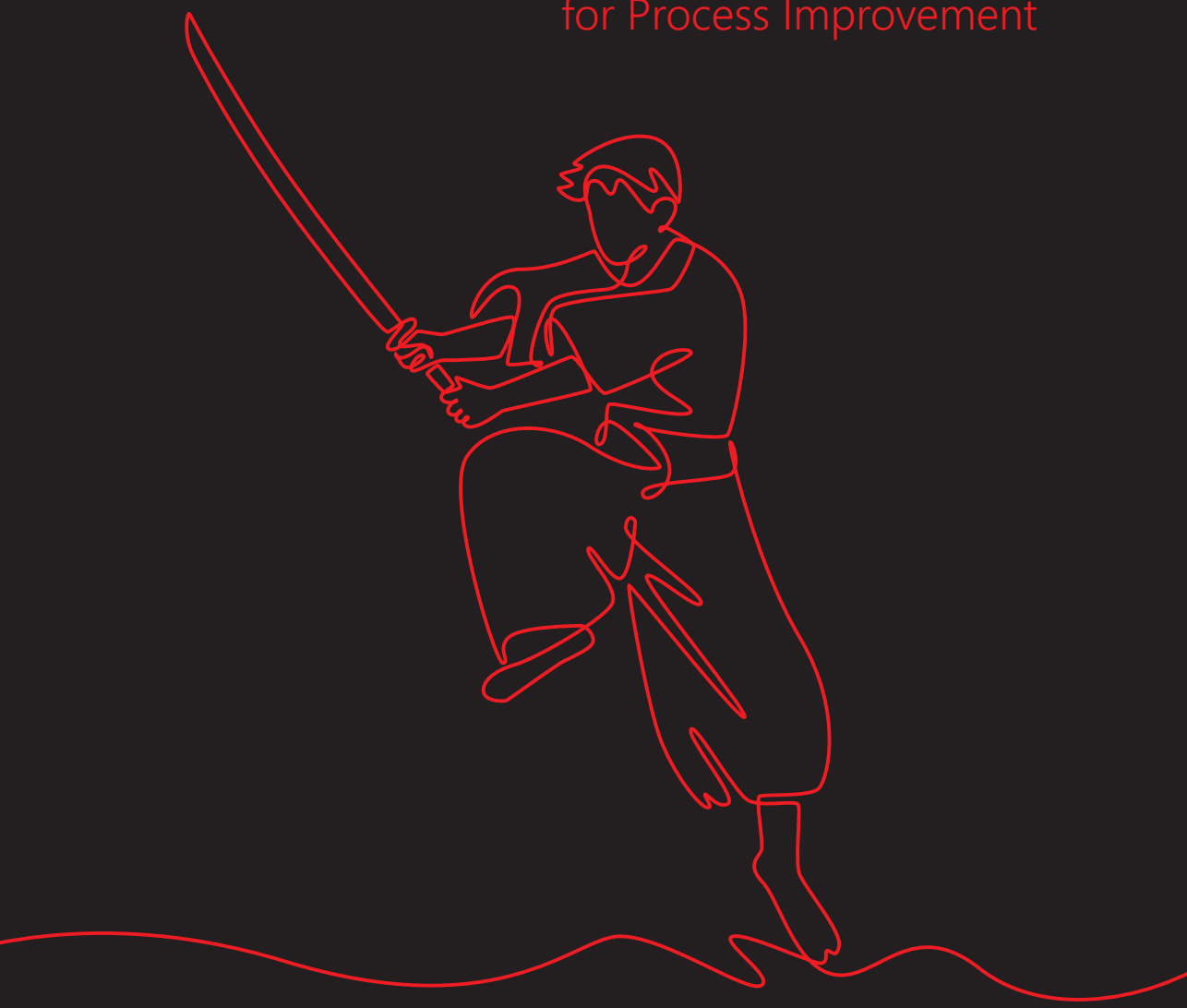
The SWORD Unsheated

Data-Driven Workaround Detection
for Process Improvement



Wouter van der Waal

The SWORD Unsheathed: Data-Driven Workaround Detection for Process Improvement

Wouter van der Waal

Copyright © 2026, Wouter van der Waal
The SWORD Unsheathed: Data-Driven Workaround Detection for Process Improvement
ISBN: 978-94-6534-652-6
DOI: 10.33540/3693
All rights reserved, unless otherwise stated
Druk: ProefschriftMaken
Cover images: Adapted from OneTime / Shutterstock.com
Cover logo: Adapted from User:TigerUnity1 / Wikimedia Commons / CC-BY-SA-3.0

The SWORD Unsheathed

Data-Driven Workaround Detection for Process Improvement

Datagedreven Workarounddetectie voor Procesverbetering

(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor aan de Universiteit Utrecht op gezag van de rector magnificus, prof. dr. ir. W. Hazeleger, ingevolge het besluit van het College voor Promoties in het openbaar te verdedigen op

woensdag 28 oktober 2026 des middags te 12.15 uur

door

Wouter Gerard van der Waal

geboren op 17 mei 1992
te Pijnacker

Promotor:
Prof. dr. ir. Hajo A. Reijers

Co-promotor:
Dr. Inge van de Weerd

Beoordelingscommissie:
Prof. dr. D. Beverungen
Prof. dr. F. Dalpiaz
Prof. dr. B.J. van den Hooff
Prof. dr. N. Martin
Prof. dr. A. del Rio Ortega

Dit proefschrift werd (mede) mogelijk gemaakt door financiële steun van NWO in het WorkAround Mining (WAM!) project [project code 18490]

Philosophy has only two purposes: to describe the world, and to change it. Everything else is vanity and self-delusion.

Straton of Stageira.

Tales and dreams are the shadow-truths that will endure when mere facts are dust and ashes, and forgot.

The Sandman.

I may be a fool, but I intend to be a live fool.

Mat Cauthon.

Dankwoord

Hoewel mijn naam op de voorkant van dit boekje staat, zou het er nooit hebben kunnen zijn zonder een heleboel andere mensen. Ik hoop dat mijn waardering wel al een beetje bekend is, maar ik wil een aantal mensen hier nog expliciet bedanken.

Inge, ik ben nog steeds onder de indruk van de balans die je weet te vinden tussen idealistisch pushen naar een hoger niveau en pragmatisch mikken op haalbare resultaten. Extra bijzonder vind ik ook dat je de keuze hiertussen maaken altijd heel natuurlijk en comfortabel was. Zowel met onderzoek als onderwijs heb ik van begin tot eind altijd fijn met je samengewerkt en ik heb op beide gebieden nog veel meer geleerd dan ik van tevoren had gedacht.

Direct daarachter staat uiteraard mijn supervisor. Hajo, je enorme enthousiasme over onderzoek in het algemeen en over zelfs kleine bevindingen tijdens studies werkte aanstekelijk, waardoor ik motivatie bleef hebben om verder te gaan tijdens het hele traject. Ik wil je bedanken voor de mogelijkheden die je me hebt geboden om mezelf verder te kunnen ontwikkelen en in het bijzonder ben ik heel erg blij met alle ruimte die je hebt gegeven om zelf keuzes te maken waar dat mogelijk of nodig was.

Veel onderzoekswerk doe je alleen of met je supervisors, maar gelukkig heb ik geluk gehad met mijn collega's in de Utrecht Process Science groep. Mari, het was ontzettend leuk om een collega te hebben om zoveel mee te kletsen over The Wheel of Time, films, statistiek, psychologie en random onderzoeken. Vanwege je unieke perspectief binnen de groep heb ik daar ook veel van geleerd. Mede daarom vind ik het ook erg fijn dat je me tijdens mijn verdediging wil ondersteunen als paranimf. Iris, sowieso zou mijn onderzoek nooit hebben bestaan zonder jouw PhD werk, maar het was ook handig om iemand te hebben die weet hoe je mijn supervisorteam het beste managet. En natuurlijk is het altijd gezellig om met jou op reis te gaan.

Nico en Jan Martijn, ik ben heel blij met jullie ondersteuning in mijn ontwikkeling als docent met de vakken die we samen hebben gegeven en de vele discussies over onderwijs, studenten en de universiteit in het algemeen. Xixi, het was altijd gezellig om na een te lange dag onderwijs een beetje bij te komen en stoom af te blazen. Het kwam mij wel goed uit dat je er zo vaak was om nog even iets af te maken, terwijl de rest van de groep alweer naar huis was. Jos en Kelly, ons kantoor was altijd hartstikke gezellig door gesprekken over werk, maar zeker ook privé of random filosofie. Misschien ben ik jullie nog wel het meest dankbaar voor me niet uit ons kantoor zetten als jullie wel wilden werken.

Vinicius and Suhwan, I'm really grateful we've been working on our PhDs and BKOs at the same time. Complaining about common problems, despite our vastly different topics, has been essential to getting through tough times. Claudio and Michele, I'm glad to call you my friends, and I'm thankful for the little Italian I've learned from you. Of course, there is no way I can repeat what I've learned here, but I'm sure it will be useful someday.

Natuurlijk ook mijn dank aan alle zorgprofessionals die op verschillende momenten mee wilden denken over de data en konden uitleggen hoe ze hun werk uitvoeren. Binnen deze groep zijn er drie mensen die ik specifiek wil bedanken: Teus, jouw gastvrijheid en vertrouwen hebben enorm veel geholpen met het project opstarten. Ook onze discussies tijdens de verschillende onderzoeken hebben een duidelijke stempel gedrukt op het werk dat hier nu staat. Saskia, hartelijk bedankt voor je enthousiasme tijdens onze vergaderingen en alle hulp bij het vinden van de juiste personen in het ziekenhuis. Of we alleen een expertinterview nodig hadden of juist volledige casestudies, jij kende altijd wel iemand die precies aansloot bij

onze wensen. Bob, onze samenwerking was zeker niet standaard, maar ik ben ontzettend blij dat we ons onderzoek samen konden doen. Juist de betrokkenheid van de niet-academische kant van de medische wereld heeft mij enorm geholpen om ons werk nog meer te waarderen.

Het zal niemand verbazen dat ik lesgeven even leuk vind als onderzoek doen. Ook voor onderwijs komt dat voor een groot deel door de mensen, in dit geval studenten. Deze groep is zeker te groot om iedereen apart te bedanken. Ondanks dat er een kans is dat ik belangrijke studenten vergeet (of namen ben vergeten), doe ik hier toch een poging om een aantal bijzondere studenten te noemen: Ahlam, Almira, Bart, Farheen, Fleur, Isa, Jari, Jelke, Jeroen, Julius, Kevin, Lara, Lisa, Lynne, Niels, Matthijs, Maurits, Michel, Mim, Quinten, Ratiba, Rijk, Rob, Robin, Ruben, Sander, Sanne, Sha're, Thijs, Tomas, en Willem. Door studenten zoals jullie ben ik zelfs statistiek leuk gaan vinden.

Naast mijn professionele leven wil ik graag iedereen bedanken die heeft geholpen door me juist zo min mogelijk aan werk te laten denken. Zonder jullie zou ik dit werk ook nooit af hebben kunnen maken. Mijn familie komt daarbij op de eerste plaats. Pap, Mam, Marjolein en Annemieke, bedankt voor al jullie hulp, support en steun voor, tijdens en na mijn PhD. Erwin, dat geldt natuurlijk ook voor jou, maar ik ben ook extra blij dat we nog steeds zo veel oude series samen kijken.

Om af te sluiten wil ik graag budoschool Jishindo bedanken. De trainingen van Joost en Marco hebben me iedere keer weer een uitstekende reden gegeven om op tijd te stoppen met werken. Ik heb de afgelopen jaren veel geleerd over onder andere balans en flow, wat zowel binnen als buiten aikibudo nuttig is. Ook blijkt met je gezicht vooruit in een mat gegooid worden een uitstekende manier te zijn om even niet aan onderzoek te denken. Natuurlijk ben ik heel erg blij dat alle leden van de club altijd klaarstaan om mij op verschillende manieren aan te vallen, rond te gooien of vast te klemmen. Steffi, ik ben jou in het bijzonder dankbaar voor je hulp en support binnen en buiten de trainingen. De afgelopen jaren is het altijd leuk geweest om samen verder te komen en naar een volgend niveau te werken. Met jou als paranimf denk ik dat de fysieke kant van mijn verdediging helemaal in orde is.

Nederlandse Samenvatting

Organisaties gebruiken vaak gespecialiseerde informatiesystemen om complexe processen te stroomlijnen en werknemers te ondersteunen. Als de ondersteuning van het dagelijks werk echter tekortschiet, gaan werknemers vaak om het systeem heen werken. Deze workarounds zijn opzettelijke afwijkingen van het voorgeschreven proces, waarmee werknemers hun werk ook kunnen uitvoeren als ze tegen onverwachte obstakels aanlopen. Workarounds laten daarmee zien welke problemen werknemers in hun werk ervaren en welke alternatieve oplossingen zij gebruiken. Procesmanagers kunnen deze informatie gebruiken om processen te verbeteren. In de praktijk kennen zij echter veel workarounds niet. Tot nu toe worden workarounds meestal in kaart gebracht door domeinexperts te observeren en te interviewen. Dit is vaak lastig: experts kunnen of willen niet altijd vertellen over onderdelen van hun werk waarin zij moeite hebben om het voorgeschreven proces te volgen. Bovendien kost deze aanpak veel tijd van domeinexperts, die het vaak al druk hebben. In dit proefschrift gebruiken we procesdata uit informatiesystemen om workarounds op een datagedreven manier te vinden met aanzienlijk minder inspanning van domeinexperts. Hiermee dragen we bij aan zowel de praktijk als wetenschap met: 1) een nieuwe, datagedreven methode voor workaroundsdetectie met het Semiautomatische WORkaround Detectie (SWORD) framework, 2) de mogelijkheid om workarounds te vinden en te kwantificeren over langere tijd met zeer beperkte benodigde moeite van domeinexperts, 3) nieuwe theoretische inzichten over hoe workarounds ontstaan en veranderen over tijd en hoe hun adoptie verschilt tussen rollen van werknemers.

English Summary

Organizations often use specialized information systems to streamline complex processes and support workers. However, when the information system does not support the day-to-day processes well, workers may start to work around the system. These workarounds, defined as intentional deviations from the prescribed process, allow workers to perform their job even when they encounter unforeseen obstacles. As such, workarounds reveal both issues workers encounter in their work and alternative paths that may be used as a solution. While process managers can therefore use workarounds to improve processes, many workarounds are unknown to them. In the current state of the art, workarounds are usually discovered by observing and interviewing domain experts. However, they may be unable, or even unwilling, to discuss parts of their work where they find difficulties following the prescribed process. Consequently, this approach requires a lot of time from already busy domain experts to discover workarounds that they are willing to share. In this dissertation, we use process data from information systems to mine workarounds in a more data-driven manner and to reduce the burden on domain experts. With this, we contribute to both practice and scientific literature by: 1) developing a novel, data-driven workaround detection approach through the Semi-automated WORkaround Detection (SWORD) framework, 2) enabling the discovery and quantification of workarounds over extended periods with very limited effort required from domain experts, and 3) providing new theoretical insights into the dynamics of workaround emergence, evolution, and adoption patterns based on actor roles.

Contents

1 Introduction **1**
1.1 Workaround Discovery . . . 1
1.2 Impact of Workaround Research . . . 2
1.3 Methods . . . 3
1.4 Contributions . . . 4
1.4.1 SWORD Framework . . . 5
1.4.2 Theory . . . 5
1.4.3 Practice . . . 5
1.5 Thesis Structure . . . 6
1.5.1 Chapter 2 - Putting the SWORD to the Test . . . 6
1.5.2 Chapter 3 - Whetting the SWORD . . . 6
1.5.3 Chapter 4 - The Emergence and Evolution of Workarounds . . . 7
1.5.4 Chapter 5 - Resource-Oriented Workaround Analysis . . . 7
1.5.5 Chapter 6 - Operating the SWORD . . . 7

2 Putting the SWORD to the Test **9**
2.1 Introduction . . . 9
2.2 Related Work . . . 11
2.3 Research Approach . . . 12
2.3.1 Phase 1: Defining the Workaround Detection Patterns . . . 12
2.3.2 Phase 2: Evaluating Patterns in a Multiple-Case Study . . . 14
2.4 SWORD Framework . . . 14
2.4.1 Workaround Detection Patterns – Literature Review . . . 15
2.4.2 Workaround Detection Pattern Analysis . . . 18
2.4.3 Patterns . . . 20
2.4.4 Ranking . . . 24
2.4.5 Application . . . 25
2.5 Multiple-Case Study . . . 26
2.5.1 Setting . . . 26
2.5.2 Data Collection . . . 26
2.5.3 SWORD Framework . . . 27
2.5.4 Evaluation . . . 27
2.5.5 Results . . . 28
2.5.6 Threats to Validity . . . 30
2.5.7 Conclusion . . . 31
2.6 Discussion . . . 31
2.6.1 Scientific Implications . . . 31
2.6.2 Practical Implications . . . 32
2.6.3 Limitations . . . 33
2.7 Conclusion . . . 33

3 Whetting the SWORD **35**
3.1 Introduction . . . 35
3.2 Related Work . . . 37
3.2.1 Workaround Mining . . . 37

3.2.2 Data Requirements . . . 38
3.3 Approach . . . 38
3.3.1 Constructing the Model . . . 38
3.3.2 Training the Model . . . 39
3.3.3 Evaluating the Model . . . 40
3.4 Case Study . . . 41
3.4.1 Constructing the Model . . . 42
3.4.2 Training the Model . . . 42
3.4.3 Evaluating the Model . . . 42
3.5 Discussion . . . 44
3.5.1 Implications . . . 44
3.5.2 Limitations . . . 46
3.6 Conclusion . . . 46

4 The Emergence and Evolution of Workarounds 49
4.1 Introduction . . . 49
4.2 Theoretical Background . . . 50
4.3 Method . . . 51
4.3.1 Context . . . 52
4.3.2 Data collection and analysis . . . 52
4.4 Results . . . 55
4.4.1 Evolution . . . 55
4.4.2 Incidents . . . 59
4.5 Discussion . . . 61
4.5.1 Temporality of Workarounds . . . 61
4.5.2 Limitations . . . 63
4.6 Conclusion . . . 64
4.6.1 Future work . . . 64

5 Resource-Oriented Workaround Analysis 67
5.1 Introduction . . . 67
5.2 Related Work . . . 68
5.2.1 Workarounds . . . 68
5.2.2 The resource perspective in BPM . . . 69
5.2.3 The resource perspective in routine dynamics . . . 69
5.3 Approach . . . 70
5.3.1 Context . . . 70
5.3.2 Data collection and preparation . . . 70
5.4 Case Study . . . 71
5.4.1 Background . . . 71
5.4.2 Data Extraction . . . 71
5.4.3 Data Results . . . 72
5.4.4 Interviews . . . 73
5.5 Discussion . . . 74
5.6 Conclusion . . . 75

6 Operating the SWORD 77
6.1 Introduction . . . 77
6.2 Related Work . . . 79
6.3 SWORD Framework . . . 80
6.3.1 Data Processing . . . 80
6.3.2 SWORD Framework Analysis . . . 80

6.3.3 Classifying . . . 81
6.3.4 Filtering . . . 82
6.4 Case Study . . . 83
6.4.1 Data extraction . . . 84
6.4.2 Round 1 . . . 84
6.4.3 Round 2 . . . 87
6.4.4 Evaluation . . . 88
6.5 Discussion . . . 88
6.5.1 Workarounds in Healthcare . . . 89
6.5.2 Process Mining in Healthcare . . . 89
6.5.3 Limitations . . . 90
6.6 Conclusion . . . 90

7 Conclusion **93**
7.1 Artefactual Contributions and Implications . . . 93
7.2 Empirical Contributions and Implications . . . 94
7.3 Theoretical Contributions and Implications . . . 94
7.4 Limitations . . . 95
7.5 Outlook on Workaround Mining . . . 96

Bibliography **99**

Curriculum Vitae **112**

Chapter 1 Introduction

For decades, most organizations have used information systems to support various parts of their day-to-day work (Symons, 1991; Boell and Cecez-Kecmanovic, 2015). For example, large companies use enterprise resource planning systems to distribute information to integrate corporate operations (Ignatiadis and Nandhakumar, 2009; Grabski et al., 2011), personnel is managed through human resource information systems (Kovach and Cathcart, 1999), and hospitals use health information systems to support patient care (Haux, 2006). Successful implementations can support organizations on various levels.

