.2	$\mathcal{L}$ – Linking	55
3.4.3	$\mathcal{S}$ – Slicing	56
3.4.4	$\mathcal{T}$ – Transformation	57
3.4.5	$\mathcal{D}$ – Discovery	58
3.4.6	$\mathcal{E}$ – Enhancement	59
3.5	Explorative Exogenous Series Analysis (EESA)	59
3.5.1	Overlapping EESA	62
3.5.2	Highlighting Points of Interest	63
3.6	Evaluation	65
3.6.1	Exogenous Influences on Model Quality	65
3.6.2	Exogenous Influences on Process Outcomes	72
3.7	Reflections and Future Work	78
3.8	Summary	79
4	Discovering the Influence of Exogenous Data on Decisions in Processes	81
4.1	Introduction	81
4.2	Motivating Example	83
4.3	Approach	93
4.4	Preliminaries	97

4.5	Stochastic Petri Nets with Exogenous Dependencies	100
4.6	Discovery	104
4.6.1	Constructing Choice Data	104
4.6.2	Handling Model Moves	105
4.6.3	Solving Equalities	108
4.6.4	Visualisation	109
4.7	Conformance Checking	112
4.8	Evaluation	113
4.8.1	Event logs and Exogenous Data	113
4.8.2	Log Completeness	114
4.8.3	Model Quality	116
4.8.4	Changes in Firing Likelihoods	117
4.9	Reflections and Future Work	118
4.10	Summary	119
5	Comparing Conformance Checking for Decision Mining: An Axiomatic Approach	124
5.1	Introduction	125
5.2	Approach	126
5.3	Problem Motivation	128
5.4	Preliminaries	130
5.5	Foundations	133
5.5.1	Running Example	134
5.5.2	Reinforcing and Weakening Guards	134
5.5.3	Playing out a model	137
5.5.4	Constructing a tree	139
5.5.5	Finding a Matching	143
5.5.6	Diagnosing a Model	146
5.6	Quality Axioms	148
5.7	Proposed Measures	151
5.7.1	Guard-Recall	151
5.7.2	Guard-Precision	152
5.8	Evaluation	153
5.8.1	Existing Techniques	153
5.8.2	Exemplar Log and Model	153
5.8.3	Counter-examples	154
5.8.4	Implementation	155
5.8.5	Results	155
5.9	Reflections and Future Work	158
5.10	Summary	159

6	Concluding the Process	161
6.1	The Contributions in Review	162
6.2	The Research in Retrospect	163
6.3	The Validity in Hindsight	166
6.4	The Future in Foresight	168
6.5	The Takeaway	171
	References	172
7	Biography	192
	Appendices	194
A	Conformance Checking	195
A.1	Existing measures	195
A.1.1	Guard-Recall Techniques	195
A.1.2	Guard-Precision Technique	196
A.2	Proof Sketches	197
B	Tooling	200
B.1	ExogenousData (ProM plug-in)	200
B.1.1	Data Structures	200
B.1.2	xPM Framework	202
B.1.3	Decision Mining with Exogenous data	206
B.1.4	EESA Analysis	208
B.1.5	Stochastic Discovery	209
B.1.6	Data-aware Conformance Checking	210
	Indexed Terms	212

List of Figures

1.1	An example of a data-aware process model, where transitions are annotated with guards. The derivation of this data-aware model is explored in Section 3.6.1.	4
1.2	Examples of choices in processes; circles represent a state of a process; rectangles represent an activity; arcs denote how actions move the state of the process from one to another.	7
1.3	Examples of exogenous data and endogenous data from the road fines log. The exogenous data are represented as coloured line series; red for the total amount of unpaid fines, and purple for the number of unpaid fines. The endogenous events are represented as vertical lines and the first event is highlighted in yellow.	9
1.4	The six key contributions established by this thesis. Red arrows show where the contributions of this thesis exist, while black (dashed) arrows show where existing process mining techniques are used.	13
2.1	Process mining's rise compared with business process management (BPM) and workflow management, based on published and indexed documents on Scopus 1986–2023 ¹ .	17
2.2	Lifecycles developed for the management of processes.	20
2.3	Common phrases used as keywords or within abstracts from process mining documents extracted in Figure 2.1.	21
2.4	van der Aalst [2] presents process mining as the bridge between the fields of process science and data science. Reproduced with permission from Springer Nature.	22
2.5	Several onions that peel back the factors that may influence processes.	25
2.6	Mannhardt, de Leoni, Reijers, <i>et al.</i> [147] presents an example of guided process discovery, where low-level events are abstracted into higher level activities. Reproduced with permission © 2018 Elsevier.	29
2.7	Agrawal, Gunopulos, and Leymann [94] presents how to discover a dependency graph using four executions $\{ABCF, ACDF, ADEF, AECF\}$. Reproduced with permission from Springer Nature.	30
2.8	Several Petri nets discovered by α miners using the road fines log [25].	31
2.9	Breakdown of the process tree $\rightarrow (A, \times(B, C), D)$ into a sound Petri Net.	32

2.10	Translated Petri nets discovered by several miners based on the inductive framework [36] for the road fines log [25].	33
2.11	A DECLARE model [164] using the extracted MIMIC III log [23], [24] for this thesis.	34
2.12	Examples of replaying the trace ‘N74775’ from Table 1.1 over Figure 2.10a using alignments. A synchronous move is highlighted in green; a model move is highlighted as orange; and, a log move is highlighted as red. Highlighted in each sub caption is the log move that can be arbitrarily moved with any cost change.	36
2.13	Several decision extensions of the process model seen in Figure 2.10b. Decision points are highlighted as green regions of the net and guards are visualised on the transitions.	42
2.14	Dependencies stochastic miner [184] applied to the process model seen in Figure 2.10b. The semantics of this model is explored in Section 4.2.	45
2.15	Data-aware stochastic miner [76] applied to the process model seen in Figure 2.10b. The semantics of this model is explored in Section 4.2.	46
3.1	The xPM framework.	53
3.2	An illustration of linking. Given a trace and an exo-panel, the linking function returns a linked exo-series for the given trace.	55
3.3	An illustration of slicing, where given trace_1 and exo-series_1 , some events are associated with the same slice (e_4, e_5), some have their own slice (e_6, e_7, e_8), and others are associated with an empty sequence (e_1, e_2, e_3).	56
3.4	An illustration of transformation, where given e_7 and a slice of an exo-series $\text{es}_{1,3}$, the transformation derives the velocity of a slice ($\text{es}'_{1,3}$), models the derivation ($\text{es}''_{1,3}$), then extracts notable features from the modelling as transformed attributes.	58
3.5	A comparison of visualisation techniques, one using the exogenous slices as-is (top), and another showing the generalisation of trends seen in raw slices (bottom). The usage of colour in each figure, is to highlight that a slice is associated with a particular process outcome.	60
3.6	Construction of shared time axis (left) and interpolation of values for shared time axis (right).	61
3.7	Demonstrations of EESA visualisation process from slices (top) to generated outcome (bottom)	62
3.8	Measurement of an overlapping EESA graph (left), where a dynamic time warping matrix is computed between median trends (middle) then the summarisation of the path produces a measurement (right).	63

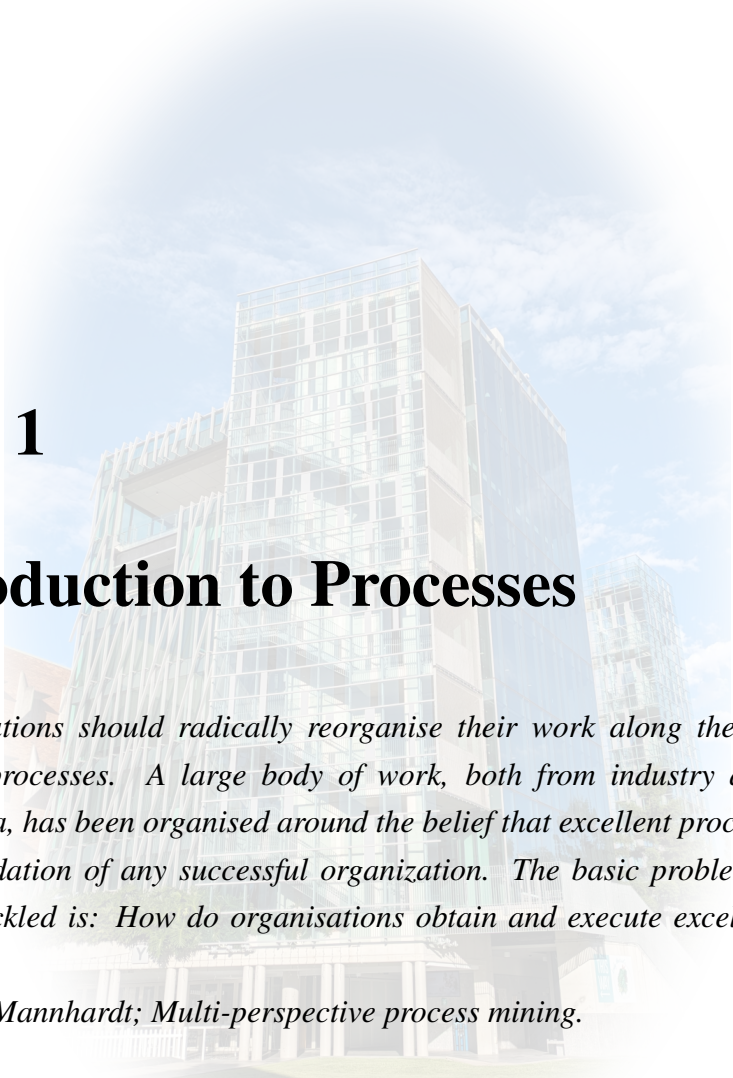
3.9	Example of ranking EESA visualisations, rank is denoted in red box, in top left. All non-overlapping EESA are ranked at $n + 1$ (4) while overlapping EESA are ranked based on distinctiveness between outcomes (1-3).	64
3.10	The three xDPNs discovered by [57] using both endogenous and exogenous data for the movements log. Transitions that are associated with a precondition are hatched.	68
3.11	The three xDPNs discovered by [58] using both endogenous and exogenous data for the procedures log. Transitions that are associated with a precondition are hatched.	69
3.12	Handmade model for analysis of ICU procedures.	74
3.13	Histogram showing the top-ranked vital signs in EESA visualisations (only considering EESA visualisation where at least 1% of procedures were associated with sepsis).	76
3.14	A top-ranked EESA of heart rate measurements linked to 20,342 intravenous cannula procedures, of which, 278 procedures were associated with a patient acquiring sepsis. Patients' heart rate measurements were taken from the two-hour window prior to the patient undergoing the intravenous cannula procedure. All procedures occurred within the first 48 hours of admission.	77
3.15	A top-ranked EESA of heart rate measurements linked to 262 intravenous cannula procedures, of which, 11 were associated with a patient acquiring sepsis. Patients' heart rate measurements were taken from the two-hour window prior to the patient undergoing the intravenous cannula procedure. All procedures occurred after the patient had been admitted for 25 days. Observe that heart rate may discriminate between patients associated with sepsis.	77
4.1	Probability of credit collection for two traces across the markings traversed using equally distanced samplings between markings for the SLDPN with 'waiting'.	85
4.2	The normative model capturing the decision points (in green) and highlighting credit collection in blue and hatching. The solid directed arcs denote the dependencies of the underlying Petri net, while the dashed directed arcs denote the sequence of moves in the alignment. Moves in the alignments that could have fired credit collection are highlighted in the sub captions. Capturing alignment for trace 'N74775', $\langle (e_{1t_1}), (e_{2t_2}), (e_{3t_5}), (e_{4t_{10}}), (e_{5t_8}), (e_{6t_{13}}), (e_{7t_{16}}), (e_{8t_{18}}), (\gg_{t_{11}}) \rangle$	86
4.3	The normative model capturing the decision points (in green) and highlighting credit collection in blue and hatching. The solid directed arcs denote the dependencies of the underlying Petri net, while the dashed directed arcs denote the sequence of moves in the alignment. Moves in the alignments that could have fired credit collection are highlighted in the sub captions. Capturing alignment for trace 'A8690', $\langle (e_{9t_1}), (e_{10t_3}), (\gg_{t_2}) \rangle$	87

4.4	The normative model capturing the decision points (in green) and highlighting credit collection in blue and hatching. The solid directed arcs denote the dependencies of the underlying Petri net, while the dashed directed arcs denote the sequence of moves in the alignment. Moves in the alignments that could have fired credit collection are highlighted in the sub captions. Capturing alignment for trace ‘A21188’, $\langle (e_{11}t_1), (e_{12}t_2), (e_{13}t_5), (e_{14}t_{10}), (e_{15}t_{12}) \rangle$	88
4.5	Discovered stochastic labelled Petri net [183, Section 3.6] and only uses base weights.	89
4.6	Discovered SLPN-SD [184, Definition 1] and uses base weights and parameters.	90
4.7	SLDPN discovered only using event attributes in Table 4.1.	91
4.8	SLDPN discovered using event attributes in Table 4.1 and the waiting time between events.	92
4.9	Changes in activity likelihoods for a discovered Exo-SLPN for the smart factory log.	94
4.10	Changes in activity likelihoods for a discovered Exo-SLPN for the MIMIC log.	95
4.11	Changes in activity likelihoods for a discovered Exo-SLPN for the road fines log.	96
4.12	Exo-series describing inter-case variables for the road fines log.	98
4.13	The trace ‘A21188’ from Table 4.1 annotated with two exo-series. Verticals denote when an event occurred in the trace, a dashed vertical denotes the first event, and e_{14} is highlighted.	99
4.14	The probability of credit collection firing between events according to the Exo-SLPN in Figure 4.15.	102
4.15	Exo-SLPN discovered from the road fines log using the normative model. Weights (φ, ψ) not set to one and associated with a transition are denoted on the transition.	103
4.16	Discovering an Exo-SLPN.	104
4.17	Illustration of where a model move in between two synchronous moves may have occurred (in green).	107
4.18	Depicting the changes in the likelihood of firing transitions for the road fines log based on the discovered Exo-SLPN using the IMf model in the evaluation (Section 4.8).	111
4.19	Results from sampling testing using the road-fines log. The line of best fit and 95% confidence levels are shown to evaluate the trends across runs.	115
4.20	Firing likelihoods changes for the MIMIC log using the IM _f control flow model.	120
4.21	Firing likelihoods changes for the MIMIC log using the IM _{po} control flow model.	121
4.22	Firing likelihoods changes for the MIMIC log using the DFM control flow model.	122
5.1	An example Petri net with data, with guards above transitions.	126
5.2	The three conformance checking contributions that address RQ2.	127

5.3	Conceptualisation of a system reacting to data and process activities [66], where the system signals that a trace is complete when it presents an empty offer $\emptyset$. . .	129
5.4	Example data-aware model using Petri net with data.	134
5.5	Example executions for the running example.	138
5.6	A step-wise introduction to the derivation of a play-out tree from a set of play-out traces.	142
5.7	Comparison of outcomes after flow reduction has occurred. Changed flows are highlighted in red.	143
5.8	A series of counterexamples using one log, where true guard evaluations are monotonically increasing from (a) to (c) with respect to L_E^k , where the number of compliant events in traces for models are: (a) 1, (b) 2, (c) 4. Measurements are denoted in sub captions. The variable d_1 has an integer domain of $[5 \dots 10]$ and is written by ‘a’.	156
B.1	GUI wizards for constructing an exogenous annotated event log.	203
B.2	The exogenous trace explorer enables users to explore exogenous and endogenous data.	205
B.3	Options for users to compare behaviour in exogenous data.	205
B.4	GUI interface for exogenous aware discovery.	206
B.5	DPN discovered by applying decision mining.	207
B.6	Visualisation of the guard for transition ‘H’.	207
B.7	GUI interface for exogenous aware discovery after applying data-aware conformance checking.	208
B.8	The ranking of EESAs within ProM.	208
B.9	Stochastic discovery plug-ins.	209
B.10	Exo-SLPN visualised within ProM.	209
B.11	Conformance checking plug-ins for ProM.	210
B.12	Visualisation of a playout tree for the exemplar included in the source repository for exogenous data, where k was 16.	211

List of Tables

1.1	An extract from the road fines log [25].	2
2.1	A snippet of the Iris flower dataset used by Fisher [113], where each row is independent from another.	23
2.2	An extract from our MIMIC III log [24], [47] in tabular like form.	24
3.1	Exemplar event log for ambulance triage.	54
3.2	Exemplar of a tabularised exo-description. All timestamps occur on 18-01-2022 in this example.	55
3.3	Breakdown of measured model quality for discovered xDPNs across two event logs and three variants. The best measurement for an event log is highlighted in bold. In both event logs, including exogenous data lead to a higher determinism.	71
4.1	An extract from the road fines log [25].	84
4.2	Choice data extracted for example 4.	106
4.3	The constructed equalities for the running example. Panel names have been abbreviated, #fines is #F and \$fines is \$F. Red equalities are not necessary for the solver.	109
4.4	Perturbations of choice data using the choice data from example 4 and the discovered Exo-SLPN in Figure 4.15 for probabilities. Perturbations are highlighted in red. The three rightmost columns are used in the visualisation tool.	112
4.5	Descriptive qualities of event logs used in our evaluation.	114
4.6	Conformance for discovered models with our method highlighted. Highest control-flow scores are bolded. Highest data-aware scores are underlined.	117
5.1	Counter-example testing across our axioms.	157
6.1	Retrospective of the contributions for research question one.	165
6.2	The retrospective outlook on the contributions for research question two.	166



Chapter 1

An Introduction to Processes

Organisations should radically reorganise their work along their value-adding processes. A large body of work, both from industry and from academia, has been organised around the belief that excellent processes are the foundation of any successful organization. The basic problem that is being tackled is: How do organisations obtain and execute excellent processes?

— Felix Mannhardt; *Multi-perspective process mining*.

The wretched come from all walks of life and may be unfamiliar with processes, as presented in this thesis. As such this chapter provides a short and non-technical primer of processes. Then, process mining is positioned as a discipline devoted to understanding and clarifying the operational behaviour of processes. Next, the focus of the thesis is introduced, which is the decision making that allows actors in processes to determine the most appropriate next action. Followed by the notion of what is not and what is *exogenous* data and its usefulness for studying decision making used in processes. The remainder of the chapter is devoted to presenting the problems addressed by the work, the research agenda, the research questions, the research approach, contributions, and publications. Finally, the structure of the thesis is clarified for the busy or well-informed wretched seeking the contributions of the thesis.

1.1 Describing and Recording Processes

Conceptually, a process describes behaviour, that is the execution of activities that achieves a goal [1]–[3]. Typically, processes have owners, for example managers or a company, which derive value from the goals of the process being achieved. To achieve these goals, activities are executed within the process by actors which include people or information systems. However, processes are often messy and complicated meaning it can be difficult to comprehend them, as describing all the possible variations may make them incomprehensible to the actors and owners [4]–[7]. Investigating processes enable both the owners and actors to understand the

Table 1.1: An extract from the road fines log [25].

trace	event	event attributes or <i>endogenous data</i> process activity	lifecycle	resource	amount	expense	pay amount	timestamp
N74775	1	create fine	completed	555	35.0			28.05.2005@00:00
	2	send fine	completed			16.0		21.09.2005@00:00
	3	insert fine notification	completed					01.10.2005@00:00
	4	add penalty	completed		71.5			30.11.2005@00:00
	5	insert date appeal to prefecture	completed					30.11.2005@00:00
	6	send appeal to prefecture	completed					09.01.2006@00:00
	7	receive result appeal from prefecture	completed					17.02.2006@00:00
	8	notify result appeal to offender	completed					31.03.2006@00:00
A8690	9	create fine	completed	557	36.0			08.02.2007@00:00
	10	payment	completed				36.0	21.02.2007@00:00
A21188	11	create fine	completed	563	22.0			14.07.2008@00:00
	12	send fine	completed			19.10		10.12.2008@00:00
	13	insert fine notification	completed			19.10		18.12.2008@00:00
	14	add penalty	completed		44.0			16.02.2009@00:00
	15	send for credit collection	completed					15.10.2010@00:00
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

past executions of a process. Analysis of processes allows for deriving insights and efficiencies, highlighting problematic activities, or checking policies, where a key benefit is to provide clarity around the process [8]–[10] for further analysis. Refining the understanding of processes through a representation based on historical recordings ensures that each refinement is based on evidence and not conjecture.

Examples of processes are abundant in our daily lives. For instance, insurance companies have processes for handling claims, each with a unique cohort of customers. To improve their processes, companies might consider the cost of their processes [11], [12], the capacity to handle claims in case of sudden influxes [13]–[15], or the fairness of these processes [16], [17]. Similarly, farmers have processes for maturing their crops to harvest, where each crop has to be carefully handled from planting to landing on shop shelves. To improve their harvests, farmers could consider optimising the timing of the harvest based on contextual information [18], such as the temperature, rainfall and fire danger ratings [19]. Likewise in the healthcare domain, a plethora of processes occur within hospitals as health professionals collaborate and interact with patients. To improve patient outcomes, analysis may focus on the journey of a patient [20]–[22] and understanding patterns in the vital signs of patients to inform decisions [23], [24]. This thesis argues that to study the behaviour of processes, analysis techniques should consider data beyond the executed activities to provide a clear representation of the process.

To analyse the behaviour of processes, historical recordings of process executions are used. Historical recordings, or *traces*, are sequences of state changes for the process. State changes, or *events*, describe a moment where an action was executed in the process, and hopefully moved the state of the process towards a goal. These events describe the activity being executed, the time it occurred, and the trace it belongs to. For example, rows in Table 1.1 are events, where an event occurred at 14.07.2008@00:00 for the activity ‘create fine’, which denoted its completion (lifecycle) by resource ‘563’ with a data attribute ‘amount’ being written with the

value of 22. Typically, events are only recorded when activities are completed, but may also be recorded when activities are assigned and started [26]. Notably, an event is not equivalent to a process activity as many events may describe the execution of a process activity. Furthermore, a single trace may be insufficient to fully describe the behaviour of a process, as choices between many activities may exist, activities may be started and completed in parallel, and previous activities may determine future behaviour.

To investigate processes, a collection of traces or an *event log* is required. Event logs are crucial to analysing processes and in many cases, extracting and constructing an event log represents 80% of the work in process improvement projects [27]. Importantly, event logs are samples of a process and reflect the nature of the process through a flattened slice of reality [2]. Table 1.1 represents a sample from an Italian information system regarding the handling of fines handed out by local police officers [25]. This event log contains events with attributes beyond the activity and timestamp, which may inform further analysis. Any data found within an event log is referred to as *endogenous* process data, or simply endogenous data, as opposed to *exogenous* data which focuses on contextual and temporal information surrounding executions of processes. Limiting process analysis to endogenous data means external influences have to be flattened to when events occurred, which means new techniques that intertwine exogenous and endogenous data are required to better study exogenous influences on processes.

Event logs represent the observed reality of a process, which often contrasts with our descriptions of processes, known as process models. A process model is a formal description of process behaviour which prescribes constraints over its activities. These constraints can be that activity *a* must be executed before *c*, that the process can repeatably execute the sequence of *c d e*, or that there is a choice between *c f* and *g*. Describing the constraints over the executed activities of a process is often called the control-flow perspective. Process models have many usages and can be derived from practice or descriptively by owners. This thesis instead, focuses on process models derived from historical recordings rather than descriptive models.

Process models avoid explicitly stating all the constraints over the activities and instead, are presented as graphs, and investigating the execution semantics of these graphs means the constraints over activities are retained. Take for example Figure 1.1, where circles (*places*) describe a possible state of a process, and rectangles (*transitions*) describe activities that could be executed in the process. Analysis of models focuses on *markings* over places and what transitions can be fired (somewhat rudely) to denote an activity being executed. The benefit of using a graph-based approach is that constraints between states and individual states of the process can be explored through a sequence of markings without an explicit denotation of all constraints.

Having a good understanding of the control-flow perspective is critical, but this perspective does not solely determine the behaviour of processes. By extending process models and their semantics, referred to as *data-aware models*, investigations and analyses can consider beyond the control-flow of the process. Explicitly, data-aware models refer to a class of models where

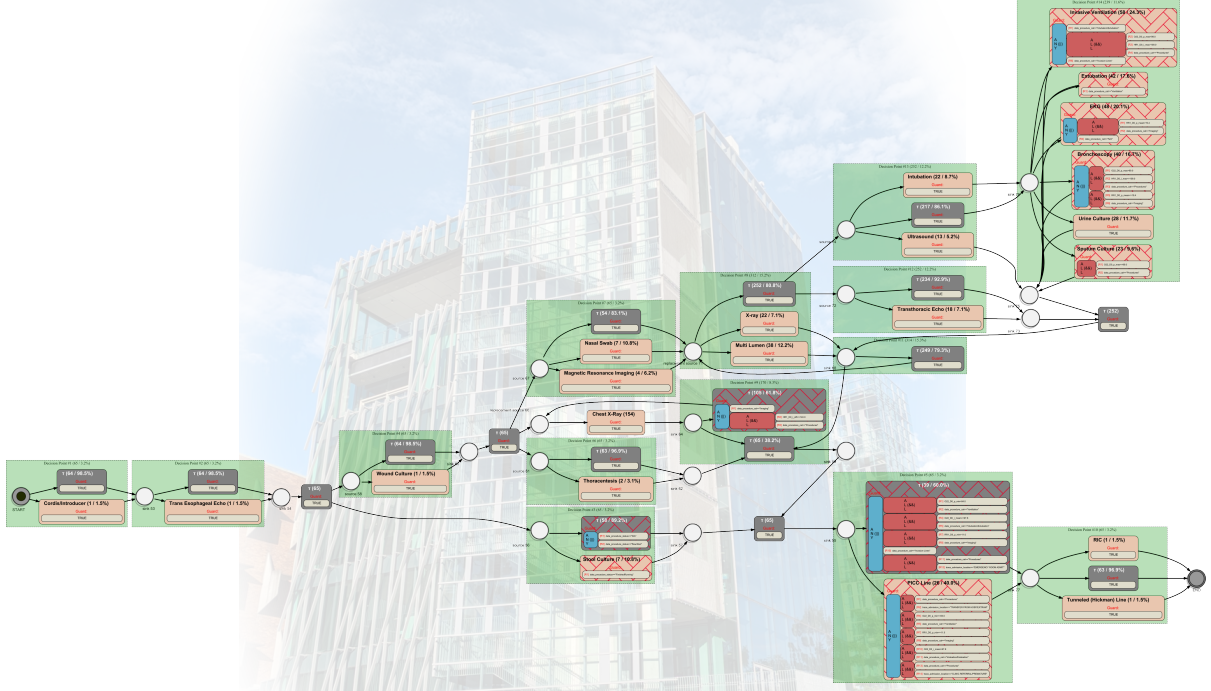


Figure 1.1: An example of a data-aware process model, where transitions are annotated with guards. The derivation of this data-aware model is explored in Section 3.6.1.

transitions have been extended to include additional parameters that influence their firing, e.g. transitions can be extended to consider data attributes to determine if they *should* have fired. Models can be extended into data-aware models in several ways. Activities in models can be extended to explicitly define what data attributes are generated by the execution of the activity. *Guards* or Boolean expressions can be included to extend the model’s semantics denoting that they must hold before the activity can be executed, as seen in Figure 1.1. Other approaches instead introduce a weight to each activity to denote how likely the execution would be in a state of the process. This thesis explores data-aware models to derive insights and rediscover how actors determine the next activity to execute at choices.

1.2 Process Mining: A Discipline for Studying Processes

Process mining is a research discipline that focuses on understanding and clarifying the operational behaviour of processes through the system use of historical recordings [2], [28], and this thesis contributes to its advancement. Process mining is rapidly maturing as a research community, with over 3,330 publications and an aggregate of 25,145 citations indexed by Scopus between 2018 and 2024¹. Most process mining techniques can broadly be categorised into process discovery, conformance checking, and process enhancement with this thesis focusing on

¹ The query was conducted on 31.07.2024 with the following search: *TITLE-ABS-KEY ("Process Mining" OR "process mining") AND PUBYEAR > 2017 AND PUBYEAR < 2024 AND (LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "ar") OR LIMIT-TO (DOCTYPE , "ch"))*.

conformance checking and process enhancement.

Process discovery techniques [29] exploit event logs to recreate the states of processes through a modelling formalism resulting in a process model. A process modelling formalism ideally has an unequivocal meaning, a well-defined formal semantics with enough expressive power to describe the behaviour of the process, and should be comprehensible and executable [30]. A typical process discovery technique is an algorithm that takes an event log E and returns a process model M . Discovering a process model is challenging because it is an unsupervised problem, meaning the truth is often not known and detecting workflow behaviour patterns [31] requires reasoning over incomplete information. Furthermore, techniques should provide guarantees around the types of models produced [29] and seek the most straightforward and simple model (Occam's razor²). Thus, to simplify the discovery of a process model, many techniques will make assumptions about the nature of the process [4], [32]–[35]. When applying analysis techniques that build from a discovered process model, one must be aware of these assumptions, as they may extend to the formalism used. For instance, some discovery techniques assume actions only occur in one state of the process [32], [36], [37], others assume that the process is without behaviour constructs and everything is choice [38]–[40] including halting. This thesis assumes the challenging task of discovery has been completed for the control-flow of the process before applying the contributions within, and accepts that the given models may be imperfect or incomplete.

Conformance checking [41] techniques quantify the quality of a given process model against an event log, by considering the intersection of observed behaviour within the event log and the described behaviour of the process model. Typically, conformance checking techniques are algorithms taking an event log E and a process model M to yield a quantification Q , usually between 0 and 1. These quantifications can vary in their semantics [6], [42]–[45] but consider one of the following: how much observed behaviour exists in the process model or *recall*, how much described behaviour is observed in an event log or *precision*, how well the described behaviour generalises the process or *generalisation*, and how understandable the process model is or *simplicity*. However, not all conformance checking techniques fit within quantification. Verification techniques confirm that a process model has a property and falls within conformance checking, these are techniques that take a process model M and return a boolean B . Examples of verification are confirming that a model is sound, has an option to complete, contains no dead activities, terminates correctly, or belongs to the workflow net class [46]. This thesis focuses on quantification to understand if extending the semantics of a model matches the observed behaviour.

Process enhancement techniques replay traces over process models to derive diagnostic information about processes. These techniques focus on the many other perspectives of a process,

² Occam's razor is a problem-solving principle that recommends searching for explanations constructed with the smallest possible set of elements, or the sharpened modern interpretation: the simplest explanation is usually the best. Occam's razor is attributed to William of Ockham, a 14th-century English philosopher and theologian.

compared to the typical control-flow perspective taken for discovery and conformance checking. Examples of process enhancement include the resource executing activities and their workloads, the timing of activities, the data generated within an execution, or the decision making occurring between activities. Enhancement techniques are usually algorithms that take an event log E and a process model M to yield either an extended model M' or a visualisation V . The goal of an enhancement algorithm is to describe an additional aspect of the behaviour of processes. For instance, this thesis presents a visualisation type in Chapter 3 and an extension type in Chapter 4.

This thesis investigates the nature and behaviour of processes, where the contributions are motivated by perspectives beyond the control-flow of processes. These contributions focus on contextual data streams, referred to as *exogenous data*, that actors may use to determine the activity that should be executed next, or the *decision making*. The aim of the thesis is to advance process mining by investigating:

*mechanisms for identifying exogenous influences on decision making in processes, and
measuring the quality of process models enhanced with decision making.*

This thesis uses a variety of techniques from process discovery, conformance checking and process enhancement to support this investigation.

1.3 Decision Making in Processes

As a process is executed, actors may face choices between possible process activities, requiring decision making to determine which activity should occur next. A choice is a state in a process where many activities could be executed next, but executing one settles the choice. An actor may settle a choice in various ways: they may rely on their expertise; they may rely on endogenous data for insight; they may rely on the contextual information, or *exogenous data*, around the process to decide the next activity. However, investigating choices in processes requires understanding the structure of processes to determine the choices that exist. Therefore this thesis assumes that the control-flow of the processes can be identified from the historical recordings of the process. This assumption does not limit the generalisation of this study, as many techniques exist to discover a model. Instead, the assumption enables the separation between several types of choices.

Many types of choices can be presented within a process. Figure 1.2 presents four possible examples of the types of choices. For some states of a process, the only option is to execute one activity (Type A, Figure 1.2a) is presented. However, this research posits there is always the choice between conforming or deviating from the model at runtime. This aspect is important to consider as in the real-world, executions can, and will be imperfect. Typically, deviation is seen as a negative aspect and the decision making leading to deviation is overlooked in research, frequently being passed off as simply “noise”. Type B choices are the most common type and

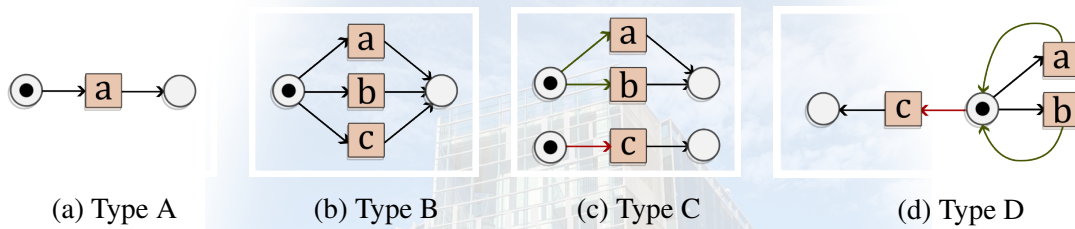


Figure 1.2: Examples of choices in processes; circles represent a state of a process; rectangles represent an activity; arcs denote how actions move the state of the process from one to another.

occur when a state allows for the execution of many activities. For instance, Figure 1.2b shows a state where activities *a*, *b*, or *c* could be executed. Type C choices occur when a state describes concurrently enabled choices. Within Figure 1.2c, a type B between *a* and *b* and a type A over *c* are concurrently enabled. Note that type C choices can easily be confused with Type B choices. Finally, type D choices are recurring choices, where executing some activities settle to the same state. Figure 1.2d describes a state where activities *a* and *b* settle to the same state, while executing activity *c* does not settle to the recurring state. In order to investigate these choices and rediscover the decision making for these choices, a process model is always required as an event log only records the settled activities and not the choices. The contributions of this thesis are not limited to any one choice type, but for the simplicity of illustrations often only type B or type C choices are presented.

This thesis aims to discover the decision making used to settle choices based on the endogenous and exogenous data. To do so, the thesis proposes techniques to derive and represent decision making through hard, soft, and visual annotations. Hard annotations are boolean expressions over data that stipulate whether an activity is allowed be executed. These annotations typically look for thresholds or exact values in the data to determine the following execution. In contrast, soft annotations focus on how likely it is to execute an activity based on data. These annotations focus on the previous behaviour observed in data rather than looking for an explicit threshold. Both soft and hard annotations allow us to investigate type B, C, and D choices in various ways, but struggle with type A choices. Type A choices are challenging as they present a state where there is no competition between activities. Visual annotations overcome the lack of competition by visualising behavioural differences within data to distinguish between process goals when activities are executed in all choice types. These annotations use visualisations over the process model, rather than extending their semantics, to clarify the process for owners and actors. By using these three types of annotations, the thesis can explore decision making in a multifaceted manner to clarify if exogenous influences are presence.

1.4 Exogenous Data

This thesis' novel contributions investigate whether processes are isolated from their contextual circumstances and clarifies how exogenous data may impact the behaviour of processes. *Ex-*

ogenous data describes the contextual surroundings of processes and are represented by time series consisting of measurements on a single concept. To differentiate between endogenous and exogenous streams, consider whether a trace and its event attributes can entirely determine a stream. If it is the case that the trace determines the stream, then the stream must be endogenous; otherwise, may be an exogenous stream.

For example, exogenous data for the road fines event log [25] could be the daily measurement of unpaid fines in the information system. These measurements can be computed from the concurrently running traces, as shown in Figure 1.3, where the number of unpaid fines (#fines) and the total amount of unpaid fines (\$fines) are shown as coloured time series. As no one trace fully determines the aggregation over all traces, these *inter-case* measurements from an event log are exogenous data. The endogenous moments in Figure 1.3 are shown as a vertical slice on the x-axis. The thesis explore the interplay between endogenous and exogenous data by investigating whether endogenous decisions are influenced by exogenous data. For example, does the number of unpaid fines affect when the batching of fines to debt collectors occurs?

Likewise, for the many event logs that were derived from MIMIC-III [23], [24], [47] in this thesis, periodic nurse observations of vital signs, such as heart rate, oxygen saturation, breaths per minute and respiratory rate, are used as exogenous data. The type of time series data considered throughout this thesis is univariate discrete time series. While data from stricter types, like equally distanced time series, can be used and may have efficiency benefits, this thesis aims to target the largest class of exogenous data sources as possible.

The impact of exogenous data on processes can be seen in several ways. Some behaviours in a process may never happen without external conditions; for example, panic buying at supermarkets often occurs when freak weather events occur. Other times, the structure of the process may change, e.g., an insurance company changing its claim handling after a natural disaster to ensure its customers get help quickly in their time of need. In some cases, the change in behaviour is more subtle. For instance, only portions of decision making may change if exogenous data is available, allowing agents to make more informed decisions about the next action. The context surrounding a process has a conditional impact and may not always be present; this makes investigating exogenous influences on processes challenging.

There are several considerations for interpreting exogenous data to avoid spurious findings. Similar to processes, analysis of time series data focuses on behaviour, where an individual measurement in exogenous data is of little interest. Consider the coloured time series in Figure 1.3a where the insight of these time series is that they continue to increase over time until a sudden decrease occurs, which repeats continuously. Another consideration for exogenous data is that visually interpreting time series data can be misleading, just like the behaviour in time series shown in Figure 1.3b. One series may be observed more often than another, making the visual comparison of between them harder. The expected mean for two stationary time series might differ, placing one above the other. The amplitude of the change between the two might differ, making the changes in one seem more extreme. After accounting for

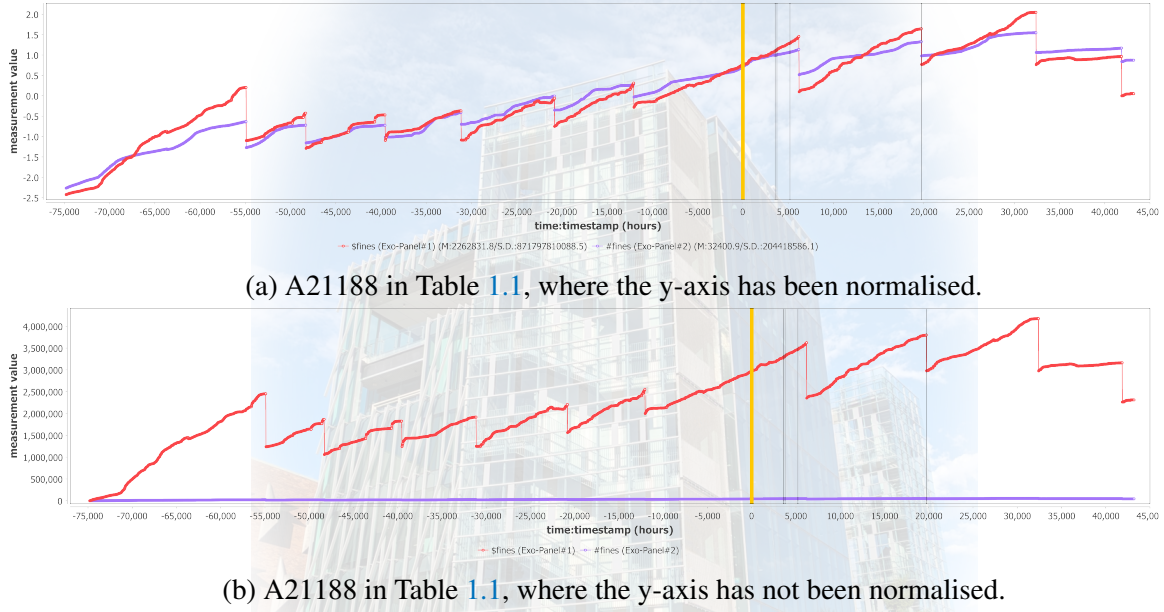


Figure 1.3: Examples of exogenous data and endogenous data from the road fines log. The exogenous data are represented as coloured line series; red for the total amount of unpaid fines, and purple for the number of unpaid fines. The endogenous events are represented as vertical lines and the first event is highlighted in yellow.

these properties of the time series, the visual inspection may show the relative behaviour is the similar as in the normalised time series shown in Figure 1.3a. Handling time series data and deriving insights from visual inspection based on them for users is challenging and requires care. Thus, techniques that preserve time series data as a first class citizens are required to investigate exogenous influences on processes.

Figure 1.3a highlights the intersection and interplay between endogenous and exogenous data. For the trace ‘A21188’ seen in Table 1.1, two sudden decreases in overall unpaid fines occurred, the latter of which correlates with the completed execution of ‘send for credit collection’ for the trace ‘A21188’. Proceeding investigations may consider if a rule be derived to determine why this execution went to credit collection and not others? Investigations may consider why the execution performed credit collection on the second decrease? That is, was it unlikely when the earlier decrease occurred and more likely when the latter occurred? Visualising the exogenous and endogenous data has highlighted an association, but does it hold for other executions, or generally across the event log? Deriving insights for these questions allows organisations to optimise their operations for contextual factors and this thesis proposes techniques to answer these questions.

1.5 Problem Statement

This section outlines the general problems that the contributions of the thesis serve to address. The thesis assumes historical recordings, or an *event log*, for the process is available, as well as

a formal description of the process, or a *control-flow process model*, is provided for analysis. Businesses are in dire need of insights from these event logs and process models to strategically plan for the future and outperform their competitors [2], [3], [48]–[51]. However, simply looking into insights into the control-flow of processes is unlikely to provide a complete picture of the problems faced by businesses [18], [52]–[55]. Thus, diagnostics into how and why executions of processes lead to (un)desirable outcomes are required to enhance the picture and clarify problems. Therefore, addressing Problem 1 enables businesses to clarify behaviour in their processes and achieve operational excellence.

Problem 1 *Given the executions of a process, a formal description of the process, and observations surrounding the process, how can algorithms identify the impact of the observations on the settlement of choices in executions of the process?*

Some businesses may be able to assume that their processes are driven by how they execute; thus, only using executions may be sufficient to ensure operational excellence [56]–[59]. For instance, an insurance company may be interested in ensuring that claims are evaluated solely based on the claim and the claimant. However, Problem 1 posits that in some cases, the enacted actions in a process do not provide enough information to derive how choices were settled by the executions [18], [52]–[54], such as patient admissions in hospitals [22], [60]–[62]. For example, when a patient is admitted to a ward, sensors or constant nurse examinations are conducted to measure the patient’s vital signs. These observations may externally influence actions in processes, like irregular heartbeats informing that an electrocardiogram should be the next action performed [23], [24]. The knowledge that observations can significantly impact processes empowers businesses and is critical for achieving operational excellence.

To understand how observations surrounding processes have affected business operations, the impact of the observations needs to be quantified. The impact of observations will be derived into the decision making based on the observations, producing a process model enhanced with decision making. Businesses need to understand how well process models enhanced with decision making reflect their operations [6], [41], [63]–[65] to avoid making strategic decisions based on ad-hoc measurements. However, a challenge in comparing process models enhanced with decision making is many different representations that could support the same behaviour [42], [45], [66]–[68], such as decision making based on observations or without observations. Therefore, addressing Problem 2 will enable businesses to effectively compare how decision making in processes changes based on the available data when processes execute.

Problem 2 *Given the executions of a process, and two formal descriptions of the process enhanced with some decision making, how can algorithms quantify and compare the quality of the formal descriptions?*

The thesis contributes advancements to address Problem 1 and Problem 2 enabling businesses to capture a more holistic picture of the problems they may face and achieve operational excellence.

1.6 Research Questions

In order to understand if executions of a process may be influenced by exogenous data, new techniques that focus on combining both endogenous and exogenous data are required. However, combining endogenous and exogenous data is not straightforward as often the temporal aspect of endogenous data in typical process mining work is overlooked. Furthermore, many research agendas within the process mining discipline [69]–[71] call for techniques that embrace a multi-perspective approach [55] to investigate processes holistically. As such, **the thesis investigates mechanisms for identifying exogenous influences on decision making in processes, and measuring the quality of data-aware process models.** To facilitate this investigation, two questions are proposed:

- (RQ1) *How may exogenous influences on decision making in processes be visualised and analysed?*
- (RQ2) *How can the quality of data-aware process models be measured and compared?*

The first research question (RQ1) addresses Problem 1 by requiring a foundational way to represent exogenous and endogenous data for process mining analyses. Secondly, a mechanism for making explicit the influence of exogenous data on process outcomes needs to be derived. Finally, as actions in processes are not equally likely and may be differently influenced by exogenous data, a mechanism needs to be derived to analyse this stochastic process behaviour. This leads to the following sub-questions:

- **RQ1.1:** How can exogenous data be represented alongside endogenous data in event logs?
- **RQ1.2:** How can exogenous influences on process outcomes be visualised and analysed?
- **RQ1.3:** How can exogenous influences on the stochastic behaviour of processes be analysed?

The second research question (RQ2) addresses Problem 2 by investigating how to compare process models enhanced with decision making, or data-aware models, and how to quantify their quality when their representation of decision making may differ. The representation may differ because one uses exogenous data and the other only uses endogenous data, but regardless it is critical to have a quantification method that is comparable for scientific endeavours. To enable the comparison of data-aware models, the thesis considers what are desirable properties [42], [43], [67], [68] to avoid ad-hoc measurements of data-aware models and how quantification methods, like recall and precision, adhere to desirable properties. This leads to the following sub-questions:

- **R2.1:** What are desirable properties for quantifying the quality of data-aware process models?

- **R2.2:** What are precision and recall measures that adhere to these desirable properties?

Together these questions serve to advance the discipline of process mining by clarifying the effect of exogenous influences on the decision making in processes.

1.7 Contributions

Figure 1.4 outlines the six key contributions established by this thesis in relation to RQ1 and RQ2. The first contribution is the xPM framework [23], [24] addressing RQ1.1 which combines endogenous and exogenous data into an exogenous annotated log. xPM is essential as it formally introduces exogenous data alongside traces as a first class citizen, allowing algorithms to be proposed that explore exogenous data for processes. A key benefit xPM is enabling domain knowledge (if available) to be used to aggregate exogenous data, and leaves the option to explore the annotated exogenous data for further process mining techniques. Chapter 3 showcases xPM in the healthcare domain using process enhancement techniques, such as decision mining which results in data-aware process models.

The second contribution (RQ1.2) is *Explorative Exogenous Series Analysis* (EESA), an automated visualisation technique which produces visual annotations over process models and investigates trends seen within exogenous data when process activities are executed. Furthermore, EESA can be used to compare trends of different process goals, where these goals can be defined in terms of endogenous or exogenous data. EESA was evaluated within the medical domain [24] using the MIMIC-III dataset [47], highlighting EESA's ability to derive an insight from vast number of seemingly scattered values for practitioners. EESA is presented alongside the xPM framework in Chapter 3, demonstrating the xPM's capacity to allow for techniques like EESA to investigate exogenous data for processes.

A process enhancement technique was developed to enhance control-flow models using stochastic process mining [72]–[74] to explore exogenous influences (RQ1.3). This technique uses a novel extension of stochastic labelled Petri nets, called Exo-SLPN, to investigate how exogenous data may influence the likelihoods of process activities through soft annotations over process models. Furthermore, a visualisation tool is proposed to identify the impact of the exogenous data on the stochastic setting, in the context of a given log. Exo-SLPNs and the visualisation tool were evaluated using publicly available datasets against existing techniques and existing conformance checking measures within stochastic process mining [75], [76] (Chapter 4).

Within Chapter 3 an evaluation of existing decision mining techniques was performed using existing data-aware conformance checking techniques. This evaluation noted difficulties in evaluating the quality of data-aware models when data is not generated by the executed process activities in the process model, namely exogenous data. This finding in our earlier work [23], [24] motivated RQ2 in Chapter 5, which addresses how to quantify data-aware process models.

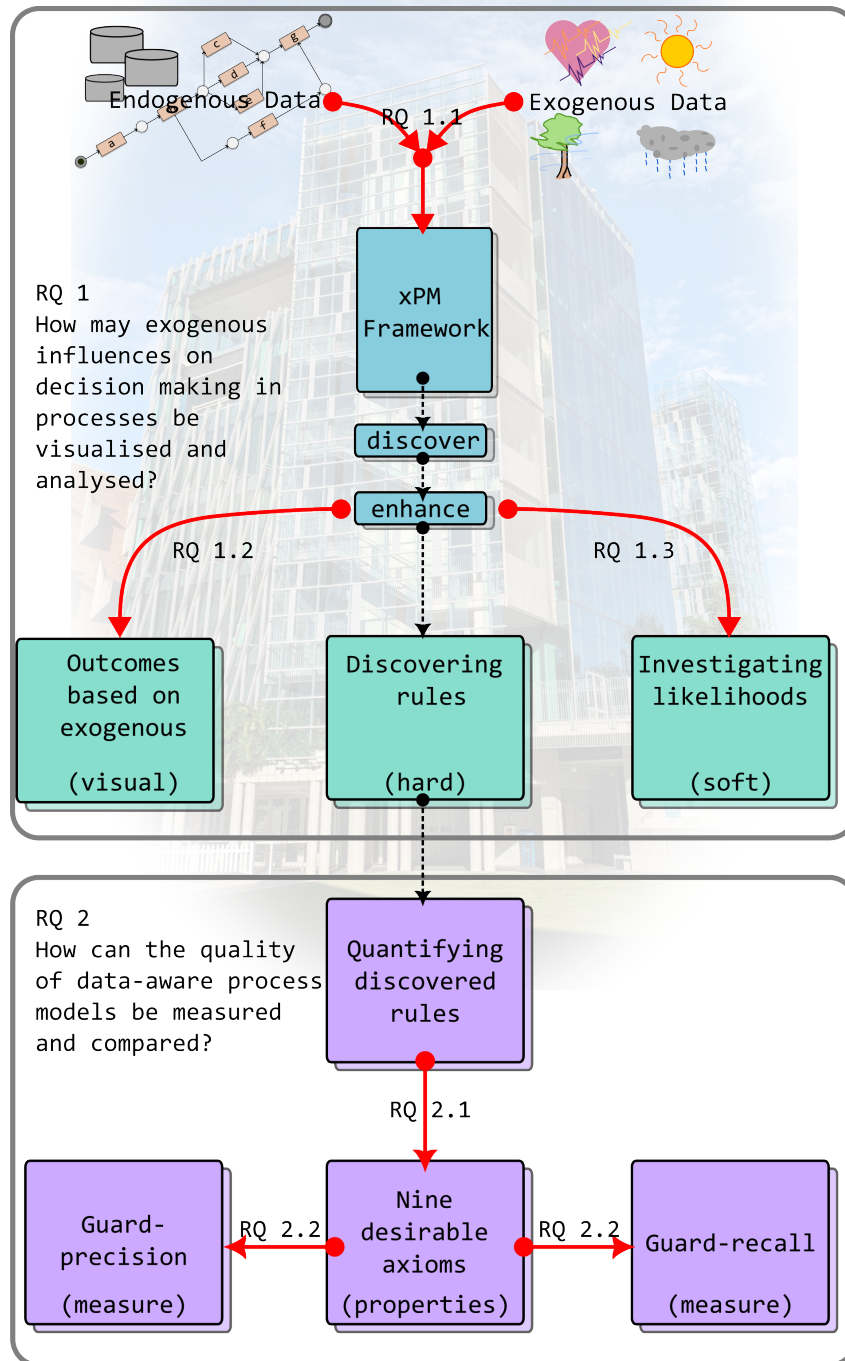


Figure 1.4: The six key contributions established by this thesis. Red arrows show where the contributions of this thesis exist, while black (dashed) arrows show where existing process mining techniques are used.

The contributions towards RQ2 are threefold. A motivation of these contributions was to follow the existing literature seeking a consensus on the desirable properties for conformance checking [42]–[44], [67], [68], [77]. As such the following contributions serve to build a consensus for quantifying data-aware models. First, a generalised formal theory for the quantification of any data-aware process model is presented. Secondly, nine quality axioms were formulated using the theory such that techniques can be assessed in terms of the axioms and their

desirable properties. Thirdly, two measures, guard-recall and guard-precision, are proposed to show that measures can adhere to proposed axioms. Chapter 5 presents the comparison of the proposed and existing measures using the proposed axioms as a yard-stick through synthetic counter-example scenarios.

1.7.1 Approach

This thesis sits between the fields of information systems (IS), computer science (CS) and mathematics so the algorithms can be precisely denoted. Earlier doctoral endeavours [30], [36], [55], [78]–[86] used a variety of methodologies, some [30], [36], [55], [78]–[81] following the methodology of CS entirely [87], others [82]–[86] combining methodologies from both CS and IS, often including Design Science Research [88]–[90] (DSR). However, this thesis takes a formal approach with ties to process mining and focuses on establishing notions of guarantees between inputs and outputs for algorithms [2], [36], [91], [92].

The thesis follows the work of Werf, Polyvyanyy, Wensveen, *et al.* [91], [92] to determine solution criteria for contributions. They present four stages of maturity [91], [92] that should be considered during design of the research, which are mostly sequentially executed but could span many projects or studies. While Werf, Polyvyanyy, Wensveen, *et al.* [91], [92] only claim their maturity methodology for process discovery, their insights can be generalised to the development of any process mining technique. In following this methodology, this thesis follows an incremental and iterative research design for algorithms that avoids *glittering* [91], [92], where a glittering algorithm is defined as one that produces high-quality output on a limited collection of input.

In terms of maturity stages, the thesis aims for the following maturity across the six contributions. All contributions should aim to have algorithms that are clearly and properly introduced, and thus, meet at least the first stage of maturity, by being well designed [92, Sec. 4.1]. Contributions that address RQ1 should consider validation in practical scenarios to meet the second stage of maturity [92, Sec. 4.2]. Contributions that address RQ2 should consider the relationship of quality of between the input and output of conformance checking algorithms to meet the third stage of maturity [92, Sec. 4.3]. The fourth maturity stage [92, Sec. 4.4], where algorithms should be tested in real-world situations to show if their assumptions and guarantees are effective, is outside the scope of the study.

1.8 Publications

The following four publications, across a variety of high-quality scholarly venues, represent the peer-reviewed work undertaken for this thesis:

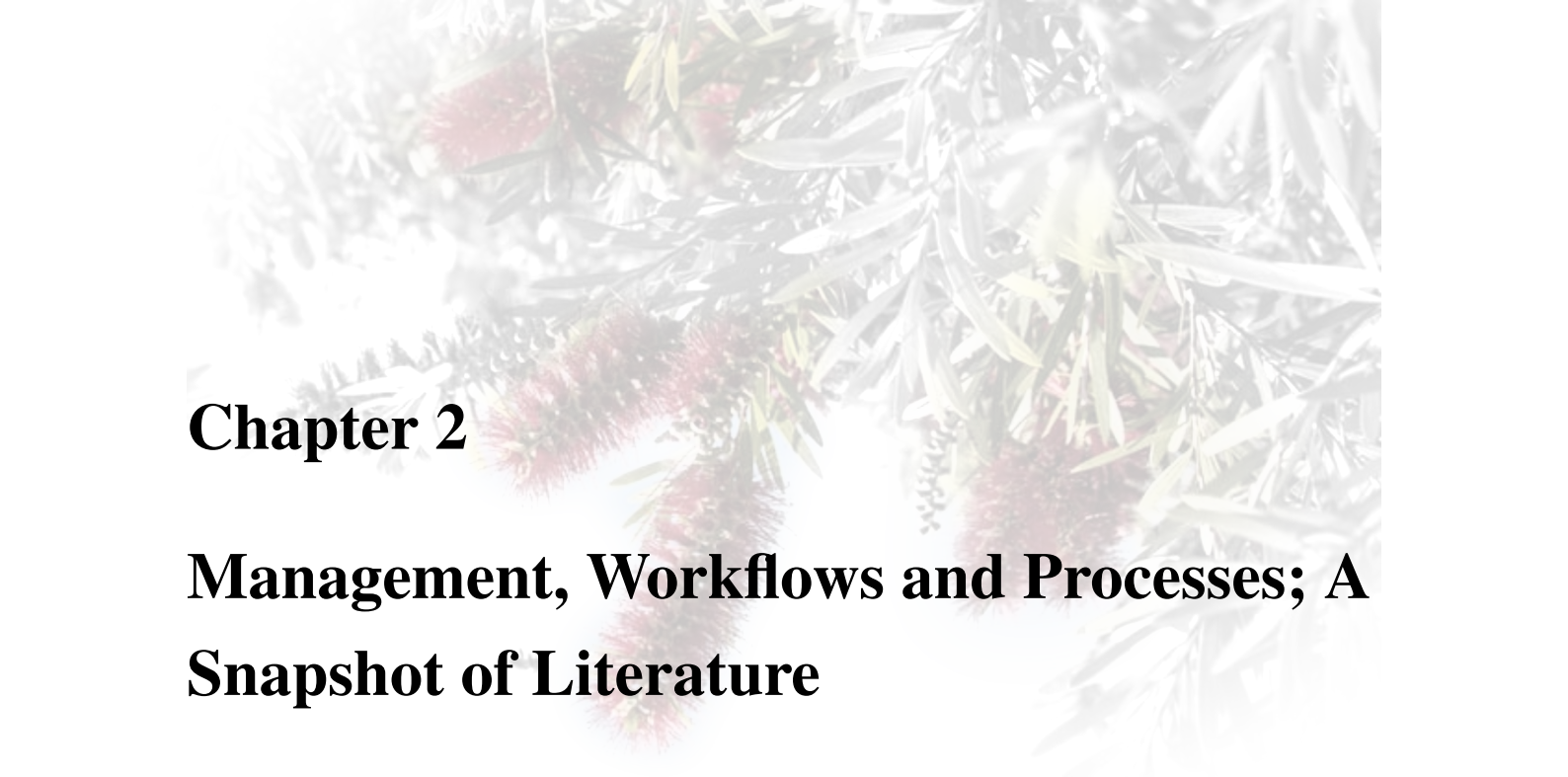
- **Adam Banham**, Sander J.J. Leemans, Moe T. Wynn, Robert Andrews. xPM: A Framework for Process Mining with Exogenous Data. In *International conference on Process*
-

Mining (ICPM), located at the Workshop on Event Data and Behavioral Analytics, Lecture Notes in Business Information Processing, vol 433, pages 85-97, 2021. This publication contributed to Chapter 3.

- **Adam Banham**, Sander J.J. Leemans, Moe T. Wynn, Robert Andrews, Kevin B. Laupland, Lucy Shinnars. xPM: Enhancing exogenous data visibility. *Artificial Intelligence in Medicine*, Volume 133, 102409, 2022. This publication contributed to Chapter 3.
- **Adam Banham**, Yannis Bertrand, Robert Andrews, Moe T. Wynn, Sander J.J. Leemans. Discovering the Influence of Exogenous Data on Decisions in Processes. In *Proceedings of Application and Theory of Petri Nets and Concurrency 2025*, Lecture Notes in Computer Science, 2025. This publication contributed to Chapter 4.
- **Adam Banham**, Arthur H. M. ter Hofstede, Sander J.J. Leemans, Felix Mannhardt, Robert Andrews, Moe T. Wynn. Comparing Conformance Checking for Decision Mining: An Axiomatic Approach. Accepted for publication in *IEEE Access* on 11.04.2024. This publication contributed to Chapter 5.

1.9 Structure of Thesis

The remaining of this thesis is structured as follows. Chapter 2 presents and analyses literature around process mining. Chapter 3 explores the difficulties of including exogenous data for process mining techniques. Chapter 4 presents the benefits of using soft annotations of decision making and how exogenous data can be investigated. Chapter 5 investigates how hard annotations of decision making can be fairly compared. Chapter 6 summaries the thesis and presents future steps. The thesis includes two appendices. Appendix A includes details to support the findings of Chapter 5. Appendix B presents the implementations of Chapter 3, Chapter 4, and Chapter 5.



Chapter 2

Management, Workflows and Processes; A Snapshot of Literature

We study history not to know the future but to widen our horizons, to understand that our present situation is neither natural nor inevitable, and that we consequently have many more possibilities before us than we imagine.
—Yuval Noah Harari; *Sapiens a brief history of humankind*.

The curious wretched might be considering if process mining has the capacity to derive bliss, does it come from a rigorous background, or is it a flashy trend within academia? These wretched needlessly worry as this chapter outlines the long history of studying processes, and presents the many benefits of using process mining for the curious. Starting with a historical perspective of the community, followed by a deeper analysis into process mining, broken down into three categories of techniques: process discovery, conformance checking and process enhancement. Lastly, the research gaps to be addressed with this thesis are presented for the wretched to digest.

2.1 The Science of Studying Processes

Before digging into process mining [2], [51], it is important to understand the journey of this emerging academic discipline. Process mining can be seen as the mixing of several disciplines focusing on information systems [48], workflow management [49], and business process management [3], [50]. To understand the merger of these disciplines, a high-level literature search was performed for each term in Scopus using a simple query string¹. No filtering was performed on the results. Then, the annual count of items published by discipline was plotted as a line series in Figure 2.1, showing it was not until 2000–2005 that the term *process mining*, was being

¹ Query string used for the following key terms, workflow management, business process management, process mining: TITLE-ABS-KEY (“[KEY]”) AND (SRCTYPE (j) OR SRCTYPE (p)) AND PUBYEAR < 2024

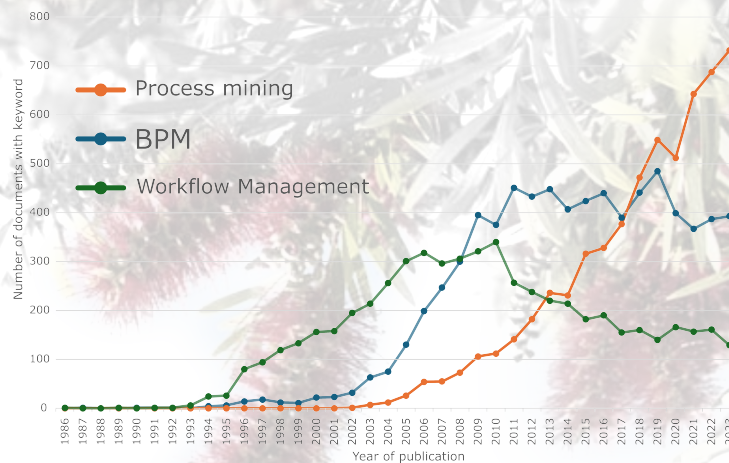


Figure 2.1: Process mining’s rise compared with business process management (BPM) and workflow management, based on published and indexed documents on Scopus 1986–2023².

actively used in publications. However, academics in these communities, regardless of name, have been studying business processes in information systems for as long as they have existed.

Information systems or process-aware information systems (PAIS) have been studied for some time [48], [49]. Dumas, van der Aalst, and ter Hofstede [48] define information systems as: “a software system that manages and executes operational processes involving people, applications, and/or information sources on the basis of process models”. The evolution of information systems has played an important role for those studying processes, and van der Aalst and van Hee [49] provide a historical summary of this evolution as being:

(1965-1975) started with hard-coded applications;

(1975-1985) focused on taking application data and placing them in databases;

(1985-1995) focused on ad-hoc generation of screens or user interfaces;

(1995-2005) focused on decomposing business processes from applications.

The goal of these developments was to obtain generic tools or interfaces for supporting business processes across all domains [48], [49], empowering owners and actors of processes to improve without needing to worry about their exact implementation in software. The abstraction of processes beyond their implementations has enabled process mining to focus on the behaviour of systems at a high level and be generalisable beyond a single process.

A survey of workflow mining [93], identified one of the earliest works on ‘process mining’ was presented by Agrawal, Gunopulos, and Leymann [94] in “*Mining Process Models from Workflow Logs*”, highlighting the potential of rediscovering the behaviour of processes from historical recordings by 1998. Furthermore, the survey [93] outlined four open problems: *how can processes be rediscovered, how should noise within logs be dealt with, how can the quality*

² Query string used for the following key terms, workflow management, business process management, process mining: TITLE-ABS-KEY (“[KEY]”) AND (SRCTYPE (j) OR SRCTYPE (p)) AND PUBYEAR < 2024

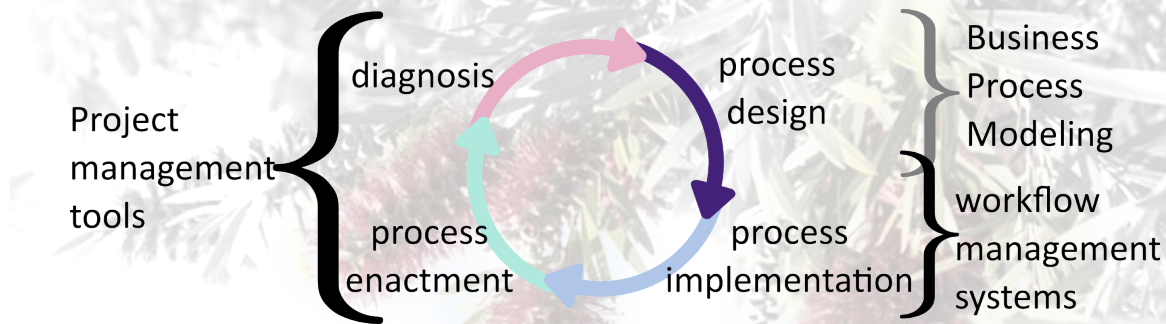
of a mined model be ascertained, and how can models with duplicated tasks be discovered? These questions continue to be of importance for academics advancing process mining, emphasising the overlap between workflow management and the process mining community. Another link between workflow management and process mining observed by the authors [93] was the call for an open and common XML format for historical recordings of processes which lead to adoption of Mining eXtensible Markup Language (MXML) [95] and the latter revisions to the eXtensible Event Stream (XES) format [96], [97] used in process mining.

A later survey of workflow management [98], highlighted the acceptance of process centered analysis as a critical factor for process management. In their survey [98], the authors mention the impact of workflow patterns [31] (over 2000 citations to date;Scopus), which describe the fundamental requirements that arise from modelling processes, where 15 commercial workflow systems were validated with these in mind. These patterns have been extended over time to cater for additional aspects of processes, advancing the control-flow [99], data within processes [100], and resources [100]. These patterns provide a mechanism to discuss and communicate the constraints between executed activities in processes with certainty, across many implementations and formalisms.

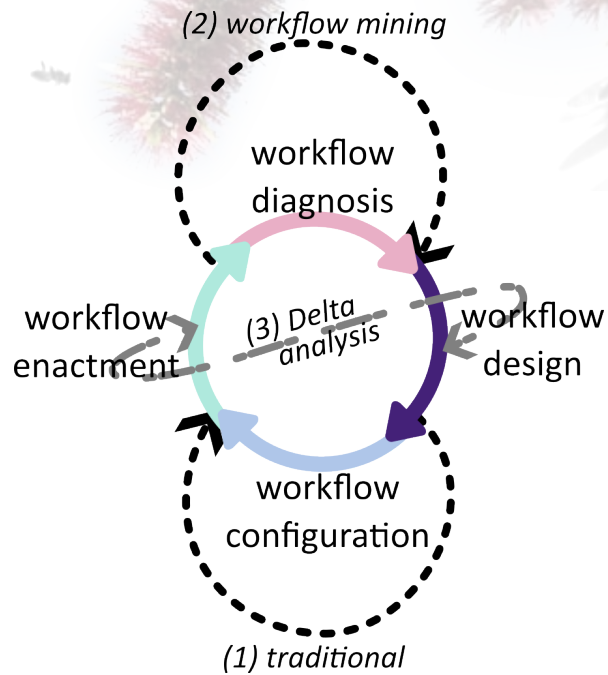
Emerging from workflow management at the same time as process mining, was the field of business process management (BPM). BPM focuses on processes as a means to organise and manage work within an organization [3]. Modelling within BPM often focuses on a descriptive model using Business Process Model and Notation (BPMN) to facilitate communication rather than providing execution semantics [98]. Thus, BPMN models enable a broad variety of constructs for communication, but the complex interactions between these may lead to serious execution semantic problems [101]. The benefit of this communication is that it enables for better process designs and organisation, which can lead to greater productivity of a process. Dumas, La Rosa, Mendling, *et al.* [3] highlight the famous example of Smith [102] as evidence that process design can lead to productivity gains. Smith describes the division of labour within the process of producing pins where 10 people could make upwards of 48,000 pins, but to say that one person makes 4,800 pins each day misrepresents the combination and division of the 10 people working together [102, Chapter 1]. Similarly, poor process design and organisation can negatively affect productivity of a process, which can be seen with the software development domain, as documented in the mythical-man-month by Brooks Jr. [103]. Brooks Jr. describes that unthoughtfully adding more resources, i.e. programmers, to the process of developing software many only further delay the process [103, p.16-19]. BPM experts therefore need to critically assess processes using process centered techniques, like workflow management and process mining, to empower organisations to make positive changes to processes in the real world.

As the fields of BPM, workflow management and process mining overlap in literature, this raises the question of what role does process mining play within this mixture of communities. To highlight the role of process mining, consider the several lifecycles of process management

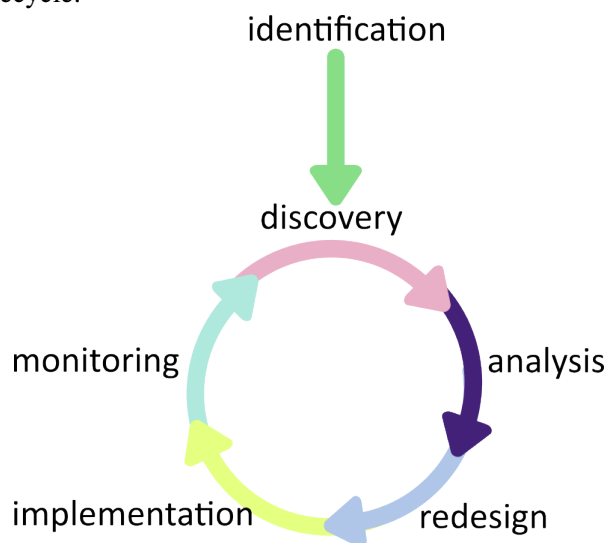
seen in Figure 2.2. The PIAS lifecycle [48] highlights how workflow systems are useful for implementation, process models are useful for design and management tools are useful for enactment and diagnosis of processes. Furthermore, workflow systems [93] were traditionally used after a design was configured and implemented, and workflow mining used recordings of the workflow to help redesign and to monitor exceptions at runtime through delta analysis. These lifecycles show that workflow management focuses on enactment or implementation and often is equated with automation. The difference between workflow management and process mining is the phases they cover in the BPM lifecycle [3], where the former focuses on implementation and monitoring, and the latter on discovery, analysis and redesign. Process mining focuses on studying the behaviour of processes through information systems to improve them, rather than how to enact, implement, or execute processes in information systems. This difference for process mining means that it is often retrospective in nature, but focuses on providing a precise description of the past for the planning of the future.



(a) Dumas, van der Aalst, and ter Hofstede [48] presented the PAIS lifecycle.



(b) van der Aalst, Dongen, Herbst, *et al.* [93] presented the Workflow lifecycle.



(c) Dumas, La Rosa, Mendling, *et al.* [3] presented the BPM lifecycle.

Figure 2.2: Lifecycles developed for the management of processes.



Figure 2.3: Common phrases used as keywords or within abstracts from process mining documents extracted in Figure 2.1.

2.2 Process Mining: A Formal Study of Processes

Academics in the field of process mining are motivated to provide clarity around the behaviour of processes, empowering their owners and actors to make meaningful changes for operational improvement [1]. In order to provide clarity, process mining assumes that the enactments of activities within processes were observed and stored in some information system [2]. The benefits from this clarity are multifaceted, as seen in Figure 2.3, whereas challenges of processes can be identified [104]–[109], compliance with company policies can be checked [41], [110], and run-time insights on bottlenecks slowing process executions can be discovered [14], [111], [112]. Process mining enables the study of complex behaviour and provides concise and impactful insights, driving organisations into the future.

Process mining might be seen as a strict subset of data science, but this is not the case. Process mining has been described as the bridge between process science and data science [2], as shown in Figure 2.4, where techniques are combined from both sides to study the world using a process-centred viewpoint. To highlight the differences between process mining and data science consider a snippet of the famous iris flower dataset shown in Table 2.1. This dataset was used by Fisher [113] to find a linear function between measurements of flower species. The iris dataset contains independent observations of two species from the same day and colony, and a third set of observations for another species from a different colony, where each observation denotes a class, the species of the bud. When using an event log such independent observations are unlikely and (ideally) all observations belong to the same class, the target process.

To contrast with the iris dataset, consider the event log shown in Table 2.2, where events are recorded states within an execution of a process and clearly previous states may influence proceeding. Likewise, traces may not be independent from each other as they can execute concurrently and share resources between these executions. The event log is our flower bud in this comparison, i.e. the whole log represents a single observation as it is a sample of the behaviour for the target process. Also, the collection of measurements for the iris dataset could have continued from 1936 to today, ever enlarging the dataset. However, the same cannot be

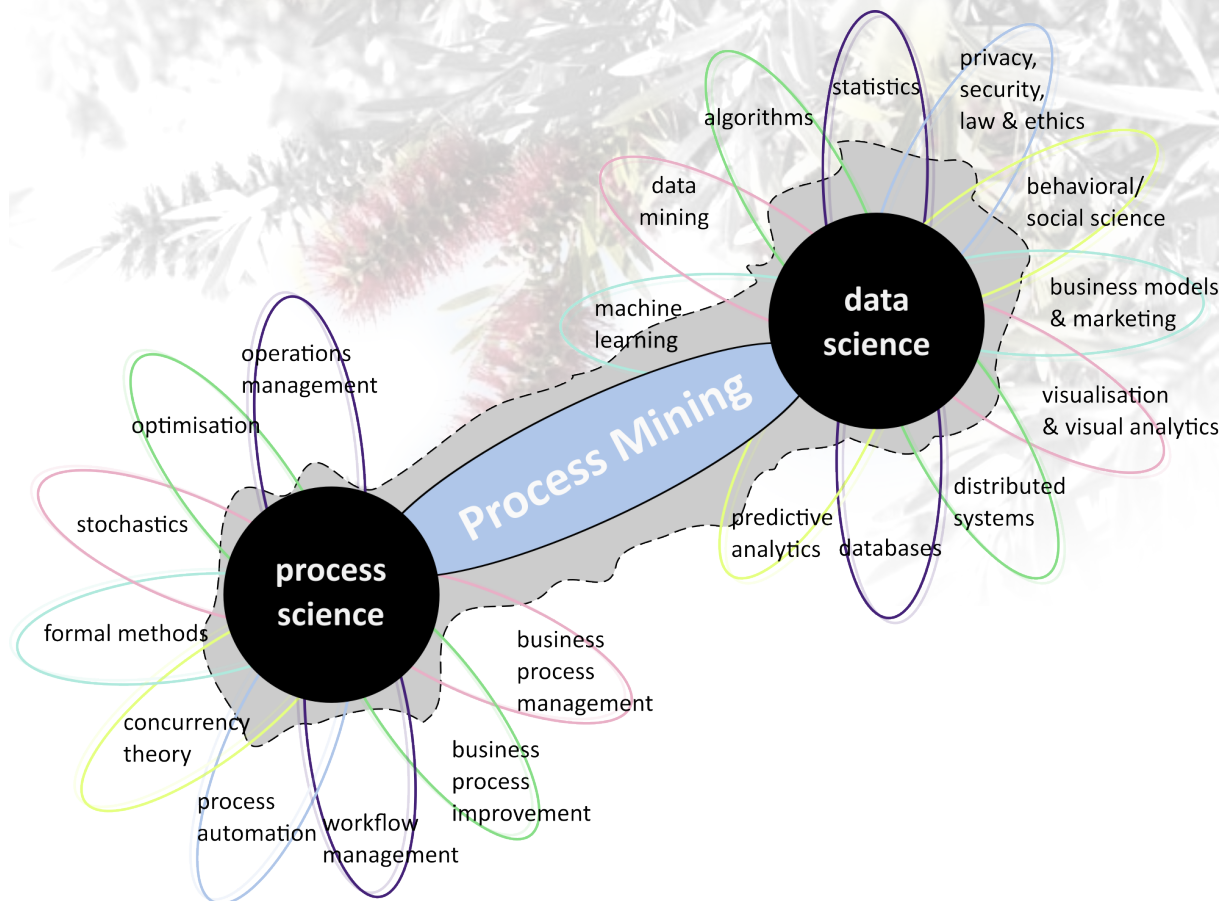


Figure 2.4: van der Aalst [2] presents process mining as the bridge between the fields of process science and data science. Reproduced with permission from Springer Nature.

said for processes, as there is no guarantee that the trigger of a process ever happens again. The point of this comparison is to highlight that event logs may appear tabular, as seen in Table 2.2, but they are not independent rows of observations and techniques must carefully handle event logs to derive insights [2], [51].

While an event log is a finite sample, say 5/50/500 or even billions of traces, it is critical to organisations that they can still derive insight from these logs to improve and continue to be competitive. The impacts of process mining in reality can be seen in many case studies [8]–[10], [114], [115], as in the domain of healthcare, to highlight areas to be improved to ensure effective and efficient patient triage and patient care [114], [116]. In a survey of business process improvement cases [8], four studies demonstrated the advantages of process mining, where a shared commonality was that process visualisations were a crucial factor for their success [116]–[119]. Reinkemeyer [9] also reported the tangible impacts from 12 case studies using process mining as being: reduction in manual activities, improving customer satisfaction, identifying root causes of inefficiencies, and transparency of as-is processes. However, another study [115] conducted interviews with process managers to highlight that challenges still exist, impeding the adoption of process mining initiatives. In response Mamudu, Bandara, Leemans, *et al.* [10] presents a process mining impacts framework, allowing managers to assess the po-

Table 2.1: A snippet of the Iris flower dataset used by Fisher [113], where each row is independent from another.

species	sepal length	sepal width	petal length	petal width
setosa	5.1	3.5	1.4	0.2
setosa	4.9	3.0	1.4	0.2
setosa	4.7	3.2	1.3	0.2
⋮	⋮	⋮	⋮	⋮
versicolor	7.0	3.2	4.7	1.4
versicolor	6.4	3.2	4.5	1.5
versicolor	6.9	3.1	4.9	1.5
⋮	⋮	⋮	⋮	⋮
virginica	6.3	3.3	6.0	2.5
virginica	5.8	2.7	5.1	1.9
virginica	7.1	3.0	5.9	2.1
⋮	⋮	⋮	⋮	⋮

tential of projects using process mining, addressing an impending challenge identified earlier by Grisold, Mendling, Otto, *et al.* [115]. Clearly, modern organisations stand to gain from the advancement of process mining to ensure their competitiveness for tomorrow.

Process mining is driven by a process-centered viewpoint of the world, however there is not one single viewpoint in this regard. The traditional viewpoint [2], limits the scope of a process to a single notion of a trace. As such, traces only contain events that relate to single process definition. For instance, traditional analysis focuses on the preparation of packages for delivery from a parcel warehouse. An alternative viewpoint called object-centric process mining [120]–[123] (OCPM) has recently emerged, where events can relate to many different notions of traces. A major breakthrough of OCPM is that all the processes within an organisation can be viewed as a single interconnected process and studied for operational excellence. For instance, object centric analysis focuses on improving the interconnected processes of: placing of an online order, its procurement and packing in a warehouse, how it is delivered from the warehouse to a last mile driver, and how it finally gets to your front door, only to be stolen by a porch pirate³. A drawback of this emerging viewpoint is that the methodologies and techniques lack maturity in comparison to the traditional viewpoint. Thus, this thesis adopts the traditional viewpoint of a single process definition [2] to focus on the decision making in processes.

³ The clear end user of a parcel delivery, i.e. a thief who takes packages left outside doors by couriers. A dangerous profession with many protective individuals setting “glitter bomb” traps to discourage theft. See Mark Rober on YouTube. YouTube is an American online video sharing platform owned by Google and Mark Rober is a content delivery expert on the platform.