Despite the overall advantages in integrating information in these systems, not all implementations are successful. If the system does not feel advantageous, users may resist its use, reducing the efficiency of the organization and thus limiting the effect of the often costly implementation (Azad and King, 2008; Safadi and Faraj, 2010; Grabski et al., 2011). Avoiding the use of the system may also occur if users actively disagree with new features, such as tools to monitor their work habits (Ferneley and Sobreperez, 2006). Finally, in the case of healthcare, errors introduced through bad implementations can even lead to dangerous situations for patients (Van den Hooff and Hafkamp, 2017).

If users do not believe the system is advantageous to use, they may work around it to reach their goal in a different way than expected or prescribed by management (Alter, 2014; Ejnefjäll and Ågerfalk, 2019). For example, users may share their login details with each other to access each other's data (Ignatiadis and Nandhakumar, 2009) or even create spreadsheets outside of the approved systems as shadow tools (Handel and Poltrock, 2011). Alternatively, if users are busy, they may delay registration of their work until a more convenient time (Ferneley and Sobreperez, 2006). Such workarounds are a good indication of problems where the information system does not support or even hinder the required process. As such, they can be used to identify and solve problems in the process. Therefore, the discovery of workarounds is valuable to learn more about the process and to identify issues and opportunities for improvements.

In this chapter, we will first discuss the way workarounds are currently discovered in Section 1.1 and explore the impact workarounds have on improving processes in practice in Section 1.2. Then, we provide an overview of the remaining chapters with a brief look into the methods used in Section 1.3, the main contributions in Section 1.4, and finally an outline of the dissertation in Section 1.5.

1.1 Workaround Discovery

To study workarounds in any way, the first step is to find them. Traditionally, the way to find workarounds is to talk with the workers involved in the process (Ejnefjäll and Ågerfalk, 2019). As workers are usually aware of the way they are supposed to be working compared to the way they are working, this method has led to the discovery of many workarounds. Workarounds have been found with interviews in, for example, the resource planning system

of a rail transport equipment contractor (Ignatiadis and Nandhakumar, 2009) or in the health information systems of multiple hospitals (Doctor et al., 2025).

To account for the issue that workers may not remember workarounds or may not even want to share how they are deviating from the prescribed rules (Debono et al., 2025), workarounds may also be discovered through direct observations (Ejnefjäll and Ågerfalk, 2019). This method has also been used successfully to find workarounds, for example, in health information systems (Azad and King, 2008), administration records of nursing homes (Vogelsmeier et al., 2008), or medication administration systems in hospitals (Koppel et al., 2008). However, care should be taken as people may alter their way of working when observed (Adair, 1984; Weinzierl et al., 2022). In addition, rare workarounds are unlikely to be found as chances of observing them in any given time frame are, by definition, low. Finally, both qualitative methods require a lot of effort from domain experts to discover any workaround, generally requiring dozens of hours for a single study.

To address these issues, two quantitative approaches have been proposed to discover workarounds. Outmazgin and Soffer (2014) propose six workaround types that may be used to recognize workarounds in information system data. For example, workers can bypass specific parts of a process or actor roles perform tasks they are not authorized for. For this, specific domain knowledge is required to recognize workarounds, like a process model or strict definitions about authorization. As such, it is an excellent way to find workarounds in well-defined contexts or to detect more instances of known workaround types, but it is not designed to discover new ones. Weinzierl et al. (2022) use neural networks to classify large numbers of traces as workarounds or normative. Using partially synthetic logs, they demonstrate that this approach can detect workarounds in large datasets. However, inherent to neural networks, it requires a large number of traces labeled as workaround or normative. As this work needs to be done manually, it is highly demanding of domain experts.

This dissertation focuses on quantitative, data-driven workaround mining to discover new workaround types while limiting the effort domain experts need to put in. We use event data that is already logged in information systems as a starting point and use insights obtained from this to have much shorter, targeted interviews with domain experts to discover workarounds. In addition, as these systems store data over years, we can study workarounds over very long periods with this longitudinal data. Such information can then be used to investigate how the workaround type itself changes in matters of emergence, evolution, and diffusion within the organization.

1.2 Impact of Workaround Research

Knowledge of workarounds provides specific insight into processes (Alter, 2014; Beerepoot and Van de Weerd, 2018; Ejnefjäll and Ågerfalk, 2019). Each workaround reveals both obstacles in the underlying process and alternative paths to circumvent them, so the types of insights they provide are also twofold: the obstacles indicate the challenges of the current process, while alternative paths provide inspiration for improved processes. Given this potential, many studies concerning workarounds note process improvement as a practical contribution. Because most studies are primarily focused on the detection of workarounds instead of their potential process optimization, even case studies focused on individual workarounds or processes traditionally tend to finish with general advice to the sector rather than giving recommendations on the specific workarounds that were studied.

For example, in the healthcare sector, after finding why clinicians were required to work around bar-coded medication administration systems, the authors of the study provide general recommendations, such as using modern user interfaces or using multidisciplinary teams to evaluate the implementation of new systems (Koppel et al., 2008). Similarly, Van Onzenoort

et al. (2008) suggest more education to increase compliance with bar-code-enabled point-of-care systems. Education is also mentioned in a more general sense to promote an environment of open communication in nursing students (Westphal et al., 2013; Debono et al., 2025). While these suggestions are inspired by the workarounds, it only touches upon their potential without making full use of the multiple perspectives they provide.

One explanation for this lack of practical impact on organizations that discovered workarounds seem to have is that even a single workaround may be very complex to address. Workarounds are not inherently good or bad and thus require careful balancing to handle (Reineke et al., 2025). Without accepting this duality in even a single workaround, problems surrounding workarounds cannot be solved (Debono et al., 2025). As the positive and negative sides of a single workaround often lie with different stakeholders in the process, the already complex management can become political (Goff et al., 2021), strengthened by the issue that some workarounds even stem from a misalignment between the stakeholders' goals (Outmazgin et al., 2020). For example, where nurses and physicians primarily want to treat their patients, upper management is generally concerned with keeping the hospital financially healthy.

Two studies address the complexities of deciding how to handle a single workaround. Discrepancies in goals can be further analyzed using the *motivational model* (Soffer et al., 2023), and workarounds can be classified with experts as something to prevent, redesign, adopt, or ignore with the *workaround action impact matrix* (Beerepoot and Van de Weerd, 2018) by determining if the process needs to change and how much managerial effort should be spent. Even so, we only find one study for each approach that aims for targeted process improvement with workarounds: Outmazgin et al. (2023) use the motivational model to suggest four explicit improvements in a public academic college, and Assbrock et al. (2024) used the workaround action impact matrix to formulate decisions about eight workarounds in two manufacturing companies and a public university. At the same time, Beerepoot and Van de Weerd (2018) and Assbrock et al. (2024) show that the potential organizational impact is strongly limited by the difficulty of their discovery: both requiring more than twenty hours of interviews, observations, or workshops for twelve and nine discovered workarounds, respectively. These studies show that workarounds, once discovered, can indeed be used to improve processes, but that the ease of discovering them needs to be improved to increase the practical impact of workaround research in real life. Therefore, we aim to make workaround detection easier through the use of event data and especially limit the effort domain experts are currently required to spend on it.

1.3 Methods

For our data-driven approach to workaround mining, process mining is the technical foundation. While workaround mining was not an established field within this context, the way of representing process pathways through event logs (Van der Aalst et al., 2007) allows for a quantitative lens to look through. Compared to qualitative methods, this enables us to separate individual process instances more strongly from each other than would be possible in interviews, where the separation is limited by the memories of the participants, or observations, where multiple processes tend to occur at the same time (Günther et al., 2010). While there are data requirements for any kind of process mining, this separation makes it possible to compare separate executions of the same process and determine differences that lead to workarounds. With the combination of patterns from various process mining fields, such as conformance checking and deviation detection, we construct the Semi-automated WORkaround Detection (SWORD) framework in Chapter 2 as a new way to use process mining to mine for workarounds, which we apply in every following chapter.

While the goals of the individual studies in this dissertation range from theoretical to practical, we perform our process mining case studies in a real-life setting. This type of study is especially useful for studying natural phenomena that are inseparable from their context (Yin, 2013). As workarounds require a rich context, such as a normative process, perceived obstacles, and motivation, case studies are a fitting method to study them in detail. The case studies start with the data-driven discovery part. However, while this enables us to find out *what* workers did, we cannot know *why* they did so. The only way to find out what led to this data is to go to the source: domain experts. Therefore, in all studies presented in this dissertation, we extend our data-driven approach with interviews. In Chapters 2 and 6, we primarily use the interviews to assess whether the deviations identified through the SWORD framework are indeed workarounds. In Chapters 3 and 5, we also use interviews to get more information about the process and to understand the differences between workarounds and normative behavior. As an alternative evaluation, we analyze historic documents instead to understand changes in the process in Chapter 4.

As repeated case studies within the same context limit the demonstrated generalizability of the underlying methodology, we applied our techniques in multiple contexts across the studies. Table 1.1 shows an overview of the location of each study. Although Chapters 2 to 5 are all executed within the Emergency Room of the University Medical Center Utrecht (UMCU), Chapters 2 and 4 also include the Cerebrovascular Nursing Ward and the Cardiovascular Care Unit, respectively. In addition to being performed in the previously unexplored Gastroenterology Ward, Chapter 6 was performed in the Jeroen Bosch Ziekenhuis (JBZ), a regional hospital instead of an academic teaching hospital. As such, we show that the underlying techniques work in a variety of hospital settings.

1.4 Contributions

The main contributions of this thesis can be combined in three groups: 1) the *SWORD framework* that enables the discovery of workarounds through workaround mining, 2) the *theoretical* insights into how workarounds evolve, change, and spread through an organization, and 3) the *practical* applications of workaround mining for process improvement.

Table 1.1: The hospitals and wards per chapter.

Chapter	Location				Period of Data
	UMCU			JBZ	
	Emergency Room	Cardiovascular Nursing Ward	Critical Care Unit	Gastro-enterology Ward	
Chapter 2	✓	✓			1 week
Chapter 3	✓				1 month
Chapter 4	✓		✓		13 years
Chapter 5	✓				13 years
Chapter 6				✓	5 years

1.4.1 SWORD Framework

In Chapter 2, we present the novel SWORD framework. With this, we combine a quantitative pattern-based approach to find workaround candidates with qualitative interviews to evaluate findings, forming the basis of workaround mining. In Chapter 3, we extend the framework with machine learning and active learning to reduce the required effort for domain experts to discover workarounds.

By design, the SWORD framework only requires an event log to start discovering workarounds. It does not require specialized domain knowledge, such as process models like Outmazgin and Soffer (2014) or labeled traces like Weinzierl et al. (2022). As it is publicly available (Van der Waal et al., 2024d), it is also accessible for everyone. Outside of the hospital settings in this dissertation, the SWORD framework has been used to discover workarounds in two industrial cases (Löhr et al., 2024) and in an academic finance setting (Outmazgin et al., 2023).

1.4.2 Theory

On a theoretical level, we look into the emergence and evolution of workarounds over longer periods than previously possible with qualitative methods. With the new workaround mining techniques, we are able to identify workarounds over a period of 13 years as described in Chapter 4. Contrary to the various theories of how workarounds spread through 'bricolage' and planned change (Alter, 2014) or unstructured socialization (Safadi and Faraj, 2010), we observe that some workarounds are stable, while others slowly become more common over time, change temporarily, or are suddenly much more applied after specific system updates.

In Chapter 5, we further look into these changes from a resource perspective. Here, we show that the role of an actor very strongly influences how quickly they adopt workarounds into their way of working. This demonstrates that workaround behavior is strongly linked to how the different perspectives between actor roles can result in vastly different behavior as studied in routine dynamics (Dionysiou and Tsoukas, 2013; Danner-Schröder, 2020).

1.4.3 Practice

The main practical goal of workaround detection is to improve processes. Workarounds provide both an obstacle to be solved and a possible solution (Alter, 2014; Ejnefjäll and Ågerfalk, 2019). Therefore, a single workaround provides multiple perspectives for process improvement (Beerepoot and Van de Weerd, 2018; Soffer et al., 2023). However, as the detection of workarounds is very time-intensive (Outmazgin and Soffer, 2014; Beerepoot and Van de Weerd, 2018; Weinzierl et al., 2022), finding these improvement opportunities is, in practice, difficult.

Using a data-driven approach on already existing data, the SWORD framework allows workarounds to be discovered much faster. As this does not require excessive time from domain experts, it is possible to find workarounds with the goal of optimizing processes. In Chapter 6, we demonstrate how to use the SWORD framework to specifically discover actionable workarounds in a hospital setting. Although the framework is partially designed based on workarounds discovered in various healthcare organizations, its application is not limited to this sector. Without any further adjustments to the framework, Outmazgin et al. (2023) show that it can also be used to discover workarounds in a public academic college, which are then used to propose specific process improvements.

1.5 Thesis Structure

As can be seen in Figure 1.1, this thesis is centered around the three main contributions: the SWORD framework, Theory, and Practice. Chapters 2 and 3 discuss the construction and improvement of the SWORD framework. Chapters 4 and 5 show new theoretical insights into workarounds that could be obtained because of the improved discovery. Finally, Chapter 6 demonstrates the added value of Workaround Mining in a practical setting aimed at process improvement. These chapters contribute to our main research question: *How can we use event data stored in HISs to detect workarounds and analyze their emergence, evolution, and diffusion in the organization?*

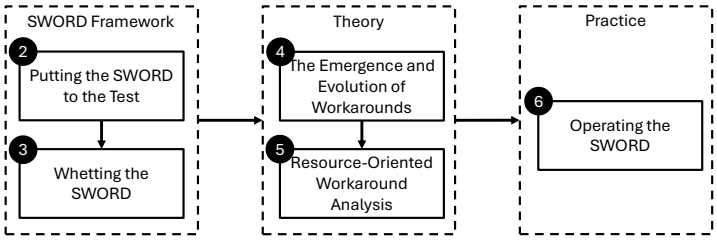


Figure 1.1: The structure of this thesis

1.5.1 Chapter 2 - Putting the SWORD to the Test

In the first chapter, we answer the question: *How can we discover workarounds using event logs?* Building on process mining literature and a large set of workarounds discovered in previous studies, we form the SWORD framework using 22 patterns that describe differences that can occur between traces. As these differences can have various explanations besides workarounds, the framework is complemented by short interviews that take place after each data analysis. We test the SWORD framework in a multiple case study of one week at the emergency room and the cerebrovascular nursing ward of the UMCU, where we discover 11 workaround types.

1.5.2 Chapter 3 - Whetting the SWORD

Building on the previous chapter, we ask the question: *How can we limit the need for expert involvement in workaround mining?* We combine the patterns from the SWORD framework using a logistic regression model, which we train through active learning. Instead of a large set of patterns, this results in a single ranking of all traces, indicating the likelihood of the trace containing any workaround. We evaluate the approach with a case study of one month at the emergency room of the UMCU. We discover seven new workaround types. Using strict rules per type to label 985 out of 1,463 traces as workarounds, we reach an F1 score of .75 with only a single one-hour interview as expert involvement.

1.5.3 Chapter 4 - The Emergence and Evolution of Workarounds

In Chapter 4, we consider the question: *How do workarounds emerge and evolve over time?* We leverage the seven workaround rules from the emergency room discovered in the previous study, as well as three rules from the critical care unit, to label all traces in the full data set spanning 13 years. We find that the prevalence of workarounds may change strongly over time: Some remain stable over long periods, while others consistently become more common. In addition, some workarounds may show sudden changes that may be temporary or permanent. We link these changes to various incidents, such as system updates.

1.5.4 Chapter 5 - Resource-Oriented Workaround Analysis

In the next chapter, we investigate the evolution of workarounds on a lower level with the question: *How does the adoption of workarounds vary between actor types?* To answer this question, we analyze one of the workaround types in the emergency room of the UMCU in detail. We find differences in the way nurses, physicians, and secretaries approach the same issue, both in the way of working, as well as their motivations for this. In line with theory about power dynamics, we find that high power roles like physicians quickly adopt a workaround they perceive as useful, whereas low power roles like secretaries adopt it primarily to support others.

1.5.5 Chapter 6 - Operating the SWORD

In the final chapter, we focus more on the practical applications of workaround mining with the question: *How can we find workarounds for process improvement?* We use previously unused data from interviews to allow for a more detailed analysis that is able to find workarounds in a more specific context. We show that adding additional interview rounds leads to more actionable workarounds. Instead of targeting a broad ward as in the previous chapters, we limit the study to a single, very common process in the gastroenterology ward of the JBZ. In this, we are able to reveal five workaround types. As the physician responsible for the process considered two of them unknown, concerning, and requiring attention, this demonstrates the added value of workaround mining for process improvement.

Chapter 2

Putting the SWORD to the Test: Finding Workarounds with Process Mining

Abstract

Workarounds, or deviations from standardized operating procedures, can indicate discrepancies between theory and practice in work processes. Traditionally, observations and interviews have been used to identify workarounds, but these methods can be time-consuming and may not capture all workarounds. This chapter presents the Semi-automated WORkaround Detection (SWORD) framework, which leverages event log traces to help process analysts identify workarounds. The framework is evaluated in a multiple-case study of two hospital departments. The results of the study indicate that with SWORD we were able to identify 11 unique workaround types, with limited knowledge about the actual processes. The framework thus supports the discovery of workarounds while minimizing the dependence on domain knowledge, which limits the time investment required by domain experts. The findings highlight the importance of leveraging technology to improve the detection of workarounds and to support process improvement efforts in organizations.

2.1 Introduction

Organizational processes can be streamlined by formalizing them into standard operating procedures. However, there are many reasons why workers may not be able to strictly follow these: the prescribed procedure may not account for all subtleties of practice (Beerepoot et al., 2021), the system may work incorrectly (Tucker, 2016), the software might not fit the procedure well (Ignatiadis and Nandhakumar, 2009), or it may not be possible to follow the procedure if an earlier step was not performed correctly (Beerepoot and Van de Weerd, 2018). Workers often solve these problems by finding a new path to reach their goals. When people

Originally published as:
van der Waal, W., van de Weerd, I., Beerepoot, I., Lu, X., Kappen, T., Haitjema, S., and Reijers, H. A. (2024a). Putting the SWORD to the Test: Finding Workarounds with Process Mining. *Business & Information Systems Engineering*, 67(2):171–190

deviate from the procedure to reach the intended goal, we consider this a workaround in line with the definition by Ejnefjäll and Ågerfalk (2019).

In some situations, a deviation from the procedure can be inefficient, costly, or outright dangerous (Ignatiadis and Nandhakumar, 2009; Ilie, 2013). In other situations, workarounds may be a practical solution to problems in the procedure and can be used to improve it (Alter, 2014; Beerepoot and Van de Weerd, 2018). Regardless of its effect, a workaround is an indication of a mismatch between a theoretical procedure and its practical execution. Knowledge of workarounds, thus, offers great potential for organizations to prevent problematic situations or improve procedures (Beerepoot and Van de Weerd, 2018). Using insights into workarounds, process owners can prevent the problem from occurring again or improve the corresponding procedure (Beerepoot and Van de Weerd, 2018). In addition, being able to identify workarounds structurally and comprehensively would allow for the monitoring of their emergence, diffusion, and evolution, which might enable process analysts to detect and respond to new workarounds quickly. This motivates our focus on workaround detection in this chapter.

Workarounds can be identified through qualitative methods, such as interviewing experts and observing their work (Beerepoot and Van de Weerd, 2018). Once identified, workarounds can be analyzed to gain insights into their mechanics and effects, as well as insights into the motivations of why people use them (Azad and King, 2008; Ignatiadis and Nandhakumar, 2009; Tucker, 2016). These methods tend to require a lot of time from both the researcher and the domain expert. Furthermore, process participants may not disclose all behavior or behave differently during observations (Weinzierl et al., 2020). A set of techniques that is particularly helpful in providing insights into the real execution of processes is *process mining*. Similar to how process mining is used to solve otherwise time-consuming problems (Van der Aalst et al., 2003), our focus is on the use of event logs and quantitative techniques.

In this chapter, we describe and evaluate a more quantitative approach for workaround detection: the Semi-automated WORkaround Detection (SWORD) framework. The SWORD framework uses 22 patterns to identify potential workarounds from an event log. If the log contains traces, which are lists of the activities with corresponding timestamps related to a single process instance, the detection of workarounds is performed in a highly automated fashion. Thirteen patterns in the framework can be applied without domain knowledge. Only after the potential workarounds have been identified, a qualitative analysis is conducted in which domain experts are interviewed to confirm the actual occurrence of workarounds. This makes the detection of workarounds less labor-intensive than finding workarounds through interviews and observation, but also less labor-intensive than conformance checking projects, in which the normative process needs to be identified with experts before the event data can be analyzed (Rozinat and Van der Aalst, 2008). Also, this non-obtrusive approach can be expected to partly mitigate the change in behavior that people may exhibit when they are aware of being observed (Holden, 2001).

We introduced the initial idea of the SWORD framework in an earlier study (Van der Waal et al., 2022), where we only demonstrated the potential of the framework on toy examples, but did not evaluate it further. In this work, we extend that work in three ways: 1) We describe the patterns that can be applied without domain knowledge in detail, which can help during implementation of the framework; 2) We expand our workaround analysis from previous studies by including workarounds from the retail domain; and 3) We replace the demonstration by a completely new evaluation. This is important to ensure that information systems artifacts find their way to application. Since workarounds are especially common in healthcare (Röder et al., 2014), we focus on this domain. We evaluate the framework in a multiple-case study, in which we applied the SWORD patterns to data from two hospital departments. By discussing the findings with domain experts, we discovered 11 workarounds. Compared to interviews, this method required limited time and effort from domain experts.

In the remainder of this chapter, we first clarify the term 'workaround' and how it relates

to other subdomains in process mining in Sect. 2.2. We explain our research method in Sect. 2.3. In Sect. 2.4, we describe the SWORD framework and how to apply it. We validate the framework with a multiple-case study in Sect. 2.5. Finally, we will reflect on the results in Sect. 2.6 and conclude in Sect. 2.7.

2.2 Related Work

When a worker tries to reach a certain goal, they may be blocked from following the official procedure. If they can solve this by working around this block to reach their intended goal, this is considered a workaround (Alter, 2014; Ejnefjäll and Ågerfalk, 2019). That means that we do not consider mistakes, deception, or fraud to be workarounds, because mistakes are not goal-oriented, and fraud and deception have a different goal than the procedure altogether. There may, however, be many intentional reasons to deviate from the procedure, such as fixing a problem that was not accounted for (Ignatiadis and Nandhakumar, 2009) or not being able to finish the task on time otherwise (Azad and King, 2008). For clarity, we distinguish between workaround instances and workaround types. The former is an individual occurrence of a workaround, the latter describes a group of similar instances that share characteristics. For example, we would consider a worker saving data to register it at a later time to be a single workaround instance that would fall under a 'batching' workaround type. An overview of these terms is also shown in Table 2.2.

There are many studies considering workaround detection in various domains, such as hospitals (Azad and King, 2008), nursing homes (Vogelsmeier et al., 2008), retail (Van de Weerd et al., 2019), and consulting agencies (Wolf and Beverungen, 2019). While most approaches aiming to find workarounds are qualitative, we are aware of two quantitative approaches. Outmazgin and Soffer (2014) define six workaround patterns, four of which are used to detect workarounds in event logs. These patterns require strict definitions of what is considered a workaround, such as a full process model or exact thresholds for activity duration. While these patterns can be used to detect instances of *known* workaround types, it is difficult to discover new types with them. Weinzierl et al. (2022) used a deep learning method to detect instances of seven workaround types. Neural networks require labeled training data, in this case, event log traces marked with 'workaround type x' or 'normative'. Since such a data set did not exist, the authors generated synthetic workarounds and added them to a real data set. With this approach, the neural network could recognize many of the synthetic workarounds. Since this method has only been tested on artificial data, it is unclear to what extent deep learning can be used to detect real workarounds. More importantly, in case that such a real dataset would be available to train a deep learning method, it would require the workaround types to be already known, which sidesteps the difficult task of evaluation within the domain as required for a real application. In addition, a high classification accuracy in neural networks does not necessarily mean high explainability (Ras et al., 2018), which is often important to apply results.

In the field of Process Mining, many conformance checking and deviation detection techniques have been proposed that make use of event log data to obtain insights into work processes. Conformance checking aims to evaluate to what extent executed processes conform to rules and regulations. As such, most of the conformance checking techniques require something to conform to, for example, a normative process model (Rozinat and Van der Aalst, 2008) or rules (Lazovik et al., 2004; Mahbub and Spanoudakis, 2004). Workarounds, however, may occur without these. Deviation detection techniques are used to discover instances that differ from the norm and generally do not require rules and regulations. Many of the techniques only take the control-flow perspective into account (Mannhardt et al., 2016). However, some workarounds can only be found by using the time, resource, and data perspectives (Beerepoot et al., 2021). We will continue this multi-perspective approach to

investigate if, in addition to recognizing workarounds, we can discover new workaround types using event logs.

2.3 Research Approach

Our study consists of two phases: a phase in which we define a list of workaround detection patterns and a phase in which we evaluate them. The two phases and their underlying activities are depicted in Fig. 2.1. In this section, we describe the approach followed in the two phases.

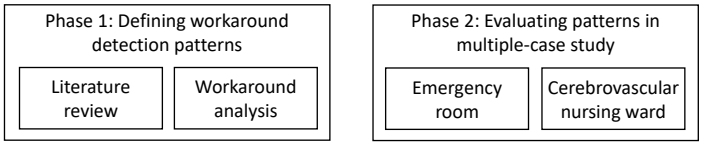


Figure 2.1: The phases of our research approach

2.3.1 Phase 1: Defining the Workaround Detection Patterns

As described in the previous section, detecting workarounds has similarities with several approaches in process mining, such as conformance checking. Both focus on differences between the intended and actual process. To investigate these approaches, we carried out a literature review to provide an overview of process mining approaches that can be applied to workaround detection.

After an initial literature search, we selected five literature reviews as the starting point for our study. The reviews focused on several (sometimes overlapping) process mining topics, namely conformance checking (Dunzer et al., 2019), process variant analysis (Taymouri et al., 2021), predictive process monitoring (Di Francescomarino et al., 2018), deviance mining (Nguyen et al., 2016), and process mining in healthcare (Batista and Solanas, 2018). The first four fields were selected based on their similarity to workaround mining. As such, techniques from these fields are more likely to be applicable in this field. The final review was included because of the high frequency of workarounds in that domain (Röder et al., 2014). In the next step, we selected relevant papers from these reviews. We used the following inclusion criteria: (1) the described approach focuses on the differences between process variants, from the control-flow, data, resource, or time perspective, and (2) the described approach uses event data for its analysis. In addition, we excluded supervised learning methods because they need a large number of labeled traces. It would require too much time to let an expert label all traces as a workaround or normative process. Note, also, that we keep to the main pattern in situations where slightly differing variants exist. For example, there are multiple versions of trace alignment; we do not distinguish between these here. Discovered papers were also searched for new references using reverse snowballing until no new patterns came up. Overall, we analyzed 37 papers in detail and included 12 papers in our final analysis, covering 16 patterns.

Second, we carried out an analysis of 29 workaround types from retail and 81 from healthcare that we observed in earlier studies. In the retail case study, the processes and workarounds in four stores of a large Dutch supermarket chain were studied through interviews and observations (Van de Weerd et al., 2019). This resulted in 29 workarounds that

were confirmed through a survey among 292 of the employees. For our analysis, we used the workaround descriptions, interview transcripts, and notes from the observations.

The healthcare workarounds were gathered from previous case studies that were carried out in five different healthcare organizations; a general hospital, two district hospitals, and two specialized care centers (Beerepoot et al., 2021). In these studies, the researchers were able to detect multiple instances of known workaround types using quantitative methods in a completely different top clinical hospital. This shows that we can expect similar behavior in other healthcare organizations. The workarounds from the healthcare studies were documented as 'workaround snapshots' and include a description of (1) the setting in which the workaround was found, (2) the workaround compared to the normative process, (3) a motivation, and (4) the expected effect of the workaround on cost, time, quality, and flexibility, and (5) transcripts of any interviews related to this workaround. To give an impression of the snapshots, we added an example snapshot in Table 1. For the transcripts, we only included a short excerpt.

Table 2.3 indicates which workarounds were found using literature or in either of the workaround analyses. Note that this framework is not exhaustive. As previous research is unlikely to have discovered all existing workarounds, more patterns can be added to the framework by investigating other departments, organizations, domains, or research fields. While not required, using additional domain knowledge for an initial extension of the framework may provide additional insights.

Table 2.1: An example of a complete workaround snapshot

Setting	The researcher is in the orthopedics and surgery ward. Several nurses are working on computers. Patients and other nurses sometimes walk by. The researcher observes the nurses and asks them what they are doing.
Workaround compared to the normative process	A patient arrives in the nursing ward after undergoing an operation. A physician is responsible for registering the medication information of this patient in the hospital information system but does not always do this. Upon arrival in the nursing ward, the nurses find out that the medication information is absent or incomplete. Nurses will try to call the physicians to recover this information. Often, the nurse has to wait until the physician is available to prescribe the medication. The nurse subsequently enters the medication information ad-hoc. This enables the nurse to administer the medication.
Motivation	Physicians are often busy and perform multiple operations per day. They are also not aware of how their workaround hinders the nursing practice. The nurse feels that the patient should no longer wait for the medication and decides to proceed with the workaround.
Effect	*Costs: negative.* Short-term positive because the nurse carries out tasks of the (assistant) physician. Long-term negative, since the process takes longer which increases costs (see time). *Time: negative.* In the short-term, it costs the (assistant) physician less time and the nurse extra time. In the long-term, it will take more time since an ad-hoc medication registration is only for one time and has also to be verified by a physician. In addition, it takes time to call the physician. *Quality: negative.* Not registering important information might lead to cases where the patient does not receive the right medication or not at the right time. *Flexibility: negative.* The nurse cannot administer the medication directly but has to call the physician. The physician is disturbed by calls during moments it might not be suitable.
Transcripts	Transcripts of interviews with one team leader and two nurses. Excerpt: "It takes the entire day to get it all in there before someone can get their medicine. You keep administering it ad-hoc . . . because ad-hoc is only for a single time. While if the physician would just verify it, it would be in there automatically and he would not need to be called all the time." (Nurse)

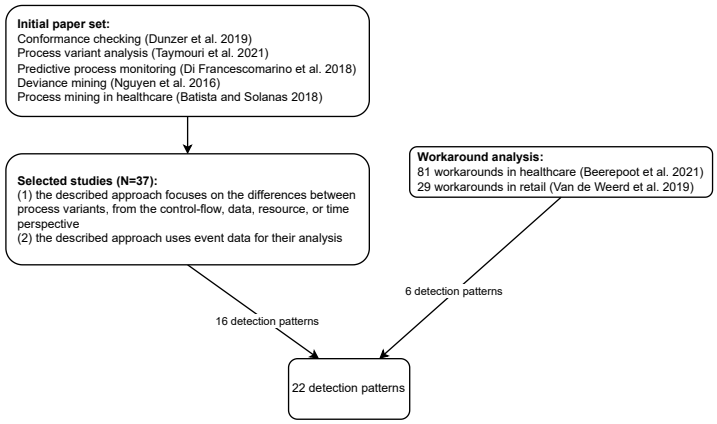


Figure 2.2: Phase 1 of our study: The construction of the patterns in the SWORD framework.

Using the comparison between the workaround and the normative process, we determined if and how each workaround could be monitored using (event log) data. Similar patterns were grouped together. This workaround analysis yielded another six patterns.

By combining the patterns from the literature study and the workaround analysis, we discovered 22 workaround detection patterns. Figure 2.2 shows a summary of this phase.

2.3.2 Phase 2: Evaluating Patterns in a Multiple-Case Study

In the second phase of our study, we evaluated the list of workaround detection patterns. We followed a two-step approach for this. First, we wanted to establish whether the data necessary to detect a pattern was stored in the Hospital Information System (HIS). Therefore, we analyzed the data structure of the tables in which the event data of the relevant processes and workarounds are stored. For each pattern, we searched for a table containing the required data to identify it. At this point, we considered if we could use that data to discover new workaround types, or if the pattern relied on specific knowledge and could only be used to monitor known workarounds. This activity resulted in an overview of data requirements per pattern.

Second, we conducted a multiple-case study at two different wards of the University Medical Center Utrecht (UMCU). In this study, we focus on discovering new workaround types, so we only consider the patterns that can be used for this. Since workarounds in healthcare are common (Röder et al., 2014), a case study at this hospital is a suitable way of evaluating the framework by providing the opportunity to both find known healthcare workaround instances (Beerepoot et al., 2021) and discover new workaround types. The details of the approach for this phase can be found in Sect. 2.5.

2.4 SWORD Framework

In this section, we present the SWORD framework. Table 2.2 shows the key terms we use. We start by analyzing the patterns we found using the literature review and workaround analyses. Specifically, we show how these patterns are applied in existing studies. We then take a further look into what data is required to use each pattern. Finally, we provide a detailed

Table 2.2: Explanation and examples of the key terms in this chapter

Key Term	Explanation	Example
Workaround Instance	An individual occurrence of a workaround	Nurse A prescribes medication ad-hoc instead of waiting for a physician
Workaround Type	A group of similar workaround instances that share characteristics	Nurses sometimes prescribe medication instead of waiting for a physician
Workaround Detection Pattern	A pattern that may be used to monitor or discover workarounds	An activity is executed by an unauthorized resource

description of the patterns that can be applied regardless of available domain knowledge and explain how to rank traces using the patterns.

2.4.1 Workaround Detection Patterns – Literature Review

Researchers in process mining fields other than workaround mining, such as conformance checking, deviance mining, predictive process monitoring, and process performance analysis, have developed patterns to detect differences between process variants. Since we can recognize different workarounds using varying perspectives, we have structured our review the same way. We start with the control-flow perspective and continue with the data, resource, and time perspective.