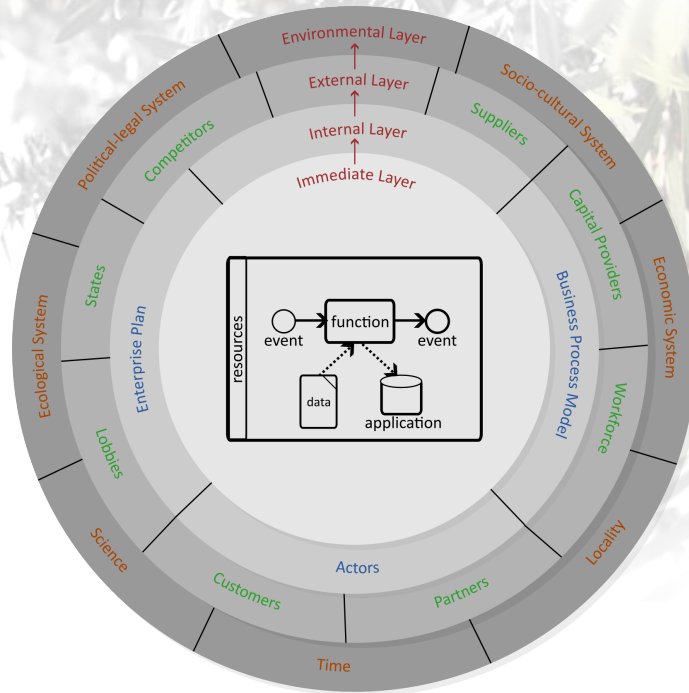
Table 2.2: An extract from our MIMIC III log [24], [47] in tabular like form.

trace	event	endogenous data process activity	lifecycle	timestamp
ICU0277104	1	chest x-ray	completed	12.05.2136@05:32
	2	chest x-ray	completed	13.05.2136@05:31
	3	48+ hours	completed	13.05.2136@20:53
	4	ct scan	completed	16.05.2136@17:04
	5	chest x-ray	completed	18.05.2136@01:31
	6	20 gauge	completed	18.05.2136@10:53
	7	20 gauge	completed	18.05.2136@11:59
	8	ct scan	completed	18.05.2136@12:41
	9	20 gauge	completed	18.05.2136@15:58
ICU0292604	10	18 gauge	completed	10.02.2177@22:03
	11	18 gauge	completed	10.02.2177@22:04
	12	invasive ventilation	completed	11.02.2177@00:00
	13	arterial line	completed	11.02.2177@22:00
	14	chest x-ray	completed	12.02.2177@11:05
	15	48+ hours	completed	12.02.2177@20:21
	16	20 gauge	completed	14.02.2177@12:47
	17	20 gauge	completed	14.02.2177@12:47
	18	chest x-ray	completed	16.02.2177@05:54
	19	20 gauge	completed	17.02.2177@06:09
	20	chest x-ray	completed	17.02.2177@06:17
	21	20 gauge	completed	27.02.2177@00:00
	22	ct scan	completed	14.03.2177@03:30
ICU0291927	23	48+ hours	completed	23.07.2139@19:14
	24	multi lumen	completed	05.08.2139@12:00
	25	invasive ventilation	completed	05.08.2139@12:00
	26	chest x-ray	completed	05.08.2139@13:45
⋮	⋮	⋮	⋮	⋮

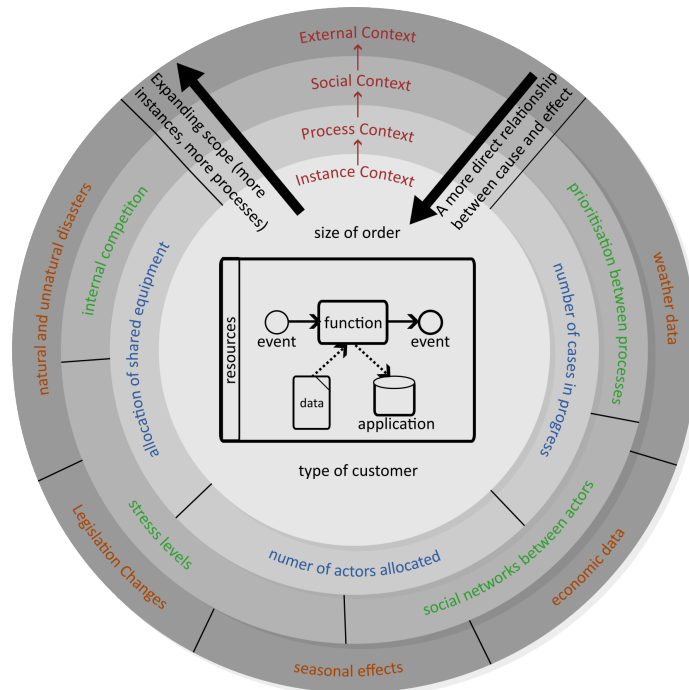
2.2.1 Contextual Data for Process Mining

An important consideration for this thesis is the data that should be considered as exogenous data for processes. Categorisation of data sources that could be used to describe processes has been discussed extensively [18], [52]–[54]. One view [52] claims that process flexibility is needed to handle extrinsic influences affecting organisations and a greater consideration of the context of processes is required. This view [52] is presented as an onion model (Figure 2.5a), where four layers are presented: an immediate layer, an internal layer, an external layer, and an environmental layer. An alternative view [53], recontextualised these layers to focus on the sources of data for processes and how these should impact process analysis, emphasising that these sources should not be abstracted away. The alternative view [53] is presented as a ‘process’ onion (Figure 2.5b), where process mining data sources are categorised four contexts: an instance context, a process context, a social context, and an external context. The differences between these onions in Figure 2.5 is that Figure 2.5a focuses more on the BPM viewpoint, while Figure 2.5b is more targeted at sourcing data for process mining efforts. To date, little formal work has been presented to explore the outer layers and data sources noted in Figure 2.5b for process mining.

A survey [54] took a broader look at the needs of analysts or process owners, showing the relationship between executions and contextual governance to be a growing need in the community. Following on, the work by vom Brocke, Baier, Schmiedel, *et al.* [18] developed



(a) The onion model presented by Rosemann, Recker, and Flender [52]. Reproduced with permission, © 2008 Inderscience.



(b) The 'process' onion model presented by van der Aalst and Dustdar [53]. Reproduced with permission, © 2012 IEEE.

Figure 2.5: Several onions that peel back the factors that may influence processes.

an in-depth framework for context-aware business process management to enable practitioners to enact planning that could adapt to contextual data. However, these frameworks are not seen as essential to process mining by recent reviewers of the field, and the contextual component remains an optional consideration during event data extraction [124]–[126]. As such, this thesis

introduces endogenous and exogenous data sources, such that they can be studied in combination to address the lacking of contextual governance for practitioners.

When basing analysis and techniques on data, it is always important to consider data quality issues. The seminal work by Wang and Strong [127] and Wand and Wang [128], outlines the notion of data quality and that data quality has the potential to impact the overall effectiveness of any organisation [128]. However, data quality issues in process mining are nuanced as the event log is the focus rather than tabular data. Studies [106], [129]–[131], have explored the nuances that occur when handling and extracting data from information systems for process mining efforts. A seminal piece of work exploring data quality for processes was Suriadi, Andrews, ter Hofstede, *et al.* [132] presenting 11 event log imperfection patterns as a means to describe common data quality issues within process mining at a high level of abstraction and generalisation. Several studies have operationalised these patterns to consider data quality in automated ways, with one study [133] presenting a quality-informed event log generation tool and another [84] presenting many techniques for crowd sourcing the repair of event logs for many data quality issues. Work in the direction of data quality has clearly established that is essential to consider data quality issues when applying process mining techniques. However, the identification and manifestations of data quality issues within exogenous data in the context of process mining remains future work.

2.2.2 Multi-Perspective Process Mining

Often work has focused on the control-flow of processes [53], i.e. the visible activities or states in processes. However, many aspects of processes exist beyond the control-flow and impact their performance. The benefits of including a variety of aspects were explored by Mannhardt [55], who introduced *multi-perspective* process mining which: (i) motivates using data attributes for distinguishing between noise and conditional process behaviour, (ii) considers if data attributes influence decision making within processes through Boolean expressions and decision trees, and (iii) studied how alignments [81] can be extended such that they balance both the control flow perspective and the data perspective. This thesis extends Mannhardt’s concepts [55] by investigating how exogenous data can be incorporated into process mining analysis, instead of being limited to endogenous data.

This thesis thereby emphasises a multi-perspective investigation of processes, where process executions generate data [55] and may monitor external data streams [13], [24] to change their decision making as the execution proceeds. The multi-perspective nature of the investigation focuses on how exogenous data influences executions of processes through changes in the decision making occurring at runtime [13], [19], [24], [54], and grounded within the work of [53] to explore the outer layers of the ‘process onion’. While, the findings from many studies [18], [53], [54] reinforce the motivation of process behaviour being influenced by the context of an execution, data-driven techniques are needed to enable analysts to investigate these influences. This gap can best be seen in large research direction consortiums [69]–[71], which have

continually emphasised the need for more multi-perspective techniques [55].

Some studies have focused on the enrichment of an event log with new types of data and explore multi-perspective process mining. Senderovich, Francescomarino, and Maggi [134] presented a framework for intra- and inter-trace predictive monitoring and introduce the notion of *bi-dimensional coding* to deal with intra- and inter-trace dependencies. Nguyen, Dumas, Rosa, *et al.* [135] presented an approach to compare process variants using the endogenous data available within an event log with a *differential perspective graph*. Similarly, the work of [136] presents an approach to characterise endogenous data, into *static*, *semi-dynamic*, and *dynamic* attributes to identify attributes of interest. Dees, Hompes, and van der Aalst [137] suggest that not all events within an event log are about the control flow, and some events are about the data flow of a process. They introduce the concept of *context events* to deal with the two types of events and show how distinguishing between the two can lead to discovering less complex models [137]. However, if this approach was applied to exogenous data would incorporate it into the control flow perspective instead of clarifying whether the context influences process execution.

The benefits of additional data attributes can be seen in the recent evolution of techniques that use such data. Leemans, Goel, and Zelst [138] presented a discovery algorithm that uses data attributes to create a hierarchical model to improve the simplicity of outcomes. Another approach proposed by Shrager, Gal, Schumacher, *et al.* [139] introduced a constraint operator for process trees notation, whereby data semantics can be expressed to explore distinctly separable behaviour of processes. While these techniques can create control flow sequences based on data attributes, no extensions have been proposed to use exogenous sources outside endogenous data within an event log.

2.2.3 Process Mining in Healthcare

The domain of healthcare offers a background rich for process mining with exogenous data. Mans, van der Aalst, Vanwersch, *et al.* [114] analysed 12 process mining applications within the healthcare domain, and identified questions frequently asked by medical professionals about healthcare processes. All these questions related to learning how a process was executed, i.e. the control-flow of processes. In a later literature review [60], the authors identified 74 papers related to applications of process mining in healthcare, confirming the focus on control-flow. That review also identified a need for better process visualisations and a reliance on domain experts to apply process mining techniques.

A subsequent systematic review [61], conducted a descriptive analysis of 55 articles, characterising the dimensions of process mining in the healthcare setting. The review noted that there was a need for techniques to capture more details about the context of a hospital intervention within their characterisation. Martin, Weerdt, Fernández-Llatas, *et al.* [22] presented ten recommendations from process mining researchers and healthcare experts to stimulate a more widespread use of process mining in healthcare. In addition, the authors encouraged efforts that

enhance the usability and understandability of techniques and a key recommendation was the need for more multi-perspective studies.

A review [62] presented a comprehensive review of 172 papers that overlap both process mining and healthcare domains between 2010 and 2020. The authors noted that the complexity of healthcare data and the heterogeneous nature of treatment pathways, has meant techniques still require fine-tuning for specific medical cases. Additionally, further work is needed to make process models more descriptive to be successfully used and understood by healthcare professionals. These reviews all highlight both a high level of involvement between the process mining community and the healthcare domain, and that little work exists for multi-perspective techniques to incorporate diverse data sources alongside process behaviour.

2.2.4 Process Mining with Internet of Things

A growing application domain for process mining is within the internet-of-things (IoT) which offers many opportunities for exogenous data to represent sensor data surrounding processes. In a research manifesto [71], the authors present 16 challenges that impede the advance of process mining with IoT, calling for further work to address these challenges. de Luzi, Leotta, Marrella, *et al.* [140] conducted a systemic review of 93 primary studies and identified most studies focused on addressing two IoT challenges [71] either: placing sensors in a process-aware way (C1) or closing the gap between event-based and process-based systems (C13). de Luzi, Leotta, Marrella, *et al.* [140] concluded that interplay between BPM and IoT in these studies is still in the development phase and many challenges may impede including IoT data with processes. Certainly IoT data can be classified as exogenous data for the purposes of this thesis, however the application of IoT data for processes can vary in several ways.

Data quality and interoperability issues in the context of IoT data have been studied with [141], describing how to plan for more resilient-aware processes in the BPM context using IoT data, reducing the impact of possible missing/unreliable data from IoT devices. Similarly, Bertrand, Belle, de Weerd, *et al.* [142] highlighted frequent data quality issues and derived patterns for identifying poor IoT data quality. Extensions of their work [142] include presenting a data format for IoT and process data [143] and validating the positive impact of IoT and process mining [144]. Both case studies conducted by Li, Joshi, Tam, *et al.* [145] and Gröger, Malburg, and Bergmann [146] highlight the real world impact of data quality issues when applying process mining over IoT data in various domains. These case studies show that considering data quality issues is important within the IoT context, but these issues as they relate to exogenous data from IoT sources remain future work.

A major focus with the application of IoT data for processes is using such data as endogenous data to discover a control-flow model. Mannhardt, de Leoni, Reijers, *et al.* [147] describes this type of process discovery as *guided process discovery* (see Figure 2.6); grouping low-level events into recognisable activities at a higher level of abstraction, which can be understood by stakeholders and used to answer typical process mining questions. However, this

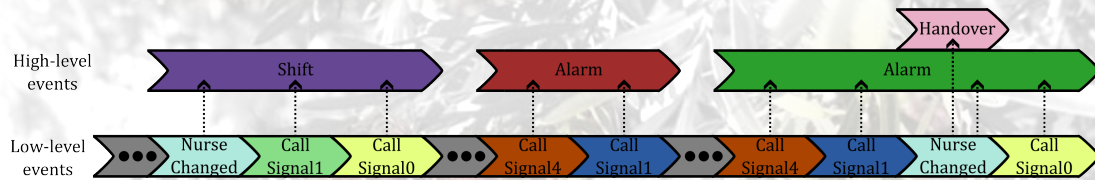


Figure 2.6: Mannhardt, de Leoni, Reijers, *et al.* [147] presents an example of guided process discovery, where low-level events are abstracted into higher level activities. Reproduced with permission © 2018 Elsevier.

definition overlooks that IoT data may not be linked to the notion of a trace and complicates applications of process discovery [148]–[151]. Nonetheless, guided process discovery has been applied across many non-typical domains such as: providing real time information from drones in disaster response [152]; smart home environments [150], [151], [153]; factory line simulation [154], [155]; smart manufacturing of chemical products [156]; logistic management at air terminals [157]; and, hands-free driving systems [149]. While work in this direction is interesting, this thesis assumes that exogenous data does not inform the control-flow of processes, but rather the decision making occurring when processes are executed.

Examples of using IoT data for the decision making in processes have been presented within literature. In one example Agostinelli, de Luzi, Canito, *et al.* [141] proposes how to annotate BPMN models during the design phase to improve the resilience of processes; where resilience is defined as decision making to handle missing or unreliable data from IoT devices. Another example presented by Beyel, Makke, Gusikhin, *et al.* [149] explored how acceleration and speed of IoT connected cars as well as traffic information may impact behaviour patterns, derived from guided process discovery, of drivers to reason about the changes in their behavior. Similarly, this thesis also investigates using IoT data as exogenous data to inform decision making within processes.

2.3 Process Discovery

Process discovery plays a pivotal role in process mining, where many different techniques have been developed and analysed for the purpose of understanding processes. The following section highlights differences within process modelling formalisms and process discovery techniques, establishing which combination thereof are ideal for studying decision making in processes.

2.3.1 Dependency Graphs

The earliest technique presented by Agrawal, Gunopulos, and Leymann [94] uses an event log to generate a dependency graph (see Figure 2.7), a directed graph of vertices V (process activities) and edges E where paths between vertices $u, v \in V$ only exist if v depends on u . Determining whether v depends on u is a complicated matter and curious readers can refer to definition 3 and definition 4 [94]. The algorithm presented does capture the dependencies in the executed

traces, but is not designed with the workflow patterns in mind [31], with the resulting graph not capturing choice, concurrency or free choice constructs as one would expect within BPM and process mining. Without choice, loops, or concurrency constructs in the semantics of resulting graph, it is challenging to investigate the decision making of processes.

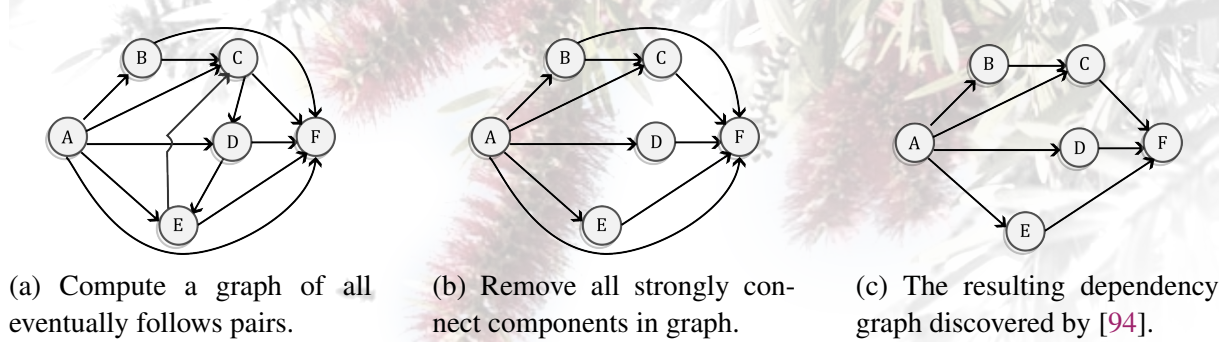


Figure 2.7: Agrawal, Gunopulos, and Leymann [94] presents how to discover a dependency graph using four executions $\{ABCF, ACDF, ADEF, AECF\}$. Reproduced with permission from Springer Nature.

2.3.2 Petri Nets

To describe the behaviour of processes, process discovery techniques will often use Petri nets as a modelling formalism. The reason that Petri nets are used so often is they cater for many of the behaviours described by the workflow patterns [31]. A Petri net is a directed graph consisting of two types of nodes, a set of places P and a set of transitions T , and a collection of flows F between places and transitions [2], [46], [158], [159]. Whether the collection of flows F is a set or a multi-set has further implications for the semantics of the graph, but does not change the execution of the graph. The advantage of Petri nets is that analysis can focus on a particular state of the net and determine the activities that could be executed. Petri nets are therefore ideal for investigating the decision making of processes as all choice types (in Section 1.3) can be identified and analysed.

Several process discovery techniques have been proposed that yield a Petri net from an event log. A family of techniques that revolve around Petri nets are derived from the α miner (alpha miner) presented by van der Aalst, Weijters, and Maruster [32], where the α miner is an algorithm that discovers a workflow net, a class of Petri nets, from an event log. In order to discover a net, the miner relies on a footprint matrix of directly follows relations to infer the required places and transitions. However, the technique struggles with activities with short loops, where a single activity can be repeated, and assumes that each activity relates to at most one transition. An extension presented in [37], the authors present $\alpha++$ miner which focuses on discovering non-free choice constructs from workflow logs. The $\alpha++$ miner has the capacity to find short loops and can discover free-choice constructs, or long term dependencies between activities that might otherwise be seen as a choice. The original α miner continues to be used even today through new extensions like the $\alpha+++$ miner, which uses plethora of discovery

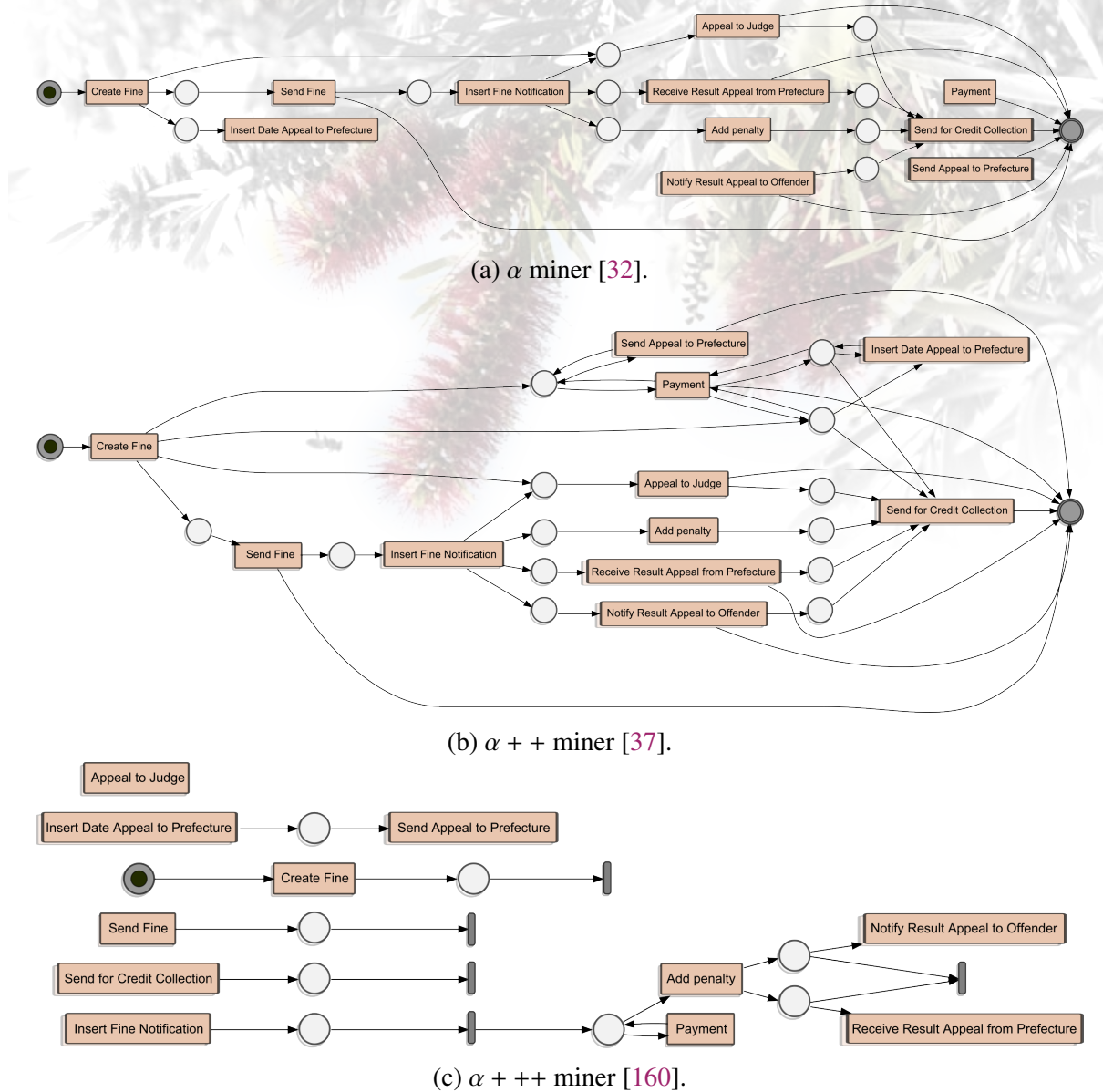


Figure 2.8: Several Petri nets discovered by α miners using the road fines log [25].

parameters to better handle real-world event logs [160]. However, not all real-world processes are nicely represented by connected graphs, in these cases Petri nets can still be used in a declarative manner [159].

While the family of α miners show it is indeed possible to discover Petri nets that represent processes, their algorithms do have drawbacks, starting with translating traces to a firing sequence of a Petri net, or computing an alignment (discussed later in Section 2.4.1). The drawback is that the discovered models may not be easy-sound [81, Def. 2.4.4, p24], and if the model is not easy-sound⁴ then alignments cannot be computed. Without the guarantee of an alignment being computable, many process enhancement techniques cannot be applied and this

⁴ The model does not have one single valid firing from its start to end state.

thesis extensively uses process enhancement techniques. Meaning that using techniques from the α miner family may result in models that are problematic and require manual intervention make them suitable for further analysis focusing on annotating these models.

2.3.3 Process Trees

Process trees enable the construction of Petri nets with guarantees around soundness, which ensures that alignments that can always be computed for any trace. Buijs [80] introduced the process tree notation as a means for process discovery techniques to guarantee sound process models and discover block-structured processes, which may be challenging to do when using a graph-based approach like Petri nets. A major benefit of this approach is that process trees can be easily translated to other notations and ensures the translation is sound [80, Sec. 3.4]. Process trees are recursive expressions using operators, over process activities, to express a language of executions. These recursive expressions are straightforwardly visualised using a nary tree. For instance, $\rightarrow (A, \times(B, C), D)$ is a process tree using the sequence operator $\rightarrow$ and the exclusive choice operator $\times$, both of which are visualised in Figure 2.9.

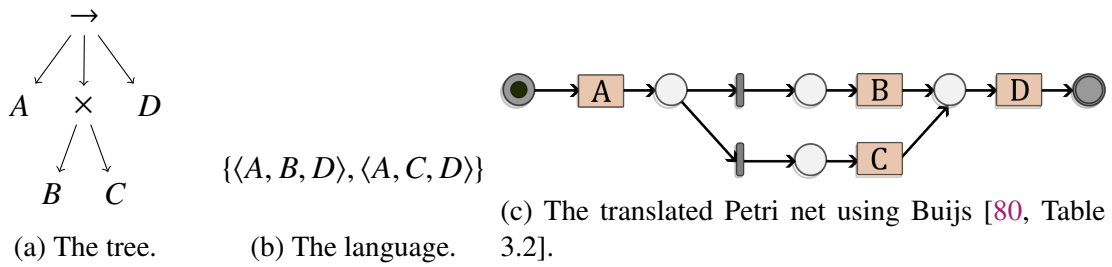


Figure 2.9: Breakdown of the process tree $\rightarrow (A, \times(B, C), D)$ into a sound Petri Net.

The genetic miner developed by Buijs [80], uses the genetic algorithm to discover a model with constructs that are challenging for other techniques, but at the cost of the runtime of the miner. Process trees are used by many process discovery techniques, but alternatives take a different approach to avoid the long runtime of the genetic miner [80]. The inductive framework presented in Leemans, Fahland, and van der Aalst [33] and Leemans [36] takes a divide and conquer approach, which addresses runtime performance issues but comes with caveats about the type of behaviour that can be discovered. Nonetheless, many extensions have used the inductive framework to explore processes; the β miner [33] focuses on rediscoverability of the process; the infrequent miner [36] focuses on handling infrequent and deviating behaviour; the approximate miner [161] focuses on finding models without parameter tuning; the directly follows miner [38] focuses on the most frequently followed activity pairs; the POWL miner [35] focuses on concurrency induced by partial orders; and the context-aware miner [139] focuses on separating behaviour based on contextual factors. These additions mean many behaviour aspects of processes can be catered for and will not effect the generalisation of further analysis based on their discovered models.

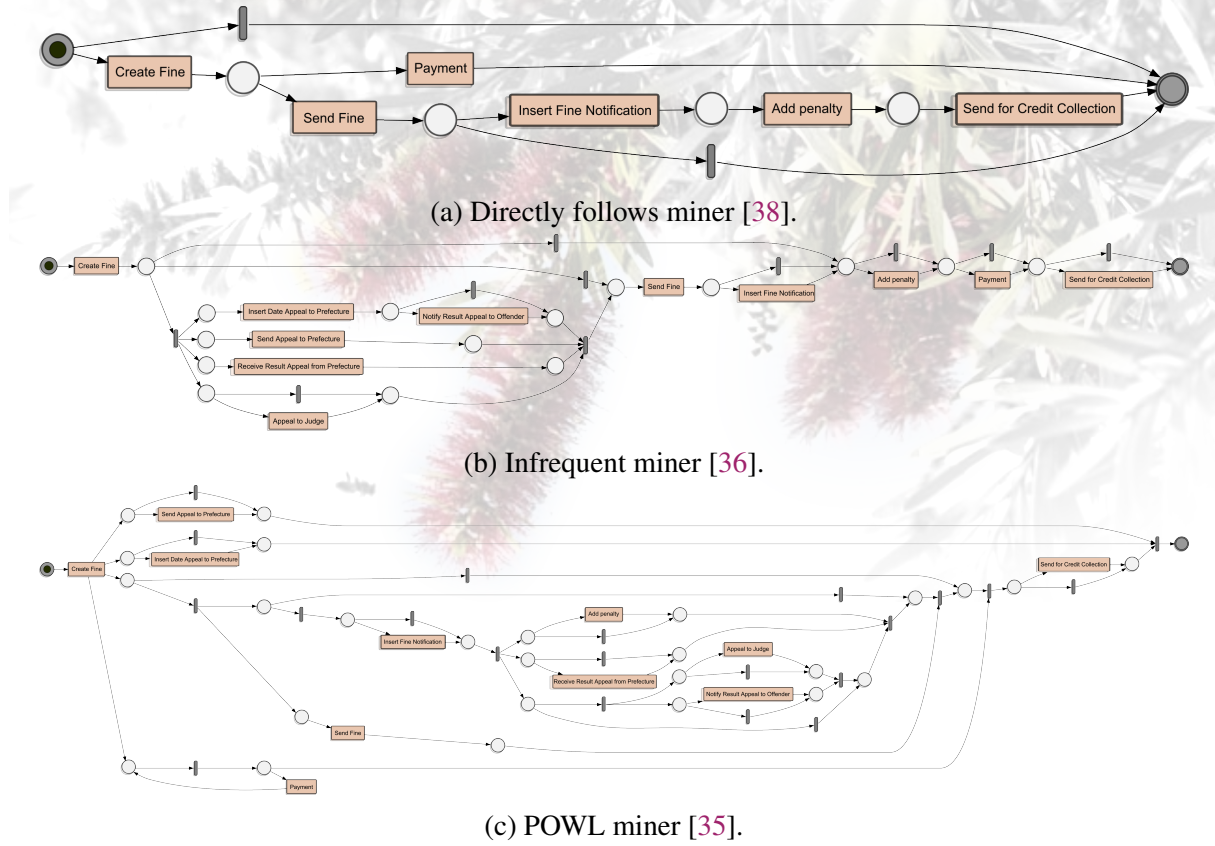


Figure 2.10: Translated Petri nets discovered by several miners based on the inductive framework [36] for the road fines log [25].

The guarantee of soundness from the inductive family of discovery techniques is extremely useful for process enhancement techniques as it ensures alignments can be computed. A comparison of the alpha miners (Figure 2.8) compared to the inductive miners (Figure 2.10), shows the inductive miners produce Petri nets that are connected and sound. While, the divide and conquer approach does have some limitations such as handling processes where activities can appear in several places, the runtime of approaches are usually within seconds. In the remainder of this thesis, inductive approaches are used so take advantage of the guarantee of soundness, rediscoverability of the given event log and the scalability of discovery for these approaches.

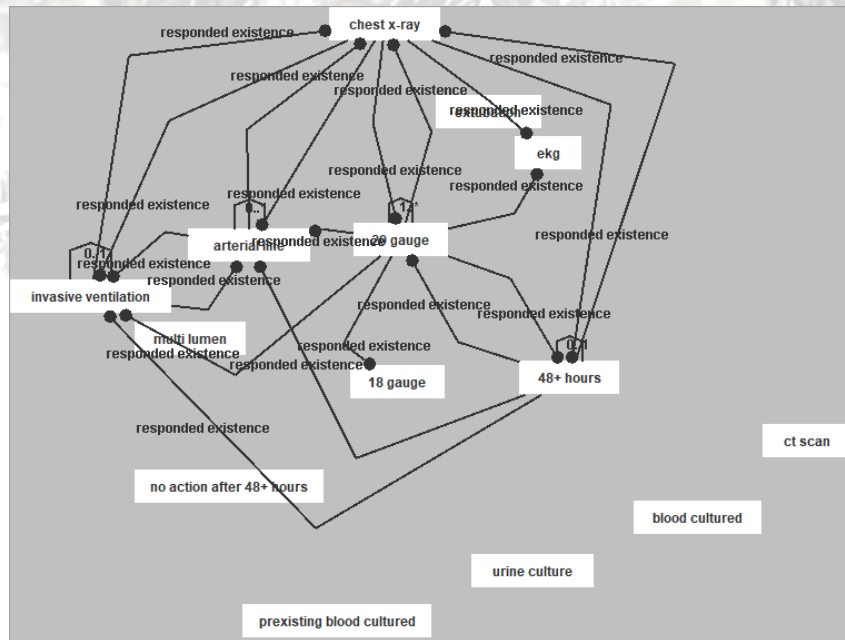


Figure 2.11: A DECLARE model [164] using the extracted MIMIC III log [23], [24] for this thesis.

2.3.4 DECLARE

The inductive [36] and alpha [32] families of techniques often focused on the imperative representation of models, strictly specifying how a process executes. However, a more flexible description for process execution exists in techniques that are focused on declarative modelling [162]. One such example of a declarative modelling approach is DECLARE, which uses a constraint-based system grounded in temporal logic [162]. The declarative modelling formalisms views processes as a single type D choice, meaning that behaviour constructs [93] are not considered. Viewing processes in this manner means these formalisms offer more analytic power to investigate processes that are less structured and may be “spaghetti” or “lasagna” like processes [20], [163].

Several discovery techniques have been proposed to discover DECLARE models from event logs. The declare miner proposed by Maggi, Bose, and van der Aalst [39] and Maggi, Ciccio, Francescomarino, *et al.* [165] finds control-flow constraints using DECLARE, while the MINERful miner [164], further improved the runtime performance of discovering a control-flow DECLARE model. A typical drawback of DECLARE models compared to process trees and Petri nets, is visualising DECLARE models can be challenging and difficult to understand without expertise. Alman, Ciccio, Haas, *et al.* [40] proposes a potential solution to interpreting DECLARE models by converting the declarative constraints into natural language. Other studies [166], [167] have used the conversion into natural language for constraints to incorporate multiple process perspectives. Schönig, Ciccio, Maggi, *et al.* [166] propose a framework for multi-perspective declarative process discovery, where different process perspectives of time, resource, or data could be used to generate a set of constraints. Following on Leno, Dumas,

Maggi, *et al.* [167] presented declarative process discovery for finding constraints based on correlated data attributes within constraint instances, explicitly focusing on the control-flow and data perspective of a process. While extensive work has been done with endogenous data within declarative process mining, most work has focused on control-flow aspects rather than contextual information and choices in processes are not directly considered.

2.4 Conformance Checking

Carmona, Dongen, Solti, *et al.* [41] describe conformance checking as the analysis of the relationship between the behaviour of a process (e.g. an event log) with a description of the process (e.g. a process model). The problem that conformance checking aims to address is quantifying if the description of the process model is accurate so that further analysis based on the description is not significantly hampered [41]. Ultimately conformance checking acts as the compass for process mining to ensure the derived insights are connected to the reality of the process as it is executed in the real-world.

Conformance checking has many sub-genres of techniques which produce explicit descriptions of the relationship between how a process is described and executed [6], [41], [46], [81], [168]. Quantification techniques [6] focus analysis on quantifying the overlap between the behaviour described and the observed behaviour. Verification techniques [46] consider if a description may be correct, sound, or have some other important behavioural property. Compliance techniques [168] check if a specific behavioural rule is or is not possible in a model and describes what executions may not be compliant w.r.t. that rule, such as the four-eye-principle [41, p. 184]. Traversal techniques [81] seek to translate process executions into a compliant and equivalent execution of the description of a process, enabling an execution to be *replayed* over a model. This thesis considers quantification and traversal techniques extensively as options to investigate the decision making in processes.

2.4.1 Traversing Models using Alignments

Traversal techniques within conformance checking play an important role within this thesis, enabling the investigate of decision making between the process activities by *replaying* the process execution over a model. Replaying a trace over a process model is a challenging task as the model can describe an infinite number of ways to replay the trace. Traversal techniques [79], [81], [169]–[173] handle infinite state space by proposing a search problem and finding an execution which limits the deviations between the trace and model.

One way to translate a trace to an execution of a model is token replay for Petri net formalisms proposed by Rozinat [79, Chapter 4]. The approach seeks to find a valid firing of a Petri net that replays a given trace by firing transitions and considering the state space enabled by the following marking of the net. Partial state space analysis of the current marking is used to

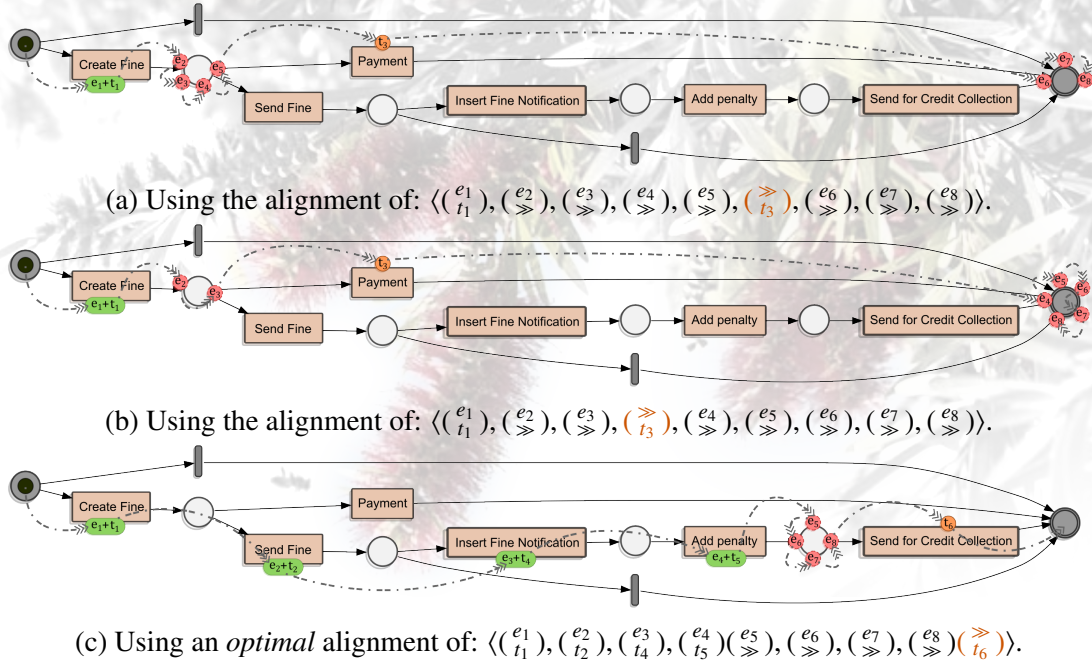


Figure 2.12: Examples of replaying the trace ‘N74775’ from Table 1.1 over Figure 2.10a using alignments. A synchronous move is highlighted in green; a model move is highlighted as orange; and, a log move is highlighted as red. Highlighted in each sub caption is the log move that can be arbitrarily moved with any cost change.

make the search problem tractable which means that to handle invisible or duplicate tasks within nets, the approach requires the use of local heuristics and these may lead to sub-optimal firing sequence at a global scale [79, Sec. 6.2.1]. Many researchers have worked to improve token replay approaches [174]–[176], but for the most part these approaches remain non-deterministic. However, these drawbacks are addressed through the seminal work by Adriansyah [81], who proposed *alignments* as an efficient solution to finding an *optimal* translation.

Alignments [81] are a key component to this thesis, enabling contributions be proposed for annotating transitions in Petri nets based on the diagnostic information generated from *replaying* traces. Adriansyah [81, Sec. 3.4] defined an alignment as pairwise comparison between the executed activities in a trace and a valid path through a process model, defined in the context of the trace. This comparison results in a sequence of moves⁵, where the sequence when projected can yield either the executed trace σ or the path through the model ρ . For example, the alignments in Figure 2.12a and Figure 2.12b, where both project to the same trace: $\sigma = \langle \text{Create Fine}, \text{Send Fine}, \text{Insert Fine Notification}, \text{Add Penalty}, \text{Insert Date Appeal To Prefecture}, \text{Send Appeal To Prefecture}, \text{Receive Result Appeal From Prefecture}, \text{Notify Result Appeal To Offender} \rangle$ and the same path through the model: $\rho = \langle \text{Create Fine}, \text{Payment} \rangle$. But simply defining an alignment between a trace and a path through process model, such as Figure 2.12a and Figure 2.12b, is usually insufficient for process analysis.

⁵ tuples of optional events and transitions, shown as (e_i^t) .

Typically, only optimal alignments [81] are considered for process analysis, as in Figure 2.12c, where optimal means the alignment has the minimal disagreement or *cost* w.r.t a cost model. A cost model defines an associated penalty for the types of moves seen within an alignment and enables users to guide the notion of what is optimal. However, in most cases, the default cost model is used as the model can dramatically change the nature of the search problem. The default cost model [81, Def. 3.4.6] applies the following cost to moves: an *synchronous* move ($\overset{e}{\rightarrow}$) usually costs zero, a *model* move ($\overset{\gg}{\rightarrow}$) usually costs one, and a *log* move ($\overset{e}{\gg}$) costs one. The costing of Figure 2.12b therefore is eight (one synchronous move, one model move, and seven log moves), while the costing of the optimal alignment in Figure 2.12c is five (four synchronous moves, one model move, and four log moves). However, there are aspects of this approach that are not deterministic in terms of the obtained optimal alignment, meaning that alternative paths through the model could be returned, traversing and settling different choices.

The non-deterministic aspects of optimal alignments revolve around the alignment returned and resolution of deviations that result in no change within the cost model. While alignments returned are always optimal, it does not mean all optimal alignments have the same projected path ρ , so best efforts are made to return the same arbitrary optimal alignment. However, it is somewhat arbitrary how model moves and log moves are ordered in alignments. For instance, Figure 2.12a and Figure 2.12a, have the same cost and the same projected path but the semantics in the data perspective are very different between these two alignments. In fact, there are many other optimal alignments like Figure 2.12c, where the ordering between ($\overset{\gg}{\rightarrow}$) and the log moves are vastly different. The effects of these aspects usually are not noticeable, but may impact process enhancement techniques making use of diagnostic information. Especially in the context of multi-perspective process mining as alignments are used to determine what endogenous data was available before firing transitions and informs how choices were settled for an execution.

2.4.2 Data-aware Conformance Checking

Conformance checking techniques [6], [41], [46], [81], [168] for the control-flow of process models have matured in process mining literature. But, not without some discussion of what these techniques should be quantifying in the general case [42], [67], [68], [177], detailed in Chapter 5. However, only a handful of approaches exist for the quantification of *data-aware* models, those extended to include an element reliant on data for its execution. The group of techniques that focus on analysis of data-aware models are referred to as *data-aware conformance checking*. Commonly these techniques use Petri nets with data (DPN) as their process modelling formalism. In terms of data-aware conformance checking, this thesis distinguishes between verification [63]–[65], [178] and the quantification of recall or precision notions [58], [169], [170], [173]. The focus of this thesis is on the latter to determine the quality of the data-aware models extended with exogenous data.

The difference between verification and quantification is that verification focuses on the properties of a model, while quantification is about the distance or disagreement between a log

and a model. Verification of properties for a data-aware model like reachability [63], relaxed data soundness [55], and data soundness [64], [65] are important for decision mining. In some cases, unsound data-aware models can be repaired into data sound versions [178], which is helpful as none of the typical decision mining techniques [55]–[57], [59] provide any guarantees about their outcomes. However, verification properties are not considered in this thesis, as they are typically focused on the model rather than log-model comparisons.

Data-aware conformance checking techniques are mostly related to either recall or precision notions [6], [41]. The first data-aware conformance checking technique was proposed by de Leoni and van der Aalst [169], using Integer Linear Programming (ILP) to compute an alignment between trace and model, which accounts for all perspectives. Alongside the alignment procedure proposed [169], the authors presented a conformance checking technique for recall, that considered the guards of a DPN. This technique applies a staged approach using the following three steps: (1) compute a control-flow alignment; (2) an ILP problem is solved for the optimal number of changes needed to make the trace compliant with the model's guards; (3) compute the recall using the harmonic-mean of (1) and (2). A downside of performing (1) and (2) in sequence is that a sub-optimal alignment could be returned by (1) if it describes an execution which is costlier than another, when both control-flow and guards are considered.

To address downsides of the staged approach [169], Mannhardt, de Leoni, Reijers, *et al.* [170] proposed an extended set of alignment moves and computes one holistic alignment. As such, the technique uses Mixed Integer Linear Programming (MILP) to solve for an alignment which included suitable write conditions for a given trace and DPN. The benefit of this approach was that each alignment move in each perspective of a process could be given a unique cost. However, the approach [170] needs many more MILP problems to be solved than the staged approach [169], which is time consuming, but always return the holistic optimal alignment rather than a sub-optimal one.

Mannhardt, de Leoni, Reijers, *et al.* [170] also presents a data-aware conformance checking technique for recall, which computes a ratio between the optimal alignment and the worst alignment for a given trace and a DPN. A downside of this approach is that both the cost-model and guards within the DPN effect the optimal and reference alignment, in turn defining a unique state-space for the resulting recall measurement and making comparisons between measurements challenging.

An alternative approach to data-aware alignment is proposed by Felli, Gianola, Montali, *et al.* [173], which introduced the ‘Computing Conformance Modulo Theories’ (CoCoMoT) framework to compute alignments in a similar vein to [169], [170]. The CoCoMoT framework uses Satisfiability Modulo Theories (SMT) and solves an SMT problem for the minimal number of edits to make a trace compliant with a given DPN. Note that this technique focuses exclusively on the alignment problem and, unlike the previous studies [169], [170], does not present a recall measure.

The CoCoMoT framework [173] performs the following four steps to compute alignments:

(1) first, the log is pre-processed to be a set of unique traces, (2) trace clustering is performed to find subsets of equivalence trace classes, (3) a representative is selected from each class, (4) a SMT problem is encoded and solved for each representative and then an alignment is decoded from the SMT solution. On completion these four steps, the decoded alignment can be used to denote the distance between the original trace and the alignment. The benefit of this approach was that only one problem per representative was needed, reducing the computation needed compared to the previous alignment approaches [169], [170].

The only data-aware conformance checking technique proposed for precision that this thesis is aware of, is the study presented by Mannhardt, de Leoni, Reijers, *et al.* [58]. A requirement of the precision technique [58] is that the log must be perfectly compliant with the given model including system behaviour (i.e. silent actions or silent transitions⁶). However, this requirement is overcome by applying an alignment procedure [170] on the given log. Then, for each state of the model, the technique computes the observed and possible behaviour for that state, after which, a ratio is computed between all observed and all possible behaviour seen at states of the model. A downside of this approach is that the original data in the log may be changed to be compliant to guards due to the alignment performed. For non-compliant logs, this downside may lead to non-intuitive precision values for the quantification of quality for data-aware models, making comparisons between outcomes challenging.

These existing data-aware conformance techniques [58], [169], [170], [173], focused on addressing the alignment problem for data-aware models, and to date, little has been presented concerning the desirable properties for the quantification of quality for data-aware models.

2.5 Process Enhancement

Process enhancement techniques typically focus on one or many different perspectives (not control-flow) of a process, to provide an integrated view of the process, using information recorded in an event log [2]. Zerbino, Stefanini, and Aloini [179] conducted a thematic analysis of 145 papers and identified a lack of maturity in techniques for supporting strategic decision making and encouraged exploring process enhancement. The call for more process enhancement techniques was identified in other systematic reviews [180]–[182]. The current maturity of process enhancement and its slow rise in comparison with process discovery and conformance checking is not unexpected. Often process enhancement combine techniques from discovery and conformance checking, where tools develop once sufficient maturity occurs within discovery and conformance checking.

Two types of process enhancement techniques, decision mining and stochastic process mining, fall within the sub-genre of *extension* within process enhancement [2], where a process

⁶ A silent transition is an action with no visible activity from the actor's perspective. Silent transitions are typically used for "routing" in discovered behaviour constructs.

model is *extended* with further semantics about a different perspective of the process. Decision mining techniques [56]–[59] extend the control-flow of a process model with guards, a function taking data and returning a Boolean, to determine if a transition should fire given the context of a partial execution, referred to as a *hard annotation*. Stochastic process mining [76], [86], [183], [184] extends the control-flow of a process model with weights, a function that returns a number, to determine how likely a transition is to fire given the context of a partial execution, referred to as a *soft annotation*. Both approaches enable the study of decision making within process executions and this thesis investigates the applications of these approaches to understand exogenous influences on processes.

2.5.1 Decision Mining

Decision mining was first presented by Rozinat and van der Aalst [56], where token replay was used to generate examples for classification problems and were solved using decision trees. The decision trees could then be used to generate a ‘human’ readable expression based on process attributes. These expressions were then attached to transitions in a Petri net to denote why a process activity would occur based on data. The next advancement in decision mining was proposed by de Leoni and van der Aalst [57], where the authors used alignments [81] to generate examples for the classification problems. This approach had greater flexibility than token replay and could handle unfitting event logs and process models. Later work presented by Mannhardt, de Leoni, Reijers, *et al.* [58] proposed a different strategy for extracting expressions, where ‘overlapping’ rules could be constructed. Lastly, Scheibel and Rinderle-Ma [59] proposed a variety of preprocessing steps to include time series data in decision mining techniques.

The process modelling formalism that decision mining techniques use is Petri nets with Data (DPN) [55], [64], [185] (see Section 3.3 for a formal definition). For instance, the decision mining techniques [57], [58] can be easily applied to process models, like the models in Figure 2.10b and paired with the road fines log [25], resulting in the data-aware models in Figure 2.13. But, the visualisations of DPNs in this thesis focus on the guards being annotated onto transitions of the Petri net and avoid describing the messy underbelly of write and read constraints that formally complete a DPN [64, Def. 3]. Write and read constraints introduced by decision mining techniques are often afterthoughts and more or less, syntactic sugar for the user. Write and read constraints are important for verification of DPNs [63]–[65], [178], but when investigating exogenous data, these constraints are not directly applicable. Thus, the contributions of the thesis abstract from write and read constraints and focus on the annotated guards in DPNs.

Studies have used decision mining to understand decision-making within processes. Dumas, La Rosa, Mendling, *et al.* [3] considered patient triage in an emergency department using decision mining to determine age groups for triage behaviour and F_1 score was used to evaluate effectiveness locally. Similarly in Mannhardt and Blinde [21] considered a Dutch emergency department and worked with healthcare experts to understand patient readmission using deci-

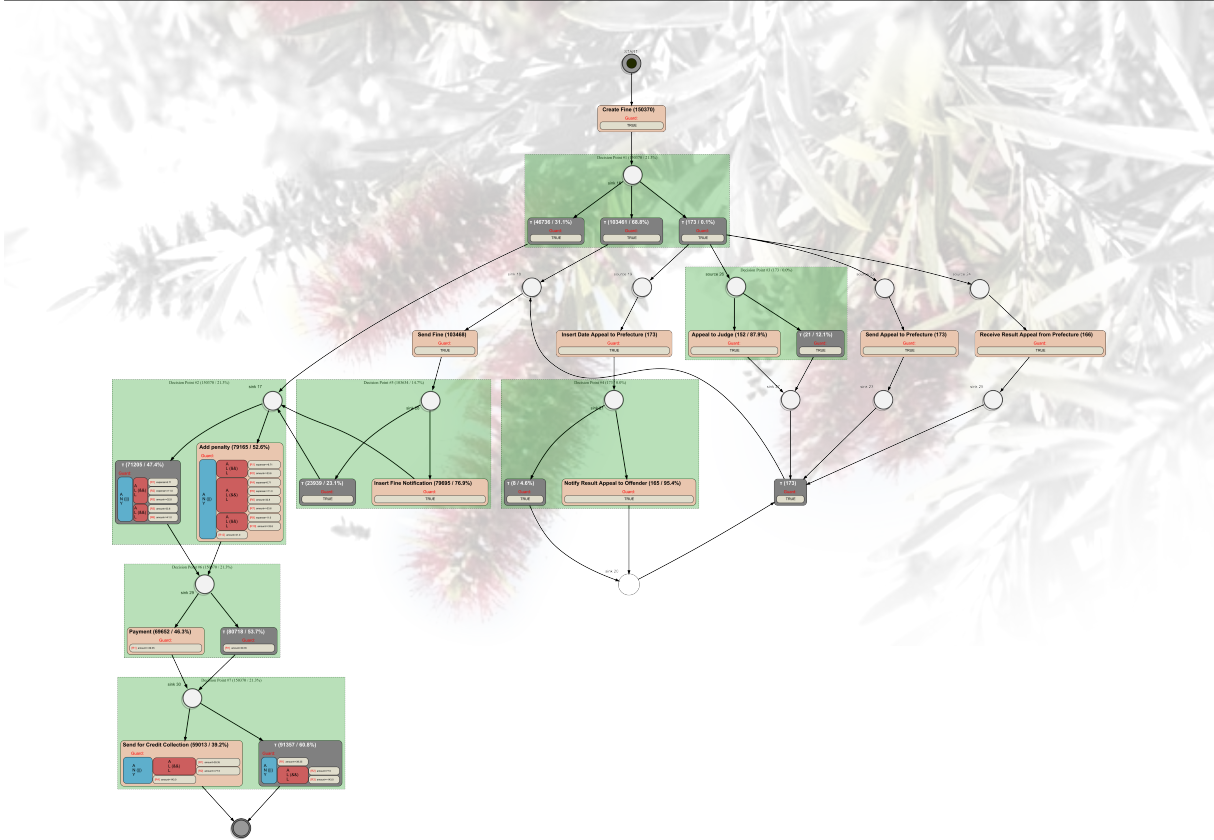
sion mining, where a data-aware conformance measure [170] was used to evaluate effectiveness. In contrast, Bolt, de Leoni, and van der Aalst [186] used decision mining to identify business rules between process variants, comparing the similarity of generated expressions. Likewise, Wais and Rinderle-Ma [187] performed a user evaluation to explore criteria for decision mining, focusing on understandability or consistency of outcomes for end users.

Jouck, de Leoni, and Depaire [188] noted the difficulty of comparing decision mining techniques, as many tailored their evaluation for the case and therefore presented a framework to evaluate and compare techniques in an artificial setting using simulation. The difficulty in comparing outcomes was also noted by work of this thesis [23], which integrated exogenous data into event logs, greatly expanding the state-space used by data-aware conformance checking techniques [170], [189]. Additionally, an ad-hoc measure was introduced by the thesis [23] (see Section 3.6.1) to investigate the number and support for the discovered guards alongside existing data-aware conformance checking. As of yet, there is yet no consensus on how to evaluate decision mining techniques or the number of criteria that should be considered.

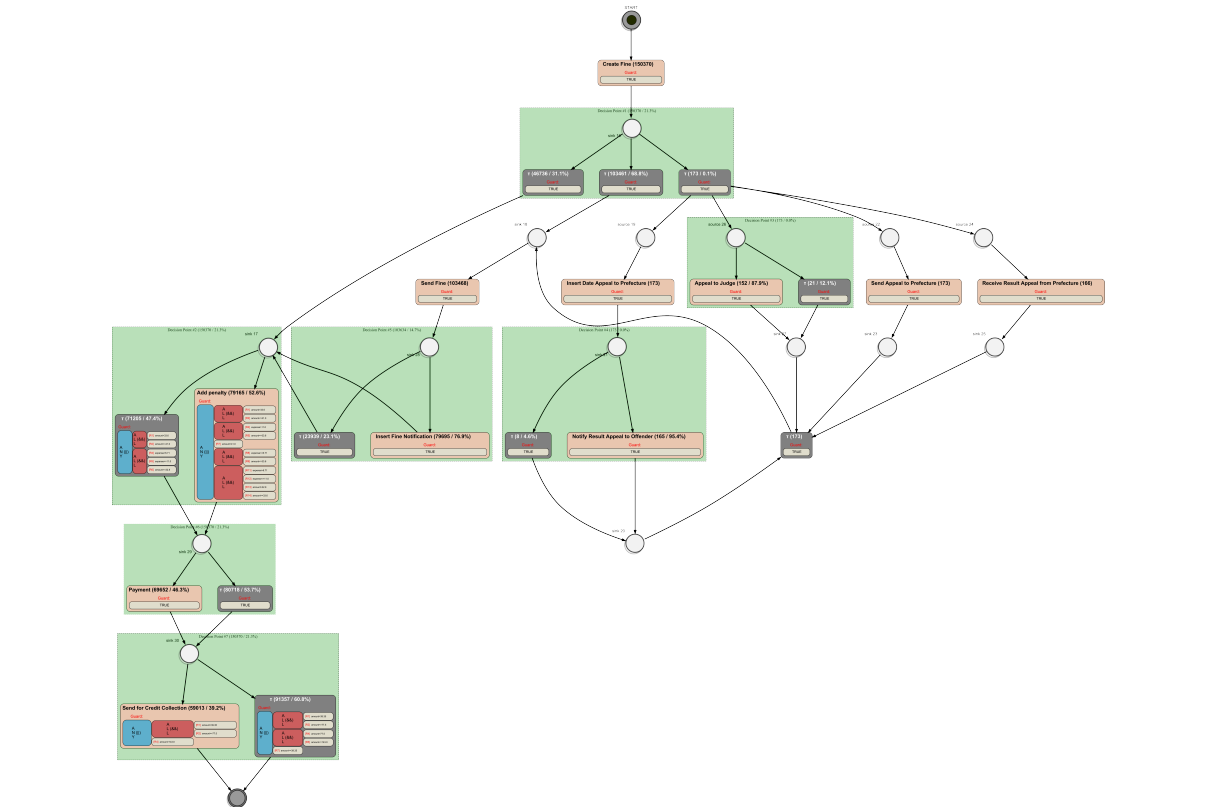
2.5.2 Stochastic Discovery

Stochastic process mining, as opposed to traditional control-flow discovery, aims to discover a process model annotated with the system weights for each activity in the process [73]. The curious reader is recommended to refer to “Process mining with labelled stochastic nets” by Burke [86] for an in-depth understanding of stochastic discovery. Work on stochastic process discovery, first considers if the transitions have a bias, or a *base weight*, that given a marking of the system, infers the likelihood of the action [183], [184]. However, discovering these base weights is not enough to investigate all perspectives of processes as they cannot change based on the context, or data, of a process execution, which means the likelihood is fixed.

To this end, techniques [76], [184] will discover both base weights and *adjustment* weights, such as the models presented in Figures 2.15 and 2.14. In the approach presented by Leemans, Mannel, and Sidorova [184], the authors proposed an adjustment for the history of the execution, allowing long distance dependencies to affect likelihoods of transitions. However, the likelihood is still fixed unless one considers changing the control-flow of the execution. Alternatively Mannhardt, Leemans, Schwanen, *et al.* [76] proposed a stochastic approach where adjustment weights are derived from the data generated from the execution. Notably for this approach [76], there exists an infinite number of likelihoods for the discovered stochastic Petri net for a single control-flow execution of the process. This thesis build on the approach used by [184] regarding adjustment weights and similarly focus on data as in [76], investigating exogenous influences on processes and the decision making occurring within them.



(a) Exclusive decision miner [57].



(b) Overlapping decision miner [58].

Figure 2.13: Several decision extensions of the process model seen in Figure 2.10b. Decision points are highlighted as green regions of the net and guards are visualised on the transitions.

2.6 Research Gaps and Summary

Many open research questions regarding process mining require advancement across process discovery, conformance checking and process enhancement. Three research consortiums [69]–[71] for advancing process mining motivate the contributions of this thesis.

The earliest consortium, the process manifesto [69], called for greater focus on complex event logs with diverse data sources ([69, C.2]) to analyse the behaviour of a process with time series data ([69, C.9]) such that the visibility of contextual influences on behaviour becomes clear ([69, C.10-11]). A later consortium focused on healthcare in their manifesto [70] and outlines open challenges within the domain. The manifesto [70] calls for frameworks to handle contextual information for process mining ([70, C.1]) and their use within healthcare. Another challenge outlined process discovery techniques lack capacity to consider the behaviour of processes beyond the enacted activities ([70, C.2]) and emphasised process outcomes that span across several departments should be analysed ([70, C.8]). This thesis investigates contextual data around processes, where IoT sensors are a common provider of such data. The IoT manifesto outlines the challenges [71] of connections between analytical techniques and IoT sources ([71, C.3]) and how to deal with unstructured environments that focus on decision making based on such data sources ([71, C.5]).

These three manifestos [69]–[71] highlight the need to advance context-aware process mining which has benefits for many domains, especially knowledge-intensive domains where actors are enabled to make ad-hoc decisions to ensure a positive goal is achieved. This snapshot of literature established the need to investigate techniques and process mining applications that can consider contextual data streams [69]–[71], summarised as:

- There is a growing need to capture the many tangible impacts from applying process mining in practice [8]–[10], [114], [115];
 - However, rarely do processes execute in complete isolation to the world around them [18], [52]–[54];
 - Yet, limited research has been conducted beyond the control-flow of processes and into contextual data such as exogenous data [55], [62], [69]–[71], [140];
 - Furthermore, existing contextual techniques focus on pre-processing data rather than considering the temporal nature of exogenous data [134], [136], [138], [139];
 - Similarly, this aspect of exogenous data is challenging to support within techniques [59], [141], [144];
 - Especially, for the quantification of decision mining [58], [169], [170], [173];
 - Finally, process enhancement presents a clear opportunity to explore contextual factors of processes, addressing many challenges faced by practitioners [69]–[71], [179].
-

The consensus within the presented literature is that the context surrounding processes can influence the behaviour of processes, but limited progress has been made in including contextual data, mostly as case or event attributes for process mining. This analysis of the literature established that no techniques are yet capable of exploiting exogenous data, as defined for this thesis, to discern its influence on the behaviour of processes. Since pre-processing exogenous data remains a mostly manual task that requires domain knowledge or expertise, this thesis serves to investigate *mechanisms for identifying exogenous influences on decision making in processes, and measuring the quality of data-aware process models.*

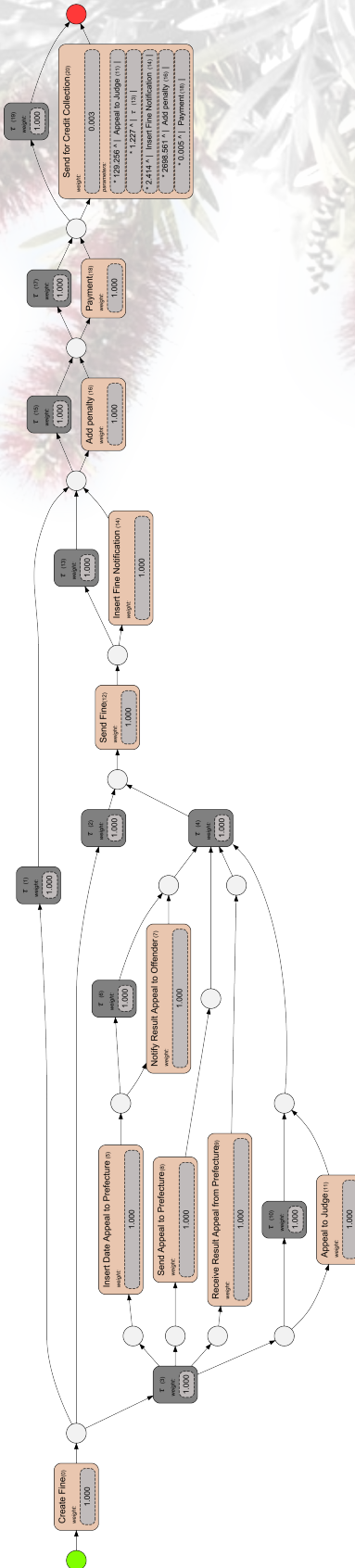


Figure 2.14: Dependencies stochastic miner [184] applied to the process model seen in Figure 2.10b. The semantics of this model is explored in Section 4.2.

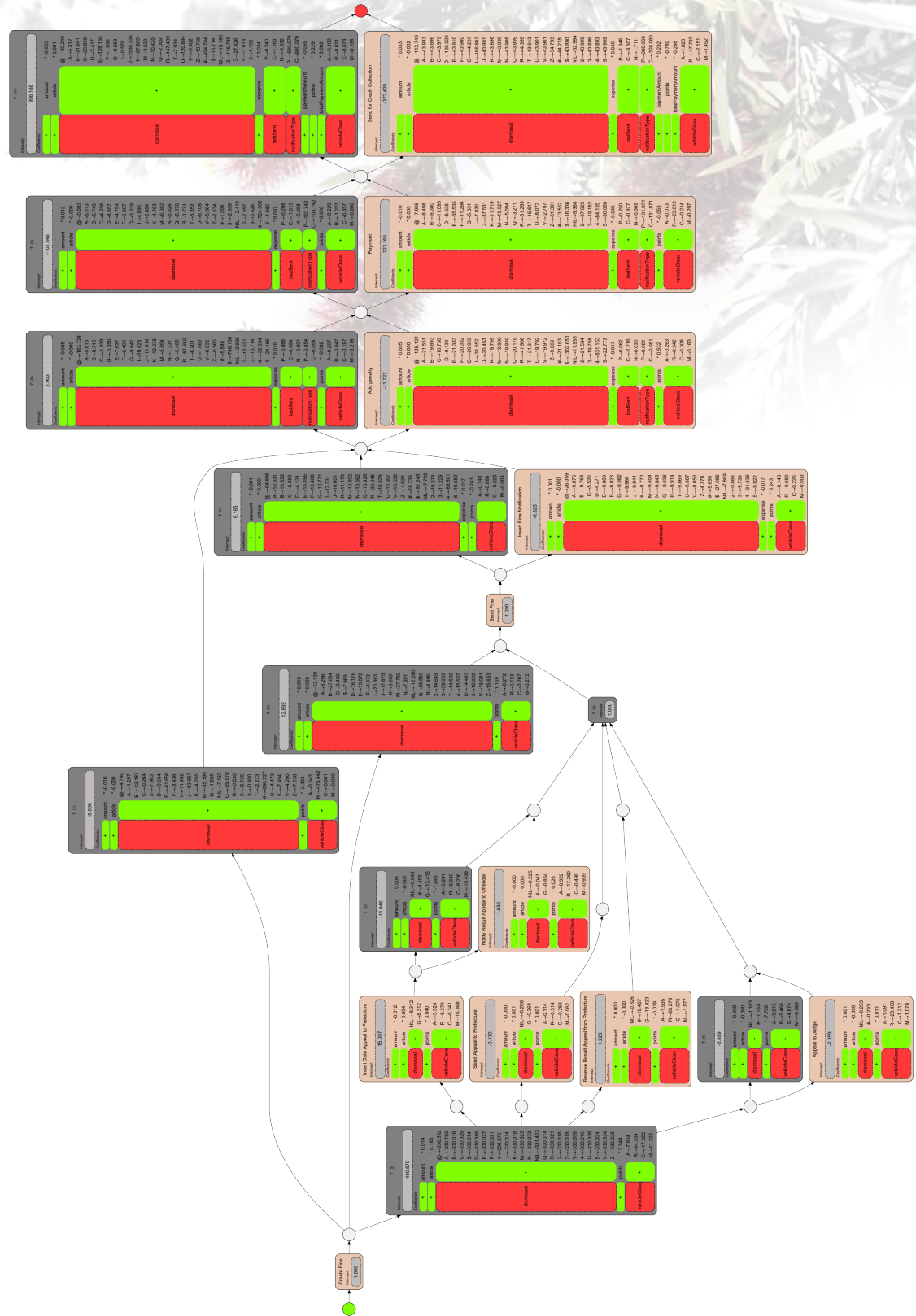


Figure 2.15: Data-aware stochastic miner [76] applied to the process model seen in Figure 2.10b. The semantics of this model is explored in Section 4.2.

Statement of Author's Contributions

The authors listed below have certified that:

1. they meet the criteria for authorship and that they have participated in the conception, execution, or interpretation, of at least that part of the publication in their field of expertise;
2. they take public responsibility for their part of the publication, except for the responsible author who accepts overall responsibility for the publication;
3. there are no other authors of the publication according to these criteria;
4. potential conflicts of interest have been disclosed to (a) granting bodies, (b) the editor or publisher of journals or other publications, and (c) the head of the responsible academic unit, and
5. they agree to the use of the publication in the student's thesis and its publication on the QUT's ePrints site consistent with any limitations set by publisher requirements.

In the case of this chapter: “xPM: A Framework for Process Mining with Exogenous Data” was published in a workshop at the *International Conference on Process Mining* in 2021.

To describe contributions, the CRediT taxonomy [190] is used for classifications.

Contributor	Statement of Contribution
Adam Banham (Student)	Conceptualisation; Methodology; Software; Validation; Formal Analysis; Investigation; Writing – Original Draft; Writing – Review & Editing; Visualisation.
Sander J. J. Leemans	Conceptualisation; Methodology; Writing – Review & Editing; Formal Analysis; Supervision.
Moe T. Wynn	Conceptualisation; Methodology; Writing – Review & Editing; Investigation; Supervision.
Robert Andrews	Conceptualisation; Methodology; Writing – Review & Editing; Investigation; Supervision.

Statement of Author's Contributions

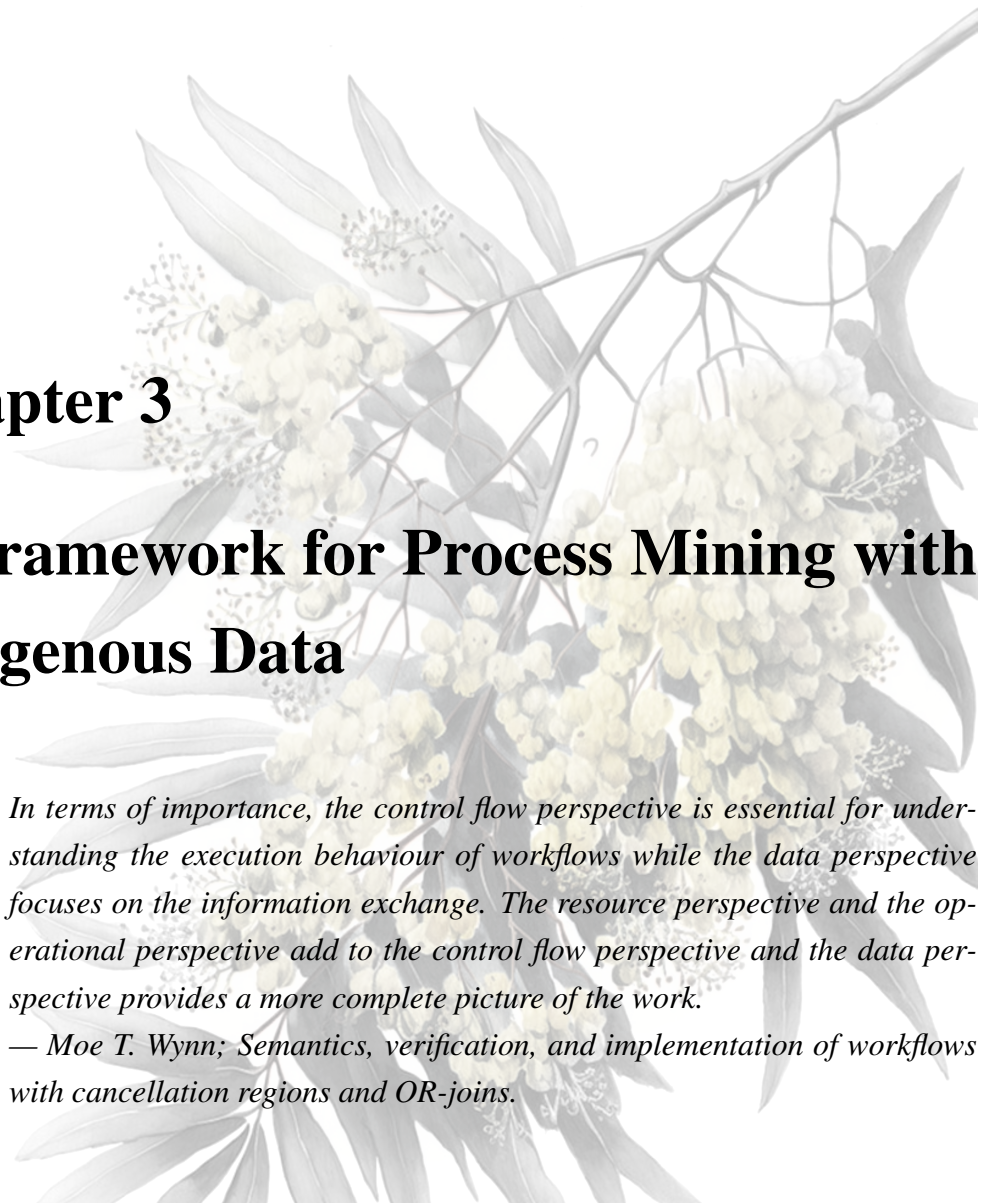
The authors listed below have certified that:

1. they meet the criteria for authorship and that they have participated in the conception, execution, or interpretation, of at least that part of the publication in their field of expertise;
2. they take public responsibility for their part of the publication, except for the responsible author who accepts overall responsibility for the publication;
3. there are no other authors of the publication according to these criteria;
4. potential conflicts of interest have been disclosed to (a) granting bodies, (b) the editor or publisher of journals or other publications, and (c) the head of the responsible academic unit, and
5. they agree to the use of the publication in the student's thesis and its publication on the QUT's ePrints site consistent with any limitations set by publisher requirements.

In the case of this chapter: “xPM: Enhancing exogenous data visibility” was published in *Artificial Intelligence in Medicine* within Volume 133 in November 2022.

To describe contributions, the CRediT taxonomy [190] is used for classifications.

Contributor	Statement of Contribution
Adam Banham (Student)	Conceptualisation; Methodology; Software; Validation; Formal Analysis; Investigation; Writing – Original Draft; Writing – Review & Editing; Visualisation.
Sander J. J. Leemans	Conceptualisation; Methodology; Writing – Review & Editing; Formal Analysis; Supervision.
Moe T. Wynn	Conceptualisation; Methodology; Writing – Review & Editing; Investigation; Supervision.
Robert Andrews	Conceptualisation; Methodology; Writing – Review & Editing; Investigation; Supervision.
Kevin B. Laupland	Resources; Data curation; Investigation; Validation.
Lucy Shinnars	Resources; Data curation; Investigation; Validation.



Chapter 3

A Framework for Process Mining with Exogenous Data

In terms of importance, the control flow perspective is essential for understanding the execution behaviour of workflows while the data perspective focuses on the information exchange. The resource perspective and the operational perspective add to the control flow perspective and the data perspective provides a more complete picture of the work.

— Moe T. Wynn; *Semantics, verification, and implementation of workflows with cancellation regions and OR-joins.*

The wretched may be motivated to explore exogenous data within their process analysis, and to do so requires some understanding of how to handle exogenous data. The eager wretched might jump to pre-processing exogenous data, but this chapter asks the wretched to hold off on this mistake and consider this chapter. This chapter investigates *how may exogenous influences on decision making in processes be visualised and analysed* (RQ1) and addresses two sub-questions, *how can exogenous data be represented alongside endogenous data in event logs* (R1.1), and *how can exogenous influences on process outcomes be visualised and analysed* (RQ1.2). The chapter presents xPM, a framework for handling exogenous data for process mining efforts, addressing RQ1.1. Then, the chapter introduces EESA, a process enhancement technique that produces a visual annotation over a process model, addressing RQ1.2. To convince the wretched, an empirical evaluation of xPM and EESA using publicly available datasets is conducted and contextualised within the healthcare domain.

3.1 Introduction

Process mining has been used in healthcare to study emergency departments of hospitals [20], [191], [192], and the clinical conditions and diseases such as sepsis [21], [193]–[197], heart

failure/chest pain [20], [198] and tuberculosis [199]. Applying process mining to healthcare processes enables clinicians to explore years of historical data and find opportunities to improve processes [22], [60], [62], [194], [200]. Such improvements can contribute to positive clinical, and beneficial financial outcomes which makes finding these opportunities is critical [201]–[203]. While process mining offers a range of techniques to support healthcare experts and processes, the application of techniques is not without challenges [22], [70], [114].

Challenges faced by process mining practitioners in the healthcare domain [70] focus largely around the process of decision making in healthcare. Examples include: the variety of data sources needed to make a clinical decision; the need for an interdisciplinary or institutional perspectives; the dependence on healthcare professionals that have the knowledge to make complex decisions; and, that process behaviour will evolve to reflect recent healthcare developments. In [22], the authors present recommendations to address these challenges, such as an interdisciplinary team approach to the project and multi-perspective techniques that provide ways to include a holistic view and application of the data available within healthcare organisations. While calls for multi-perspective techniques are not new, little work has, so far, been presented that handles these challenges.

This chapter presents a multi-perspective framework that establishes a distinction between process behaviour and rich data flows to provide the potential to address the challenges around complex decision making. An interdisciplinary approach with expert healthcare professionals is adopted, providing context to the evaluation, and enriching the process mining technique. The following terms: *endogenous* process data and *exogenous* data are used to distinguish between different types of data sources.

Endogenous data is defined as process data [2], data internal to a process, meaning data directly produced by process activities, or describes the moment of activity. Examples of endogenous data that describe a moment of a healthcare process include the name of the procedure being performed, the nurse performing the procedure, and the patient's level of discomfort during the procedure. In contrast, this thesis defines *exogenous* data as data external to a process, meaning data not tied directly to a process' activities, but provides the context in which process instances execute. Examples of exogenous data sources include availabilities of surgeons, ICU bed availability, and patient vital sign measurements.

The remainder of this chapter is structured as follows: Section 3.2 outlines the approach of this chapter, Section 3.3 introduces the formalisation and notation, Section 3.4 presents the xPM framework, Section 3.5 presents (EESA) explorative exogenous series analysis visualisation, Section 3.6 evaluates the xPM and EESA, Section 3.7 reflects on the chapter, and finally Section 3.8 summarises the chapter and discusses future work.

3.2 Approach

This chapter presents a framework for process mining with exogenous data, referred to as **xPM**, prescribing the steps for connecting contextual data relevant to the endogenous moments of a process, and for conducting process discovery and enhancement. Also presented is a new process mining technique, EESA, that uses xPM to understand how contextual factors influence the behaviour of processes. Specifically, EESA can visualise differences between changes in exogenous data and process outcomes. Furthermore, an automatic ranking procedure to identify similar trends between process outcomes is presented to avoid manually processing a large collection of EESAs for insight.

A quantitative analysis of data-aware techniques using both endogenous and exogenous data is conducted to establish the effectiveness of xPM and EESA. The evaluation considers activities in ICU wards and associated occurrences of hospital acquired sepsis were used and extracted from the publicly available MIMIC-III [47] dataset from the healthcare domain. In terms of data, events relating to patient treatment in ICU (endogenous data), and patient vital signs measurements (exogenous data) were extracted.

The benefits xPM for analysts is that the framework enables them to better capture the reality of processes by making multiple contextual data sources available to process mining techniques. Furthermore, xPM affords opportunities for process participants to explore decisions occurring within complex environments in terms of the context surrounding executions of the process.

3.3 Preliminaries

This section introduces events logs, exogenous data sets, Petri nets and a sub-formalisation for process descriptions used within contributions, xDPN. To formalise these concepts, the notation from [2], [51] is adopted and included here for completeness.

Universes. $\mathcal{U}_E$ is the universe of events, $\mathcal{U}_D = \{case, act, time, \dots\}$ is the universe of data variables, $\mathcal{U}_V$ is the universe of values, and $\mathcal{U}_{map}$ is the universe of variable-value mappings. $\mathcal{U}_T$ is the universe of timestamps, $\mathcal{U}_A$ is the universe of activities. Hence, $\mathcal{U}_A \cup \mathcal{U}_T \subset \mathcal{U}_V$ [51].

Event logs are a collection of events. An event e is an execution step of a process, recorded through data variables. Formally, an event log is (E, π) , where $E \subseteq \mathcal{U}_E$ is a set of events and $\pi \in (\mathcal{U}_E \rightarrow \mathcal{U}_{map})$ is a function that maps an event e to a variable-value mapping. The shorthand $e_{\pi,A}$ is used to denote the value of a data variable $A \in \text{dom}(\pi(e))$ for event e , i.e. $\pi(e)(A) = e_{\pi,A}$ [51].

An event e is assumed to have at least the following data variables: $\{case, act, time\}$, such that *case* is the trace related to an event, *act* is the process activity related to an event, and *time* denotes when an event occurred. The following shorthand denotes these data variables as needed, $\lambda : \mathcal{U}_E \rightarrow \mathcal{U}_A$ and $\lambda(e) = \pi(e)(act)$, also $\Gamma : \mathcal{U}_E \rightarrow \mathcal{U}_T$ and $\Gamma(e) = \pi(e)(time)$, finally $C : \mathcal{U}_E \rightarrow \mathcal{U}_V$ and $C(e) = \pi(e)(case)$.

Traces. An end-to-end execution of a process, or a trace ($T \in \mathbb{T}$) is a sequence of events derived from an event log (E, π) , such that $\mathbb{T} = \{\langle e_1, \dots, e_n \rangle \mid e_1, \dots, e_n \in E \wedge \forall_{1 \leq i < n} C(e_i) = C(e_{i+1}) \wedge \Gamma(e_i) \leq \Gamma(e_{i+1})\}$. The mapping function π is extended to also apply to traces, where it denotes the set of all values present in the trace, i.e. $\Pi : \mathbb{T} \rightarrow (\mathcal{U}_D \rightarrow 2^{\mathcal{U}_V})$, with $\Pi(T)(A) = \{\pi(e)(A) \mid e \in T\}$ or denoted using the shorthand of $T_{\Pi, A}$.