The control-flow perspective is used in most process mining fields, often by comparing traces to process models (Beerepoot et al., 2021; Borrego and Barba, 2014; Rebuge and Ferreira, 2012). Alternatively, some approaches analyze traces to check for activities that should never co-occur (Borrego and Barba, 2014) or that should occur close to specific others (Lu et al., 2016). Some individual events can already be interesting to monitor (Bolt et al., 2018; Bose and Van der Aalst, 2013; Nguyen et al., 2016). Repeated behavior can also be an indication something is going wrong. We can look at how often activities repeat (Borrego and Barba, 2014; Bose and Van der Aalst, 2013; Nguyen et al., 2016; Swennen et al., 2016) or whether a trace contains loops (Bose and Van der Aalst, 2013; Lo et al., 2009; Nguyen et al., 2016; Swennen et al., 2016; Wynn et al., 2017).

The data perspective is mostly used in the conformance checking field, where the values of data objects can be simply evaluated (Beerepoot et al., 2021; Borrego and Barba, 2014; Bose and Van der Aalst, 2013; Burattin et al., 2016; De Leoni and Van der Aalst, 2013; Taymouri et al., 2021). If they deviate too much, this can indicate unintended behavior in the trace. Alternatively, the exact value might not be important, but the value should not change during the trace (Borrego and Barba, 2014; Burattin et al., 2016; De Leoni and Van der Aalst, 2013).

The resource perspective shows similar patterns. Some activities should always be executed by a specific resource or the resource needs to stay the same during the trace (Beerepoot et al., 2021; De Leoni and Van der Aalst, 2013). For example, certain medications should only be prescribed by a physician. While not a workaround detection pattern in itself, earlier mentioned patterns can also be used using a resource as case ID. In this way, we can investigate the behavior of resources. For example, if we do so and notice a resource is repeating the same activity, something might be going wrong (Swennen et al., 2016).

We can find deviating patterns using the time perspective in multiple process mining fields. From a conformance checking viewpoint, some activities may need to be executed at a

specific time (De Leoni and Van der Aalst, 2013). Process performance analysis naturally takes time into account too: The time of activity since the start of the trace can predict the entire performance of the entire trace (Bolt et al., 2018). Multiple fields distinguish between process variants by looking at the time between activities (Burattin et al., 2016; De Leoni and Van der Aalst, 2013; Swennen et al., 2016; Wynn et al., 2017), the duration of a single activity (Swennen et al., 2016; Taymouri et al., 2021; Wynn et al., 2017), or the total time of a trace (Taymouri et al., 2021).

Each study listed in this section describes at least one aspect that is used to distinguish traces in the various process mining fields, which we translate into more general patterns. Table 2.3 shows an overview and description of all 22 workaround detection patterns, together with the studies they were described in and both workaround analyses we performed. As can be observed from the table, we found several workaround detection patterns in the healthcare data that have not been documented in literature. We did not find any new workaround detection patterns in the retail data.

Table 2.3: Workaround detection patterns

	Workaround detection pattern	Explanation	Reference	Health-care	Retail
Control-flow	Occurrence of an activity	A specific activity occurs	Bolt et al. (2018); Bose and Van der Aalst (2013); Nguyen et al. (2016)	X	X
	Occurrence of recurrent activity sequence	A recurrent activity sequence occurs within a trace	Bose and Van der Aalst (2013); Lo et al. (2009); Nguyen et al. (2016); Swennen et al. (2016); Wynn et al. (2017)	X	X
	Activity frequency out of bounds	There is a deviation in the frequency of an activity within a trace	Borrego and Barba (2014); Bose and Van der Aalst (2013); Nguyen et al. (2016); Swennen et al. (2016)	X	X
	Occurrence of activities in an order different from process model	The order of activities in a trace is other than in a predefined process model	Borrego and Barba (2014); Rebuge and Ferreira (2012)	X	X
	Occurrence of mutually exclusive activities	Specific activities occur that are mutually exclusive within a trace	Borrego and Barba (2014)		X
	Occurrence of unusual neighboring activities	An activity is directly followed by an activity other than usual	Lu et al. (2016)	X	
	Occurrence of directly repeating activity	An activity is immediately repeated within a trace		X	
	Missing occurrence of activity	A specific activity is missing in the trace		X	X

Continued on next page

Table 2.3: Workaround detection patterns (Continued)

	Workaround detection pattern	Explanation	Reference	Health-care	Retail
Data	Data object with value outside boundary	The value of a data object deviates from the usual values	Borrego and Barba (2014); Bose and Van der Aalst (2013); Burattin et al. (2016); De Leoni and Van der Aalst (2013); Taymouri et al. (2021)	X	X
	Change in value between events	Data values change unexpectedly between events	Borrego and Barba (2014); Burattin et al. (2016); De Leoni and Van der Aalst (2013)	X	X
	Specific information in free-text fields	Information is logged in free-text fields instead of dedicated fields		X	X
Resource	Activity executed by unauthorized resource	An activity is executed by a resource other than those authorized	De Leoni and Van der Aalst (2013)	X	
	Activities executed by multiple resources	Activities within the same trace are executed by multiple resources	De Leoni and Van der Aalst (2013)	X	X
	Activities executed by a single resource	Activities within the same trace are all executed by the same resource	Swennen et al. (2016)	X	
	Activity frequency out of bounds for a resource	There is a deviation in the frequency of an activity within a trace for one resource compared to other resources		X	X
	Value frequency out of bounds for a resource	There is a deviation in the frequency of a data value within a trace for one resource compared to other resources		X	
Time	Occurrence of activity outside of time period	An activity occurs outside of the usual time period	De Leoni and Van der Aalst (2013)	X	
	Delay between start of trace and activity is out of bounds	There is a deviation in the delay between the start of the trace and the time of an activity	Bolt et al. (2018)	X	
	Time between activities out of bounds	There is a deviation in the time between activities	Burattin et al. (2016); De Leoni and Van der Aalst (2013); Swennen et al. (2016); Wynn et al. (2017)	X	
	Duration of activity out of bounds	There is deviation in the duration of an activity	Swennen et al. (2016); Taymouri et al. (2021); Wynn et al. (2017)	X	
	Duration of trace out of bounds	There is a deviation in the duration of a trace	Taymouri et al. (2021)	X	
	Delay between event and logging is out of bounds	There is a deviation in the delay between the time of event and time of logging		X	

2.4.2 Workaround Detection Pattern Analysis

To investigate what data is required to detect each workaround detection pattern, we have used the data structure of HiX.[1] Using this specific HIS, we explored which columns in the data we would need to use to find each pattern. In total, there were 386 tables, where we mostly used one of the 12 tables containing information concerning measurements. While we used a specific implementation of an HIS, we used the data structure as a guide to investigating what information would be required. Other information systems structure data differently, but this does not affect the data requirements for the patterns. Therefore, this analysis is relevant for any other information system, even if it is unrelated to healthcare.

To use these patterns, it should always be clear to which trace an event belongs. Depending on the available data, we can use timestamps (e.g., by grouping events that are temporally close), specific activities (e.g., a trace starts with logging in and ends with logging out), or dedicated case IDs. Table 2.4 shows an overview of the required data. We distinguish between four data types that may be required: activity, time, data, and resource. Each pattern may require a different level of quality for these types:

- Some patterns need *specific* data. This data must be known beforehand. For example, a data field may require a certain value. Because of the huge number of data fields, it is not feasible to find these values automatically. We cannot find new workaround types with these patterns, only monitor discovered ones.
- We generally require *high-quality* data. Activity names should be distinguishable from each other, timestamps need to determine when an event happened, data needs to be complete, or we need to know which resource executed an event. The exact requirements differ per process. For example, in one case, "register measurement" is sufficient, but in another, we need the measurement type.
- For the time dimension, *low-quality* data may be sufficient if high-quality is not available. In that case, timestamps only need to be precise enough to determine a correct event order.
- Some data types are *not needed* to find a pattern. For example, if we investigate if the right resource performed an activity, we do not require timestamps.

[1] https://www.chipsoft.com/solutions/ (Last accessed 15 November 2023)

Table 2.4: Workaround detection patterns with the required data. Those marked with * can only be used to monitor known workaround types.

	Workaround detection patterns	Activity	Time	Data	Resource
Control-flow	**Occurrence of activity*	*Specific*	-	-	-
	Occurrence of recurrent activity sequence	High Quality	Low Quality	-	-
	Activity frequency out of bounds	High Quality	-	-	-
	**Occurrence of activities in an order different from process model*	*Specific+ Process Model*	*Low Quality*	-	-
	**Occurrence of mutually exclusive activities*	*Specific*	-	-	-
	**Occurrence of unusual neighboring activities*	*Specific*	*Low Quality*	-	-
	Occurrence of directly repeating activity	High Quality	Low Quality	-	-
	**Missing occurrence of activity*	*Specific*	-	-	-
Data	**Data object with value outside boundary*	-	-	*Specific*	-
	Change in value between events	-	Low Quality	High Quality	-
	**Specific information in free-text fields*	-	-	*Specific*	-
Resource	Activity executed by unauthorized resource	-	-	-	High Quality
	Activities executed by multiple resources	High Quality	Low Quality	-	High Quality
	Activities executed by a single resource	High Quality	Low Quality	-	High Quality
	**Activity frequency out of bounds for a resource*	*High Quality*	-	-	*Specific*
	**Value frequency out of bounds for a resource*	-	-	*High Quality*	*Specific*
Time	Occurrence of activity outside of time period	High Quality	High Quality	-	-
	Delay between start of trace and activity is out of bounds	High Quality	High Quality	-	-
	Time between activities out of bounds	High Quality	High Quality	-	-
	Duration of activity out of bounds	High Quality	High Quality	-	-
	Duration of trace out of bounds	-	High Quality	-	-
	Delay between event and logging out of bounds	-	High Quality	High Quality	-

2.4.3 Patterns

Nine of the workaround detection patterns presented in Table 2.4 require specific domain knowledge concerning the workaround to use. Therefore, these patterns can only be employed to find new instances of known workaround types, but not to discover new types. These patterns can be of use when monitoring known workarounds. For example, in the Emergency Room, a patient should only be discharged after at least two pain scores have been obtained. The '*Missing occurrence of activity*' may be applied to check if the workaround type occurs in a dataset by counting if the number of times this activity occurs in a trace is at least two. However, applying this pattern without the domain knowledge results in a long list of activities that are not always executed in the same frequency, which is perfectly fine in most cases. After all, not all patients require the same care. For this chapter, we are interested in discovering new workaround types, so we will further describe the 13 patterns that can be used for this. We provide an example of the patterns based on previously discovered workarounds and then further define them.

Pattern 1: Occurrence of Recurrent Activity Sequence

A repeating activity sequence signals that the same task is performed multiple times. While some tasks consisting of multiple activities are expected to be performed regularly, other sequences should not. For example, when a physician writes a discharge note for a patient, they first have to select if and what medication is still necessary and then print the letter. If the quantity of the medication is not correct on the printed letter, due to a system error, the physician needs to repeat the process, which results in the same activity sequence showing in the event log multiple times: $< \ldots, \textit{open note}, \textit{edit}, \textit{print}, \ldots, \textit{open note}, \textit{edit}, \textit{print}, \ldots >$.

Finding non-trivial recurrent activity sequences, such as $< \ldots, a, b, c, \ldots, a, b, c, \ldots >$ is an advanced pattern recognition problem that is beyond the scope of this chapter. We refer to other studies focusing on this exact question, such as one by Bose and Van der Aalst (2013).

Pattern 2: Activity Frequency out of Bounds

If an activity occurs either more or less often than usual, this may signal that people are working around a problem. For example, when patients have a health issue that requires a multidisciplinary approach, they should be linked to a specialist for each field. In some cases, a physician does not register their colleague as a co-practitioner but instead requests multiple peer consultations, which is usually used for incidental occurrences. A high frequency of peer consultations would make this observable in the data. Patients receive the required expertise, although not in an intended way.

We implement this pattern by counting how many times an activity occurs in each case. This results in a mean and standard deviation for every unique activity. If an activity frequency deviates strongly from the mean, either higher or lower, in a trace, it is considered more likely to contain a workaround.

Pattern 3: Occurrence of a Directly Repeated Activity

Activities that occur multiple times in succession indicate repeated behavior. This can indicate multiple people executing the same task or a single person repeatedly attempting it. For example, when patients get discharged from the hospital, they may get a discharge letter containing information for their follow-up. Nurses often have to wait until a physician has finished this letter. If they expect that a physician will not notify them after having written the letter, they repeatedly check the same page in the HIS, without performing any other

task within the system between the checks. This final activity could be observed as directly repeating in an event log.

This pattern can be checked by simply counting if an activity occurs directly after the same one in the (time-sorted) event log. For example, in a case with $< \ldots, x, x, \ldots, x, x, x, \ldots >$, we count three repetitions of x, i.e., an x directly succeeded by another x without any other event in between.

Pattern 4: Change in Value Between Events

Aside from the well-known activities, timestamps, and resources, there are often many data fields linked to a process. Many values here are expected to change over the course of a trace. For example, heart rate measurements can change every measurement. Other data tends to be more static; for example, a patient's height stays constant during hospital stays.

One example we have seen where data fields changed where they were expected to stay constant, concerned appointment times that were planned in advance. In some cases, due to either user or system errors, multiple patients were scheduled to visit the same physician at the same time. These could be solved by updating the times for the patients. If we monitor this data over time, we would see that the scheduled appointment times change during the workaround but would stay static otherwise.

We can use this pattern to identify workarounds for each data field by counting the number of times it changes within a single trace. Highly dynamic data fields would have a high mean on this count, while the mean of the less dynamic data fields would be close to zero. We can then evaluate a trace by comparing the number of times each data field changed to the average of that particular field.

Pattern 5: Activity Executed by Unauthorized Resource

Some activities should be executed by specific resources or resources with specific rights. For example, medication for a patient should generally be prescribed by a physician. In the ER, some nurses are also allowed to prescribe medication for emergency patients when there is no time to consult a physician. This authority to prescribe medication can also be used in non-urgent cases to save time. After all, if nurses do it themselves, they do not have to wait.

To investigate this pattern, we can use resource roles (such as nurse, physician, or administrator) and check which roles usually perform each activity. This results in a distribution of the roles for each unique activity. For a single trace, we can use this pattern by checking the probability that the executing resource for an activity is normal by comparing it to this distribution. If an activity is usually executed by a different resource type, we would further investigate the reason for this.

Pattern 6: Activities Executed by Multiple Resources & Pattern 7: Activities Executed by a Single Resource

When activities are performed by more resources than usual, it may be a sign that people are doing work for others. In Fig. 2.3, for example, nurses on a day shift are sometimes too busy to register the vital signs they measure. The night shift can support them by registering the measurements later, using the paper notes of the day shift. This results in two resources being involved in the trace, instead of one.

Conversely in Fig. 2.4, the involvement of fewer resources can also indicate workarounds. Before administering certain medications, two nurses should check the correct dosage. Since the check is done outside the system, they should register it manually. In some cases, one nurse registers both checks. While there are two resources involved, only one would be visible in the data.

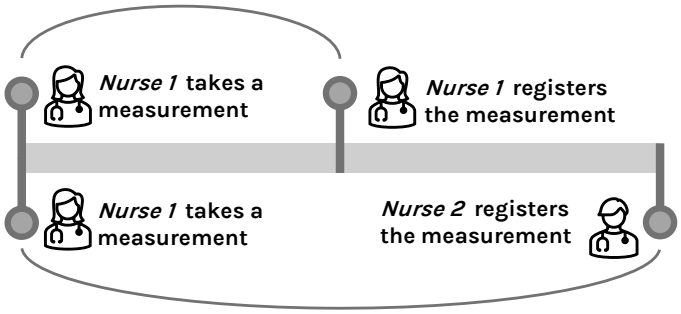


Figure 2.3: In the normative process (top), nurse 1 takes a measurement and registers it directly afterward. In the workaround (bottom) nurse 2 registers the measurement at a later time.

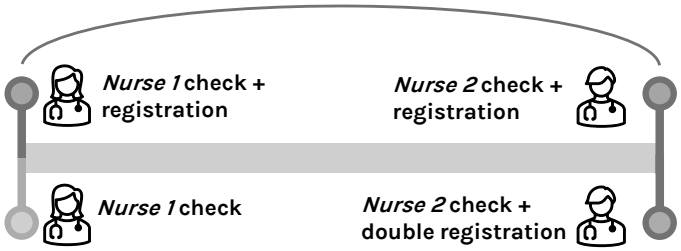


Figure 2.4: In the normative process (top), nurses 1 and 2 check the medication and register it themselves. In the workaround (bottom), nurse 1 checks the medication, but does not register it. then nurse 2 checks it and registers it for both nurses.

We check this pattern by counting the number of resources involved in a trace. This results in a mean and standard deviation of the number of resources involved in a trace. The higher the deviation from the mean for the number of resources in a trace, the more likely a workaround may be. Trace length differs strongly between patients. Some visit the hospital for a few hours, while others are there for weeks.

We observed an issue with these two patterns: if there are more events in a trace, more resources are generally involved. To account for this, we have a second instance for this pattern. We divide the number of resources by the number of events in the trace. In this way, we see how many resources were involved per activity. We are still able to determine a mean and standard deviation for each trace.

Pattern 8: Occurrence of Activity Outside of Time Period

Some activities are more common during certain times. If resources logged measurements directly after taking them, we would expect to see a relatively constant occurrence of the event over time. It would be higher during the day shift compared to the night shift, but within shifts, the changes in frequency would be limited. Similar to Fig. 2.3, a commonly used workaround is that while the measurements are taken on time, the logging is done in batches at a later time, especially during the end of a shift. If this happens frequently, we can see that logging an event occurs more often during those times.

Since this pattern is strongly based on specific workarounds occurring at the same time, we do not rank individual traces. Instead, we create a graph showing when specific activities

occur, such as Fig. 2.5. A taller-than-expected bar in the graph shows events occurring at a non-standard time.

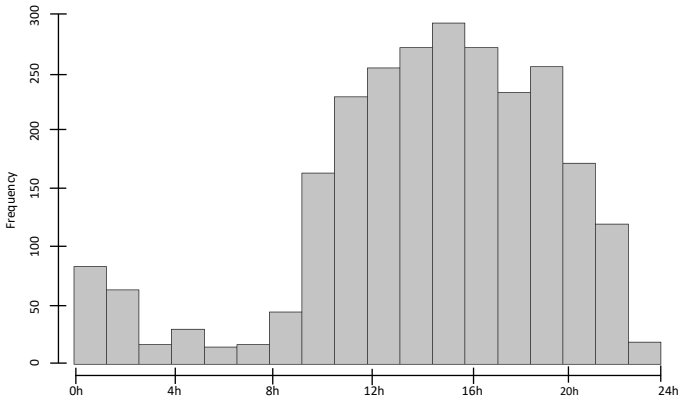


Figure 2.5: Timestamps of all orders in the ER. The distribution follows the usual patient load, except for the taller bar before 20h. This coincides with a shift change. In addition, there is a large difference between the final bar before 24h and the first bar before 0h, where 0h is a common time to round to if the exact time of measurement cannot be remembered.

Pattern 9: Delay Between Start of Trace and Activity is Out of Bounds

Some activities tend to occur at the same time since the start of the trace. For example, in Fig. 2.6, a patient in the emergency room should be seen by a nurse near the beginning of the visit. If this happens near the end of the trace, either the patient waited for a long time, or the registration was delayed, which could indicate a workaround.

We can detect this pattern by calculating the time since the first event in the trace for every unique activity. Based on these values we calculate the mean and standard deviation per activity. If the time since the start of a trace differs strongly for an event compared to the mean, we can suspect a workaround.

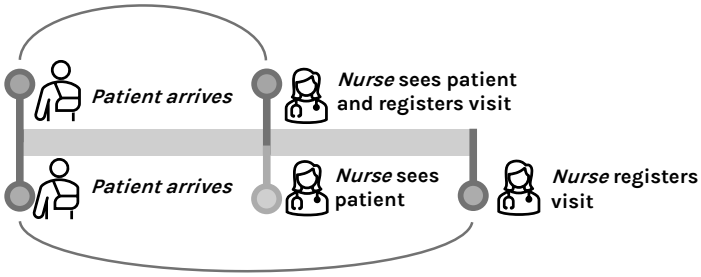


Figure 2.6: In the normative process (top), the nurse registers her visit to the patient immediately. In the workaround (bottom), she visits the patient at the same time but registers it later.

Pattern 10: Time Between Activities Out of Bounds

In many processes, the time between different events is comparable. In Fig. 2.4, for example, if multiple nurses check the medication, this takes a certain amount of time. In the workaround process, it is performed by a single resource, resulting in less time between the checks. With Pattern 6 we check the number of resources to detect the workaround, but with this pattern, we focus on the time between the medication checks.

We can detect this by calculating the time between every subsequent event in a trace. This results in a mean and standard deviation of the time between events for a process. If there is a long time between two events compared to the mean, this suggests a workaround.

Pattern 11: Duration of Activity Out of Bounds

To prevent data leaks, many organizations apply the Clear-Screen policy; if you are not at your desk, you should lock your screen or log out, so no one will be able to access your system. Logging in after returning is often seen as a waste of time, so people do not always log out when leaving their computers. When investigating the usage time of the system, this would show as a continuous multi-hour activity, as opposed to multiple shorter instances, separated by breaks.

If the data shows the duration of each activity, we can simply calculate the average and standard deviation of it for each unique event. If a trace then shows an exceptionally long or short duration for an activity, we are interested in investigating that trace further.

Pattern 12: Duration of Trace Out of Bounds

Many traces take roughly the same amount of time. If the duration of a full trace is significantly longer or shorter, this can indicate a workaround. We can see an example in Fig. 2.3: one nurse takes a measurement and another nurse registers it later. The normal process finishes shortly after the measurement, while the workaround does not finish until a later shift.

If there are only timestamps available for the start of events, we can detect this pattern by calculating the duration between the first and the last event for a trace. If ending timestamps are available too, the results will be more precise. In either case, this results in a mean and standard deviation of the duration of all traces. A trace that takes a lot more time compared to the mean points to a potential workaround.

Pattern 13: Delay Between Event and Logging is Out of Bounds

In most cases, an event should be logged shortly after it takes place according to official procedures. Otherwise, details about the event may be lost or other healthcare providers are not able to respond timely. Depending on the event, different time delays may be acceptable. For example, it is acceptable to delay a routine length measurement for a patient, because it does not serve as a warning score. A temperature measurement is more important to log quickly since a fever can indicate acute health issues.

If there is a dedicated timestamp available for both the measurement and the logging, we can determine the mean and standard deviation of the logging delay per activity. If an event has a logging delay that differs strongly from the mean, this is an indication of a workaround.

2.4.4 Ranking

In the final step of the application of the SWORD framework, we determine if there is a potential workaround in a trace using the patterns. For each pattern that can be applied to the event log in line with Table 2.4, we can create a ranking that determines how strongly each

trace deviates from the norm. Table 2.5 shows an example of such a ranking. Only patterns 1, 3, and 8 are exceptions to this approach.

Patterns 2, 5, 9, 11, and 13 result in a value for each event in a trace. We determine the mean for each unique activity and compare the value for each event in a trace to it using a Z-score. A Z-score expresses how many standard deviations an item deviates from the mean (Field, 2013). To determine the strongest deviating traces, we assign this maximum deviation score as the final score for this pattern to each trace. The scoring of pattern 4 is similar, although it results in a value for each data field instead of each activity.

Patterns 6, 7, 10, and 12 result in a value for each trace. We calculate the mean of the values of all traces. We can then determine the Z-score for each trace, which is the score for this pattern.

Patterns 1 and 3 detect repeated events. Since more repeats of a single event or event sequence do not necessarily increase the chance of a workaround, we do not use this frequency for ranking traces. However, if there are few repeats in the dataset, we can evaluate all traces where at least one repeat occurs. Instead, for Pattern 1, we can evaluate the activity sequences themselves and determine if specific variants indicate a workaround. For Pattern 3, we are only interested in the high repetitions of events. As such, we can use the raw score instead of a Z-score that would also highlight traces with few repetitions.

As mentioned before, we can only detect deviations in pattern 8 if we consider at which moments all events of a single activity occur. Since we work with timestamps for this, we cannot simply create a mean to compare to. Instead, we show the distribution of the timestamps as seen in Fig. 2.5.

We can create a ranking of the traces per pattern by sorting the related scores. A high-ranked trace may be more likely to contain a workaround, but since a workaround requires intent to reach a goal, which is rarely registered in an event log, it needs to be evaluated by an expert. Experts may not be able to provide the complete reasoning around the process; however, they are able to determine the intended end goal for a trace. The details of such an evaluation can be found in Sect. 2.5.

Table 2.5: Ranking of the traces using Z-scores. Sorted on pattern 2. Higher values for each trace indicate a stronger deviation from the mean.

Trace	Pattern 2	Pattern 5a	Pattern 5b	...
T-023	8.186	0.847	1.821	...
T-104	8.186	1.332	1.523	...
T-243	8.186	0.847	0.846	...
T-247	8.186	0.847	0.226	...
T-281	8.186	0.847	0.505	...
T-219	7.446	2.070	1.442	...
...	...	...	...	...

2.4.5 Application

To summarize, we can apply the framework to an event log by following Fig. 2.7. We start by selecting the applicable patterns (Sect. 2.4.2), and then we apply the selected patterns (Sect. 2.4.3), which results in rankings of the traces for all patterns (Sect. 2.4.4). This ranking is then evaluated by a domain expert, which we describe next in Sect. 2.5.



Figure 2.7: The required steps to apply the SWORD framework

2.5 Multiple-Case Study

2.5.1 Setting

The SWORD framework consists of both the patterns and the pattern selection process, in which the applicability of the patterns is directly dependent on the available data. To evaluate the framework, we performed a multiple-case study in two different departments of the same hospital: the emergency room (ER) and the cerebrovascular nursing ward. These departments handle very different processes. The ER mostly handles patients with strongly varying complaints for a short time and therefore needs to work with flexible processes, whereas patients on the ward get more specific care, often for a longer period. Because of these different process characteristics, we obtain a more complete view to evaluate the framework.

Both case studies were carried out at the University Medical Center Utrecht (UMCU), a Dutch academic hospital. The hospital takes care of more than 220,000 patients annually and has around 12,000 employees. Since this study makes use of patient data, it was evaluated and approved according to the Medical Research Ethics Committee (MREC) NedMec policy (research protocol number 22/1055).

In consultation with a data manager, we selected one week representative of typical hospital work. Thus we avoided the data being influenced by uncommon patient loads such as during holidays or centered around large-scale events like an outbreak of a specific disease.

2.5.2 Data Collection

The HIS stores all data in hundreds of separate tables. To identify the tables that contain the data that reflects the actions performed on patients in each department, we consulted a data manager of the UMCU with experience in extracting similar data from the HIS.

The ER uses a subsystem dedicated to track all its patients. This system assigns an identifier to each unique visit and keeps track of specific information, such as current patient location, the severity of the issue, and whether the patient was seen by a physician and a nurse. Especially the mutation log for this table provided a lot of events about the ER patients, as the table contained a timestamp for all activities and a resource for all except the 'seen by nurse/physician' activities. Since patients rarely stay longer than 24 hours in the ER, we selected all traces that started within the selected week.

The ward does not use a ward-specific system, but patients can be tracked in a so-called admissions table. This logs to which ward patients are admitted, as well as a timestamp and executing resource for admittance, discharge, and transfers between wards. By filtering for the related ward in this table, we were able to extract which patients were at the ward during the specific week. Only selecting patients with a stay fully within the selected week would not have given a representative view, since many patients on this ward stay for multiple weeks. Therefore, we selected all hospitalizations at the ward during the selected week.

For both the ER and the ward, the HIS uses additional tables to keep track of what happens with patients. After consulting the data manager, we used the measurements table and the orders table. The measurements table logs the values of clinical measurements, such as heart

rate and blood pressure, as well as pain scores, early warning scores, and other form-based measurements. For each measurement, the measurement time, the logging time, and the executing resource are registered. The orders table contains information about various orders. Among others, these can be orders for blood tests, radiology scans, or reminders for various other tasks. The logging time, planned time for the execution of the order, and resource are all registered. Note that to respect patient privacy, we do not use values or results from these tables.

We used Microsoft SQL to capture the event data from the HIS. Patients who object to their data being used for research purposes were tracked and removed from the data. One of the 32 patients in the ward and two of the 343 patients in the ER were filtered out for this reason. We pseudonymized the data as early as possible by assigning random unique values to all patient, resource, and other identifiers.

We combined the data into one event log for each department. The event log of the ER contained 7764 events over 341 unique cases; the log of the cerebrovascular nursing ward had 2148 events over 31 cases.

2.5.3 SWORD Framework

To apply the SWORD framework, we first determined the patterns that apply to the data. By investigating the event logs created in the previous section, we found that we could use nine of the patterns, namely '*Activity frequency out of bounds*', '*Occurrence of a directly repeated activity*', '*Event executed by too multiple resources*', '*Event executed by a single resource*', '*Occurrence of activity outside of time period*', '*Delay between start of trace and event is out of bounds*', '*Time between event is out of bounds*', '*Duration of trace is out of bounds*', and '*Delay between event and logging is out of bounds*'. The remaining patterns require data that was not available, such as a process model or start- and end-times for each activity.

Applying these patterns as described in Sect. 2.3 results in a ranking of all traces for each pattern, with the higher-ranking ones deviating more strongly from the norm. The higher-ranked traces were therefore considered to be more likely to contain a workaround. This ranking was used to structure the evaluation of workarounds in these traces, which we explain further in the next section.

2.5.4 Evaluation

To evaluate the top-ranked traces, we conducted semi-structured interviews with one expert per department. These were recorded and later transcribed. The ER expert was a specialist in acute internal medicine. For the cerebrovascular nursing ward, we interviewed a neurologist.

For each of the patterns, we followed a similar approach during the interview. If a pattern provided a ranking of the traces, we selected the top traces to present to the expert. For most patterns, this was an overview of the top 30. Since the ward had 31 patients in total, patterns that result in a single measure per trace, such as 'Duration of trace out of bounds', only had 31 measures. Since presenting the top 30 traces would give an overview of almost all data, instead of the most deviating traces, we presented the top 6 (20%) instead. For these top-ranked traces, we presented simple, understandable measures for each pattern as described in Sect. 2.3, namely the mean and standard deviation of the top-ranked traces compared to those of all traces. If the patterns did not provide a ranking, we presented the corresponding figures instead. The expert was asked to indicate if the metrics pointed to a workaround or something else. If necessary, there was a short exchange between the expert and the researcher to determine what was happening in the trace.

In addition, we presented the highest-ranked trace for each pattern to the experts. We explained what happened during that trace and asked whether there was a workaround

present in the trace. One trace was too long to present in this way (286 events). The trace concerned the delay between taking and logging measurements. Since they are often taken at the same time, we presented batches as a single event.

For example, Pattern 5 would be presented as follows: 'Seen by physician' usually occurs 36 minutes (SD = 101 minutes) after the patient arrives, but in the top-ranked traces, this happens after 71 hours (SD = 5 hours). The expert would then determine if the deviation was normal, a workaround, or something else.

2.5.5 Results

After ranking the traces using the patterns of the SWORD framework, we found at least one workaround for *each* pattern at the ER. With the nine patterns we were able to use, we found seven unique workaround types. In five of those workarounds, the overview of the top patterns was sufficient for the expert to recognize the behavior. They recognized the other two workarounds by inspecting the (pseudonymized) full, most-deviating trace. It should be noted that we detected three workaround types multiple times using different patterns. Table 2.6 shows an overview of the number of discovered workarounds per pattern.

For the ward, we found four unique workaround types. The expert recognized workaround behavior for three based on the overview of the top-ranked traces and one while investigating a most deviating trace.

Table 2.6: Detected workarounds per pattern

Type	Pattern	ER	Ward
Control-flow	Activity frequency out of bounds	1	
	Occurrence of a directly repeated activity	1	1
Resource	Event executed by too many or too few resources	1	
	Event executed by too many or too few resources (accounted for trace length)	1	
Time	Occurrence of activity outside of time period	1	1
	Delay between start of trace and event is out of bounds	4	
	Time between events out of bounds	2	1
	Duration of trace out of bounds	2	
	Delay between event and logging is out of bounds	1	1

The patterns in the framework are grouped into four main workaround dimensions (Mannhardt et al., 2016; Beerepoot et al., 2021): Control-flow, Data, Resource, and Time. To show that various workaround types can be discovered, we present an example for each dimension for both departments. Note that we could not use any of the Data-dimension patterns because of the requirements as described in Sect. 2.3.

Control-Flow Perspective

We detected three workaround types with our patterns from the control-flow perspective; two were found in the ER and one in the ward.

Example ER 'Activity frequency out of bounds' – The expert indicated that if the 'seen by nurse' activity did not occur in a trace, this pointed to a workaround. If a patient was not seen by a nurse, there are two options: (1) the patient left before this could happen, or (2) the patient was only administratively entered at the ER. Our expert indicated that the second option can happen when a patient is referred to a ward for treatment by an outside party but does not have a patient number. New patients are difficult to register during weekends, except in the ER, since unexpected patients are handled by design. The new patients are then administratively admitted through the ER to get a patient number and get assigned to the ward immediately. Although it looks like the patients visited the ER, they never do: The functionality of the ER is merely used to work around the obstacle of registering patients during weekends.

Example Ward 'Directly Repeated Event' – According to the expert, the 'save treatment overview' activity should not reoccur directly in normative traces. When a nurse registers or updates the treatment overview of a patient, they should write this text themselves. Instead, they often open an old overview to copy (parts of) the text. Opening and closing the overviews would register in the system as saving the document multiple times, which shows this workaround.

Resource Perspective

We detected two workaround types with patterns from the resource perspective, both in the ER.

Example ER 'Activities executed by multiple resources' – After seeing the high number of resources in the most extreme trace, the expert explained that this should not occur normally. In some cases, there are more nurses involved than usual in moving patients between treatment rooms in the ER and taking measurements. These data should be immediately registered. However, especially when the ER is very busy, often one nurse checks the measured data and another nurse registers them in the HIS.

Time Perspective

We detected 13 workaround types with patterns from the time perspective; 10 were found in the ER and three in the ward.

Example ER 'Duration of Trace out of Bounds' (Note that this example was also detected with other patterns) – The expert indicated that an extremely long trace with a duration of more than three days would most likely contain a workaround. Patients always leave the ER within 24 hours, usually even much faster. Before patients can be discharged in the system, there are checks that need to be performed, usually by a nurse. Some checks are not relevant for the illness, e.g., a pain score for a patient who has troubled breathing, but they are still enforced by the system before they can be officially discharged. If the ER is busy, nurses don't have time for these checks, but patients get sent home regardless. Since they cannot be removed from the system, staff register these patients in a field meant to keep track of patients that are sent somewhere else in the hospital with the label 'Administrative' until these checks can be logged at a later time. See Fig. 2.8 for a screenshot of the ER interface.

Example Ward 'Time between Activities out of Bounds' – The expert explained that a long interval between a patient's 'arrival at the ward' and 'change room' pointed to a workaround. When treatment has stopped because there is no chance of recovery for the patient, the patient is not unnecessarily disturbed and moved to a private room. When there are many patients on

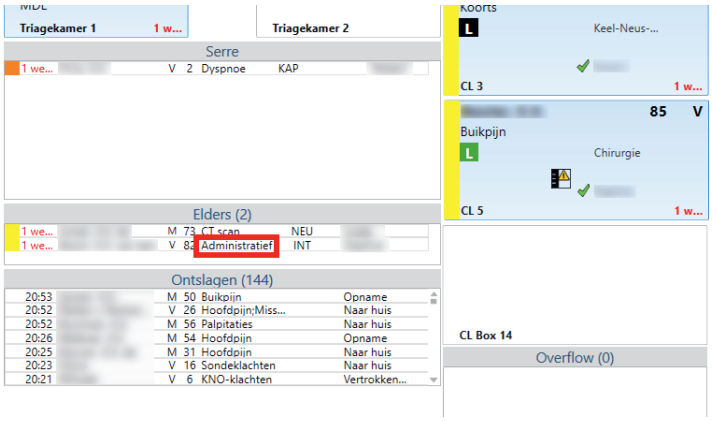


Figure 2.8: Part of the ER interface that keeps track of patients, with all identifiable information blurred. The left side is used to track patients outside of specific ER rooms. From top to bottom: 'Serre' is a *waiting room*. The field 'Elders' (*Somewhere else*) is used to track patients that are somewhere else in the hospital (e.g., for a CT scan). 'Ontslagen' keeps track of the patients that were *discharged*. The right side shows patients in ER rooms. If patients are sent home, but not all required checks are done, they are temporarily put in the 'Elders' field and marked with 'Administratief' (*Administrative*).

the ward and there are few private rooms available, patients can get assigned to special rooms that are not used at the time, for example, a cleanroom that minimizes chances of infection. This is not an issue, but a workaround may occur when this happens and the special room needs to be used. Then, the patient may be moved to another room anyway. While that room is free and all care can technically continue, this does disturb the terminal patient.