Exogenous Data. An *exogenous measurement* (exo-measurement) m^t is a single measurement $m \in \mathcal{U}_V$ at timestamp $t \in \mathcal{U}_T$. For instance, $37^{18-01-202211:55:23}$ may represent the result of a nurse measuring the temperature of a particular patient at the given timestamp. An *exogenous series* (exo-series) $\langle m_1^{t_1}, \dots, m_n^{t_n} \rangle^A$ is a sequence of exo-measurements, annotated with a variable-value mapping A . Furthermore, the universe of exo-series is denoted as $\mathcal{U}_{es}$. For instance, $\langle 37^{18-01-202211:55:23}, 37.5^{18-01-202212:13:58}, 38^{18-01-202213:14:32} \rangle_{\text{patient_id:1034847}}$ may represent the repeated measuring of body temperature by a nurse for a particular patient. The following shorthand denotes the i th element of an exo-series es , such that $es(i)$ is the i th exo-measurement recorded within an exo-series. Furthermore, to denote the value of a variable A for an exo-series es , $es_{\#A}$ is used.

An *exogenous panel* (exo-panel or $\mathbb{P}$) is a collection of exo-series ($\mathbb{P} \subseteq \mathcal{U}_{es}$). For instance, $\{\langle 37^{18-01-202211:55:23}, 38^{18-01-202213:14:32} \rangle_{\text{patient_id:1034847}}, \langle 37.8^{18-01-202215:47:39}, 37.8^{18-01-202220:15:41} \rangle_{\text{patient_id:1034882}}\}$. An *exogenous description* (exo-description or $\mathcal{U}_{exo}$) is a collection of exo-panels. Typically, an exo-panel would contain all related exo-measurements; for instance: all blood pressure measurements for all patients. An exo-description then contains several exo-panels; for instance: one for blood pressure, one for heart rate, and one for body temperature.

Labelled Petri Nets. A *Petri net* is a triple $N = (P, T, F)$, where P is a finite set of places, T is a finite set of transitions such that $P \cap T = \emptyset$ and $F \subset (P \times T) \cup (T \times P)$ is a set of directed arcs, called a flow relation [2]. A *labelled Petri net* is a quintuple, $(P, T, F, \Sigma, \lambda)$, where (P, T, F) is a Petri net, $\Sigma \subseteq \mathcal{U}_A$ is a set of activity labels and λ is overloaded to apply to transitions or an event labelling function $\lambda : T \rightarrow \mathcal{U}_A$ [2].

Semantics. Places may hold tokens, which are produced and consumed when transitions fire according to the flow relation. A transition is *enabled* if each input place contains a token. The state of a Petri net is a *marking*, which records what places have tokens and how many. An enabled transition l can *fire*, which updates the marking according to the flow relation F and, if l is labelled by λ , denotes the execution of activity $\lambda(l)$. An *initial marking* denotes the initial state of a Petri net before the first transition is fired. Elements of $P \cup T$ are called nodes. A node a is an input node of another node b if and only if there is a directed arc from a to b . A node a is an output node of another b if and only if there is a directed arc from b to a . For any $a \in P \cup T$, $\bullet a = \{b \mid (b, a) \in F\}$ and $a \bullet = \{b \mid (a, b) \in F\}$.

Petri Nets with Exogenous Data (xDPN). A *precondition* (ϕ) is a boolean expression describing a subset of values for data variables (e.g. temperature is higher than 20°C). A *Petri Net with Exogenous Data* (xDPN) is a sextuple $(P, T, F, \Sigma, \lambda, \Phi)$, where $(P, T, F, \Sigma, \lambda)$ is a labelled Petri net and $\Phi : T \rightarrow \phi$ associates transitions with preconditions.

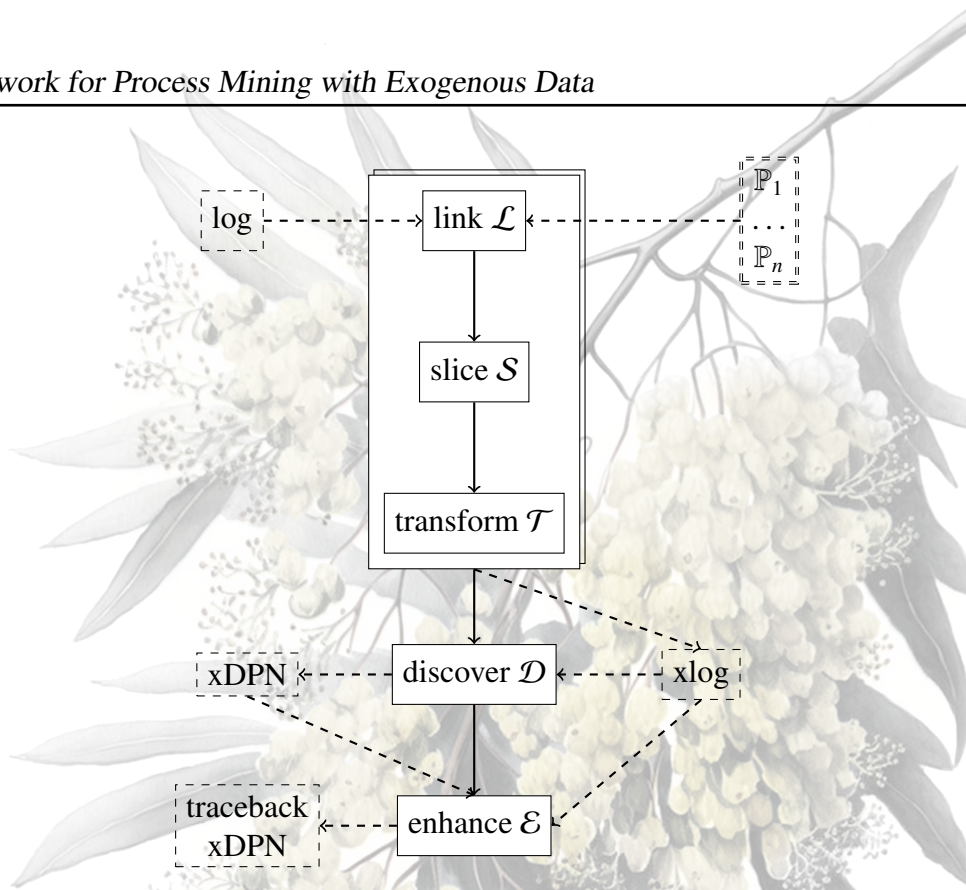


Figure 3.1: The xPM framework.

Semantics. The state of an xDPN is a marking and an assignment of endogenous and exogenous variables. A transition is *data enabled* if the precondition attached to a transition is satisfied by the current assignment of variables, or if there is no attached precondition. In an xDPN, a transition can fire if it is enabled and data enabled. An xDPN is a sub-formalism of DPN (a complete formalisation of DPN can be found in [55], [185], [204]): in contrast to xDPN, DPN distinguish between variable states (e.g. *read* or *written*) and enforce that transitions in the model update variable assignments. By dropping this last requirement in xDPN, exogenous data variables can be updated during execution.

3.4 The xPM Framework

This section introduces xPM, a framework for process mining with exogenous data (see Figure 3.1). xPM takes as input an event log and an exo-description ($\mathcal{U}_{exo}$). xPM applies several *determinations*, quadruples of $(\mathbb{P}, \mathcal{L}, \mathcal{S}, \mathcal{T})$, and creates as output an *exogenous-annotated* log (xlog). The purpose of a determination is to capture the conceptual bounds of one exogenous influence, targeting the likely window of interest for actors and their decision making. In each determination, $\mathbb{P} \in \mathcal{U}_{exo}$ is an exo-panel, $\mathcal{L}$ is a linking function which connects a trace and an exo-panel to an exo-series, $\mathcal{S}$ is a slicing function which creates sub-series (*slice*) of an exo-series for each event, and $\mathcal{T}$ is a transformation function, which transforms a slice into one or more event attributes.

The xPM framework applies each determination to annotate events with temporal data and flattened temporal event attributes, creating an *exogenous-annotated* log (xlog).

Table 3.1: Exemplar event log for ambulance triage.

$C(e_i)$	e_i	$\Gamma(e_i)$	$\lambda(e_i)$	suburb	ambulance	incident
1	1	18-01-2022-02:25	request	Runcorn		15
1	2	18-01-2022-02:30	assign		A	
1	3	18-01-2022-03:00	arrival			
1	4	18-01-2022-04:00	pickup			
1	5	18-01-2022-04:30	delivery	Brisbane		17
2	6	18-01-2022-06:18	request	Runcorn		
2	7	18-01-2022-06:20	assign		A	
2	8	18-01-2022-06:40	arrival			
2	9	18-01-2022-06:55	pickup			
2	10	18-01-2022-07:10	hand-over		B	
2	11	18-01-2022-07:35	delivery	Brisbane		

The xlog-specific additions include: a mapping from traces to exo-series; a mapping from events to portions of these exo-series; and possible flattened and simplistic representations as event attributes. This thesis focuses on treating temporal data as a first-class citizen through these mappings. The xPM framework emphasises these mappings and provides a means for connecting with pre-existing process mining techniques.

From the xlog, a discovery function $\mathcal{D}$ can discover an xDPN. In the last step, an enhancement function $\mathcal{E}$ aligns an xlog with a process model to visualise differences in process executions. After enhancement, exogenous data can be *traced back* to an exo-series for fine-grain analysis. Illustrations of xPM will use time series of continuous measurements, but the xPM framework does not limit the type of time series that can be used. In Section 3.4.1, an exemplar scenario to describe xPM is outlined.

3.4.1 Exemplar Scenario

This exemplar scenario considers how ambulance triage occurs and what influences whether a patient is treated on-site, transported to a hospital or handed over to another ambulance. Table 3.1 presents a snippet of this event log (i.e. the endogenous data). Events have the following variables; *suburb* is the destination of the ambulance, *ambulance* is the identifier for an ambulance, and *incident* is the identifier for an ambulance request. After arriving onsite, paramedics may observe the patient to determine if the patient needs to be transported and if so, they may also consider traffic, weather or the nearest ambulance with a senior paramedic. An analyst may want to understand how these factors influence how paramedics determine the best way to transport a patient.

To consider exogenous data in this scenario, an exo-description containing two exo-panels is used: the temperature in each suburb and the heart rate of the patient. Table 3.2 presents a snippet of exo-series from different exo-panels. This example data is used to demonstrate how the xPM framework conceptually constructs determinations. For instance, the example will

Table 3.2: Exemplar of a tabularised exo-description. All timestamps occur on 18-01-2022 in this example.

<i>exo-panel for weather of a suburb</i>		
exo-series id	attributes	exo-measurements (temperature reading)
1	{suburb: Runcorn }	$\langle -5^{03:00}, -2^{04:00}, 3^{05:00} \rangle$
2	{suburb: Brisbane}	$\langle 0^{03:00}, 2^{04:00}, 5^{05:00} \rangle$
<i>exo-panel for heart rate of patients</i>		
exo-series id	attributes	exo-measurements (Beats per second)
4	{ambulance: A, incident: 15}	$\langle 95^{03:15}, 97^{03:20}, 110^{03:25} \rangle$
6	{ambulance: B, incident: 17}	$\langle 105^{06:55}, 110^{07:00}, 115^{07:05} \rangle$
7	{ambulance: A, incident: 17}	$\langle 120^{07:15}, 122^{07:20}, 125^{07:25} \rangle$

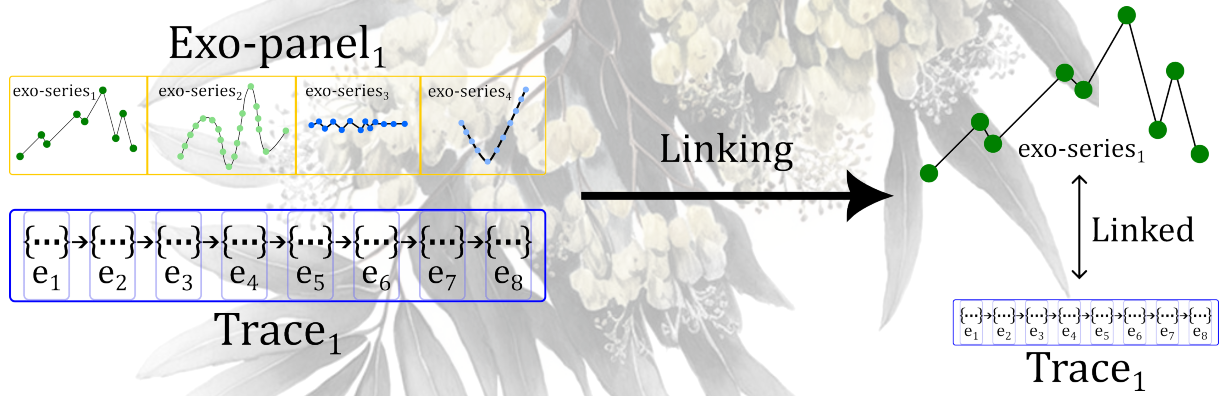


Figure 3.2: An illustration of linking. Given a trace and an exo-panel, the linking function returns a linked exo-series for the given trace.

consider linking the weather and heart rate measurements in Table 3.2 as exogenous influences with the trace data shown in Table 3.1. This example is used throughout the following sections to show that our approach can be generalised to investigate any possible exogenous influences on processes (with the selection of these influences typically being user-driven).

3.4.2 $\mathcal{L}$ – Linking

The framework starts by linking exo-panels to traces, as shown in Figure 3.2. The act of linkage is described in a linking function ($\mathcal{L}$), and if a link exists, finds an exo-series for the trace. For example, a linking function could connect an ICU admission to the workload of an attending nurse, to laboratory results, or to weather conditions surrounding the hospital.

Formally, $\mathcal{L}$ takes a trace $T \in \mathbb{T}$ and an exo-panel $P \in \mathbb{P}$, and finds an exo-series, i.e. $\mathcal{L} : (\mathbb{T} \times \mathbb{P}) \rightarrow \mathcal{U}_{es}$. To illustrate $\mathcal{L}$, consider the exemplar scenario with two exo-panels, temperature and heart rate, and two possible linkage options. A $\mathcal{L}$ for temperature would likely be $L_1 \in \mathcal{L}$, showing the suburb attribute from events ($T_{\Pi, \text{suburb}}$) is used to find a matching exo-

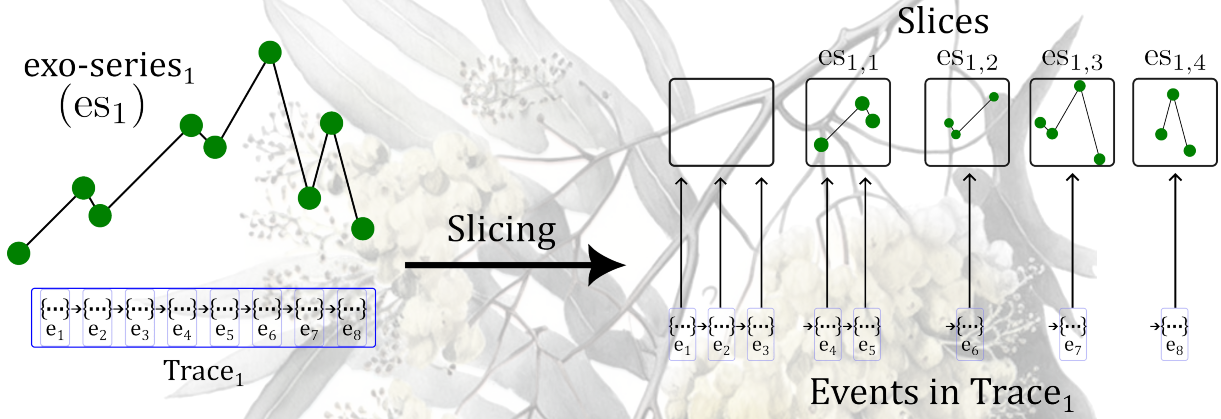


Figure 3.3: An illustration of slicing, where given trace_1 and exo-series_1 , some events are associated with the same slice (e_4, e_5), some have their own slice (e_6, e_7, e_8), and others are associated with an empty sequence (e_1, e_2, e_3).

series. A $\mathcal{L}$ for heart rate would likely be $L_2 \in \mathcal{L}$, showing the ambulance and incident variables from events ($T_{\Pi, \text{incident}}, T_{\Pi, \text{ambulance}}$) is used to find a matching exo-series .

$$L_1(T, P) = es \iff es \in P \wedge es_{\#_{\text{suburb}}} \in T_{\Pi, \text{suburb}}$$

$$L_2(T, P) = es \iff es \in P \wedge es_{\#_{\text{incident}}} \in T_{\Pi, \text{incident}} \wedge es_{\#_{\text{ambulance}}} \in T_{\Pi, \text{ambulance}}$$

In case $\mathcal{L}$ finds two or more exo-series for a trace, $\mathcal{L}$ must merge these into a single exo-series . Merging time series is not trivial and will require a thorough understanding of exogenous data and domain knowledge. For example, if two heart rate sensors are monitoring a patient, but they are not time-synchronised, then they need to be handled within the linking function to find the truthful exo-series . At this stage handling this concern is left to the user and would be interesting future work.

3.4.3 $\mathcal{S}$ – Slicing

Slicing functions ($\mathcal{S}$) associate a sub-series of an exo-series (*slice*) with each event in a trace. After applying a slicing function, some events may be associated with an empty slice, some are associated with the same slice, and others have their own slice (as seen in Figure 3.3). Examples of how a slicing function may create slices include selecting all exo-measurements before a patient undergoes a procedure, all exo-measurements in the hour before a procedure, or all exo-measurements between major interventions.

Formally, a slicing function $\mathcal{S}$ is applied to each event in a trace. It takes an event and an exo-series , to return a sub-series of the exo-series , i.e. $\mathcal{S}: (\mathcal{U}_E \times \mathcal{U}_{es}) \rightarrow \mathcal{U}_{es}$. Thus, creating a slice of relevant exogenous data from an exo-series , for each event. Domain knowledge will often inform the choice of slicing functions and assisting the automatic selection of a slicing function is an interesting area of further research. Such options for automating slicing could consider the frequency of measurements, autocorrelation with a time lag, or investigating

the amount of variance within an exo-panel. Another approach could be to consider research around databases or OLAP operations [205], [206] to automate or provide complete functions for slicing time series data.

To illustrate $\mathcal{S}$, consider the exemplar scenario. In this scenario, the two exo-panels have different durations of relevance, for temperature longitude trends are crucial. Thus, a suitable $\mathcal{S}$ for temperature is $S_1 \in \mathcal{S}$, where slices are created using a two-hour window of exo-measurements (before and after each event). In contrast, for heart rate, short-term changes are significant. Therefore, a suitable $\mathcal{S}$ for heart rate is $S_2 \in \mathcal{S}$, where slices are created from a 15-minute window of exo-measurements before each event.

$$S_1(e, es) = \langle m^t \mid m^t \in es \wedge (\Gamma(e) - 1 \text{ hour}) \leq t \leq (\Gamma(e) + 1 \text{ hour}) \rangle$$

$$S_2(e, es) = \langle m^t \mid m^t \in es \wedge (\Gamma(e) - 15 \text{ minutes}) \leq t \leq \Gamma(e) \rangle$$

3.4.4 $\mathcal{T}$ – Transformation

Next, transforming a slice for existing process mining techniques and future techniques is introduced, as seen in Figure 3.4. The result of a transform¹ for a slice, is a set of one or more event attributes. This transformation is handled through a transformation function ($\mathcal{T}$), which takes an event and an associated slice, to create event attributes (*transformed attributes*) for the notable features of a slice.

Formally, a transformation function is $\mathcal{T} : (\mathcal{U}_E \times \mathcal{U}_{es}) \rightarrow 2^{\mathcal{U}_D \times \mathcal{U}_V}$. Examples of transformations may include whether an extreme measurement exists, the number of peaks and lows, or whether the trend line of a slice was increasing or decreasing over time. Transforms $\mathcal{T}$ typically take one of three forms: (i) $\mathcal{T}$ can return a single attribute; such a transformation might return the minimum, maximum or mean of a slice; (ii) $\mathcal{T}$ can return a set of attributes; e.g. coefficients of the n th Taylor polynomial of the slice; (iii) $\mathcal{T}$ can be recursive to meet either case (i) or (ii). Such a transformation finds the n th derivative of a slice, then applies any previously mentioned examples, to meet the form of either (ii) or (iii) as shown in Figure 3.4.

To illustrate $\mathcal{T}$, consider the exemplar scenario. In this scenario, the interpolation is vastly different between exo-panels and each requires a unique transformation function; xPM does not limit the user in their choice of transformation and what follows is a demonstration of a few possibilities. One example of $\mathcal{T}$ for temperature could be $\mathcal{T}_1 \in \mathcal{T}$, showing a slice being transformed into an attribute that states if extreme weather occurred. Another example of $\mathcal{T}$ for heart rate could be $\mathcal{T}_2 \in \mathcal{T}$, showing a slice being transformed into many attributes, such as the

¹ Transformation can be any aggregation such as min, max, mean, or any more complex aggregation.

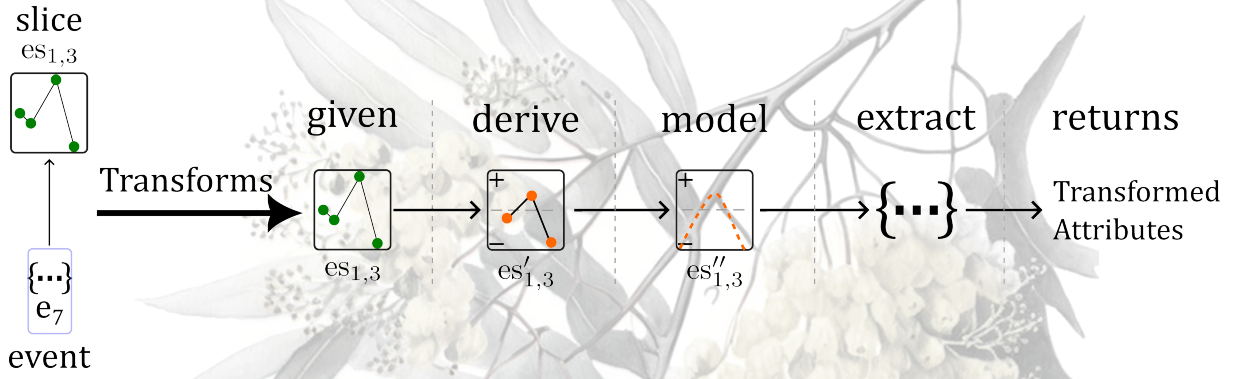


Figure 3.4: An illustration of transformation, where given e_7 and a slice of an exo-series $es_{1,3}$, the transformation derives the velocity of a slice ($es'_{1,3}$), models the derivation ($es''_{1,3}$), then extracts notable features from the modelling as transformed attributes.

velocity of a heart rate and if a projected heart rate pattern will go outside guidelines.

$$\text{abnormal}(es) \equiv \exists_{m^t \in es} -10^\circ\text{C} \geq m^t \vee m^t \geq 40^\circ\text{C}$$

$$\mathcal{T}_1(e, es) = \{('extreme', \text{abnormal}(es))\}$$

$$\text{velocity}(es) = \left\langle \left(\frac{m_{i+1} - m_i}{t_{i+1} - t_i} \right)^{t_{i+1}} \middle| m_i^t = es(i) \wedge m_{i+1}^{t_{i+1}} = es(i+1) \wedge 1 \leq i < |es| \right\rangle$$

$$\text{avg}(es) = \sum_{m^t \in es} \frac{m}{|es|}$$

$$\text{dng}(m) \equiv -5 \geq m \vee m \geq 5$$

$$\mathcal{T}_2(e, es) = \{('velocity', \text{avg}(\text{velocity}(es))), ('danger', \text{dng}(\text{avg}(\text{velocity}(es))))\}$$

In this scenario, two determinations for xPM were constructed, (temperature, $L_1, S_1, \mathcal{T}_1$) and (heart rate, $L_2, S_2, \mathcal{T}_2$). Access to exogenous data is facilitated after xPM has applied all determinations to an event log, such that some events are annotated with (i) exogenous slices and (ii) transformed attributes. After applying determinations to a log, an *exogenous-annotated* log (xlog) is produced.

3.4.5 $\mathcal{D}$ – Discovery

This step uses the newly created xlog to discover if the behaviour of a process varied based on the available exogenous data. A discovery function ($\mathcal{D}$) is applied to the xlog to achieve this goal. Given the plenitude of process discovery techniques and process notations in literature, both the slices and transformed attributes can be used to investigate the behaviour of a process using existing techniques.

One example of process discovery techniques that could use transformed attributes is decision mining, which finds insightful business rules for decision points. Examples of such techniques are [207] and [58]. However, researchers should be careful about assumptions that these techniques make about the nature of data variables. One assumption made is the constant

data placement within process executions, which suits endogenous data variables but may have unintended consequences when considering exogenous data variables. Another approach for process discovery techniques could be to use the lower-level information stored in the slices. Using the trends stored within slices, techniques could extract unique process behaviour associated with the changes within the slice. The description of process behaviour based on exogenous influences will require new process notations and execution semantics but may lead to valuable insights.

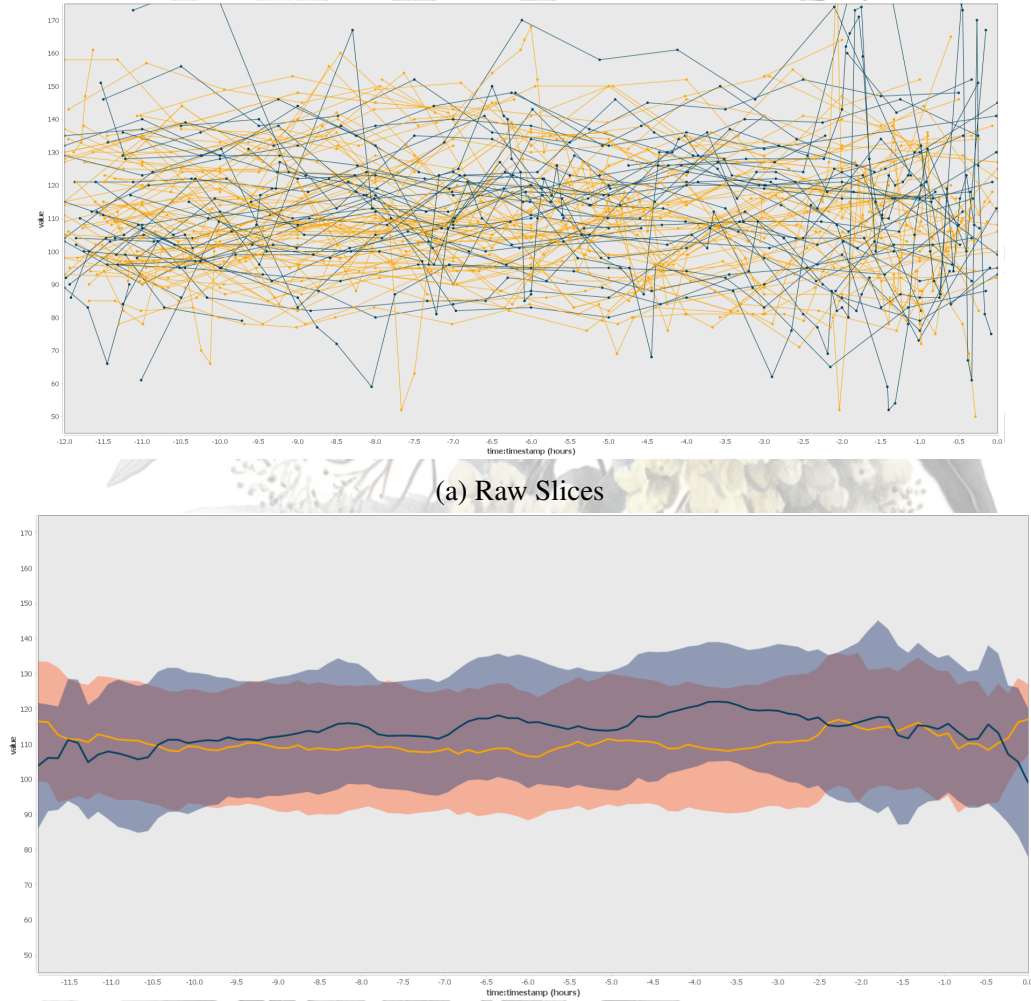
3.4.6 $\mathcal{E}$ – Enhancement

The final step of xPM is the enhancement step $\mathcal{E}$ which uses both endogenous and exogenous data in an xlog and a process model, to visualise the process and the exogenous influences on the process. One approach for techniques could be to investigate process activities linked to exogenous changes and use explainable machine learning techniques [208]–[210] to explain why variance occurs. Another approach could be to use the exogenous data, find cohorts of traces with similar contextual settings, and then visualise differences in process executions between cohorts. Furthermore, techniques describe trends within slices on process models to show differences between process outcomes.

Process mining techniques in this step are likely to rely on process conformance techniques that produce an alignment between a process model and an event log. An (optimal) alignment is a mapping between observed process behaviour occurring in reality (traces/events) and a process model such that the deviation between is minimised [81]. While data-aware alignments exist (e.g. [169], [170]), [169] only considers the writing of variables by transitions and [169], [170] correct the data written by transitions using Integer Linear Programming. However, exogenous data should not be adjusted in conformance techniques as it occurs outside the internal process execution. Further work is needed to generalise these techniques for usage outside business processes, as assumptions introduced in these techniques about the nature of data variables may have negative consequences when applied to exogenous data.

3.5 Explorative Exogenous Series Analysis (EESA)

Explorative Exogenous Series Analysis (EESA) is an enhancement technique within xPM and this section outlines the contribution. This technique visualises changes in exogenous sources seen at a point within a process model by building upon the xPM framework to visualise slices. Naively, one could visualise the slices by using a line for each slice observed, as seen in Figure 3.5a. Identifying trends or similarities at a glance may only be possible when the number of slices is relatively low or when the variance between slices is low. EESA overcomes this drawback and presents the observed trends regardless of the number of slices used, as seen in Figure 3.5b. At a glance, users can see the variance between slices and the median trend



(b) The proposed overlapping EESA visualisation showing the median trend line for blood pressure readings between two groups of patients shown via colour (oranges vs blues). The shaded area show the amount of variance seen within groups.

Figure 3.5: A comparison of visualisation techniques, one using the exogenous slices as-is (top), and another showing the generalisation of trends seen in raw slices (bottom). The usage of colour in each figure, is to highlight that a slice is associated with a particular process outcome.

between slices in an exogenous source between activities in a process model. EESA focuses on using continuous measurements within slices leaving other types for future work.

This section outlines the steps to handle slices and generate an EESA visualisation. These steps start with collecting slices to form observations, constructing a shared time axis, interpolation, and visualisation. Section 3.5.1 presents a variation of these steps that allows for multiple process outcomes to be considered when visualising slices. Lastly, Section 3.5.2 presents an automatic analysis procedure for ranking a collection of EESA visualisations.

EESA uses a collection of observations (O) as an input data source. Each observation (e, s, A, X) , is a quadruple of an event (e), a slice (s), the activity label of e ($A = \lambda(e)$) and the exogenous data source (X). Each observation used as input must have the same A and X , to ensure that all slices are comparable when visualised. Before visualisation, each s is resampled

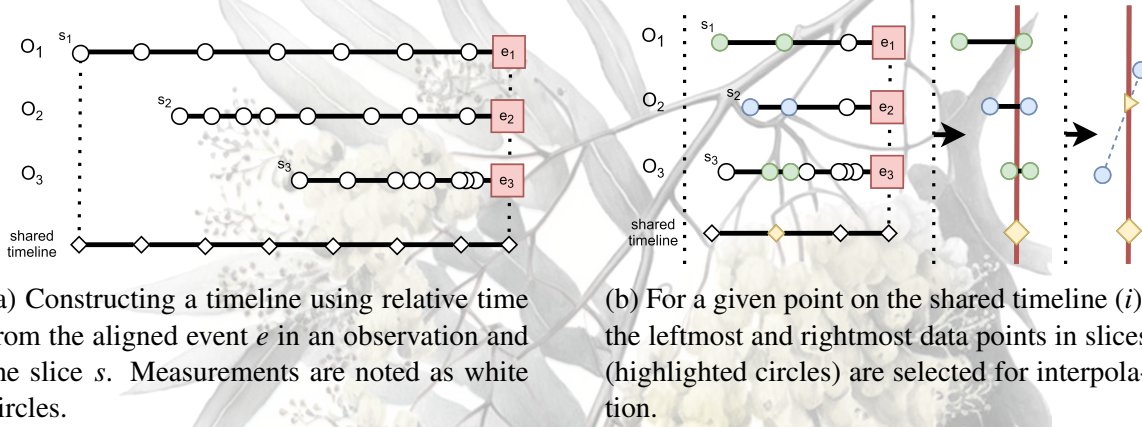


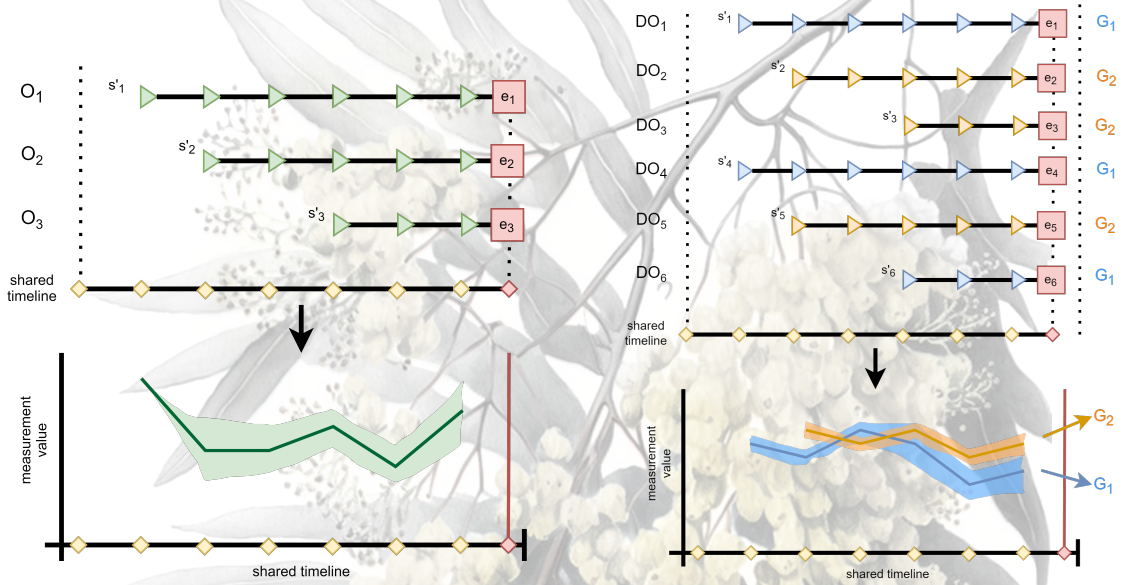
Figure 3.6: Construction of shared time axis (left) and interpolation of values for shared time axis (right).

across a shared time axis (on relative time from e) using linear interpolation to ensure that a consistent view is presented. Finally, slices are interpolated for each point on the shared time axis and the median value and one standard deviation on each side of the median value are plotted as a trend.

EESA's first step is to construct a shared time axis covering the longest timespan seen in observations. To construct this time span, all slices (s) are considered in relative time difference from their event (e), which ensures that at time $00:00:00$ the event is always observed. Then, the oldest and latest measurement recorded across all slices (s) are found. Using these two extreme measurements, EESA constructs a timeline ($TL \subseteq \mathcal{U}_T$) for resampling. This timeline has intervals ($i \in TL$) that are evenly spaced between these measurements, as seen in Figure 3.6a. The number of intervals to consider for a given timeline is left to the analyst to choose. Depending on the amount of variance within slices or frequency of measurements, more or less distance between intervals may be preferred. However, for the following section, 100 intervals within the timeline is assumed and automating this choice is left for future work.

The second step of EESA uses $i \in TL$, to find interpolated values at i for all s in the given observations. For a given time i and slice s , the nearest earlier measurement and the nearest older measurement in s is found. Then using linear interpolation, an interpolated value at i is calculated as shown in Figure 3.6b. If a measurement cannot be found either before or after i , then no value is recorded from this s at i . This step generates a mapping that returns all possible interpolated values at the given i .

The final step of EESA collects statistics needed for each $i \in TL$ to create the visualisation. The visualisation consists of a line and a shaded area, the median is plotted as a line, and one standard deviation ($-/+$ from the median) is plotted as a shaded area. The median is chosen as it is less prone to outliers, shows an actual value rather than a summary value, and is a robust measure of the central tendency [211]. These choices allow an EESA visualisation to show the evolution of slices, and where variation in measurements would likely occur across slices, as seen in Figure 3.7a.



(a) How an EESA visualisation uses interpolated values for a given i to plot the median trend (solid) and one standard deviation (shaded).

(b) How an overlapping EESA visualisation uses group identifiers (G_i) to create comparable trends.

Figure 3.7: Demonstrations of EESA visualisation process from slices (top) to generated outcome (bottom)

3.5.1 Overlapping EESA

This section presents a variation of the EESA visualisation, where multiple trends can be presented within a single visualisation on the same axis. This variation, referred to as an *overlapping* EESA visualisation, characterises each observation to allow trends to be compared within graphs rather than between graphs, where the shared time axis and measurement range could influence interpretation. This variation adds a discrimination process of observations, a recursive step around the last step of the EESA procedure, whereby observations are extended to a quintuple, referred to as a distinguished observation (DO).

A distinguished observation is $DO = (e, s, A, X, G)$, where (e, s, A, X) is an observation, and G is a set of group identifiers for the observation. If G is limited to a size of one, this signals that DO cannot be used with replacement. However, when G is not limited to a size of one, the group identifiers are less distinctive, and DO can be used with replacement across groups. The identification of group identifiers can vary, as the process model, process behaviour, and exogenous data could inform how to characterise an observation and the number of possible groups. These group identifiers are used to create subsets of the original collection of interpolated values, whereby each subset generates a separate EESA visualisation on the same graph.

To find group identifiers for an observation O , a discrimination function is introduced before the visualisation step in EESA. This function takes a process model, a partial process execution, and all data associated with the execution and finds a set of group identifiers for an observation O , thus creating distinguished observation DO . Applying the discrimination function to each

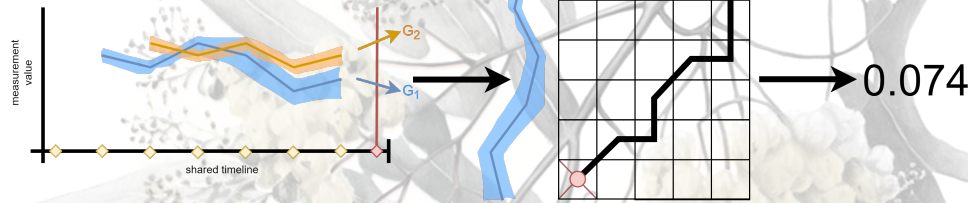


Figure 3.8: Measurement of an overlapping EESA graph (left), where a dynamic time warping matrix is computed between median trends (middle) then the summarisation of the path produces a measurement (right).

O , a collection of DO is created. Next, subsets of DO are created where all DO have a single group identifier in common and an EESA visualisation is generated for the subset. Finally, all generated EESAs are overlapped in a single visualisation, thus creating an overlapping EESA, as seen in Figures 3.5b, 3.7b and 3.8.

3.5.2 Highlighting Points of Interest

This section presents an automatic macro-analysis to complement EESA visualisations that provides micro-analysis of process executions. The analysis' goal is to find where distinctions exist between process outcomes across a process model using a collection of EESAs. To highlight where the most/least amount of distinction exists, all EESA visualisations are ranked based on the amount of distinction seen. This analysis focuses on the following properties; the shape of the trends should be considered (not just the magnitude of change), usability on differing lengths of time series, and measure differences in behaviour. The solution for ranking uses a scaling function to make EESA visualisations cross-compatible and Dynamic Time Warping (DTW) [212], [213] for ranking.

The ranking procedure takes a collection of EESAs as input, then for each EESA, computes a measure of distinctiveness and assigns a rank. The lower the assigned rank, the higher measure of distinctiveness between median trends observed in the visualisation. Only overlapping EESAs are measured, and all other visualisations are given a pseudo rank, the highest possible rank plus one ($n + 1$). For example, given the collection of EESAs in Figure 3.9, the three EESAs that are overlapping would be given a rank based on their distinctiveness. Then, all remaining EESAs are given the highest possible rank, which is $3 + 1$, as the last rank given to an overlapping EESA was three.

The analysis uses a process model and an xlog to create a collection of EESA visualisations from the transitions in the process model using the slices in an xlog. Using this collection, the analysis starts by making two subsets, one for overlapping EESAs and another for non-overlapping. For each overlapping EESA, the median trend for each G_i is extracted and scaled, then a summation of DTW between medians in a pairwise manner is computed. After each

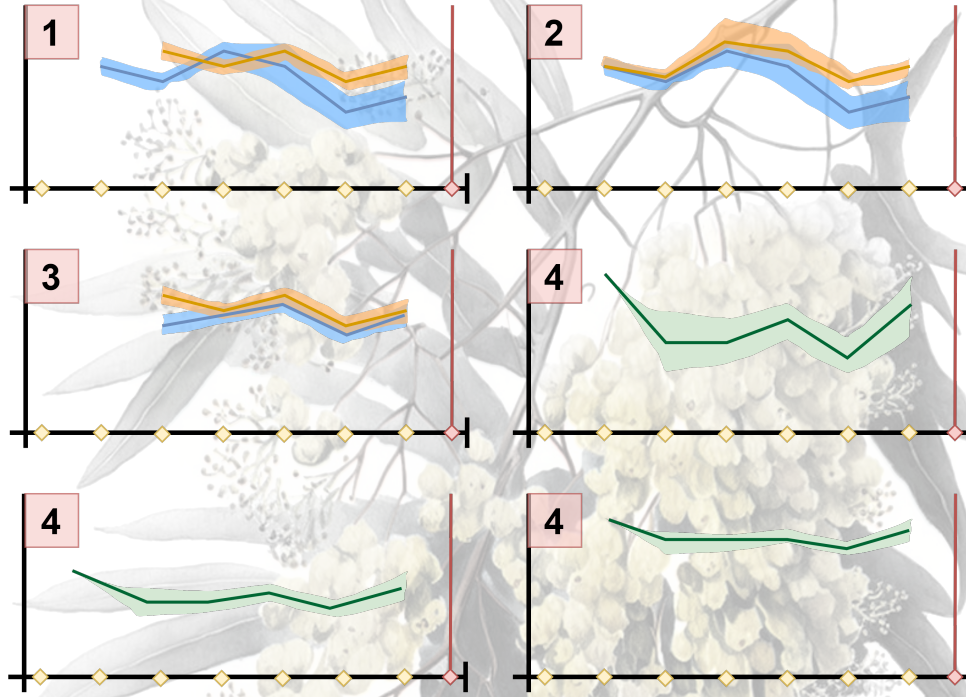


Figure 3.9: Example of ranking EESA visualisations, rank is denoted in red box, in top left. All non-overlapping EESA are ranked at $n + 1$ (4) while overlapping EESA are ranked based on distinctiveness between outcomes (1-3).

overlapping EESAs has been measured, the subset is sorted such that the first element is the one with the largest summation of DTW. Finally, ranks are assigned based on this ordering for overlapping EESA (1, 2, 3, ..., n), while EESAs in the second subset are given a pseudo rank ($n + 1$).

Analysts may be interested in different ends of the ranking system, depending on the discrimination function and the analysed process. For example, using the ranking, analysts could use the discrimination function to consider if slices agreed, or disagreed, with the associated precondition. Then using ranking, analysts could consider the relatively highly distinctive graphs (near rank 1), the relatively least distinctive graphs (near rank n), or rank $n+1$ for non-overlapping EESAs to understand how exogenous data could be informing decision making.

By using DTW to compute the total difference between trends, the method has the following benefits. Firstly, the method is elastic, meaning that the two trends under comparison do not need to be equal in length. Second, the method compares the similarities between shape and time, so differences will be noted when peaks and lows occur differently. Finally, the method computes a mapping between trends through a matrix and finds the path with the least amount of resistance that can be adapted for both numerical and discrete cases, as seen in Figure 3.8. Furthermore, by using min-max scaling on trends before computing DTW, exogenous sources with different magnitudes/measurements will be comparable through the ranking.

3.6 Evaluation

This section evaluates both contributions, xPM and EESA, presented thus far. Section 3.6.1 evaluates the model quality of models discovered by xPM. Section 3.6.2 illustrates the influences of exogenous data on clinical outcomes using EESA and the ranking procedure for a real-world healthcare setting. Both evaluations use a publicly available dataset, MIMIC-III [47] to construct event logs. This dataset contains anonymised data about forty thousand patients who stayed in critical care units at a large tertiary care hospital between 2001 and 2012.

An MVP interface for xPM is implemented in Java², which uses the Weka³ package for machine learning, JFreeChart⁴ for visualisation, and several ProM⁵ plug-ins libraries. The interface was implemented as a plug-in to the ProM framework (an open-source project for process mining) in the ‘ExogenousData’ Package.

3.6.1 Exogenous Influences on Model Quality

This section evaluates the influence of exogenous data on the quality of discovered xDPNs, using two event logs from the MIMIC-III dataset. This evaluation investigates the capacity of xPM to find preconditions using decision mining techniques and the effects on model quality. The initial hypothesis was that including exogenous data would increase the possibility of finding data-driven rules, as more data attributes would be available to techniques. However, by adding more attributes which are prone to being more conditionally available, quality measurements will penalise outcomes for a lack of consistency across traces. Meaning that finding some data-driven rules based on exogenous data, will likely be a double-edged sword in terms of model quality.

Procedure

Model quality is evaluated in three settings: (i) an event log with endogenous data attributes (*endo*); (ii) an event log with exogenous attributes where endogenous attributes have been removed (*exo*); and (iii) an event log with both endogenous and exogenous attributes (*endo+exo*). In the settings with exogenous data (ii,iii), the outlined xPM configuration below was used to produce exogenous attributes to compare with the endogenous case (i). The following instantiation of xPM was used:

$\mathcal{L}$ For each exogenous data set, a linking function was defined that links datasets to the patient of the trace and that occurred during the admission.

² <https://github.com/promworkbench/ExogenousData>

³ <https://www.cs.waikato.ac.nz/ml/weka/>

⁴ <https://www.jfree.org/jfreechart/>

⁵ <https://promtools.org/>

- $\mathcal{S}$ Two slicing functions are used. Let $\langle e_1 \dots e_n \rangle$ be a trace. Then, for event e_i the first slicing function (S_1) finds sub-time series between events e_{i-1} and e_i , while the second slicing function (S_2) finds the sub-time series between e_1 and e_i .
- $\mathcal{T}$ Four transformation functions were introduced: a minimum, an average, a maximum and a cumulative sum of a Fourier transform [214].
- $\mathcal{D}$ To discover a control-flow model, the Inductive Miner - infrequent [33] with path filtering of 0.25 was applied. To discover an xDPN, two decision mining techniques were used: Mutually Exclusive Decision Tree (me) [57] and Overlapping Rules Decision Tree (or) [58]. These techniques each take a parameter *min instances* (mi) that sets the minimum level of observed decision point instances that support a clause in a precondition. The experiment was repeated for several settings, $mi \in \{0.05, 0.15, 0.25\}$.

$\mathcal{E}$ was not part of this evaluation.

In total, 6 xDPNs were discovered for each variation of the source event log. For instance, Figure 3.10 and Figure 3.11 show the three xDPNs produced by a decision mining approach for a log with both exogenous and endogenous data. The number of decision points within these xDPNs are consistent across all logs as the same control-flow model is used. Thus, the only change is in the annotated preconditions from the application of the decision mining technique. To investigate what, how many, and the quality of the preconditions introduced, these models are measured in several ways.

The quality of the discovered xDPNs is assessed using fitness, precision and determinism. For fitness, balanced multi-perspective conformance checking [170] is used (see def. 41 for an in-depth understanding). For precision, multi-perspective precision [189] is used (see def. 45 for an in-depth understanding). For determinism, the following measure is proposed, which expresses the number of decision points in the model that are deterministic. That is, the fraction of transitions in the model, that are in the postset of places with two or more outgoing arcs (decision points), and have an associated precondition (i.e. $\Phi(t) \neq \perp \wedge \Phi(t) \neq \text{true}$).

Definition 1 (Determinism) Let $N = (P, T, F, \Sigma, \lambda, \Phi)$ be an xDPN. Then, the following functions take N and quantify the number of transitions with an associated precondition, defined as:

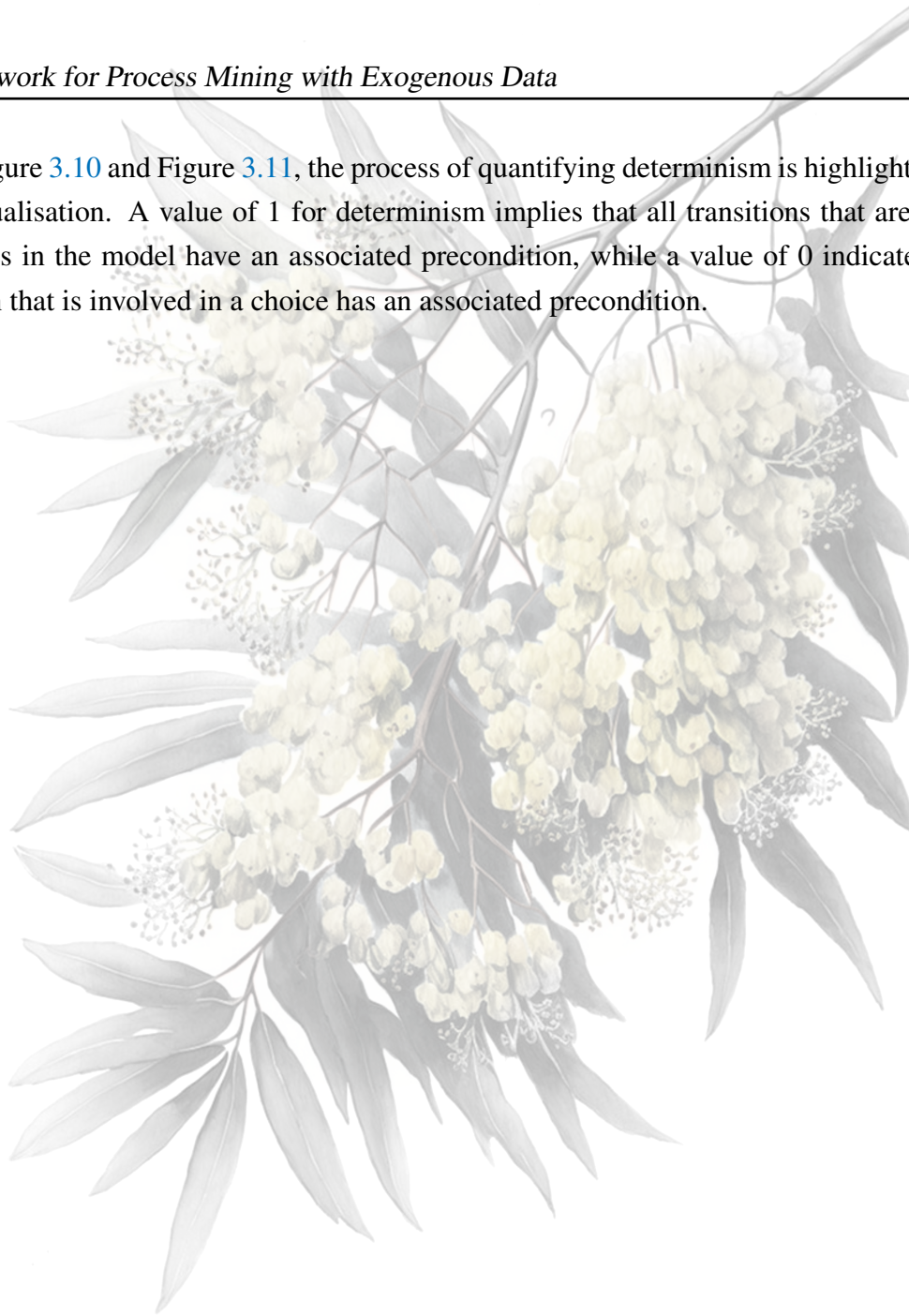
$$\text{decision transitions or } dtrans(N) = \{t \in p^\bullet \mid p \in P \wedge |p^\bullet| \geq 2\}, \text{ and} \quad (3.1)$$

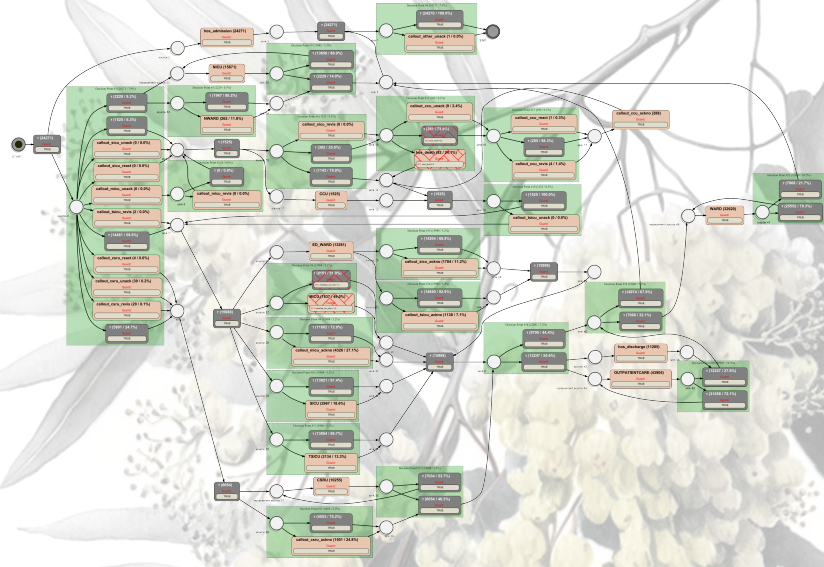
$$\text{associated transitions or } atrans(N) = \{t \in dtrans(N) \mid \Phi(t) \neq \perp \wedge \Phi(t) \neq \text{true}\}. \quad (3.2)$$

Now, determinism $deter$ is introduced as a function that takes N and returns a number between 0 and 1, defined as:

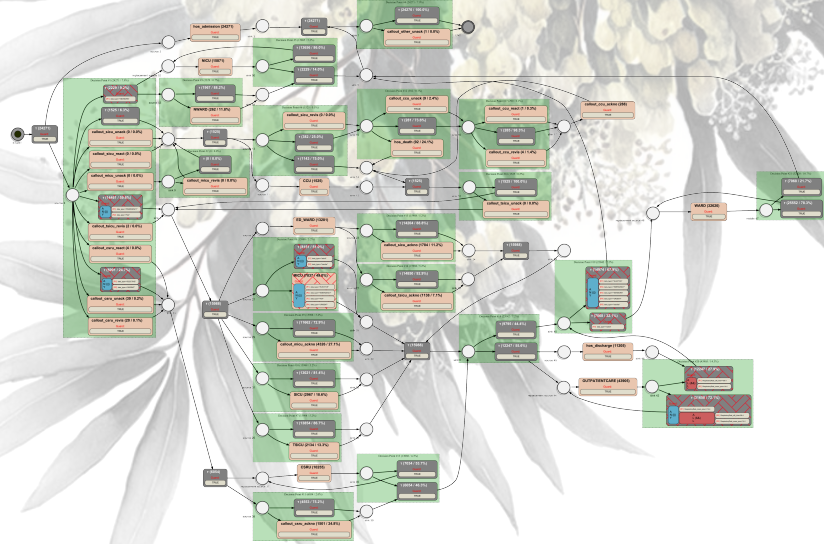
$$\text{Determinism or } deter(N) = \begin{cases} \frac{|atrans(N)|}{|dtrans(N)|} & \text{if } |dtrans(N)| > 0, \\ 1 & \text{otherwise.} \end{cases} \quad (3.3)$$

In Figure 3.10 and Figure 3.11, the process of quantifying determinism is highlighted within each visualisation. A value of 1 for determinism implies that all transitions that are involved in choices in the model have an associated precondition, while a value of 0 indicates that no transition that is involved in a choice has an associated precondition.

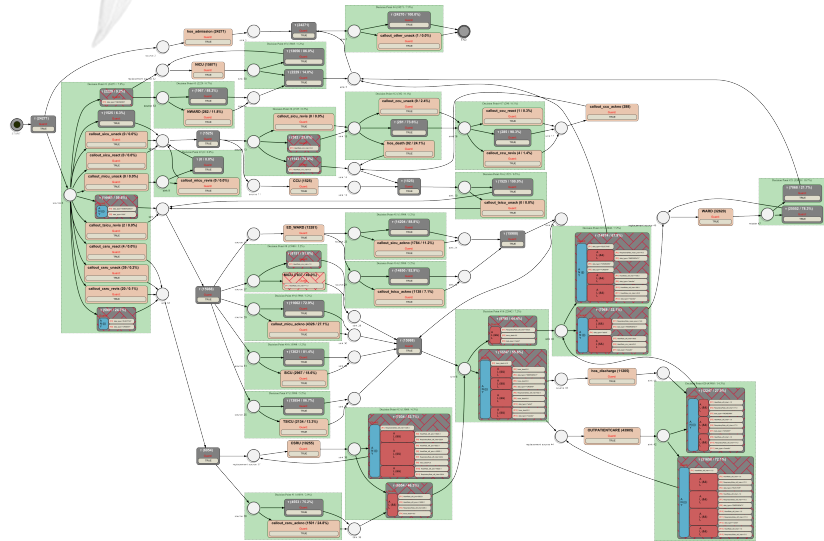




(a) $mi = 0.25$. $|\text{atrans}(N)| = 4$. $|\text{atrans}(N)| = 54$. $\text{deter}(N) = 0.074$.



(b) $mi = 0.15$. $|\text{atrans}(N)| = 9$. $|\text{atrans}(N)| = 54$. $\text{deter}(N) = 0.167$.



(c) $mi = 0.05$. $|\text{atrans}(N)| = 15$. $|\text{atrans}(N)| = 54$. $\text{deter}(N) = 0.278$.

Figure 3.10: The three xDPNs discovered by [57] using both endogenous and exogenous data for the movements log. Transitions that are associated with a precondition are hatched.

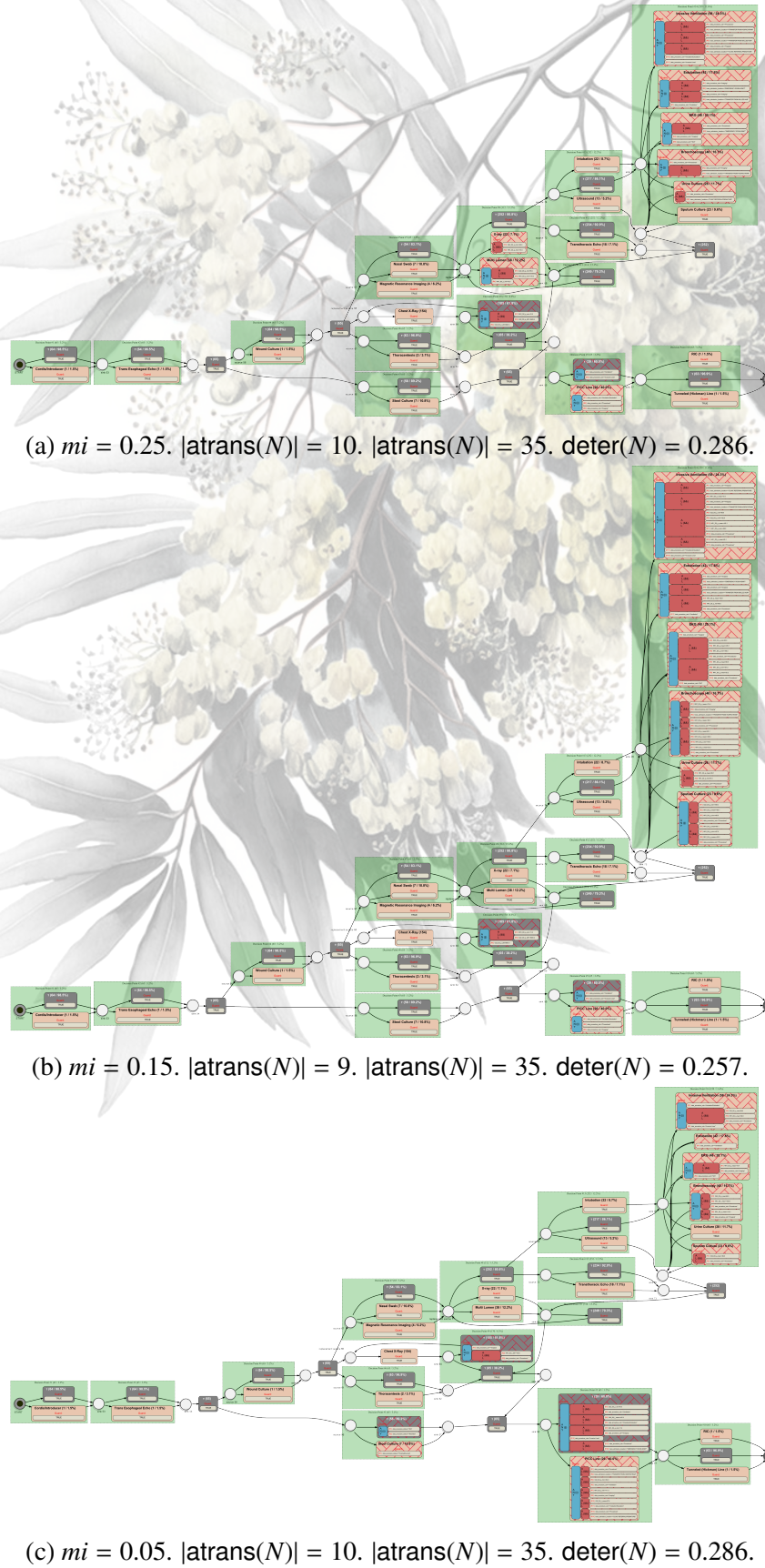


Figure 3.11: The three xDPNs discovered by [58] using both endogenous and exogenous data for the procedures log. Transitions that are associated with a precondition are hatched.

Data

Two event logs from the MIMIC-III dataset were created: a log of patient movements (*movements log*) and a log of procedures for respiratory failures (*procedures log*). The extraction scripts for these two event logs can be found in a public repository⁶. The *movements log* captures the movements of patients between ICU wards, and contains 24 271 traces, 290 462 events, 65 activities and 6 endogenous attributes. The *procedures log* describes the procedures that a patient received, and contains 65 traces, 610 events, 34 event classes and 4 endogenous attributes. Both logs use an exo-description including exo-panels for respiratory rate, heart rate, oxygen saturation, and arterial blood pressure measurements. The *movements log* linked to 25 684 680 exo-measurements; the *procedures log* linked to 590 285 exo-measurements.

Results

Table 3.3 shows the results. The best results for each log appear in **boldface**. When considering the *movements log*, using exogenous data only (exo) does not introduce preconditions in most cases, and henceforth the fitness and precision values are high. In cases where determinism is greater than zero, fitness is very low but precision is competitive. From the results of this log the conclusion is that exogenous data by itself does not suffice. For exo+endo, typically more preconditions are discovered, which lowers fitness and precision (at most 0.11 lower than endo). This decrease in quality is to be expected, as adding more preconditions means that multi-perspective measures will consider more data attributes from the event log, thus increasing the state space on which precision is based.

When considering the *procedures log*, surprisingly, larger values of the parameter *mi* did not always decrease the number of preconditions found as is to be expected as *mi* is a support threshold. The findings suggest that the rather small size of the procedures log and the nature of the overlapping rules (or) algorithm is at play here, which after building a first precondition, does not have enough observations left for a second precondition to meet the *mi* threshold. For this log, using the exogenous data increased the determinism and hence the number of preconditions found (exo+endo and exo vs. endo). Consequently, for endo and endo+exo, fitness goes up with *mi* for me and goes down for or, but in the exogenous variant these patterns are not observed. When considering exo and endo+exo vs. endo, then fitness consistently decreases, precision consistently increases, and determinism consistently increases. These findings suggest that the preconditions cover a larger fraction of the increased state space than for the *movements log*, which has a greater variety of process executions.

The conclusion from this evaluation is that the inclusion of exogenous data can lead to greater determinism, a decrease in fitness, and a possible increase in precision. A caveat to this outcome is that the same exogenous data may have more or less impact on different processes.

⁶ <https://github.com/adambanham/icpm2021>

Table 3.3: Breakdown of measured model quality for discovered xDPNs across two event logs and three variants. The best measurement for an event log is highlighted in bold. In both event logs, including exogenous data lead to a higher determinism.

Variant	Discovery	mi	Movements log			Procedures log		
			fitness	precision	determinism	fitness	precision	determinism
endogenous	me	0.25	0.586	0.644	0.037	0.739	0.395	0.171
		0.15	0.573	0.656	0.130	0.739	0.392	0.171
		0.05	0.573	0.644	0.167	0.761	0.418	0.229
	or	0.25	0.586	0.657	0.037	0.785	0.393	0.200
		0.15	0.583	0.656	0.130	0.785	0.393	0.200
		0.05	0.587	0.647	0.167	0.761	0.419	0.229
endogenous and exogenous	me	0.25	0.575	0.591	0.074	0.692	0.409	0.200
		0.15	0.518	0.537	0.167	0.705	0.411	0.200
		0.05	0.465	0.533	0.278	0.717	0.459	0.286
	or	0.25	0.586	0.649	0.037	0.731	0.445	0.286
		0.15	0.536	0.559	0.167	0.739	0.430	0.257
		0.05	0.465	0.550	0.259	0.717	0.463	0.286
exogenous	me	0.25	0.654	0.640	0.000	0.722	0.413	0.171
		0.15	0.654	0.623	0.000	0.697	0.436	0.171
		0.05	0.173	0.567	0.185	0.709	0.420	0.229
	or	0.25	0.654	0.628	0.000	0.701	0.512	0.371
		0.15	0.654	0.639	0.000	0.697	0.425	0.171
		0.05	0.099	0.582	0.167	0.709	0.424	0.229

In some cases, exogenous data may tell analysts nothing, while in other domains, exogenous data may be the only relevant part. Furthermore, the evaluation did not investigate if the discovered preconditions provide greater insights with exogenous data; it only investigated if using additional data is beneficial to the quality of discovered process models, in a single domain.

The results of the evaluation include some unintuitive observations when considering that preconditions can only make the model more precise. The testing showed that determinism was inversely proportional to the mi parameter used to construct preconditions. This observation makes sense; the lower is mi , then the easier it is to construct guards, leading to determinism increasing. Similarly, the greater is mi , the fewer preconditions are introduced, increasing fitness as the model reverts to a control-flow model. An unintuitive observation from the evaluation was that including more preconditions (making the model more precise) led to a reduction or similar precision. This relationship can occur due to decision mining techniques introducing both preconditions *and* write constraints; the latter also affecting the precision measure. I posit that the quality of the discovered preconditions can be misinterpreted if these conformance measures are naively applied to decision mining outcomes. This revelation led to the need for exploring means for quantifying the quality of models with preconditions in greater detail within Chapter 5.

3.6.2 Exogenous Influences on Process Outcomes

This section demonstrates and evaluates how xPM and EESA visualisations can be used to understand how changes in exogenous data influence process outcomes. In contrast to typical process outcomes from business processes (e.g., a loan application is either accepted or rejected), clinical outcomes in healthcare processes can change throughout a patient's healthcare journey several times or not at all (e.g., a patient may require assistance breathing intermittently during a hospital stay). Thus, this evaluation uses the dynamic characterisation of slices within EESA to study clinical outcomes. The assumption is that clear distinctions within exogenous data could lead to earlier detection of adverse clinical outcomes. Furthermore, the instantiation of xPM was chosen such that the enhancement step could be focused on within the evaluation.

Clinical Outcomes in ICUs

This evaluation focuses on a healthcare scenario considering invasive procedures performed on patients admitted to an intensive care unit (ICU) at a hospital. Intensive care units (ICU) are clinical areas that are designed to manage the most critically ill patients. Critically ill patients are those that have failure of one or more organ systems that require advanced supportive therapies. Patients admitted to ICUs will require one or more intravascular accesses (i.e. intravenous cannulae, central venous catheters, arterial monitoring lines) and are typically managed with an array of monitoring and therapeutic devices.

Sepsis is a condition that is defined by serious organ dysfunction associated with a dysregulated host response, and it has been highlighted by the World Health Organization as a global health priority and a major threat to patient safety [215]. Affecting more than 50 million people each year [216], septic shock is sepsis that is associated with hypotension, lactatemia and a range of etiologies, which are most commonly associated with serious infections.

Infections (e.g., severe pneumonia) are not only among the most common reasons for patients to be admitted to ICUs, but also frequently complicate a patient's stay. Growth of microorganisms in cultures of specimens from patients are typically required to diagnose an infection. Bloodstream infections are those where microorganisms are present in the bloodstream of a patient and may be related to infections at any body site such as the respiratory (pneumonia), gastrointestinal, or genito-urinary tracts.

Recognising patients with early or incubating infections and/or predicting those who will develop subsequent infections/sepsis is a major challenge. As a result of their inherent complexity due to severe illness and the plethora of clinical variables arising from physiologic observations and interventions, usual bedside examination and monitoring of patients will occasionally fail to detect patients at very early stages of infection/sepsis when treatment would be most efficacious. Thus, developing approaches to support decision-making around the early detection of infection/sepsis in critically ill patients is a priority research objective.

To date, technology developments in healthcare centre around the support of clinical decision-

making to improve patient care and outcomes [217]–[219]. Process mining has recently been used to explore the process of care and the diagnostic trajectory of a disease [21], [70], [114], [220]. However, several key issues arise when technology is employed to support clinical decision making in healthcare, such as rule based systems or Bayesian neural networks [219]. These issues include the quality of data to produce accurate and reliable outputs [218], [219]; the lack of mining techniques that can provide understandable, high-level information [217], [218]; and the ethical responsibility to maintain healthcare professionals trust in those outputs [217], [219], [221]. Therefore, this evaluation investigates if EESA visualisations could fill a gap, by providing understandable information to enhance the clarity of a clinical outcome in retrospective analysis.

Procedure

The following scenario will investigate how the combination of the proposed techniques, process behaviour, and exogenous data can identify, within patient vital sign measurements, clinical outcomes related to a sepsis/infection. The instantiation of xPM for this evaluation is as follows:

- $\mathcal{L}$ The patient (`subject_id`), hospital admission (`hadm_id`) and ICU (`icustay_id`) identifiers were used to link all exogenous data sets (except laboratory results) to traces. Laboratory results were linked using patient and hospital admission identifiers.
- $\mathcal{S}$ Four slicing functions were introduced: a slicing function collects all *exo*–measurements either 2, 4, 6 or 12 hours before an event (i.e. a procedure).
- $\mathcal{T}$ A single transformation function was used to determine if a vital sign increased or decreased over time. The slice is used to find the line of best fit for all *exo*–measurements, then returns the gradient of the line as an event attribute for annotation.
- $\mathcal{D}$ Instead of applying a discovery technique, a hand-made process model was used, shown in Figure 3.12. This model used a flower-like pattern to describe the process, and repeated the same process behaviour for different admission intervals (lengths of stay) to help identify different cohorts of patients.
- $\mathcal{E}$ This step is the focus of the experiment. The EESA procedure and the ranking procedure, outlined in Section 3.5 & Section 3.5.2, were applied. Using these contributions, our analysis focused on identifying points of interest between patients associated with sepsis and those that are not.

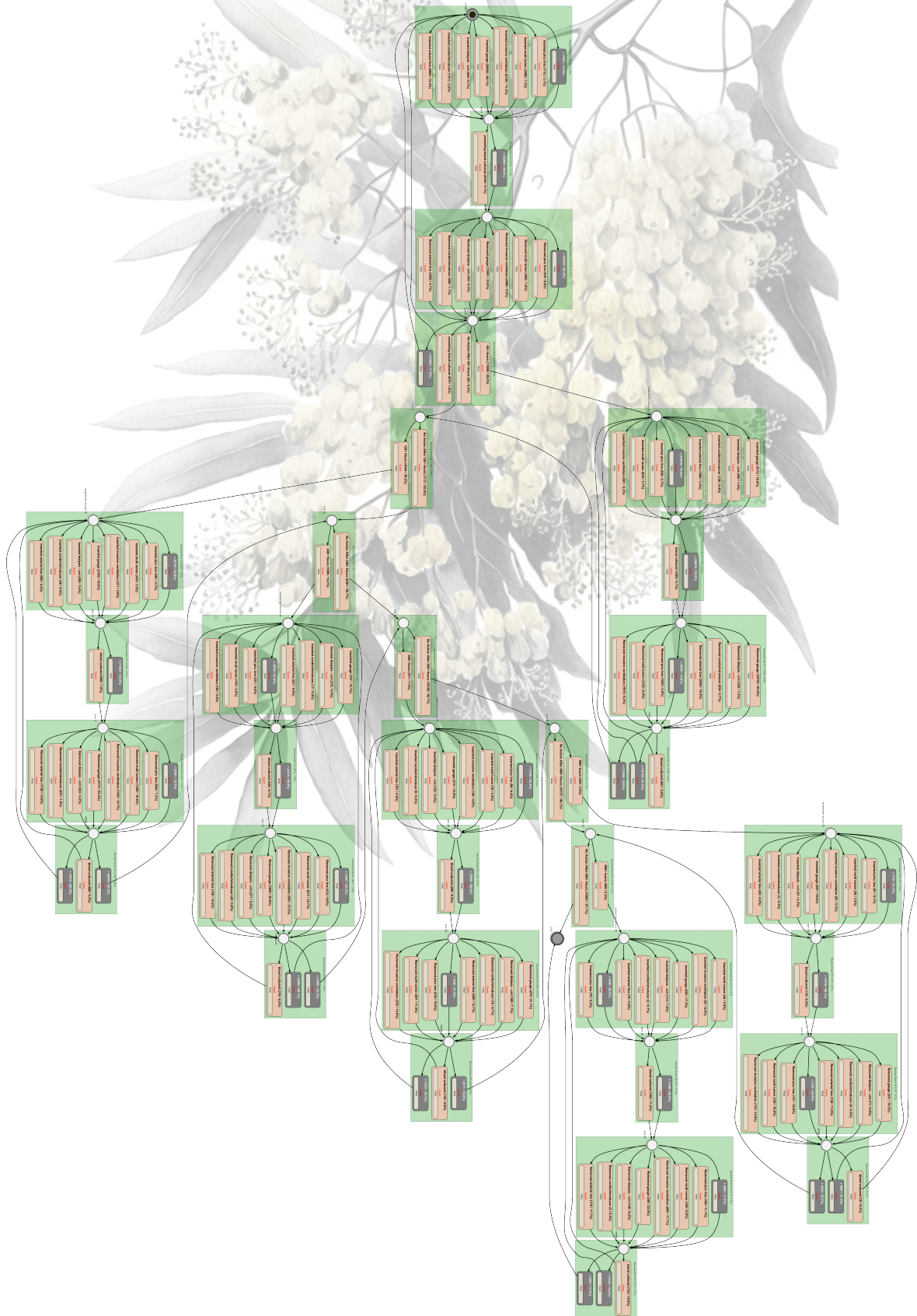


Figure 3.12: Handmade model for analysis of ICU procedures.

Data

This evaluation uses the MIMIC-III dataset [47] to collect ICU admissions into an event log. Patient admissions to an ICU ward were included if: sepsis was not noted at the time of admission to the hospital (noted in admissions table), at admission the patient was at least 18 years old and the ICU stay lasted at least 48 hours. These requirements exclude cases of pre-existing sepsis and focus on sepsis that could be associated with the ICU stay.

This evaluation does not distinguish between ICU wards, resulting in 11,161 admissions being included using the admissions and icustay tables. The following procedures were used as activities from the procedureevents_mv table: mutli-lumen, intravenous cannula (recorded as gauge 18-22), picc line, invasive ventilation, cordis/introducer, dialysis and arterial line. Each event was split into a start and stop event as these require clinicians involvement.

The exo-description included the following exo-panels representing vital sign measurements of patients: heart rate (beats/min), respiratory rate (insp/min), non-invasive blood pressure (mmHg), blood oxygen saturation (SpO2 %) and body temperature (celsius). These exo-panels were extracted from the chartevents table and represent nurses updating the patient's electronic chart.

Also included was an exo-panel for laboratory results from the microbiologyevents table, specifically blood tests, where a blood culture was tested for the presence of organisms. The outcome of these laboratory results were used to identify when a patient had a sepsis/infection during a ICU admission. In total, across six exo-panels, 111,312 exo-series and 18,014,496 exo-measurements were extracted. The extraction scripts for both the endogenous and exogenous data from the MIMIC-III can be found in a public repository⁷.

Results

In total, 2,240 EESA visualisations were produced, where 2028 EESAs described trends between patients with sepsis and those without sepsis. Meaning sepsis was observed in 81.77% of all process activities within admissions. However, not all of these activities had an interesting number of instances where sepsis was detected, i.e., only one or two instances vs. thousands of non-sepsis instances. To explore in greater detail which activities had a noticeable number of sepsis instances when they occurred, a filter on the EESA visualisations was introduced to focus on the diagnostically interesting activities. The applied filter ignored all EESAs where the sepsis group did not contain at least 1% of the overall instances. After its application, only 36.05% of process activities still had an EESA where the sepsis group was large enough to make diagnostic statements. Meaning that roughly a third of the procedures in our model involved sepsis-affected patients, so it was not uncommon to see possible sepsis infections in patients.

The histogram in Figure 3.13 shows the frequency of basic vital sign measures seen in the

⁷ <https://github.com/adambanham/icpm2021>

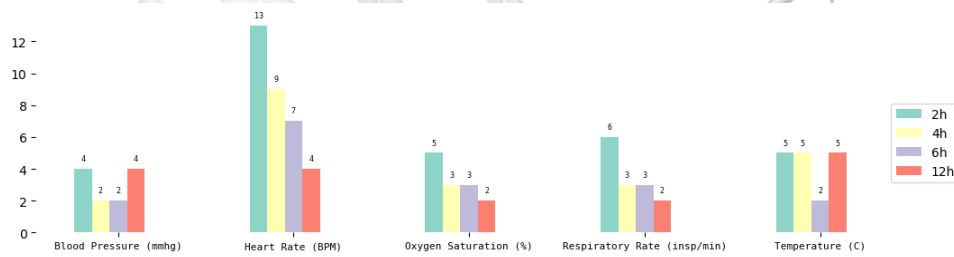


Figure 3.13: Histogram showing the top-ranked vital signs in EESA visualisations (only considering EESA visualisation where at least 1% of procedures were associated with sepsis).

top ranked (rank < 90) EESA visualisations, with at least 1% of procedures were associated with sepsis. The majority of EESAs with a higher level of discrimination, mostly used the heart rate of the patient. Moreover, drilling down to the length of observations, the findings show using two hours of vital sign measurement provides the greatest discrimination between sepsis outcomes across all vital sign measurements.

Figures 3.14 and 3.15 show two EESAs that were ranked highly after applying the proposed ranking procedure (see Section 3.5.2), and both are focused on the procedure named “intravenous cannula”. This activity is interesting to see in the top ranked EESAs as it is a procedure where an external feed line is inserted into the patient, providing fluids or medicine. As such, the activity presents an entry point for bacteria to enter the bloodstream of the patient, which may lead to a blood-borne infection like sepsis. Recall that the used determinations (Section 3.6.2) focus on the vital signs before the activity and that the groups are identified in hindsight; that is, they include lab results from after the activity. Both EESAs show that the heart rate (measured in beats per second, or BPM) of patients before this procedure were elevated compared to the non-sepsis affected patients, which could be used to inform decision making for sepsis identification.

These results demonstrate that this approach can transform clinical data from being clinically uninterpretable (Figure 3.5a) to interpretable and having potential clinical value given further interrogation (Figure 3.14 & Figure 3.15). Figure 3.14 shows an EESA visualisation that converted 20,340 exo-series into a clinically interpretable visualisation. Furthermore, in Figure 3.15, the EESA visualisations shows how those patients who develop infection/sepsis may be discriminated based on heart rate values alone.

Discussion

This analysis was primarily based on standard, routinely measured, vital signs. In the course of a given day, a patient admitted to ICU may be expected to have tens to hundreds of intermittent measurements and thousands of continuous measurements. While clinicians do examine trends, typically most clinical decisions are made on “snap shot” intermittent measures. The possibility exists that patients with slow deterioration in continuously measured parameters (i.e. heart rate) may not be recognised until other measures of worsening (i.e. shock onset) become overt. One

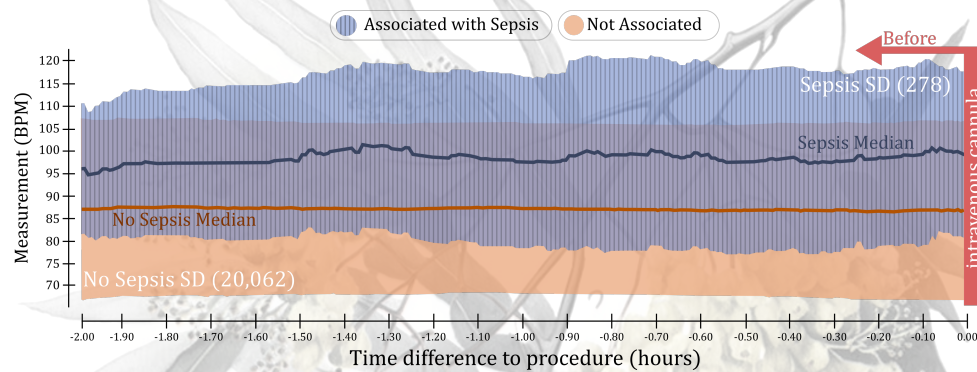


Figure 3.14: A top-ranked EESA of heart rate measurements linked to 20,342 intravenous cannula procedures, of which, 278 procedures were associated with a patient acquiring sepsis. Patients' heart rate measurements were taken from the two-hour window prior to the patient undergoing the intravenous cannula procedure. All procedures occurred within the first 48 hours of admission.