2.5.6 Threats to Validity

To ensure construct validity, we need to establish the correct operational methods for the concepts under study (Yin, 2013). The main threat to construct validity is related to the implementation of the patterns, namely that our code may not reflect the patterns sufficiently. Since the complexity of all implemented patterns is low, this threat is limited, but as a safety measure our code is publicly available.[2]

In this study, we try to generalize the use of the SWORD framework to general workaround discovery. A traditional threat to this type of generalization is that the results might only apply to the investigated organization (Yin, 2013). Most of the patterns in the framework are based on existing process mining literature. Besides healthcare, these patterns originate from many different domains, such as education, finance, industry, insurance, logistics, or public administration (Di Francescomarino et al., 2018; Taymouri et al., 2021; Nguyen et al., 2016) and should therefore be applicable to multiple domains.

The newly defined patterns of the SWORD framework are based on 1) workarounds found in *healthcare* organizations other than our case organization and 2) a set of workarounds found in four *retail* organizations. Since the same workarounds tend to occur in multiple healthcare organizations (Beerepoot et al., 2021), we believe the newly defined patterns should be easily applicable in the wide field of healthcare. However, we see several similarities between the investigated domains that challenge further generalizability. First, it is worth mentioning that

[2] https://git.science.uu.nl/w.g.waal/sword-framework

both fields contain strong hierarchical relationships between workers. Nurses and physicians in healthcare have a different standing, which is similar between management and salesperson roles in retail (Van de Weerd et al., 2019). As the role of power has been shown to influence the emergence of workarounds (Beerepoot et al., 2019a), it may not be possible to apply our results to fields where there is little hierarchy present. For a more complete view, domains with less hierarchical forms of organizing should be investigated, such as startups as well as certain research and development firms. Second, all case studies have been conducted in the Netherlands. Other, non-Western, cultures may have different cultural dimensions, such as Power Distance, Uncertainty Avoidance, or Individualism (Hofstede, 2011). Since these differences may affect workarounds, we cannot transfer outcomes to non-Western cultures at this point. Finally, we would like to note that none of the patterns in the time perspective were found in the workarounds from the retail domain. Although the framework in general would be applicable, it is possible that not all perspectives may be informative in every case.

There are also two differences between the domains that positively influence generalizability: 1) Retail work, specifically stocking and sales, tends to be relatively straightforward, while the processes in medical work may be highly complex (Safadi and Faraj, 2010). 2) As opposed to retail, the healthcare domain is known to be knowledge-intensive, which influences processes in general (Di Ciccio et al., 2012). Because of these differences, we find it likely that even though our case study is focused on healthcare, our results can be generalized to apply to multiple domains, within the constraints we mentioned.

To ensure that this study could be repeated, we saved all MS SQL queries that were used to extract the data from the database. With access to the database, these can be employed to extract the same datasets as used in this study, if the data has not been changed. As an extra safety measure, we also saved a frozen copy of the data. As mentioned before, the code to detect and rank the patterns in the datasets is publicly available. Finally, we recorded and transcribed both evaluation interviews.

2.5.7 Conclusion

In this multiple-case study, we applied nine patterns from the SWORD framework to data from the ER and a nursing ward of the UMCU. This resulted in a ranking of all traces for each pattern. We discussed the top-ranked traces with experts to evaluate the effectiveness of the framework. In total, we discovered 11 unique workaround types over the three investigated workaround perspectives.

2.6 Discussion

In this study, we described and evaluated the SWORD framework. We applied workaround detection patterns from the framework in a multiple-case study and found that many of the top-ranked traces indeed pointed to actual workarounds. In this section, we will discuss the scientific and practical implications of our work, as well as its limitations.

2.6.1 Scientific Implications

Compared to existing process mining approaches (Beerepoot et al., 2021; Outmazgin and Soffer, 2014), we have discovered completely new workaround types, as opposed to detecting new instances of known types or discovering new ones using qualitative methods. In our quantitative approach, we start by investigating already existing data. This mitigates the chance of people exhibiting non-standard behavior, which can happen during interviews

and observations (Holden, 2001). Thus, our data-driven approach reflects the real process more objectively than the traditional methods. Compared to a quantitative neural network approach (Weinzierl et al., 2020), our patterns have simpler data requirements. This is underlined by the data from the investigated departments. The event log for the ER contained 341 cases, while the log for the nursing ward contained only 31. With this study, we show that we can indeed discover workarounds with limited data. The framework also does not require data to be labeled as a workaround or normative process beforehand, which is useful in scenarios where access to this data is limited.

Our scientific contributions can be extended from the workaround domain to the wider process mining domain. While the purpose of our framework is to discover workarounds, the patterns in the framework may also be applied to other fields, such as conformance checking (Dunzer et al., 2019), process variant analysis (Taymouri et al., 2021), predictive process monitoring (Di Francescomarino et al., 2018), and deviance mining (Nguyen et al., 2016). Patterns that originate from one field may be effective in the other fields. In addition, newly defined patterns may indicate new directions to conform to, deviate from, or monitor, depending on the field. In short, whenever one is interested in finding differences between processes, if an event log is available, the patterns may be applied to discover these differences.

In addition, we offer a framework that can be used to uncover and explain, at least partly, the dynamics of routines. Routine dynamics (cf. Feldman and Pentland (2022)) have a strong tradition in qualitative research, but with the advances in data science, it is possible to analyze event data to uncover them. Specifically, we are able to uncover their performative aspects, i.e., the specific actions people take, which might differ from the ostensive aspects of routines, i.e., what people understand the process is (Feldman and Pentland, 2003). For example, workarounds can be used as an indication of resistance behavior (Wolf and Beverungen, 2019). The ability to uncover workarounds with our framework opens up opportunities to study resistance and change.

Our work also furthers the conceptualization of workarounds. While Ejnefjäll and Ågerfalk (2019) show that multiple definitions of the term "workaround" are used in existing literature, we argue that the *intent to reach the goal* is essential to distinguish between workarounds and other behavior. For example, we present a nurse prescribing ad-hoc medication to a patient as a workaround, because they do not have time to wait for a physician. In this case, the *goal* is the same as waiting, i.e., prescribing medication before administering it. However, the *path* is different. If a nurse would forget the prescription and administer the medication immediately, the goal would accidentally be different; It would now only be administering medication. While this is a deviation from the designed process, this would be considered a mistake instead. Finally, a nurse could also intentionally change the goal, by aiming to use specific medication as much as possible for financial gain. This hypothetical scenario would clearly be considered fraud instead of a workaround. If the goal is the same, but only the path differs, this should be considered a workaround, but once one attempts to reach a different goal, a different term should be used. Unintended deviations from the goal can be considered mistakes, but if the worker intentionally aims towards a different goal, this may be considered deception or even fraud.

2.6.2 Practical Implications

In addition to the scientific contributions, there are additional practical aspects of the framework: it is easy to apply and it requires limited effort from experts.

For process analysts, applying the SWORD framework is simple enough not to require domain knowledge until the evaluation step. Starting from an event log containing traces of only a single week, selecting and using the patterns is straightforward. Since most patterns result in a clear ranking of all traces, it is easy to select traces to evaluate with an expert.

This shows that the framework may also be applied outside of research in a more practical scenario, even if there is little data available.

An important motivation for the quantitative approach of the SWORD framework is the limited time that domain experts need to spend on it compared to the traditional qualitative approaches. The bulk of the time for this study was spent on data since both extraction and preparation are difficult for hospital data. After implementing the patterns, using them to identify workarounds only takes a few moments. The domain experts are not required until after this step. This has vastly less of an impact on them compared to observations and interviews. Where previous studies making use of these qualitative measures required between 20 and 30 hours of direct expert involvement for a single case study (Beerepoot and Van de Weerd, 2018; Beerepoot et al., 2019b), this study only required a single one-and-a-half-hour interview per expert for each case.

After this study, a process analyst from the UMCU expressed interest in investigating the implementation of the framework in their systems for more structural workaround detection. While an expression of interest is still far from actual application, it shows that this study, and workaround mining in general, is practically interesting to the healthcare sector. The newly discovered workarounds may be used to gain insights into processes and potentially improve them (Beerepoot and Van de Weerd, 2018).

2.6.3 Limitations

This study has the following limitations. First, the way we construct the ranking of the traces is somewhat limited. Not all traces that deviate from the norm necessarily point to workarounds. To the best of our knowledge, there is no previous work on this in workaround mining, so we opted for this simple ranking. Nevertheless, we described the procedure in detail in Sect. 2.4.4. In addition, we do not claim that the high-ranked traces indicate workarounds but instead use them as a starting point for the interviews.

Second, we have only interviewed a single expert per ward. While the physicians could easily judge traces involving the process they perform themselves, this was more difficult if the traces only concerned other resource types. Interviewing additional domain experts, such as nurses, would most likely result in more discovered workarounds. However, since we were more interested in investigating if we could discover workarounds and not so much in getting a complete view of their occurrence in a ward, the effect on this study is negligible.

Finally, it is important to note that this framework can only discover workarounds that are in the data. For example, from interviews, we know that nurses sometimes call the pharmacy to check if a prescription has arrived, instead of checking this in the system. Neither the call nor the check is registered in the data, so while interviews and observation can be used to detect these, there is no way to detect such workarounds using the framework or with any method that solely considers data from the system.

2.7 Conclusion

In this chapter, we have described the SWORD framework and evaluated it. With a multiple-case study in a healthcare setting, we have discovered workarounds in two departments. We show that without much domain knowledge, we can use the framework to identify traces that point to workaround behavior. This requires limited effort from experts and can both identify known and discover unknown workaround types.

There are multiple research directions we would like to explore in further work. First, we make use of simple Z-scores for the current ranking of the patterns. However, we found that for most patterns only deviations higher than the mean were reflected in the top-ranked

traces. While these deviations are interesting, we mostly found exceptionally high deviations and few low ones. The underlying data may be skewed because for most patterns there is a natural zero point, but no maximum value. For example, the duration of a trace can never be lower than zero, but a trace can always take longer. Using standard scores instead of Z-scores would already improve this, but more advanced statistics may be applicable too.

Second, in our study, we rank the likelihood of a workaround in the traces, but we only consider the top-ranked ones per pattern. We have seen that the patterns that lead to workarounds differ per ward, i.e., not all patterns point to workarounds in all contexts. For future work, we would like to look further into which patterns are the most interesting depending on the context.

Third, while we have investigated the 13 patterns that may be used without domain knowledge, we were only able to apply nine. The data we used for this study was insufficient to test the remaining four patterns, but we would like to investigate these patterns in the future. This would require a dataset containing data lacking in this study. As noted earlier, there may be differences in which patterns are applicable in different cases. Future research could investigate this both within and between domains.

Fourth, the SWORD framework uses the patterns as singular options to detect differences between traces, but we have seen that some workarounds can be detected with multiple patterns. To improve detection, we could use machine learning methods, such as classification (Osuna et al., 1997) or clustering (Ester et al., 1996). These can combine detection patterns, allowing us to effectively consider processes from multiple angles at the same time.

Finally, the ability to discover more workarounds in a limited time can be leveraged to further research into workarounds. Since workarounds discovered using the SWORD framework can be found in data, they can be monitored over time to see how they originated, spread through an organization, or evolved.

Chapter 3

Whetting the SWORD: Detecting Workarounds by Using Active Learning and Logistic Regression

Abstract

In many organizations, especially in healthcare, workers may work around prescribed procedures. Detecting these workarounds can give insights into difficulties concerning the procedures, which in turn can be used to improve them. Previous studies have shown that workarounds may be discovered from an event log using a set of predefined patterns such as the duration of a trace or the number of resources involved in one. However, domain experts may find it difficult to evaluate and monitor results if there are multiple patterns that indicate workarounds. Training a model that merges the features is often difficult because there are no available datasets covering workarounds. Labeling traces generally requires a lot of time from domain experts. In addition, this would have to be repeated for every new domain, company, or even department since the types of workarounds that occur may differ strongly between them. In this work, we propose to combine the features using a Logistic Regression model and train through Active Learning. In a case study at a hospital, we find that after training the model on only 10 to 15 traces, it stabilizes with an approximate F1 score of .75. This shows that we create and train a model that can detect workarounds well without requiring a large amount of labeled data or a lot of time from a domain expert.

3.1 Introduction

Workarounds are deviations from the prescribed procedure to reach an intended goal (Ejnefjäll and Ågerfalk, 2019; Van der Waal et al., 2022). Especially in hospital settings, they are a widespread phenomenon (Azad and King, 2008; Röder et al., 2014; Beerepoot et al., 2021; Van der Waal et al., 2022), but workarounds are also commonplace in other sectors, such as retail (Van de Weerd et al., 2019), or consulting agencies (Wolf and Beverungen, 2019). The

Originally published as:
van der Waal, W., van de Weerd, I., Haitjema, S., Kappen, T., and Reijers, H. A. (2024c). Whetting the SWORD: Detecting Workarounds by Using Active Learning and Logistic Regression. In *Proceedings of the 57th Hawaii International Conference on System Sciences*, pages 3687–3696

discovery of workarounds can help to identify problems in processes and systems. They contain valuable information on how the process could be improved.

Workarounds may be discovered through interviews and observations, but recent work shows that they can be detected in a more efficient and automated way by using event logs that contain sequences of events, also called traces. Various studies have identified patterns that indicate that a trace may contain workaround behavior. For example, certain activities may be skipped (Outmazgin and Soffer, 2014) or repeated (Alter, 2014; Weinzierl et al., 2022), certain process participants may perform tasks they are not authorized for (Outmazgin and Soffer, 2014; Beerepoot et al., 2021), or activities are executed at unlikely times (Van der Waal et al., 2022).

To classify a deviating trace as a workaround instead of user error or malicious behavior, we need to know the intent behind the diverging trace. Thus, even if patterns are used to indicate workaround candidates, these candidates need to be evaluated by an expert before workarounds can be used for process improvements (Bezemer et al., 2019). Previous work (Van der Waal et al., 2024a) proposes to create a ranking of the traces that result from applying a pattern, which reflects the likelihood that a trace is a workaround. This already helps to prioritize which traces should be evaluated. The idea of a ranking is that there is always one most interesting trace to evaluate. However, if there are multiple patterns taken into account, there are also multiple most interesting traces, which limits the effectiveness of the approach. Combining the patterns is not straightforward since every ranking follows a completely different distribution. Minimum and maximum scores may also differ strongly per pattern. Even when normalized between 0 and 1, means and standard deviations are rarely similar.

One way to combine patterns is to express them as scores that indicate how strong the pattern matches and use this as input features for machine learning models. However, training complex models usually requires a large dataset. Such a dataset containing many traces labeled as normative or workaround does not currently exist. Constructing one would require an expert to manually label hundreds of traces, which would not be feasible. Another challenge is that such a model would be specific to the domain and dependent on the workaround patterns used. Since workarounds can differ strongly per domain (Weinzierl et al., 2022), organization (Ejnefjäll and Ågerfalk, 2019), or even department (Van der Waal et al., 2024a), the model needs to be retrained from scratch for every new application, which would require a sizable time investment from interested process managers.

To address the problem of combining the patterns, as well as the data requirement and training issues that arise when we do, we propose to make efficient use of the limited time domain experts have available. We do so by using Active Learning (Settles, 2009) to train a Logistic Regression model that combines multiple features into a single score. We evaluate this approach with a case study at the emergency room of the University Medical Center Utrecht. We train the model during a single one-hour interview in which we identify several workaround types. To evaluate the performance of the model, we create a testing set by defining specific rules to detect all instances of these types in the event log. Based on the discovered types, we define rules that can be used to discover all instances of a specific workaround type in the data. Using this testing set, we show that our approach can distinguish traces as a normative or workaround after a short training session.

The primary contribution of this chapter is an approach to detect workarounds in event data. It is an extension of the SWORD framework (Van der Waal et al., 2022) to make it more applicable and useful. The extended approach involves Active Learning to ensure that the data requirements can be fulfilled *with limited queries for a domain expert*. By employing Logistic Regression it then becomes possible to *combine multiple features* to establish the occurrence of a workaround from various perspectives.

From a practical point of view, the model can be quickly tailored to a new application domain, because of to the little training time required. In addition, combining features makes

complex multi-feature frameworks much easier to use for real-life workaround detection by process managers. For scientific purposes, our approach requires very little time and effort from domain experts to both train and test a model. This can help future studies where no suitable datasets are available or possible to create. While our focus is on workaround detection, this approach may apply to other process mining fields where traces need to be classified by domain experts.

The remainder of the chapter is structured as follows: In Section 3.2 we explore the current state of the art concerning workaround mining, as well as common approaches in statistics and data science to create models using limited data. In Section 3.3 we further explain our approach to create, train, and evaluate our model. Then in Section 3.4, we evaluate the performance of our model using a case study. Finally, we discuss our findings and limitations in Section 3.5 and conclude the chapter in Section 3.6.

3.2 Related Work

3.2.1 Workaround Mining

There are two main goals for workaround mining: discovering new workaround types and detecting instances based on these types. Traditionally, workaround types can be discovered through qualitative measures, such as interviews and observations (Beerepoot and Van de Weerd, 2018). By defining a rule based on a discovered type, new instances of this type can be detected in new datasets (Beerepoot et al., 2021). In recent work, there has been an increase in attention to discovering workaround types using quantitative, data-driven methods, which we discuss here.

Outmazgin and Soffer (2014) define six patterns to mine workarounds from event logs. While the patterns are set up broadly, they require specific domain knowledge to detect in an event log. For example, the "Incompliance to role definition" pattern requires a known set of users that are allowed to perform a specific activity. As such, while this approach can find new instances of known workaround types, discovering completely new types may be difficult.

The SWORD framework makes use of 22 patterns (Van der Waal et al., 2022). Depending on the data requirements of a corresponding pattern, new instances of known workaround types can be detected, or new types may be discovered. Depending on the case, some patterns may turn out to be more useful than others. This can make it challenging for domain experts to evaluate the large number of patterns.

Weinzierl et al. (2022) use neural networks to detect workarounds using seven patterns defined in various other papers. While this approach manages to reach F1 scores up to 0.83, depending on the case, it has only been tested on synthetic data for now. In addition, training a neural network usually requires a large dataset, which is not always available for workaround mining.

Starting from an event log, Wijnhoven et al. (2023) first discover a process model. This model is then used to investigate 11 patterns that may indicate workarounds. While they show that this approach can be applied in practice, there is the underlying assumption that a process model is discoverable in the applied context. This may not hold in more complex scenarios, such as healthcare. For example, there may be theoretical procedures in the emergency room, but patients rarely follow the same path. To complicate matters even more, patients may arrive with multiple ailments that are treated in parallel, making process model discovery infeasible using the limited data available.

3.2.2 Data Requirements

Existing workaround detection tends to rely on readily labeled data, where the labels indicate the specific workarounds. When only very limited labeled data is available, common techniques such as Support Vector Machines or Decision Trees perform poorly (Sordo and Zeng, 2005) or require users to manually inspect many patterns and instances to identify workarounds. Active learning (Schein and Ungar, 2007; Settles, 2009) is a stream of algorithms in machine learning that can interactively query a user to optimize the learning rate and improve performance. As a result, Active Learning can handle initially unlabeled data and minimize the user's effort. Active Learning aims to do this by predicting the information gain for every possible query. Selecting the option with the highest expected gain would then lead to training a strong model using only limited data. Since expected information gain must be determined, Active Learning does restrict the possible models that may be used with it. Two common, straightforward classification models that have been used successfully with Active Learning are Naive Bayes and Logistic Regression.

The Naive Bayes classifiers (Lewis, 1998) use probabilistic reasoning to classify items by predicting the highest chance label based on a set of features as input. While they can reach high accuracy levels, they assume the individual features are independent. However, many of the features currently used in workaround mining are strongly related. For example, one indication of a workaround is the number of process participants involved in a trace. This may initially seem unrelated to another pattern that relates to the total duration of the trace. However, if a trace takes a long enough time, depending on the exact domain, it may be natural that more participants start being involved because of, for example, shift changes.

Logistic regression (Kleinbaum and Klein, 2010) is a regression technique that assigns a score between 0 and 1 to a set of features as input. Binary classification is simply done by setting a cut-off value. Any result above this value would be considered one option (e.g., a workaround), and the ones below it the other (e.g., a normative trace). The model is trained by adjusting the weight of each feature, making certain patterns either more or less important than the others to classify traces as workarounds. As opposed to the Naive Bayes classifier, Logistic Regression has no issues with strongly related features, making it very well suited for workaround mining. Thus, for this study, we will use Logistic Regression as the model and train it through Active Learning.

3.3 Approach

In this section, we explain the proposed approach to combine multiple workaround patterns into a single score. As shown in Figure 3.1, we start by describing how we construct the Logistic Regression model. Then, we discuss how we train this with domain experts using Active Learning. Finally, we talk about how we can evaluate the performance of the model.

3.3.1 Constructing the Model

To construct our Logistic Regression model, we start with an event log. As we discussed in Section 3.2, we use the SWORD framework (Van der Waal et al., 2022) to assign features to individual traces from such a log in this chapter. Table 3.1 shows two example patterns for each perspective from the full framework.

All 22 patterns in the SWORD framework have separate data requirements, meaning that depending on the data available in the event log, specific patterns may be applied. For example, activities executed by a different-than-usual resource, i.e., a process participant, can

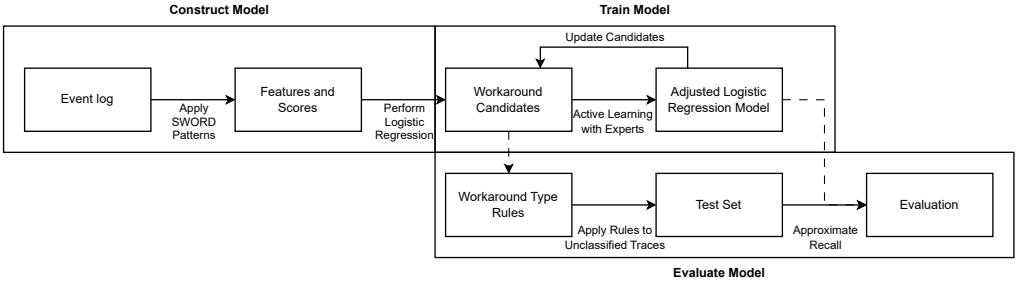


Figure 3.1: A schematic overview of how we create, train, and evaluate a model starting from an event log.

indicate a workaround, but to detect it, information about resources must be part of the event log. Based on these requirements, we select the patterns that can be used for the specific log. When applied, each pattern leads to one score per trace that reflects its deviation from the norm as established by the full set of traces. While these scores were used previously to create a ranking per pattern, in this study, we use the scores as input for the model.

As discussed in Section 3.2, we use Logistic Regression for actively detecting workarounds, since the features from the SWORD framework are not independent. As illustrated in Table 3.2, each trace and pattern is assigned a score. The set of scores for all traces and patterns is the input of the model. After training, each trace has a value between 0 and 1. When a trace is assigned a value close to 0, this indicates there is likely no workaround; when the value is close to 1, there is very likely a workaround. Since this is binary classification, the default cutoff point is 0.5. If a score is lower than 0.5, the trace will be classified as normative, otherwise as a workaround.

3.3.2 Training the Model

To optimize the limited data we can query from domain experts, we use Active Learning to train the model. Active Learning aims to optimize the training speed of a model by selecting the item that will most likely provide the highest information gain (Schein and Ungar, 2007). Depending on the model, the selection process may be very different. In the case of binary classification, the item with the highest uncertainty generally provides the most information gain (Settles, 2009). Since Logistic Regression assigns a score between 0 and 1 to items, continuously selecting the item that is closest to 0.5 will likely train the model fastest (Lewis and Catlett, 1994).

To train the Logistic Regression model, we interviewed a domain expert. In doing so, we present the trace with the highest expected information gain and ask whether it should be classified as normative or a workaround. Since the model is relatively simple, we can immediately use this classification to retrain the model. We then present the trace that offers the highest gain for this new model and ask the same question. We continue this process until the expert is out of time. Note that if multiple traces provide the same highest expected information gain, such as at the start when all traces are equally uncertain, we select one randomly.

Table 3.1: Two example patterns for each perspective in the SWORD framework (Van der Waal et al., 2022).

Detection pattern	Explanation
Control-flow perspective	
Occurrence of an activity	A specific activity occurs
Activity frequency out of bounds	There is a deviation in the frequency of an activity within a trace
Data perspective	
Data object with value outside boundary	The value of a data object deviates from the usual values
Change in value between events	Data values change unexpectedly between events
Resource perspective	
Activity executed by unauthorized resource	An activity is executed by a resource other than those authorized
Activities executed by a single resource	Activities within the same trace are all executed by the same resource
Time perspective	
Occurrence of activity outside of time period	An activity occurs outside of the usual time period
Delay between start of trace and activity is out of bounds	There is a deviation in the delay between the start of the trace and the time of an activity

3.3.3 Evaluating the Model

Common measures to quantify the performance of classification models are recall and precision (Powers, 2020). Recall compares the true positives and the false negatives to discover how many of the relevant items are found. Precision compares the number of true positive and false positive classifications to determine how many of the matches are relevant in the model. For further evaluation, precision and recall can be combined into an F_β score. The β can be varied to put more emphasis on either precision or recall. We follow a recent workaround mining study by weighing them equally using an F1 score (Weinzierl et al., 2022).

To determine a reasonable recall score, we need many labeled traces, but we face the same issue as before: There is no large dataset with labeled workarounds available. Creating such a dataset with domain experts would require them to evaluate a lot of traces, which would require a large time investment. In addition, since workarounds may differ strongly, it would have to be repeated for every new application (e.g., department). To solve this problem, we propose to use the discovered workaround types to generate labels for a test set.

When a workaround type is discovered, new instances of that type may be discovered using a rule specific to the type. Any trace that the expert indicates as a workaround for reasons that do not fall under an existing rule is considered a new type. For example, after taking measurements of a patient, a nurse may be required to log them as soon as possible. While a small delay can be acceptable, an expert may be able to explain that *a delay of more than an*

Table 3.2: The patterns scores and the score assigned by the Logistic Regression model.

Case ID	P1	P2	...	LR score
Case-0001	2.294017	1.672582	...	0.000
Case-0002	6.641274	0.643885	...	1.000
...	...	...	...	...
Case-0728	1.228111	0.384811	...	0.997
...	...	...	...	...
Case-1488	1.000074	0.899160	...	1.000
Case-1489	1.143432	0.899160	...	0.000

hour always means that the measurement was written down somewhere else before recording it into the system. We can define such a rule together with the expert as soon as they indicate a trace is a workaround for a new reason.

Using such rules, we can identify all instances of the discovered workaround types. Since these instances were not included during the training of the model, they can serve as a testing set containing traces that an expert would classify as a workaround. Note that this approach will not return all workarounds, only a complete set of the workaround types that were discovered beforehand. Even though this approach will most likely not result in the true recall value of the model, it does reflect all workarounds that the model should be able to classify. Thus, this approach serves as a reasonable performance approximation.

We approximate precision using the same data by assuming any trace that is not marked by any rule as a workaround is normative. In truth, these traces are not evaluated, so these traces may still contain workarounds. As such, this is a lower-bounded approximation for precision. Note that the undershoot is inconsistent and may differ strongly per iteration. Therefore, it may not be usable to investigate the change in performance while training the model.

3.4 Case Study

To test our approach, we performed a case study with the University Medical Center Utrecht (UMCU), a Dutch academic teaching hospital with around 12,000 employees, which cares for more than 220,000 patients annually. Before we started, our study was positively evaluated according to the Medical Research Ethics Committee (MREC) NedMec policy (research protocol number 22/1055). In the UMCU, HiX[1] is used as the Hospital Information System (HIS) to store patient data across 386 tables. In collaboration with a data manager, we extracted data covering one September for the Emergency Room (ER). This department has 19 treatment rooms and treats around 18,000 patients annually. With the data, we created an event log that contained 34,929 events for 1,489 completed cases. To evaluate our findings, we interviewed a specialist in acute internal medicine who works in the department and is directly involved with many of the patients. In this section, we will elaborate on how we used this dataset for testing our method.

[1] https://chipsoft.com/solutions

3.4.1 Constructing the Model

To construct a classification model for the ER, we started by creating a set of workaround indication features for each trace in the event log. In this study, we did this by applying the SWORD framework. Each pattern requires certain data to be available in the event log. For example, the "Event executed by a single resource" pattern requires a resource, i.e., a process participant, linked to each activity, whereas the "Event executed by an unauthorized resource" requires at least resource roles to be available. Taking such requirements into account, we could apply nine of the patterns of the SWORD framework:

- Activity frequency out of bounds
- Occurrence of a directly repeated activity
- Event executed by too multiple resources
- Event executed by a single resource
- Occurrence of activity outside of time period
- Delay between start of trace and event is out of bounds
- Time between event is out of bounds
- Duration of trace is out of bounds
- Delay between event and logging is out of bounds

The "Occurrence of a directly repeated activity" pattern resulted in the number of direct repeats that occurred within a trace. The remaining patterns each resulted in a Z-score that reflected the deviation from the norm per trace. These scores served as input for the Logistic Regression model.

3.4.2 Training the Model

We trained the model through Active Learning by presenting the trace with the highest expected information gain to the involved expert. In a single one-hour session, the expert evaluated 26 traces: 12 traces were considered normative, 14 contained at least one workaround. In total, we discovered seven workaround types, which can be seen in Table 3.3. When a new workaround type was discovered during the interview, we defined a rule to find similar instances of the type with the expert. These rules were used during the evaluation of the model for labeling test data.

3.4.3 Evaluating the Model

After training the model, we construct a test set by applying the expert-defined rules to the 1,463 unlabeled traces. Only if a rule showed that a workaround type occurred within the trace, it was marked as a workaround. In total, the test set contained 985 workaround traces. With this, we calculated the approximate precision and recall of the model after each training iteration, the results of which are shown in Figure 3.2.

The points on the left of the dotted line indicate the "warming-up" iterations. During these, the model was primarily trained on workarounds. This leads to it classifying all traces as

Table 3.3: The workaround types discovered at the ER, their explanation, and the rule to discover similar instances. Note that rules have a little slack time (10 - 15 minutes) to account for the order of activity registrations. For example, if an activity can only occur after a patient was seen by a nurse, the activity may be performed first and the checkmark placed directly afterward.

	Workaround type	Reasoning	Rule
1	Activities after discharge	Activities cannot be performed on a patient after they have left the ER. If this is observed in a trace, registration was late.	If an activity occurs after a patient has been discharged, there is a workaround.
2	Orders before the patient was seen by a physician and after initial intake	Orders can be placed in two ways: 1) a physician inspects the patient and places orders, or 2) the patient is expected and tests were ordered before their arrival.	If an order is placed more than 10 min before "Seen by physician", there is a workaround. Exception: Orders placed within 10 min of triage are normative.
3	Extra tests performed before a patient was seen by a physician	Nurses generally only order standard tests. A physician may determine that "extra tests" are required. These should naturally only occur after they saw the patient.	If the "extra test" activity occurs more than 10 min before "Seen by physician", there is a workaround.
4	Late triage registration	Triage (determining the urgency of care) always occurs at the start of a treatment process. It should also be registered quickly.	If the registration of the triage is performed after more than 15 min after the first activity by a nurse, there is a workaround.
5	Activities were executed before the patient was seen	Similar to the previous type, nurse activities, such as taking measurements, should only occur after they saw the patient	If triage, measurement, or order is registered more than 10 min before "Seen by nurse", there is a workaround.
6	Discharge without two pain scores	Any patient in the ER is asked twice how much pain they have to see if any needs to be addressed and to see if treatment helped.	If a patient was discharged with fewer than two registered pain scores, there is a workaround.
7	Immediate release after an x-ray	After an x-ray is performed, there is some time required to interpret and explain the results before a patient can be released.	If a patient was released less than 20 min after an x-ray was taken, there is a workaround.

a workaround and thus reaching a perfect recall score. Starting from iteration 5, we see a steep curve, showing quick improvement in performance, which stabilizes around 0.85 from iteration 11 onward.

The approximate precision barely changes over the training iterations. As stated earlier, this value is a lower bound on the actual precision, which may therefore be higher. Nevertheless, the minimum precision for the final model is about 0.67. This means that the corresponding F1 score for this model is 0.75.

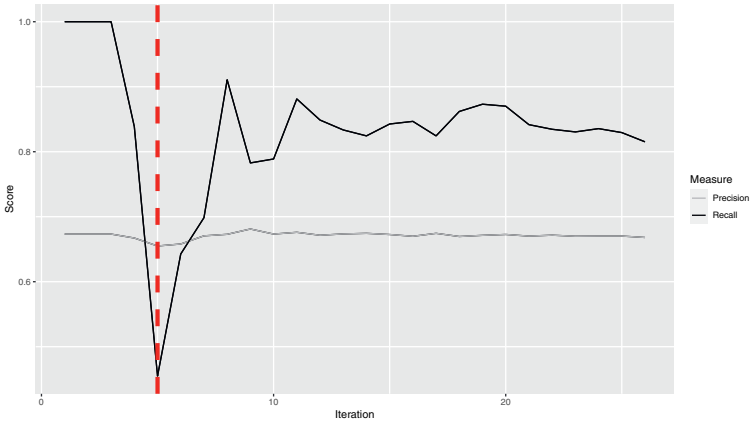


Figure 3.2: The approximate precision and recall of the model. The x-axis shows how many traces the model is trained on, the y-axis shows the approximate score for precision and recall. The focus is on the right side of the dotted line. The training iterations before this are part of the "warming-up" phase of the model.

To investigate the performance of the model on each workaround type, we determined the approximate recall per type for the final model. The results are in Table 3.4 with an overview of in which training iterations the workarounds occur. Since we only classify as "workaround" or "normative" and do not further distinguish between these types, we cannot determine an accompanying precision and F1 score. However, these scores can be used to compare performance between types. Note that within our context of workaround mining, recall is more important than precision; A normative trace that has been falsely identified as a workaround can quickly be dismissed by an expert, but a missed workaround may not be found at all.

3.5 Discussion

In this study, we combined multiple features that can indicate workarounds using Logistic Regression. To train the model with limited data availability and time involvement of a domain expert, we applied Active Learning. We evaluated this approach on real hospital data during a case study at the emergency room of the UMCU and found an approximate F1 score of 0.75 after training the model. In the remainder of this section, we discuss the implications and limitations of our work.

3.5.1 Implications

Our study has several implications we would like to discuss. First, while the recall scores per workaround type differ, all types are recognized with a score of at least 0.75. This shows that the model can be used even for strongly varying types of workarounds. In addition, we notice that a workaround type can already be recognized with few examples in the training

Table 3.4: The approximate recall of the final model per workaround type.

	Workaround Type	Occurs in Training Iteration	Approx. Recall
1	Activities after discharge	1, 7, 10	0.86
2	Orders before the patient was seen by a physician and after initial intake	2, 5, 8, 15, 17, 19	0.76
3	Extra tests performed before a patient was seen by a physician	19	0.81
4	Late triage registration	5, 11, 15, 17	0.75
5	Activities were executed before the patient was seen	5, 8, 11, 16, 17	0.76
6	Discharge without two pain scores	5, 24	0.81
7	Immediate release after an x-ray	3, 6	0.82

set. This is especially clear from workaround type 3, which occurs only in training iteration 19 but is recognized with a recall of 0.81. This is multiple rounds after the model already stabilizes. Even before the expert marked this new workaround type, it is already classified correctly. This shows that the model can also classify traces that contain workarounds that have not been discovered yet. Therefore, investigating high-ranked traces that do not fit any of the known workarounds can lead to new workarounds that may be interesting for process managers.