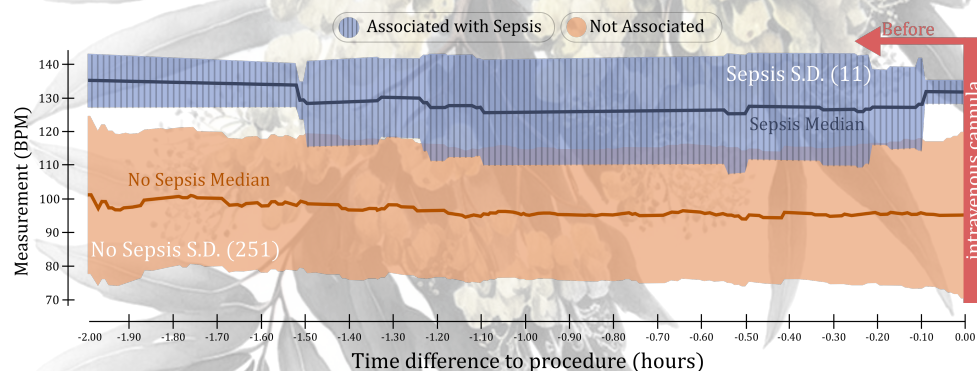


Figure 3.15: A top-ranked EESA of heart rate measurements linked to 262 intravenous cannula procedures, of which, 11 were associated with a patient acquiring sepsis. Patients' heart rate measurements were taken from the two-hour window prior to the patient undergoing the intravenous cannula procedure. All procedures occurred after the patient had been admitted for 25 days. Observe that heart rate may discriminate between patients associated with sepsis.

of the values of EESA is that it integrates the vast number of often seemingly scattered values into patterns that can be of clinical value for decision making and flag patients for an in-depth bedside review of their clinical status.

While the processed data does not, in its current format (EESA visualisations), predict whether or not the patient will develop infection/sepsis, it is important that techniques can demonstrate the discrimination between these two outcomes. The approach allows domain experts and process analysts to investigate a broader interpretation of the context, then refine the interpolation in an iterative manner based on data-driven outcomes. However, further studies are needed to see if the approach would help find variables that have greater discrimination for multivariable predictive modelling.

When reviewing the procedure's ranking, the ranking results included sources with vastly different measurement units and variability, as seen in Figure 3.13. While heart rate was the most common exo-panel included in the top-ranked EESAs, as it does have large variability

naturally, the other exo-panels were present in the histogram. This inclusion of exo-panels may reflect the design choice to have min-max scaling occur on trends before computing the DTW distance measurement used for ranking. However, the results may reflect the distinct nature of sepsis, and the effectiveness of this design choice may not be universal. As such, future work is needed to understand if the design choice allows for an unbiased comparison of exo-panels, but also highlights the potential generalisation.

There are some limitations and assumptions about the presented analysis. Process activities (i.e. arterial line, ventilator, insertion of an intravenous cannula) were selected in a somewhat arbitrary fashion to explore the potential use of the contributions. Furthermore, a very liberal understanding of the process structure (process model used for $\mathcal{D}$), to emphasise the analysis of EESA visualisations was used. The evaluation also assumed that data quality issues had not affected analysis and the extraction of vital sign measurements was representative. Finally, the identification of patients with sepsis using a blood culture test was assumed to be effective and provide a reliable understanding of this clinical outcome.

The findings from this evaluation suggests that EESA can provide clarity in an otherwise noisy analysis area for healthcare professionals. Further study is needed to evaluate the feasibility of deploying the contributions within a clinical setting. For example, the complexity of an approach could be a reason for non-acceptance. The adoption of an approach may be subject to a detailed cost benefit analysis. However, it is imperative that insights must be comprehensible by healthcare professionals in order to build trust and acceptance.

3.7 Reflections and Future Work

This section reflects on development of xPM and EESA presented in this chapter to highlight future work. Best efforts have been made to ensure the generalisation of the investigation presented in this chapter, but some limitations exist. The xPM framework was derived from scientific literature rather than interviews, survey data, or informed by industry needs and so its practical benefits for society at large are not considered here. The process mining existing literature [18], [52]–[54] shows that owners and actions of processes are looking for techniques that cater for contextual information. Therefore xPM was motivated by literature and developed to primarily offer an avenue for researchers to formally approach the use of time series data within process mining for algorithmic contributions.

In the most simplistic view of the xPM framework, it should be acknowledged that the contribution may be boiled down to adding a single attribute to each event per determination. However, knowing how to convert a complex time series into a single static attribute is non-trivial and rarely realistic [222]. Future studies should consider whether the xPM framework captures the general case of including temporal data for process mining efforts.

The enhancement technique for xPM, Explorative Exogenous Series Analysis (EESA), presented in this chapter also has some limitations. The visualisation tool does requires some

domain knowledge or iterative explorative analysis to define group identifiers for the overlapping variant. EESA presented in this chapter focus on numerical signals presented as time series data. Investigating categorical signals or how these could be handled to produce an EESA was not undertaken, future work may research how to incorporate categorical signals so a larger class of exogenous influences can be explored.

This chapter focused on the healthcare domain, as a clear connection between endogenous and exogenous data could be presented through the publicly available dataset, MIMIC III [47]. Further extracts from the same location, i.e. MIMIC IV [223], could be used to revisit the findings in Section 3.6 with a comparison of samples (event logs) undertaken from these extensions. It is challenging to provide a clear use case for process mining with exogenous data without a clear connection between endogenous and exogenous data. The development of techniques to explore exogenous influences, as presented here do not clarify whether a statistical relationship exists between the given endogenous and exogenous data. Future work may explore if such a statistical relation could be used to filter exo-panels that may only include noise for the analysis.

There are many possibilities to extending the contributions of this chapter. Firstly, the semantics of xDPN could be expanded to introduce a means for expressing temporal changes in data, so that exogenous data can be presented alongside the process structure. Secondly, new decision mining techniques could be developed which have operators in preconditions that consider temporal sequences instead of merely event attribute values. These extensions, while clearly useful in the medical domain, would be applicable to other domains such as business processes, transport processes and manufacturing processes. Lastly, many opportunities exist for process enhancement techniques to take advantage of xPM and provide new types of analysis for exploring exogenous influences.

3.8 Summary

This chapter presented xPM and EESA to enable process mining with exogenous data, which analyses the associations between exogenous data and the behaviour of processes. The approaches were evaluated from the two perspectives of a process mining analyst, and that of a clinician. The evaluation demonstrated that xPM can provide automated analysis, and can find useful insights within exogenous data for processes. Furthermore, evaluations of EESA showed that contributions can be used to integrate large amounts of physiologic data and interventions, providing clinically interpretable information. This chapter sets the stage for further studies aimed at investigating exogenous influences on processes and decision making occurring within processes at runtime.

Statement of Author's Contributions

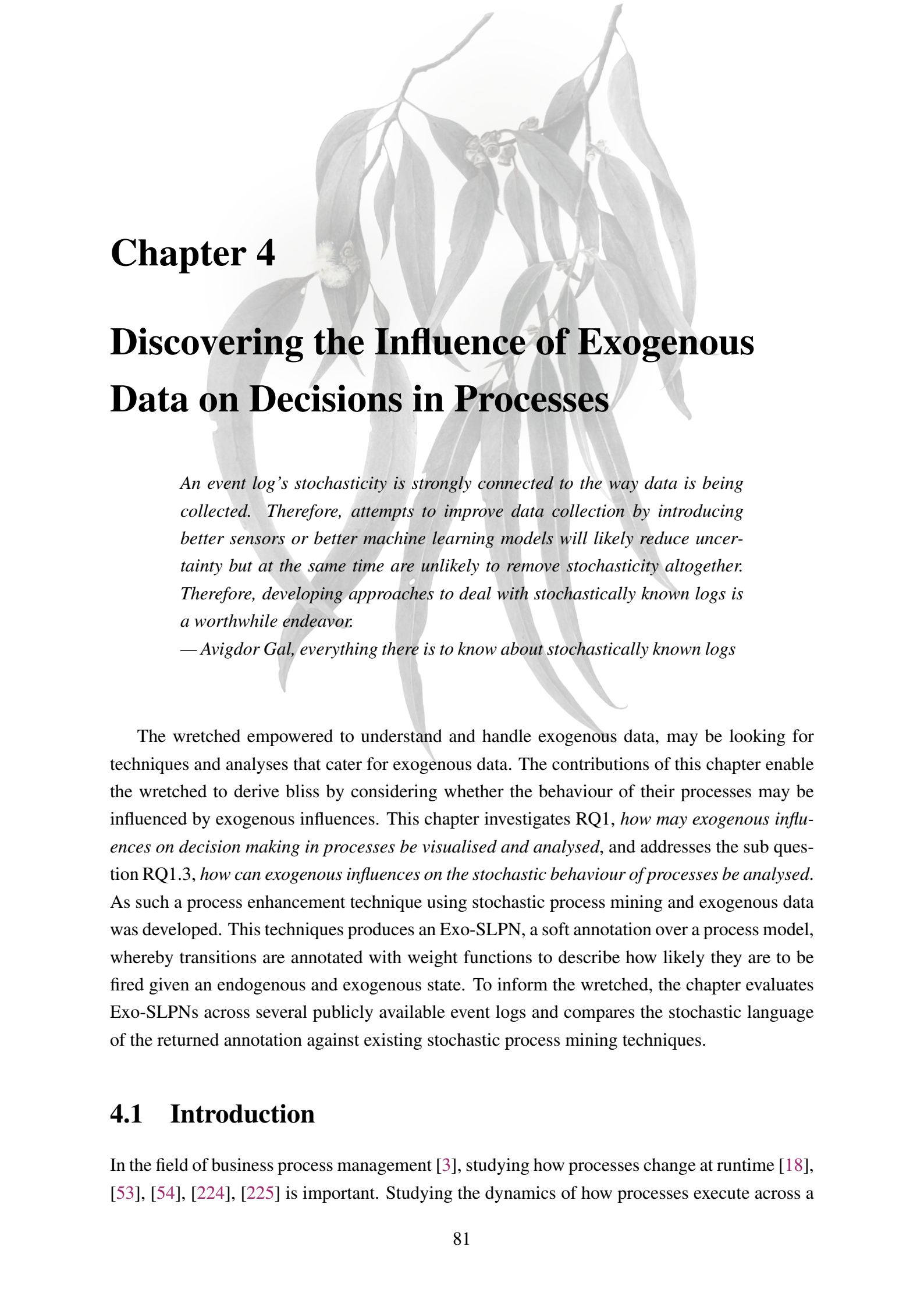
The authors listed below have certified that:

1. they meet the criteria for authorship and that they have participated in the conception, execution, or interpretation, of at least that part of the publication in their field of expertise;
2. they take public responsibility for their part of the publication, except for the responsible author who accepts overall responsibility for the publication;
3. there are no other authors of the publication according to these criteria;
4. potential conflicts of interest have been disclosed to (a) granting bodies, (b) the editor or publisher of journals or other publications, and (c) the head of the responsible academic unit, and
5. they agree to the use of the publication in the student's thesis and its publication on the QUT's ePrints site consistent with any limitations set by publisher requirements.

In the case of this chapter: "Discovering the Influence of Exogenous Data on Decisions in Processes" was accepted and will be published in the proceedings for Application and Theory of Petri Nets and Concurrency 2025, Lecture Notes in Computer Science, 2025.

To describe contributions, the CRediT taxonomy [190] is used for classifications.

Contributor	Statement of Contribution
Adam Banham (Student)	Conceptualisation; Methodology; Software; Validation; Formal Analysis; Investigation; Writing – Original Draft; Writing – Review & Editing; Visualisation.
Yannis Bertrand	Resources; Data Curation; Validation; Visualisation.
Robert Andrews	Methodology; Writing – Review & Editing; Supervision.
Moe T. Wynn	Methodology; Writing – Review & Editing; Supervision.
Sander J. J. Leemans	Conceptualisation; Methodology; Software; Validation; Investigation; Writing – Original Draft; Writing – Review & Editing; Supervision.



Chapter 4

Discovering the Influence of Exogenous Data on Decisions in Processes

An event log's stochasticity is strongly connected to the way data is being collected. Therefore, attempts to improve data collection by introducing better sensors or better machine learning models will likely reduce uncertainty but at the same time are unlikely to remove stochasticity altogether. Therefore, developing approaches to deal with stochastically known logs is a worthwhile endeavor.

— Avigdor Gal, *everything there is to know about stochastically known logs*

The wretched empowered to understand and handle exogenous data, may be looking for techniques and analyses that cater for exogenous data. The contributions of this chapter enable the wretched to derive bliss by considering whether the behaviour of their processes may be influenced by exogenous influences. This chapter investigates RQ1, *how may exogenous influences on decision making in processes be visualised and analysed*, and addresses the sub question RQ1.3, *how can exogenous influences on the stochastic behaviour of processes be analysed*. As such a process enhancement technique using stochastic process mining and exogenous data was developed. This technique produces an Exo-SLPN, a soft annotation over a process model, whereby transitions are annotated with weight functions to describe how likely they are to be fired given an endogenous and exogenous state. To inform the wretched, the chapter evaluates Exo-SLPNs across several publicly available event logs and compares the stochastic language of the returned annotation against existing stochastic process mining techniques.

4.1 Introduction

In the field of business process management [3], studying how processes change at runtime [18], [53], [54], [224], [225] is important. Studying the dynamics of how processes execute across a

collection of varying scenarios, enables businesses to plan for future events based on the likelihood of occurrence of these events. Process mining [2] techniques enable businesses to derive data-driven insights about the behaviour of their processes by extracting and analysing historical data within their information about processes. Process discovery techniques generate process models that show the activities that can occur in a process and how they are structured [2]. Conformance checking quantifies the degree of commonality of behaviour allowed by the model and behaviour observed in an event log [41]. But, to understand how likely an activity is to occur at a given point a different approach such as stochastic process mining is used [73].

Stochastic process mining provides a window into understanding process-level decisions by describing which activities could occur next, and how likely these activities are to occur [73]. Various mechanisms for determining next activity likelihood exist including basing the likelihood on the enabled activities [76], [183], [184] with recent extensions also considering the dependency on trace history [184] or arbitrary case data [76]. Quantifying the likelihood of an activity may require the consideration of many different dependencies, both internal and external of the process. Addressing this challenge with *stochastic process mining* [76], [184], [226] is ongoing. This chapter explores stochastic process mining by investigating whether the historical executions of a process were influenced by external data sources, i.e. *exogenous data* [24].

Process behaviour may have many stochastic dependencies on exogenous data. For example, many processes are influenced by their context [18]: agriculture supply chains can be influenced by external factors like climate variability [19], [71]; patient care in hospitals considers the continuous observation of the patient's vital signs [24], [70]; and even handling insurance claims can have dependencies on resource bottlenecks occurring at the inter-case level [13]. Notably, for these cases cases, the techniques available to date do not capture the dependency of these exogenous influences on the likelihood of an activity in a process.

The contributions of this chapter have a common aim with the work of Mannhardt, Leemans, Schwanen, *et al.* [76] which focused on including endogenous data in the firing likelihoods of activities. This chapter focuses on exogenous data to cater for contextual factors that may influence the behaviour of processes, rather than focusing on endogenous data. Exogenous data considers time series measurements of a single concept, or an *exo-panel* [24], like temperature, blood pressure, or the number of free printers in the firing likelihoods. Handling time series data introduces a temporal component, not considered by [76] and would typically require domain knowledge to process to a static value [24]. Therefore, this chapter extends the state of the art to investigate exogenous data [24] when quantifying the likelihood of activities.

In this chapter, Section 4.2 presents the problem space, Section 4.3 outlines the contributions, Section 4.4 denotes the notation used, Section 4.5 presents the stochastic modelling formalism used, Section 4.6 defines a discovery method for such a formalism, Section 4.7 introduces how to quantify the discovered nets, Section 4.8 evaluates the contributions, Section 4.9 reflects on the chapter, and Section 4.10 summarises the chapter.

4.2 Motivating Example

This section establishes the use of stochastic process mining and highlights the existing techniques [76], [183], [184] and their limitations. A running example using a sample of traces from road fines log [25] (Table 4.1) and a normative model for the log (from [55, Sec 12.1.3]) is introduced as well as the optimal alignments (see Sections 2.4.1 and 4.4) for the snippet, shown in Figures 4.2, 4.3 and 4.4. These alignments are important for stochastic process mining as the probability of transitions is based on the traversed markings for the execution. The motivation for including this example is to explore the probability of ‘Sent to Credit Collection’ (abbreviated to **credit collection**) occurring. When this activity occurs in the process, the fine is set aside and a separate credit collection process is triggered (which not visible in the event log) to recover the fine. The answer for the trace ‘A8690’ is straightforward; as the transition for credit collection is never enabled, it was 0%. However, the question of the probability of credit collection is more challenging to answer for the other traces.

For the trace ‘N74775’ in Table 4.1 and the alignment in Figure 4.2, credit collection is enabled but never fired during the execution. Several stochastic techniques can be applied to understand the probability of credit collection when it was enabled for this trace. One approach is to consider the frequency of the fired transitions [183] to estimate a weight for each transition such as in Figure 4.5. Following estimation, the probability of credit collection is its weight over the sum of weights for all enabled transitions. A drawback of this approach is that long-term dependencies between transitions or dependencies outside of the control-flow perspective are not included. In this case, credit collection has a weight of 59,013, and the weight of all enabled transitions is 185,115 meaning that the probability of credit collection using this approach is 31.87% for all three times that it is enabled. Another drawback of this approach [183] is the probability is constant over the temporal domain and cannot account for the fine being overturned, which occurs when the ‘dismissal’ attribute is set to anything other than ‘NIL’.

Further information about the running execution can be used to contextualise the probability of transitions using *parameterised* weight functions. Parameterised weight functions can cater for long-term dependencies between activities [184] as well as dependencies on data values for activities [76]. An example of a process modelling formalism that uses parameters is stochastic labelled Petri nets with stochastic dependencies [184] (SLPN-SD) shown in Figure 4.6. In SLPN-SDs, transitions are associated with a base weight, and several parameters for the previous actions in the running execution.

SLPN-SDs are helpful as their semantics can cater for when a transition is unlikely based on the already fired transitions. As such, the ‘Appeal to Judge’ transition in Figure 4.6 is associated with a base weight of 0.002 and has several non-one parameters for two of the ‘Payment’ transitions (negative relationship, 0.002), ‘Add Penalty’ (positive relationship, 374.688), and itself (negative relationship, 0.002). Note that before any transitions are fired, these parameters resolve to 1 as any number n (including 0) raised to the power of 0 is 1, i.e. $n^0 = 1$.

Next, consider the partial alignment for trace ‘N74775’ of $\langle (e_1^{e_1}), (e_2^{e_2}), (e_3^{e_3}) \rangle$ which results in a

Table 4.1: An extract from the road fines log [25].

trace	event	event attributes or <i>endogenous data</i> process activity	amount	dismissal	pay amount	timestamp
N74775	1	create fine	35.0	NIL		28.05.2005@00:00
	2	send fine				21.09.2005@00:00
	3	insert fine notification				01.10.2005@00:00
	4	add penalty	71.5			30.11.2005@00:00
	5	insert date appeal to prefecture				30.11.2005@00:00
	6	send appeal to prefecture		#		09.01.2006@00:00
	7	receive result appeal from prefecture				17.02.2006@00:00
	8	notify result appeal to offender				31.03.2006@00:00
A8690	9	create fine	36.0	NIL		08.02.2007@00:00
	10	payment			36.0	21.02.2007@00:00
A21188	11	create fine	22.0	NIL		14.07.2008@00:00
	12	send fine				10.12.2008@00:00
	13	insert fine notification				18.12.2008@00:00
	14	add penalty	44.0			16.02.2009@00:00
	15	send for credit collection				15.10.2010@00:00
⋮	⋮	⋮	⋮	⋮	⋮	⋮

state where the following transitions are enabled: ‘Payment’; ‘Insert Date Appeal to Prefecture’; ‘Appeal to Judge’; ‘Add Penalty’; a silent action (τ); and **credit collection**. In this state, the firing likelihood of credit collection is 19.99%, given by $1/(1+1+1+1+1+(0.002 \cdot 0.002^0 \cdot 0.002^0 \cdot 374.688^0 \cdot 0.002^0)) = 0.1999$. However, after the ‘Notify Result Appeal to Offender’ transition has been fired and credit collection is re-enabled, it has a new probability of 17.39%. In this case, the probability of credit collection differs between markings that enabled it. However, the small change in the probabilities does not reflect the fact that the fine has been dismissed.

Another approach is to use stochastic labelled data Petri nets [76] (SLDPN), which incorporates the endogenous data of the execution into the weight of transitions via a logistic function. The logistic function can incorporate when fines are overturned through one-hot-encoding of literal values into either 0 or 1. However, this approach is limited in that only linear combinations of relationships are discovered, and complex relationships may not be able to be derived. A second issue with this approach is the one-hot-encoding of non-numerical attributes complicates the visualisation of the SLDPN as seen in Figures 4.7 and 4.8.

Consider the example of the two SLDPNs in Figures 4.7 and 4.8. Figure 4.7 was discovered using the attributes in Table 4.1, while Figure 4.8 was discovered with an additional attribute ‘waiting’ denoting the hours since the last event. Note that the weight function for a transition in an SLDPN and calculated as follows: $\frac{1}{1+e^{-f(d)}}$ where f is the fitted logistic function and d is the data state used to evaluate f to a numerical value. In Figure 4.7, the base weight of credit collection is -80.427 (intercept β_0 of the logistic function), and both (fine) ‘amount’ and ‘pay amount’ attributes have small positive relations. However, the weight heavily depends on the value of ‘dismissal’ as it swings between overly negative to somewhat neutral or positive. For trace ‘N74775’, the data state when credit collection is first enabled is {amount $\mapsto$ 35, pay amount $\mapsto$ 0, dismissal $\mapsto$ NIL} and the state when credit collection is enabled for the third time is {amount $\mapsto$ 71.5, pay amount $\mapsto$ 0, dismissal $\mapsto$ #}. In these states,

credit collection has weights of 0.265 and 4.203⁻⁵⁶ and probabilities of 8.77% and roughly 0%, respectively. For trace ‘N74775’, it has become clear the credit collection was unlikely, but the same does not apply to trace ‘A21188’ as in Table 4.1.

The difference between trace ‘N74775’ and trace ‘A21188’ is that the latter is not dismissed, and the duration of execution spans two batches of credit collections (see Figure 4.13). For trace ‘A21188’, credit collection is enabled twice and using Figure 4.7, its weight is 0.253 and 0.273 with probabilities of 8.17% and 9.20% respectively. In this case, the change in probabilities is limited to the changes, if any, in the endogenous data and the temporal factors or the impact of other concurrently running executions are not included. One could ignore the semantics of the underlying Petri net with Data, and include the time between events (i.e. Figure 4.8), to consider if time influences the likelihood. In doing so, the weights of credit collection are changed to 0.634 and 1.0, where the probabilities are 33.46% and 33.33%, showing the probability of credit collection is much larger than before. Importantly, introducing ‘waiting’ means that any transition with a positive relationship will reach an equilibrium with others in terms of probability after some point in time. Figure 4.1 demonstrates this phenomenon through the probability of credit collection for traces ‘N74775’ and ‘A21188’.

This example shows existing techniques can cater for many endogenous factors that may influence the probability of a transition firing. Techniques can capture complex relationships, such as long term dependencies within activities or dependencies on the values of endogenous data. However, without a transition firing, these techniques are unable to adjust the probability of transitions or account for temporal considerations like exogenous data. Therefore this chapter explores both how to present exogenous influences on the firing likelihood of transitions, and the challenges of including exogenous data in the discovery of stochastic nets.

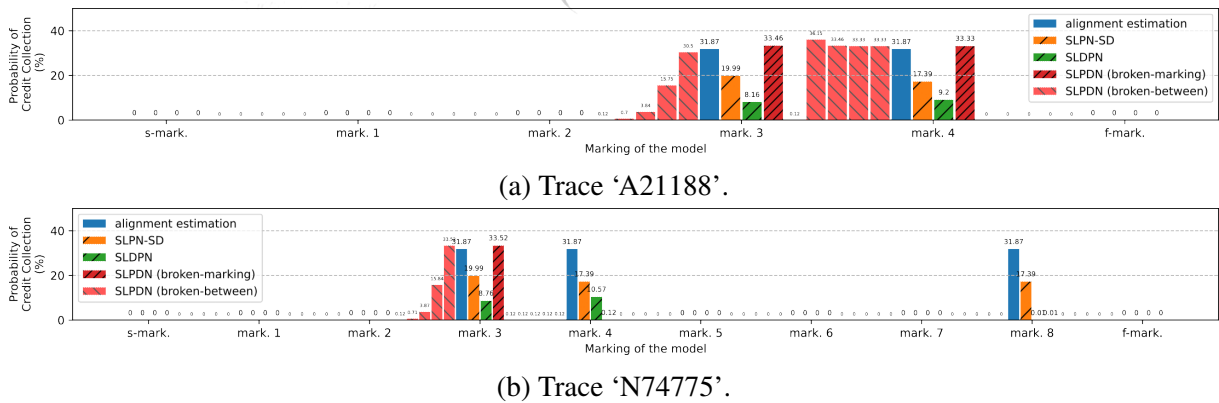


Figure 4.1: Probability of credit collection for two traces across the markings traversed using equally distanced samplings between markings for the SLDPN with ‘waiting’.

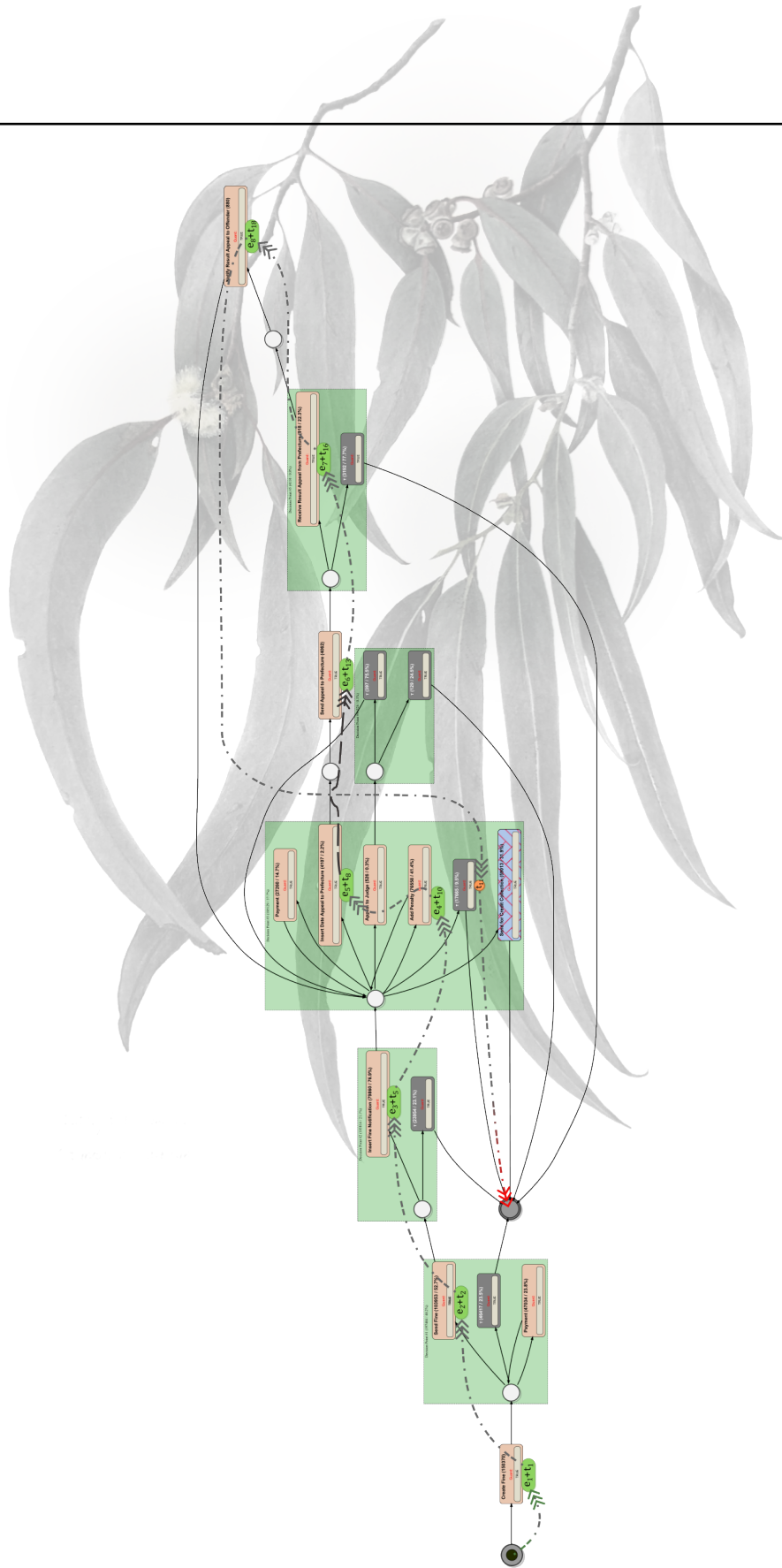


Figure 4.2: The normative model capturing the decision points (in green) and highlighting credit collection in blue and hatching. The solid directed arcs denote the dependencies of the underlying Petri net, while the dashed directed arcs denote the sequence of moves in the alignment. Moves in the alignments that could have fired credit collection are highlighted in the sub captions. Capturing alignment for trace ‘N74775’, $\langle (e_1^{t_1}), (e_2^{t_2}), (e_3^{t_5}), (e_4^{t_{10}}), (e_5^{t_8}), (e_6^{t_{13}}), (e_7^{t_{16}}), (e_8^{t_{18}}), (\gg^{t_{11}}) \rangle$.

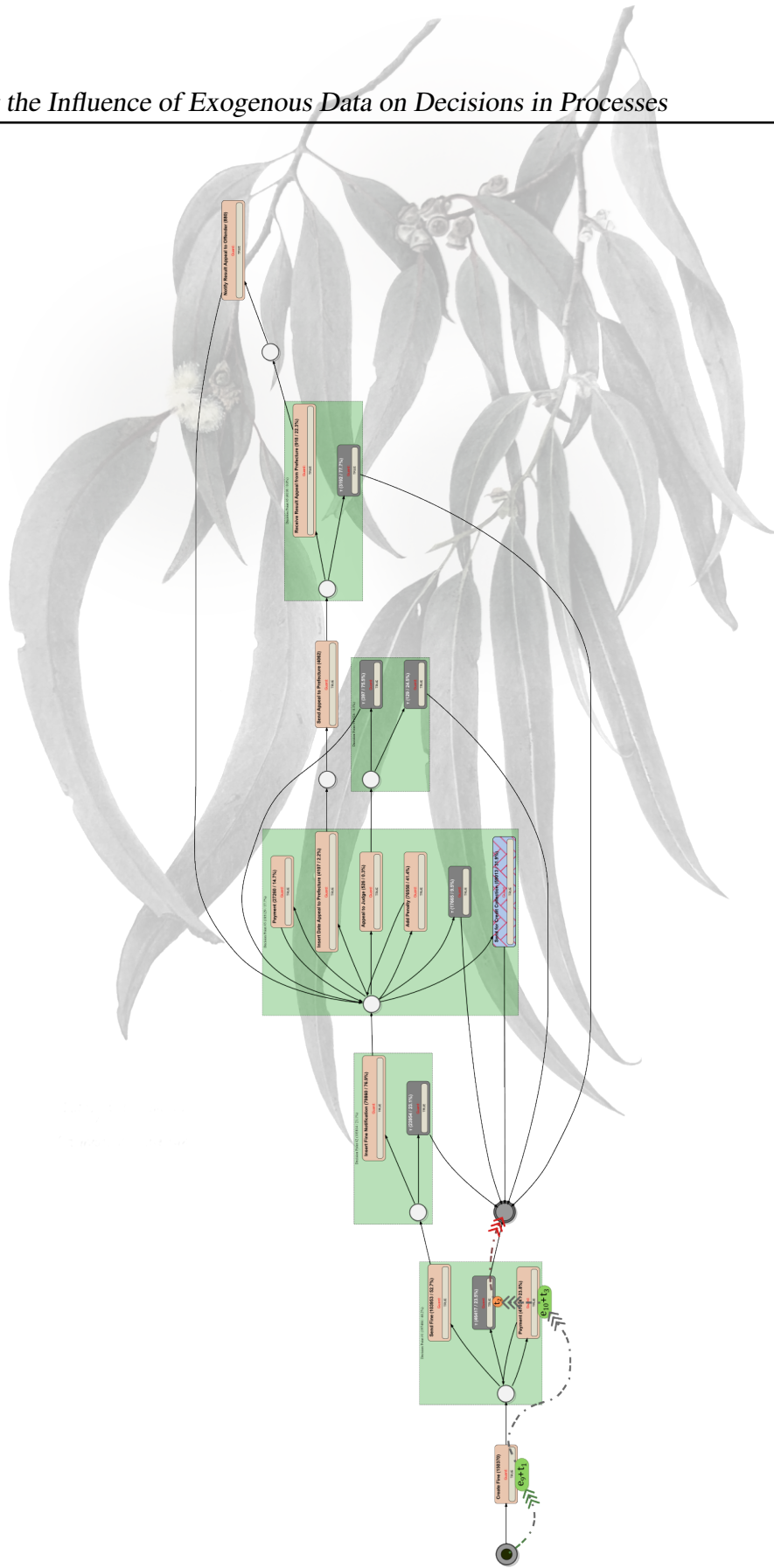


Figure 4.3: The normative model capturing the decision points (in green) and highlighting credit collection in blue and hatching. The solid directed arcs denote the dependencies of the underlying Petri net, while the dashed directed arcs denote the sequence of moves in the alignment. Moves in the alignments that could have fired credit collection are highlighted in the sub captions. Capturing alignment for trace ‘A8690’, $\langle (e_9, t_1), (e_{10}, t_3), (e_{12}, t_2) \rangle$.

Figure 4.4: The normative model capturing the decision points (in green) and highlighting credit collection in blue and hatching. The solid directed arcs denote the dependencies of the underlying Petri net, while the dashed directed arcs denote the sequence of moves in the alignment. Moves in the alignments that could have fired credit collection are highlighted in the sub captions. Capturing alignment for trace ‘A21188’, $\langle (e_{t_1}^{e_{11}}), (e_{t_2}^{e_{12}}), (e_{t_5}^{e_{13}}), (e_{t_{10}}^{e_{14}}), (e_{t_{12}}^{e_{15}}) \rangle$.

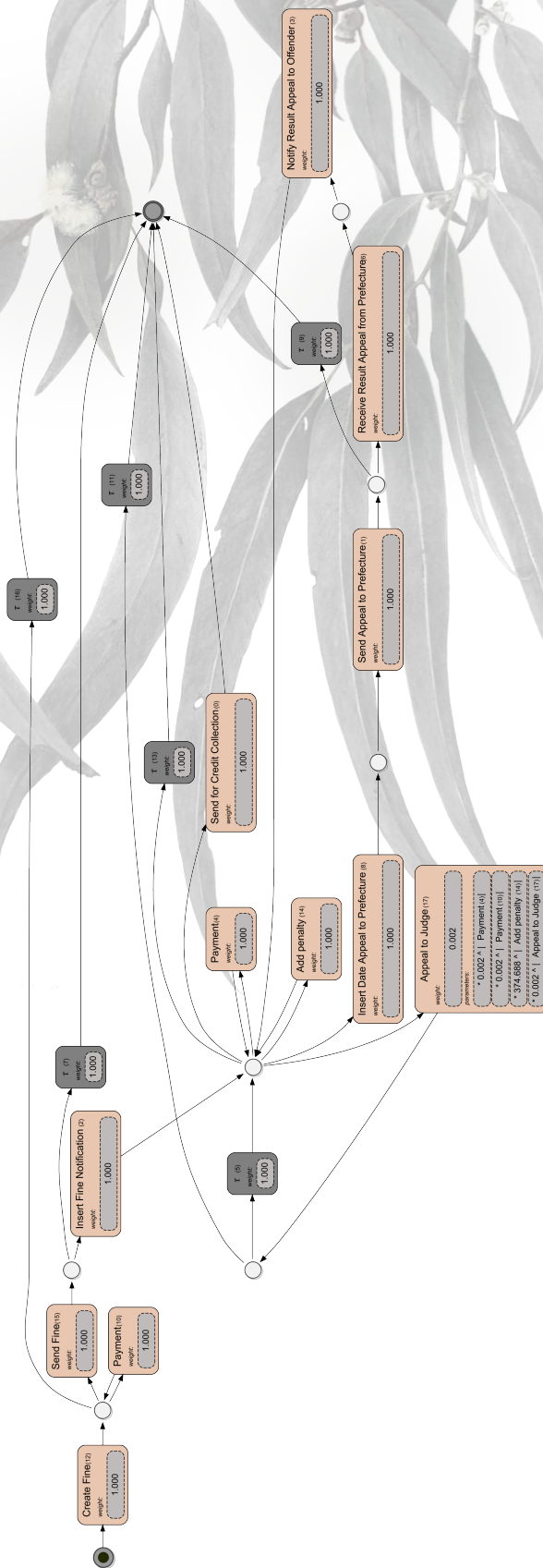


Figure 4.6: Discovered SLPN-SD [184, Definition 1] and uses base weights and parameters.

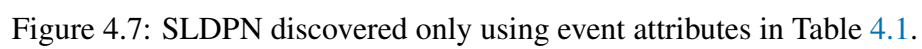


Figure 4.8: SLDPN discovered using event attributes in Table 4.1 and the waiting time between events.

4.3 Approach

This chapter's investigation serves to advance stochastic process mining by developing a preliminary step to determine the exogenous influences that exist before performing further analysis using exogenous data. The developed approach advances the state-of-the-art of stochastic process mining [73], [76], [183], [184] to include exogenous influences [24] by developing stochastic labelled Petri nets with exogenous dependencies, or Exo-SLPN. This process modelling formalism enables the study of exogenous data and its impact on the probability of transitions firing. Furthermore, the chapter proposes an approach for discovering Exo-SLPNs using alignments and a given control-flow process model. This approach considers the impact of model moves within alignments when deriving information for discovery and the challenges of including a temporal dimension. The approach is evaluated against three existing techniques [76], [183], [184], and across three publicly available datasets [25], [47], [227].

The chapter also develops a visualisation tool, demonstrated in Figures 4.9, 4.10 and 4.11 to investigate the changes of firing likelihoods for a given log and a discovered Exo-SLPN. The visualisations show the less likely (redder bars) activities, or more likely (greener bars) activities based on exogenous data, and allow for many interesting observations on the exogenous influences on process activities. These visualisations can be used to consider if the exogenous data has had any influence, where bars are not all at zero change (middle) of the visualisation. For example, does the number of unpaid fines affect how likely credit collection is to occur within the road fines log [25]? This running example continues from Section 4.2, and is used throughout the chapter to illustrate the discovery process for Exo-SLPNs.

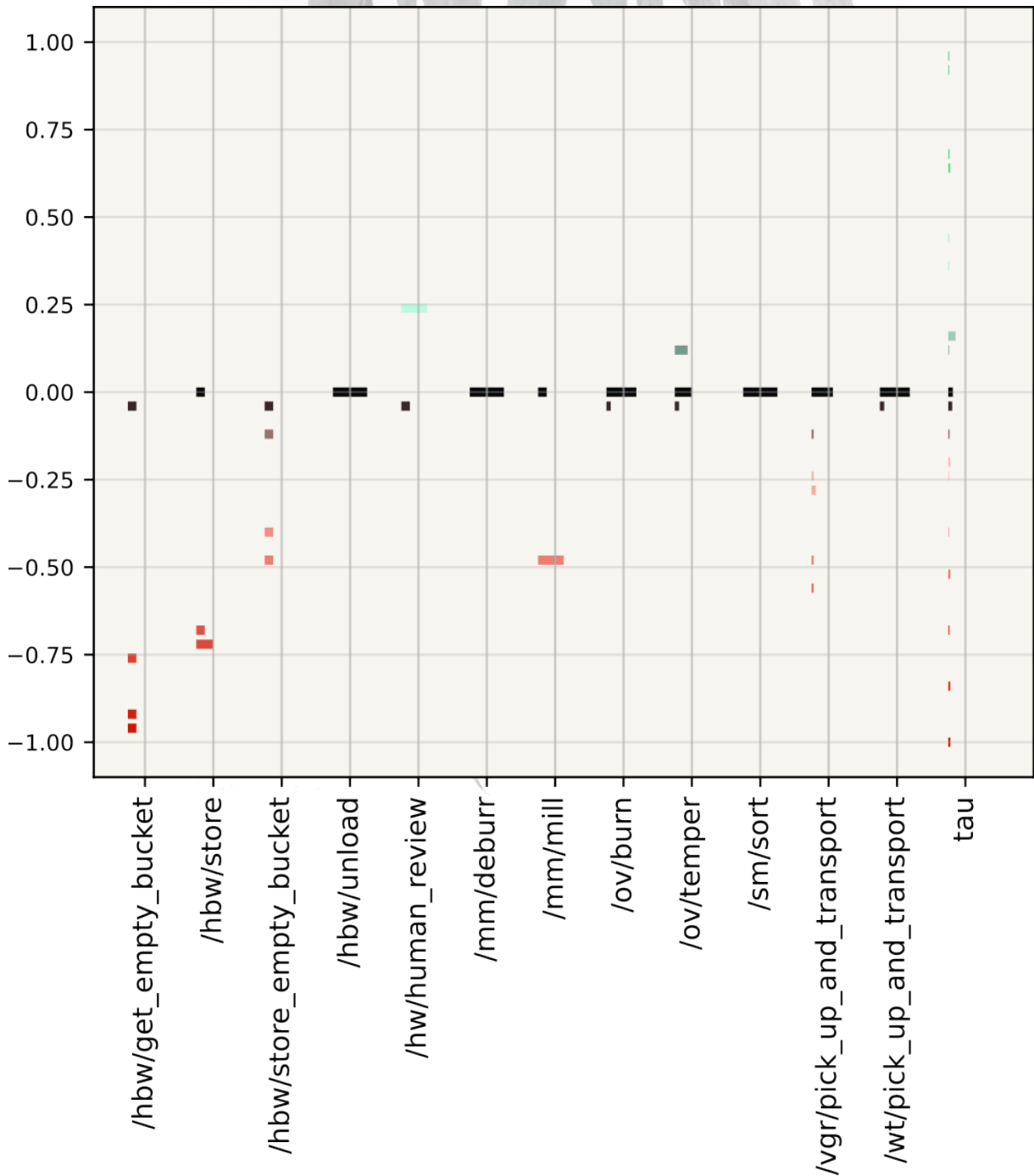


Figure 4.9: Changes in activity likelihoods for a discovered Exo-SLPN for the smart factory log.

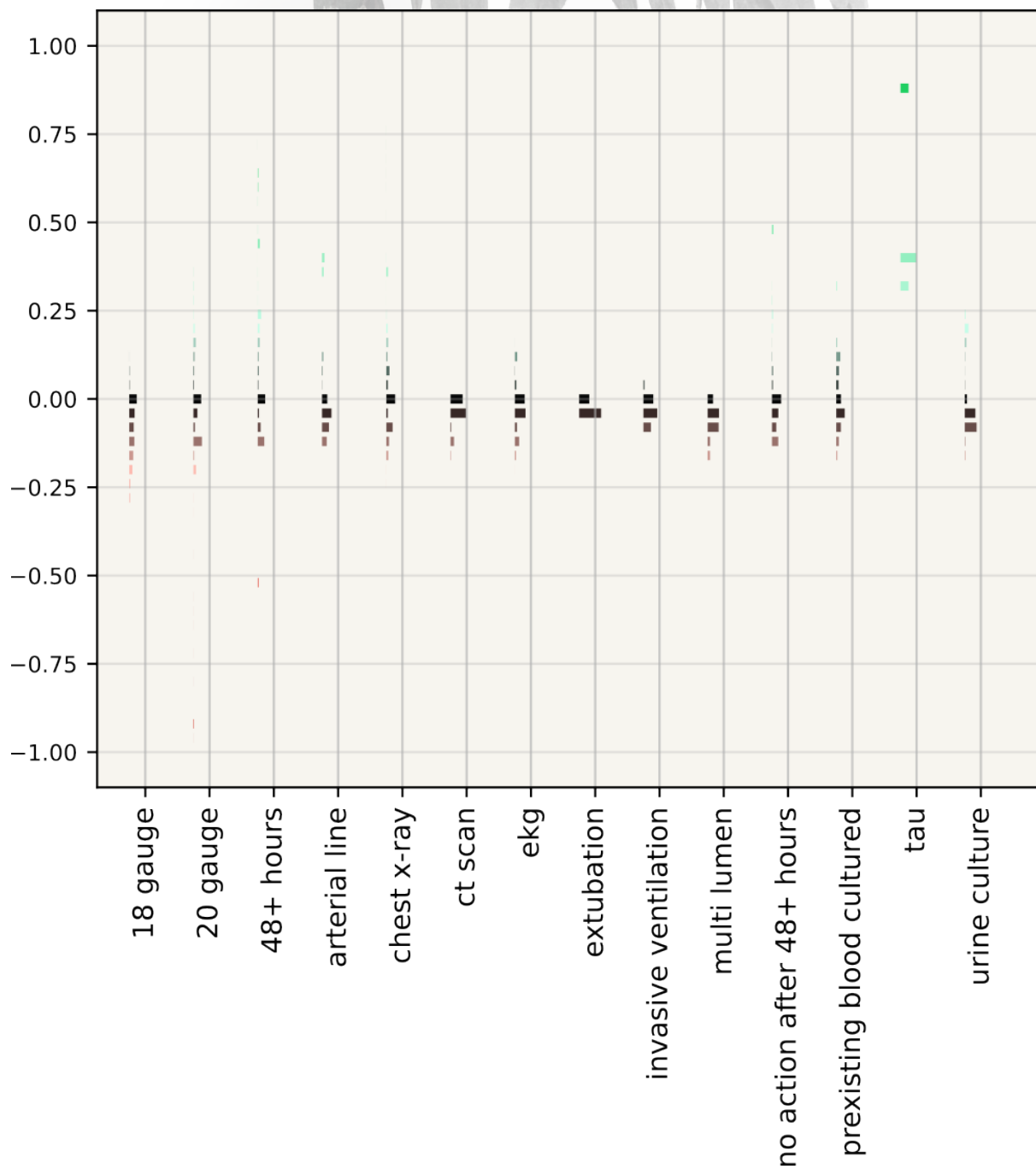


Figure 4.10: Changes in activity likelihoods for a discovered Exo-SLPN for the MIMIC log.

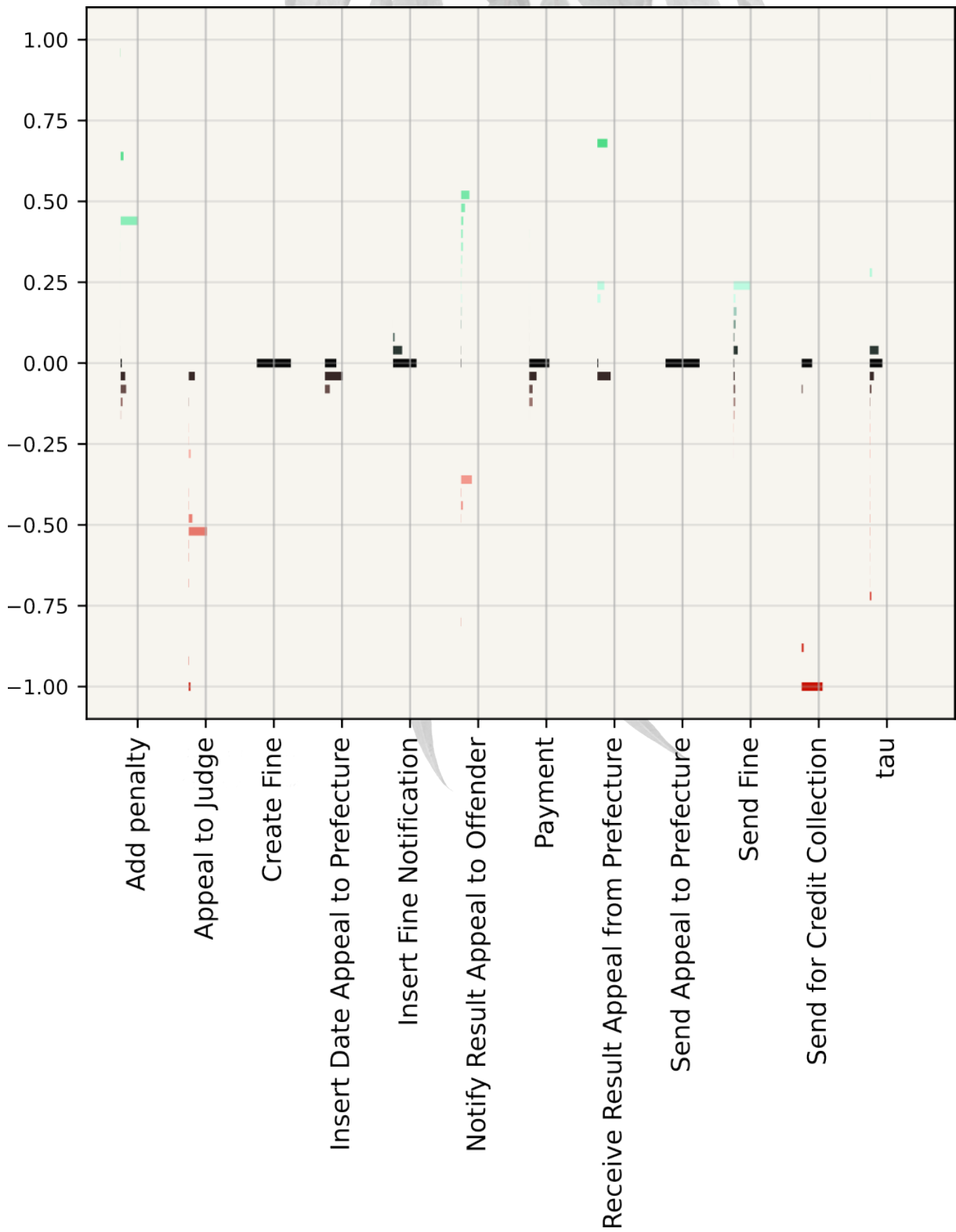


Figure 4.11: Changes in activity likelihoods for a discovered Exo-SLPN for the road fines log.

4.4 Preliminaries

This section describes the notation used for formalisms, and then formally introduce event logs, process models and alignments.

Sets. Curly brackets denote sets, e.g. $\Sigma = \{a, b, c\}$. To denote the number of members in a set Σ , $|\Sigma|$ is written. Given some set, say Σ , a multiset $M: \Sigma \mapsto \mathbb{N}^+$ maps the elements of Σ to the positive natural numbers or zero. Square brackets denote multisets, e.g. $[a^2, b^1]$ s.t. $a, b \in \Sigma$ and for all other elements of Σ , these are mapped to zero and are not denoted. To denote the infinite set of all multisets over a given set Σ , $\mathcal{M}(\Sigma)$ is used. To state relations between a set S and a multiset M , $S \subseteq M$ denotes that S is a subset of M if and only if: $\forall_{s \in S} M(s) > 0$. To denote the cardinality of a multiset M , $\|M\| = \sum_{m \in M} M(m)$ is written.

Sequences. Angle brackets denote sequences, e.g. $\langle a, b, c \rangle$. To concatenate two sequences s, s' , $s: s'$ is written. The length of a sequence s is denoted as $|s|$, and if $|s| = 0$ then s is empty. To access an element of a sequence s , s_i is used to access the i -th element of the sequence s.t. $1 \leq i \leq |s|$. To denote a set of possible sequences over the elements of a set Σ , Σ^* is used to denote all possible non-empty and empty sequences, while Σ^+ is used to denote all possible non-empty sequences.

Events. An event describes the execution of a process step (activity). Each *event* consists of the activity that occurred and a timestamp of the step [2]. For this chapter, the time of event is focused on, accessed through the function `time`. A non-empty sequence of events is a *trace* and describes an execution of a process.

Universes. $\mathcal{U}_{ev}, \mathcal{U}_{time}$ denote the sets of identifies for events, and timestamps. $\mathcal{U}_{att}, \mathcal{U}_{val}$ denote the sets of attributes and all possible values that these can take on. $\mathcal{U}_{ev}, \mathcal{U}_{time}, \mathcal{U}_{att}, \mathcal{U}_{val}$ are mutually exclusive.

Time Series. This chapter focuses on exogenous data which is formed as a numerical discrete univariate time series, referred to as an *exo-series*. Such a time series, is a sequence consisting of measurement and timestamps, say $x \in (\mathcal{U}_{val} \times \mathcal{U}_{time})^*$, focusing on measuring a single concept, or an *exo-panel*, such as temperature, blood pressure or the price of gold. To access the value of the measurement at the ordered element i , x_i is written. Likewise, to access the time of the measurement at i , `time( $x_i$ )` is written. $\mathcal{U}_{es} = (\mathcal{U}_{val} \times \mathcal{U}_{time})^*$ denotes the set of all possible *exo-series*.

Example 1. Consider the following *exo-series* in Figure 4.12 for the number of unpaid fines (`#fines`) and the total amount of unpaid fines (`$fines`) as an inter-case variables derived from the road fines log (discussed later in Section 4.8.1). Each measurement within these *exo-series* describes the number/amount of unpaid fines in the system as of that day and will be used as exogenous data in the following examples.

Exogenous Annotated Logs. The source of time series data will come from applying the xPM framework (see Section 3.4) for this chapter. Where an event log is annotated with exogenous data through several determinations, creating an *exogenous annotated log* (xlog). This chapter solely focuses on linked exogenous data, that is it does not consider user selected

$$\langle \dots, 31188^{14.02.2009}, 31222^{15.02.2009}, 31228^{16.02.2009}, \dots \rangle$$

(a) Number of unpaid fines (#fines).

$$\langle \dots, 2.007e6^{14.02.2009}, 2.008e6^{15.02.2009}, 2.009e6^{16.02.2009}, \dots \rangle$$

(b) Total amount of unpaid fines (\$fines).

Figure 4.12: Exo-series describing inter-case variables for the road fines log.

slices (Section 3.4.3) or transformed attributes (Section 3.4.4) in this contribution. Thus, an xlog is considered where traces have been annotated with several exo-series and the chapter ignores the further affects of determinations.

Definition 2 (Exogenous Annotated Logs) An exogenous-annotated log $L = (\rho, \Sigma)$ is a tuple, where ρ is the set of exo-panel names and $\Sigma = \mathcal{M}(\mathcal{U}_{ev}^* \times (\rho \mapsto \mathcal{U}_{es}))$ is a multiset of traces annotated with a mapping of exo-series.

To consider exo-series x or a mapping of exo-series X , relative to a given timestamp r , referred to as a *truncated* exo-series, the following functions are used:

$$\begin{aligned} \text{trun}(X, r) &= \bigcup_{X(p)=x} \{p \mapsto \text{trun}(x, r)\}, \\ \text{and } \text{trun}(\langle (v, t) \rangle : x, r) &= \begin{cases} \langle (v, r - \text{time}(t)) \rangle : \text{trun}(x, r) & \text{if } \text{time}(t) \leq r, \\ \langle \rangle & \text{otherwise.} \end{cases} \end{aligned} \quad (4.1)$$

It is called *truncated* as the original exo-series has been “cut short” before the given event timestamp.

Example 2. Consider the following visualisation in Figure 4.13 of trace ‘A21188’ from Table 4.1 annotated with exo-series for ‘#fines’ and ‘\$fines’ from Example 1. To demonstrate *trun*, consider applying *trun* in the context of e_{14} (highlighted vertical in Figure 4.13), i.e. $r = \text{time}(e_{14})$, which yields:

$$\begin{aligned} \text{trun}(\{\#fines \mapsto \langle \dots \rangle, \$fines \mapsto \langle \dots \rangle\}, r) &= \{\#fines \mapsto \text{trun}(\langle \dots \rangle, r), \$fines \mapsto \text{trun}(\langle \dots \rangle, r)\} \\ &= \{\#fines \mapsto \langle \dots, 31188^2, 31222^1, 31228^0 \rangle, \\ &\quad \$fines \mapsto \langle \dots, 2.007e6^2, 2.008e6^1, 2.009e6^0 \rangle\} \end{aligned}$$

For each (event) vertical in Figure 4.13, truncated exo-series can be extracted to understand how firing likelihoods might change based on this information. Also by using this approach all times between the events can be considered by re-truncating the exo-series for any arbitrary time.

Petri Nets. Petri nets are used as the modelling notation to describe modelled processes. Two such extensions, labelled Petri nets (LPN) and stochastic Petri nets (introduced later), are considered.

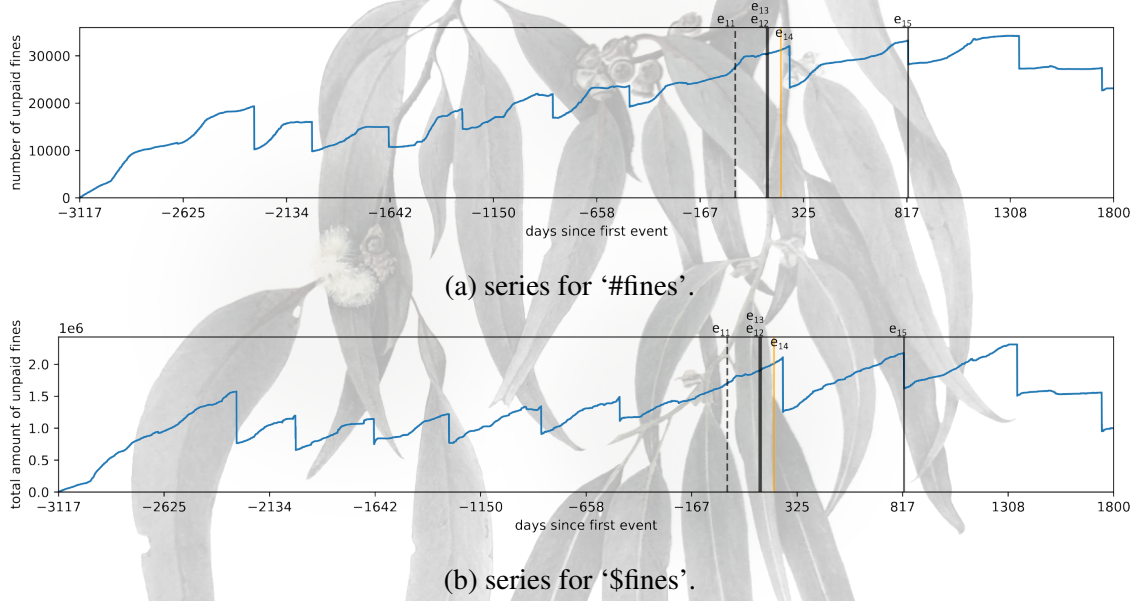


Figure 4.13: The trace 'A21188' from Table 4.1 annotated with two exo-series. Verticals denote when an event occurred in the trace, a dashed vertical denotes the first event, and e_{14} is highlighted.

Definition 3 (Labelled Petri Nets) A labelled Petri net (LPN) is a tuple $(P, T, F, \Sigma, \lambda, M_o)$, where P is a set of places, T is a set of transitions s.t. $P \cap T = \emptyset$, $F \subseteq ((P \times T) \cup (T \times P))$ is a set of directed flows, Σ is a set of process activities, $\lambda : T \mapsto \Sigma \cup \{\tau\}$ is a labelling function, and $M_o \in \mathcal{M}(P)$ is an initial marking.

The preset of a transition t is denoted as $\bullet t = \{p \mid (p, t) \in F\}$, likewise the postset of a transition is $t^\bullet = \{p \mid (t, p) \in F\}$. A transition is enabled in a marking $M \in \mathcal{M}(P)$ if a marking includes all places with an outgoing arc to the transition, i.e. $\bullet t \subseteq M$. A marking represents what places have at least a token, e.g. $M = [p_1]$ then there is one token at p_1 . A run of an LPN starts from the initial marking M_o and fires transition until no transitions are enabled. The firing of a transition consumes tokens from its preset and generates fresh tokens in its postset places. The set of enabled transitions from a marking M is denoted as $\text{enbl}(M) = \{t \in T \mid \bullet t \subseteq M\}$.

Alignments. An alignment (see Section 2.4.1) γ is a sequence of moves, where a move is a combination of an event from the log (or a skip $\gg$) and a transition in the model (or a skip $\gg$). However a move in an alignment cannot consist of two skips $\gg$, i.e. $(\gg)$. A move $(\overset{e}{\underset{t}{\gg}})$ is a synchronous move; a move $(\overset{e}{\underset{t}{\gg}})$ is a log move; and a move $(\overset{e}{\underset{t}{\gg}})$ is a model move. Also, given a move m , the following functions, synm , logm , modm , identify the type of move using the previous cases. Alignments [81] allow for traces to be mapped to a run of an LPN.

Sojourn Times. The contributions of this chapter need to consider when a model move could have occurred. To do so, the sojourn times, the waiting between two fired transitions, are used to inform when model moves could have occurred. In particular, a histogram of the time between synchronous moves in alignments is computed. For instance, say $\gamma = \langle (\overset{e}{\underset{t}{\gg}}), (\overset{e'}{t'}), (\overset{e''}{t''}) \rangle$ is an alignment, then the sojourn time of $\text{soj}(t')$ is $\text{time}(e') - \text{time}(e)$. That is to say that the

sojourn time is the duration from the firing of t to the firing of t' , or in the case of events the duration between e occurring until e' occurs. Formally the function *hist* computes all sojourn times for a given transition, as:

Definition 4 (Sojourn times) *Let LPN be a net, let L be a log, let Γ be the multiset of alignments between LPN and L and let $t \in T$ be a transition. Then, *hist* is a function that takes a transition t to yield a multi-set of sojourn times:*

$$\text{hist}(t) = \bigcup_{\substack{\gamma \in \Gamma, \\ 1 \leq i \leq \Gamma(\gamma)}} [\text{time}(e) - \text{time}(e') \mid \forall_{\gamma': \langle (e'_i) \rangle: \gamma'' : \langle (e'_i) \rangle: \gamma''' = \gamma} \#_{1 \leq j \leq |\gamma''|} \text{synm}(\gamma''_j)]$$

4.5 Stochastic Petri Nets with Exogenous Dependencies

This section introduces stochastic labelled Petri nets with exogenous dependencies or Exo-SLPN, where a weight function that considers exogenous data is introduced. The following modelling formalisation is based on other existing works on stochastic labelled Petri nets [76], [86], [184].

Definition 5 (Stochastic Labelled Petri Nets With Exogenous Dependencies) *An Exo-SLPN is a tuple $(P, T, F, \Sigma, \lambda, M_o, \rho, \text{trans}, \omega)$, where $(P, T, F, \Sigma, \lambda, M_o)$ is an LPN, ρ is a set of exo-panels, $\text{trans}: (\rho \mapsto \mathcal{U}_{es}) \mapsto (\rho \mapsto (\mathbb{R} \cup \perp))$ transforms a mapping of exo-panels and exo-series into a mapping of exo-panels and numbers or $\perp$, and $w: (T \times (\rho \mapsto (\mathbb{R} \cup \perp))) \mapsto \mathbb{R}$ is a weight function.*

The control-flow semantics of an Exo-SLPN are those of the corresponding LPN. Additionally, an Exo-SLPN expresses the likelihood of firing an enabled transition as a ratio of the sum of the weights of all enabled transitions. Formally, given a marking M and a mapping of exo-series E , the probability of firing an enabled transition $t \in \text{enbl}(M)$ is:

$$\text{prob}(M, E, t) = \frac{w(t, \text{trans}(E))}{\sum_{t' \in \text{enbl}(M)} w(t', \text{trans}(E))}. \quad (4.2)$$

While Exo-SLPNs support any representation for a weight function w , this chapters consider forms that use three weights for each transition t . That is, a *base* weight ϕ_t and, for each exo-panel p , two influencing parameters, being an *adjustment* parameter $\varphi_{t,p}$ when exogenous data is available and an *alternate* parameter $\psi_{t,p}$ when exogenous data is unavailable. These parameters quantify how the base weight ϕ_t of a transition t is influenced by exogenous data through $\varphi_{t,p}$ and $\psi_{t,p}$. Given a transition t and a mapping of transforms E , this chapter considers several

forms for a weight function w :

$$w(t, E) = \phi_t \cdot \prod_{E(p)=e} \begin{cases} (\varphi_{t,p})^e & \text{if } e \neq \perp \\ \psi_{t,p} & \text{otherwise} \end{cases} \quad \text{with } \phi_t, \varphi_{t,p}, \psi_{t,p} > 0 \quad (4.3)$$

$$w(t, E) = \phi_t + \sum_{E(p)=e} \begin{cases} e \cdot \varphi_{t,p} & \text{if } e \neq \perp \\ \psi_{t,p} & \text{otherwise} \end{cases} \quad \text{with } \phi_t, \varphi_{t,p}, \psi_{t,p} > 0 \quad (4.4)$$

$$w(t, E) = \phi_t + \sum_{E(p)=e} \begin{cases} e \cdot \varphi_{,p} & \text{if } e \neq \perp \\ \psi_{,p} & \text{otherwise} \end{cases} \quad \text{with } \phi_t, \varphi_{,p}, \psi_{,p} > 0 \quad (4.5)$$

Equation 4.3 expresses exogenous influences could be multiplicative with the base weight of a transition, where they can have a dramatic impact on the likelihood of firing. Equation 4.4 and Equation 4.5 express exogenous influences could be additive with the base weight of a transition, where they can only have a positive impact on the likelihood of firing. However, Equation 4.5 only introduces one set of parameters per influence (i.e. exo-panel $p : \varphi_{,p}, \psi_{,p}$), rather than individually $(\varphi_{t,p}, \psi_{t,p})$ on transitions. These options are explored to understand which one may have the best generalisation.

Similarly, Exo-SLPNs support any transformation function trans , and the methods in the remainder of this paper do not pose any restrictions on these functions. But for this paper, a default transformation for numerical time series is used. This transformation returns a weighted average of z-scores (Equation 4.7), where measurements further away from the truncation have a smaller influence on the transformed value. The intuition is that changes that happened furthest away from the decision point will have less impact on the firing likelihood. Z-scores are used to capture the variability and volatility in exogenous data, quantifying exogenous data for firing likelihoods. However, when there is no variance within a panel, the transformation maps to zero to identify that zero change can be observed from this panel. Similarly, when the truncation results in an empty sequence, the transformation maps to $\perp$ to use of the alternate parameter.

Definition 6 (Default Numerical Transformation) *Let X be a mapping of panels to truncated exo-series, let $p \in \rho$ be an exo-panel, let $\mu(p)$ be the mean of p , let $\sigma(p)$ be the standard deviation of p , and let $X(p) = x$ be a truncated exo-series for p . Then, trans' maps a collection of truncated exo-series into numbers:*

$$\text{trans}'(X)(p) = \begin{cases} \perp & \text{if } X(p) = \langle \rangle \\ 0 & \text{if } \sigma(p) = 0 \wedge X(p) \neq \langle \rangle \\ z & \text{if } \sigma(p) \neq 0 \wedge X(p) = \langle x_1, \dots, x_n \rangle \end{cases} \quad (4.6)$$

$$\text{where } z = \frac{\sum_{i=1}^n \frac{1}{1+\text{time}(x_i)} \cdot \left| \frac{x_i - \mu(p)}{\sigma(p)} \right|}{\sum_{i=1}^n \frac{1}{1+\text{time}(x_i)}} \quad (4.7)$$

Each transformed-truncated-exo-series is referred to as a TTES.

Example 3. Consider the Exo-SLPN shown in Figure 4.15 and trace ‘A21188’ annotated with exogenous data from Example 2. The goal is to understand the change in probability for credit collection based on the exogenous data in the duration between e_{13} and e_{14} . In order to perform transformation, both the mean (mean) and standard deviation (std) of the panels are needed. As such, for #fines these are 20928.975 and 8084.747, for \$fines these are 1308182.069 and 479526.257. Therefore, applying trans on the truncated exo-series presented in Example 2, yields a mapping E between panels and TTESes in the following manner:

$$\begin{aligned} E &= \text{trans}'(\{\#fines \mapsto \langle \dots, 31228^0 \rangle, \$fines \mapsto \langle \dots, 2.009e6^0 \rangle\}) \\ &= \{\#fines \mapsto 0.864, \$fines \mapsto 0.919\}. \end{aligned}$$

Note that a shorthand is used in the following calculation for panels, ‘#F’ for ‘#fines’ and ‘\$F’ for ‘\$fines’. Then, the probability of credit collection (CC) firing when the trace ‘A21188’ executed ‘add penalty’ (AP) was:

$$\begin{aligned} \text{prob}(M, E, \text{CC}) &= \frac{w(\text{CC}, E)}{\sum_{t \in \text{enbl}(M)} w(t, E)} \\ &\approx \frac{3.4321 \cdot \begin{cases} 0.6786^{E(\#F)} & \text{if } E(\#F) \neq \perp \\ 0.0005 & \text{if } E(\#F) = \perp \end{cases} \cdot \begin{cases} 0.8875^{E(\$F)} & \text{if } E(\$F) \neq \perp \\ 0.0005 & \text{if } E(\$F) = \perp \end{cases}}{\dots + 0.7401 \cdot \begin{cases} 0.0882^{E(\#F)} & \text{if } E(\#F) \neq \perp \\ 59.188 & \text{if } E(\#F) = \perp \end{cases} \cdot \begin{cases} 2.4169^{E(\$F)} & \text{if } E(\$F) \neq \perp \\ 59.188 & \text{if } E(\$F) = \perp \end{cases} + \dots} \\ &\approx \frac{3.4321 \cdot 0.6786^{0.864} \cdot 0.8875^{0.919}}{3.4321 \cdot 0.6786^{0.864} \cdot 0.8875^{0.919} + 0.7401 \cdot 0.0882^{0.864} \cdot 2.4169^{0.919} + \dots} \\ &\approx 0.4635 \text{ or } 46.35\%. \end{aligned}$$

Then, using equally distance time points between events, the computation of the probability for credit collection between the events is shown in Figure 4.14. This figure shows that credit collection was likely ($\sim 47\%$) to fire in comparison to the other activities.

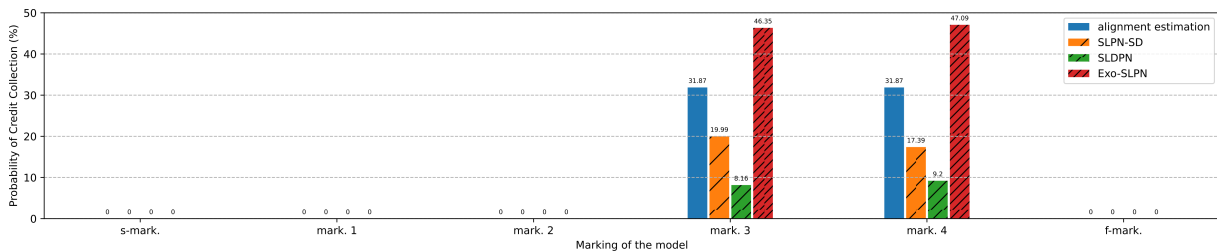


Figure 4.14: The probability of credit collection firing between events according to the Exo-SLPN in Figure 4.15.

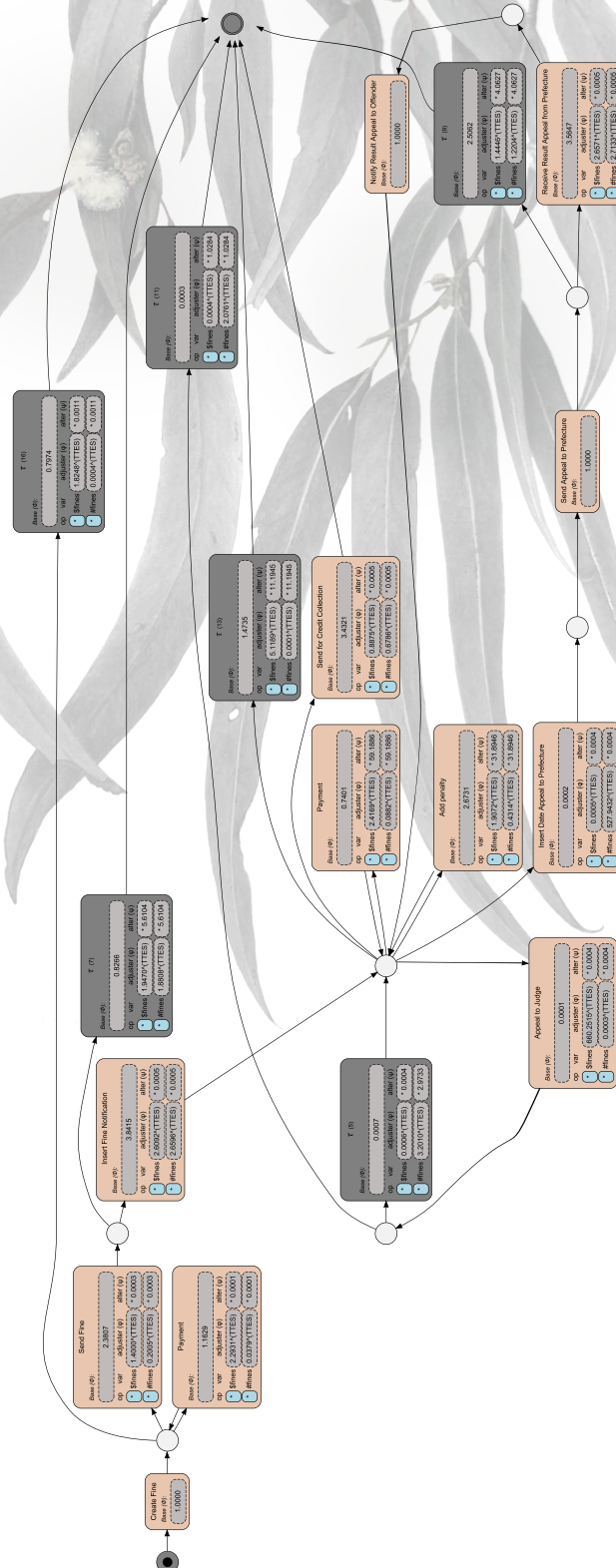


Figure 4.15: Exo-SLPN discovered from the road fines log using the normative model. Weights (φ, ψ) not set to one and associated with a transition are denoted on the transition.

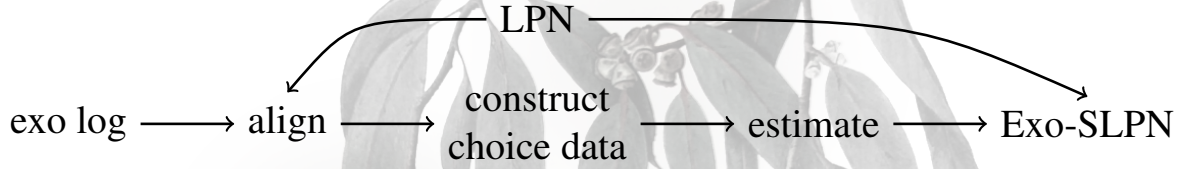


Figure 4.16: Discovering an Exo-SLPN.

4.6 Discovery

This section outlines the approach used to discover an Exo-SLPN, which uses an event log and an LPN and casts the discovery as an optimisation problem over equalities. The approach is illustrated in Figure 4.16, and starts with an LPN and an exogenous log to estimate the parameters for each transition. First, the process of reconstructing the choices that were made in the model is described, referred to as *choice data*. Then, the estimation of the weights is cast as an optimisation problem. Next, the optimisation process for the equalities to obtain values for the parameters of each transition is introduced, i.e. a base weight ϕ , many adjustment φ parameters and many alternative ψ parameters. Lastly, a visualisation tool for the changes in firing likelihoods is proposed.

4.6.1 Constructing Choice Data

After alignments have been obtained for traces in a log, the next step is to gather the model choices, as well as the TTES values for these model choices. That is, for each transition fired in the model to replay the log, the following is recorded: (i) the transitions that were enabled, (ii) the TTES values as they were at the time the transition fired, and (iii) the transition that fired. For synchronous moves, obtaining the TTES is easy, as moves have events with a timestamp, which straightforwardly yields TTES values (Equation 4.9). For a log move, no transition is recorded, hence no model decision was made, so these moves do not introduce any choice data (Equation 4.10). For model moves, no timestamp is available and Section 4.6.2 discusses how TTES values are obtained for these moves using the function *fill* (Equation 4.11).

Formally, for a trace, choice data is obtained through the recursion denoted in Definition 7. The choice data for an event log is then simply the multiset union of the choice data for its traces.

Definition 7 (Choice data) *Let $(\sigma, \#) \in \Sigma$ be a trace σ annotated with *exo-series* $\#$ in an xlog, let LPN be a net, and let γ be an alignment between σ and LPN . Then, *choice* takes a marking*

and an alignment returns a multiset of choice data:

$$\text{choice}(M, \langle \rangle) = [], \quad (4.8)$$

$$\text{choice}(M, \langle (e_t) \rangle : \gamma') = [(\text{enbl}(M), \text{trans}'(\text{trun}(\#, \text{time}(e))), t)] \quad (4.9)$$

$$\cup \text{choice}(M \setminus \bullet t \cup t^\bullet, \gamma'),$$

$$\text{choice}(M, \langle (e_{\geq}) \rangle : \gamma') = \text{choice}(M, \gamma'), \quad (4.10)$$

$$\text{choice}(M, \langle (e_t) \rangle : \gamma') = [(\text{enbl}(M), \text{fill}(\#, \gamma, |\gamma| - |\gamma'|), t)] \quad (4.11)$$

$$\cup \text{choice}(M \setminus \bullet t \cup t^\bullet, \gamma').$$

Example 4. Consider the snippet of the road fines log (Table 4.1) and the alignments in Figure 4.4, where the abbreviations are used for activities (see Figure 4.15). Then, the recursion of choice is explored for trace ‘A21188’ to demonstrate how choice data is derived from the diagnostic information stored in the alignment.

$$\begin{aligned} \text{choice}(M_o, (e_{11}^{\text{CF}}) : \langle \dots \rangle) &= \\ & [(\{ \text{CF} \}, \{ \# \text{fines} \mapsto 0.551, \$ \text{fines} \mapsto 0.529 \}, \text{CF})] \cup \text{choice}(M', (e_{12}^{\text{SF}}) : \langle \dots \rangle) \\ &= [(\{ \text{CF} \}, \{ \dots \}, \text{CF})] \\ & \quad \cup [(\{ \text{SF}, \tau_{16} \}, \{ \# \text{fines} \mapsto 0.797, \$ \text{fines} \mapsto 0.789 \}, \text{SF})] \\ & \quad \cup \text{choice}(M', (e_{13}^{\text{IFN}}) : \langle \dots \rangle) \\ &= [(\{ \text{CF} \}, \{ \dots \}, \text{CF}), (\{ \text{SF}, \tau_{16} \}, \{ \dots \}, \text{SF})] \\ & \quad \cup [(\{ \text{IFN}, \tau_7 \}, \{ \# \text{fines} \mapsto 0.802, \$ \text{fines} \mapsto 0.803 \}, \text{IFN})] \\ & \quad \cup \text{choice}(M', (e_{14}^{\text{AP}}) : \langle \dots \rangle) \\ &= [(\{ \text{CF} \}, \{ \dots \}, \text{CF}), (\{ \text{SF}, \tau_{16} \}, \{ \dots \}, \text{SF}), (\{ \text{IFN}, \tau_7 \}, \{ \dots \}, \text{IFN})] \\ & \quad \cup [(\{ \text{CC}, \text{AP}, \text{PYb}, \text{AtJ}, \tau_{13} \}, \{ \# \text{fines} \mapsto 0.864, \$ \text{fines} \mapsto 0.919 \}, \text{AP})] \\ & \quad \cup \text{choice}(M', (e_{15}^{\text{CC}}) : \langle \rangle) \\ &= [(\{ \text{CF} \}, \{ \dots \}, \text{CF}), (\{ \text{SF}, \tau_{16} \}, \{ \dots \}, \text{SF}), (\{ \text{IFN}, \tau_7 \}, \{ \dots \}, \text{IFN}), \\ & \quad (\{ \text{CC}, \text{AP}, \text{PYb}, \text{AtJ}, \tau_{13} \}, \{ \dots \}, \text{AP})] \\ & \quad \cup [(\{ \text{CC}, \text{AP}, \text{PYb}, \text{AtJ}, \tau_{13} \}, \{ \# \text{fines} \mapsto 1.052, \$ \text{fines} \mapsto 1.174 \}, \text{CC})] \end{aligned}$$

Thus for trace ‘A21188’, five triples of choice data are extracted and are combined with choice data from the other traces in the snippet as shown in Table 4.2. Often the choice data derived is distinctly unique, which can lead to a state explosion as seen in Table 4.2.