Second, there is one existing workaround mining study we can compare our results to. The performance of the neural network approach by Weinzierl et al. (2022) differs strongly with F1 scores ranging from 0.33 to 0.83. These scores vary strongly depending on the exact dataset used. Our score of 0.75 is slightly lower than their highest score. Since we used different data, these scores are not sufficient to conclude that either approach performs better in general, but we can see our performance is in the same range. In addition, it should be noted that the neural network approach has been tested on partially synthetic data, so it is unclear how it performs on real data. One advantage of our model is that it requires little data compared to a neural network. Where our model stabilizes after 11 instances, the neural network is trained on 7,000 to 16,000 instances. Even if our approach does not match the performance in the optimal case, the low data requirements make it easier to apply in a practical scenario.

Finally, while any feature-based model can be used to indicate traces of interest, domain experts are required for actual classification. To limit the time and effort requirements of these experts, we optimize training the model by only querying the most valuable traces using Active Learning. With nine features as input, the performance of the model still stabilizes after only 10 to 15 items, which we reached well within a single one-hour session with a domain expert. This shows that training a model with Active Learning to recognize workarounds requires little time and data, especially when compared to qualitative studies that spent between 20 and 30 hours on interviews and observations (Beerepoot and Van de Weerd, 2018; Beerepoot et al., 2019b). This makes it easier for interested process managers to apply our research in practice.

3.5.2 Limitations

As with most studies, there are limitations to this work, which we will discuss in this section. First, we can only approximate precision and recall in this study. We cannot calculate a true score for either because this would require labeling all workaround traces in the log. Recall varies strongly over time and follows a curve that shows a learning effect. However, precision varies very little. It is around 0.67 during all training iterations. We see two possible explanations for this behavior. 1) The model learns to recognize workarounds, but not normative traces, or 2) The test set, which consists of 67% workarounds, is not suitable to approximate precision. The second option would also explain why the F1 score during the warming-up phase is higher than after the model is trained. As to the impact of this on our research, we stated earlier that the approximate precision does provide a lower bound. This means the F1 score can still be used to give a lower bound for our results.

The second limitation is related to the use of event logs for the discovery of workarounds. We discover seven workaround types in this study but based on previous research that used different data from the same department (Van der Waal et al., 2024a), we are aware of more workarounds that occur there. Some of these workarounds could likely be discovered with a second interview, but other workarounds cannot be discovered at all through event logs. For example, a nurse that calls the pharmacy for current information concerning a medication order instead of looking it up in the HIS can never be detected if these activities are not logged. While we show that we can discover new workarounds using our approach, there is no guarantee we find all workarounds in the data.

Thirdly, the workarounds we can find with our approach depend heavily on the knowledge of the domain expert. We interviewed a physician who was aware of many of the workarounds that other workers, such as nurses and administrative personnel, employ. If an expert is less familiar with their day-to-day tasks, evaluating traces that primarily focus on their process will be difficult to classify. To put the results of our approach to use, expert involvement will always be of paramount importance conform Bezemer et al. (2019). Even though our approach minimizes experts' involvement, it still relies on them for evaluation, so the involved experts should be carefully selected.

Finally, we only evaluate a single ward in this study. Previous research shows that different contexts may have strong effects on performance (Weinzierl et al., 2022), which makes generalizing the results of detection models problematic. During this study, we also investigated data from the Critical Care Unit (CCU). Because the care is by definition critical in the department, there is a strong sense of urgency to follow the official procedures, which leads to few workarounds in this department.

3.6 Conclusion

With this study, we extend the field of quantitative workaround mining by creating a Logistic Regression model that combines multiple features that are known to be useful when discovering workarounds. To train such a model, we required labeled training data, which is not readily available. We make the most use of the experts' limited time by training the model through Active Learning. In this way, domain experts only need to evaluate the traces that provide the most information to the model. In a case study, we approximate the performance of the model by creating a test set based on the workaround types we discovered during a one-hour interview with a domain expert. Using this approximation, we find an F1 score of 0.75 after training the model. To conclude, we propose multiple directions for future research: the evaluation can be broadened, expert involvement can be improved, and there are alternatives to several choices for our approach.

First, as we state in the discussion, we only test this approach in one case. The evaluation can be improved by investigating different departments or even different domains. We expect that the performance can change depending on how common workarounds are and what types of workarounds occur most often. While we initially tried this at the CCU, the related evaluation was limited by the few workarounds that are applied here. Thus, it is important to note that this should be tested in a context where we can expect workarounds to occur.

Second, during the interview, the domain expert did not feel like the presented rank and standardized pattern scores were useful to determine what happened in a trace. Instead, they felt more comfortable looking through a full trace without focusing on any part indicated by the patterns. In future work, it may be more useful to give more specific guidance. For example, if an activity occurs relatively late in a trace, this specific deviation could be highlighted instead of the more abstract score for the whole set.

Finally, we suggest six alternatives to the choices we made: (1) Precision and recall may be approximated by querying a small number of random traces to the expert. We decided against this approach in this work because we lacked information concerning the ratio of workaround traces compared to normative traces. If this ratio is strongly skewed to either side, an approximation with too few traces would not be accurate. This made it difficult to determine how many traces should be labeled by the expert (Powers, 2020). In addition, it was unclear how much time an expert would require to label a trace. Since their time is limited, we opted for our rule-based evaluation. (2) To score the performance of our model, we used F1 scores to strike a balance between precision and recall. However, there are many alternatives, such as F scores (Verbeek, 2020) or Cohen's Kappa (Warrens, 2015). We decided to use F1 scores to be able to compare our approach with the existing workaround detection technique by Weinzierl et al. (2022). (3) Based on the discovered workaround types, we define rules with the expert to create a test set. The domain experts were confident about the rules they posed, but it may be interesting to look into rule-mining (Alman et al., 2020) instead to automate this process and reduce the expert's time investment even further. (4) For reasons we outline in this chapter, we decided to create our model based on Logistic Regression. More complex models, such as Support Vector Machines or Decision Trees can outperform simple models, but they may require more labeled data (Sordo and Zeng, 2005). (5) Both previous suggestions require more data than our current approach. In this chapter, we optimize data selection through Active Learning. Alternatively, we could construct a dataset through, for example, crowdsourcing with medical students. While the individual quality of the having students classify traces would be lower, the group may be able to provide quality labels in a larger quantity than possible for "expensive" physicians. (6) To create our model, we have only investigated supervised techniques. However, unsupervised methods may be an alternative option to circumvent the data requirements. Methods such as trace clustering (Song et al., 2009) may be used to train models without labeled data or it could be used as a starting point to further optimize evaluating traces with domain experts.

Chapter 4

The Emergence and Evolution of Workarounds

Abstract

Workarounds are often assumed to be quick fixes or stable practices. As most processes tend to change over time, we investigate this assumption. In an embedded case study, we analyze 13 years of process data from a large university teaching hospital. We investigate how the ratio of 10 workarounds changes over this period. We find that some workarounds remain stable over time, while others may change suddenly, temporarily, or continuously over time. We use historical documents to identify incidents that influence the use of workarounds and find that the dynamics of a workaround may switch between temporal categories. In contrast to earlier theories, we find that workers seem to adopt workarounds based on an individual need rather than a collective adoption after one person discovers it. In addition to these additions to current theory, we suggest future work that can answer currently open questions surrounding the evolution of workarounds.

4.1 Introduction

Management often prescribes specific ways of how a process should be executed. Even when workers try to follow these procedures, they may be blocked from doing so by a misaligned IT system, limited time, limited resources, or mistakes from colleagues. Such obstacles can be worked around by deviating from the prescribed procedure to still reach the original goal (Alter, 2014; Ejnefjäll and Ågerfalk, 2019).

By raising attention to perceived obstacles, workarounds indicate issues with the underlying process or system. While the identification of obstacles can be used directly to prevent the workaround from occurring, the resulting workaround may show advantages over the original process, such as being more efficient (Cresswell et al., 2016). During the redesign of procedures, organizations may choose to adopt such advantageous workarounds in their official workflow. However, as there could also be dangers linked to the workaround, prevention is sometimes preferable instead (Beerepoot and Van de Weerd, 2018). Whether a workaround is eventually

Based on:
van der Waal, W., van de Weerd, I., Beerepoot, I., and Reijers, H. A. (2024b). The Emergence and Evolution of Workarounds: A Study of Stability and Change. In *International Conference on Information Systems*

prevented or adopted, it is clear that it is beneficial for management to be aware of their existence.

Often, the temporality of workarounds is assumed to consist of either workarounds as quick fixes or workarounds leading to long-term stability (Ejnefjäll et al., 2023). This stability may be general or through planned institutionalization after a brief period. Some scholars explored a more dynamic perspective of workarounds and found, for example, that workarounds could evolve (Wong et al., 2022), gradually disappear (Zhou et al., 2011) and go through different stages of development (Safadi and Faraj, 2010; Davison et al., 2021). These studies are all conducted with qualitative research methods, which means that the workarounds could only be studied individually, over a relatively short period. Some quantitative studies have been conducted that show that it is possible to automatically identify many workarounds within a process (Outmazgin and Soffer, 2014; Van der Waal et al., 2024a; Weinzierl et al., 2022; Wijnhoven et al., 2023). However, the focus in these works is on finding workarounds in data at a single point in time, often a long time after the original execution of the process. As such, they do not provide insights into how workarounds may evolve. Given that processes tend to be very dynamic over time (Cloutier and Langley, 2020; Van de Ven and Poole, 1995), we assume that workarounds are more dynamic than currently described. With the present study, we address this research gap by investigating the evolution of workarounds over a sustained period. Understanding the dynamics of workarounds across time would further help manage workarounds, potentially even helping to predict or prevent harmful workarounds proactively.

In this chapter, we present the results of an embedded case study, in which we examine ten types of workarounds in an academic teaching hospital over a period of 13 years. By applying automated detection rules on a dataset containing process data from the hospital's information systems, we can detect specific IS-related incidents that alter the evolution of workarounds. While we observe that some workarounds are stable, we also find three additional categories that describe more dynamic behavior. Furthermore, individual workarounds may switch between these categories. After the background, method, and results of this study, we will further discuss how these categories relate to each other and how our new insights into the long-term changes in workarounds fit with current theories.

4.2 Theoretical Background

Workarounds are a specific type of deviation from prescribed processes. We adopt the definition of Alter (2014, p. 1044): "A workaround is a goal-driven adaptation, improvisation, or other change to one or more aspects of an existing work system to overcome, bypass, or minimize the impact of obstacles, exceptions, anomalies, mishaps, established practices, management expectations, or structural constraints that are perceived as preventing that work system or its participants from achieving a desired level of efficiency, effectiveness, or other organizational or personal goals." Notably, this definition does not assume any temporal nature of workarounds.

Some scholars describe workarounds as short-term fixes (Kobayashi et al., 2005; Debono et al., 2013). Others extend this view by making a distinction between workarounds as temporary practices and workarounds that are persistent over extended periods (cf. Bartelheimer et al. (2023)). As Ejnefjäll et al. (2023) explain, most workaround studies focusing on institutionalized workarounds, assume that these workarounds are stable. Some exceptions are related to workarounds that are connected to clear incidents. For example, the sudden emergence of a workaround due to the implementation of a new IS implementation (Alvarez, 2008; Boudreau and Robey, 2005; Koppel et al., 2008) or the sudden disappearance because the organization started to actively prevent it (Malaurent and Karanasios, 2020).

Several scholars have qualitatively analyzed the more dynamic and temporal nature of

workarounds, which is in line with the assumption that any process evolves over time (Cloutier and Langley, 2020; Van de Ven and Poole, 1995). For example, Zhou et al. (2011) show that workarounds gradually disappear as users get more familiar with the designed path. Workarounds may also become more common when they are shared among users, during the training of new staff, or during the redesign of processes (Safadi and Faraj, 2010; Weinzierl et al., 2022). These practices can spread the workaround through the organization, potentially even institutionalizing it Azad and King (2012); Choudrie and Zamani (2016); Cresswell et al. (2016); Davison et al. (2019), or serving as "generative mechanisms for bottom-up process innovation" (Bartelheimer et al., 2023). The dynamic nature of workarounds is further explored by Wong et al. (2022), who suggest that workarounds typically evolve unless they are simple one-time fixes that resolve a situation immediately.

Approaching workarounds as a dynamic, rather than static, phenomenon, aligns with the study of routines. Pentland and Feldman (2008, p. 236) define routines as "generative systems that produce repetitive, recognizable patterns of inter-dependent action carried out by multiple participants". Routines are recurring and collective and may contribute to organizational change (Becker, 2004). Since the inherent nature of these routines is highly dynamic, insights into how people work can be obtained by investigating various types of changes. Indeed, Zhou et al. (2011) note that workarounds can be 'transitory', potentially vanishing as new training is implemented or as changes are made to system usage policies. Delving into the subtleties of workaround evolution, Alter (2014) introduces the concept of 'bricolage', which captures not only immediate improvisations but also the gradual adaptation of routines over time. In a similar vein, Safadi and Faraj (2010), along with Davison et al. (2021), empirically explore the developmental trajectory of workarounds but focus only on the first phases of workaround evolution. Weinzierl et al. (2022) also study the early stages of workaround emergence but note that temporary workarounds may eventually be routinized and "stabilize as an established practice". This makes the lifecycle of these practices crucial to understanding.

As Goff et al. (2021) highlight, the temporal dimensions of workarounds remain an under-researched area. One problem with studying the long-term evolution of workarounds is that qualitative methods such as interviews and observations are less suitable for identifying changes in processes over periods longer than a few months. Some studies use process mining techniques to analyze IS data to overcome this problem. However, up to date, these studies focus only on the detection of workarounds (cf. Outmazgin and Soffer (2014); Beerepoot et al. (2021); Van der Waal et al. (2024a); Wijnhoven et al. (2023)). Although Weinzierl et al. (2022) emphasize that "workaround detection methods should consider the life-cycle stage of a workaround", they also solely focus on the emergence of workarounds in their study and not their evolution.

In conclusion, empirical studies predominantly distinguish between temporary and stable workarounds, with little attention given to other possible manifestations. While some researchers explore the evolving nature of workarounds, this has not been rigorously studied, partly due to the lack of appropriate analytical tools previously available. We argue that it is important to study the dynamic nature of workarounds since that would allow us to reveal how workarounds change over time. Unpacking how workarounds emerge and spread through the organization enables organizations to act proactive instead of reactive.

4.3 Method

In this embedded case study, we investigate the emergence and evolution of ten different types of workarounds analyzing process data from a Hospital Information System (HIS), supported by interviews and historical documentation concerning system updates. The case

study method is especially suitable for complex, contemporary phenomena within their natural context, where the phenomena and context are inseparable (Yin, 2013). Using an embedded case study design allows for an in-depth examination of specific workarounds within the context of two different wards over an extended period.

4.3.1 Context

We conduct our study at the University Medical Center Utrecht (UMCU). This Dutch academic teaching hospital currently employs more than 12,000 people and treats around 220,000 patients annually. The two wards in our study are the Emergency Room (ER) and the Cardiovascular Care Unit (CCU). We selected these two embedded cases because they share commonalities but also have differences such as the number of patients and type of activities that are conducted. The ER has 19 treatment rooms and treats around 18,000 patients annually. The CCU has 10 beds and treats around 1300 patients annually. Both wards use the same HIS: HiX[1].

4.3.2 Data collection and analysis

Our project was approved by the Medical Research Ethics Committee (MREC) NedMec (research protocol number 22/1055). While the data of all patients is collected for medical purposes, patients can opt out of their data being used for research. We removed patients who indicated this after each data collection step.

Table 4.1 shows the four steps of this study and the data used for each. While the first two steps are essential but not the focus of this study, we do not describe them in detail but instead refer to the previous work (Van der Waal et al., 2024c). Note that while we build upon previous work, our previous study was purely focused on workaround detection using data from only one month while we are currently applying the insights for a 13-year longitudinal study with a focus on the evolution of workarounds. The remainder of this section further details the steps of this study.

Steps 1 and 2

In step 1, we used a logistic regression model that combined multiple patterns indicative of workarounds to analyze one month of HIS data. At this point, the goal was to simplify and speed up workaround detection compared to interviews and observations, not to extract all workarounds from the data. With this model, we were able to identify multiple workarounds in the Emergency Room (ER) and the Cardiovascular Care Unit (CCU). In step 2, domain experts were consulted to determine precise rules indicating a workaround's occurrence in a log of a single process execution based on the discovered workarounds. While monitoring any process over long periods is difficult due to inherent process changes (Cloutier and Langley, 2020), these indicators are of such a general nature that they can be applied to strongly differing processes. The resulting workaround indication rules, shown in Table 4.2, enabled us to recognize ten types of workarounds in various process variants.

Step 3

In our current study, we apply the workaround indication rules to 13 years of HIS data of the UMCU. This allows us to investigate how the process changes over time with a clear focus on distinguishing workaround evolution from its normative counterpart.

[1] https://www.chipsoft.com/en

Table 4.1: Data Usage Overview

	Step	Goal	Data Source	Timespan
Focus of previous study	1	Discover workaround candidates	Initial HIS dataset containing two event logs covering 1,613 patients.	1 month
	2	Detect workarounds and determine strict indicator rules	1-hour interview with ER physician 1-hour interview with CCU physician	
Focus of current study	3	Monitor workaround indicators over time	Full HIS dataset, containing two event logs covering 249,633 patients.	13 years
	4	Identify process-altering incidents	Historical documents: presentations, meeting minutes, implementation reports	

Data collection: For every known workaround, we identify the process in which it occurs and extract the corresponding process data from the HIS. We use all data that was recorded in the system; in most cases, the earliest point the process is registered is June 2011 with the introduction of the first version of the current HIS, although for two processes the first logged data is in March 2018 which coincides with a large update of the HIS. To account for the start-up effect of the system, we did not use the data from the initial few months, as this consisted largely of tests related to the implementation, instead of real usage. For the processes started in 2011, we only used data starting from January 2012, for the process in 2018, we started this in July 2018. In all cases, our cutoff point where we stopped the data collection was January 1, 2024. After formatting this data into event logs (Van der