4.6.2 Handling Model Moves

To obtain TTES values from model moves, two types of model moves are distinguished: moves on silent transitions, i.e. $\lambda(t) = \tau$, and moves on labelled transitions, i.e. $\lambda(t) \neq \tau$. To define the fill function, first synchronous (which have TTES values) and log moves (which do not have

Table 4.2: Choice data extracted for example 4.

enabled (T)	TTESes (E)	fired (t)
{ CF }	{#fines $\mapsto$ 0.282, \$fines $\mapsto$ 0.357}	[CF]
{ CF }	{#fines $\mapsto$ 0.530, \$fines $\mapsto$ 0.552}	[CF]
{ CF }	{#fines $\mapsto$ 0.633, \$fines $\mapsto$ 0.598}	[CF]
{ IFN , τ_7 }	{#fines $\mapsto$ 0.424, \$fines $\mapsto$ 0.363}	[IFN]
{ IFN , τ_7 }	{#fines $\mapsto$ 0.803, \$fines $\mapsto$ 0.803}	[IFN]
{ PYa , SF , τ_{16} }	{#fines $\mapsto$ 0.299, \$fines $\mapsto$ 0.360}	[PYa]
{ PYa , SF , τ_{16} }	{#fines $\mapsto \perp$, \$fines $\mapsto \perp$ }	[τ_{16}]
{ PYa , SF , τ_{16} }	{#fines $\mapsto$ 0.439, \$fines $\mapsto$ 0.380}	[SF]
{ PYa , SF , τ_{16} }	{#fines $\mapsto$ 0.789, \$fines $\mapsto$ 0.798}	[SF]
{ CC , PYb , IAP , τ_{13} , AP , AtJ }	{#fines $\mapsto$ 0.310, \$fines $\mapsto$ 0.313}	[AP , IAP]
{ CC , PYb , IAP , τ_{13} , AP , AtJ }	{#fines $\mapsto$ 0.919, \$fines $\mapsto$ 0.865}	[AP]
{ CC , PYb , IAP , τ_{13} , AP , AtJ }	{#fines $\mapsto$ 1.174, \$fines $\mapsto$ 1.052}	[CC]
{ SAP }	{#fines $\mapsto$ 0.275, \$fines $\mapsto$ 0.310}	[SAP]
{ RAP , τ_9 }	{#fines $\mapsto$ 0.256, \$fines $\mapsto$ 0.289}	[RAP]
{ NRA }	{#fines $\mapsto$ 0.505, \$fines $\mapsto$ 0.427}	[NRA]

TTES values, as they do not represent transitions) are considered:

$$\text{fill}(\#, \gamma, i) = \begin{cases} \perp & \text{if } i > |\gamma| & (4.12a) \\ \text{trans}'(\text{trun}(\#, \text{time}(e))) & \text{if } \text{synm}(\gamma_i) \text{ with } \gamma_i = (\frac{e}{i}) & (4.12b) \\ \text{fill}(\#, \gamma, i + 1) & \text{if } \text{logm}(\gamma_i) & (4.12c) \\ \dots & \end{cases}$$

Silent Model Moves

A silent transition models a change in the state of a system without directly visible external consequences. As such, they have neither timestamps nor TTES. Nevertheless, the decision to fire the silent transition and not a potential competing visible transition depends on the TTES value, and as such must be assigned a TTES. As the firing of a silent transition has no corresponding action in the process (i.e. just in the system), this chapter posits that the moment they happen is the moment the next labelled transition fires:

$$\text{fill}(\#, \gamma, i) = \begin{cases} \dots \\ \text{fill}(\#, \gamma, i + 1) & \text{if } \text{modm}(\gamma_i) \wedge \lambda(t) = \tau \text{ with } \gamma_i = (\frac{\tau}{i}) \\ \dots \end{cases} \quad (4.12d)$$

Labelled Model Moves

A labelled transition models the execution of a process step; a labelled model move indicates that this step was executed, but not logged, and hence there is no timestamp available to compute

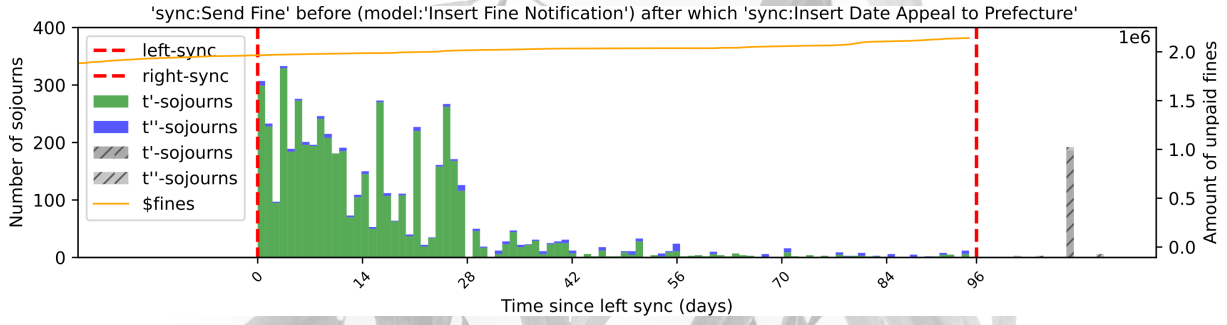


Figure 4.17: Illustration of where a model move in between two synchronous moves may have occurred (in green).

the TTES values. Thus, an TTES is obtained through estimation based on other moves with timestamps around the model move.

For the sake of explanation, assume an alignment with two synchronous moves and a model move in between: $\langle \dots (e_t), (\overset{e}{t}), (e'') \dots \rangle$. TTES values depends on when the transition t' was fired, and in this case it was between e and e'' . Time wise, this means that the time of e plus the sojourn times of t' and t'' is the time of e'' ($\text{time}(e) + \text{soj}(t') + \text{soj}(t'') = \text{time}(e'')$). Neither $\text{soj}(t')$ or $\text{soj}(t'')$ are known, though a distribution from all sojourn times of t' and t'' in the log can be observed. Figure 4.17 illustrates this partially: from the moment that (e_t) happens (vertical line on the left), the distribution of sojourn times of t' (green bars) can be projected. Furthermore, note that $\text{soj}(t')$ cannot have lasted beyond (e'') and that some sojourns times are not appropriate (grayed out bars).

Observe that at each potential timestamp in the green region, the TTES value would be computed using the exo-measures up till the last observed exo-measure. Thus, to estimate the TTES value, the number points in the distribution of $(\overset{e}{t})$ (the green region) that occurred before to each exo-measure is used to compute a weighted average accordingly:

$$\text{fill}(\#, \gamma, i) = \begin{cases} \dots \\ \{p \mapsto \begin{cases} \frac{1}{\|S\|} \sum_{s \in S} E(p)S(s) & \text{if } \|S\| > 0 \\ \perp & \text{otherwise} \end{cases} \mid p \in \# \} \\ \text{if } \text{modm}(\gamma_i) \wedge \lambda(t) \neq \tau \end{cases} \quad (4.12e)$$

with $(e_t), (e'')$ being the most recent, next sync moves in γ

and sojourns $S = [s^{\text{hist}(t')(s)} \mid s \in \text{hist}(t') \wedge s \leq \text{time}(e'') - \text{time}(e) \wedge E(p) \neq \perp]$

and transforms of $E = \text{trans}'(\text{trun}(\#, \text{time}(e) + s))$

The edge cases in which no synchronous move precedes or succeeds a model move are handled similarly. Please note two factors are explicitly left out: (i) the distribution over sojourn times of t'' is not leveraged, and (ii) that there may be multiple model moves in between two synchronous moves. These factors may be included in future work.

4.6.3 Solving Equalities

This section outlines how to estimate base weights ϕ , adjustments φ , and alternatives ψ parameters to discover an Exo-SLPN, by casting the problem of estimating these parameters as an linear optimisation problem. To do so, equalities are constructed from the obtained choice data. For each observed choice, the observations in the log are equated with the probability of that choice in the model (expressed in the weight parameters). A standard solver is then applied to minimise the disagreement between the sides of these equations. Two solution strategies are considered, one where all the equations are solved in one-shot, and a two-shot approach where base variables are minimised first, then the adjustment and alternative parameters are minimised, both using the following equations.

Recall that the choice data C consists of triples of enabled transitions, the observed TTES values and which transitions fired. For each pair of enabled transitions (T) and TTES values (E) in Equation 4.13, an equated fraction is introduced for every observed transition $o \in T$ through Equation 4.14. The equated fraction in Equation 4.15 states that the relative occurrence of the transition o should match the probability in the model, given the weight parameters denoted in Equation 4.16:

$$\forall_{(T,E) \in \{(T,E) | (T,E,t) \in C\}} \quad (4.13)$$

$$\forall_{o \in \{o | (T,E,o) \in C\}} \quad (4.14)$$

$$\frac{||[(T, E, o) \in C]||}{||[(T, E, t') \in C]||} \quad (4.15)$$

$$= \frac{\phi_o \prod_{E(p)=e} \begin{cases} \varphi_{o,p}^e & \text{if } e \neq \perp \\ \psi_{o,p} & \text{otherwise} \end{cases}}{\sum_{t' \in T} \phi_{t'} \prod_{E(p)=e} \begin{cases} \varphi_{t',p}^e & \text{if } e \neq \perp \\ \psi_{t',p} & \text{otherwise} \end{cases}} \quad (4.16)$$

If all enabled transitions have been observed at least once when constructing equalities, then one of the equations (an observed transition o from Equation 4.14) for every (T, E) (from Equation 4.13) can be removed without loss of information.

Example 5. For instance, the equalities corresponding to the running example of Table 4.2 are shown in Table 4.3. The equalities that are not necessary for the solver are given in red. By passing these equalities to a solver, the parameters ϕ, φ, ψ for each transition are assigned a value which minimally reduces the difference between these equalities. The examples thus far have shown the necessary steps needed to discover an Exo-SLPN.

Table 4.3: The constructed equalities for the running example. Panel names have been abbreviated, #fines is #F and \$fines is \$F. Red equalities are not necessary for the solver.

enabled (T)	TTES (E)	fired (t)	equality
{CF}	{...}	[CF]	$\frac{1}{1} = \frac{\phi_{CF} \cdot \psi_{CF, \#F}^{0.282} \cdot \psi_{CF, \$F}^{0.357}}{\phi_{CF} \cdot \psi_{CF, \#F}^{0.282} \cdot \psi_{CF, \$F}^{0.357}}$
{CF}	{...}	[CF]	$\frac{1}{1} = \frac{\phi_{CF} \cdot \psi_{CF, \#F}^{0.530} \cdot \psi_{CF, \$F}^{0.552}}{\phi_{CF} \cdot \psi_{CF, \#F}^{0.530} \cdot \psi_{CF, \$F}^{0.552}}$
{CF}	{...}	[CF]	$\frac{1}{1} = \frac{\phi_{CF} \cdot \psi_{CF, \#F}^{0.633} \cdot \psi_{CF, \$F}^{0.598}}{\phi_{CF} \cdot \psi_{CF, \#F}^{0.633} \cdot \psi_{CF, \$F}^{0.598}}$
{IFN, τ_7 }	{...}	[IFN]	$\frac{1}{1} = \frac{\phi_{IFN} \cdot \psi_{IFN, \#F}^{0.424} \cdot \psi_{IFN, \$F}^{0.363}}{\phi_{IFN} \cdot \psi_{IFN, \#F}^{0.424} \cdot \psi_{IFN, \$F}^{0.363} + \phi_{\tau_7} \cdot \psi_{\tau_7, \#F}^{0.424} \cdot \psi_{\tau_7, \$F}^{0.363}}$
{IFN, τ_7 }	{...}	[IFN]	$\frac{1}{1} = \frac{\phi_{IFN} \cdot \psi_{IFN, \#F}^{0.803} \cdot \psi_{IFN, \$F}^{0.803}}{\phi_{IFN} \cdot \psi_{IFN, \#F}^{0.803} \cdot \psi_{IFN, \$F}^{0.803} + \phi_{\tau_7} \cdot \psi_{\tau_7, \#F}^{0.803} \cdot \psi_{\tau_7, \$F}^{0.803}}$
{PYa, SF, τ_{16} }	{...}	[PYa]	$\frac{1}{1} = \frac{\phi_{PYa} \cdot \psi_{PYa, \#F}^{0.803} \cdot \psi_{PYa, \$F}^{0.803}}{\phi_{PYa} \cdot \psi_{PYa, \#F}^{0.803} \cdot \psi_{PYa, \$F}^{0.803} + \dots}$
{PYa, SF, τ_{16} }	{...}	[τ_{16}]	$\frac{1}{1} = \frac{\phi_{\tau_{16}} \cdot \psi_{\tau_{16}, \#F}^{0.439} \cdot \psi_{\tau_{16}, \$F}^{0.380}}{\phi_{\tau_{16}} \cdot \psi_{\tau_{16}, \#F}^{0.439} \cdot \psi_{\tau_{16}, \$F}^{0.380} + \dots}$
{PYa, SF, τ_{16} }	{...}	[SF]	$\frac{1}{1} = \frac{\phi_{SF} \cdot \psi_{SF, \#F}^{0.439} \cdot \psi_{SF, \$F}^{0.380}}{\phi_{SF} \cdot \psi_{SF, \#F}^{0.439} \cdot \psi_{SF, \$F}^{0.380} + \dots}$
{PYa, SF, τ_{16} }	{...}	[SF]	$\frac{1}{1} = \frac{\phi_{SF} \cdot \psi_{SF, \#F}^{0.789} \cdot \psi_{SF, \$F}^{0.798}}{\phi_{SF} \cdot \psi_{SF, \#F}^{0.789} \cdot \psi_{SF, \$F}^{0.798} + \dots}$
{CC, PYb, IAP, τ_{13} , AP, AtJ}	{...}	[AP, IAP]	$\frac{1}{2} = \frac{\phi_{AP} \cdot \psi_{AP, \#F}^{0.310} \cdot \psi_{AP, \$F}^{0.313}}{\phi_{AP} \cdot \psi_{AP, \#F}^{0.310} \cdot \psi_{AP, \$F}^{0.313} + \dots}$
{CC, PYb, IAP, τ_{13} , AP, AtJ}	{...}	[AP]	$\frac{1}{1} = \frac{\phi_{AP} \cdot \psi_{AP, \#F}^{0.919} \cdot \psi_{AP, \$F}^{0.865}}{\phi_{AP} \cdot \psi_{AP, \#F}^{0.919} \cdot \psi_{AP, \$F}^{0.865} + \dots}$
{CC, PYb, IAP, τ_{13} , AP, AtJ}	{...}	[CC]	$\frac{1}{1} = \frac{\phi_{CC} \cdot \psi_{CC, \#F}^{1.174} \cdot \psi_{CC, \$F}^{1.052}}{\phi_{CC} \cdot \psi_{CC, \#F}^{1.174} \cdot \psi_{CC, \$F}^{1.052} + \dots}$
{SAP}	{...}	[SAP]	$\frac{1}{1} = \frac{\phi_{SAP} \cdot \psi_{SAP, \#F}^{0.275} \cdot \psi_{SAP, \$F}^{0.310}}{\phi_{SAP} \cdot \psi_{SAP, \#F}^{0.275} \cdot \psi_{SAP, \$F}^{0.310}}$
{RAP, τ_9 }	{...}	[RAP]	$\frac{1}{1} = \frac{\phi_{RAP} \cdot \psi_{RAP, \#F}^{0.256} \cdot \psi_{RAP, \$F}^{0.289}}{\phi_{RAP} \cdot \psi_{RAP, \#F}^{0.256} \cdot \psi_{RAP, \$F}^{0.289} + \phi_{\tau_9} \cdot \psi_{\tau_9, \#F}^{0.256} \cdot \psi_{\tau_9, \$F}^{0.289}}$
{NRA}	{...}	[NRA]	$\frac{1}{1} = \frac{\phi_{NRA} \cdot \psi_{NRA, \#F}^{0.505} \cdot \psi_{NRA, \$F}^{0.427}}{\phi_{NRA} \cdot \psi_{NRA, \#F}^{0.505} \cdot \psi_{NRA, \$F}^{0.427}}$

4.6.4 Visualisation

This section presents a visualisation to show the change in firing probabilities for a given log and an Exo-SLPN, highlighting the impact of exogenous data on process behaviour. This section continues with the running example of the road fines event log and using the unpaid fines in the system as exogenous data.

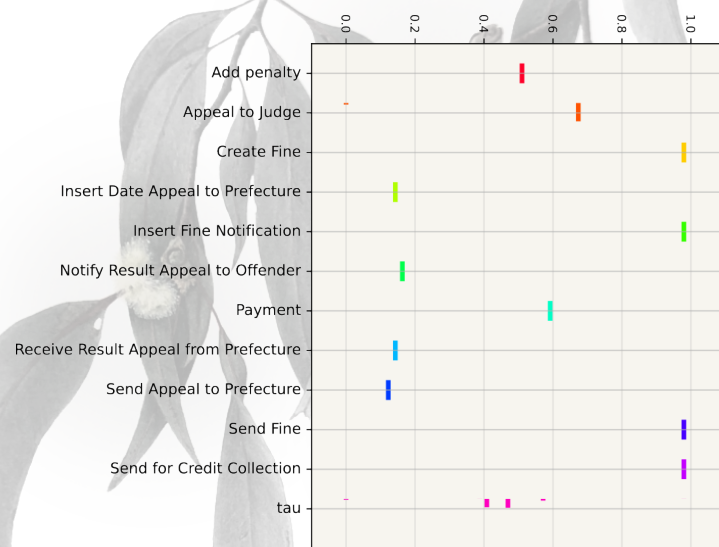
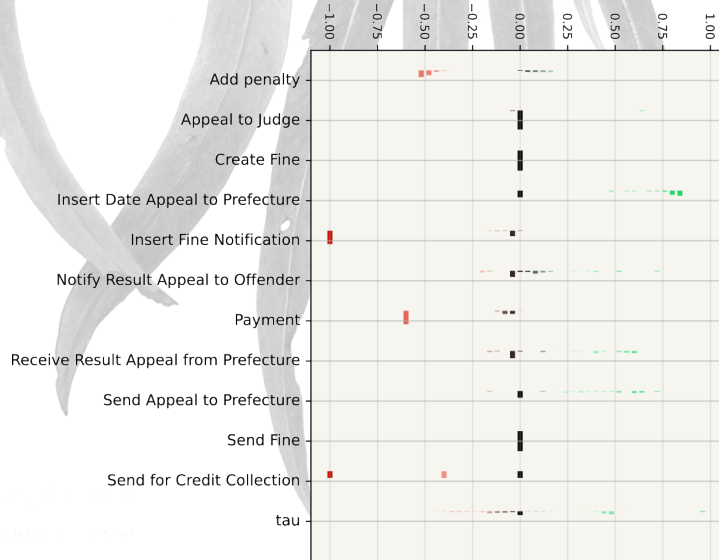
Data. Figure 4.18 visualises the choice data for a discovered Exo-SLPN by using the choice data to plot vertical histograms of firing probabilities for process activities. In Figure 4.18a, the histograms of firing probabilities per activity are computed only considering the base weight. While in Figure 4.18c, the histograms are computed using the proposed weight function in Equation 4.3 with all the discovered parameters for the Exo-SLPN. Finally, Figure 4.18b shows histograms that are computed on the difference between these probabilities, such that the right probability is the base plus the difference. On top of the choice data used for the discovery,

all perturbations of choice where a denoted TTES is removed are included, i.e. set to $\perp$ for an exo-panel.

Presentation. In Figure 4.18a and Figure 4.18c, colour is used to denote the transition fired. While the colour in Figure 4.18b denotes an increase in the probability (greener), a decrease (redder), or little change (black). The width of the bars shows the relative portion of choice data points with that probability. The x-axis groups firing probabilities by transition. Note that all silent transitions are grouped into one ‘tau’ group for simplicity.

Interpretation. Figure 4.18b allows the user to quickly see if any of the introduced exogenous data has changed the likelihood of firing a transition. That is, if Figure 4.18b only contains long black bars, then no change in the firing probabilities based on exogenous data would have been introduced. This is not the case for Figure 4.18, where there are increases in the likelihood of firing for three groups (green bars) and decreases for four groups (red bars). Insight that can be drawn from Figure 4.18 is that the likelihood of ‘payment’ based on exogenous data is unlikely, while the base weight would say otherwise. This chapter focuses on the difference visualisation (Figure 4.18b) when discussing Exo-SLPNs as it captures the change in firing likelihoods based on exogenous data.

Example 6. To consider how the visualisation uses the choice data to provide information back to user, Table 4.3 is revisited and the perturbations generated are highlighted in Table 4.4. Using the three rightmost columns in Table 4.4, histograms are computed for the fired transitions and data for histograms is grouped based on the transition’s label. Note that TTESes are only removed if it was original present, missing $\perp$ TTESes are not imputed. Using these perturbations, an Exo-SLPN can be considered in the context of the xlog used for discovery and focuses the interpretation on how the additional parameters influence firing likelihoods.

(a) prob with ϕ .

(b) likelihood change.

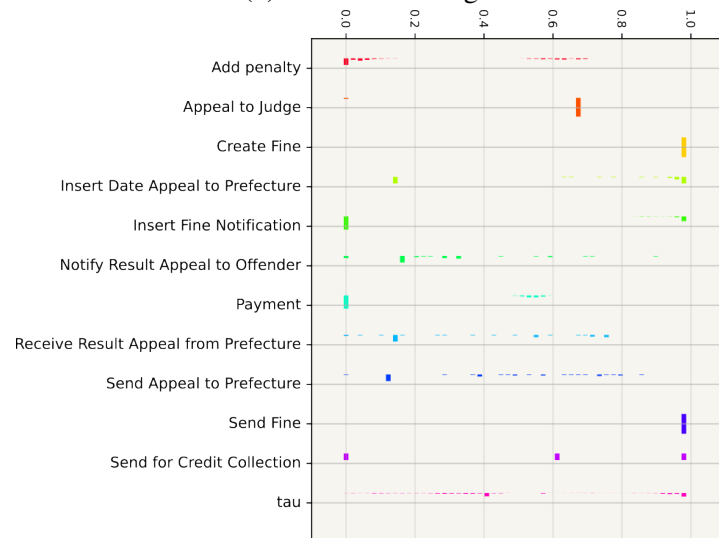
(c) prob with ϕ, φ, ψ .

Figure 4.18: Depicting the changes in the likelihood of firing transitions for the road fines log based on the discovered Exo-SLPN using the IMf model in the evaluation (Section 4.8).

Table 4.4: Perturbations of choice data using the choice data from example 4 and the discovered Exo-SLPN in Figure 4.15 for probabilities. Perturbations are highlighted in red. The three rightmost columns are used in the visualisation tool.

TTEs (E)	fired (t)	prob with ϕ	change	prob with ϕ, φ, ψ
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\{\#fines \mapsto 0.423, \$fines \mapsto 0.362\}$	IFN	0.999	0.000	0.999
$\{\#fines \mapsto 0.423, \$fines \mapsto \perp\}$	IFN	0.999	-0.223	0.777
$\{\#fines \mapsto \perp, \$fines \mapsto 0.362\}$	IFN	0.999	-0.712	0.288
$\{\#fines \mapsto \perp, \$fines \mapsto \perp\}$	IFN	0.999	-1.000	0.000
$\{\#fines \mapsto 0.803, \$fines \mapsto 0.802\}$	IFN	0.999	0.000	0.999
$\{\#fines \mapsto 0.803, \$fines \mapsto \perp\}$	IFN	0.999	-0.222	0.777
$\{\#fines \mapsto \perp, \$fines \mapsto 0.802\}$	IFN	0.999	-0.712	0.288
$\{\#fines \mapsto \perp, \$fines \mapsto \perp\}$	IFN	0.999	-1.000	0.000
$\{\#fines \mapsto 0.299, \$fines \mapsto 0.359\}$	PYa	0.445	-0.029	0.417
$\{\#fines \mapsto \perp, \$fines \mapsto 0.359\}$	PYa	0.445	-0.438	0.008
$\{\#fines \mapsto 0.299, \$fines \mapsto \perp\}$	PYa	0.445	-0.445	0.001
$\{\#fines \mapsto \perp, \$fines \mapsto \perp\}$	PYa	0.445	-0.446	0.000
$\{\#fines \mapsto \perp, \$fines \mapsto \perp\}$	τ_{16}	0.019	0.980	0.999
$\{\#fines \mapsto 0.438, \$fines \mapsto 0.380\}$	SF	0.535	0.031	0.566
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

4.7 Conformance Checking

This section outlines how to perform conformance checking on an Exo-SLPN, where the data-aware unit Earth mover's distance [76] (duEMSC) is adapted for the contributions of the chapter. Data-aware unit Earth movers extends the unit Earth-movers [75] by adding the data perspective as an additional dimension when comparing and calculating probability distributions. The measurement of data-aware unit Earth movers is expressed as follows [76]:

$$\text{duEMSC}(L, M) = 1 - \sum_{D \in L_\Delta} \sum_{A \in L_\Sigma} \max(p_L(A \wedge D) - (p_M(A|D) \cdot p_L(D)), 0)$$

Where L, M denote an event log and a model, and L_Δ, L_Σ are distributions of data sequences and activity sequences observed in the log. In order to apply duEMSC to a given log L and Exo-SLPN M , data distribution L_Δ needs to be defined and the joint probabilities p_L, p_M for a given activity and data sequence, but more so in this case a sequence of TTES mappings. The first step is to consider the representation of xlog (Definition 2) as a multiset of pairs for activities and data, stepwise walking traces and generating TTESes for each step.

$$L = (\rho, \Sigma) = \bigcup_{(\sigma, E) \in \Sigma} \bigcup_{k=1}^{\Sigma((\sigma, E))} [(\sigma, \bigcup_{1 \leq i \leq |\sigma|} \langle \text{trans}'(\text{trun}(E, \text{time}(\sigma_i))) \rangle)] \quad (4.17)$$

Where Σ is a multiset of traces annotated with a mapping of exo-series E (Definition 2). Using this form the distributions for sequences over activities A and TTES $\mathcal{E}$ are straightforwardly derived from the multiset representation. Noting that TTES mappings E are continuous, requiring discretisation using a precision reduction factor. Now the probabilities from the log L , given a sequence of activities A and a sequence of TTES mappings $\mathcal{E}$ can be expressed as:

$$p_L(A \wedge \mathcal{E}) = \frac{L((A, \mathcal{E}))}{|L|} \quad p_L(\mathcal{E}) = \frac{||\mathcal{E}^{L((A, \mathcal{E}))} || (A, \mathcal{E}) \in L||}{|L|} \quad (4.18)$$

The last step to consider is the probabilities for all paths T through the Exo-SLPN M that induce the activity sequence A under the assumption of the sequence of mapped TTESes $\mathcal{E}$. Note that the sequence $\mathcal{E}$ must be as long as a path $t \in T$, as such padding is applied to ensure that silent transitions use the next following mapping if available, or the last preceding mapping. Now, the model probability for a given activity sequence A under the assumption of $\mathcal{E}$, is the sum probability of all paths T that induce A , expressed as:

$$p_M(A|\mathcal{E}) = \sum_{\langle t_1, \dots, t_n \rangle \in T} \prod_{i=1}^n \frac{w(t_i, \mathcal{E}_i)}{\sum_{t' \in \text{enbl}(M_i)} w(t', \mathcal{E}_i)} \quad (4.19)$$

Where M_i is the deterministic marking before firing a transition, e.g. $M_1 = M_o$ and $M_o \xrightarrow{t_1} M' \rightarrow \dots \xrightarrow{t_{i-1}} M_i$. Now, duEMSC can be denoted using the definition of an xlog L and Exo-SLPN M , expressed as:

$$\text{duEMSC}(L, M) = 1 - \sum_{\mathcal{E} \in L_\Delta} \sum_{A \in L_\Sigma} \max(p_L(A \wedge \mathcal{E}) - (p_M(A|\mathcal{E}) \cdot p_L(\mathcal{E})), 0) \quad (4.20)$$

4.8 Evaluation

This section compares the proposed discovery technique, presented in Section 4.6, against several existing techniques across several publicly available event logs. The proposed technique has been implemented within the ProM framework¹, and evaluation data has been made publicly available to the best of my ability and inline with ethics².

4.8.1 Event logs and Exogenous Data

This evaluation uses three event logs, from different domains, paired with exogenous data. The first event log comes from the MIMIC III dataset [47] which contains ‘MICU’ ward admissions focusing on the first 48 hours of admission and follows the preparation outlined in [24]. For this log, blood pressure measurements collected from nurse observations are used as exogenous

¹ Found in the ExogenousData Package: github.com/promworkbench/ExogenousData.

² Evaluation data can be found in: github.com/adambanham/exo-slpn-testing

Table 4.5: Descriptive qualities of event logs used in our evaluation.

log	#traces	#events	#acts	exo-data	mean	std	#datapoints
MIMIC [47]	4482	28524	13	blood pressure	8	106	976367
S.Factory [227]	34	418	12	crane jib x	139	128	13038
				crane jib y	43	46	13038
				motorspeed	-21	274	1602128
R. Fines [25]	150370	561470	11	#unpaid fines	32400	14297	4829
				\$unpaid fines	2261806	933700	4898

data to understand what activities are related to blood pressure changes. The second log comes from a smart factory [227], where only the ‘WF_101’ process is considered and the start events within these executions (as these are associated IoT sensors). This log was paired with IoT sensors used in the factory as exogenous data, where extracted time series for the positioning of a crane and the motor speed of the crane are used. These sensors were selected as the crane may require human adjustment or trigger further actions.

The third log used is the road fines event log [25]. For this log, inter-case variables from the log were used as exogenous data, i.e. the total number of unpaid fines and the amount of unpaid fines seen in the event log. These inter-case time series were created using data points from all events, and then aggregated into daily observations. These variables are used to understand and answer the running example’s question in Section 4.2. All three logs are described in terms of control flow, size and exogenous data in Table 4.5. The testing used an AMD Ryzen 9 3900XT with 64GB RAM, but during testing excessive amounts of system RAM usage was observed (upwards of 256GB) to compute conformance checking, as such both the MIMIC III and road fines log were sampled³.

4.8.2 Log Completeness

To show that the approach can identify the stochastic nature of a process, this section considers how the model quality changes as progressively more complete samples of a given log are used. The intuition being that with a more complete understanding of the original log, the discovered stochastic nature should better reflect the original or at least not degrade. 25 sample logs of the road fines log were created, where the n -th sample consists of $1000 \cdot n$ traces, and each progressive larger sample contains all samples from previous sample, i.e. given samples s_i, s_{i+1}, s_{i+j} then $s_i \subset s_{i+1}$ and $(s_i \cup s_{i+1}) \subset s_{i+j}$ where $i \geq 1$ and $j > 1$. Note that the approach considers the temporal nature of each trace, so traces with the same control-flow will have a minor temporal difference and these introduce/expand the choice data used for discovery

³ See github.com/adambanham/exo-slpn-testing.

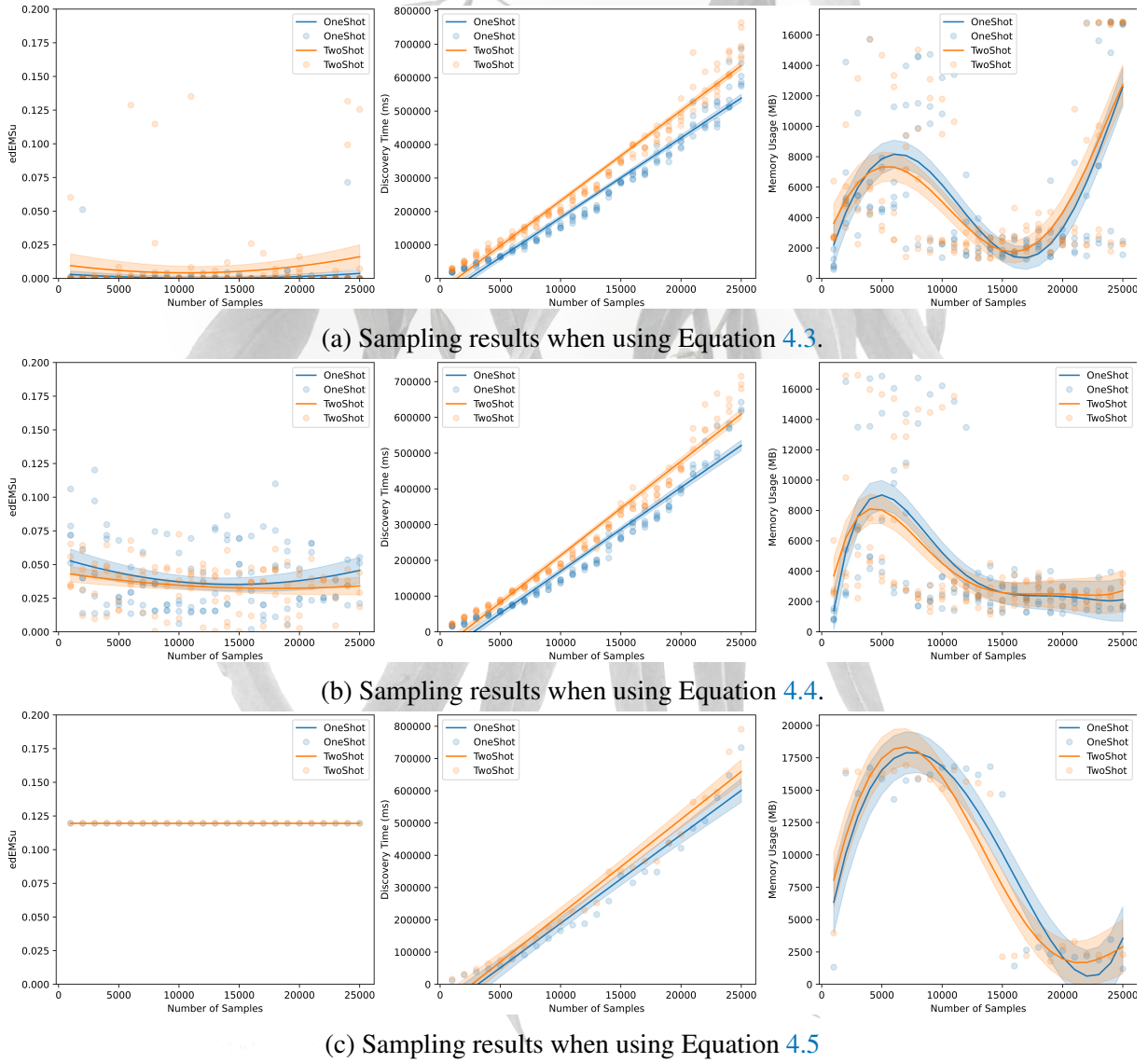


Figure 4.19: Results from sampling testing using the road-fines log. The line of best fit and 95% confidence levels are shown to evaluate the trends across runs.

(Section 4.6.1). Furthermore, as the sample gets larger, the number of sojourns considered for model moves (Section 4.6.2) will be increased and will affect the runtime of discovery.

For each sample, an Exo-SLPN was discovered (recording runtime and memory usage) and then using the complete log, duEMSC was computed to quantify the quality of the discovered Exo-SLPN. A sample size of 25,000 traces for the road fines log was used for testing to avoid excessive system requirements. For a control-flow model the normative model (Figure 4.2) was used. Then, testing was conducted several times for both one-shot and two-shot optimisation (Section 4.6.3). The results of the sampling testing across the alternative weight functions are shown in Figure 4.19.

The testing on the multiplicative form (Equation 4.3) in Figure 4.19a shows that the equation may overly focus on the exogenous influences and struggle to find the balance between endogenous and exogenous factors. As demonstrated by duEMSC measurements being on av-

erage below 0.01 across the sampling with some minor spikes. This struggle may be due to the exogenous influence being modelled as a power of ψ , but interestingly higher quality models were produced by the two-shot approach. Next, the testing for the additive form (Equation 4.4) in Figure 4.19b shows that the equation has a higher baseline than Equation 4.3 when capturing the stochastic nature, likely due to the exogenous influence being less pronounced. However, Equation 4.3 produced a higher quality model than Equation 4.4 highest, albeit rather inconsistently, but a large amount of variance can be seen for both equations. The opposite trend occurs for the global additive form (Equation 4.5), which reports no variance at all and a constant reading of 0.119 regardless of sample size. I posit that using Equation 4.5 means that exogenous influences are treated as constants and cancel each other out when competition occurs between transitions, which in turn removes all difficulty for optimisation.

Interestingly, the pairwise comparison of two-shot and one-shot optimisation, showed that two-shot has the potential to produce higher quality models, but the one-shot approach was more consistently better in Figure 4.19a and Figure 4.19b. Unsurprisingly, the discovery time is linear with the size of the log due to the nature of sojourn calculations, and memory usage was similar between all approaches. Furthermore, the additional runtime needed to perform the two-shot optimisation approach is minor in comparison to the possible quality increase, as such the two-shot optimisation was adopted in the following evaluation.

4.8.3 Model Quality

To compare the proposed approach against existing ones, a variety of stochastic extensions of Petri nets were discovered and considered how close these nets represent the stochastic nature of a log. As existing techniques and the proposed approach require a control-flow model to discover a stochastic Petri net, the following discovery techniques were used: the inductive miner [36] (IM_f), the directly flows miner [38] (DFM), and the POWL miner [35] (IM_{po}) using the default settings. Additionally the normative model [55, Sec 12.1.3] for the road fines log was included, which contains repeated activities which inductive miners cannot find. These techniques were selected as they discover process trees which ensure that the converted Petri net is sound and alignments can be computed. To quantify the quality of a stochastic Petri net, the unit earth movers distance [75] or the data-aware variant [76] was used (see Section 4.7). For these measures, a measurement closer to 1.0 reports that the model perfectly describes the stochastic nature of the log. Note that directly comparing between control flow and data aware stochastic conformance measures is discouraged as they are comparing on different state spaces.

The procedure for discovering and quantifying a model consisted of: (i) discovering a control-flow model with the original log, (ii) sampling the original log with replacement, (iii) discovering stochastic weights using the sampled log and control-flow model, and (iv) measure the discovered stochastic model using the original log. The same sampled log is used across techniques. As noted in Section 4.8.2, the variability of outcomes produced from the approach is high, as such the evaluation was repeated five times and the results reported on the highest

Table 4.6: Conformance for discovered models with our method highlighted. Highest control-flow scores are bolded. Highest data-aware scores are underlined.

	Disc.	Alig. Est.[183]	Sign. Deps.[184]	Data Deps.[76]	Equation 4.3	Equation 4.4	Equation 4.5
Unit Earth Movers [75]				Data-Aware Unit Earth Movers [76]			
MIMIC	IM _f	0.0546	t/o	<u>0.0020</u>	0.0012	0.0001	0.0001
	IM _{po}	0.0031	t/o	0.0006	<u>0.0187</u>	0.0012	0.0000
	DFM	0.1713	t/o	0.0064	<u>0.0147</u>	0.0063	0.0067
S. Factory	IM _f	0.0000	0.0000	0.0000	<u>0.0000</u>	0.0000	0.0000
	IM _{po}	0.1205	0.0000	0.0001	<u>0.1176</u>	0.1149	0.0592
	DFM	0.1282	0.1258	0.0006	<u>0.1176</u>	0.1150	0.0295
R. Fines	normative	0.2573	0.2933	<u>0.1479</u>	<u>0.0253</u>	0.0540	0.1179
	IM _f	0.1003	0.3865	<u>0.1654</u>	0.0701	0.0760	0.0413
	IM _{po}	0.2201	0.3955	<u>0.3326</u>	0.0437	0.0101	0.0100
	DFM	0.8178	0.8178	<u>0.5100</u>	0.4266	0.3779	0.2148

quality outcome.

Results

Table 4.6 shows the testing results for existing and proposed approaches (far-right columns). Neither control-flow techniques reported 1.0 for any of the used logs, highlighting the need to consider factors outside the control-flow of processes. However, Sign. Deps. [184] was unable to discover a model within 24 hours for the mimic log as it was solving over 50,000 parameters. Surprisingly, neither Alig. Est. [183] or Sign. Deps. reported to capture more than 50% of the stochastic nature of the road fines log using the normative model. All techniques favoured using a more literal and less structured model, i.e. DFM, for the road fines log. Notably, all measurements for the IM_f model and the S. Factory log reported 0.000 due to the model being unable to replay traces.

Moving to the data-aware setting, across the mimic and factory log, the proposed approach outperforms the existing Data Deps. [76] technique in four out of five cases. Showing that Exo-SLPNs can be competitive with other stochastic nets in unstructured processes. However, Data Deps. outperformed in the more structured case, i.e., the road fines log where there is a clear connection between event attributes and outcomes. Comparing the different weight functions, Equation 4.3 produced the highest quality model in seven out of nine cases. These results may indicate that the multiplicative form has a greater capacity to capture the stochastic natures of processes, albeit with greater complexity for optimisation.

4.8.4 Changes in Firing Likelihoods

Consider the discovered Exo-SLPN's for the MIMIC log and their changes in likelihoods in Figures 4.20, 4.21 and 4.22. Across all control-flow models, all activities show some change in their likelihood based on blood pressure. This is to be expected as it is unlikely that a medical procedure occurs solely based on this one vital sign. The visualisations for IM_f and IM_{po}

Exo-SLPNs (Figures 4.20 and 4.21) show that the most influenced activity is the 18 gauge⁴, which has the widest distribution of changes in firing likelihoods. However, the silent transitions in the DFM Exo-SLPN (Figure 4.22) show the greatest change in firing likelihoods. It is interesting to note that the IM_f and IM_{p0} reported that the ‘no action...’ transition, which signals the end of the 48 admission period, is unlikely given recent updates to blood pressure. Also, this finding is not reflected in the Exo-SLPN extended from the DFM model. Further work is needed to validate and evaluation the proposed visualisation, at this stage I posit it simplifies the complexity of parametrised stochastic Petri nets.

4.9 Reflections and Future Work

This section reflects on the development of Exo-SLPNs to highlight future work. The proposed discovery technique for Exo-SLPN is able to explore and describe exogenous influences, but some limitations remain. The casting of the optimisation problem for finding suitable ϕ , φ , ψ in a one-shot manner over choice data may induce a large memory requirement on the system responsible for solving the optimisation. This issue draws on fact that the choice data presented here, will likely be unique for each event of the log, or the number of equations produced will approach the number of events observed in the log. Typically, the optimisation requires a matrix of $N \times M$, where N is the number of parameters (ϕ , φ , ψ) and M is the number of unique equations, which may induce a large memory requirement. An opportunity for future work is investigating if the optimisation problem can be distributed into smaller problems and solved independently, which would reduce runtime and memory requirements of the current implementation.

The visualisation tool proposed in Section 4.6.4 and its usefulness for users is under-explored. For the purpose this chapter, the visualisation was used to simplify the often complex semantics of data-aware stochastic process mining as seen in Section 4.2. Further work should be conducted using Design Science Research [88]–[90] on the development of the artefact and its capacity to convey the change in the stochastic language between base weights ϕ and parameters φ , ψ . It is important to consider how non-experts will interpret such visualisations as well as researchers.

The efficiency of the estimation of log-moves could be improved by considering a probability density function for handling sojourn times. Alternative forms for transforming exo-series for binary, discrete, categorical or ordinal data could be considered. Next, a phased approach for discovering weights could be developed which may enable arbitrary stochastic Petri nets to be used. Further investigations into reductions and groups as identified by Leemans, Marnel, and Sidorova [184] could help runtime performance for stochastic process mining using an optimisation approach. It would be interesting to investigate how an arbitrary weight func-

⁴ A type of catheter used to feed/remove fluid from a patient.

tion proposed for control-flow or data-aware version could be extended to include exogenous parameters as considered in this chapter. The interaction between an arbitrary weight function and exogenous parameters may be very difficult to develop. Lastly, work into handling an almost uniquely defined equation space as opposed to a dense equation space would be useful for both control-flow and data-aware stochastic process mining for estimating weights.

4.10 Summary

This chapter presented a novel means which analysts can study whether exogenous data influences the behaviour of processes. A new process modelling formalism Exo-SLPN enables the quantification of firing likelihoods based on exogenous data. Then, a discovery technique is was developed for Exo-SLPNs, which cater for imperfect alignments between traces and the control-flow model. Also presented is a new visualisation tool which highlights how the additional parameters, φ and ψ , change the likelihood of a transition firing (Figures 4.9 and 4.18). Lastly, the evaluation of Exo-SLPNs showed that the approach could outperform state-of-art techniques in less structured processes.

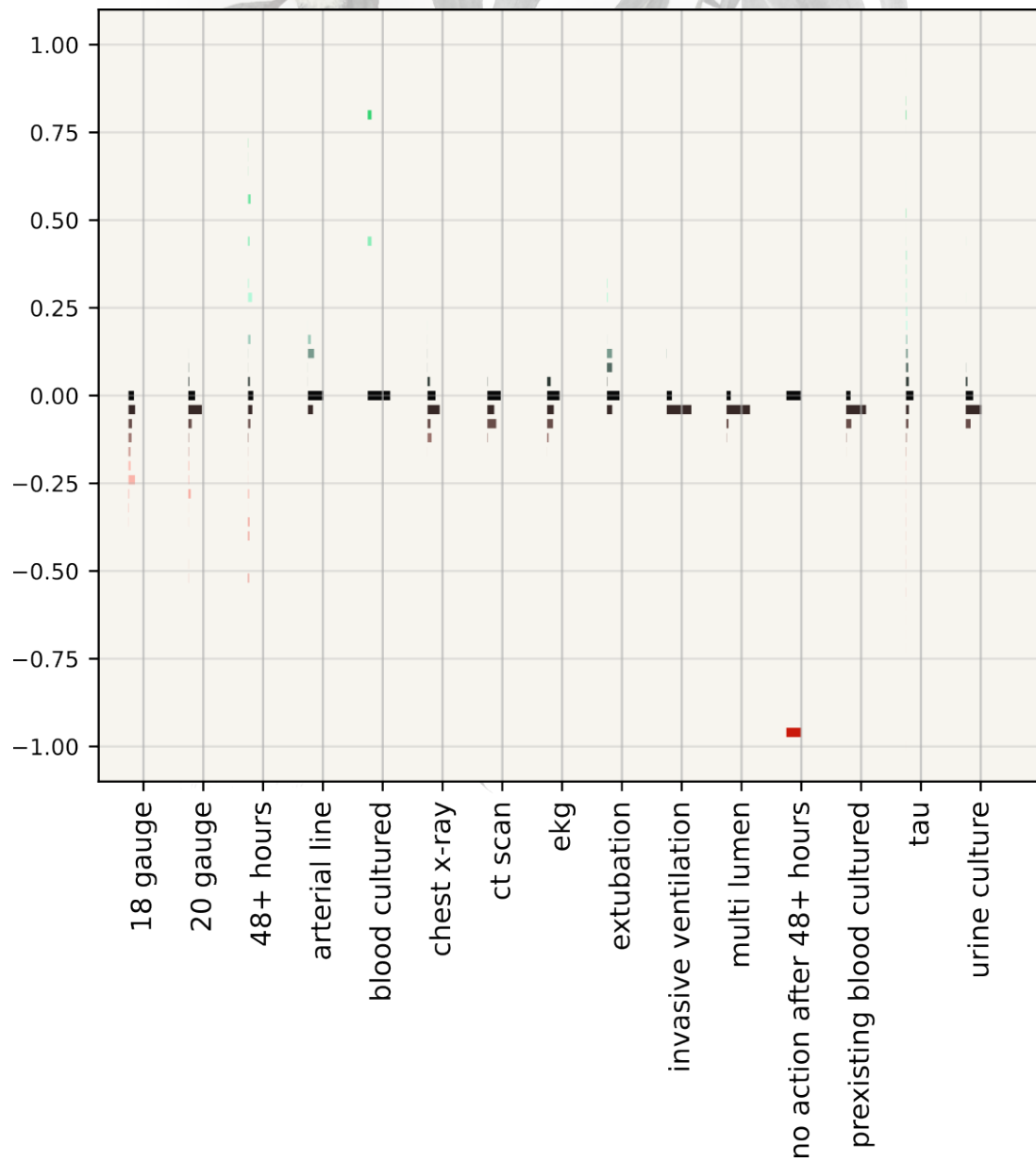


Figure 4.20: Firing likelihoods changes for the MIMIC log using the IM_f control flow model.

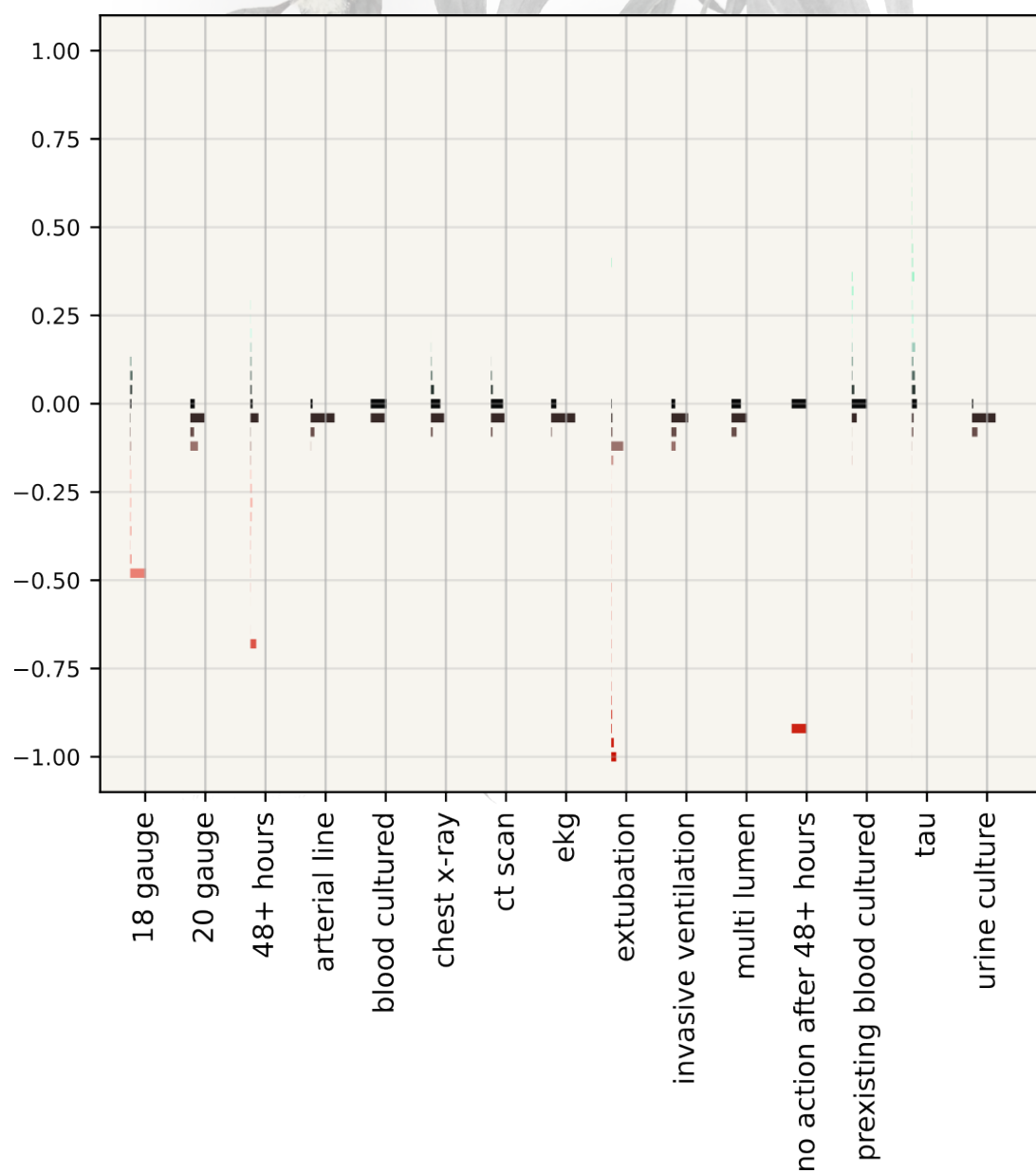


Figure 4.21: Firing likelihoods changes for the MIMIC log using the IM_{po} control flow model.

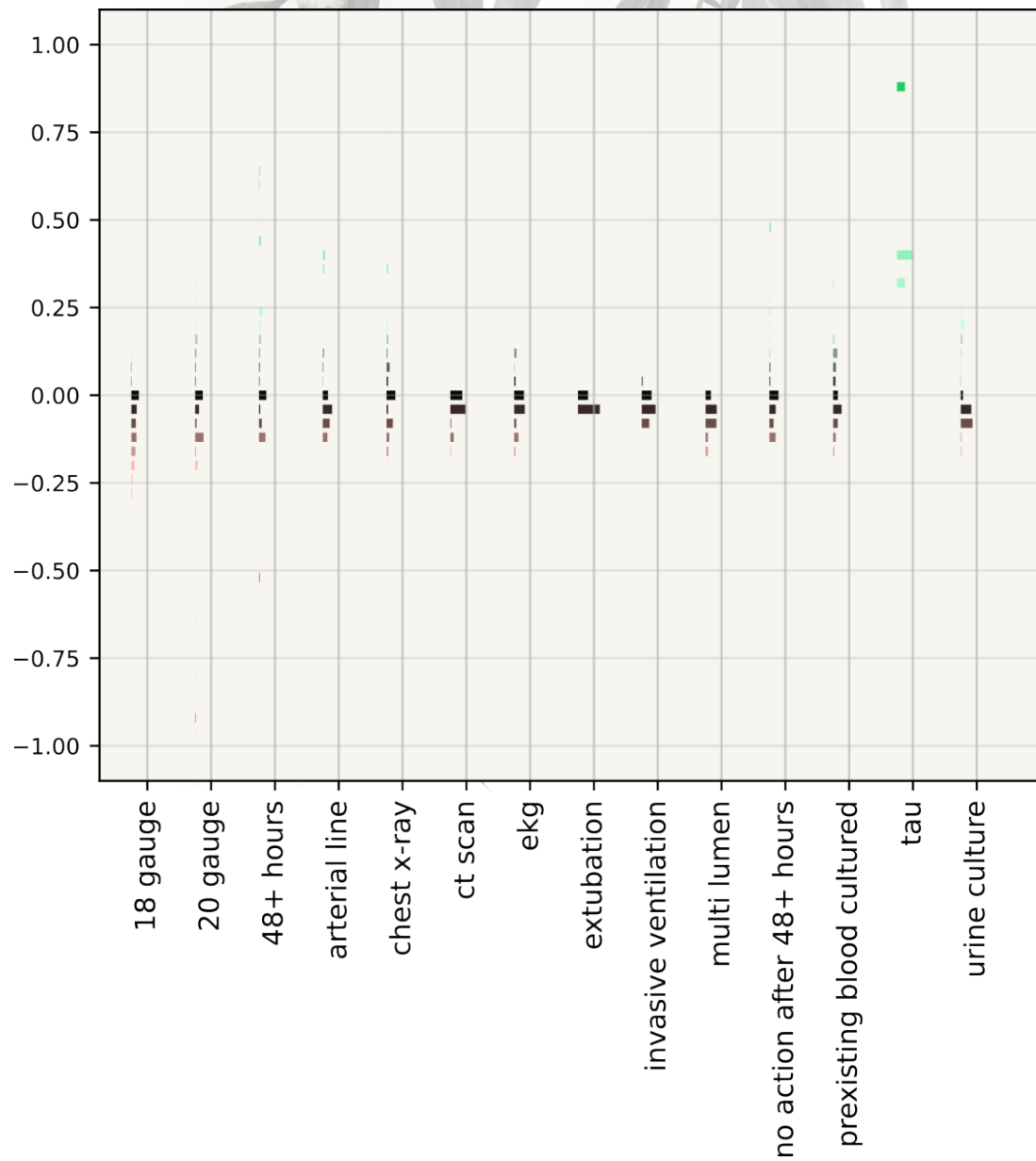


Figure 4.22: Firing likelihoods changes for the MIMIC log using the DFM control flow model.

Statement of Author's Contributions

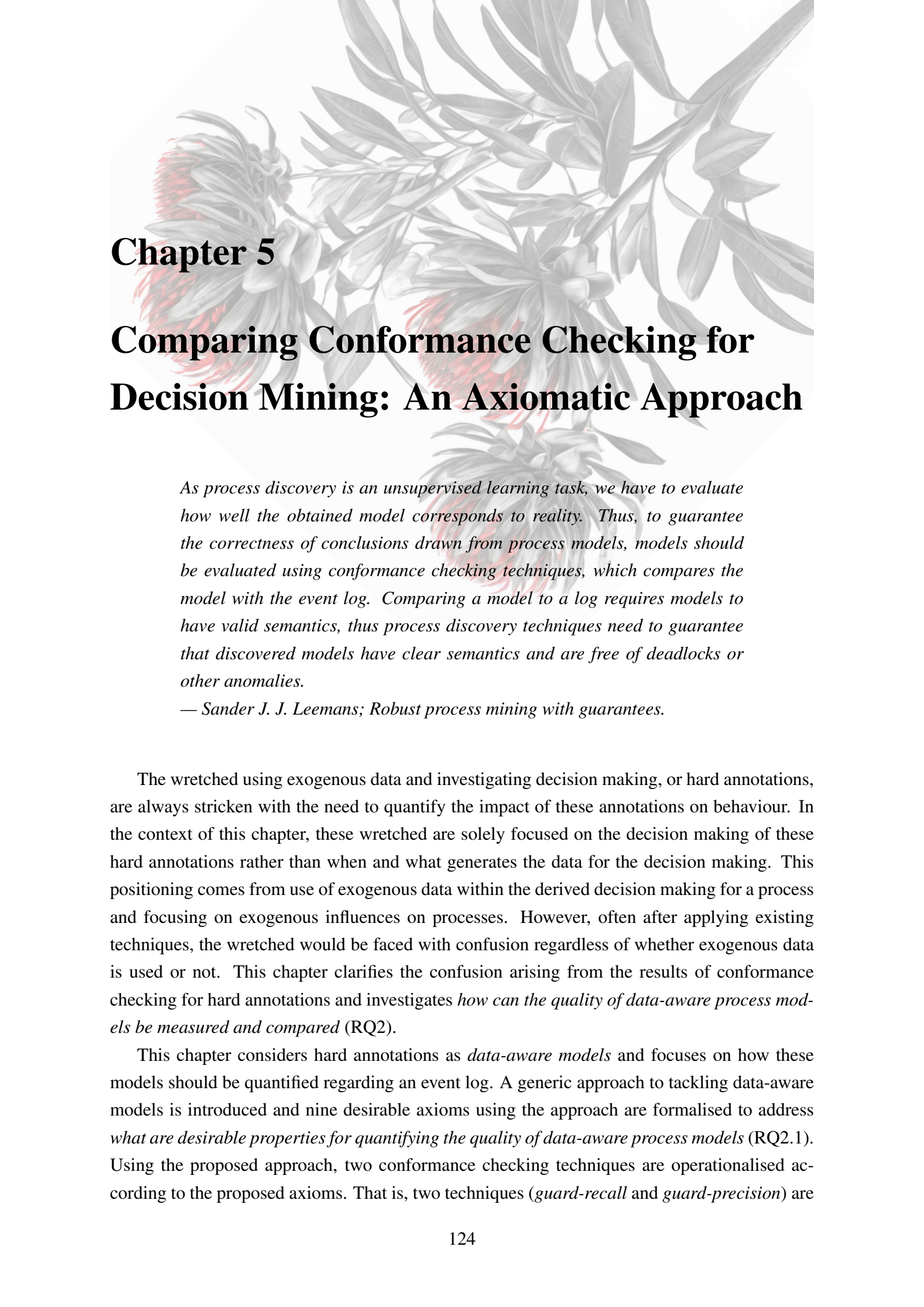
The authors listed below have certified that:

1. they meet the criteria for authorship and that they have participated in the conception, execution, or interpretation, of at least that part of the publication in their field of expertise;
2. they take public responsibility for their part of the publication, except for the responsible author who accepts overall responsibility for the publication;
3. there are no other authors of the publication according to these criteria;
4. potential conflicts of interest have been disclosed to (a) granting bodies, (b) the editor or publisher of journals or other publications, and (c) the head of the responsible academic unit, and
5. they agree to the use of the publication in the student's thesis and its publication on the QUT's ePrints site consistent with any limitations set by publisher requirements.

In the case of this chapter: “Comparing Conformance Checking for Decision Mining: An Axiomatic Approach” was published in the journal *IEEE Access* on 18 April 2024.

To describe contributions, the CRediT taxonomy [190] is used for classifications.

Contributor	Statement of Contribution
Adam Banham (Student)	Conceptualisation; Methodology; Software; Validation; Formal Analysis; Investigation; Writing – Original Draft; Writing – Review & Editing; Visualisation.
Arthur H. M. ter Hofstede	Conceptualisation; Methodology; Formal Analysis; Investigation; Writing – Review & Editing.
Sander J. J. Leemans	Conceptualisation; Methodology; Validation; Formal Analysis; Investigation; Writing – Review & Editing; Supervision.
Felix Mannhardt	Conceptualisation; Methodology; Validation; Formal Analysis; Investigation; Writing – Review & Editing.
Robert Andrews	Conceptualisation; Methodology; Validation; Investigation; Writing – Review & Editing; Supervision.
Moe T. Wynn	Conceptualisation; Methodology; Validation; Investigation; Writing – Review & Editing; Supervision.



Chapter 5

Comparing Conformance Checking for Decision Mining: An Axiomatic Approach

As process discovery is an unsupervised learning task, we have to evaluate how well the obtained model corresponds to reality. Thus, to guarantee the correctness of conclusions drawn from process models, models should be evaluated using conformance checking techniques, which compares the model with the event log. Comparing a model to a log requires models to have valid semantics, thus process discovery techniques need to guarantee that discovered models have clear semantics and are free of deadlocks or other anomalies.

— Sander J. J. Leemans; *Robust process mining with guarantees*.

The wretched using exogenous data and investigating decision making, or hard annotations, are always stricken with the need to quantify the impact of these annotations on behaviour. In the context of this chapter, these wretched are solely focused on the decision making of these hard annotations rather than when and what generates the data for the decision making. This positioning comes from use of exogenous data within the derived decision making for a process and focusing on exogenous influences on processes. However, often after applying existing techniques, the wretched would be faced with confusion regardless of whether exogenous data is used or not. This chapter clarifies the confusion arising from the results of conformance checking for hard annotations and investigates *how can the quality of data-aware process models be measured and compared* (RQ2).

This chapter considers hard annotations as *data-aware models* and focuses on how these models should be quantified regarding an event log. A generic approach to tackling data-aware models is introduced and nine desirable axioms using the approach are formalised to address *what are desirable properties for quantifying the quality of data-aware process models* (RQ2.1). Using the proposed approach, two conformance checking techniques are operationalised according to the proposed axioms. That is, two techniques (*guard-recall* and *guard-precision*) are

proposed and compared against existing techniques using the presented axioms as a yardstick to investigate *what are precision and recall measures that adhere to these desirable properties* (RQ2.2).

5.1 Introduction

Process mining is a discipline that uses observed, historical data from an organisation's business process(es) to understand the actual behaviour and performance of the process(es) [2]. Such historical data is recorded in event logs, consisting of process instances (traces), where each trace describes a series of process activities that unfolded over time [1]. Process discovery techniques [36] exploit these traces to recreate the structuring of activities (control-flow) in the form of a process model. Then, by using conformance checking [41] the quality of a given process model against an event log can be quantified by considering the intersection of observed and described behaviour to measure concepts such as precision, recall, generalisation and simplicity [6], [228]. However, consensus has yet to be reached on the intuitive properties for conformance checking [42], [67], [68], [177].

To motivate which intuitive properties for conformance measures are desirable, a high-level summary of existing discussions [6], [42], [67], [68], [177] is provided. Much of the existing work derives from Buijs, Dongen, and van der Aalst [6], who showed how the behaviour in an event log, process model, and the underlying system can be disjoint or overlapping. Other studies [67], [68] observed that existing conformance measures were not formulated against a set of desirable properties, indicating such measures were not easy to interpret or compare against each other. This issue was confirmed through evaluations performed by [42], [177]. Together, these four studies [6], [42], [67], [68] identified desirable properties, or propositions/axioms, for conformance checking, limited to the control-flow perspective (sequences of activities).

Conformance checking beyond control-flow is also important for processes. In [55], the author considers multi-perspective process mining, which includes activities, resources, data attributes, and expressions (*guards*) for activities. Although [55] introduced conformance checking that considers guards, the study did not consider desirable properties. Therefore, this chapter extends existing discussions [42], [55], [67], [68], by considering desirable properties for data-aware process models [55], [57], [59] to ensure that data-aware models can be compared.

Existing data-aware conformance checking [169], [170], [173], [189] predominantly utilises an alignment-based approach [81] to firstly compute a mapping between a log and a model, and then compute a distance between them [169], [170], [189]. The distance between an alignment and trace in the log, to some extent, is subjective as it is reliant on a domain-relevant cost-model, which specifies what steps are realistic and what are not (and are therefore penalised more). Furthermore, the cost-model defines which alignments are considered “optimal” which means measures over alignments may not be comparable if the cost-models differ between techniques.

A property of alignments that must taken into account when comparing models and an

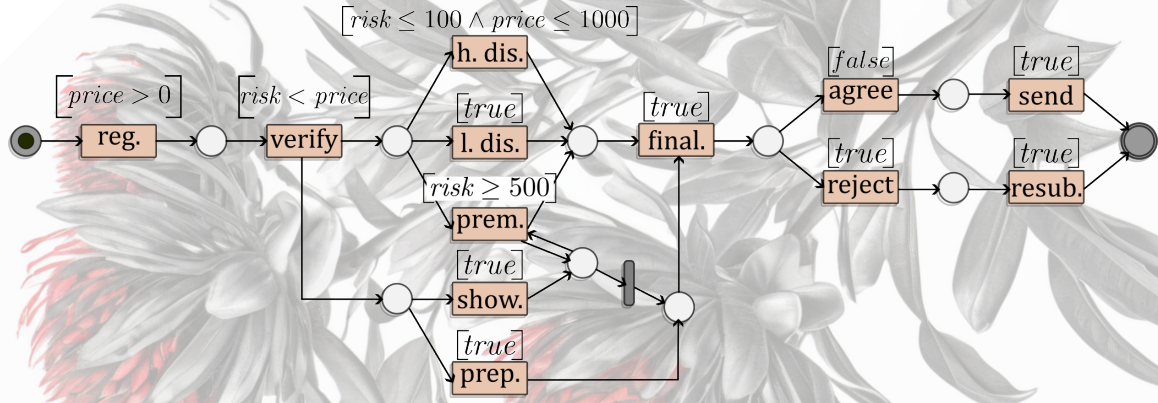


Figure 5.1: An example Petri net with data, with guards above transitions.

event log, is that existing measurements [169], [170], [189] have their state-space uniquely tied to the process model (and underlying cost-model). Thus, it is easy to identify the most compliant log when comparing a single model with many logs. However, it is challenging to identify the highest quality model from a collection of models when considering a single log, as any model-log distance measurement is not guaranteed to be comparable to another due to the possibility that each model and underlying cost-model induce a larger/smaller state-space. Comparing decision mining techniques to identify which technique excels in model-recall or model-precision [6], [41], [68], requires a comparison of many models with a single log. This chapter therefore addresses the challenge of identifying the highest quality model when considering a single log, and the desirable properties that should be considered in order to compare data-aware models.

This chapter is structured as follows: Section 5.2 outlines the approach to the investigation; Section 5.3 discusses the problem space; Section 5.4 presents the notation used; Section 5.5 proposes the new quantification theory; Section 5.6 formalises quality axioms for conformance checking; Section 5.7 defines two conformance checking measures; Section 5.8 evaluates the proposal against existing measures using the proposed quality axioms; Section 5.9 reflects on the chapter; and lastly Section 5.10 summarises the contributions of this chapter.

5.2 Approach

To motivate data-aware conformance checking, consider a representation of a data-aware process model in Figure 5.1. This model describes an example process for the purchase of a financial product and depicts the concepts of choices, data and guards.

Central to the contributions of this chapter is the notion of ‘choices’ in models. A choice represents a state in the model where a set of more than one process activities could be executed next, and from which only one activity can happen. For instance, Figure 5.1 shows that after completion of ‘final’, both ‘agree’ and ‘reject’ are possible next activities, but the occurrence of one means the other cannot occur. Transitions in this model have been annotated with *guards*,

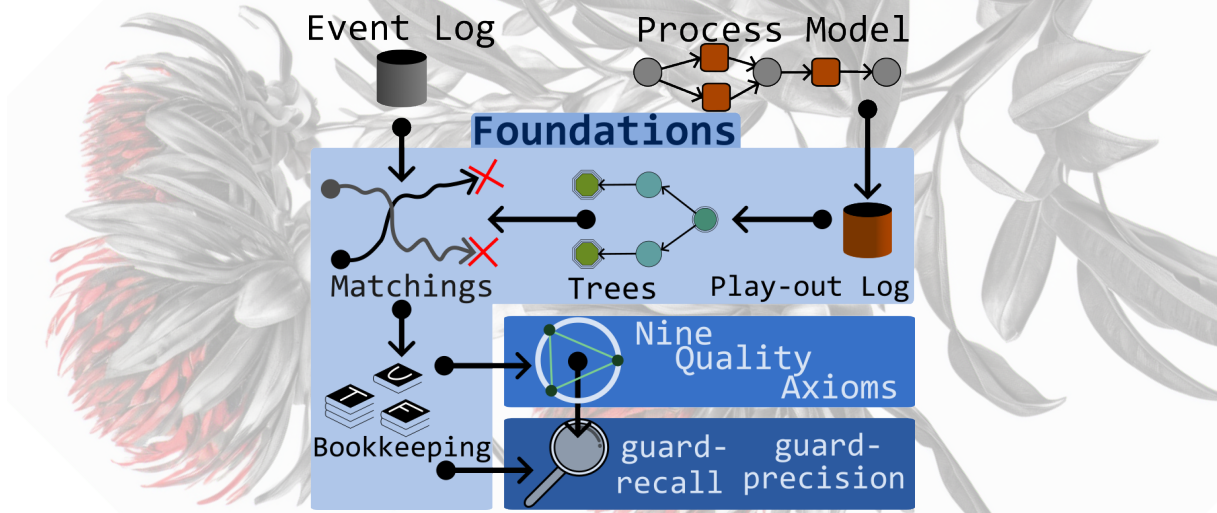


Figure 5.2: The three conformance checking contributions that address RQ2.

some of which for example, the guard for ‘h.dis’, are defined over process data attributes, while others, like guards for ‘show’ and ‘agree’, are indifferent to data.

The semantics of guards may differ (see Section 2.5.1). For instance, the first two guards (for ‘reg.’ and ‘verify’) are domain policies that should be enforced in practice, and are challenging to discover (see [229]). One could consider guards that prescribe how data should change after a process activity [185], [204], however, the following three guards for ‘h.dis.’, ‘l.dis.’, ‘prem.’ are guards that determine, from data, what process activity should occur next. The latter is a specific concern for decision mining techniques [55], [57], [59] and this viewpoint is emphasised for the contributions in this chapter.

Quantifying the quality of guards is often restricted to their respective model representations required by the techniques performing the quantification [55], [57], [170], [189], [204]. A comparison of any data-aware model with an event log, means foregoing specific representations of these models. While, visualisation of models may use Petri nets with data (DPN) notation, the proposed approach does not rely on DPNs. Notably, any use of alignments require the control-flow of a model to always have the option to complete (see [2, clause 3 of Def.3.7]), but the approach undertaken in this chapter does not require models to be relaxed data sound¹ (see [55, Def. 5.1]) as existing decision mining techniques do not guarantee this property in their outcomes [55], [57], [59].

Figure 5.2 outlines the contributions for the chapter. The foundations (Section 5.5) present a generalised modelling formalism which enables quantification through following four steps of: (i) inducing a set of execution paths from an executable process model [68] or a *play-out log*; (ii) constructing a *finite transition tree* from these execution paths based on a given event log; (iii) mapping traces in the log to paths through the tree through a *matching*; and, (iv) collecting diagnostic information about the guards within the tree through *bookkeeping*

¹ The model does not have one single valid firing from its start to end state.

functions for conformance checking. The next step is to formulate the desirable properties for data-aware conformance checking using the bookkeeping through nine quality axioms (Section 5.6). Following on two new measures, guard-recall and guard-precision were developed to address these axioms (Section 5.7). Finally, these measures and existing techniques [170], [173], [189] are evaluated for adherence to the axioms (Section 5.8) to establish how to compare data-aware models.

5.3 Problem Motivation

At a very high level, decision mining finds ‘human-readable’ guards for activities in a process model [230] which guide the execution of a process model at runtime, and may lead to a deterministic execution. Leewis, Smit, and Zoet [231] observed that different techniques exist for achieving this goal, techniques which may either consider an existing control-flow and annotate the control-flow with guards, or adopt a decision-aware approach to process discovery [230], [231]. This chapter focuses on the control-flow annotation approach and its properties for data-aware conformance checking.

A typical decision mining technique using the annotation approach will consist of the following five steps for a given model and event log: a) identify decision points in the model, b) find a control-flow mapping between the log and the model, c) construct classification problems for each decision point, d) learn a classifier for each problem, and then e) extract guards over a set of attributes from each classifier, which are then annotated to activities within the model as *guards*.

A natural consequence of applying this approach is that all possible outcomes (i.e. data-aware models) will have the same control-flow, with differences between the outcomes arising due to the restrictions on model behaviour imposed by the various, different guard annotations. As such, this chapter investigates desirable qualities for conformance checking when considering models derived from the same control-flow to ensure measurements over these models are comparable.

An important consideration for conformance checking is that measures should not be bound to representation, e.g. Petri nets or extensions thereof, such as in Figure 5.1. However, this requirement does pose a problem with step a) identifying decision points or choices between process activities, as it requires some understanding of the representation of the model. An abstraction of process models is used, namely transition trees [66] to address the representation problem. This model abstraction enables the identification of decision points, however some cases might result in an infinite tree when considering loops and concurrency. But, given the focus on what is observed for conformance checking, the cases that result in infinite tree are largely not a concern and can be avoided.

A further justification for the use of transition trees comes from the generalisation of process execution described in [66] and illustrated in Figure 5.3. Where process execution is conceived

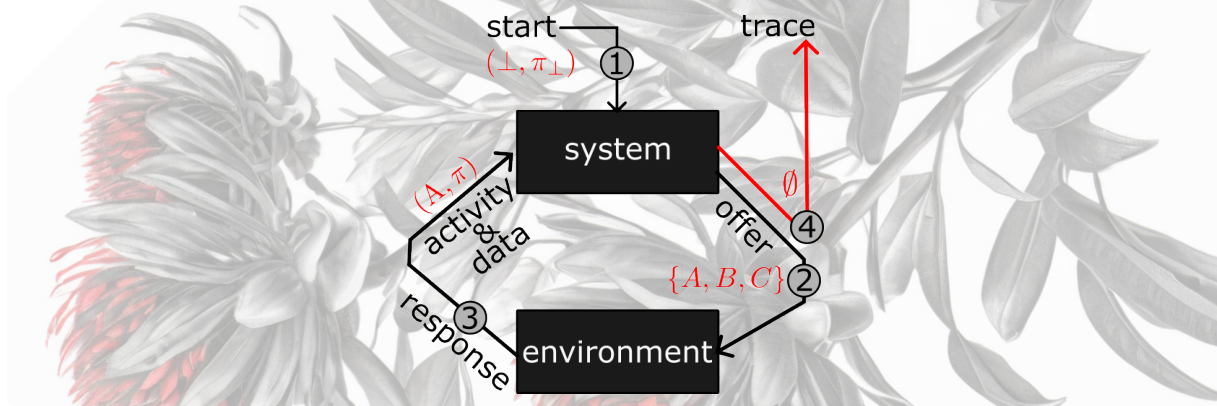


Figure 5.3: Conceptualisation of a system reacting to data and process activities [66], where the system signals that a trace is complete when it presents an empty offer $\emptyset$.

as a system which reacts to process activities and data (1 or 3), by presenting offers (i.e. a set of process activities) to an environment (2). Next, the environment responds to the offer, with a process activity and additional data back to the system (3). Then, the system adapts to the response, and this cycle of offers and responses is repeated until a process execution is completed (4). The iterative relationship between system and environment both builds a trace, and is naturally represented as a transition tree. Transition trees are therefore used for process models for this chapter and will show that executable models can be represented by a tree.

This chapter uses the offers generated by the system to position conformance checking regarding precision and recall. For example, where a decision mining outcome restricts all offers presented at runtime by the system to only a single process activity, the process is entirely deterministic, and precision therefore is maximised. If any offer(s) consist of more than one process activity, the process is non-deterministic, implying precision will be less than maximal. Likewise, if any offer(s) are empty at runtime, the process will terminate early, which will negatively impact on recall. As such, these issues will be introduced as the notions of guard-precision and guard-recall, a specialisation of data-aware conformance checking for the quantification of data-aware models.

Typically, an alignment procedure [81] is used for step b), to find a control-flow mapping between the event log and model. An alignment between a log and a model is a case-specific sequence of steps, which are either *sync steps* where an event is matched with an execution step, *log steps* where an event without an execution step, or *model steps* where an execution step without an event. For Petri net-based representations, alignment procedures that find the “optimal” alignments have been studied at length [81], [169]–[173], but the agreement on the notion of optimal and its exact properties is an active research endeavour.

However, in step c), constructing classification problems for each decision point, alignments are generalised to include only *sync* and *log* steps to ensure that only behaviour observed in the log is used in learning guards for the model. This generalisation of a control-flow alignment is referred to as a “*matching*” and used throughout the rest of this chapter.

5.4 Preliminaries

This section outlines the notation used in this chapter as well as introducing guards, events, traces, event logs and process models.

Functions. Functions are denoted as $f: X \mapsto Z$, where f is a function that maps each element of the set X to an element of the set Z . A function f is a bijection if no two different elements of X are mapped to the same element in Z and all elements in Z have an original, i.e. an element from X that is mapped to it.

Bags. A *bag* over X is a function $f: X \mapsto \mathbb{N}$. The bag f captures the frequency of each element from X in the bag. Square brackets denote bags and only non-zero frequencies of elements in X are shown as superscripts, e.g. $[x_1^1, x_2^5, x_3^{10}]$.

Sequences. Angled brackets denote sequences, e.g. $\langle a, b, c \rangle$. A sequence can be constructed by a first element, the head, say h , and another sequence, say t , the tail. The construction of such a sequence is denoted as $h:t$. This construction is widened to $t:h$, where a sequence t is extended with element h .

The length of a sequence, say $s = \langle a, b, c \rangle$, is denoted as $|s|$; e.g. $|s| = 3$. To access an element of a sequence s , s_i is written such that $1 \leq i \leq |s|$, e.g. $s_3 = c$. To access the last element of a non-empty sequence s , the function `last` is used, defined as: $\text{last}(s) = s_{|s|}$. To denote the concatenation of two sequences, the concatenation operator $+$ is used, e.g. $\langle a, b \rangle + \langle c, d \rangle = \langle a, b, c, d \rangle$. To take a segment from position i up to and including position j from a sequence s , the notation $s[i:j]$ is used. For example, $\langle a, b, c, d \rangle[2:3] = \langle b, c \rangle$. By definition, if $i > j$, $s[i:j] = \langle \rangle$. Also, if $j > |s|$ and $i \leq |s|$, then $s[i:j] = s[i:|s|]$.

Universes. Several universes that the notation will build upon are now introduced.

Definition 8 (Universes) $\mathcal{U}_{ev}$ is the universe of event identifiers; $\mathcal{U}_{case}$ is the universe of case identifiers; $\mathcal{U}_{act}$ is the universe of process activities; $\mathcal{U}_{time}$ is the universe of timestamps; $\mathcal{U}_{att}$ is the universe of process attribute identifiers; $\mathcal{U}_{val}$ is the universe of possible values that process attributes can take. The latter universe includes $\perp$, a value that can be used to indicate that an attribute is undefined. All these sets are mutually exclusive.

Data. An *assignment* is the representation of data and is a function from process attributes to values.

Definition 9 (Assignments) $\Pi = \mathcal{U}_{att} \mapsto \mathcal{U}_{val}$ is the set of all possible attribute assignments.

As a process executes, assignments need to be correspondingly updated. Thus, an override $\oplus$ operator as an operator on assignments is used, where the values of the right assignment take precedence over the left assignment.

Definition 10 (Overriding Assignments) Let $\pi_1, \pi_2 \in \Pi$ be assignments, then $\pi_1 \oplus \pi_2 \in \Pi$ is an assignment defined as follows. For all $a \in \mathcal{U}_{att}$:

$$\pi_1 \oplus \pi_2(a) = \text{if } \pi_2(a) \neq \perp \text{ then } \pi_2(a) \text{ else } \pi_1(a).$$

Guards. Guards are used to guide process execution. This chapter assumes no particular language for the specification of guards. In the context of a particular assignment, a guard may evaluate to *true* or *false* or it may not be possible to evaluate the guard in this context.

Definition 11 (Guards) $\mathcal{U}_{\text{grd}}$ is the universe of guards. Then for any guard $g \in \mathcal{U}_{\text{grd}}$, the attributes that the guard is formed over is accessed through $\text{var}: \mathcal{U}_{\text{grd}} \mapsto 2^{\mathcal{U}_{\text{att}}}$, and guards can be evaluated in the context of attribute assignments. In particular, for a guard $g \in \mathcal{U}_{\text{grd}}$ and an assignment $\pi \in \Pi$, $\llbracket g \rrbracket(\pi)$ denotes the evaluation of guard g in the context of assignment π and yields true (T), false (F), or undefined (U).

Events. An event, represented by a unique identifier, describes a state change in a process execution. Each *event identifier* is associated with an assignment of process attributes, a process activity, a case identifier, and a timestamp.

Definition 12 (Events) For every event $e \in \mathcal{U}_{\text{ev}}$, its attribute assignment is accessed through the function $\# : \mathcal{U}_{\text{ev}} \mapsto \Pi$, its activity through the function $\text{act} : \mathcal{U}_{\text{ev}} \mapsto \mathcal{U}_{\text{act}}$, its case identifier through the function $\text{case} : \mathcal{U}_{\text{ev}} \mapsto \mathcal{U}_{\text{case}}$, and its timestamp through the function $\text{time} : \mathcal{U}_{\text{ev}} \mapsto \mathcal{U}_{\text{time}}$.

Traces. A sequence of events σ denotes a *trace* or a single execution of a process if all events – all different – refer to the same case and the time ordering is preserved.

Definition 13 (Trace) A trace σ is a nonempty sequence of events, i.e. $\sigma \in \mathcal{U}_{\text{ev}}^+$, such that:

1. All events of σ belong to the same case: $\forall_{1 \leq i < j \leq |\sigma|} [\text{case}(\sigma_i) = \text{case}(\sigma_j)]$;
2. The sequence of events in σ adheres to their timing: $\forall_{1 \leq i < j \leq |\sigma|} [\text{time}(\sigma_i) \leq \text{time}(\sigma_j)]$;
3. All events in σ are different: $\forall_{1 \leq i < j \leq |\sigma|} [\sigma_i \neq \sigma_j]$.

For convenience, the shorthand of $\text{case}(\sigma)$ is used to access the case of a trace and to obtain the activity sequence of a trace σ , or the *variant* of the trace, $\hat{\sigma} = \langle \text{act}(\sigma_1), \dots, \text{act}(\sigma_{|\sigma|}) \rangle$ is used. Next, to evaluate guards, access is needed to the assignment that holds after a particular position of a trace.

Definition 14 (Running Assignment) Let σ be a trace, let $j \in \mathbb{N}$ be the trace position, and let $\pi \in \Pi$ be an initial assignment. Then, the function μ yields the running assignment of the trace defined as:

$$\mu(\pi, \langle \rangle, j) = \pi \quad \text{and} \quad \mu(\pi, e : \sigma, j) = \begin{cases} \mu(\pi \oplus \#(e), \sigma, j-1) & \text{if } j > 0, \\ \pi & \text{otherwise.} \end{cases}$$

For the rest of the paper, we write $\mu_{\sigma,j}$ instead of $\mu(\emptyset, \sigma, j)$.

Logs. An *event log* contains the completed executions of a process over some period.

Definition 15 (Event Logs) An event log L is a set of traces, such that no two traces $\sigma', \sigma'' \in L$, $\sigma' \neq \sigma''$, share a case:

$$\text{case}(\sigma') \neq \text{case}(\sigma'').$$

In some cases, only the control-flow variants observed in an event log are considered.

Definition 16 (Control-flow Variants) Given an event log L , the set of its control-flow variants $\mathcal{V}_L \subset \mathcal{U}_{act}^+$ is defined as:

$$\mathcal{V}_L = \{\hat{\sigma} \mid \sigma \in L\}.$$

To represent event logs in this chapter, event identifiers are not shown, and instead the variants and bags over the sequences of assignments that occurred for those variants are shown. For example see the representation below, which shows a view on an event log with two process variants and two different assignments for one of these variants:

$$L = \left\{ \langle a, b, c \rangle \mapsto [\langle \pi_1, \pi_2, \pi_3 \rangle^1, \langle \pi_1, \pi_4, \pi_3 \rangle^3], \langle a, d, c \rangle \mapsto [\langle \pi_1, \pi_5, \pi_3 \rangle^2] \right\}.$$

Enlarging. Often notation requires a log which has been enlarged with respect to distribution of variants and assignments. To enlarge a log, traces need to be *cloned*.

Definition 17 (Cloning Traces) Let $\sigma \in \mathcal{U}_{ev}^+$ be a trace. Then, $\bar{\sigma}$ is a clone of σ which satisfies the following properties:

1. the clone is also a trace;
2. the length of the trace is preserved: $|\sigma| = |\bar{\sigma}|$;
3. the sequence of process activities is preserved: $\hat{\bar{\sigma}} = \hat{\sigma}$;
4. the sequence of assignments is preserved: $\forall_{1 \leq i \leq |\sigma|} [\#(\sigma_i) = \#(\bar{\sigma}_i)]$;
5. the trace and its clone do not share events: $\forall_{1 \leq i \leq |\bar{\sigma}|, 1 \leq j \leq |\sigma|} [\bar{\sigma}_i \neq \sigma_j]$;
6. the cloned trace belongs to a different case: $\text{case}(\bar{\sigma}) \neq \text{case}(\sigma)$.

Now that traces can be cloned, an enlargement of an event log is achieved by cloning the traces in the log a number of times.

Definition 18 (Enlarging Event Logs) Let L be an event log, and let $k \in \mathbb{N}$ be a integer scaling factor s.t. $k \geq 1$. For each $\sigma \in L$, we assume $k - 1$ clones: $\bar{\sigma}_2, \dots, \bar{\sigma}_k$ with none of these clones sharing case identifiers or events among them or with any other such clones created for other traces in log L . Then, L^k denotes the enlarging of the event log via cloning, defined as:

$$L^k = L \cup \left\{ \bar{\sigma}_j \mid \sigma \in L \wedge 2 \leq j \leq k \right\}.$$

For example, consider the following enlargement of an event log:

$$\begin{aligned} \text{if } L &= \{ \langle a, b, c \rangle \mapsto [\langle \pi_1, \pi_2, \pi_3 \rangle^1, \langle \pi_1, \pi_4, \pi_3 \rangle^3], \langle a, d, c \rangle \mapsto [\langle \pi_1, \pi_5, \pi_3 \rangle^2] \}, \\ \text{then } L^3 &= \{ \langle a, b, c \rangle \mapsto [\langle \pi_1, \pi_2, \pi_3 \rangle^3, \langle \pi_1, \pi_4, \pi_3 \rangle^9], \langle a, d, c \rangle \mapsto [\langle \pi_1, \pi_5, \pi_3 \rangle^6] \}. \end{aligned}$$

Note that each instance in the bags on the right represents a unique trace; therefore, enlargement of the log in visualisations is seen as multiplying these frequencies by the enlargement, i.e. $1 \cdot 3$, $2 \cdot 3$ and $3 \cdot 3$.

Trees. This chapter abstracts from a specific representation for process models but requires abstractions that capture control-flow and data dependencies. Thus, transition trees [66] are used as an abstraction to denote process models.

Definition 19 (Models) A process model $M = (N, F, n_r, N_f, \text{src}, \text{tgt}, \text{gd}, \text{actf})$ is a rooted and non-cyclic tree, with N a set of nodes, F a set of flows, a root node $n_r \in N$, a set of terminal nodes $N_f \subseteq N$, a function $\text{src}: F \mapsto N$ to yield the source node of a flow, a function $\text{tgt}: F \mapsto N$ to yield the target node of a flow, a function $\text{gd}: F \mapsto \mathcal{U}_{\text{grd}}$ to yield the guard of a flow, and a function $\text{actf}: F \mapsto \mathcal{U}_{\text{act}}$ to yield the activity of a flow.

In the context of a model $M = (N, F, n_r, N_f, \text{src}, \text{tgt}, \text{gd}, \text{actf})$, the following shorthand notation is used. If there is an $f \in F$ such that $\text{src}(f) = n$ and $\text{tgt}(f) = n'$ for $n, n' \in N$, then $n \rightarrow n'$ and $n \xrightarrow{f} n'$ is written. To access the flows that are outgoing from a given node n , then $n \bullet$ is written to denote:

$$n \bullet = \{ f \in F \mid n \xrightarrow{f} n' \}.$$

As a convenience, $\text{splits}(M)$ is used to denote flows in the process model whose source node is source node of at least one other flow in the model, i.e. flows at splits in the model:

$$\text{splits}(M) = \{ f \in F \mid |\text{src}(f) \bullet| > 1 \}.$$

To determine whether two process models M and M' have the same behavioural structure, an isomorphism between them: $M \simeq M'$ is required. For an in-depth consideration of isomorphism see [232]. Two process trees are isomorphic if and only if there exists a bijection between their flows and nodes that preserves sources and targets of flows as well as their activities. Such a bijection is termed an isomorphism, denoted as $\text{iso}_{M'}^M$ or iso for short, when the models involved are clear. Note that this bijection is only considering the control-flow of the models in question.

5.5 Foundations

This section introduces the formal quantification of quality for data-aware models. First, an example data-aware model is introduced, which is used to demonstrate the transformation to a model-agnostic form for data-aware conformance checking. Next, how to combine guards through binary operators on guards referred to as reinforcing or weakening and their formal

properties are presented. These properties allow for simplifications when combining guards from partial executions of a model consisting of infinitely many silent actions (τ) followed by a non-silent action or when alternative execution paths leading to the same visible action are required to be denoted.

The following two subsections introduce how to produce a model-agnostic form of any executable data-aware process model, such that the form conforms with the conceptual view of a system reacting to an environment (see Figure 5.3). The next step introduces how to study the intersection of an event log and a transition tree through a generalisation of alignments called a matching, which only considers synchronised and log moves to avoid the propagation of dead assignments. Lastly, these steps are combined to present the quantification mechanism, called *bookkeeping*, which denotes for a part of the tree, how often it was traversed and how often particular outcomes of the associated guards occurred.

Together, these advancements will be used to present desirable properties for data-aware conformance checking in Section 5.6 and to define guard-recall and guard-precision in Section 5.7.

5.5.1 Running Example

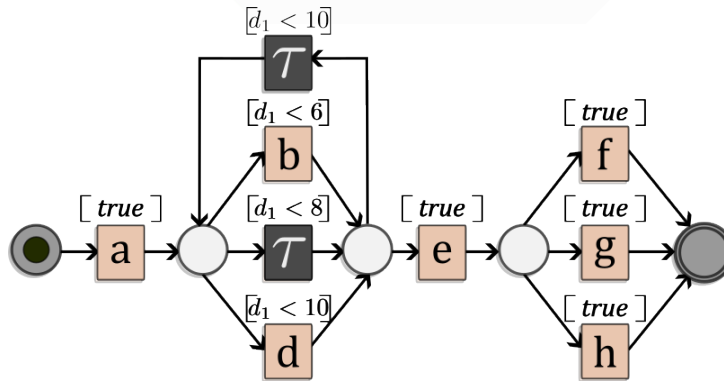


Figure 5.4: Example data-aware model using Petri net with data.

Figure 5.4 introduces the example model for demonstration purposes. This model has one write constraint, i.e. ‘a’ writes attribute d_1 which has an integer domain. This model contains silent actions (transitions labelled with τ) and a rework loop that can occur silently. In this model, infinite system behaviour can occur between ‘a’ and ‘e’ due the silent actions. For simplicity, the guards in Figure 5.4 will be referenced in the following manner:

$$g_{\tau} \equiv \text{true}, \quad g_1 \equiv d_1 < 6, \quad g_2 \equiv d_1 < 8, \quad g_3 \equiv d_1 < 10.$$

5.5.2 Reinforcing and Weakening Guards

This section advances the theory and notation around guards. The motivation for these advancements is that to unfold executions of a model requires combining a series of guards into a single

guard. Such a case appears in Figure 5.4, where the two silent transitions (τ) can be fired many times and requires encoding their guards alongside the guard of the next visible fired transition (either b, d, or e). However, there can be many partial executions that lead to the firing of ‘e’ after ‘a’ in Figure 5.4, and these cases need to be captured as well. To be able to encode these cases, a simplification process over a series of guards is introduced.

The first step toward this goal is to introduce the concept of neutral guards, which yield the same value regardless of the assignment.

Definition 20 (Neutral Guards) A guard $g \in \mathcal{U}_{grd}$ is neutral iff for any assignments $\pi, \pi' \in \Pi$, $\llbracket g \rrbracket(\pi) = \llbracket g \rrbracket(\pi')$. A guard is referred to as truth neutral, $\text{truth}(g)$, iff for any assignment $\pi \in \Pi$, $\llbracket g \rrbracket(\pi) = \top$, while it is referred to as undefined neutral, $\text{undef}(g)$, iff for any assignment $\pi \in \Pi$, $\llbracket g \rrbracket(\pi) = \perp$.

While, the form of guards is abstracted from, contributions need to reason about their evaluation behaviour. Thus, equivalence between two guards is introduced.

Definition 21 (Equivalent Guards) Guards $g_1, g_2 \in \mathcal{U}_{grd}$ are equivalent, $g_1 \equiv g_2$, iff for all $\pi \in \Pi$, $\llbracket g_1 \rrbracket(\pi) = \llbracket g_2 \rrbracket(\pi)$.

To create stricter guards, the *reinforce* operator $\otimes$ is introduced. Reinforcing a guard with another, means that the reinforcement is a guard that returns undefined if *either* guard is undefined, returns true if *both* guards are true, otherwise it returns false.

Definition 22 (Reinforcing Guards) Let $g_1, g_2 \in \mathcal{U}_{grd}$ be guards, and let $\pi \in \Pi$ be an assignment. Then, the reinforce operator $\otimes$ over guards is defined as:

$$\llbracket g_1 \otimes g_2 \rrbracket(\pi) = \begin{cases} \perp & \text{if } \llbracket g_1 \rrbracket(\pi) = \perp \text{ or } \llbracket g_2 \rrbracket(\pi) = \perp, \\ \top & \text{if } \llbracket g_1 \rrbracket(\pi) \text{ and } \llbracket g_2 \rrbracket(\pi), \\ \text{F} & \text{otherwise.} \end{cases}$$

From this definition, the following proposition immediately follows.

Proposition 1 (Properties of $\otimes$) Let $g_1, g_2, g_3, \varrho_{\top} \in \mathcal{U}_{grd}$ be guards such that $\text{truth}(\varrho_{\top})$. Then, it follows that the reinforce operator $\otimes$ has the following properties:

- a. if $g_1 \equiv g_2$ then $g_1 \otimes g_2 \equiv g_1$; (idempotent)
- b. $g_1 \otimes g_2 \equiv g_2 \otimes g_1$; (commutative)
- c. $(g_1 \otimes g_2) \otimes g_3 \equiv g_1 \otimes (g_2 \otimes g_3)$; (associative)
- d. $g_1 \otimes \varrho_{\top} \equiv \varrho_{\top} \otimes g_1 \equiv g_1$. (neutral element)

Using these properties, then the following reduction can be applied to any reinforcement of guards.

Lemma 1 (Strong Idempotence of $\otimes$) Let $g_1, \dots, g_n \in \mathcal{U}_{\text{grd}}$ be guards and let $g \in \mathcal{U}_{\text{grd}}$ be another guard, then if $g \equiv g_i$ for some $1 \leq i \leq n$:

$$\bigotimes_{1 \leq i \leq n} g_i \otimes g \equiv \bigotimes_{1 \leq i \leq n} g_i.$$

To create laxer guards, the *weaken* operator $\ominus$ is introduced. Weakening a guard with another, means that the weakening is a guard that returns undefined if *both* guards are undefined, it returns true if *either* guard is true, otherwise it returns false.

Definition 23 (Weakening Guards) Let $g_1, g_2 \in \mathcal{U}_{\text{grd}}$ be guards, and let $\pi \in \Pi$ be an assignment. Then, the weaken operator $\ominus$ over guards is defined as:

$$\llbracket g_1 \ominus g_2 \rrbracket(\pi) = \begin{cases} \text{U} & \text{if } \llbracket g_1 \rrbracket(\pi) = \text{U} \text{ and } \llbracket g_2 \rrbracket(\pi) = \text{U}, \\ \text{T} & \text{if } \llbracket g_1 \rrbracket(\pi) \text{ or } \llbracket g_2 \rrbracket(\pi), \\ \text{F} & \text{otherwise.} \end{cases}$$

Similar to before, the following proposition arises from Definition 23.

Proposition 2 (Properties of $\ominus$) Let $g_1, g_2, g_3, \varrho_{\text{U}} \in \mathcal{U}_{\text{grd}}$ be guards such that $\text{undef}(\varrho_{\text{U}})$. Then, it follows that the weaken operator $\ominus$ has the following properties:

- a. if $g_1 \equiv g_2$ then $g_1 \ominus g_2 \equiv g_1$; (idempotent)
- b. $g_1 \ominus g_2 \equiv g_2 \ominus g_1$; (commutative)
- c. $(g_1 \ominus g_2) \ominus g_3 \equiv g_1 \ominus (g_2 \ominus g_3)$; (associative)
- d. $g_1 \ominus \varrho_{\text{U}} \equiv \varrho_{\text{U}} \ominus g_1 \equiv g_1$. (neutral element)

Using these properties, the following reduction to any weakening of guards can be applied.

Lemma 2 (Strong Idempotence of $\ominus$) Let $g_1, \dots, g_n \in \mathcal{U}_{\text{grd}}$ be guards and let $g \in \mathcal{U}_{\text{grd}}$ be another guard, then if $g \equiv g_i$ for some $1 \leq i \leq n$:

$$\bigominus_{1 \leq i \leq n} g_i \ominus g \equiv \bigominus_{1 \leq i \leq n} g_i.$$

Example Reduction

Consider the following reinforcement r over the guards from the running example (Figure 5.4), where $\text{truth}(\varrho_{\text{T}})$:

$$r = (((\varrho_{\text{T}} \otimes g_2) \otimes g_3) \otimes g_2) \otimes g_3.$$

Now, consider if r is in its simple form, i.e. contains the minimal amount of reinforcement to represent the same behaviour. To do this, the applications of Lemma 1 and Proposition 1 are

used as seen below:

$$\begin{aligned}
 r &= (((\varrho_{\top} \otimes g_2) \otimes g_3) \otimes g_2) \otimes g_3 \\
 &\text{(neutral element)} \\
 &= ((g_2 \otimes g_3) \otimes g_2) \otimes g_3 \\
 &\text{(commutative)} \\
 &= ((g_3 \otimes g_2) \otimes g_2) \otimes g_3 \\
 &\text{(associative)} \\
 &= (g_3 \otimes (g_2 \otimes g_2)) \otimes g_3 \\
 &\text{(idempotence)} \\
 &= (g_3 \otimes g_2) \otimes g_3 \\
 &\text{(strong idempotence)} \\
 &= g_3 \otimes g_2
 \end{aligned}$$

The above reduction is important to show, as Figure 5.4 may produce infinite reinforcements of guards from system behaviour. But, these infinite reinforcements can always be simplified down to handful of finite reinforcements.

5.5.3 Playing out a model

This section proposes how to abstract away from the representation of a model through the executable play-outs from the model. A single execution of a data-aware model generates two sequences of the same length, one containing activities and another containing guards. However, the sequence of activities may include silent actions, which cannot be encoded into a transition tree. Therefore, the function `playout` is used to proliferate guards from silent activities to non-silent activities. This proliferation creates a play-out trace and to denote proper termination, where a halt symbol is denoted as ‘*halt*’.

Definition 24 (Play-out Traces) Let $a \in \mathcal{U}_{act} \cup \{\tau\}$ be an activity or the silent action, and $tl_a \in (\mathcal{U}_{act} \cup \{\tau\})^*$ be a sequence of activities with silent actions (τ), $g \in \mathcal{U}_{grd}$ be a guard, $tl_g \in \mathcal{U}_{grd}^*$ be a sequence of guards such that $|a: tl_a| = |g: tl_g|$, and $g', \varrho_{\top} \in \mathcal{U}_{grd}$ be guards such that $\text{truth}(\varrho_{\top})$. Then, to compute a play-out trace using these sequences, we use the function `playout` defined as follows:

$$\begin{aligned}
 \text{playout}(\langle \rangle, \langle \rangle, g') &= \langle (\text{halt}, \varrho_{\top}) \rangle \text{ and} \\
 \text{playout}(a: tl_a, g: tl_g, g') &= \begin{cases} (a, g' \otimes g): \text{playout}(tl_a, tl_g, \varrho_{\top}) & \text{if } a \neq \tau, \\ \text{playout}(tl_a, tl_g, g' \otimes g) & \text{otherwise.} \end{cases}
 \end{aligned}$$

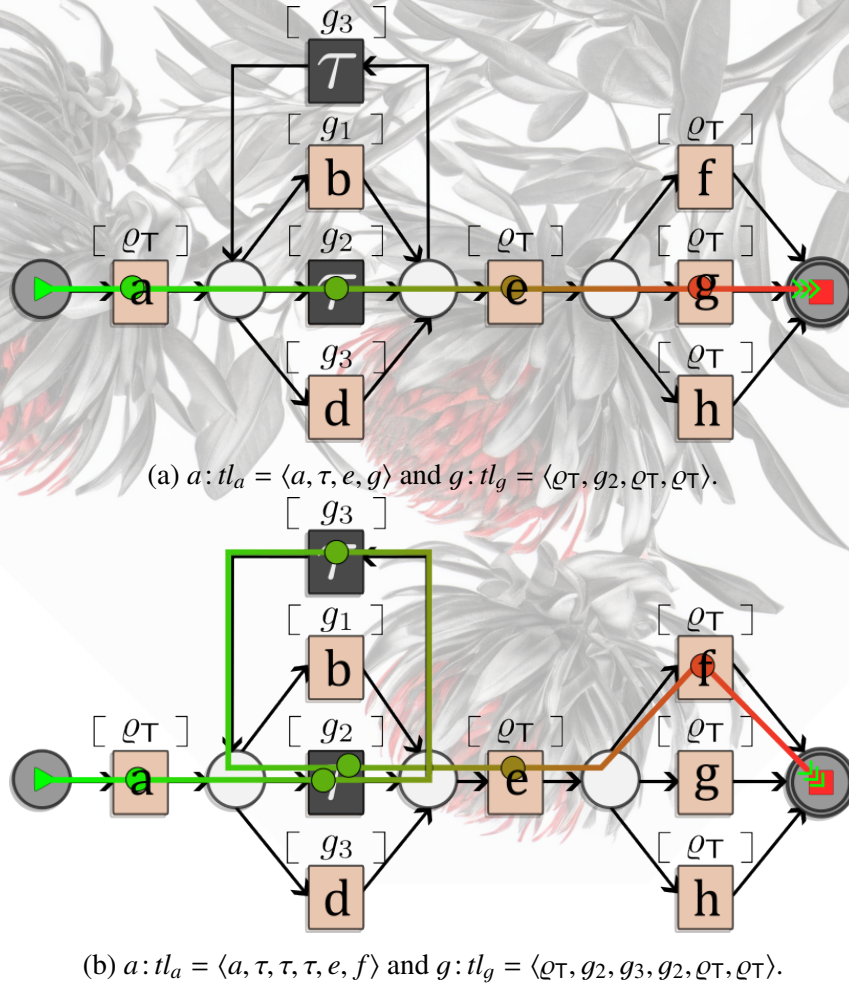


Figure 5.5: Example executions for the running example.

Example Play-out

For example, Figure 5.5 shows two executions of the model and the associated sequences of activities and guards. Now consider how these executions are handled by the playout function. Starting with the execution described in Figure 5.5a, where the applications of Lemma 1 and Proposition 1 are used in the recursion of the playout function below:

$$\begin{aligned}
 \text{playout}(a: \langle \tau, e, g \rangle, \varrho_T: \langle g_2, \varrho_T, \varrho_T \rangle, \varrho_T) &= (a, \varrho_T): \text{playout}(\tau: \langle e, g \rangle, g_2: \langle \varrho_T, \varrho_T \rangle, \varrho_T) \\
 &= (a, \varrho_T): \text{playout}(e: \langle g \rangle, \varrho_T: \langle \varrho_T \rangle, \varrho_T \otimes g_2) \\
 &= (a, \varrho_T): ((e, \varrho_T \otimes g_2): \text{playout}(g: \langle \rangle, \varrho_T: \langle \rangle, \varrho_T)) \\
 &= (a, \varrho_T): ((e, \varrho_T \otimes g_2): ((g, \varrho_T): \text{playout}(\langle \rangle, \langle \rangle, \varrho_T))) \\
 &= (a, \varrho_T): ((e, \varrho_T \otimes g_2): ((g, \varrho_T): (\text{halt}, \varrho_T))) \\
 &= \langle (a, \varrho_T), (e, \varrho_T \otimes g_2), (g, \varrho_T), (\text{halt}, \varrho_T) \rangle
 \end{aligned}$$

In the recursion, some reinforcements were reduced, e.g. $\varrho_T \otimes g_2 \otimes \varrho_T$ to $\varrho_T \otimes g_2$, or $\varrho_T \otimes \varrho_T$ to ϱ_T . The same demonstration is repeated for the execution described in Figure 5.5b and

denoted below:

$$\begin{aligned}
& \text{playout}(a : \langle \tau, \tau, \tau, e, f \rangle, \varrho_T : \langle g_2, g_3, g_2, \varrho_T, \varrho_T \rangle, \varrho_T) \\
&= (a, \varrho_T) : \text{playout}(\tau : \langle \tau, \tau, e, f \rangle, g_2 : \langle g_3, g_2, \varrho_T, \varrho_T \rangle, \varrho_T) \\
&= (a, \varrho_T) : \text{playout}(\tau : \langle \tau, e, f \rangle, g_3 : \langle g_2, \varrho_T, \varrho_T \rangle, \varrho_T \otimes g_2) \\
&= (a, \varrho_T) : \text{playout}(\tau : \langle e, f \rangle, g_2 : \langle \varrho_T, \varrho_T \rangle, \varrho_T \otimes g_2 \otimes g_3) \\
&= (a, \varrho_T) : \text{playout}(e : \langle f \rangle, \varrho_T : \langle \varrho_T \rangle, \varrho_T \otimes g_2 \otimes g_3) \\
&= (a, \varrho_T) : ((e, \varrho_T \otimes g_2 \otimes g_3) : \text{playout}(f : \langle \rangle, \varrho_T : \langle \rangle, \varrho_T)) \\
&= (a, \varrho_T) : ((e, \varrho_T \otimes g_2 \otimes g_3) : ((f, \varrho_T) : (\text{halt}, \varrho_T))) \\
&= \langle (a, \varrho_T), (e, \varrho_T \otimes g_2 \otimes g_3), (f, \varrho_T), (\text{halt}, \varrho_T) \rangle
\end{aligned}$$

5.5.4 Constructing a tree

Introduced next is how to derive a transition tree from the executable play-out traces of a model. Firstly, the notion of a play-out log from a process model and access functions for play-out traces is presented.

Definition 25 (Play-out Logs) Let $\varrho_T \in \mathcal{U}_{grd}$ be a guard such that $\text{truth}(\varrho_T)$, let M be a model, and let $I \subseteq (\mathcal{U}_{act} \cup \{\tau\})^+ \times \mathcal{U}_{grd}^+$ denote the set of possible execution instances generated from the model. Then, the play-out log $L_M \subset (\mathcal{U}_{act} \times \mathcal{U}_{grd})^+$ for M is defined as:

$$L_M = \{ \text{playout}(s_a, s_g, \varrho_T) \mid (s_a, s_g) \in I \}.$$

To denote a play-out trace, ρ is written. Furthermore, the following access functions are used, poact and pogd , for play-out traces to retrieve either the activities or guards within the play-out.

Definition 26 (Accessing Play-out Traces) Let L_M be a play-out log, and let $\rho \in L_M$ be a play-out trace. Then, the function $\text{poact} : L_M \mapsto \mathcal{U}_{act}^+$ yields the variant for ρ , and the function $\text{pogd} : L_M \mapsto \mathcal{U}_{grd}^+$ yields the sequence of guards for ρ . Both of these functions, poact and pogd are defined as:

$$\begin{aligned}
& \text{poact}(\langle \rangle) = \langle \rangle \text{ and } \text{poact}((a, g) : \rho) = a : \text{poact}(\rho), \\
& \text{pogd}(\langle \rangle) = \langle \rangle \text{ and } \text{pogd}((a, g) : \rho) = g : \text{pogd}(\rho).
\end{aligned}$$

Now, deriving a directed acyclic graph (*dag*) from a play-out log is introduced. To clarify, these graphs are always constructed in the context of an event log, as the longest observed trace is used to limit the construction of a possible infinite transition tree. By using the longest observed trace length, the length of the play-out traces used to derive these graphs are also limited to this length.

Definition 27 (Play-out Dags) Let M be a model, let L_M be a play-out log of that model, let k be an integer for the longest length of observed traces, then a *dag* from the play-out log,

$\text{dag}_{L_M} = (N, F, n_r, N_f, \text{src}, \text{tgt}, \text{gd}, \text{actf})$, is defined as:

$$n_r = \langle \rangle, \quad (5.1)$$

$$N = \{\text{poact}(\rho)[1:i] \mid \rho \in L_M \wedge 0 < i < \min(k+1, |\rho|)\} \cup \{n_r\}, \quad (5.2)$$

$$N_f = \{\text{poact}(\rho)[1:i] \mid \rho \in L_M \wedge 0 \leq i < \min(k+1, |\rho|) \wedge \text{poact}(\rho)_{i+1} = \text{halt}\}, \quad (5.3)$$

$$F = \{(\text{poact}(\rho)[1:i], (\text{poact}(\rho)_{i+1}, \text{pogd}(\rho)_{i+1}), \text{poact}(\rho)[1:i+1]) \mid \rho \in L_M \wedge 0 \leq i < \min(k+1, |\rho| - 1)\}. \quad (5.4)$$

However, this definition does have a drawback, it does always produce a directed acyclic graph but not always a tree. If the play-out log comes from a process model with silent actions or duplicated activities, the approach may define more than one flow between two nodes in the graph. Having multiple flows between the same two nodes is problematic for the discussions of precision as quantification needs to consider if a path of the graph only had guards evaluated to be true for the taken path. Hence, a flow reduction step is introduced to ensure only one flow exists between nodes, which when combined with Definition 27 always reduces a play-out dag to a play-out tree.

Definition 28 (Flow Reduction) Let $\text{dag}_{L_M} = (N, F, n_r, N_f, \text{src}, \text{tgt}, \text{gd}, \text{actf})$ be a play-out dag, let $A = \{\text{actf}(f) \mid f \in F\}$ be the set of activity labels, and $a \in A$ be a label. Then, $\text{tree}_{L_M} = (N, F', n_r, N_f, \text{src}, \text{tgt}, \text{gd}, \text{actf})$ is the play-out tree of dag_{L_M} where F is replaced with F' , such that only one flow exists in F' between two nodes $n, n': a \in A$, which already had at least one flow between them in F , defined as:

$$F' = \{(n, (a, \bigoplus_{f \in F, n \xrightarrow{f} n'} \text{gd}(f)), n') \mid n \in N \wedge n \rightarrow n': a\}.$$

For the rest of this chapter, a *play-out tree* is referring to a play-out dag constructed from Definition 27 that has had flow reduction (Definition 28) applied to its flows.

Example Tree

Continuing with the running example, consider a play-out log for Figure 5.4 and constructing a play-out tree. For the construction of the tree, assume that the longest trace length is four, i.e. for Definition 27, assume $k = 4$.

A partial derivation of the play-log L_M for Figure 5.4 is shown below, with the caveat that

only the play-out traces that are shorter than six elements are needed:

$$\begin{aligned}
 L_M = \{ & \langle (a, g_T), (b, g_1), (e, g_T), (f, g_T), (halt, \varrho_T) \rangle, \\
 & \langle (a, g_T), (b, g_1), (e, g_T), (g, g_T), (halt, \varrho_T) \rangle, \\
 & \langle (a, g_T), (b, g_1), (e, g_T), (h, g_T), (halt, \varrho_T) \rangle, \\
 & \langle (a, g_T), (e, g_2 \otimes g_T), (f, g_T), (halt, \varrho_T) \rangle, \\
 & \langle (a, g_T), (e, g_2 \otimes g_T), (g, g_T), (halt, \varrho_T) \rangle, \\
 & \langle (a, g_T), (e, g_2 \otimes g_T), (h, g_T), (halt, \varrho_T) \rangle, \\
 & \langle (a, g_T), (d, g_3), (e, g_T), (f, g_T), (halt, \varrho_T) \rangle, \\
 & \langle (a, g_T), (d, g_3), (e, g_T), (g, g_T), (halt, \varrho_T) \rangle, \\
 & \langle (a, g_T), (d, g_3), (e, g_T), (h, g_T), (halt, \varrho_T) \rangle \\
 & \langle (a, g_T), (b, g_1), (b, g_3 \otimes g_1), (e, g_T), (f, g_T) \rangle, \\
 & \langle (a, g_T), (b, g_1), (b, g_3 \otimes g_1), (e, g_T), (g, g_T) \rangle, \\
 & \langle (a, g_T), (b, g_1), (b, g_3 \otimes g_1), (e, g_T), (h, g_T) \rangle, \\
 & \langle (a, g_T), (d, g_1), (b, g_3 \otimes g_1), (e, g_T), (f, g_T) \rangle, \\
 & \langle (a, g_T), (d, g_1), (b, g_3 \otimes g_1), (e, g_T), (g, g_T) \rangle, \\
 & \langle (a, g_T), (d, g_1), (b, g_3 \otimes g_1), (e, g_T), (h, g_T) \rangle, \dots \}.
 \end{aligned}$$

Next, the derivation of Definition 27 and the resulting directed graph using the play-out log is shown in Figure 5.6. This demonstration shows that the flow reduction step is required to ensure tree is constructed, as Figure 5.6h clearly shows that many flows can exist between two nodes. The comparison between the resulting directed graph produced by Definition 27, and the flow reduced tree using Definition 28 is shown in Figure 5.7. In this comparison, the flows which have been reduced into a single flow are highlighted in red.

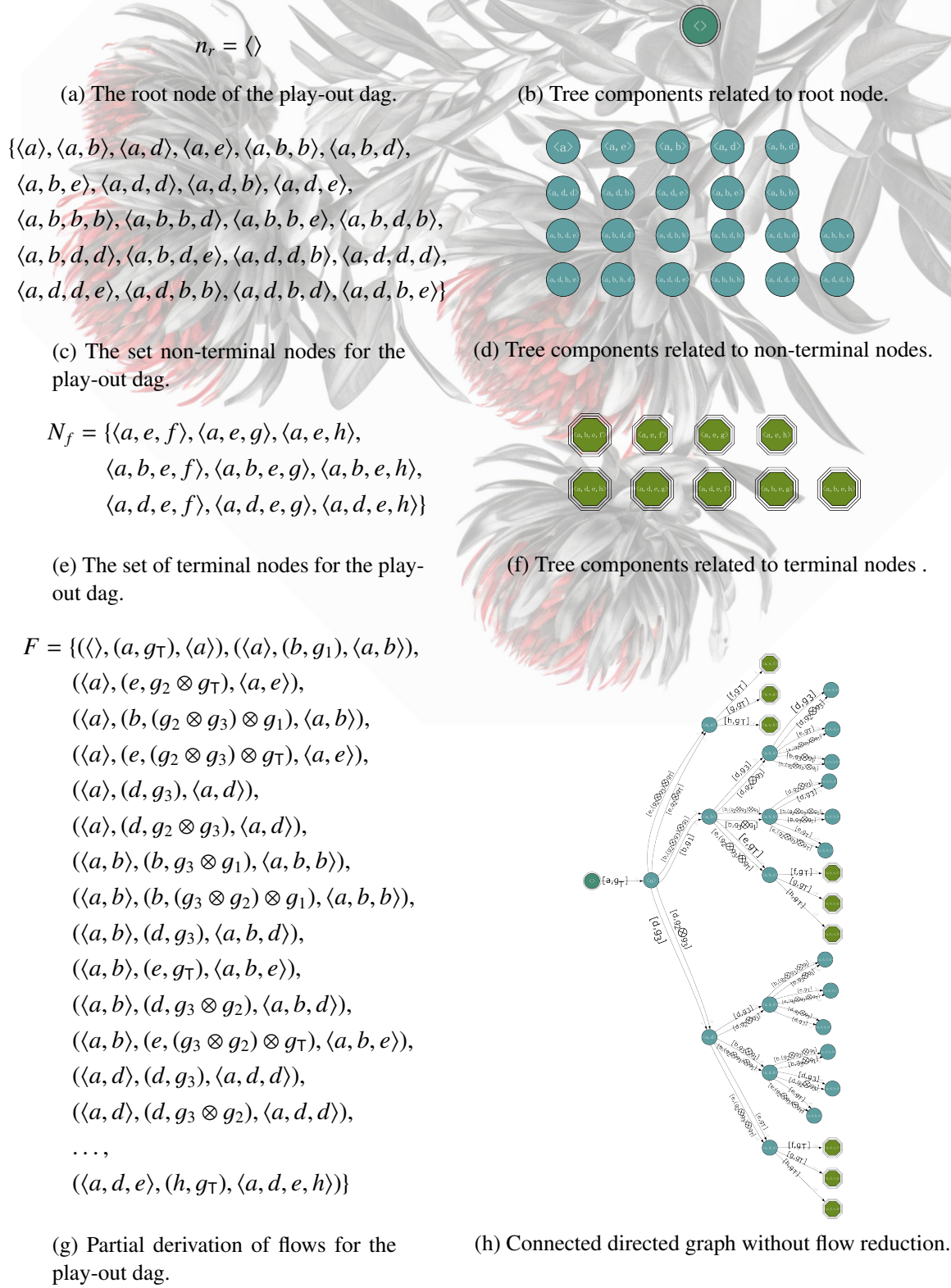


Figure 5.6: A step-wise introduction to the derivation of a play-out tree from a set of play-out traces.

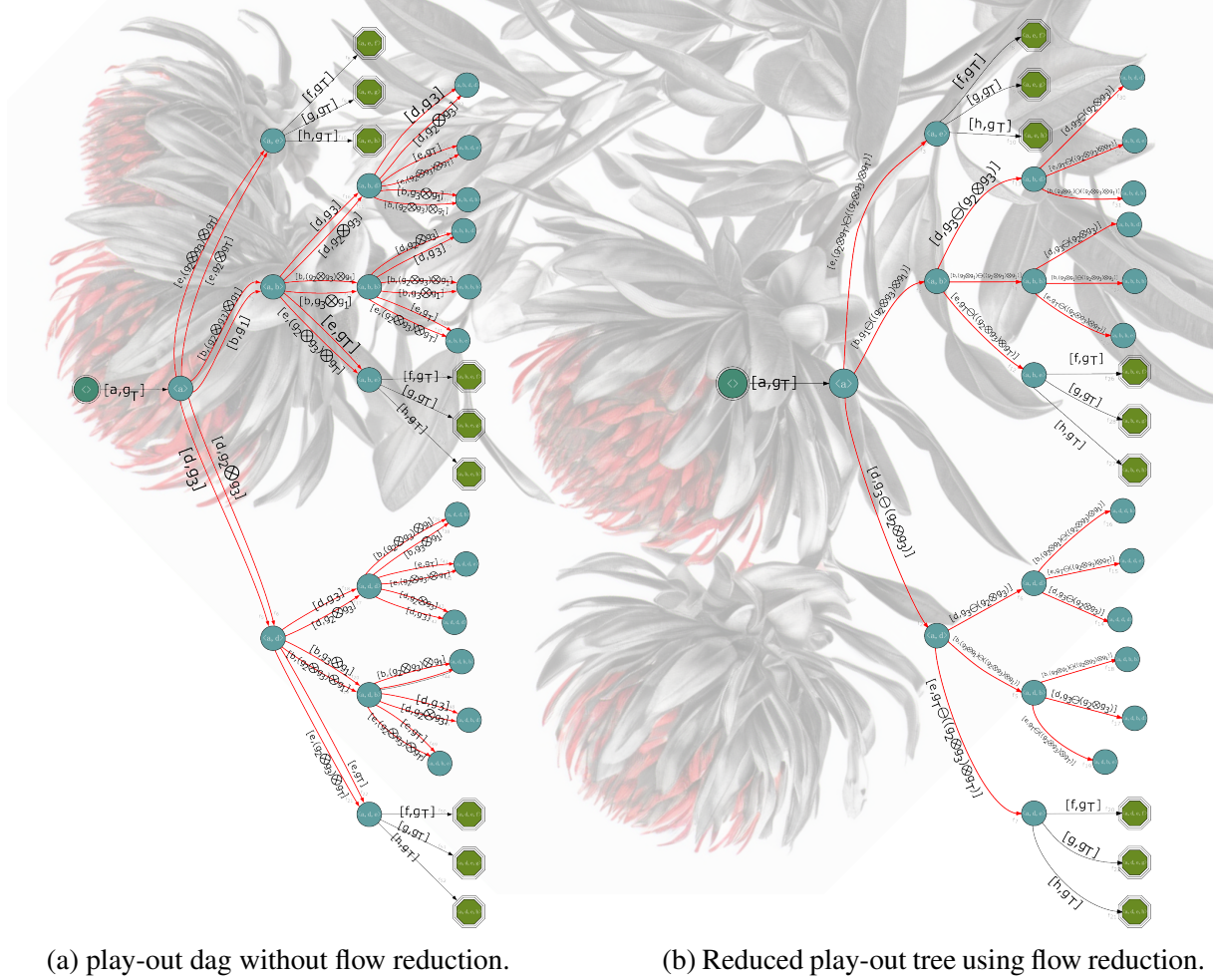


Figure 5.7: Comparison of outcomes after flow reduction has occurred. Changed flows are highlighted in red.

5.5.5 Finding a Matching

This section introduces the notion of a *matching* between variants of a log and paths in a model, i.e. a control-flow focused alignment which only consists of synchronised and log moves (as per the problematisation in Section 5.3).

Paths. For a particular process model M , the set of paths-with-skips through this model is denoted as P^M . P will be used if the context is clear. Each $p \in P$ takes a particular form: a sequence of flows potentially alternated with skips, i.e. $P \subseteq (F \cup \{\gg\})^*$; with $\gg$ denoting the skip symbol, a recognition that paths cannot always follow a particular step in the model. Note that paths in M do not have skips (treating M as a graph), but paths through M may have them. The function `noskip` removes all skips from a path and is defined as follows:

$$\text{noskip}(\langle \rangle) = \langle \rangle \quad \text{and} \quad \text{noskip}(s:p) = \begin{cases} s:\text{noskip}(p) & \text{if } s \neq \gg, \\ \text{noskip}(p) & \text{otherwise.} \end{cases}$$

Matching. A matching function between a log and a model captures how variants in the log

are mapped onto paths in the model. This mapping may not be perfect, i.e. it may imply that guards are violated, process activities may not match between events and flows, or some paths may not end in final nodes. Note, that at this stage no restrictions on the matching have been placed, such as being “optimal”. Furthermore, a matching is a generalisation of alignments [81], which only considers log models (i.e. $\gg$) or sync moves (i.e. the path denotes a flow) which may or may not be ideal.

Definition 29 (Matchings) *Let L be a log, let $\mathcal{V}_L$ be the set of variants captured in L , let $M = (N, F, n_r, N_f, \text{src}, \text{tgt}, \text{gd}, \text{actf})$ be a model, and let $P \subset (F \cup \{\gg\})^*$ be the set of paths through M . Then, a matching function $\gamma: \mathcal{V}_L \mapsto P$ maps control-flow variants to paths in the model. For any variant $\hat{\sigma} \in \mathcal{V}_L$ and its reduced path $p = \text{noskip}(\gamma(\hat{\sigma}))$ the following holds:*

1. *the reduced path always starts at the root: if $|p| > 0$ then $n_r \xrightarrow{p_1} \text{tgt}(p_1)$;*
2. *the reduced path is between connected flows: for all $1 \leq i < |p|$: $\text{src}(p_i) \xrightarrow{p_i} \text{src}(p_{i+1})$;*
3. *and the length of the path is never longer than the variant: $|\gamma(\hat{\sigma})| \leq |\hat{\sigma}|$.*

One to Many Matchings

Now the process of finding a suitable collection of candidate paths to use as a matching function is described, where the goal is to find “optimal” paths through the model for a given variant. To find these paths one might use an alignment procedure to efficiently find an optimal path for a variant, but this section abstracts from the method used. Instead, this section focuses on describing a ‘many matching’, which produces all paths that are least costly or optimal in the best case. The difference between a matching and a many matching, is that the former is a one-to-one while the latter is a one-to-many relationship.

Costing. To consider if a path is optimal for a variant, the quality of this pairing is considered. The quality of the pairing is impacted by differences in length, non-matching activities at corresponding positions, and improper termination. The function `cost` is used to determine this quality, where `cost` yields 0 to denote an optimal pairing and anything else denotes a less-than-optimal pairing.

Definition 30 (Cost of a Path) *Let $M = (N, F, n_r, N_f, \text{src}, \text{tgt}, \text{gd}, \text{actf})$ be a play-out tree, let $p \in P_M$ be path through this tree, and let $\hat{\sigma} \in \mathcal{U}_{act}^+$ be a variant. Then, `cost` is a function that yields a non-zero integer for a non-optimal path or zero for an optimal path:*

$$\text{cost}(p, \hat{\sigma}) = (|\hat{\sigma}| - |\text{noskip}(p)|) + \sum_{1 \leq i \leq |p|} [\text{actf}(p_i) \neq \hat{\sigma}_i] + [\text{tgt}(\text{last}(\text{noskip}(p))) \notin N_f].$$

Now, the process for selecting candidate paths is denoted, which prioritises the optimal paths over non-optimal paths and results in a many matching Γ . Furthermore, in-line with Definition 29 of a matching, only considers paths through the model of at most the length of the variant.

To denote this process, the function Γ is introduced, which takes a variant $\hat{\sigma}$, and yields the set of the least costly paths through a model for a variant.

Definition 31 (Many Matching) *Let $M = (N, F, n_r, N_f, \text{src}, \text{tgt}, \text{gd}, \text{actf})$ be a play-out tree, and let P be the set of all paths through and in M , and let $\hat{\sigma} \in \mathcal{U}_{\text{act}}^+$ be a variant, then we define:*

$$\Gamma(i, \hat{\sigma}) = \{s \in P \mid \text{cost}(s, \hat{\sigma}) = i \wedge |s| \leq |\hat{\sigma}|\}.$$

Now $\Gamma(\hat{\sigma}) = \Gamma(i, \hat{\sigma})$ for the minimal $i \in \mathbb{N}$ for which $\Gamma(i, \hat{\sigma}) \neq \emptyset$.

Weighing. At times, the quantification will need to be able to determine the likelihood of a path in the context of a variant. As such a weighting function ω is introduced, which in the context of a set of paths T and a given path p and a variant $\hat{\sigma}$, yields the likelihood that the path is the true system path for the variant. Thus, if the given set of paths only contains optimal paths, ω acts as a full-distribution, i.e. $\sum_{p \in T} \omega(\hat{\sigma}, p) = 1$, otherwise ω acts as sub-distribution, i.e. $\sum_{p \in T} \omega(\hat{\sigma}, p) < 1$.

Definition 32 (Weighting Paths) *Let L be an event log, let $\hat{\sigma} \in \mathcal{V}_L$ be a variant, let M be a play-out tree, let $T = \Gamma(\hat{\sigma})$ be the least costly paths of the variant through the model, let $p \in T$ be one such path, and let $\kappa \in \mathbb{R}$ denote a small cost-offset such that $0 < \kappa < 1$. Then, $\omega : \mathcal{V}_L \times T \mapsto \mathbb{R}$ is a function which takes a variant and a path and yields the quality of this pairing, defined as:*

$$\omega(\hat{\sigma}, p) = \frac{1}{|T|} \cdot \kappa^{\text{cost}(p, \hat{\sigma})}.$$

For the rest of the chapter, the contributions will assume κ to be 0.9. This selection of κ means that for short paths with minor issues, they have a weight near the best score. While, for longer paths with issues these have a smooth transition towards zero as they have more and more issues.

Example Matching

Now, a demonstration for the construction of the set of likely candidates paths for a single trace when considering Figure 5.7b is shown. Consider the following trace, $\sigma = \langle e_1^a, e_2^b, e_3^z \rangle$, and the following partial derivation of P_M which considers paths to nodes and paths to nodes with one skip:

Assuming the following flows from Figure 5.7b:

$$\begin{aligned} f_1 &= (\langle \rangle, (a, g_\top), \langle a \rangle), & f_2 &= (\langle a \rangle, (d, (g_3) \ominus (g_2 \otimes g_3)), \langle a, d \rangle), \\ f_3 &= (\langle a \rangle, (e, (g_2 \otimes g_3) \ominus ((g_2 \otimes g_3) \otimes g_\top)), \langle a, e \rangle), & f_4 &= (\langle a \rangle, (b, (g_1) \ominus ((g_2 \otimes g_3) \otimes g_1)), \langle a, b \rangle), \\ f_5 &= (\langle a, d \rangle, (b, (g_3 \otimes g_1) \ominus ((g_2 \otimes g_3) \otimes g_1)), \langle a, d, b \rangle), & f_6 &= (\langle a, d \rangle, (d, g_3 \ominus (g_2 \otimes g_3)), \langle a, d, d \rangle), \\ f_7 &= (\langle a, d \rangle, (e, g_\top \ominus ((g_2 \otimes g_3) \otimes g_\top)), \langle a, d, e \rangle), & f_8 &= (\langle a, e \rangle, (f, g_\top), \langle a, e, f \rangle), \\ f_9 &= (\langle a, e \rangle, (g, g_\top), \langle a, e, g \rangle), & f_{10} &= (\langle a, e \rangle, (h, g_\top), \langle a, e, h \rangle), \end{aligned}$$

$$f_{11} = (\langle a, b \rangle, (b, (g_3 \otimes g_1) \ominus ((g_2 \otimes g_3) \otimes g_1)), \langle a, b, h \rangle),$$

$$f_{12} = (\langle a, b \rangle, (e, g_\top \ominus ((g_2 \otimes g_3) \otimes g_\top)), \langle a, b, e \rangle),$$

$$f_{13} = (\langle a, b \rangle, (d, g_3 \ominus (g_2 \otimes g_3)), \langle a, b, d \rangle).$$

Then, consider a partial derivation of P_M :

$$P_M = \{ \langle f_1, f_2, f_5 \rangle, \langle f_1, f_2, f_6 \rangle, \langle f_1, f_2, f_7 \rangle, \langle f_1, f_3, f_8 \rangle, \langle f_1, f_3, f_9 \rangle, \langle f_1, f_3, f_{10} \rangle, \langle f_1, f_4, f_{11} \rangle, \\ \langle f_1, f_4, f_{12} \rangle, \langle f_1, f_4, f_{13} \rangle, \langle \gg \rangle, \langle \gg, f_1 \rangle, \langle f_1, \gg \rangle, \langle \gg, f_1, f_2 \rangle, \langle f_1, \gg, f_2 \rangle, \langle f_1, f_2, \gg \rangle, \\ \langle \gg, f_1, f_3 \rangle, \langle f_1, \gg, f_3 \rangle, \langle f_1, f_3, \gg \rangle, \langle \gg, f_1, f_4 \rangle, \langle f_1, \gg, f_4 \rangle, \langle f_1, f_4, \gg \rangle, \dots \}.$$

For the given σ , observe that no optimal path exists as there is no flow labeled with ‘z’ (f_{1-13}). Moreover, any sequence in P_M will occur a cost from either not being the same length $\|\sigma\| - \|\text{noskip}(p)\|$, or from improper termination $\text{tgt}(\text{last}(\text{noskip}(p))) \notin N_f$. Next, working up until Γ yields a non-empty set is shown, with the cost of a candidate displayed above it in red, below:

Applying cost to P_M the following costing occurs:

$$P_M = \left\{ \begin{array}{l} \overset{0+2+1}{\langle f_1, f_2, f_5 \rangle}, \overset{0+2+1}{\langle f_1, f_2, f_6 \rangle}, \overset{0+2+1}{\langle f_1, f_2, f_7 \rangle}, \overset{0+2+0}{\langle f_1, f_3, f_8 \rangle}, \overset{0+2+0}{\langle f_1, f_3, f_9 \rangle}, \overset{0+2+0}{\langle f_1, f_3, f_{10} \rangle}, \overset{0+2+0}{\langle f_1, f_3, f_{11} \rangle}, \\ \overset{0+1+1}{\langle f_1, f_4, f_{11} \rangle}, \overset{0+1+1}{\langle f_1, f_4, f_{12} \rangle}, \overset{0+1+1}{\langle f_1, f_4, f_{13} \rangle}, \overset{3+1+1}{\langle \gg \rangle}, \overset{2+2+1}{\langle \gg, f_1 \rangle}, \overset{2+1+1}{\langle f_1, \gg \rangle}, \overset{1+3+1}{\langle \gg, f_1, f_2 \rangle}, \overset{1+2+1}{\langle f_1, \gg, f_2 \rangle}, \\ \overset{1+2+1}{\langle f_1, f_2, \gg \rangle}, \overset{1+3+1}{\langle \gg, f_1, f_3 \rangle}, \overset{1+2+1}{\langle f_1, \gg, f_3 \rangle}, \overset{1+2+1}{\langle f_1, f_3, \gg \rangle}, \overset{1+3+1}{\langle \gg, f_1, f_4 \rangle}, \overset{1+2+1}{\langle f_1, \gg, f_4 \rangle}, \overset{1+1+1}{\langle f_1, f_4, \gg \rangle}, \\ \overset{1+0+1}{\langle f_1, f_4 \rangle}, \dots \end{array} \right\}.$$

Then, Γ is incrementally applied to find the minimal i :

$$\Gamma(\sigma, 0) = \emptyset, \quad \Gamma(\sigma, 1) = \emptyset, \quad \Gamma(\sigma, 2) = \{ \langle f_1, f_3, f_8 \rangle, \langle f_1, f_3, f_9 \rangle, \langle f_1, f_3, f_{10} \rangle, \langle f_1, f_4, f_{11} \rangle, \\ \langle f_1, f_4, f_{12} \rangle, \langle f_1, f_4, f_{13} \rangle, \langle f_1, f_4 \rangle \}.$$

Thus for the given σ , a many-matching $\Gamma(\sigma)$ is constructed that returns a set of seven non-optimal paths. Where each candidate path has a weighted share, computed by ω , of 0.115 out of a maximal share of 0.142 if they were optimal paths.

5.5.6 Diagnosing a Model

This section introduces how to quantify the quality of guards, and how to count their evaluations using ‘bookkeeping’ functions.

Bookkeeping. The notion of recall, in terms of data-aware conformance, considers how often a guard evaluated to a particular outcome when an associated flow is presented in the matching. Thus, the following functions capture information about guard evaluations given a flow and an evaluation outcome, referred to as *traversal* bookkeeping functions. In the following definitions, the Iverson bracket is used: $[\phi] = 1$ if ϕ evaluates to *true* and $[\phi] = 0$ if ϕ evaluates to *false*.

Definition 33 (Traversal Bookkeeping) Let L be a log, let $M = (N, F, n_r, N_f, \text{src}, \text{tgt}, \text{gd}, \text{actf})$ be a model, let γ be a matching between them, and let $f \in F$ be a flow. Then, $\text{Grd}_T^\rightarrow, \text{Grd}_F^\rightarrow, \text{Grd}_U^\rightarrow$ are functions that compute the traversal bookkeeping for a flow, for a particular outcome of a guard, defined as:

$$\text{Grd}_T^\rightarrow(f, \gamma, L) = \sum_{\sigma \in L} [\exists_{1 \leq i < |\sigma|} [\gamma_{\hat{\sigma}, i} = f \wedge \llbracket \text{gd}(f) \rrbracket(\mu_{\sigma, i-1})]], \quad (5.5)$$

$$\text{Grd}_F^\rightarrow(f, \gamma, L) = \sum_{\sigma \in L} [\exists_{1 \leq i < |\sigma|} [\gamma_{\hat{\sigma}, i} = f \wedge \neg \llbracket \text{gd}(f) \rrbracket(\mu_{\sigma, i-1})]], \quad (5.6)$$

$$\text{Grd}_U^\rightarrow(f, \gamma, L) = \sum_{\sigma \in L} [\exists_{1 \leq i < |\sigma|} [\gamma_{\hat{\sigma}, i} = f \wedge \llbracket \text{gd}(f) \rrbracket(\mu_{\sigma, i-1}) = \text{U}]]. \quad (5.7)$$

Next, the notion of precision considers how restricted were choices of the system given the matching and how many times a guard could have been evaluated to be true. Thus, *exploratory* bookkeeping functions are introduced to capture information about guard evaluations that could have been evaluated for a given flow, i.e. all paths (and their associated assignments) that visited the source node for the given flow are considered.

Definition 34 (Exploratory Bookkeeping) Let L be a log, let $M = (N, F, n_r, N_f, \text{src}, \text{tgt}, \text{gd}, \text{actf})$ be a model, let γ be a matching between them, and let $f \in F$ be a flow. Then, $\text{Grd}_T, \text{Grd}_F, \text{Grd}_U$ are functions that compute the exploratory bookkeeping for a flow, for a particular outcome of a guard, defined as:

$$\text{Grd}_T(f, \gamma, L) = \sum_{\sigma \in L} [\exists_{1 \leq i < |\sigma|} [\text{src}(\gamma_{\hat{\sigma}, i}) = \text{src}(f) \wedge \llbracket \text{gd}(f) \rrbracket(\mu_{\sigma, i-1})]], \quad (5.8)$$

$$\text{Grd}_F(f, \gamma, L) = \sum_{\sigma \in L} [\exists_{1 \leq i < |\sigma|} [\text{src}(\gamma_{\hat{\sigma}, i}) = \text{src}(f) \wedge \neg \llbracket \text{gd}(f) \rrbracket(\mu_{\sigma, i-1})]], \quad (5.9)$$

$$\text{Grd}_U(f, \gamma, L) = \sum_{\sigma \in L} [\exists_{1 \leq i < |\sigma|} [\text{src}(\gamma_{\hat{\sigma}, i}) = \text{src}(f) \wedge \llbracket \text{gd}(f) \rrbracket(\mu_{\sigma, i-1}) = \text{U}]]. \quad (5.10)$$

Now using the bookkeeping functions, guard equivalence can be determined to understand if two flows from separate models have equivalent bookkeeping.

Definition 35 (Relaxed Guard Equivalence) In the context of model M (with flow f) and M' (with flow f') and with respect to log L with matchings γ (between L and M) and γ' (between L and M'), flows f and f' are relaxed guard equivalent, $\text{GrdEq}(f, f', \gamma, \gamma')$, iff:

$$\text{Grd}_T(f, \gamma, L) = \text{Grd}_T(f', \gamma', L) \wedge \text{Grd}_F(f, \gamma, L) = \text{Grd}_F(f', \gamma', L) \wedge \text{Grd}_U(f, \gamma, L) = \text{Grd}_U(f', \gamma', L).$$

In the case that an isomorphism exists between the models, the following corollary exists.

Corollary 35.1 Guard equivalence implies relaxed guard equivalence. Let L be a log, let $M = (N, F, n_r, N_f, \dots)$ and M' be models such that an isomorphism iso exists between these models, let γ be a matching between L and M , let γ' be its isomorphic equivalent between L and M' , and

let $f \in F$ be a flow.

$$\text{gd}(f) \equiv \text{gd}(\text{iso}(f)) \implies \text{GrdEq}(f, \text{iso}(f), \gamma, \gamma').$$

Agreement. While not being the direct goal of data-aware conformance checking, it is important to understand the agreement between a log and a model. To this end, the quantification needs to know whether events were skipped, whether process activities always matched, and whether traces always ended properly.

Definition 36 (Optimal Matchings) *Let L be a log, let M be a model, and let γ be a matching between them. When there are no skips, no activity mismatches, and all traces end in terminal nodes, then the matching is considered optimal:*

$$\text{optimal}(L, \gamma, M) \equiv \sum_{\sigma \in L} \text{cost}(\gamma(\hat{\sigma}), \hat{\sigma}) = 0.$$

5.6 Quality Axioms

This section characterises the notions of *guard-recall* and *guard-precision* (grec and gprec). These notions help us position recall and precision in the context of data in a way similar to how recall and precision were positioned in a control-flow context in earlier work [42], [67], [68]. Guard-recall and guard-precision are functions that require a log, a matching, and a model to quantify a value between min and max (inclusive). While the minimal value can be zero and the maximum can be one, this work does not require approaches to adhere to this notion, as some metrics take on values greater than one [233]. Along this line of thought, several axioms that guard-recall and guard-precision should at least satisfy are formulated.

Representation. Following Proposition 1 presented by Syring, Tax, and van der Aalst [42], guard-recall and guard-precision are required to be deterministic functions, i.e. their outcome is fully determined by their input: the log, the model, and the matching (Axiom 6 and Axiom 9, where $k = 1$). Proposition 2 of [42] requires independence from process representation. However, this work has chosen transition trees as an abstraction of process models, so the approach taken does not fully abide by this proposition. However, regarding the data perspective, the approach does not prescribe how guards should be formulated and abides by the proposition (Axiom 1 and Axiom 2).

Axiom 1 (Inspired by Proposition 2 in [42]) *Let L be a log, let $M = (N, F, n_r, N_f, \dots)$ be a model, let M' be another model using a different guard representation such that an isomorphism iso exists between these models, let γ be a matching between L and M , and let γ' be its isomorphic equivalent between L and M' . Then, grec is fully determined by the behaviour observed in the log and the behaviour of guards (representation of guards does not matter):*

$$\forall_{f \in F} \text{GrdEq}(f, \text{iso}(f), \gamma, \gamma') \implies \text{grec}(L, \gamma, M) = \text{grec}(L, \gamma', M').$$

Axiom 2 (Inspired by Proposition 2 in [42]) Let L be a log, let $M = (N, F, n_r, N_f, \dots)$ be a model, let M' be another model using a different guard representation such that an isomorphism iso exists between these models, let γ be a matching between L and M , and let γ' be its isomorphic equivalent between L and M' . Then, gprec is fully determined by the behaviour observed in the log and the behaviour of guards (representation of guards does not matter):

$$\forall_{f \in F} \text{GrdEq}(f, \text{iso}(f), \gamma, \gamma') \implies \text{gprec}(L, \gamma, M) = \text{gprec}(L, \gamma', M').$$

Guard-Recall. Ideally, flows with guards that do not evaluate to true are not taken. Guard-recall quantifies if guards have been formed over the observed data and if they evaluate to true when taken. Hence, when comparing similar models with the same event log, guard-recall takes on a larger value for the model with more guard evaluations that were true (Axiom 3). This Axiom implies that if a guard evaluates to false or couldn't be evaluated, then guard-recall is negatively affected.

Axiom 3 Let L be a log, let $M = (N, F, n_r, N_f, \dots)$ be a model, let M' be another model with iso an isomorphism between these models, let γ be a matching between L and M and let γ' be its isomorphic equivalent between L and M' . Guard-recall takes on a larger value for the model with more guard evaluations that were true:

$$\sum_{f \in F} (\text{Grd}_{\top}^{\rightarrow}(f, \gamma, L) - \text{Grd}_{\top}^{\rightarrow}(\text{iso}(f), \gamma', L)) < 0 \implies \text{grec}(L, \gamma, M) < \text{grec}(L, \gamma', M').$$

Furthermore, if no guard evaluates to true then guard-recall takes on the minimal value (Axiom 4).

Axiom 4 Let L be a log, let $M = (N, F, n_r, N_f, \dots)$ be a model, and let γ be a matching between them. Guard-recall takes on the minimal value if and only if, no guard was ever evaluated to true when taken:

$$\forall_{f \in F} [\text{Grd}_{\top}^{\rightarrow}(f, \gamma, L) = 0] \iff \text{grec}(L, \gamma, M) = \min.$$

Next, it could be the case, that the given matching is not optimal meaning that some events in the log may not have been considered in the bookkeeping functions. Hence, using a non-optimal matching, opposed to an optimal matching, even when evaluations are similar, will negatively affect guard-recall (Axiom 5).

Axiom 5 Let L be a log, let M be a model, and let γ, γ' be different matching functions between log and model. Guard-recall takes on a larger value for the matching that is optimal, even if evaluations are similar:

$$\begin{aligned} \left(\sum_{f \in M} \text{Grd}_{\top}^{\rightarrow}(f, \gamma, L) \leq \sum_{f \in M} \text{Grd}_{\top}^{\rightarrow}(f, \gamma', L) \wedge \neg \text{optimal}(L, \gamma, M) \wedge \text{optimal}(L, \gamma', M) \right) \\ \implies \text{grec}(L, \gamma, M) < \text{grec}(L, \gamma', M). \end{aligned}$$

This chapter argues that enlarging an event log should not influence guard-recall (Axiom 6).

Axiom 6 (Proposition 6 in [42]) *Let L be a log, let $M = (N, F, n_r, N_f, \dots)$ be a model, let γ be a matching between L or L^k and M , and let $k \geq 1$ be an integer factor. Guard-recall cannot take on a different value if a log is enlarged:*

$$\text{grec}(L, \gamma, M) = \text{grec}(L^k, \gamma, M).$$

Guard-Precision. Ideally, flows with guards only evaluate to true when they are observed to be taken. Hence, guard-precision quantifies if guards only evaluated to true when the associated flow is observed and taken. When comparing similar models with the same event log, guard-precision takes on a larger value for the model with fewer guard evaluations that were true (Axiom 7).

Axiom 7 *Let L be a log, let $M = (N, F, n_r, N_f, \dots)$ be a model, let M' be another model with iso an isomorphism between these models, let γ be a matching between L and M and let γ' be its isomorphic equivalent matching between L and M' . Guard-precision takes on a larger value for the model with fewer guard evaluations that were true when the flow was not taken:*

$$\sum_{f \in F} (\text{Grd}_T(f, \gamma, L) - \text{Grd}_T(\text{iso}(f), \gamma', L)) > 0 \implies \text{gprec}(L, \gamma, M) < \text{gprec}(L, \gamma', M').$$

Next, guard-precision takes on the maximal value if and only if all guards only evaluated to true when they are observed and taken, and no control-flow issues exist in the matching function (Axiom 8).

Axiom 8 *Let L be a log, let $M = (N, F, n_r, N_f, \dots)$ be a model, and let γ be a matching between them. Guard-precision takes on the maximal value if and only if, the matching is optimal and guards only evaluated to true when taken:*

$$(\text{optimal}(L, \gamma, M) \wedge \forall_{f \in F} [\text{Grd}_T^{\rightarrow}(f, \gamma, L) = \text{Grd}_T(f, \gamma, L)]) \iff \text{gprec}(L, \gamma, M) = \max.$$

Furthermore, this chapter argues that enlarging an event log should not influence guard-precision (Axiom 9).

Axiom 9 (Proposition 11 in [42]) *Let L be a log, let $M = (N, F, n_r, N_f, \dots)$ be a model, let γ be a matching between L or L^k and M , and let $k \geq 1$ be an integer factor. Guard-precision cannot take on a different value if a log is enlarged:*

$$\text{gprec}(L, \gamma, M) = \text{gprec}(L^k, \gamma, M).$$

However, it should be noted that many things are implied by Axiom 8 beyond what was previously stated. Firstly, the following always holds for any flow f not part of a split in the model:

$$f \notin \text{splits}(M) \implies \text{Grd}_T^{\rightarrow}(f, \gamma, L) = \text{Grd}_T(f, \gamma, L).$$

Hence, a model without a split may have perfect guard-precision as the function can only be affected by splits. Also implied by Axiom 8, is that guard-precision does not penalise a model-log combination with unobserved behaviour and unobserved guards, as the following holds for any flow $f \in F$:

$$\nexists_{\sigma \in L, 1 \leq i \leq |\sigma|} [\gamma_{\hat{\sigma}, i} = f] \implies \text{Grd}_{\top}^{\rightarrow}(f, \gamma, L) = \text{Grd}_{\top}(f, \gamma, L) = 0.$$

Hence, a model with unobserved flows for a given log may have perfect guard-precision. To quantify if the control-flow is imprecise, one should not use guard-precision as control-flow imprecision is outside the scope of guard-precision.

These axioms are presented as the minimal set to consider data-aware quantification and this chapter does not claim them to account for all cases. For example, future work may consider if data-aware conformance checking should quantify models that are not relaxed data sound at all.

5.7 Proposed Measures

This section formally introduces a guard-recall measure and a guard-precision measure according to the extended theory in Section 5.5.

5.7.1 Guard-Recall

Firstly, the proposed guard-recall function grec is introduced. Guard-recall quantifies if guards have been formed over observed data attributes and if they evaluate to true when taken. Ideally, guard-recall differentiates between data-aware models that have guards that evaluate to false or undefined more often than others with respect to an event log. The proposed formulation uses bookkeeping functions as the axioms do, but allows for one-to-many matchings (see Definition 31). We start by introducing a weighted traversal bookkeeping function, which uses one-to-many matchings to count guard evaluations that were true.

Definition 37 (Weighted Traversal Bookkeeping) *Let L be a log, let M be a model, let L_M be the play-out log of M (Definition 25), let $M' = (N, F, n_r, N_f, \text{src}, \text{tgt}, \text{gd}, \text{actf})$ be the play-out tree (Definition 27) with flow reduction (Definition 28), let Γ be a one-to-many matching between L and M' (Definition 31), let ω be a weighting function for paths (Definition 32), and let $f \in F$ be a flow. Then, $\text{MGrd}_{\top}^{\rightarrow}$ is a function that yields the weighted sum of evaluations that were true for a given flow:*

$$\text{MGrd}_{\top}^{\rightarrow}(f, \Gamma, L) = \sum_{\sigma \in L} \sum_{\rho \in \Gamma(\hat{\sigma})} \omega(\hat{\sigma}, \rho) \cdot \left[\exists_{1 \leq i \leq |\rho|} [\rho_i = f \wedge \llbracket \text{gd}(f) \rrbracket(\mu_{\sigma, i-1})] \right].$$

Next, the proposed guard-recall measure is formally introduced using weighted traversal bookkeeping.

Definition 38 (Guard-Recall) Let L be a log, let M be a model, let L_M be the play-out log of M (Definition 25), let $M' = (N, F, n_r, N_f, \text{src}, \text{tgt}, \text{gd}, \text{actf})$ be the play-out tree (Definition 27) with flow reduction (Definition 28), and let Γ be a one-to-many matching between L and M' (Definition 31). Then, grec is proposed as a function taking a log L , a matching Γ , and a tree M' , defined as:

$$\text{grec}(L, \Gamma, M') = \frac{\sum_{f \in F} \text{MGrd}_{\vec{\Gamma}}(f, \Gamma, L)}{\sum_{\sigma \in L} \sum_{\rho \in \Gamma(\hat{\sigma})} \frac{1}{|\Gamma(\hat{\sigma})|} \cdot |\sigma|}.$$

The proposed definition for grec consists of a numerator which is the total weighted sum of guard evaluations that were true, accounting for whether they are optimal paths or not; while the denominator is the weighted sum of the flows recorded in the paths, where each flow is given the optimal share.

5.7.2 Guard-Precision

Now, the proposed guard-precision function gprec is introduced. Guard-precision quantifies whether guards have been formed over the observed data attributes and whether they are only evaluated to true when the associated flow is taken in paths. Ideally, guard-precision differentiates between data-aware models that have guards that are more exclusive than another with respect to an event log. Like the previous formulation, the proposal for gprec is constructed using bookkeeping functions. As such, an exploratory bookkeeping function that uses a one-to-many matching to count all the possible guard evaluations that were true is introduced.

Definition 39 (Weighted Exploratory Bookkeeping) Let L be a log, let M be a model, let L_M be the play-out log of M (Definition 25), let $M' = (N, F, n_r, N_f, \text{src}, \text{tgt}, \text{gd}, \text{actf})$ be the play-out tree (Definition 27) with flow reduction (Definition 28), let Γ be a one-to-many matching between L and M' (Definition 31), and let $f \in F$ be a flow. Then, $\text{MGrd}_{\vec{\Gamma}}$ is a function that yields the total number of possible evaluations that were true for that flow:

$$\text{MGrd}_{\vec{\Gamma}}(f, \Gamma, L) = \sum_{\sigma \in L} \sum_{\rho \in \Gamma(\hat{\sigma})} \frac{[\exists_{1 \leq i \leq |\rho|} [\text{src}(f) = \text{src}(\rho_i) \wedge \llbracket \text{gd}(f) \rrbracket(\mu_{\sigma, i-1})]]}{|\Gamma(\hat{\sigma})|}.$$

Next, the proposed guard-precision measure is formally denoted using weighted exploratory bookkeeping.

Definition 40 (Guard-Precision) Let L be a log, let M be a model, let L_M be the play-out log of M (Definition 25), let $M' = (N, F, n_r, N_f, \text{src}, \text{tgt}, \text{gd}, \text{actf})$ be the play-out tree (Definition 27) with flow reduction (Definition 28), let Γ be a one-to-many matching between L and M' (Definition 31), and let $c \in \mathbb{R}$ be a constant factor s.t $c > 0$. Then, gprec is proposed as a function taking a log L , a matching Γ , and a tree M' , defined as:

$$\text{gprec}(L, \Gamma, M') = \frac{c \cdot [\text{optimal}(L, \gamma, M')] + \sum_{f \in F} \text{MGrd}_{\vec{\Gamma}}(f, \Gamma, L)}{c + \sum_{f \in F} \text{MGrd}_{\vec{\Gamma}}(f, \Gamma, L)}.$$

Note that $MGrd_{\top}(f, \Gamma, L)$ will always be at least as large as $MGrd_{\top}^{\rightarrow}(f, \Gamma, L)$, so $gprec$ will always be a ratio between zero and one. However, for the cases where for all $f \in F$: $MGrd_{\top}^{\rightarrow}(f, \Gamma, L) = MGrd_{\top}(f, \Gamma, L) = 0$, the constant c is included so that the ratio returned is 1, to allow for adherence to Axiom 8, which requires guard-precision to return max.

5.8 Evaluation

This section evaluates the adherence of existing measures [169], [170], [173], [189] and the two proposed measures (in Section 5.7) with the axioms proposed in Section 5.6. This evaluation focuses not only on adherence to the axioms but also whether existing techniques encapsulate the notions of guard-recall or guard-precision. To this end, the evaluation adopts a ‘counter-example’ approach, as existing work has done [42], where measure by measure, axiom by axiom, adherence is tested. The determination that a measure does not adhere to the relevant axiom, is based on if a single counter-example (drawn from exemplar logs and models) can be found.

5.8.1 Existing Techniques

This evaluation selected, from existing literature, all known data-aware conformance checking techniques for which a working implementation exists. The existing techniques for a $grec$ function are (i) $grec_M$ using recall reported by [170], and (ii) $grec_F$ using a weighted average of distances from equivalent trace classes reported by [173]. Then, the only existing technique for a $gprec$ function is (iii) $gprec_M$ using precision reported by [189]. For each of these techniques, the default cost-model was used for its underlying alignment technique. These techniques are formally introduced in Appendix A.1, noting that for $grec_F$ [173] this work proposes a metric in a similar manner to the other techniques [169], [170].

5.8.2 Exemplar Log and Model

As in similar previous work [42], [67], [68], this evaluation makes use of counter-examples to test adherence to an axiom. Accordingly, a synthetic event log and a control-flow model for counter-examples are used. In terms of control-flow, the log and model are perfectly matching, to exclude control-flow issues. The base model has one guard already introduced and this guard remains unchanged, i.e. the guard for activity ‘a’ is always true (see Figure 5.8). This ensures that any constructed counter-example from this model could be relaxed data-sound [55, Def. 5.1], which requires at least one valid firing sequence from initial state to final state for the model.

The exemplar event log L_E is shown below. It consists of nine variants and 18 sequences of assignments, and considers one attribute, d_1 , with an integer domain $[5 \dots 10]$. However, when

testing for adherence to the proposed axioms, an enlargement of this log is used, i.e. L_E^k with $k = 100$ (Definition 18).

$$L_E = \{ \langle a, b, e, f \rangle \mapsto [\langle \pi_1, \emptyset, \emptyset, \emptyset \rangle^3, \langle \pi_2, \emptyset, \emptyset, \emptyset \rangle^3], \langle a, b, e, g \rangle \mapsto [\langle \pi_1, \emptyset, \emptyset, \emptyset \rangle^2, \langle \pi_2, \emptyset, \emptyset, \emptyset \rangle^2], \\ \langle a, b, e, h \rangle \mapsto [\langle \pi_1, \emptyset, \emptyset, \emptyset \rangle^1, \langle \pi_2, \emptyset, \emptyset, \emptyset \rangle^1], \langle a, c, e, f \rangle \mapsto [\langle \pi_3, \emptyset, \emptyset, \emptyset \rangle^3, \langle \pi_4, \emptyset, \emptyset, \emptyset \rangle^3], \\ \langle a, c, e, g \rangle \mapsto [\langle \pi_3, \emptyset, \emptyset, \emptyset \rangle^2, \langle \pi_4, \emptyset, \emptyset, \emptyset \rangle^2], \langle a, c, e, h \rangle \mapsto [\langle \pi_3, \emptyset, \emptyset, \emptyset \rangle^1, \langle \pi_4, \emptyset, \emptyset, \emptyset \rangle^1], \\ \langle a, d, e, f \rangle \mapsto [\langle \pi_5, \emptyset, \emptyset, \emptyset \rangle^3, \langle \pi_6, \emptyset, \emptyset, \emptyset \rangle^3], \langle a, d, e, g \rangle \mapsto [\langle \pi_5, \emptyset, \emptyset, \emptyset \rangle^2, \langle \pi_6, \emptyset, \emptyset, \emptyset \rangle^2], \\ \langle a, d, e, h \rangle \mapsto [\langle \pi_5, \emptyset, \emptyset, \emptyset \rangle^1, \langle \pi_6, \emptyset, \emptyset, \emptyset \rangle^1] \}$$

and $d_1 \in \mathcal{U}_{att}$ with:

$$\pi_1 = \{(d_1, 5)\}, \pi_2 = \{(d_1, 6)\}, \pi_3 = \{(d_1, 7)\}, \pi_4 = \{(d_1, 8)\}, \pi_5 = \{(d_1, 9)\}, \pi_6 = \{(d_1, 10)\}.$$

5.8.3 Counter-examples

Figure 5.8 shows three models used in the construction of counter-examples. Figure 5.8a, denotes a model with an ideal guard-precision annotation, but struggles with guard-recall as the guards are never enabled. Figure 5.8b, denotes a model with laxer guards in terms of guard-precision, but an improvement in guard-recall as some guards in the first choice can be enabled. Note, that for these two models, that activity ‘e’ has a guard that constricts the language of these models as d_1 must be written to be less than 5. Figure 5.8c denotes a model which performs the best in terms of guard-recall, w.r.t. L_E^k , but also the worst in terms of guard-precision as the later choice always enables all options. Finally, all three models have an isomorphism up to the control-flow with each other.

For most axioms, many counter-examples were constructed to show that a candidate does not adhere to an axiom. However, for Axiom 1 & Axiom 2, a more straightforward reasoning is used to determine adherence as these relate to how candidates are formulated. For Axioms Axiom 6 & Axiom 9, any of the following were used: a matching γ , a model M , a log L_E^k combinations, and a further enlarged log L_E^K s.t. $k \ll K$ to test that for a measure m , $m(L_E^k, \gamma, M) = m(L_E^K, \gamma, M)$ holds.

For Axiom 3, possible counter-examples were constructed using: L_E^k , a matching γ and an isomorphic version γ' , and any of the following pairs of models: $(M:5.8a, M':5.8b)$, $(M:5.8a, M':5.8c)$, or $(M:5.8b, M':5.8c)$. All of these pairs have been constructed to be such that they are strictly increasing in terms of (true) traversal bookkeeping from left to right. These counter-examples expect that for any measure that is a candidate for guard-recall, $\text{grec}(L_E^k, \gamma, M) < \text{grec}(L_E^k, \gamma', M')$ should hold, otherwise the candidate does not adhere.

The potential counter-examples for Axiom 4 used: L_E^k , a matching γ , and a modified model M of Figure 5.8a, where the guard for ‘a’ was changed, preventing it from being evaluated to true w.r.t L_E^k . Such a guard could be $d_1 < 5$, $(d_1 \neq \perp \wedge d_1 = \perp)$ or simply false. The expectation was that for any candidate of guard-recall, $\text{grec}(L_E^k, \gamma, M) = \min$ should hold, otherwise the candidate does not adhere.

Next, the counter-examples for Axiom 5 used: L_E^k , a model M (Figure 5.8b), an optimal

matching γ s.t. $\text{optimal}(L_E^k, \gamma, M)$, and a non-optimal γ' to construct possible counter-examples. The construction γ' ensured all paths ended with the same flow (either 'f', 'g', or 'h'), thus for some traces the associated path would be optimal but not for all. Also, by using Figure 5.8b and only changing the last flow in paths, the (true) traversal bookkeeping will be the same between matchings. These counter-examples expect that for any candidate for guard-recall, $\text{grec}(L, \gamma, M) < \text{grec}(L, \gamma', M)$ should hold, otherwise the candidate does not adhere.

Changing to guard-precision, the potential counter-examples for Axiom 7 used: L_E^k , a matching γ and an isomorphic version γ' , and any of the following pairs of models: (M:5.8c, M':5.8b), (M:5.8c, M':5.8a), or (M:5.8b, M':5.8a). All of these pairs have been constructed to be such that they are strictly decreasing in terms of (true) exploratory bookkeeping from left to right. The expectation of these was that for any candidate of guard-precision, $\text{gprec}(L_E^k, \gamma, M) < \text{gprec}(L_E^k, \gamma', M')$ should hold, otherwise the candidate does not adhere.

Next, the counter-examples for Axiom 8 used: L_E^k , a model M (Figure 5.8a), an optimal matching γ s.t. $\text{optimal}(L_E^k, \gamma, M)$, and a non-optimal γ' to construct possible counter-examples. The expectation of this counter-example was that for any candidate of guard-precision, both $\text{gprec}(L_E^k, \gamma, M) = \max$ and $\text{gprec}(L_E^k, \gamma', M) < \max$ should hold, otherwise the candidate did not adhere. The usage of γ' was to ensure that the measure did indeed account for a less-than-optimal matching before returning $\max$.

5.8.4 Implementation

The evaluation and proposed techniques have been implemented as an MVP using Python. The implementation takes an event log in the IEEE XES format and any DPN using the PNML format. This implementation could be easily extended to consider any other executable data-aware process model, so long as a play-out log is derivable. Given that existing decision mining techniques only produce DPNs, limiting to DPNs is not seen as a major limitation and the implementation can be extended in future work. The evaluation of existing techniques and the proposed measures, including all counter-examples, can be found here². This repository describes the detailed steps taken for the evaluation to ensure the reproducibility of the presented results.

5.8.5 Results

For each axiom, this evaluation investigated if a counter-example existed to show that a candidate does not adhere to that axiom (which can be found here³). If no negative counter-example was found, a proof sketch is included for adherence in these cases in Appendix A.2. The findings of the evaluation is summarised in Table 5.1, where '✓' denotes adherence to an axiom,

² github.com/AdamBanham/data-aware-conformance

³ github.com/AdamBanham/data-aware-conformance/tree/main/axioms

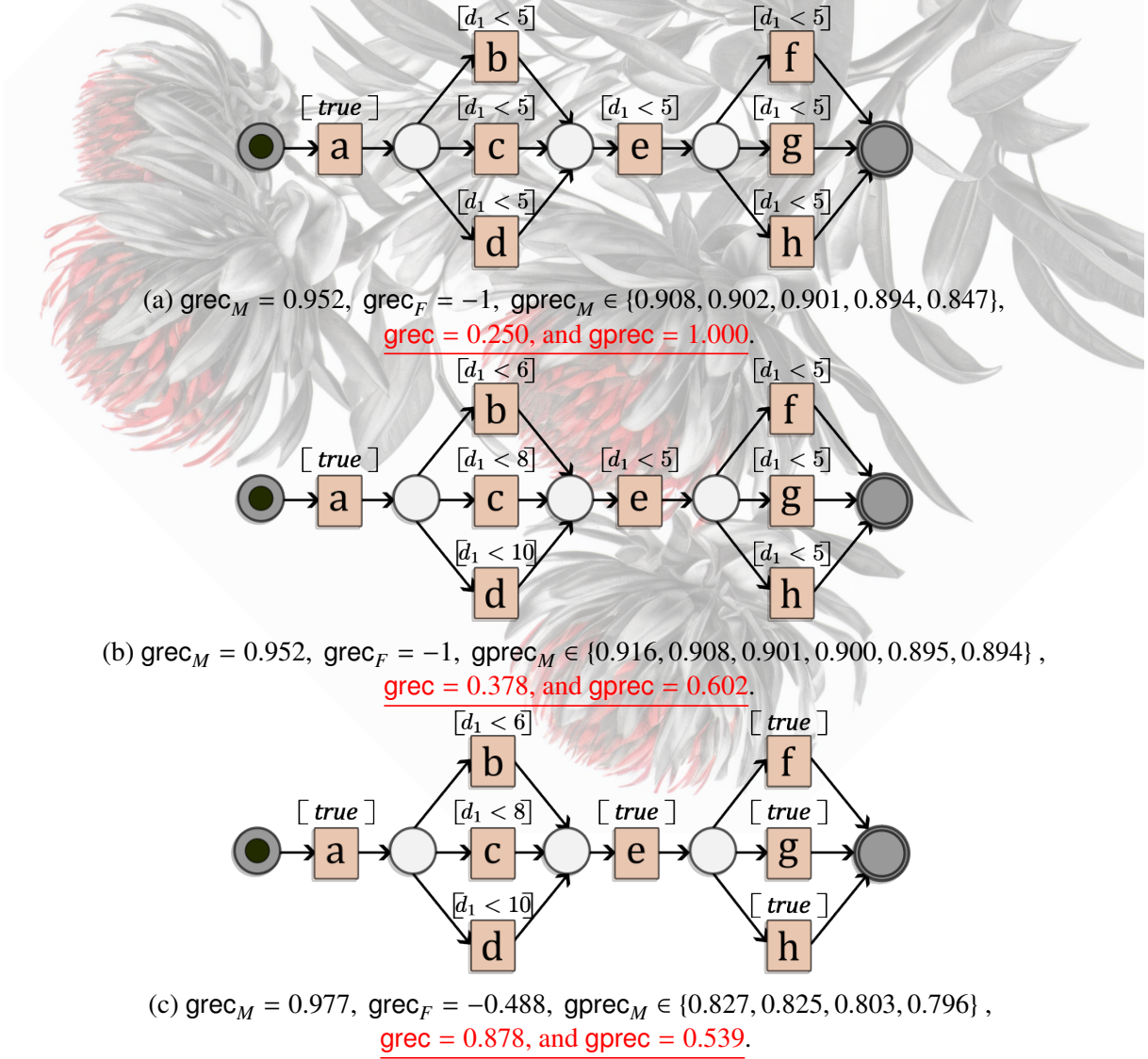


Figure 5.8: A series of counterexamples using one log, where true guard evaluations are monotonically increasing from (a) to (c) with respect to L_E^k , where the number of compliant events in traces for models are: (a) 1, (b) 2, (c) 4. Measurements are denoted in sub captions. The variable d_1 has an integer domain of $[5 \dots 10]$ and is written by ‘a’.

‘ $\times$ ’ denotes disregard for an axiom, ‘-’ denotes irrelevance, and ‘*’ denotes impediments during testing. Impediments could be caused by tightly coupled concepts or by the requirements of a candidate. The proposed measures from Section 5.7, grec (Definition 38) and gprec (Definition 40), are highlighted in Table 5.1.

The findings in Table 5.1 show that no existing grec technique adheres to Axiom 1, i.e., that guard representation should not matter, only behaviour does. That is, these techniques rely on guards being in a specific form (representation) so they can perform their optimisations. Also, this chapter reasons that gprec_M does not adhere to Axiom 2, as gprec_M requires traces that exactly match the model with no deviations in either control-flow or data perspective. Therefore, the implementation requires creating a log of fitting traces that fulfil guard evaluations and write constraints of a DPN [55], [204], and the resulting log will only contain assignments that

Table 5.1: Counter-example testing across our axioms.

Axiom	grec_M	grec_F	grec	gprec_M	gprec
A1/A2	×	×	✓	×	✓
A3	×	×	✓	-	-
A4	×	×	✓	-	-
A5	-*	-*	✓	-	-
A6/A9	✓	✓	✓	×	✓
A7	-	-	-	×	✓
A8	-	-	-	×	✓

evaluate guards to be true.

For the remaining axioms related to guard-recall, the following was observed for existing grec techniques. Consider Figure 5.8, from (a) to (c), where the number of guard evaluations being true increases monotonically. As all candidates return the same value for Figure 5.8a & Figure 5.8b, they do not adhere to Axiom 3, which states that there should be a difference. All guard-recall candidates require a relaxed data sound model [55], [173], hence they cannot adhere to Axiom 4, as if no guard can be true when taken, the model cannot be relaxed data sound. Axiom 5 focuses on dissimilar matchings and as all existing guard-recall candidates do not encounter dissimilar matchings (i.e. they always use the “optimal” alignment), the axiom does not apply. Finally, given a log or an enlargement for any counter-example, candidates return the same value, showing adherence to Axiom 6.

As gprec_M requires compliant traces and uses an alignment technique [170] to generate compliant assignments, and these may take on a range of possible values, means that an implementation of gprec_M has the potential to be non-deterministic. In this testing, it was noted that [170] does not consistently return the same assignments, this induces non-deterministic behaviour in gprec_M [189] for a given log and model, i.e gprec_M returned several values over ten unique runs shown in Figure 5.8. The resulting behaviour from this requirement impedes gprec_M from adhering to any axiom, but with a deterministic implementation, it would theoretically adhere to many.

Now, consider Figure 5.8, from (c) to (a), where the total number of guards evaluating to true is monotonically decreasing. Then, gprec_M does not return strictly increasing measurements across these models. Thus, it does not adhere to Axiom 7. Next, gprec_M does not adhere to Axiom 8, as it could not return max for Figure 5.8a. Finally, given a log or enlargement for any counter-example, gprec_M does not return the same value, and thus does not adhere to Axiom 9.

The findings of this evaluation shows that the existing candidates for either guard-recall or guard-precision, do not adhere to all related axioms. In comparison to existing techniques, the proposed measures (grec or gprec) can cater for guards of any representation, as each uses bookkeeping, and the theory for guards does not optimise over the form of a guard. Hence, each proposed measure grec and gprec , adhere to Axiom 1 & Axiom 2.

As for Axiom 3, consider Figure 5.8, from (a) to (c), where the number of guard evalua-

tions being true increases monotonically. In this case, `grec` produces measurements that strictly increase between these models, showing adherence to Axiom 3. Next, for Axiom 4, the proposed `grec` uses bookkeeping based on guards evaluating to true and returns zero if and only if no guards across all paths evaluate to true. Several more counter-examples were constructed with differing guard representations for this axiom and in each case `grec` returned zero (min) showing adherence to Axiom 4.

Then, for Axiom 5, counter-examples were constructed using Figure 5.8b and L_E^k , where the examples compared using the optimal matching (Definition 36) and a strictly worst matching (e.g. all variants were mapped a path that lead to ‘f’). Thus, the explorative bookkeeping should either be the same or less as the model describes that no guard of latter choice can evaluate to true w.r.t L_E^k . In these cases, `grec` returned a larger value when using the optimal matching over non-optimal matching, showing adherence to Axiom 5. Furthermore, if the log is enlarged and compared with the measurement using the original log, for any counter-example, `grec` and `gprec` returns the same measurement for the enlarged or original log, showing adherence to Axiom 6 & Axiom 9.

For the proposed `gprec` function, consider Figure 5.8, from (c) to (a), where the number of guard evaluations being true is monotonically decreasing. In this case, `gprec` returns measurements that are strictly increasing across these models, showing adherence to Axiom 7. To consider adherence to Axiom 8, Figure 5.8a was considered and various other guard representations with the same behaviour in terms of guard evaluations. However, to further test this axiom another counter-example model which only contained control-flow behaviour up to ‘e’ in Figure 5.8a was used to test that `gprec` does not return max as Axiom 8 describes. In all counter-examples for this axiom, the proposed `gprec` showed adherence by either returning max when expected or not max when expected.

Thus, existing techniques, or their implementations, do not adhere to many of the proposed axioms for guard-recall or guard-precision. However in comparison, the proposed measures have been shown to outperform existing techniques in terms of adherence to the proposed axioms. Furthermore, the biggest difference can be seen between the proposed `gprec` and the existing `gprecM`, where the existing measure does not adhere to Axiom 9, but the proposed measure does. This finding means it is challenging for existing techniques to return a reliable measurement to compare decision mining techniques in terms of guard-precision. But, this is not the case for the proposed technique `gprec`.

5.9 Reflections and Future Work

This section reflects on the development of the quality axioms as well as the proposed measures, guard-recall and guard-precision. While the work of this chapter has been formally denoted and explained at length, further work is still required. This chapter claims the nine axioms (Section 5.6) do describe desirable properties that conformance checking over data-aware models should

have, or at least shows the consequences of not considering them. These quality axioms are yet to be evaluated by experts from process mining or through some qualitative analysis to determine if indeed these axioms are describing what the community would determine to be “desirable”. An interesting outcome for future work could be to conclude whether conformance checking over data-aware models that are not relaxed data sound should not be possible.

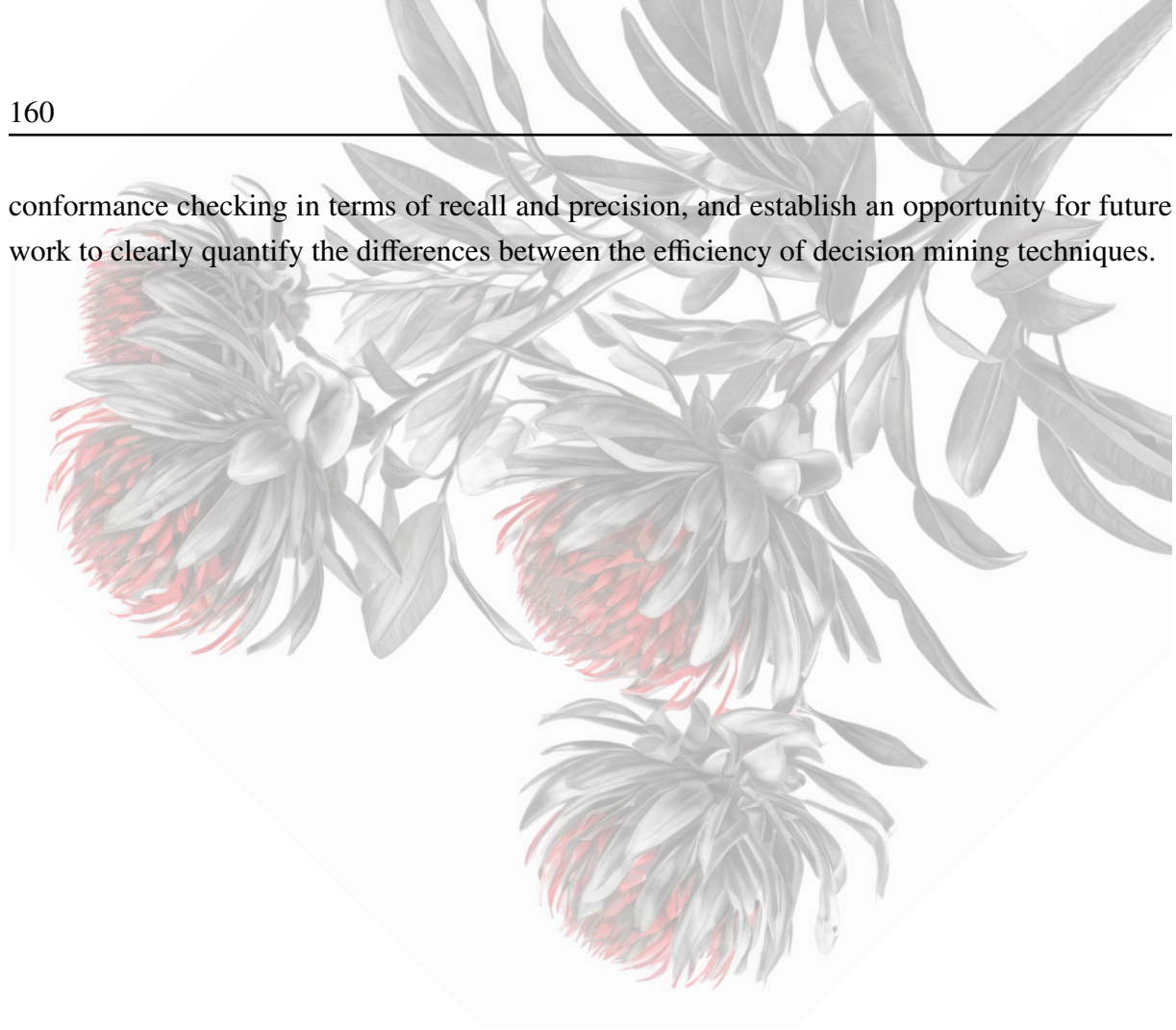
The new measures of guard-recall and guard-precision are a straightforward operationalisation of the foundation section, as such they are not algorithmic contributions like the existing techniques [170], [173], [189]. The current implementation of these measures are limited by the capacity of producing a possible infinite number of playout traces and if rework loops exist in the model, the longest variant in the log. Together these raise very practical concerns about if a performant implementation of the present underlying mechanism for the axioms exists. Future work should consider how to quickly or efficiently compute guard-recall and guard-precision, and evaluate potential implementations runtime performance, memory performance or any other efficiency metrics. To this end, future work could explore how alignment procedures could be altered to cater for the data-aware conformance as presented here and how to propose adhering measures of guard-recall and guard-precision using only alignments.

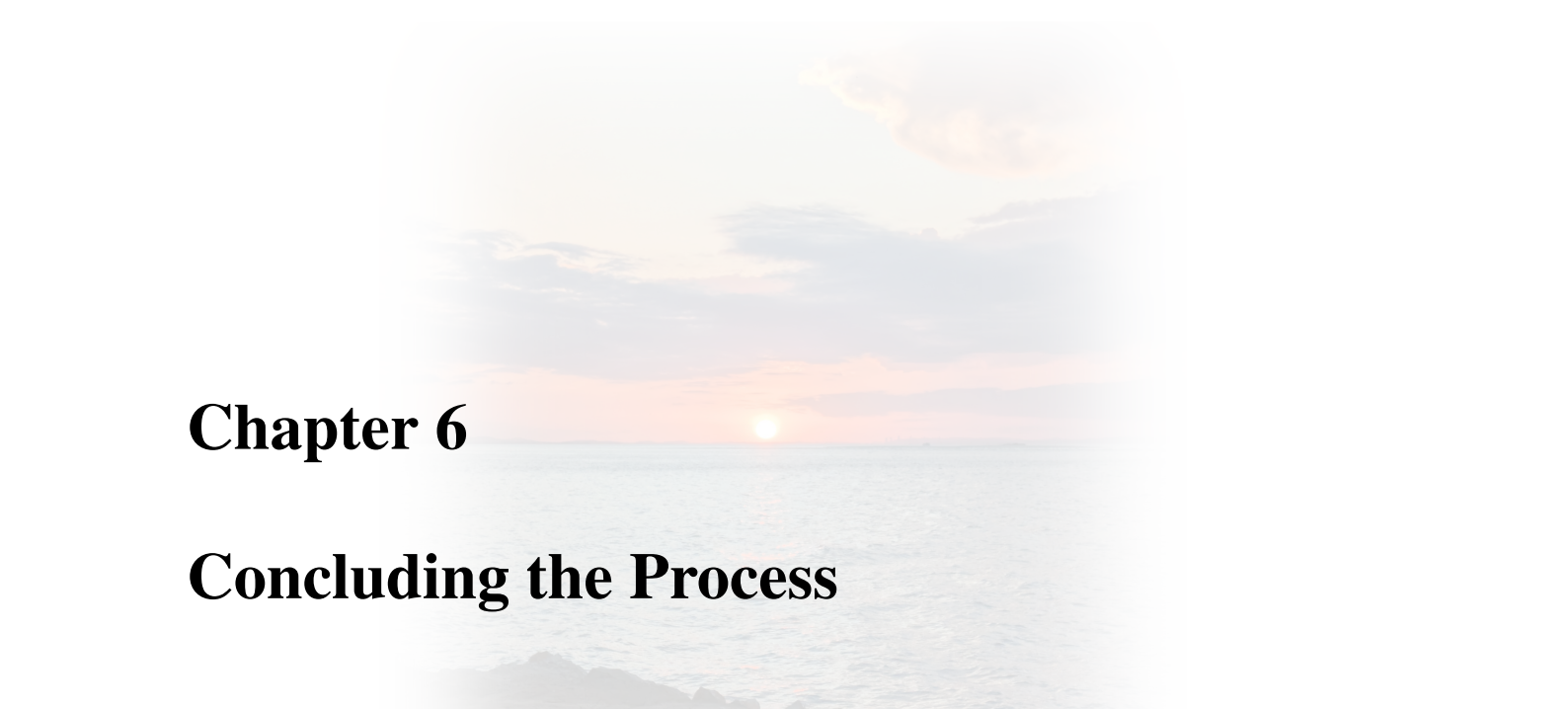
There are many avenues for extensions in future work, including investigating the possible existence of decision mining techniques that guarantee perfect guard-recall or guard-precision, or even both. Data-aware conformance checking could be extended to include generalisation and simplicity, and investigating whether changing the semantics of guards to be policies, as seen in Figure 5.1, affects the desirable properties of conformance checking. Finally, the sensitivity of the axioms to differences in data quality and process model quality should be investigated in a manner similar to studies like Janssenswillen, Donders, Jouck, *et al.* [77] to confirm any bias in measures.

5.10 Summary

This chapter was motivated by a need to compare decision mining outcomes and was concerned that existing conformance checking may run into difficulties when quantifying an outcome. To address these concerns, the chapter introduced the notion of guard-recall and guard-precision in terms of data-aware conformance checking through a novel quantification theory that is model agnostic for executable data-aware models as seen in Figure 5.2. Next, the chapter formalised and presented the desirable properties for data-aware conformance checking as 9 axioms (4 related to work from [42]) which guard-recall and guard-precision functions should satisfy. After which, an evaluation tested existing techniques [170], [173], [189] against counter-examples for the proposed axioms and showed no existing candidates adhere entirely to these axioms. The findings of the evaluation showed a clear need to develop new measures for the notions of guard-recall and guard-precision, and the proposed measures show that it is indeed possible to adhere to the axioms. These contributions demonstrate the importance of having comparable

conformance checking in terms of recall and precision, and establish an opportunity for future work to clearly quantify the differences between the efficiency of decision mining techniques.





Chapter 6

Concluding the Process

The past decades have shown that the development of large software applications is a difficult problem, which is still far from fully understood. In fact, it appears that this problem only increases as applications tend to become more and more complex. The development of information systems, one of the more frequently occurring applications, is a typical example of this problem. Information systems that are reliable, delivered in time, and meet users' expectations are rare.

[Perhaps even more relevant today than ever before] Arthur ter Hofstede; Information modelling in data intensive domains, 1993.

This thesis explored how process mining can take advantage of exogenous data to derive bliss¹. For the wretched, the thesis established three new avenues relevant to their quantification of behaviour towards achieving their goals. For the field of process mining, the thesis presented six advancements towards multi-perspective process mining, decision mining, conformance checking, and process enhancement. For researchers of processes, the thesis proposed algorithms, tools, techniques, and approaches for handling exogenous data to understand how the behaviour of a process may change based on the context surrounding the process. For the curious reader, the thesis has hopefully enabled them to explore process mining through the many visualisations presented, and to see the benefit of clarifying the behaviour of processes. Overall, the thesis is a deep exploratory analysis of *processes*, showing why they are important to consider within organisations and the world at large.

This conclusion is structured in the following manner: Section 6.1 reviews the contributions presented, Section 6.2 reconsiders the research investigation goal and questions, Section 6.3 assesses the threats to validity of the investigation, Section 6.4 envisages the future of process mining with exogenous data, and lastly, Section 6.5 provides retrospective takeaway for the busy.

¹ The definition of 'bliss' is irrelevant and whatever it is, will come from the efforts of the wretched, individuals or organisations, improving their behaviour to obtain more of it.

6.1 The Contributions in Review

This section highlights the six key contributions from Chapters 3, 5 and 4.

Chapter 3 presented two contributions that handle exogenous data for process mining efforts. Firstly, a framework for process mining with exogenous data or **xPM** is presented. The benefit of xPM is that it provides both formal notation and clear semantics relevant to the handling of exogenous data (consisting of time series data). Furthermore, xPM ensures the transformation of exogenous data to event attributes is traceable back to the original time series and opens up the possibility to explore exogenous data for processes. As such, the second contribution presented in Chapter 3 was a visualisation technique called Explorative Exogenous Series Analysis or **EESA**. EESA explores exogenous data and highlights patterns that occur when activities are executed in processes, simplifying a noisy state space into a simple visualisation. A variation of EESA which focused on comparing outcomes was developed to further analyse process outcomes. To overcome the manual task of exploring a large collection of EESAs, an automatic ranking procedure is introduced to highlight where the most or least distinction exists between outcomes visualised by EESAs.

Chapter 3 presented two evaluations, one focusing on xPM and another on EESA. The first evaluation focused on a quantitative approach considering recall and precision of returned models and reported that by incorporating exogenous data within analysis techniques, greater insights within hard annotations could be found. The findings from this evaluation show including exogenous data can have a positive impact on process analysis in knowledge-intensive domains like healthcare. The second evaluation showed that the visualisation of process outcomes based on exogenous data through EESAs can inform subsequent process mining analyses. The analysis highlighted that ranking EESAs can direct practitioners to insights by integrating a vast number of seemingly scattered values into patterns.

Next, Chapter 4 presented two contributions which serve to advance stochastic process mining by considering exogenous influences on activity likelihoods. To this end, a new modelling formalism based on stochastic labelled Petri nets is presented as *stochastic labelled Petri nets with exogenous dependencies* or **Exo-SLPN**. The Exo-SLPN formalism enables the consideration that the firing likelihoods of transitions within the model may be influenced by exogenous data. To discover Exo-SLPNs, a process enhancement technique was developed to extend an existing control-flow model to an Exo-SLPN. Then, a visualisation tool was included to simplify analysis of exogenous influences for the wretched and overcome the somewhat complex semantics of stochastic process mining outcomes.

The evaluation presented in Chapter 4 considered the impact of Exo-SLPN and the proposed visualisation tool. Several publicly available event logs and control-flow models were used to understand the capability of the proposed process enhancement technique to discover meaningful Exo-SLPNs. The visualisation tool was used on these outcomes to highlight where changes in firing likelihoods occurred and considered their usefulness in this matter. The findings of this evaluation showed that Exo-SLPNs are competitive in terms of quality, but are nonetheless are

describing a conditional aspect and as such can fall behind advanced control-flow techniques.

Chapter 5 presented the quantification of data-aware models, a critical contribution to those studying the rediscovering of decision making within process or the influences thereof. To formally explore the quantification of data-aware models, an extensive foundation of formal properties is presented. These foundations enable the expression of recall and precision within data-aware models. To avoid ad hoc operationalisations of recall and precision, desirable properties of the quantification are concluded through **nine quality axioms** and two adhering measures called **guard-recall** and **guard-precision** are presented.

To evaluate the contributions of Chapter 5, existing data-aware techniques and proposed measures are compared against the presented quality axioms as a yardstick. In order to perform the evaluation, counter-examples are constructed to understand what techniques adhere or violate the quality axioms. As such, a synthetic event log and process model fit for purpose was used to construct counter-examples. The evaluation established that many existing techniques struggled to adhere to the quality axioms, while **guard-recall** and **guard-precision** fully adhered to the presented axioms.

6.2 The Research in Retrospect

This thesis investigated two research questions (RQ1) “*how may exogenous influences on decision making in processes be visualised and analysed*” and (RQ2) “*how can the quality of data-aware process models be measured and compared*”. This section reconsiders the contributions for addressing RQ1 and RQ2 and their maturity using the framework described by Werf, Polyvyanyy, Wensveen, *et al.* [91], [92] inline with the research approach established in Section 1.7.1.

To assess the maturity of the six key contributions, a retrospective is performed using the maturity stages described in [92] and presented in tabular forms as in Tables 6.1 and 6.2. To denote that a contribution has sufficient evidence for a stage to be confirmed, the tables denote ✓. To denote that a contribution lacks evidence for the stage to be confirmed, the tables denote ✗. When there is evidence to realising a stage but further work is needed for a firm conclusion, the tables denote ↑. To understand the application of each maturity stage, consider the following evaluation of a known and well established discovery technique, the inductive miner - infrequent (IM_f) [36, Section 6.2.2]. The first stage of maturity [92], asks the contribution to be well designed and should clearly denote:

- what is the class of outcomes produced by the contribution,
 - evidence for the properties that outcomes have,
 - denote the requirements on the given input (or assumptions of the true process),
-

- include an evaluation/exemplar over synthetic data showing the application of the contribution.

IM_f demonstrates maturity for stage one by presenting a formal specification of the algorithm in [36, Chapter 6], although some may ask that an working example implementation would be included in this stage as well. The second maturity stage [92] states contributions should be validated over a collection of real-life event logs to consider the practical use of the contribution. Once again [36, Section 8.1] shows that IM_f was validated over known real-world event logs, presenting evidence for maturity stage two and many other studies including this thesis have conducted similar evaluations for IM_f confirming this stage. It may be feasible within one academic piece of work, say a conference paper or journal article, that these stages are demonstrated and cleared. However, the following stages may require a full piece on their own, as well as many more applications outside of the original study.

The third maturity stage [92] states contributions should establish a relationship between the sample quality of the log and the quality of the outcomes, where evidence may include a formal proof, defining algorithm-specific log quality measures or even statistical evidence for the relationship between sample quality and outcome quality. Furthermore, this relationship may and should consider more than one notion of quality, for example recall, precision, generalisation, and simplicity. The evaluations in [36, Section 8.3] and [36, Section 8.4] show exemplar that confirms a relationship between sample quality and outcome quality, highlighting that IM_f demonstrates maturity for stage three.

The fourth maturity stage [92] states contributions should validate how effective the algorithm is in real-life situations, where validation should focus on situations where the true process is unknown. The focus of this stage is to consider the effectiveness of the underlying assumptions and the impact on outcomes if the assumptions are challenged in practice. For instance, Werf, Polyvyanny, Wensveen, *et al.* [92] argues that IM_f assumes that event logs are generated by process trees, but no test is conducted to know the practicality of this assumption, which sets a very high bar for what evidence is required for a firm conclusion on this maturity stage. However, IM_f has been applied and used by companies around the world for the approach's stability as well as researchers for easy-of-use, guarantee of soundness [36, Section 6.1.3], and rediscoverability [36, Theorem 6.18], which shows the practical effectiveness of IM_f². Therefore, the presumable conclusion on a retrospective for IM_f is the following: ✓, ✓, ✓, ↑ for the four stages of maturity.

The contributions developed for RQ1 and RQ2 are now retrospectively analysed to ascertain their maturity [92]. RQ1 had three sub-questions to guide the research:

- **RQ1.1:** how can exogenous data be represented alongside endogenous data in event logs?
- **RQ1.2:** how can exogenous influences on process outcomes be visualised and analysed?

² Although, an argument could be made that the rediscoverability of processes is always (practically) possible through the directly follow relations of observations, may somewhat limit the application of IM_f.

- **RQ1.3:** how can exogenous influences on the stochastic behaviour of processes be analysed?

The outcomes of RQ1.1 and RQ1.2 were presented in Chapter 3, with the contributions of xPM and EESA. Likewise, the outcomes of RQ1.3 was presented in Chapter 4, with the contributions of Exo-SLPN.

Outcomes for RQ1 — xPM, EESA, Exo-SLPN, show the maturity of being well designed through formal denotation and include running examples of their applications, as outlined in Sections 3.4, 3.5 and 4.6. These outcomes were validated with real-life logs and their evaluations meet the requirements of being validated, as outlined in Sections 3.6.1 and 3.6.2. Testing these outcomes under extensive sampling and in the face of noise has not been conducted as these issues were not in scope for this thesis, meaning the requirements for the third stage were not met. All outcomes were tested on an event log without a known ground truth model, specifically the MIMIC-III dataset [47]. The evaluations of xPM and EESA involved healthcare domain experts testing these outcomes to find useful insights (Section 3.6.2). This retrospective posits there is evidence for the fourth stage of maturity, but not enough for a firm conclusion on the matter. A sampling analysis could be conducted to confirm the findings for xPM and EESA across several domains and with many more experts before claiming practical effectiveness. Therefore the retrospective for outcomes addressing RQ1 show maturity across stage one, two, and four, showing the outcomes have been developed to maturity across the sates that were within scope for the thesis.

Table 6.1: Retrospective of the contributions for research question one.

Maturity Stages [92]	xPM	EESA	Exo-SLPN
S1, of being well designed	✓	✓	✓
S2, being validated on real-life logs	✓	✓	✓
S3, of studying noise/imperfection	✗	✗	✗
S4, of testing practical effectiveness	↑	↑	✗

RQ2 had two sub-questions to guide the research:

- **RQ2.1:** what are desirable properties for quantifying the quality of data-aware process models?
- **RQ2.2:** what are precision and recall measures that adhere to these desirable properties?

The significant outcomes of RQ2.1 and RQ2.2 were nine quality axioms and two new measures, guard-recall and guard-precision, as presented in Chapter 5.

The contributions related to RQ2 are now evaluated across the four maturity stages [92]. The axioms and measures demonstrate maturity of being well designed through their formal denotation and their applications in examples, as outlined in Sections 5.5, 5.6 and 5.7. These axioms and measures focused on formal properties held by techniques and did not validate

findings over real-life logs and instead used synthetic event logs. But the findings nonetheless are the same, regardless of such validation, and are not limited by not reaching maturity stage two. While extensive sampling of the relationship of noise was not included, this retrospective posits that evidence (Section 5.8) is presented showing maturity for stage three, but without a firm conclusion. Furthermore, the proposed axioms describe an explicit relationship between recall and precision through the presented foundations and demonstrate maturity for stage four by understanding if an algorithm is effective through the testing of existing techniques. But unlike existing techniques which explored an in-depth analysis of their implementations, the proposed measures are not algorithmic in nature. As such, this retrospective argues that the axioms and testing in Sections 5.6 and 5.8 are evidence towards stage four without a firm conclusion. Therefore the retrospective concludes that contributions for RQ2 show maturity across stage one, three, and four, as outlined in Table 6.2.

Table 6.2: The retrospective outlook on the contributions for research question two.

Maturity Stages [92]	Quality Axioms	Guard-recall	Guard-precision
S1, of being well designed	✓	✓	✓
S2, being validated on real-life logs	✗	✗	✗
S3, of studying noise/imperfection	↑	↑	↑
S4, of testing practical effectiveness	↑	↑	↑

The research presented here has shown strong evidence that contributions for RQ1 and RQ2 are sound, applicable and useful in terms of answering these research questions. Further research will always be required, but these contributions offer the first opportunity to explore exogenous influences on processes, not seen in literature to date.

6.3 The Validity in Hindsight

This section considers the threats to validity for this thesis, where four major threats to validity are noted. Firstly, often evaluations lacked a confirmed ground truth and could not confirm if whether outcomes, like xPM, EESA, or Exo-SLPN, could always derive a known exogenous influence. Secondly, data quality issues of exogenous data and the consequences for process mining with exogenous data were not considered, which is an unrealistic expectation for real world data which is often messy, noisy, and incomplete. Thirdly, as exogenous data is positioned as time series data and this research often required the interpretation of large collections of such series, noise within these series can and may impact interpretation. Lastly, the practical effectiveness and impact of the contributions were not tested in real-world case studies with domain experts. These issues and their implications for this thesis are detailed below.

Missing Ground Truths

This thesis did not test that an applications of xPM, EESA, or Exo-SLPN could always derive a known exogenous influence on a process. To explore this issue, a synthetic test using a model and log could be conducted, where an outcome of choice within the process is settled by a known interpretation of a time series data, also synthetically generated. However, the contributions within this thesis are not capable of automatically determining this known interpretation so any testing would need to overlook this component. Furthermore, no statistical testing was introduced to conclude the derived exogenous influences were associations, correlations, causations, or due to random chance³. The implications of this threat mean that often the evaluations of this thesis focused on determining if a positive impact was achieved by including exogenous data, rather than concluding that a relationship was found between the behaviour of a process and exogenous data.

Data Quality Being Out Of Scope

The implications of data qualities issues were determined to be outside of the scope of this thesis. Literature for data quality was reviewed Chapter 2, but data issues within exogenous data was not investigated by this thesis. Time series data can have several data quality issues such as misreported measurements by ill-configured tools, measurements can be missing, or there may be problems with the timestamp allocated to measurements. Several issues that can be noted for process mining data quality [127], [128], [132], [133] could apply to the exogenous data or time series data, these issues were not considered. This thesis instead assumed the world and its data is always imperfect and decision making still occurs based on such data. Research should be conducted on exogenous data regarding quantifying the effect of various data quality issues for processes.

The Effect Of Noisy Signals Was Overlooked

An in-depth investigation of time series properties [222] was not considered when proposing the contributions. Computing summary statistics over a large collection of times series may result in computations that do not reflect the reality of the collection. Series within the collection can consist of stationary time series that are clearly separable and as such summaries like the mean for a given time may misrepresent the properties of the collective. Further research could be conducted on the statistical and descriptive components of time series to strengthen the contributions presented. Statistical instruments could be used to identify white or Gaussian white noise within time series used as potential exogenous influences to avoid spurious findings.

³ www.abs.gov.au/statistics/understanding-statistics/statistical-terms-and-concepts/correlation-and-causation

Practical Effectiveness Had Limited Testing

A case study was not conducted on processes and exogenous data, the research often relied on publicly available data. Without a case study it is difficult to assess the real-world impact of the presented contributions. Furthermore, it is easy for the applications of the contributions to require complexity beyond what is practically possible. To explore this issue, a case study into how actors of processes look at exogenous data while performing activities within processes would be of benefit to the outcomes of this thesis.

6.4 The Future in Foresight

When considering the future, five lines of research are envisaged that could extend the work presented in this thesis.

The Future for Decision Mining

In this thesis decision mining has been used extensively, where decision trees are used to derive ‘human readable’ expressions for annotating process models. Other machine learning techniques exist that could perform this task and even more techniques could be used to annotate guards if the property of human readability is dropped. The procedure of handling the classification problem from diagnostic alignments could be approached from several different perspectives, such as using stochastic information or by considering the markings over the postsets of places. A more direct extension to this thesis could be a decision mining technique built to handle time series data as a variable as opposed to only catering for static values, meaning exogenous data and patterns within could be annotated as guards.

The Future for Stochastic Process Mining

Approaching processes in a more flexible manner is crucial for the future of process mining, as more often contextual factors impact the behaviour of processes, like global pandemics and supply chain events. Stochastic process mining offers techniques that are more flexible in their semantics compared with having only a binary notion of deviation. However, work is still needed to improve the understanding of stochastic models as the greater flexibility does come with greater complexity. Future work could include the stochastic write and read constraints within Petri nets with data. The extension of stochastic process mining in this manner may enable write constraints that relate to data beyond the running execution of a process to study inter-case variables for systems of processes.

Process Mining for Time Series Classification

Process models describe the behaviour of sequences and many formalisms exist to cater for complex relationships between actions. The application of process models as a classifier of sequences for time series classification is as yet, unexplored. A process model could be proposed to ‘accept’ a given sequence if it belongs to its language and process mining could provide the means to find such models for classification efforts. The additional benefit of using a process model is that classification can be done easily for non-numerical data, noting that eight categorical values in concurrency as a process model would allow for $8! = 40,320$ different sequences to be accepted, while nine values would allow for 362,880. However, constructing a process model as a classifier over numerical data may not be tractable so the applications may be limited.

Automated Selection of Determinations

Future work that could serve to advance the xPM framework would be the automated selection of determinations. Achieving this feat without an extensive search algorithm would remove the need for users to have significant domain knowledge. A smarter search algorithm could incorporate some domain constraints to prune the state space for the search. The time series literature may present options to prune this extensive state space and would require a full investigation to explore the automation of configuring the ideal xPM instantiation if only the linking function is considered. Extending the configuration to selecting slicing and transform functions may be even more difficult and have its own massive state space.

Exploring Simplicity and Generalisation for Data-Aware Models

The notions of recall and precision were explored for data-aware process models in this thesis. However, further notions exist in the conformance checking literature, e.g. simplicity and generalisation. Investigating simplicity would be an interesting direction as guards introduced by techniques could be judged by the potential of humans to actually read them. Such an investigation would allow techniques that perhaps may not be performance, but have a high capacity to be insightful for actors to have a quantifiable impact and would serve to advance the understanding of processes. Understanding what the notion of generalisation and what it entails for data-aware models, as this thesis defined and investigated, may require a full investigation to even get started and establish a new branch of process mining focused on the rediscovery of decision making for process mining.

What if the world is indeed a rainbow spaghetti process?

A common metaphor used within the process mining community for the object-centric viewpoint is using rainbow spaghetti [234] (similar to complex metro maps) to discuss inter-connected

processes. Tackling the differences between the object-centric viewpoint, [235] where a singular notion of a case is not required, and traditional singular notion of case, was not explored. However, that does not mean that the contributions of the thesis could not apply to the more general viewpoint of object-centric process mining (OCPM). In many cases, the contributions relied on mapping events to exogenous data, and the notion of events is still present in OCPM. The challenge will be connecting exogenous data without introducing noise into the process analysis, as many linkage functions rely on trace attributes (often, these are preprocessed after the trace is completed). Additionally future work involving exogenous data will play an important role in understanding how external influences apply to interconnected processes, where they could have a far more significant impact on these types of processes.



6.5 The Takeaway

If the wretched, owners and actors of processes, the reader, or even the curious should takeaway one thing from this thesis, let it be **that far too often the context surrounding processes are not holistically considered** when investigating processes. *Resist* the temptations of limiting analysis to the control-flow perspective of processes. *Embrace* that our world is complex and messy which can be reflected in data. *Demand* that decision making should consider the world as holistically as possible through data-driven analysis, opposed to relying on conjecture. *Apply* the contributions of this thesis in the search and derivation of ‘bliss’ regardless of wretchedness.

References

- [1] J. vom Brocke, W. M. P. van der Aalst, T. Grisold, *et al.*, “Process science: The interdisciplinary study of continuous change,” *SSRN Elect Journal*, 2021 (cit. on pp. [1](#), [21](#), [125](#)).
- [2] W. M. P. van der Aalst, *Process Mining - Data Science in Action, Second Edition*. Springer, 2016 (cit. on pp. [1](#), [3](#), [4](#), [10](#), [14](#), [16](#), [21–23](#), [30](#), [39](#), [50–52](#), [82](#), [97](#), [125](#), [127](#)).
- [3] M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of Business Process Management, Second Edition*. Springer, 2018 (cit. on pp. [1](#), [10](#), [16](#), [18–20](#), [40](#), [81](#)).
- [4] C. W. Günther and W. M. P. van der Aalst, “Fuzzy mining - adaptive process simplification based on multi-perspective metrics,” in *BPM*, ser. Lecture Notes in Computer Science, vol. 4714, Springer, 2007, pp. 328–343 (cit. on pp. [1](#), [5](#)).
- [5] W. M. P. van der Aalst, “Process mining: Discovering and improving spaghetti and lasagna processes,” in *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, 2011, pp. 1–7 (cit. on p. [1](#)).
- [6] J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, “Quality dimensions in process discovery: The importance of fitness, precision, generalization and simplicity,” *Int. J. Cooperative Inf. Syst.*, vol. 23, no. 1, 2014 (cit. on pp. [1](#), [5](#), [10](#), [35](#), [37](#), [38](#), [125](#), [126](#)).
- [7] D. Fahland and W. M. P. van der Aalst, “Simplifying discovered process models in a controlled manner,” *Inf. Syst.*, vol. 38, no. 4, pp. 585–605, 2013 (cit. on p. [1](#)).
- [8] J. vom Brocke and J. Mendling, Eds., *Business Process Management Cases, Digital Innovation and Business Transformation in Practice* (Management for Professionals). Springer, 2018 (cit. on pp. [2](#), [22](#), [43](#)).
- [9] L. Reinkemeyer, *Process Mining in Action; Principles, Use Cases and Outlook*. Springer, 2020 (cit. on pp. [2](#), [22](#), [43](#)).
- [10] A. Mamudu, W. Bandara, S. J. J. Leemans, and M. T. Wynn, “A process mining impacts framework,” *Bus. Process. Manag. J.*, vol. 29, no. 3, pp. 690–709, 2023 (cit. on pp. [2](#), [22](#), [43](#)).

-
- [11] M. Wynn, J. D. Weerdt, A. H. M. ter Hofstede, *et al.*, “Cost-aware business process management: A research agenda,” in *ACIS*, 2013, p. 110 (cit. on p. 2).
 - [12] W. Z. Low, J. D. Weerdt, M. T. Wynn, A. H. M. ter Hofstede, W. M. P. van der Aalst, and S. K. L. M. vanden Broucke, “Perturbing event logs to identify cost reduction opportunities: A genetic algorithm-based approach,” in *IEEE Congress on Evolutionary Computation*, IEEE, 2014, pp. 2428–2435 (cit. on p. 2).
 - [13] R. Andrews and M. T. Wynn, “Shelf time analysis in CTP insurance claims processing,” in *PAKDD (Workshops)*, ser. Lecture Notes in Computer Science, vol. 10526, Springer, 2017, pp. 151–162 (cit. on pp. 2, 26, 82).
 - [14] Z. Toosinezhad, D. Fahland, Ö. Köroglu, and W. M. P. van der Aalst, “Detecting system-level behavior leading to dynamic bottlenecks,” in *ICPM*, IEEE, 2020, pp. 17–24 (cit. on pp. 2, 21).
 - [15] D. Fahland, V. Denisov, and W. M. P. van der Aalst, “Inferring unobserved events in systems with shared resources and queues,” *Fundam. Informaticae*, vol. 183, no. 3-4, pp. 203–242, 2021 (cit. on p. 2).
 - [16] L. T. Ly, F. M. Maggi, M. Montali, S. Rinderle-Ma, and W. M. P. van der Aalst, “Compliance monitoring in business processes: Functionalities, application, and tool-support,” *Inf. Syst.*, vol. 54, pp. 209–234, 2015 (cit. on p. 2).
 - [17] M. S. Qafari and W. M. P. van der Aalst, “Fairness-aware process mining,” in *OTM Conferences*, ser. Lecture Notes in Computer Science, vol. 11877, Springer, 2019, pp. 182–192 (cit. on p. 2).
 - [18] J. vom Brocke, M. Baier, T. Schmiedel, K. Stelzl, M. Röglinger, and C. Wehking, “Context-aware business process management,” *Bus. Inf. Syst. Eng.*, vol. 63, no. 5, pp. 533–550, 2021 (cit. on pp. 2, 10, 24, 26, 43, 78, 81, 82).
 - [19] J. Yang, C. Ouyang, G. Dik, P. Corry, and A. H. M. ter Hofstede, “Crop harvest forecast via agronomy-informed process modelling and predictive monitoring,” in *CAiSE*, ser. Lecture Notes in Computer Science, vol. 13295, Springer, 2022, pp. 201–217 (cit. on pp. 2, 26, 82).
 - [20] A. Partington, M. T. Wynn, S. Suriadi, C. Ouyang, and J. Karnon, “Process mining for clinical processes: A comparative analysis of four australian hospitals,” *ACM Trans. Manag. Inf. Syst.*, vol. 5, no. 4, 19:1–19:18, 2015 (cit. on pp. 2, 34, 49, 50).
 - [21] F. Mannhardt and D. Blinde, “Analyzing the trajectories of patients with sepsis using process mining,” in *RADAR+EMISA@CAiSE*, ser. CEUR Workshop Proceedings, vol. 1859, CEUR-WS.org, 2017, pp. 72–80 (cit. on pp. 2, 40, 49, 73).
 - [22] N. Martin, J. D. Weerdt, C. Fernández-Llatas, *et al.*, “Recommendations for enhancing the usability and understandability of process mining in healthcare,” *Artif. Intell. Medicine*, vol. 109, p. 101962, 2020 (cit. on pp. 2, 10, 27, 50).
-

-
- [23] A. Banham, S. J. J. Leemans, M. T. Wynn, and R. Andrews, “xPM: A framework for process mining with exogenous data,” in *ICPM Workshops*, ser. Lecture Notes in Business Information Processing, vol. 433, Springer, 2021, pp. 85–97 (cit. on pp. [2](#), [8](#), [10](#), [12](#), [34](#), [41](#)).
 - [24] A. Banham, S. J. J. Leemans, M. T. Wynn, R. Andrews, K. B. Laupland, and L. Shinnars, “xPM: Enhancing exogenous data visibility,” *Artif. Intell. Medicine*, vol. 133, p. 102 409, 2022 (cit. on pp. [2](#), [8](#), [10](#), [12](#), [24](#), [26](#), [34](#), [82](#), [93](#), [113](#)).
 - [25] M. de Leoni and F. Mannhardt, *Road traffic fine management process*, en, 2015 (cit. on pp. [2](#), [3](#), [8](#), [31](#), [33](#), [40](#), [83](#), [84](#), [93](#), [114](#)).
 - [26] B. F. A. Hompes and W. M. P. van der Aalst, “Lifecycle-based process performance analysis,” in *OTM Conferences (1)*, ser. Lecture Notes in Computer Science, vol. 11229, Springer, 2018, pp. 336–353 (cit. on p. [3](#)).
 - [27] M. L. van Eck, X. Lu, S. J. J. Leemans, and W. M. P. van der Aalst, “PM2 : A process mining project methodology,” in *CAiSE*, ser. Lecture Notes in Computer Science, vol. 9097, Springer, 2015, pp. 297–313 (cit. on p. [3](#)).
 - [28] W. M. P. van der Aalst, “Process mining: A 360 degree overview,” in *Process Mining Handbook*, ser. Lecture Notes in Business Information Processing, vol. 448, Springer, 2022, pp. 3–34 (cit. on p. [4](#)).
 - [29] S. J. J. Leemans, “Automated process discovery,” in *Encyclopedia of Big Data Technologies*, Springer, 2019 (cit. on p. [5](#)).
 - [30] A. H. M. ter Hofstede, “Information modelling in data intensive domains,” Ph.D. dissertation, Katholieke Universiteit Nijmegen (since renamed to Radboud Universiteit), 1993 (cit. on pp. [5](#), [14](#)).
 - [31] W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros, “Workflow patterns,” *Distributed Parallel Databases*, vol. 14, no. 1, pp. 5–51, 2003 (cit. on pp. [5](#), [18](#), [30](#)).
 - [32] W. M. P. van der Aalst, T. Weijters, and L. Maruster, “Workflow mining: Discovering process models from event logs,” *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 9, pp. 1128–1142, 2004 (cit. on pp. [5](#), [30](#), [31](#), [34](#)).
 - [33] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, “Discovering block-structured process models from event logs containing infrequent behaviour,” in *Business Process Management Workshops*, ser. Lecture Notes in Business Information Processing, vol. 171, Springer, 2013, pp. 66–78 (cit. on pp. [5](#), [32](#), [66](#)).
 - [34] A. Augusto, R. Conforti, M. Dumas, M. L. Rosa, and A. Polyvyanyy, “Split miner: Automated discovery of accurate and simple business process models from event logs,” *Knowl. Inf. Syst.*, vol. 59, no. 2, pp. 251–284, 2019 (cit. on p. [5](#)).
-

- [35] H. Kourani and S. J. van Zelst, “POWL: partially ordered workflow language,” in *BPM*, ser. Lecture Notes in Computer Science, vol. 14159, Springer, 2023, pp. 92–108 (cit. on pp. 5, 32, 33, 116).
 - [36] S. J. J. Leemans, *Robust Process Mining with Guarantees - Process Discovery, Conformance Checking and Enhancement* (Lecture Notes in Business Information Processing). Springer, 2022, vol. 440 (cit. on pp. 5, 14, 32–34, 116, 125, 163, 164).
 - [37] L. Wen, W. M. P. van der Aalst, J. Wang, and J. Sun, “Mining process models with non-free-choice constructs,” *Data Min. Knowl. Discov.*, vol. 15, no. 2, pp. 145–180, 2007 (cit. on pp. 5, 30, 31).
 - [38] S. J. J. Leemans, E. Poppe, and M. T. Wynn, “Directly follows-based process mining: Exploration & a case study,” in *ICPM*, IEEE, 2019, pp. 25–32 (cit. on pp. 5, 32, 33, 116).
 - [39] F. M. Maggi, R. P. J. C. Bose, and W. M. P. van der Aalst, “A knowledge-based integrated approach for discovering and repairing declare maps,” in *CAiSE*, ser. Lecture Notes in Computer Science, vol. 7908, Springer, 2013, pp. 433–448 (cit. on pp. 5, 34).
 - [40] A. Alman, C. D. Ciccio, D. Haas, F. M. Maggi, and A. Nolte, “Rule mining with rum,” in *ICPM*, IEEE, 2020, pp. 121–128 (cit. on pp. 5, 34).
 - [41] J. Carmona, B. F. van Dongen, A. Solti, and M. Weidlich, *Conformance Checking - Relating Processes and Models*. Springer, 2018 (cit. on pp. 5, 10, 21, 35, 37, 38, 82, 125, 126).
 - [42] A. F. Syring, N. Tax, and W. M. P. van der Aalst, “Evaluating conformance measures in process mining using conformance propositions,” *Trans. Petri Nets Other Model. Concurr.*, vol. 14, pp. 192–221, 2019 (cit. on pp. 5, 10, 11, 13, 37, 125, 148–150, 153, 159).
 - [43] S. J. J. Leemans and A. Polyvyanyy, “Stochastic-aware precision and recall measures for conformance checking in process mining,” *Inf. Syst.*, vol. 115, p. 102 197, 2023 (cit. on pp. 5, 11, 13).
 - [44] A. T. Burke, S. J. Leemans, M. T. Wynn, W. M. van der Aalst, and A. H. ter Hofstede, “A chance for models to show their quality: Stochastic process model-log dimensions,” *Information Systems*, vol. 124, p. 102 382, 2024, issn: 0306-4379 (cit. on pp. 5, 13).
 - [45] A. Banham, A. H. M. T. Hofstede, S. J. J. Leemans, F. Mannhardt, R. Andrews, and M. T. Wynn, “Comparing conformance checking for decision mining: An axiomatic approach,” *IEEE Access*, vol. 12, pp. 60 276–60 298, 2024 (cit. on pp. 5, 10).
 - [46] W. M. P. van der Aalst, “Verification of workflow nets,” in *ICATPN*, ser. Lecture Notes in Computer Science, vol. 1248, Springer, 1997, pp. 407–426 (cit. on pp. 5, 30, 35, 37).
-

- [47] A. E. Johnson, T. J. Pollard, L. Shen, *et al.*, “MIMIC-III, a freely accessible critical care database,” *Scientific Data*, vol. 3, no. 1, 2016 (cit. on pp. [8](#), [12](#), [24](#), [51](#), [65](#), [75](#), [79](#), [93](#), [113](#), [114](#), [165](#)).
 - [48] M. Dumas, W. M. P. van der Aalst, and A. H. M. ter Hofstede, Eds., *Process-Aware Information Systems: Bridging People and Software Through Process Technology*. Wiley, 2005 (cit. on pp. [10](#), [16](#), [17](#), [19](#), [20](#)).
 - [49] W. M. P. van der Aalst and K. M. van Hee, *Workflow Management: Models, Methods, and Systems* (Cooperative information systems). MIT Press, 2002 (cit. on pp. [10](#), [16](#), [17](#)).
 - [50] J. vom Brocke and M. Rosemann, Eds., *Handbook on Business Process Management I, Introduction, Methods, and Information Systems, 2nd Ed*, International Handbooks on Information Systems, Springer, 2015 (cit. on pp. [10](#), [16](#)).
 - [51] W. M. P. van der Aalst and J. Carmona, Eds., *Process Mining Handbook* (Lecture Notes in Business Information Processing). Springer, 2022, vol. 448 (cit. on pp. [10](#), [16](#), [22](#), [51](#)).
 - [52] M. Rosemann, J. Recker, and C. Flender, “Contextualisation of business processes,” *Int. J. Bus. Process. Integr. Manag.*, vol. 3, no. 1, pp. 47–60, 2008 (cit. on pp. [10](#), [24](#), [25](#), [43](#), [78](#)).
 - [53] W. M. P. van der Aalst and S. Dustdar, “Process mining put into context,” *IEEE Internet Computing*, vol. 16, no. 01, pp. 82–86, 2012, issn: 1941-0131 (cit. on pp. [10](#), [24–26](#), [43](#), [78](#), [81](#)).
 - [54] G. D. Kerpedzhiev, U. M. König, M. Röglinger, and M. Rosemann, “An exploration into future business process management capabilities in view of digitalization,” *Bus. Inf. Syst. Eng.*, vol. 63, no. 2, pp. 83–96, 2021 (cit. on pp. [10](#), [24](#), [26](#), [43](#), [78](#), [81](#)).
 - [55] F. Mannhardt, “Multi-perspective process mining,” Ph.D. dissertation, Eindhoven University of Technology, Eindhoven, The Netherlands, 2018 (cit. on pp. [10](#), [11](#), [14](#), [26](#), [27](#), [38](#), [40](#), [43](#), [53](#), [83](#), [116](#), [125](#), [127](#), [153](#), [156](#), [157](#)).
 - [56] A. Rozinat and W. M. P. van der Aalst, “Decision mining in prom,” in *Business Process Management*, ser. Lecture Notes in Computer Science, vol. 4102, Springer, 2006, pp. 420–425 (cit. on pp. [10](#), [38](#), [40](#)).
 - [57] M. de Leoni and W. M. P. van der Aalst, “Data-aware process mining: Discovering decisions in processes using alignments,” in *SAC*, ACM, 2013, pp. 1454–1461 (cit. on pp. [10](#), [38](#), [40](#), [42](#), [66](#), [68](#), [125](#), [127](#)).
 - [58] F. Mannhardt, M. de Leoni, H. A. Reijers, and W. M. P. van der Aalst, “Decision mining revisited - discovering overlapping rules,” in *CAiSE*, ser. Lecture Notes in Computer Science, vol. 9694, Springer, 2016, pp. 377–392 (cit. on pp. [10](#), [37](#), [39](#), [40](#), [42](#), [43](#), [58](#), [66](#), [69](#)).
-

-
- [59] B. Scheibel and S. Rinderle-Ma, “Decision mining with time series data based on automatic feature generation,” in *CAiSE*, ser. Lecture Notes in Computer Science, vol. 13295, Springer, 2022, pp. 3–18 (cit. on pp. 10, 38, 40, 43, 125, 127).
- [60] E. Rojas, J. Munoz-Gama, M. Sepúlveda, and D. Capurro, “Process mining in healthcare: A literature review,” *J. Biomed. Informatics*, vol. 61, pp. 224–236, 2016 (cit. on pp. 10, 27, 50).
- [61] E. Batista and A. Solanas, “Process mining in healthcare: A systematic review,” in *IISA*, IEEE Computer Society, 2018, pp. 1–6 (cit. on pp. 10, 27).
- [62] A. Guzzo, A. Rullo, and E. Vocaturo, “Process mining applications in the healthcare domain: A comprehensive review,” *WIREs Data Mining Knowl. Discov.*, vol. 12, no. 2, 2022 (cit. on pp. 10, 28, 43, 50).
- [63] R. D. Masellis, C. D. Francescomarino, C. Ghidini, and S. Tessaris, “Solving reachability problems on data-aware workflows,” *Expert Syst. Appl.*, vol. 189, p. 116 059, 2022 (cit. on pp. 10, 37, 38, 40).
- [64] P. Felli, M. de Leoni, and M. Montali, “Soundness verification of data-aware process models with variable-to-variable conditions,” *Fundam. Informaticae*, vol. 182, no. 1, pp. 1–29, 2021 (cit. on pp. 10, 37, 38, 40).
- [65] P. Felli, M. Montali, and S. Winkler, “Soundness of data-aware processes with arithmetic conditions,” in *CAiSE*, ser. Lecture Notes in Computer Science, vol. 13295, Springer, 2022, pp. 389–406 (cit. on pp. 10, 37, 38, 40).
- [66] J. Hidders, M. Dumas, W. M. P. van der Aalst, A. H. M. ter Hofstede, and J. Verelst, “When are two workflows the same?” In *CATS*, ser. CRPIT, vol. 41, Australian Computer Society, 2005, pp. 3–11 (cit. on pp. 10, 128, 129, 133).
- [67] N. Tax, X. Lu, N. Sidorova, D. Fahland, and W. M. P. van der Aalst, “The imprecisions of precision measures in process mining,” *Inf. Process. Lett.*, vol. 135, pp. 1–8, 2018 (cit. on pp. 10, 11, 13, 37, 125, 148, 153).
- [68] W. M. P. van der Aalst, “Relating process models and event logs - 21 conformance propositions,” in *ATAED@Petri Nets/ACSD*, ser. CEUR Workshop Proceedings, vol. 2115, CEUR-WS.org, 2018, pp. 56–74 (cit. on pp. 10, 11, 13, 37, 125–127, 148, 153).
- [69] W. M. P. van der Aalst, A. Adriansyah, A. K. A. de Medeiros, *et al.*, “Process mining manifesto,” in *Business Process Management Workshops*, ser. Lecture Notes in Business Information Processing, vol. 99, Springer, 2011, pp. 169–194 (cit. on pp. 11, 26, 43).
- [70] J. Munoz-Gama, N. Martin, C. Fernández-Llatas, *et al.*, “Process mining for healthcare: Characteristics and challenges,” *J. Biomed. Informatics*, vol. 127, p. 103 994, 2022 (cit. on pp. 11, 26, 43, 50, 73, 82).
-

- [71] C. Janiesch, A. Koschmider, M. Mecella, *et al.*, “The internet of things meets business process management: A manifesto,” *IEEE Systems, Man, and Cybernetics Magazine*, vol. 6, no. 4, pp. 34–44, 2020 (cit. on pp. [11](#), [26](#), [28](#), [43](#), [82](#)).
 - [72] S. J. J. Leemans, F. M. Maggi, and M. Montali, “Reasoning on labelled petri nets and their dynamics in a stochastic setting,” in *BPM*, ser. Lecture Notes in Computer Science, vol. 13420, Springer, 2022, pp. 324–342 (cit. on p. [12](#)).
 - [73] A. Gal, “Everything there is to know about stochastically known logs,” in *ICPM*, IEEE, 2023, pp. xvii–xxiii (cit. on pp. [12](#), [41](#), [82](#), [93](#)).
 - [74] S. J. J. Leemans, F. M. Maggi, and M. Montali, “Enjoy the silence: Analysis of stochastic petri nets with silent transitions,” *Information Systems*, p. 102 383, 2024, issn: 0306-4379 (cit. on p. [12](#)).
 - [75] S. J. J. Leemans, W. M. P. van der Aalst, T. Brockhoff, and A. Polyvyanyy, “Stochastic process mining: Earth movers’ stochastic conformance,” *Inf. Syst.*, vol. 102, p. 101 724, 2021 (cit. on pp. [12](#), [112](#), [116](#), [117](#)).
 - [76] F. Mannhardt, S. J. J. Leemans, C. T. Schwanen, and M. de Leoni, “Modelling data-aware stochastic processes - discovery and conformance checking,” in *Petri Nets*, ser. Lecture Notes in Computer Science, vol. 13929, Springer, 2023, pp. 77–98 (cit. on pp. [12](#), [40](#), [41](#), [46](#), [82–84](#), [93](#), [100](#), [112](#), [116](#), [117](#)).
 - [77] G. Janssenswillen, N. Donders, T. Jouck, and B. Depaire, “A comparative study of existing quality measures for process discovery,” *Inf. Syst.*, vol. 71, pp. 1–15, 2017 (cit. on pp. [13](#), [159](#)).
 - [78] C. Günther, “Process mining in flexible environments,” English, Proefschrift., Phd Thesis 1 (Research TU/e / Graduation TU/e), Eindhoven University of Technology, Eindhoven, The Netherlands, 2009 (cit. on p. [14](#)).
 - [79] A. Rozinat, “Process mining : Conformance and extension,” English, Proefschrift., Phd Thesis 1 (Research TU/e / Graduation TU/e), Eindhoven University of Technology, Eindhoven, The Netherlands, 2010 (cit. on pp. [14](#), [35](#), [36](#)).
 - [80] J. Buijs, “Flexible evolutionary algorithms for mining structured process models,” English, Phd Thesis 1 (Research TU/e / Graduation TU/e), Eindhoven University of Technology, Eindhoven, The Netherlands, 2014 (cit. on pp. [14](#), [32](#)).
 - [81] A. Adriansyah, “Aligning observed and modeled behavior,” Ph.D. dissertation, Eindhoven University of Technology, Eindhoven, The Netherlands, 2014 (cit. on pp. [14](#), [26](#), [31](#), [35–37](#), [40](#), [59](#), [99](#), [125](#), [129](#), [144](#)).
 - [82] A. Pika, “Mining process risks and resource profiles,” Ph.D. dissertation, Queensland University of Technology, Brisbane, Australia, 2015 (cit. on p. [14](#)).
-

-
- [83] W. Z. Low, “Towards cost model-driven log-based business process improvement,” Ph.D. dissertation, Queensland University of Technology, Brisbane, Australia, 2016 (cit. on p. 14).
 - [84] S. Sadeghianasl, “The quality guardian: Improving activity label quality in event logs through gamification,” Ph.D. dissertation, Queensland University of Technology, Brisbane, Australia, 2022 (cit. on pp. 14, 26).
 - [85] J. Yang, “Discovering organizational models from event logs for workforce analytics,” Ph.D. dissertation, Queensland University of Technology, Brisbane, Australia, Oct. 2023 (cit. on p. 14).
 - [86] A. Burke, “Process mining with labelled stochastic nets,” Ph.D. dissertation, Queensland University of Technology, Brisbane, Australia, Feb. 2024 (cit. on pp. 14, 40, 41, 100).
 - [87] T. Colburn, “Methodology of computer science,” in *The Blackwell Guide to the Philosophy of Computing and Information*. John Wiley & Sons, Ltd, 2004, ch. 24, pp. 318–326, ISBN: 9780470757017 (cit. on p. 14).
 - [88] V. Vaishnavi, B. Kuechler, and S. Petter. “Design science research in information systems,” Association for Information Systems. (Jun. 2004) (cit. on pp. 14, 118).
 - [89] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design science in information systems research,” *Management Information Systems Quarterly*, vol. 28, no. 1, p. 75, 2004 (cit. on pp. 14, 118).
 - [90] A. R. Hevner, “A three cycle view of design science research,” *Scandinavian Journal of Information Systems*, vol. 19, p. 4, 2007 (cit. on pp. 14, 118).
 - [91] J. M. E. M. van der Werf, A. Polyvyanyy, B. R. van Wensveen, M. J. S. Brinkhuis, and H. A. Reijers, “All that glitters is not gold - towards process discovery techniques with guarantees,” in *CAiSE*, ser. Lecture Notes in Computer Science, vol. 12751, Springer, 2021, pp. 141–157 (cit. on pp. 14, 163).
 - [92] J. M. E. M. van der Werf, A. Polyvyanyy, B. R. van Wensveen, M. J. S. Brinkhuis, and H. A. Reijers, “All that glitters is not gold: Four maturity stages of process discovery algorithms,” *Inf. Syst.*, vol. 114, p. 102 155, 2023 (cit. on pp. 14, 163–166).
 - [93] W. M. P. van der Aalst, B. F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A. J. M. M. Weijters, “Workflow mining: A survey of issues and approaches,” *Data Knowl. Eng.*, vol. 47, no. 2, pp. 237–267, 2003 (cit. on pp. 17–20, 34).
 - [94] R. Agrawal, D. Gunopulos, and F. Leymann, “Mining process models from workflow logs,” in *EDBT*, ser. Lecture Notes in Computer Science, vol. 1377, Springer, 1998, pp. 469–483 (cit. on pp. 17, 29, 30).
-

- [95] B. F. van Dongen and W. M. P. van der Aalst, “A meta model for process mining data,” in *EMOI-INTEROP*, ser. CEUR Workshop Proceedings, vol. 160, CEUR-WS.org, 2005 (cit. on p. 18).
 - [96] G. Acampora, A. Vitiello, B. N. D. Stefano, W. M. P. van der Aalst, C. W. Günther, and E. Verbeek, “IEEE 1849: The XES standard: The second IEEE standard sponsored by IEEE computational intelligence society [society briefs],” *IEEE Comput. Intell. Mag.*, vol. 12, no. 2, pp. 4–8, 2017 (cit. on pp. 18, 200).
 - [97] M. T. Wynn, W. M. P. van der Aalst, E. Verbeek, and B. N. D. Stefano, “The IEEE XES standard for process mining: Experiences, adoption, and revision [society briefs],” *IEEE Comput. Intell. Mag.*, vol. 19, no. 1, pp. 20–23, 2024 (cit. on pp. 18, 200).
 - [98] C. Ouyang, M. Adams, M. T. Wynn, and A. H. M. ter Hofstede, “Workflow management,” in *Handbook on Business Process Management (1)*, ser. International Handbooks on Information Systems, Springer, 2015, pp. 475–506 (cit. on p. 18).
 - [99] N. Russell, A. Hofstede, ter, W. Aalst, van der, and N. Mulyar, *Workflow control-flow patterns : a revised view* (BPM reports), English. BPMcenter. org, 2006 (cit. on p. 18).
 - [100] N. Russell, W. M. P. van der Aalst, A. H. M. ter Hofstede, and D. Edmond, “Workflow resource patterns: Identification, representation and tool support,” in *CAiSE*, ser. Lecture Notes in Computer Science, vol. 3520, Springer, 2005, pp. 216–232 (cit. on p. 18).
 - [101] R. M. Dijkman, M. Dumas, and C. Ouyang, “Semantics and analysis of business process models in BPMN,” *Inf. Softw. Technol.*, vol. 50, no. 12, pp. 1281–1294, 2008 (cit. on p. 18).
 - [102] A. Smith, *An Inquiry into the Nature and Causes of the Wealth of Nations*. Oxford University Press, 1776 (cit. on p. 18).
 - [103] F. P. Brooks Jr., *The Mythical Man-Month: Essays on Software Engineering*. Addison-Wesley Publishing Company, 1975 (cit. on p. 18).
 - [104] W. M. P. van der Aalst, “Challenges in business process analysis,” in *ICEIS (Selected Papers)*, ser. Lecture Notes in Business Information Processing, vol. 12, Springer, 2007, pp. 27–42 (cit. on p. 21).
 - [105] R. P. J. C. Bose and W. M. P. van der Aalst, “Process diagnostics using trace alignment: Opportunities, issues, and challenges,” *Inf. Syst.*, vol. 37, no. 2, pp. 117–141, 2012 (cit. on p. 21).
 - [106] R. Mans, W. M. P. van der Aalst, R. J. B. Vanwersch, and A. J. Moleman, “Process mining in healthcare: Data challenges when answering frequently posed questions,” in *ProHealth/KR4HC*, ser. Lecture Notes in Computer Science, vol. 7738, Springer, 2012, pp. 140–153 (cit. on pp. 21, 26).
-

-
- [107] W. M. P. van der Aalst, J. Carmona, T. Chatain, and B. F. van Dongen, “A tour in process mining: From practice to algorithmic challenges,” *Trans. Petri Nets Other Model. Concurr.*, vol. 14, pp. 1–35, 2019 (cit. on p. 21).
- [108] W. M. P. van der Aalst, T. Brockhoff, A. F. Ghahfarokhi, M. Pourbafrani, M. S. Uysal, and S. J. van Zelst, “Removing operational friction using process mining: Challenges provided by the internet of production (iop),” in *DATA (Revised Selected Papers)*, ser. Communications in Computer and Information Science, vol. 1446, Springer, 2020, pp. 1–31 (cit. on p. 21).
- [109] N. Martin, D. A. Fischer, G. D. Kerpedzhiev, *et al.*, “Opportunities and challenges for process mining in organizations: Results of a delphi study,” *Bus. Inf. Syst. Eng.*, vol. 63, no. 5, pp. 511–527, 2021 (cit. on p. 21).
- [110] F. M. Maggi, M. Montali, and W. M. P. van der Aalst, “An operational decision support framework for monitoring business constraints,” in *FASE*, ser. Lecture Notes in Computer Science, vol. 7212, Springer, 2012, pp. 146–162 (cit. on p. 21).
- [111] C. Li, T. Shinde, W. He, *et al.*, “Unveiling bottlenecks in logistics: A case study on process mining for root cause identification and diagnostics in an air cargo terminal,” in *ICSOC (2)*, ser. Lecture Notes in Computer Science, vol. 14420, Springer, 2023, pp. 291–307 (cit. on p. 21).
- [112] S. Suriadi, M. T. Wynn, J. Xu, W. M. P. van der Aalst, and A. H. M. ter Hofstede, “Discovering work prioritisation patterns from event logs,” *Decis. Support Syst.*, vol. 100, pp. 77–92, 2017 (cit. on p. 21).
- [113] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936 (cit. on pp. 21, 23).
- [114] R. Mans, W. M. P. van der Aalst, R. J. B. Vanwersch, and A. J. Moleman, “Process mining in healthcare: Data challenges when answering frequently posed questions,” in *ProHealth/KR4HC*, ser. Lecture Notes in Computer Science, vol. 7738, Springer, 2012, pp. 140–153 (cit. on pp. 22, 27, 43, 50, 73).
- [115] T. Grisold, J. Mendling, M. Otto, and J. vom Brocke, “Adoption, use and management of process mining in practice,” *Bus. Process. Manag. J.*, vol. 27, no. 2, pp. 369–387, 2021 (cit. on pp. 22, 23, 43).
- [116] R. Andrews, S. Suriadi, M. Wynn, A. H. M. ter Hofstede, and S. Rothwell, “Improving patient flows at st. andrew’s war memorial hospital’s emergency department through process mining,” in *Business Process Management Cases*, ser. Management for Professionals, Springer, 2018, pp. 311–333 (cit. on p. 22).
- [117] R. Andrews, M. Wynn, A. H. M. ter Hofstede, *et al.*, “Exposing impediments to insurance claims processing,” in *Business Process Management Cases*, ser. Management for Professionals, Springer, 2018, pp. 275–290 (cit. on p. 22).
-

- [118] T. Thaler, S. Norek, V. D. Angelis, D. Maurer, P. Fettke, and P. Loos, “Mining the usability of process-oriented business software: The case of the ARIS designer of software AG,” in *Business Process Management Cases*, ser. Management for Professionals, Springer, 2018, pp. 291–310 (cit. on p. 22).
 - [119] J. Suchý, M. Suchy, M. Rosik, and A. Valkova, “Automate does not always mean optimize: Case study at a logistics company,” in *Business Process Management Cases*, ser. Management for Professionals, Springer, 2018, pp. 463–483 (cit. on p. 22).
 - [120] J. N. Adams, D. Schuster, S. Schmitz, G. Schuh, and W. M. P. van der Aalst, “Defining cases and variants for object-centric event data,” in *ICPM*, IEEE, 2022, pp. 128–135 (cit. on p. 23).
 - [121] W. M. P. van der Aalst, “Learning colored petri nets using object-centric event data (OCED2CPN),” in *CiSt*, IEEE, 2023, pp. 1–6 (cit. on p. 23).
 - [122] A. Berti and W. M. P. van der Aalst, “OC-PM: analyzing object-centric event logs and process models,” *Int. J. Softw. Tools Technol. Transf.*, vol. 25, no. 1, pp. 1–17, 2023 (cit. on p. 23).
 - [123] A. Berti, I. Koren, J. N. Adams, *et al.*, “OCEL (object-centric event log) 2.0 specification,” *CoRR*, vol. abs/2403.01975, 2024 (cit. on p. 23).
 - [124] J. D. Smedt, S. K. L. M. vanden Broucke, J. Obregon, A. Kim, J. Jung, and J. Vanthienen, “Decision mining in a broader context: An overview of the current landscape and future directions,” in *Business Process Management Workshops*, ser. Lecture Notes in Business Information Processing, vol. 281, 2016, pp. 197–207 (cit. on p. 25).
 - [125] K. Diba, K. Batoulis, M. Weidlich, and M. Weske, “Extraction, correlation, and abstraction of event data for process mining,” *WIREs Data Mining Knowl. Discov.*, vol. 10, no. 3, 2020 (cit. on p. 25).
 - [126] A. E. M. Chamorro, K. Revoredo, M. Resinas, A. del-Río-Ortega, F. M. Santoro, and A. Ruiz-Cortés, “Context-aware process performance indicator prediction,” *IEEE Access*, vol. 8, pp. 222 050–222 063, 2020 (cit. on p. 25).
 - [127] R. Y. Wang and D. M. Strong, “Beyond accuracy: What data quality means to data consumers,” *J. Manag. Inf. Syst.*, vol. 12, no. 4, pp. 5–33, 1996 (cit. on pp. 26, 167).
 - [128] Y. Wand and R. Y. Wang, “Anchoring data quality dimensions in ontological foundations,” *Commun. ACM*, vol. 39, no. 11, pp. 86–95, 1996 (cit. on pp. 26, 167).
 - [129] J. C. J. C. Bose, R. S. Mans, and W. M. P. van der Aalst, “Wanna improve process mining results?” In *CIDM*, IEEE, 2013, pp. 127–134 (cit. on p. 26).
 - [130] R. Eden, R. Syed, T. Makasi, *et al.*, “Revealing the root causes of digital health data quality issues: A qualitative investigation of the odigos framework,” in *HICSS*, ScholarSpace, 2023, pp. 3035–3044 (cit. on p. 26).
-

-
- [131] K. Goel, S. Sadeghianasl, R. Andrews, *et al.*, “Digital health data imperfection patterns and their manifestations in an australian digital hospital,” in *HICSS*, ScholarSpace, 2023, pp. 3235–3244 (cit. on p. 26).
 - [132] S. Suriadi, R. Andrews, A. H. M. ter Hofstede, and M. T. Wynn, “Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs,” *Inf. Syst.*, vol. 64, pp. 132–150, 2017 (cit. on pp. 26, 167).
 - [133] R. Andrews, C. G. J. van Dun, M. T. Wynn, W. Kratsch, M. Röglinger, and A. H. M. ter Hofstede, “Quality-informed semi-automated event log generation for process mining,” *Decis. Support Syst.*, vol. 132, p. 113 265, 2020 (cit. on pp. 26, 167).
 - [134] A. Senderovich, C. D. Francescomarino, and F. M. Maggi, “From knowledge-driven to data-driven inter-case feature encoding in predictive process monitoring,” *Inf. Syst.*, vol. 84, pp. 255–264, 2019 (cit. on pp. 27, 43).
 - [135] H. Nguyen, M. Dumas, M. L. Rosa, and A. H. M. ter Hofstede, “Multi-perspective comparison of business process variants based on event logs,” in *ER*, ser. Lecture Notes in Computer Science, vol. 11157, Springer, 2018, pp. 449–459 (cit. on p. 27).
 - [136] J. Cremerius and M. Weske, “Supporting domain data selection in data-enhanced process models,” in *Wirtschaftsinformatik*, AISEL, 2022 (cit. on pp. 27, 43).
 - [137] M. Dees, B. Hompes, and W. M. P. van der Aalst, “Events put into context (epic),” in *ICPM*, IEEE, 2020, pp. 65–72 (cit. on p. 27).
 - [138] S. J. J. Leemans, K. Goel, and S. J. van Zelst, “Using multi-level information in hierarchical process mining: Balancing behavioural quality and model complexity,” in *ICPM*, IEEE, 2020, pp. 137–144 (cit. on pp. 27, 43).
 - [139] R. Shraga, A. Gal, D. Schumacher, A. Senderovich, and M. Weidlich, “Inductive context-aware process discovery,” in *ICPM*, IEEE, 2019, pp. 33–40 (cit. on pp. 27, 32, 43).
 - [140] F. de Luzi, F. Leotta, A. Marrella, and M. Mecella, “On the interplay between business process management and internet-of-things,” *Bus. Inf. Syst. Eng.*, 2024 (cit. on pp. 28, 43).
 - [141] S. Agostinelli, F. de Luzi, U. di Canito, J. Ferraro, A. Marrella, and M. Mecella, “A data-centric approach to design resilient-aware process models in BPMN,” in *BPM (Forum)*, ser. Lecture Notes in Business Information Processing, vol. 458, Springer, 2022, pp. 38–54 (cit. on pp. 28, 29, 43).
 - [142] Y. Bertrand, R. V. Belle, J. de Weerd, and E. Serral, “Defining data quality issues in process mining with iot data,” in *ICPM Workshops*, ser. Lecture Notes in Business Information Processing, vol. 468, Springer, 2022, pp. 422–434 (cit. on p. 28).
 - [143] Y. Bertrand, S. Veneruso, F. Leotta, M. Mecella, and E. Serral, “NICE: the native iot-centric event log model for process mining,” in *ICPM Workshops*, ser. Lecture Notes in Business Information Processing, vol. 503, Springer, 2023, pp. 32–44 (cit. on p. 28).
-

- [144] Y. Bertrand, J. de Weerd, and E. Serral, “An expert-validated bridging model for iot process mining,” *Bus. Inf. Syst. Eng.*, 2024 (cit. on pp. 28, 43).
 - [145] C. Li, A. Joshi, N. T. L. Tam, *et al.*, “Rectify sensor data in iot: A case study on enabling process mining for logistic process in an air cargo terminal,” in *CoopIS*, ser. Lecture Notes in Computer Science, vol. 14353, Springer, 2023, pp. 293–310 (cit. on p. 28).
 - [146] J. Grüger, L. Malburg, and R. Bergmann, “Iot-enriched event log generation and quality analytics: A case study,” *it Inf. Technol.*, vol. 65, no. 3, pp. 128–138, 2023 (cit. on p. 28).
 - [147] F. Mannhardt, M. de Leoni, H. A. Reijers, W. M. P. van der Aalst, and P. J. Toussaint, “Guided process discovery - A pattern-based approach,” *Inf. Syst.*, vol. 76, pp. 1–18, 2018 (cit. on pp. 28, 29).
 - [148] R. D. Fazio, A. Balzanella, S. Marrone, *et al.*, “Caseid detection for process mining: A heuristic-based methodology,” in *ICPM Workshops*, ser. Lecture Notes in Business Information Processing, vol. 503, Springer, 2023, pp. 45–57 (cit. on p. 29).
 - [149] H. H. Beyel, O. Makke, O. Gusikhin, and W. M. P. van der Aalst, “Analyzing behavior in cyber-physical systems in connected vehicles: A case study,” in *Business Process Management Workshops*, ser. Lecture Notes in Business Information Processing, vol. 492, Springer, 2023, pp. 92–104 (cit. on p. 29).
 - [150] M. de Leoni and L. Pellattiero, “The benefits of sensor-measurement aggregation in discovering iot process models: A smart-house case study,” in *Business Process Management Workshops*, ser. Lecture Notes in Business Information Processing, vol. 436, Springer, 2021, pp. 403–415 (cit. on p. 29).
 - [151] D. Janssen, F. Mannhardt, A. Koschmider, and S. J. van Zelst, “Process model discovery from sensor event data,” in *ICPM Workshops*, ser. Lecture Notes in Business Information Processing, vol. 406, Springer, 2020, pp. 69–81 (cit. on p. 29).
 - [152] A. Rebmann, J. Rehse, M. Pinter, M. Schnaubelt, K. Daun, and P. Fettke, “Iot-based activity recognition for process assistance in human-robot disaster response,” in *BPM (Forum)*, ser. Lecture Notes in Business Information Processing, vol. 392, Springer, 2020, pp. 71–87 (cit. on p. 29).
 - [153] P. Rashidi, G. M. Youngblood, D. J. Cook, and S. K. Das, “Inhabitant guidance of smart environments,” in *HCI (2)*, ser. Lecture Notes in Computer Science, vol. 4551, Springer, 2007, pp. 910–919 (cit. on p. 29).
 - [154] L. Malburg, R. Seiger, R. Bergmann, and B. Weber, “Using physical factory simulation models for business process management research,” in *Business Process Management Workshops*, ser. Lecture Notes in Business Information Processing, vol. 397, Springer, 2020, pp. 95–107 (cit. on p. 29).
-

- [155] F. Weyers, R. Seiger, and B. Weber, “Method to identify process activities by visualizing sensor events,” in *Business Process Management Workshops*, ser. Lecture Notes in Business Information Processing, vol. 460, Springer, 2022, pp. 76–88 (cit. on p. 29).
 - [156] Y. Bertrand, J. D. Weerd, and E. Serral, “A novel multi-perspective trace clustering technique for iot-enhanced processes: A case study in smart manufacturing,” in *BPM*, ser. Lecture Notes in Computer Science, vol. 14159, Springer, 2023, pp. 395–412 (cit. on p. 29).
 - [157] C. Li, A. Joshi, N. T. L. Tam, *et al.*, “Rectify sensor data in iot: A case study on enabling process mining for logistic process in an air cargo terminal,” in *CoopIS*, ser. Lecture Notes in Computer Science, vol. 14353, Springer, 2023, pp. 293–310 (cit. on p. 29).
 - [158] W. Reisig and G. Rozenberg, Eds., *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets, the volumes are based on the Advanced Course on Petri Nets, held in Dagstuhl, September 1996*, vol. 1491, Lecture Notes in Computer Science, Springer, 1998 (cit. on p. 30).
 - [159] W. M. P. van der Aalst, “Everything you always wanted to know about petri nets, but were afraid to ask,” in *BPM*, ser. Lecture Notes in Computer Science, vol. 11675, Springer, 2019, pp. 3–9 (cit. on pp. 30, 31).
 - [160] A. Küsters and W. M. P. van der Aalst, “Revisiting the alpha algorithm to enable real-life process discovery applications,” in *ATAED/PN4TT@Petri Nets*, ser. CEUR Workshop Proceedings, vol. 3424, CEUR-WS.org, 2023 (cit. on p. 31).
 - [161] J. N. van Detten, P. Schumacher, and S. J. J. Leemans, “An approximate inductive miner,” in *ICPM*, IEEE, 2023, pp. 129–136 (cit. on p. 32).
 - [162] M. Pesic, H. Schonenberg, and W. M. P. van der Aalst, “DECLARE: full support for loosely-structured processes,” in *EDOC*, IEEE Computer Society, 2007, pp. 287–300 (cit. on p. 34).
 - [163] W. M. van der Aalst, “Process mining: Discovering and improving spaghetti and lasagna processes,” in *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, 2011, pp. 1–7 (cit. on p. 34).
 - [164] C. D. Ciccio and M. Mecella, “On the discovery of declarative control flows for artful processes,” *ACM Trans. Manag. Inf. Syst.*, vol. 5, no. 4, 24:1–24:37, 2015 (cit. on p. 34).
 - [165] F. M. Maggi, C. D. Ciccio, C. D. Francescomarino, and T. Kala, “Parallel algorithms for the automated discovery of declarative process models,” *Inf. Syst.*, vol. 74, no. Part, pp. 136–152, 2018 (cit. on p. 34).
 - [166] S. Schöning, C. D. Ciccio, F. M. Maggi, and J. Mendling, “Discovery of multi-perspective declarative process models,” in *ICSOC*, ser. Lecture Notes in Computer Science, vol. 9936, Springer, 2016, pp. 87–103 (cit. on p. 34).
-

- [167] V. Leno, M. Dumas, F. M. Maggi, M. L. Rosa, and A. Polyvyanyy, “Automated discovery of declarative process models with correlated data conditions,” *Inf. Syst.*, vol. 89, p. 101 482, 2020 (cit. on p. [34](#)).
 - [168] M. Weidlich, A. Polyvyanyy, N. Desai, J. Mendling, and M. Weske, “Process compliance analysis based on behavioural profiles,” *Inf. Syst.*, vol. 36, no. 7, pp. 1009–1025, 2011 (cit. on pp. [35](#), [37](#)).
 - [169] M. de Leoni and W. M. P. van der Aalst, “Aligning event logs and process models for multi-perspective conformance checking: An approach based on integer linear programming,” in *BPM*, ser. Lecture Notes in Computer Science, vol. 8094, Springer, 2013, pp. 113–129 (cit. on pp. [35](#), [37–39](#), [43](#), [59](#), [125](#), [126](#), [129](#), [153](#), [195](#), [196](#)).
 - [170] F. Mannhardt, M. de Leoni, H. A. Reijers, and W. M. P. van der Aalst, “Balanced multi-perspective checking of process conformance,” *Computing*, vol. 98, no. 4, pp. 407–437, 2016 (cit. on pp. [35](#), [37–39](#), [41](#), [43](#), [59](#), [66](#), [125–129](#), [153](#), [157](#), [159](#), [195–197](#)).
 - [171] B. F. van Dongen, “Efficiently computing alignments - using the extended marking equation,” in *BPM*, ser. Lecture Notes in Computer Science, vol. 11080, Springer, 2018, pp. 197–214 (cit. on pp. [35](#), [129](#)).
 - [172] D. Sommers, N. Sidorova, and B. F. van Dongen, “Exact and approximated log alignments for processes with inter-case dependencies,” in *Petri Nets*, ser. Lecture Notes in Computer Science, vol. 13929, Springer, 2023, pp. 99–119 (cit. on pp. [35](#), [129](#)).
 - [173] P. Felli, A. Gianola, M. Montali, A. Rivkin, and S. Winkler, “Data-aware conformance checking with SMT,” *Inf. Syst.*, vol. 117, p. 102 230, 2023 (cit. on pp. [35](#), [37–39](#), [43](#), [125](#), [128](#), [129](#), [153](#), [157](#), [159](#), [196](#)).
 - [174] E. Verbeek and W. M. P. van der Aalst, “Decomposing replay problems: A case study,” in *PNSE+ModPE*, ser. CEUR Workshop Proceedings, vol. 989, CEUR-WS.org, 2013, pp. 219–235 (cit. on p. [36](#)).
 - [175] H. M. W. Verbeek and W. M. P. van der Aalst, “Merging alignments for decomposed replay,” in *Petri Nets*, ser. Lecture Notes in Computer Science, vol. 9698, Springer, 2016, pp. 219–239 (cit. on p. [36](#)).
 - [176] A. Berti and W. M. P. van der Aalst, “Reviving token-based replay: Increasing speed while improving diagnostics,” in *ATAED@Petri Nets/ACSD*, ser. CEUR Workshop Proceedings, vol. 2371, CEUR-WS.org, 2019, pp. 87–103 (cit. on p. [36](#)).
 - [177] A. Augusto, A. Armas-Cervantes, R. Conforti, M. Dumas, and M. L. Rosa, “Measuring fitness and precision of automatically discovered process models: A principled and scalable approach,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 4, pp. 1870–1888, 2022 (cit. on pp. [37](#), [125](#)).
 - [178] P. Felli, M. Montali, and S. Winkler, “Repairing soundness properties in data-aware processes,” in *ICPM*, IEEE, 2023, pp. 41–48 (cit. on pp. [37](#), [38](#), [40](#)).
-

-
- [179] P. Zerbino, A. Stefanini, and D. Aloini, “Process science in action: A literature review on process mining in business management,” *Technological Forecasting and Social Change*, vol. 172, p. 121 021, 2021, issn: 0040-1625 (cit. on pp. 39, 43).
- [180] T. G. Erdogan and A. Tarhan, “Systematic mapping of process mining studies in health-care,” *IEEE Access*, vol. 6, pp. 24 543–24 567, 2018 (cit. on p. 39).
- [181] F. Taymouri, M. L. Rosa, M. Dumas, and F. M. Maggi, “Business process variant analysis: Survey and classification,” *Knowl. Based Syst.*, vol. 211, p. 106 557, 2021 (cit. on p. 39).
- [182] M. Imran, S. Hamid, and M. A. Ismail, “Advancing process audits with process mining: A systematic review of trends, challenges, and opportunities,” *IEEE Access*, vol. 11, pp. 68 340–68 357, 2023 (cit. on p. 39).
- [183] A. Burke, S. J. J. Leemans, and M. T. Wynn, “Stochastic process discovery by weight estimation,” in *ICPM Workshops*, ser. Lecture Notes in Business Information Processing, vol. 406, Springer, 2020, pp. 260–272 (cit. on pp. 40, 41, 82, 83, 89, 93, 117).
- [184] S. J. J. Leemans, L. L. Mannel, and N. Sidorova, “Significant stochastic dependencies in process models,” *Inf. Syst.*, vol. 118, p. 102 223, 2023 (cit. on pp. 40, 41, 45, 82, 83, 90, 93, 100, 117, 118).
- [185] M. de Leoni, P. Felli, and M. Montali, “A holistic approach for soundness verification of decision-aware process models,” in *ER*, ser. Lecture Notes in Computer Science, vol. 11157, Springer, 2018, pp. 219–235 (cit. on pp. 40, 53, 127).
- [186] A. Bolt, M. de Leoni, and W. M. P. van der Aalst, “Process variant comparison: Using event logs to detect differences in behavior and business rules,” *Inf. Syst.*, vol. 74, no. Part, pp. 53–66, 2018 (cit. on p. 41).
- [187] B. Wais and S. Rinderle-Ma, “Towards a comprehensive evaluation of decision rules and decision mining algorithms beyond accuracy,” in *Advanced Information Systems Engineering*, Cham: Springer Nature Switzerland, 2024, pp. 403–419 (cit. on p. 41).
- [188] T. Jouck, M. de Leoni, and B. Depaire, “A framework to evaluate and compare decision-mining techniques,” in *Business Process Management Workshops*, ser. Lecture Notes in Business Information Processing, vol. 342, Springer, 2018, pp. 482–493 (cit. on p. 41).
- [189] F. Mannhardt, M. de Leoni, H. A. Reijers, and W. M. P. van der Aalst, “Measuring the precision of multi-perspective process models,” in *Business Process Management Workshops*, ser. Lecture Notes in Business Information Processing, vol. 256, Springer, 2015, pp. 113–125 (cit. on pp. 41, 66, 125–128, 153, 157, 159, 196, 197).
- [190] A. Brand, L. Allen, M. Altman, M. M. K. Hlava, and J. Scott, “Beyond authorship: Attribution, contribution, collaboration, and credit,” *Learn. Publ.*, vol. 28, no. 2, pp. 151–155, 2015 (cit. on pp. 47, 48, 80, 123).
-

- [191] G. Ibanez-Sanchez, C. Fernandez-Llatas, A. Martinez-Millana, *et al.*, “Toward value-based healthcare through interactive process mining in emergency rooms: The stroke case,” *International Journal of Environmental Research and Public Health*, vol. 16, no. 10, p. 1783, 2019 (cit. on p. 49).
 - [192] R. C. Basole, H. Park, M. Gupta, M. L. Braunstein, D. H. Chau, and M. Thompson, “A visual analytics approach to understanding care process variation and conformance,” in *VAHC*, ACM, 2015, 6:1–6:8 (cit. on p. 49).
 - [193] H. S. G. Caballero, A. Corvò, P. M. Dixit, and M. A. Westenberg, “Visual analytics for evaluating clinical pathways,” in *VAHC*, IEEE, 2017, pp. 39–46 (cit. on p. 49).
 - [194] R. A. Q. Neira, B. F. A. Hompes, G. de Vries, *et al.*, “Analysis and optimization of a sepsis clinical pathway using process mining,” in *Business Process Management Workshops*, ser. Lecture Notes in Business Information Processing, vol. 362, Springer, 2019, pp. 459–470 (cit. on pp. 49, 50).
 - [195] E. Rojas and D. Capurro, “Characterization of drug use patterns using process mining and temporal abstraction digital phenotyping,” in *Business Process Management Workshops*, ser. Lecture Notes in Business Information Processing, vol. 342, Springer, 2018, pp. 187–198 (cit. on p. 49).
 - [196] G. de Vries, R. A. Q. Neira, G. Geleijnse, P. M. Dixit, and B. F. Mazza, “Towards process mining of EMR data - case study for sepsis management,” in *HEALTHINF*, SciTePress, 2017, pp. 585–593 (cit. on p. 49).
 - [197] H. D. Oliveira, M. Prodel, L. Lamarsalle, *et al.*, ““bow-tie” optimal pathway discovery analysis of sepsis hospital admissions using the hospital episode statistics database in england,” *JAMIA Open*, vol. 3, no. 3, pp. 439–448, 2020 (cit. on p. 49).
 - [198] A. Najjar, D. Reinharz, C. Girouard, and C. Gagné, “A two-step approach for mining patient treatment pathways in administrative healthcare databases,” *Artif. Intell. Medicine*, vol. 87, pp. 34–48, 2018 (cit. on p. 50).
 - [199] A. C. Apunike, L. Oliveira-Ciabati, T. L. M. Sanches, *et al.*, “Analyses of public health databases via clinical pathway modelling: TBWEB,” in *ICCS (4)*, ser. Lecture Notes in Computer Science, vol. 12140, Springer, 2020, pp. 550–562 (cit. on p. 50).
 - [200] E. D. Roock and N. Martin, “Process mining in healthcare - an updated perspective on the state of the art,” *J. Biomed. Informatics*, vol. 127, p. 103 995, 2022 (cit. on p. 50).
 - [201] E. Rojas, M. Sepúlveda, J. Munoz-Gama, D. Capurro, V. Traver, and C. Fernandez-Llatas, “Question-driven methodology for analyzing emergency room processes using process mining,” *Applied Sciences*, vol. 7, no. 3, p. 302, 2017 (cit. on p. 50).
 - [202] C. Alvarez, E. Rojas, M. Arias, *et al.*, “Discovering role interaction models in the emergency room using process mining,” *J. Biomed. Informatics*, vol. 78, pp. 60–77, 2018 (cit. on p. 50).
-

-
- [203] K. S. Cosby, R. Roberts, L. Palivos, *et al.*, “Characteristics of patient care management problems identified in emergency department morbidity and mortality investigations during 15 years,” *Ann. Emerg. Med.*, vol. 51, no. 3, pp. 251–261, 2008 (cit. on p. 50).
- [204] P. Felli, M. de Leoni, and M. Montali, “Soundness verification of decision-aware process models with variable-to-variable conditions,” in *ACSD*, IEEE, 2019, pp. 82–91 (cit. on pp. 53, 127, 156).
- [205] E. Gonzalez Lopez de Murillas, “Process mining on databases: Extracting event data from real-life data sources,” Ph.D. dissertation, Eindhoven University of Technology, Eindhoven, The Netherlands, 2019 (cit. on p. 57).
- [206] S. Chaudhuri and U. Dayal, “An overview of data warehousing and OLAP technology,” *SIGMOD Rec.*, vol. 26, no. 1, pp. 65–74, 1997 (cit. on p. 57).
- [207] F. Mannhardt, M. de Leoni, and H. A. Reijers, “The multi-perspective process explorer,” in *BPM (Demos)*, ser. CEUR Workshop Proceedings, vol. 1418, CEUR-WS.org, 2015, pp. 130–134 (cit. on p. 58).
- [208] S. M. Lundberg and S. Lee, “A unified approach to interpreting model predictions,” in *NIPS*, 2017, pp. 4765–4774 (cit. on p. 59).
- [209] M. T. Ribeiro, S. Singh, and C. Guestrin, ““why should I trust you?”: Explaining the predictions of any classifier,” in *KDD*, ACM, 2016, pp. 1135–1144 (cit. on p. 59).
- [210] C. Moreira, Y. Chou, M. Velmurugan, C. Ouyang, R. Sindhgatta, and P. Bruza, “LINDA-BN: an interpretable probabilistic approach for demystifying black-box predictive models,” *Decis. Support Syst.*, vol. 150, p. 113 561, 2021 (cit. on p. 59).
- [211] R. Chattamvelli and R. Shanmugam, *Statistics for Scientists and Engineers*. WILEY, 2015, ISBN: 1118228960 (cit. on p. 61).
- [212] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978 (cit. on p. 63).
- [213] S. R. Aghabozorgi, A. S. Shirkhorshidi, and Y. W. Teh, “Time-series clustering - A decade review,” *Inf. Syst.*, vol. 53, pp. 16–38, 2015 (cit. on p. 63).
- [214] D. Griffin and J. Lim, “Signal estimation from modified short-time fourier transform,” *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984 (cit. on p. 66).
- [215] K. Reinhart, R. Daniels, N. Kissoon, F. R. Machado, R. D. Schachter, and S. Finfer, “Recognizing sepsis as a global health priority — a who resolution,” *N. Engl. J. Med.*, vol. 377, no. 5, pp. 414–417, 2017 (cit. on p. 72).
-

- [216] L. J. Schlapbach, N. Kissoon, A. Alhawsawi, *et al.*, “World sepsis day: A global agenda to target a leading cause of morbidity and mortality,” *American Journal of Physiology-Lung Cellular and Molecular Physiology*, vol. 319, no. 3, pp. L518–L522, 2020 (cit. on p. 72).
 - [217] K. Kawamoto, C. A. Houlihan, E. A. Balas, and D. F. Lobach, “Improving clinical practice using clinical decision support systems: A systematic review of trials to identify features critical to success,” *BMJ*, vol. 330, no. 7494, p. 765, 2005 (cit. on p. 73).
 - [218] D. F. Sittig, A. Wright, J. A. Osheroff, *et al.*, “Grand challenges in clinical decision support,” *J. Biomed. Informatics*, vol. 41, no. 2, pp. 387–392, 2008 (cit. on p. 73).
 - [219] J. Varghese, M. Kleine, S. I. Gessner, S. Sandmann, and M. Dugas, “Effects of computerized decision support system implementations on patient outcomes in inpatient care: A systematic review,” *J. Am. Medical Informatics Assoc.*, vol. 25, no. 5, pp. 593–602, 2018 (cit. on p. 73).
 - [220] R. S. Mans, H. Schonenberg, M. Song, W. M. P. van der Aalst, and P. J. M. Bakker, “Application of process mining in healthcare - A case study in a dutch hospital,” in *BIOSTEC (Selected Papers)*, ser. Communications in Computer and Information Science, vol. 25, Springer, 2008, pp. 425–438 (cit. on p. 73).
 - [221] A. Holzinger, “Interactive machine learning for health informatics: When do we need the human-in-the-loop?” *Brain Informatics*, vol. 3, no. 2, pp. 119–131, 2016 (cit. on p. 73).
 - [222] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to time series analysis and forecasting* (Wiley Series in Probability and Statistics), eng, Second edition. Hoboken, New Jersey: Wiley, 2015 (cit. on pp. 78, 167).
 - [223] A. E. Johnson, L. Bulgarelli, L. Shen, *et al.*, “MIMIC-IV, a freely accessible electronic health record dataset,” *Scientific Data*, vol. 10, no. 1, 2023 (cit. on p. 79).
 - [224] S. Rinderle, M. Reichert, and P. Dadam, “Correctness criteria for dynamic changes in workflow systems - a survey,” *Data Knowl. Eng.*, vol. 50, no. 1, pp. 9–34, 2004 (cit. on p. 81).
 - [225] C. W. Günther, S. Rinderle-Ma, M. Reichert, W. M. P. van der Aalst, and J. Recker, “Using process mining to learn from process changes in evolutionary systems,” *Int. J. Bus. Process. Integr. Manag.*, vol. 3, no. 1, pp. 61–78, 2008 (cit. on p. 81).
 - [226] A. Rogge-Solti, W. M. P. van der Aalst, and M. Weske, “Discovering stochastic petri nets with arbitrary delay distributions from event logs,” in *Business Process Management Workshops*, ser. Lecture Notes in Business Information Processing, vol. 171, Springer, 2013, pp. 15–27 (cit. on p. 82).
 - [227] L. Malburg, J. Gröger, and R. Bergmann, “An iot-enriched event log for process mining in smart factories,” *Zendo*, 2023 (cit. on pp. 93, 114).
-

-
- [228] S. J. J. Leemans, W. M. P. van der Aalst, T. Brockhoff, and A. Polyvyanyy, “Stochastic process mining: Earth movers’ stochastic conformance,” *Inf. Syst.*, vol. 102, p. 101 724, 2021 (cit. on p. 125).
- [229] M. de Leoni, M. Dumas, and L. García-Bañuelos, “Discovering branching conditions from business process execution logs,” in *FASE*, ser. Lecture Notes in Computer Science, vol. 7793, Springer, 2013, pp. 114–129 (cit. on p. 127).
- [230] M. de Leoni and F. Mannhardt, “Decision discovery in business processes,” in *Encyclopedia of Big Data Technologies*, Springer, 2019 (cit. on p. 128).
- [231] S. Leewis, K. Smit, and M. Zoet, “Putting decision mining into context: A literature study,” in *Digital Business Transformation*, R. Agrifoglio, R. Lamboglia, D. Mancini, and F. Ricciardi, Eds., Cham: Springer, 2020, pp. 31–46 (cit. on p. 128).
- [232] S. Lindell, “A logspace algorithm for tree canonization (extended abstract),” in *STOC*, ACM, 1992, pp. 400–404 (cit. on p. 133).
- [233] H. Alkhamash, A. Polyvyanyy, A. Moffat, and L. García-Bañuelos, “Entropic relevance: A mechanism for measuring stochastic process models discovered from event data,” *Inf. Syst.*, vol. 107, p. 101 922, 2022 (cit. on p. 148).
- [234] W. M. P. van der Aalst, *Rainbow spaghetti, metro maps, and object-centric process mining*. [Online]. Available: <https://vdaalst.com/publications/p1434.pdf> (cit. on p. 169).
- [235] W. M. P. van der Aalst, “Object-centric process mining: Unraveling the fabric of real processes,” *Mathematics*, vol. 11, no. 12, 2023, ISSN: 2227-7390. doi: 10.3390/math11122691 (cit. on p. 170).
- [236] B. F. van Dongen, A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and W. M. P. van der Aalst, “The prom framework: A new era in process mining tool support,” in *ICATPN*, ser. Lecture Notes in Computer Science, vol. 3536, Springer, 2005, pp. 444–454 (cit. on p. 200).
- [237] W. M. P. van der Aalst, B. F. van Dongen, C. W. Günther, A. Rozinat, E. Verbeek, and T. Weijters, “Prom: The process mining toolkit,” in *BPM (Demos)*, ser. CEUR Workshop Proceedings, vol. 489, CEUR-WS.org, 2009 (cit. on p. 200).
- [238] E. Verbeek, J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, “Prom 6: The process mining toolkit,” in *BPM (Demos)*, ser. CEUR Workshop Proceedings, vol. 615, CEUR-WS.org, 2010 (cit. on p. 200).
-

Biography

Dr Adam Banham was born on November 8, 1991, in the outer suburbs of Brisbane, Queensland, Australia, and has been immersed in the world of processes for some time. Although the process that led to the completion of this thesis has been ongoing for some time, it included many choices often influenced by the context surrounding the process and, across the way, led to significant personal development and growth. The start of this process began with a bachelor of design, focusing on architecture, which I did not complete. I had a creative mindset but not that type of creativity. To some extent, my living situation significantly hindered my ability to focus on learning. Having to travel for two and a half hours each way and only having one meal rarely allows one to focus on anything but survival. However, what I did learn is that my cure for boredom was learning, tinkering, and investigating.

Throughout this journey towards completing this thesis, I have refined my understanding of how to excel at learning. I present my findings around learning through a metaphor: consider pouring water into a cup. The cup represents the context surrounding the learner; many things influence the shape of the cup, how tall the cup is, how well made it is to capture water, and how many holes exist in the cup. During my first attempt at university, my cup was filled with holes and was quickly made so I could get back home in time to make money for tomorrow's meal. The flow of water into the cup represents access to mentors or teachers for the subject matter. Many things influence the flow of the water, such as whether you are allowed to use the tap it flows from and whether the water is contaminated or not. Given the cup I had at my first attempt at university, it made little difference how much I focused on my studies; I couldn't capture a word of the many mentors and teachers who tried to fill it. After this experience and some hard lessons about how little one can do without an education, I started to learn to make better cups.

Making a metaphor is easy, and having an idea may also come just as easy; however, changing one's living conditions to support these ideas always comes with challenging choices and difficult solutions. For the next five years, I would endure working as a store manager and merchandiser for convenience stores. I would work late into the night, being held up several times at gunpoint, to make enough to support my siblings and my single mother. Why endure? Well, these types of jobs often are long hours with little agency over what to do and cleaning up the mistakes of others. I also gained my first experience with processes (unknown to me at the time) and ad-hoc decision making in these positions. I saw firsthand the difference between

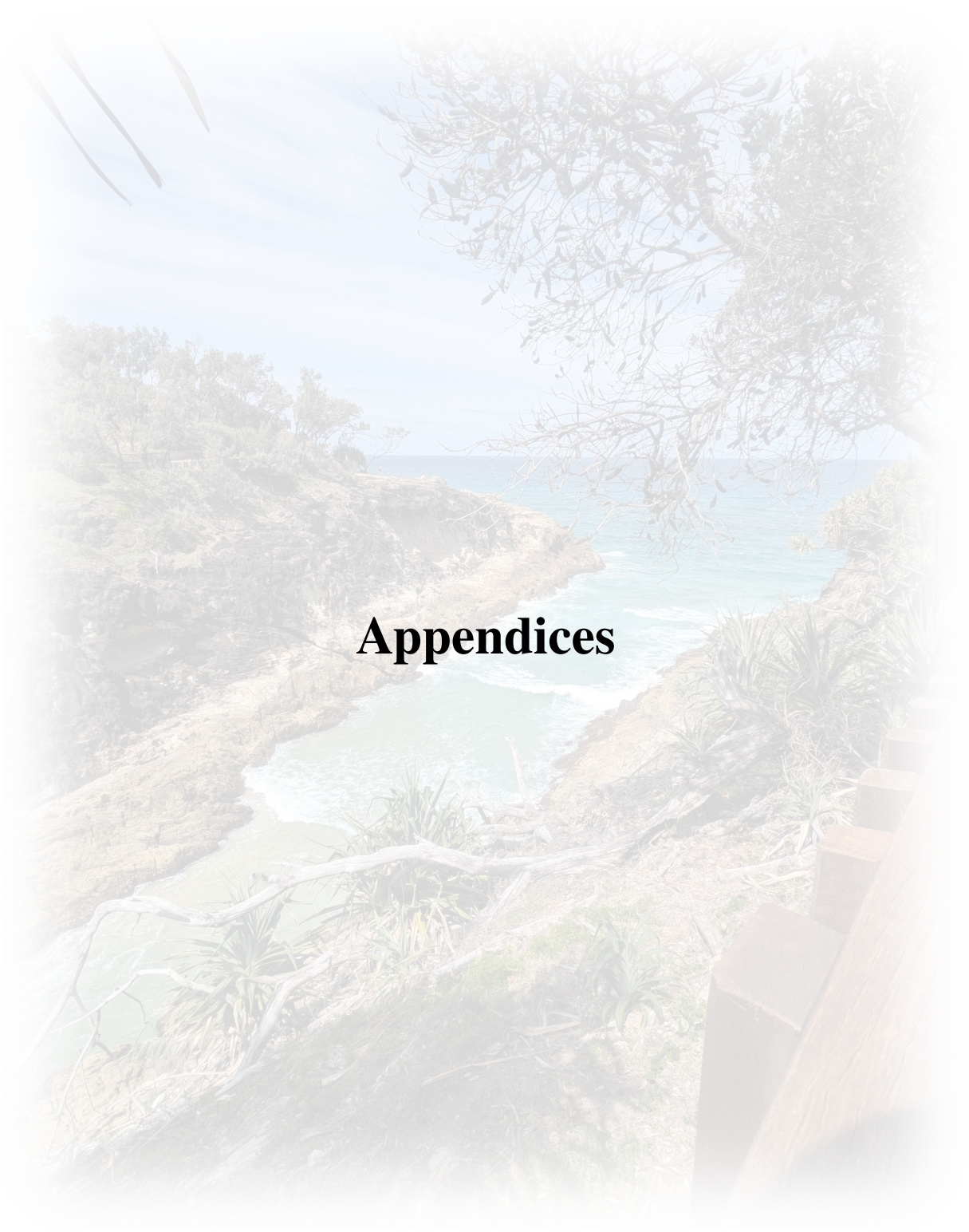
evidence-based and instinct-based decision making. The latter often yields huge wins once or twice, while the former builds a worldwide network of stores by making thousands of small wins and ensuring a constant cash flow. The challenge of a convenience store is its appearance; people don't want to come to a store that is unclean, unsafe, or does not have fresh items. If people don't come to the store, items stay on the shelves too long, and the store slowly becomes a shady place to visit. All you can do is clean and hope people come in to look at your goods. If a store sours, it often experiences a slow, irreversible decline, often at the expense of the franchisee owner, regardless of the decision making that follows.

A choice arose when all my siblings had finished school, and I no longer needed to do the school run each day after working until 1 or 2 in the morning. What should I do with my life? For too long, it had been about making ends meet and rotting in place to make them meet. I made a promise to myself to never stay in such a position for any amount of time again; if I wasn't learning or couldn't see how I would learn, it was time to leave. Now, with better living conditions, I returned to pursue my bachelor's degree in information technology, progressing to honours, which would ultimately lead to this thesis almost a decade later. I do not regret the choices I made to get to this point. In fact, they were the only ones I could have made to be able to look at myself in the mirror today. However, in reflection, if I had obtained a university degree in computer science or information systems (as opposed to architecture) right the first time around, it would have been a dramatic and impactful advantage.

So, what was the significant personal development and growth that occurred during the completion of this thesis? It wasn't that I had some undiscovered talent, had suddenly become "smart", or had achieved the impossible¹. The process of completing a thesis and obtaining a PhD had taught me to be an expert in learning through long hours of hard work. Without this mastery of learning, I could never compete with those who have talent, far greater resources, better-planned upbringings, or were geniuses (they do exist!) in their subject matter. I had become a sponge that absorbed the surrounding context, improving my decision making to grow exponentially. I greedily threw my cups towards the many wonderful souls at the Queensland University of Technology, who gracefully filled them up to the brim through their mentorship. I owe the rate of my growth to the mentorship I have received, I owe my passion for processes to my supervisors, I owe the timing of the process to the many choices I had to settle, and I owe the bliss I have derived from the process to my wretchedness². **Embrace your wretchedness!** At worst, you make a thousand small steps towards your life goals and learn a million things to grow into tomorrow; at best, you confidently choose the best possible thousand steps to take and wish you had more time to learn another thousand steps.

¹ Roughly 206,000 (36% increase) people had obtained a PhD in 2021 census of Australia, compared to roughly 151,000 in 2016, see www.abs.gov.au/statistics/people/education/education-and-training-census/2021.

² The definition of 'bliss' is irrelevant and whatever it is, it will come from the efforts of the wretched, individuals or organisations, improving their behaviour to obtain more of it.



Conformance Checking

In this section, we outline existing data-aware conformance checking techniques that were evaluated in Chapter 5. We also present proof sketches to affirm that our proposed measures, guard-recall and guard-precision, adhere to the presented axioms in Section 5.6.

A.1 Existing measures

We now discuss existing data-aware techniques using the theory and notation presented in Section 5.4.

A.1.1 Guard-Recall Techniques

The first existing technique in the literature, proposed by de Leoni and van der Aalst [169], extends the classical alignment by using ILP to compute a data assignment matching the initial control-flow alignment. However, the original implementation of this technique is no longer available, thus we only consider the subsequent continuation of this work in [170] for our evaluation, which ensures that an optimal alignment is always produced.

The first technique considered in our evaluation, (grec_M), was proposed by Mannhardt, de Leoni, Reijers, *et al.* [170], and uses MILP to solve for an holistic alignment between a given trace and DPN. The approach applies a cost function ($\mathcal{K}$) which accounts for misalignments in the data-perspective, as well as the control-flow in a fully customisable way (i.e. the weights can be changed to penalise any one perspective more or less than the others). The formal computation of grec_M is noted below:

Definition 41 (grec_M as proposed by [170]) *Let L be a log, let M be a model, let P be the set of paths through M , let $\gamma_O : L \mapsto P$ be the computed optimal alignments using [170], let $\gamma_R : L \mapsto P$ be the worst reference alignments according to [170], and let $\mathcal{K} : (L \times P) \mapsto \mathbb{R}$ be the cost function with respect to a given cost-model. Hence, grec_M is defined as:*

$$\text{grec}_M(L, \gamma, M) = \frac{1}{|L|} \cdot \sum_{\sigma \in L} 1 - \frac{\mathcal{K}(\sigma, \gamma_O(\sigma))}{\mathcal{K}(\sigma, \gamma_R(\sigma))}.$$

Note that for our testing, we used the default cost-model for this technique where a log-move costs 1, a model-move costs 2, and any data-move costs 3.

The second candidate, (grec_F), uses the theory proposed in Felli, Gianola, Montali, *et al.* [173] to formulate a potential recall measurement. In [173], the CoCoMoT framework is used to compute multi-alignments in the a similar vein to [169], [170]. However, their work focuses exclusively on an alignment problem and, unlike the already described techniques [169], [170], does not present a fitness/recall measure. Thus, we propose a metric using their theory for our evaluation.

To describe grec_F (from [173]), we take advantage of the standard cost function to convert an outcome from this technique into a numeric. However, unlike other already described techniques, we report the weighted average of edits needed without a normalisation factor. Hence, grec_F will range between $[-\infty, 0]$, where 0 means that on average no edits were required to make traces compliant with the given DPN. The formal computation of a recall value derivable from [173] is noted below:

Definition 42 (Proposed grec_F using [173]) *Let L be a log, let M be a model, let P be the set of paths through M , let $C \subset L$ be the set of trace classes computed by [173], let $\text{rep} : L \mapsto C$ be the representative selection function for each class, let $\gamma_O : C \mapsto P$ be the optimal alignments for each class computed by [173], and let $\mathcal{K} : (C \times P) \mapsto [0, \infty]$ be the standard cost function in [173]. Then, we propose grec_F is defined as:*

$$\text{grec}_F(L, \gamma, M) = \frac{1}{|L|} \cdot \sum_{\sigma \in C} -1 \cdot |\{\sigma' \in L \mid \text{rep}(\sigma') = \sigma\}| \cdot \mathcal{K}(\sigma, \gamma_O(\sigma)).$$

A.1.2 Guard-Precision Technique

We now introduce the only existing technique for guard-precision, gprec_M , proposed in Mannhardt, de Leoni, Reijers, *et al.* [189]. This technique computes a precision measurement by considering the ‘compliant’ version of the log, and computes a ratio between the observed and the possible behaviour.

To formally denote gprec_M , we need to introduce the notions of *possible* behaviour and *observed* behaviour. Firstly, we formally denote the possible behaviour for a given point in the model, as the set of possible next steps based on the assignments seen at that point.

Definition 43 (Possible Behaviour) *Let L be a log, let $M = (N, F, n_r, N_f, \text{src}, \text{tgt}, \text{gd}, \text{actf})$ be a model, and $\hat{L} \subset (F \times \Pi)$ be the transformed compliant log with respect to M computed by [170]. Then, $\text{pos} : F \mapsto \mathcal{P}(F \times \Pi)$ is a function yielding the set of possible behaviour, defined as:*

$$\text{pos}(f) = \{(f', \pi) \mid (f, \pi) \in \hat{L} \wedge f' \in \text{src}(f) \bullet \wedge \llbracket \text{gd}(f') \rrbracket(\pi)\}.$$

Now, we formally denote the observed behaviour for a given point in the model, as the set of recorded next steps based on the assignments seen at that point.

Definition 44 (Observed Behaviour) *Let L be a log, let $M = (N, F, n_r, N_f, \text{src}, \text{tgt}, \text{gd}, \text{actf})$ be a model, and $\hat{L} \subset (F \times \Pi)$ be the transformed compliant log with respect to M computed by*

[170]. Then, $\text{obs} : F \mapsto \mathcal{P}(F \times \Pi)$ is a function yielding the set of observed behaviour, defined as:

$$\text{obs}(f) = \{(f, \pi) \in \hat{L} \mid \llbracket \text{gd}(f) \rrbracket(\pi)\}.$$

Finally, we formally denote the candidate gprec_M as proposed by Mannhardt, de Leoni, Reijers, et al. [189] below:

Definition 45 (gprec_M as proposed by [189]) *Let L be a log, let $M = (N, F, n_r, N_f, \text{src}, \text{tgt}, \text{gd}, \text{actf})$ be a model, and $\hat{L} \subset (F \times \Pi)$ be the transformed compliant log with respect to M computed by [170]. Hence, gprec_M is defined as:*

$$\text{gprec}_M(L, \gamma, M) = \frac{\sum_{f \in F} |\text{obs}(f)|}{\sum_{f \in F} |\text{pos}(f)|}.$$

A.2 Proof Sketches

Axiom 3 states that guard-recall increases proportionally to the (true) traversal bookkeeping. Thus, between two isomorphic models, one log, and one matching, the model with larger (true) traversal bookkeeping yields a larger guard-recall.

For our proposed measure grec , Axiom 3 holds.

Reasoning. Based on Definition 38, our proposed measure yields a measurement based on weighted traversal bookkeeping function (Definition 37) in the numerator of the proposed ratio. Given that both of the log and matching are fixed, the bookkeeping function will yield a number that is proportional to the number of truthful guard evaluations for a given model. Hence, the model with larger (true) traversal bookkeeping will always yield a larger guard-recall.

Axiom 4 states that guard-recall only returns min when the (true) traversal bookkeeping is zero for a given model, log and matching.

For our proposed measure grec , Axiom 4 holds.

Reasoning. Based on Definition 38, our proposed measure will only return zero (min) when the weighted traversal bookkeeping function (Definition 37) returns zero. This bookkeeping only returns zero when no guard evaluations were true, in line with the axiom. Hence, our proposed measure only yields min when there were no guard evaluations that were true.

Axiom 5 states that guard-recall is negatively affected by the paths in a matching being non-optimal, in the context of a log, a model, an optimal and non-optimal matching.

For our proposed measure grec , Axiom 5 holds.

Reasoning. Based on Definition 38, our proposed measure is proportional to the weighted traversal bookkeeping (Definition 37). This bookkeeping is a weighted sum over guard evaluations, where the weight is based on the path in the matching being optimal. Optimal paths are given larger weights than non-optimal paths through the weight function (Definition 32). If the (true) traversal bookkeeping between matchings is the same, then our proposed measure grec is larger for the matching that is optimal as the bookkeeping will assign more weight to

the optimal case. If the non-optimal matching has strictly less guard evaluations that were true, then the weighted sum from the bookkeeping will be less than the optimal matching.

Axiom 6 states that guard-recall is not influenced by log enlargements.

For our proposed measure grec, Axiom 6 holds.

Reasoning. Based on Definition 38, our proposed measure is a ratio where the numerator and denominator are iterations over the traces in a log. As such, enlarging a log by cloning (Definition 18) will increase both numerator and denominator by a factor of k as the clones result in the same outcomes being counted k times. This factor k cancels out and the same ratio is returned regardless of the size of the factor.

Axiom 7 states that guard-precision is inversely proportional to (true) explorative bookkeeping (Equation 4.8 in Definition 34). Thus, guard-precision will be larger for a model that has a smaller sum over the explorative bookkeeping than another, for a given log and a matching.

For our proposed measure gprec, Axiom 7 holds.

Reasoning. Based on Definition 40, our proposed measure uses two bookkeeping functions, a weighted traversal function for the numerator and a weighted explorative function (Definition 39) for the denominator. The weighted explorative bookkeeping is proportional to the (true) explorative bookkeeping, thus if the explorative bookkeeping is larger, the denominator of our proposed ratio for gprec will be larger. As the denominator will always be at least as large as the numerator, our proposed ratio will be inversely proportional to the (true) explorative bookkeeping. Thus, a model will have a larger gprec than another, if the model has at least one flow where the explorative bookkeeping was smaller than its isomorphic reflection in the other model, reducing the size of the denominator in our proposed measure.

Axiom 8 states that guard-precision only returns max when both traversal and explorative bookkeeping are the same and the matching is optimal, w.r.t. to a log and a model.

For our proposed measure gprec, Axiom 8 holds.

Reasoning. Based on Definition 40, our proposed measure is a ratio between the weighted traversal bookkeeping (Definition 37) and the weighted explorative bookkeeping (Definition 39). As such this ratio can only be 1 (max), when both bookkeeping functions return the same value. The bookkeeping in the numerator can only return the same value as the bookkeeping in the denominator when the matching is optimal, as the denominator assigns weight to each path as if they were optimal. Hence, our proposed measure only returns max when the results of both forms of bookkeeping are the same, or if guards only evaluated to true when they were taken in paths within the matching.

Axiom 9 states that guard-precision is not influenced by log enlargements.

For our proposed measure gprec, Axiom 9 holds.

Reasoning. Based on Definition 40, our proposed measure is a ratio where the numerator and denominator are iterations over the traces in a log. As such, enlarging a log by cloning (Definition 18) will increase both numerator and denominator by a factor of k as the clones result in the same outcomes being counted k times. This factor k cancels out and the same ratio

is returned regardless of the size of the factor.



Tooling

This section outlines the implementations completed to support the contributions of this thesis. For these contributions, a combination of Java and Python were used and to help future researchers, many implementations are available as plug-ins for the ProM framework [236]–[238]. The ProM framework is an open-source academic tool, which had previously been managed by SVN, but is now supported on GitHub¹. The Package within the ProM framework containing work related to this thesis is called “ExogenousData” and the many plug-ins included are explained in detail throughout this appendix.

B.1 ExogenousData (ProM plug-in)

The package “ExogenousData” source repository can be found on GitHub². The following sections outline the functionality captured in the package as of the version ‘0.1.0’ (tagged release). Starting with the data structures used to store exogenous data. The introduced data structures extend the eXtensible Event Stream (XES) [96], [97] format for event logs to ensure the future interoperability of outcomes generated from the package. Then, the implementations for xPM framework (Chapter 3) within the package is explained, highlighting how exogenous annotated logs (xlog) can be obtained as well as the possibilities for extensions. To explore xlogs, two GUI interfaces were implemented to handle exploring and investigating processes with exogenous data. Next, the implementations for stochastic techniques introduced by Chapter 4 are explained. These implementations allow for the discovery, visualisation, and quantification of Exo-SLPNs. Lastly, the implementation of guard-recall and guard-precision introduced by Chapter 5 are explained. These measures have plug-ins but are also expanded on in the GUI interface for investigating processes, which can overlay the measures on process models.

B.1.1 Data Structures

In order to support exo-panels, this section outlines the necessary five attributes within an XES formatted event log needed to parse into an ExogenousDataset (exo-panel), as shown in Listings

¹ The GitHub repository related to this chapter can be found here: github.com/promworkbench/.

² ExogenousData source code can be found here: github.com/promworkbench/ExogenousData/.

B.1 and **B.2**. The first attribute `exogenous:dataset` denotes that the event log should be parsed as an `exo-panel`, of which the value should be `TRUE`. Nested within this attribute, are two additional attributes, where `exogenous:type` denotes the type for values of `exo-measurements`, and `exogenous:link` outlines how a linking function should be constructed for the `exo-panel`. Currently, two types of linkages are supported, `exo-series` are linked based on matching trace attributes (`match`) or an arbitrary `exo-series` will be linked (`any`), the latter allows inter-case variables to be quickly linked. The attribute `exogenous:link:method` denotes which linkage should be used and the nested attributes of `exogenous:link:matching` denote trace attributes to be considered for matching if applicable.

The remaining two attributes relate to the semantics of `exo-series` and `exo-measurements`. The trace attribute `exogenous:name` denotes the `exogenous` influence being studied, optionally this attribute can be removed and the `concept:name` of the log will be used instead. Finally, two attributes are required to describe `exo-measurements`, namely the time and the value. For the time, the default attribute typically used in an event log is repurposed, i.e. the attribute `time:timestamp`. The value of the measurement is denoted by the attribute `exogenous:value`.

Listing B.1: Expected attributes for an event log to be considered an `exo-panel`.

```

1 <?xml version="1.0" encoding="utf-8" ?>
2 <log>
3   ...
4   <string key="concept:name" value="MIMIC III - Blood Pressure Type 1" />
5   ...
6   <!-- the following block is required to signal to the package that the log is an exo-panel -->
7   <string key="exogenous:dataset" value="TRUE">
8     <!-- the type of measurements should be specified below -->
9     <string key="exogenous:type" value="numerical" />
10    <!-- some exo-panels match exo-series to trace attributes -->
11    <string key="exogenous:link:method" value="match" />
12    <string key="exogenous:link:matching" value="" >
13      <!-- an exo-series is linked if the concept:name between trace and exo-series are matched -->
14      <string key="attribute_1" value="concept:name" />
15      <!-- more than one attribute can be specified -->
16      <!-- document order defines ordering of checks -->
17      ...
18    </string>
19  </string>
20  ...
21  <!-- traces are used to represent exo-series -->
22  <trace>
23    <!-- exo-series should name what exogenous factor they are observing -->
24    <string key="exogenous:name" value="BP1"/>
25    ...
26    <!-- event are used to represent exo-measurements -->
27    <event>
28      ...
29      <!-- the typical time:timestamp attribute is expected -->
30      <date key="time:timestamp" value="2000-01-01T00:00:00+01:00" />
31      <!-- the value of the measurement is specified as the following attribute -->
32      <string key="exogenous:value" value="137"/>
33      ...
34    </event>
35    ...

```



```

36     </trace>
37     ...
38 </log>

```

Listing B.2: Alternative linking support for exo-panels as used for experiments in Section 4.8.

```

1 <?xml version="1.0" encoding="utf-8" ?>
2 <log>
3   <string key="concept:name" value="road fines unpaid fines"/>
4   <!-- some exo-panels contain a single exo-series,
5   the following ensures that the exo-panel always links that exo-series -->
6   <string key="exogenous:dataset" value="TRUE">
7     <string key="exogenous:type" value="numerical"/>
8     <string key="exogenous:link:method" value="any"/>
9   </string>
10  ...
11  <trace>
12    <!-- one really long exo-series -->
13    <!-- useful for describing inter-case variables for an event log -->
14    ...
15  </trace>
16 </log>

```

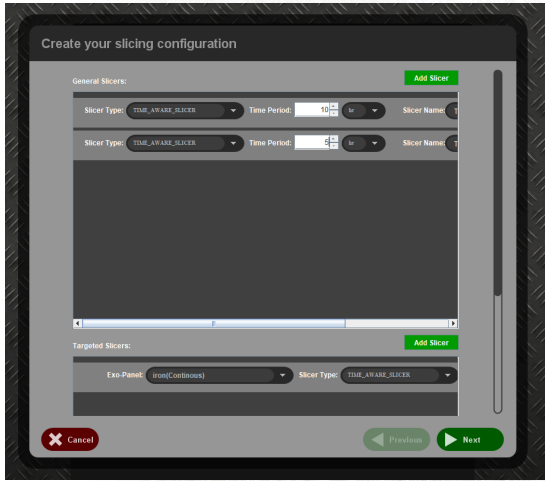
In order to use the following plug-ins for the package, users are expected to be able to construct these event logs themselves and then load them into ProM through pre-existing XES readers. Automating extracting, loading and transforming process of endogenous and exogenous data may be of interest for future endeavours. However, for the moment, this process requires domain knowledge and flexibility so that the package can broadly accept input from all domains.

In some cases, converting the imported EventLog or XLog interface in ProM into an ExogenousDataset can make handling data easier for the user, the plug-in to support this casting is called “Make Log into Exogenous Dataset”. If the user wishes to collect some statistics about an ExogenousDataset, the plug-in “Describe Exogenous Dataset” provides information through the ProM GUI.

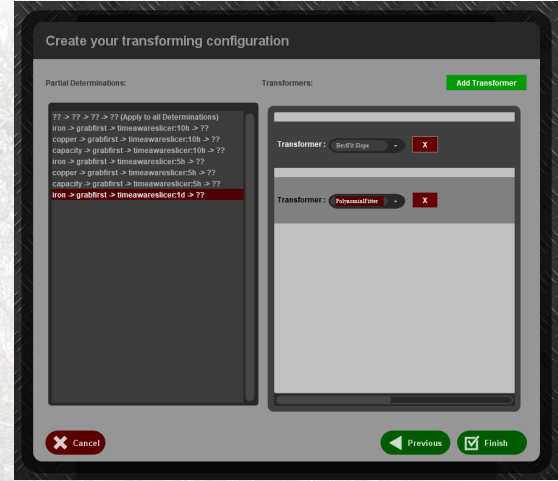
B.1.2 xPM Framework

Instantiating an xPM instance through ProM is facilitated by five plug-ins, namely a log preparation step where determinations are constructed, the application of determinations to create an exogenous annotated log (xlog), and the exploration of the resulting xlog through a visualiser. To demonstrate the implementations a simple example scenario is included within the source for Exogenous data³, the scenario describes a production process where material costs for production are used as exogenous data. The preparation step is facilitated through plug-ins with the name “Exogenous Annotated Log Preparation (X)”, where X usually refers the input taken for

³ The data for the wretched looking to follow along can be found here: github.com/promworkbench/ExogenousData/tree/main/exemplar



(a) Slicing Wizard.



(b) Transformation Wizard.

Figure B.1: GUI wizards for constructing an exogenous annotated event log.

exo-panels, either as XLogs or when left unspecified implies a number of ExogenousDataset are expected.

The preparation plug-in presents a GUI wizard to the user to help construct determinations, as shown in Figure B.1. The linkage functions are created by the ExogenousDataset so the user does not get any options for linkage. Slicing functions are constructed either for all panels, or slicing can be targeted for a specific dataset. Transformation functions are also constructed in the similar manner, for all partial determinations or for a specific partial to complete them.

Alternatively determinations can be constructed pragmatically using the builder pattern instead of the GUI wizard, as shown in Listing B.3. The current selection of slicing and transformation functions are somewhat limited as it is difficult to predict all the possible options that may be needed. To enable future development several interfaces for linking, slicing and transformers are included in the source code.

Listing B.3: Source code for the xPM instantiation for experiments in Section 3.6.2.

```

1 public class AIIM2022 {
2
3     static private List<Double> slicingRanges = new ArrayList<Double>() {{
4         add(2.0);
5         add(4.0);
6         add(6.0);
7         add(12.0);
8     }};
9
10    static private long HOUR = 1000 * 60 * 60;
11
12    private AIIM2022() {}
13
14    static public List<Determination> getConfiguration(
15        List<ExogenousDataset> datasets) {
16        List<Determination> configuration = new ArrayList();
17
18        for(ExogenousDataset dataset: datasets) {
19            if (dataset.getDataType() == ExogenousDatasetType.NUMERICAL) {

```

```

20     for(double range : slicingRanges) {
21         configuration.add(
22             Determination.builder()
23                 .panel(dataset)
24                 .linker(dataset.getLinker())
25                 .slicer(TimeAwareSlicer.builder()
26                     .timePeriod((long) (range * HOUR)).build())
27                 .transformer( SlopeTransformer.builder().build())
28                 .build()
29         );
30     }
31 } else {
32     configuration.add(
33         Determination.builder()
34             .panel(dataset)
35             .linker(dataset.getLinker())
36             .slicer(FutureOutcomeSlicer.builder()
37                 .timePeriod(HOUR * 48).build())
38             .transformer(PossibleOutcomeTransformer.builder()
39                 .Outcome("SEPSIS INFECTION").build())
40             .build()
41     );
42 }
43 }
44 return configuration;
45 }
46 }

```

Listing B.4: Constructing an exogenous annotated log using the configuration in Listing B.3 (AIIM2022).

```

1 public ExogenousAnnotatedLog AIIM2022preparation(final UIPluginContext context,
2     final XLog endogenous, final ExogenousDataset[] exogenous) throws Throwable {
3     final List<ExogenousDataset> exoLogs = Arrays.asList(exogenous);
4     ExogenousAnnotatedLog annotated = ExogenousAnnotatedLog
5         .builder()
6         .endogenousLog(endogenous)
7         .exogenousDatasets(exoLogs)
8         .classifiers(endogenous.getClassifiers())
9         .extensions(endogenous.getExtensions())
10        .useDefaultConfiguration(true)
11        .determinations(AIIM2022.getConfiguration(exoLogs))
12        .globalEventAttributes(endogenous.getGlobalEventAttributes())
13        .globalTraceAttributes(endogenous.getGlobalTraceAttributes())
14        .attributes(endogenous.getAttributes())
15        .parsed(false)
16        .build()
17        .setup(context);
18 }

```

Once an xlog has been constructed, the xlog can be visualised through ProM, as shown in Figure B.2. Users select an endogenous trace using the list of traces (1), after which the right side of the explorer is populated with information related to exogenous data. To enable users to find a desirable trace, the list can be filtered by searching for traces with a `concept:name` starting with a string (1a). An overview graph (2) highlights where endogenous events occurred (vertical lines) and when exo-measurements occurred for the exo-series linked to the trace from

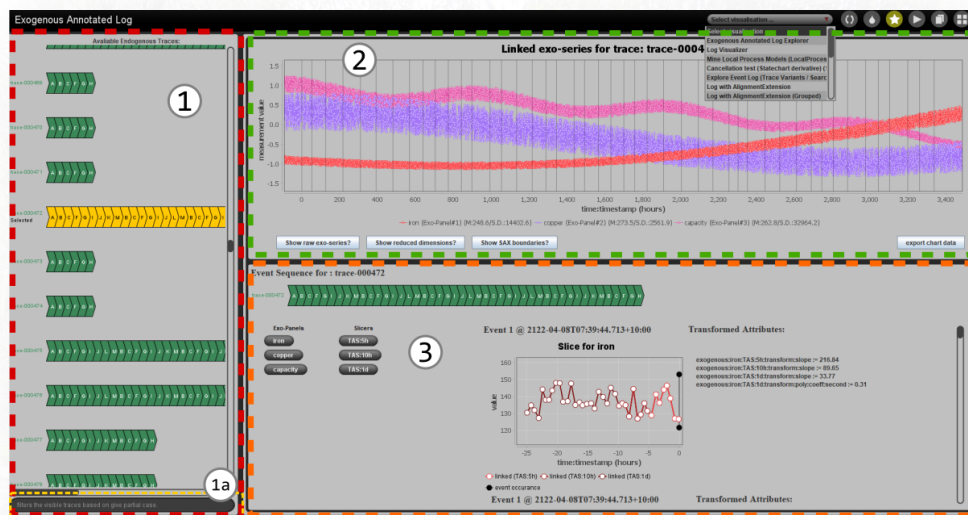


Figure B.2: The exogenous trace explorer enables users to explore exogenous and endogenous data.

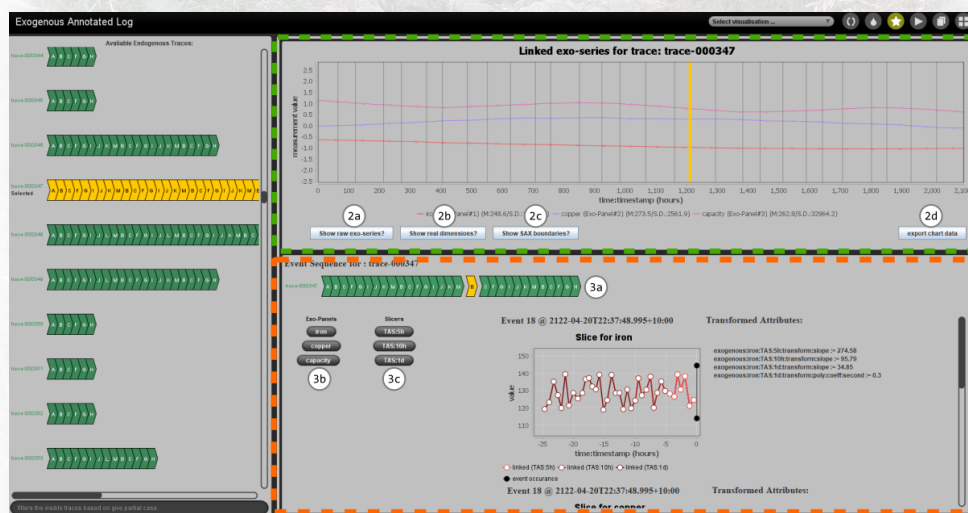


Figure B.3: Options for users to compare behaviour in exogenous data.

exo-panels. The slices related to the trace are denoted below (3) and the transformed attributes associated with events are denoted next to each graph.

Users can explore traces in greater detail using the explorer (2/3) by filtering the slices as shown in Figure B.3. To reduce the difficulty of comparing exo-series, several transforms can be applied using the buttons under the overview graph (2a,2b,2c). Exo-series can be normalised (2a), reduced using piecewise aggregate approximation (2b), and symbolic aggregate approximation (2c) can be highlighted. The data for the overview graph can be exported in csv from using the export button (2d), which includes data for the events and each exo-series. Users can highlight a vertical slice for an event by clicking on the associated chevron in the breakdown under the graph (3a). The slices can also be filtered by the exo-panel (3b) or the duration (3c) to simplify the event level graphs.

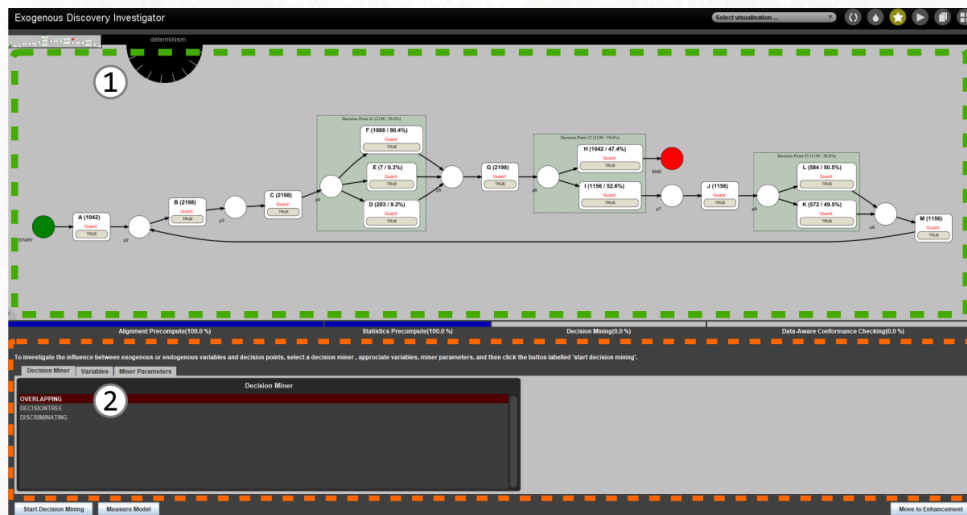


Figure B.4: GUI interface for exogenous aware discovery.

B.1.3 Decision Mining with Exogenous data

After an xlog is constructed, users can explore annotating process models, assuming that the model can be translated into a Petri net, using several plug-ins following convention of “Exogenous Aware Discovery...”. The interface enables users to select an decision mining approach, event attributes to consider for guards, and adjust the parameters of the underlying technique (2) as shown in Figure B.4. The resulting Petri net with Data is displayed (1) to the user using graphviz, namely using Dot, and can be exported in several formats, such as: pdf, png, svg, dot. Users can move the display around and zoom to read guards annotated to a transition. Decision points are highlighted in green for the user to quickly identify where guards can be annotated by decision mining techniques.

The button “Start Decision Mining” (2a) applies a decision mining approach to the original Petri net, based on the selected parameters, as shown in Figure B.5. When the resulting Petri net with Data includes guards that are not set to TRUE, the default case, they are visualised in a user friendly manner (1a) on the transitions. The guard can also be extracted in plain text by clicking the transition (1a), where an information panel will pop out providing further information about the transition. Note that both endogenous and exogenous data attributes are available in this interface and users can use the interface for endogenous decision mining if desired. The progress to complete an alignment, applying decision mining, or measuring the result DPN are tracked through progress bars (1b) to inform the users of the likely time to completion.

These guards are visualised in a way to encourage them to be read visually, interpreting the position of a guard in relation to either weakenings or reinforcements (see Section 5.5.2). These guards are read left to right, where a blue box denotes a weakening, a red box denotes a strengthening, and any other box denotes a guard. For instance, Figure B.6 visualises transition ‘H’ which is annotated with a weakening across, three guards, two reinforcements and a guard (R9) of “cost > 49996”. This weakening evaluates to true, enabling the transition, if any of

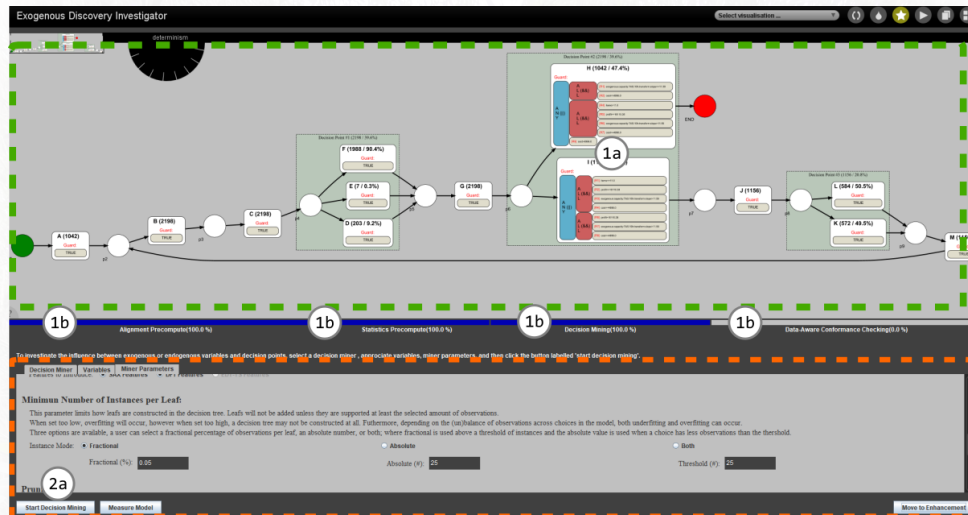


Figure B.5: DPN discovered by applying decision mining.

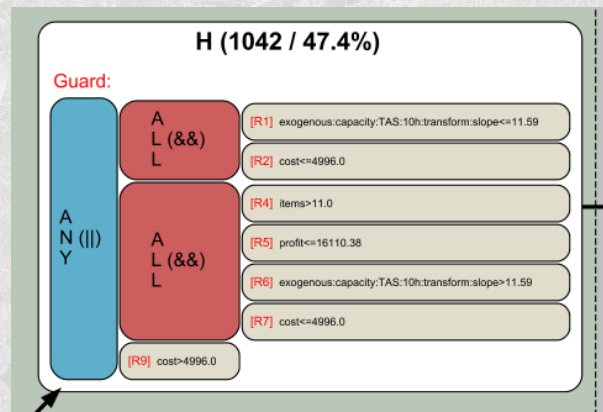


Figure B.6: Visualisation of the guard for transition ‘H’.

three guards evaluate to true. The top reinforcement evaluates to true if both guards R1 and R2 are true. The bottom reinforcement evaluates to true if guards R4, R5, R6, R7 are true. This visualisation in combination with the implemented decision miners, is prone to producing tall guards rather than wide guards, due the implementations ORing between paths through the decision trees.

To enable users to justify the quality of the discovered DPNs, models can be measured through the interface, where pressing the button “Measure Model” triggers quantification. To help the understandability of the quantification, the visualisation of the model is updated to denote, for a measure, how much the transition played a positive role within quantification, shown in Figure B.7. A semi circle is displayed to show the outcome of quantification, where the outer circle is used denote the overall metric and the individual contribution of decision points are denoted using inner wedges of the semi circle. For example, Figure B.7 shows the determinism for the outcome was 0.29, but only one decision point contributed towards the overall metric. Some quantification methods will take longer than others, and all quantification methods to have separate plug-ins that can be called instead if users wish to streamline the

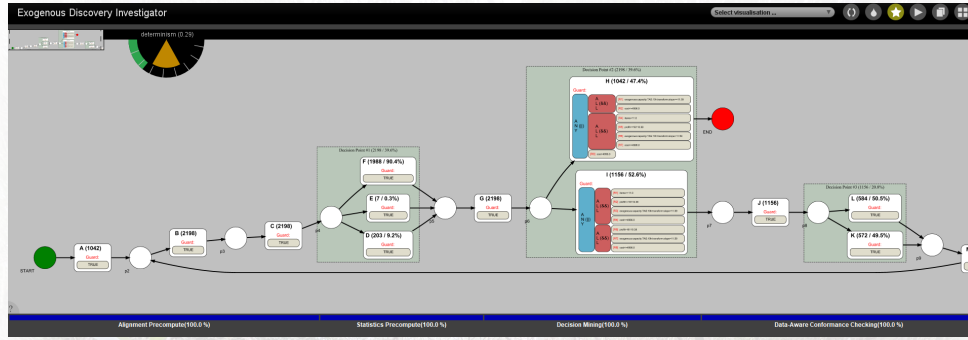


Figure B.7: GUI interface for exogenous aware discovery after applying data-aware conformance checking.

process for scientific endeavours. Currently, the interface supports the following metrics:

- Determinism, as presented in Definition 1;
- Guard-recall, as presented in Definition 38;
- Guard-precision, as presented in Definition 40.

B.1.4 EESA Analysis

It is possible to generate EESA visualisations within the GUI interface, as shown in Figure B.8. Furthermore, any generated EESAs can be ranked using the contributions presented in Section 3.5 and users can explore this ranking to explore any insight graphs. However, the current implementation is somewhat hard coded for the application of experiments conducted in Section 3.6.2 and further work is needed to allow users to generate grouping functions for overlapping EESAs. Users wishing to export all EESAs (which can be memory expensive) can press the “Export” button to dump all graphs out into a directory for further analysis.

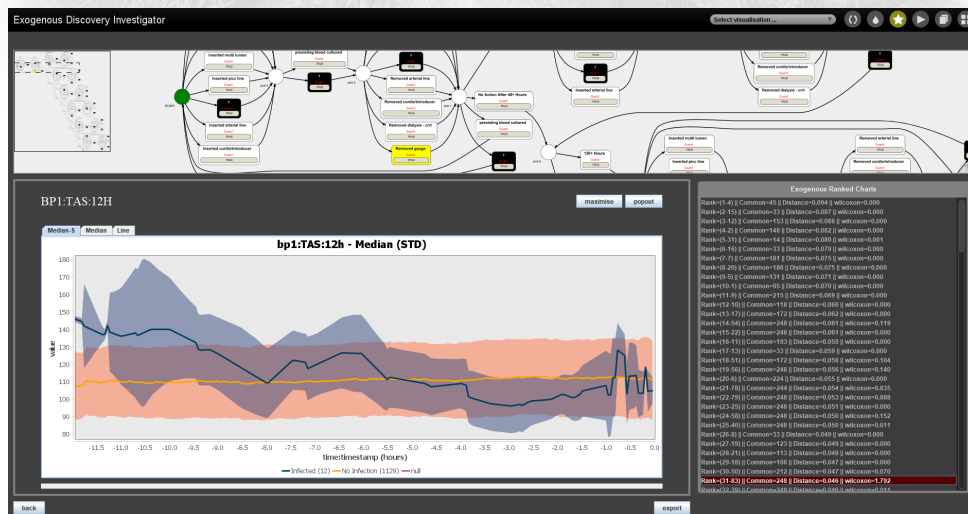


Figure B.8: The ranking of EESAs within ProM.

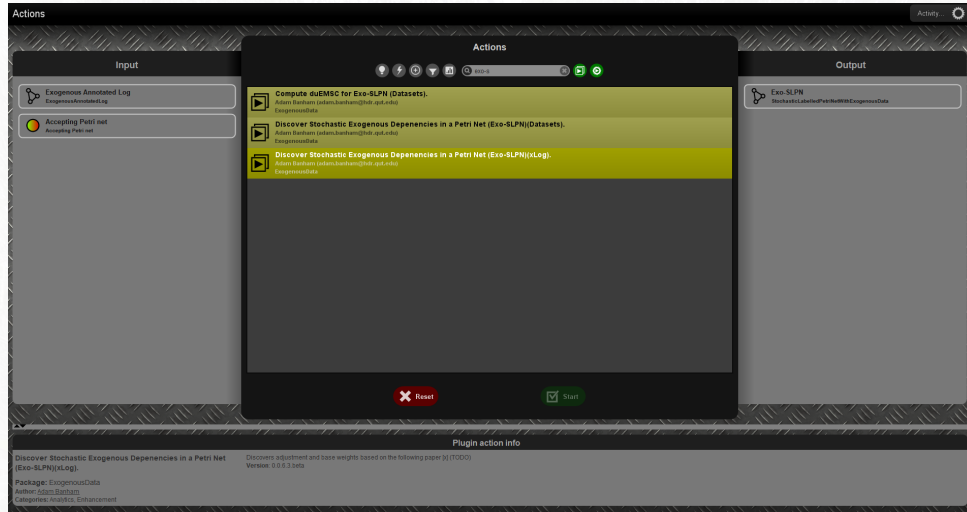


Figure B.9: Stochastic discovery plug-ins.

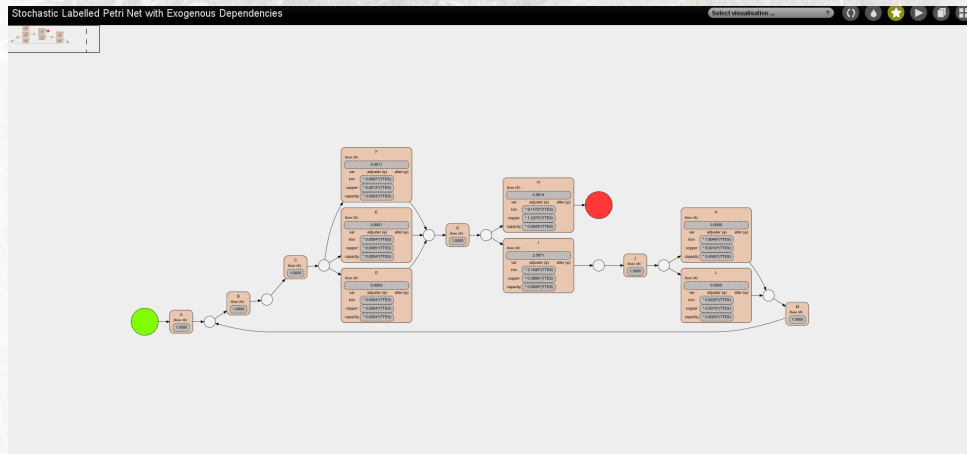


Figure B.10: Exo-SLPN visualised within ProM.

B.1.5 Stochastic Discovery

To apply the techniques presented in Chapter 4, three implementations are available as plug-ins, as shown in Figure B.9. Users can discovery Exo-SLPNs by using an xlog, or by providing an endogenous log and several ExogenousDatasets using the plug-ins named “Discover Stochastic Exogenous Dependencies in a Petri Net”. The process of discovering Exo-SLPNs can be both time consuming and memory expensive due the nature of the problem space being solved, users are encouraged to use a sample of the original log will if they have restricted computing resources. Once an Exo-SLPN is discovered, users can quantify the quality of the outcome by applying data-aware Earth Movers’ Stochastic Conformance, this quantification can also be time consuming and bases the quantification on perfect control-flow matches, meaning for unfitting event logs the results may be unintuitive.

Exo-SLPNs can be visualised by the user through ProM, as shown in Figure B.10. The base weights and parameters are visualised on the transitions, where the base weight is shown first, then a table of adjustments and alternative weights are shown below. Only weights which are

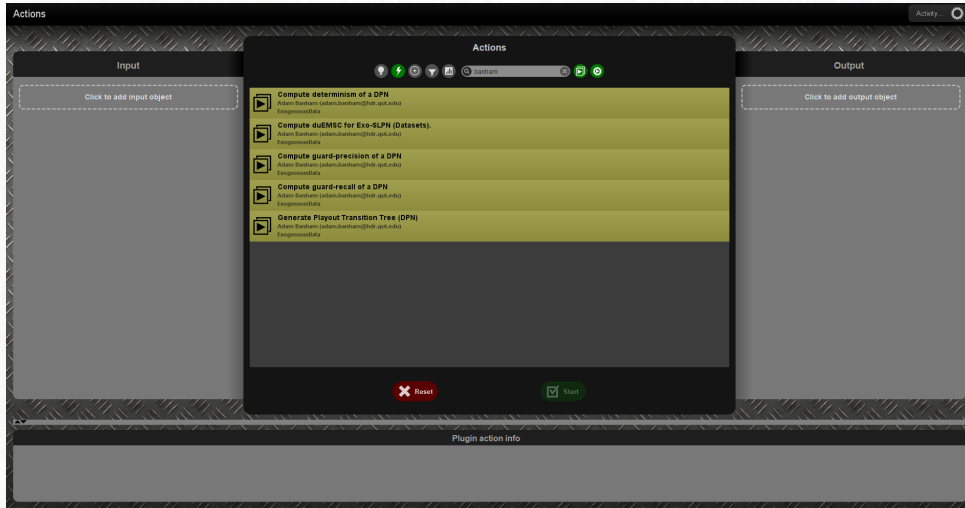


Figure B.11: Conformance checking plug-ins for ProM.

not set to 1.0 are shown to the user to reduce the complexity. Notably when parameters are set to 1.0, parameters do not affect the outcome of weight functions and the information shown can still be used to hand compute weights. The visualisation is produced managed through graphviz (Dot) meaning the visualisation can be exported in many formats, such as: .png, .svg, or .dot.

B.1.6 Data-aware Conformance Checking

The implementations for Chapter 5 were originally presented using python, which be found here⁴, but ProM implementations have also been developed. In total five conformance checking plug-ins were developed as show in Figure B.11, three relating to the contributions of Chapter 5, namely guard-recall and guard-precision. For the interested wretched, a plug-in for deriving a playout tree (Definition 27 with flow reduction) is available and the resulting tree can be visualised by ProM, as shown in Figure B.12. The visualisation of the playout tree is handled by graphviz (Dot) and exports out to many formats.

⁴ Source repository for python implementations github.com/AdamBanham/data-aware-conformance.

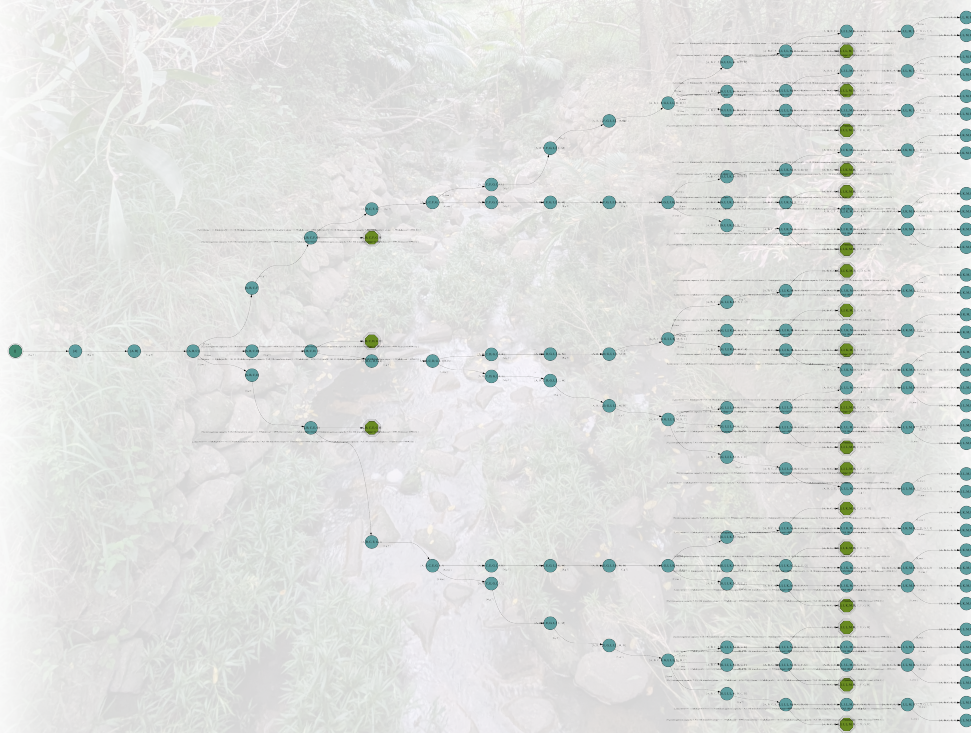


Figure B.12: Visualisation of a playout tree for the exemplar included in the source repository for exogenous data, where k was 16.

Indexed Terms

- alignments, 35
 - data-aware, 38, 39
- business process management (BPM), 18
- conformance checking, 5, 35, 125, 195
 - data-aware, 37
 - desirable qualities, 148
 - existing, 195
 - desirable properties, 125
 - determinism, 66
 - guard-precision
 - definition, 153
 - sketches, 198
 - guard-recall
 - definition, 152
 - sketches, 197
- data-aware models
 - choices, 127
 - guards, 127, 134
- decision making, 6, 50
 - choices, 7
- decision mining, 40, 70, 128
- endogenous data, 3
 - examples, 50
- exogenous data, 8
 - contextual data, 24
 - onion models, 24
 - examples, 8, 50, 54, 82, 114
 - notation, 52
- healthcare, 27, 50
- historical recordings
 - event logs, 2, 22
- information systems, 17
- IoT, 28
- modelling formalisms
 - DECLARE, 34
 - directed graphs, 29
 - Petri nets, 30
 - notation, 52
 - process models, 3
 - process trees, 32
- multi-perspective process mining, 26, 50
- process discovery, 5, 29
- process enhancement, 6, 39
- process mining, 4, 21, 50, 82
- processes
 - annotations, 7
 - definition, 2
 - examples, 2
 - perspectives
 - control-flow, 4, 29
 - viewpoints, 23
- research
 - approach, 14
 - contributions, ix, 12
 - data-aware conformance checking, 133, 134, 139, 146, 148, 151
 - EESA, 51, 59, 62, 63



Exo-SLPN, [100](#), [104](#), [109](#)
structure, [12](#)
xPM, [51](#), [53](#)
goal, [6](#)
implementations, [65](#)
literature gaps, [43](#)
publications, [14](#)
questions, [11](#)
thesis overview, [15](#)
stochastic process mining, [41](#), [82](#)
tabular data, [22](#)
time series, [9](#)
workflow mining, [17](#)
wretched, [viii](#), [1](#), [49](#), [81](#), [124](#)
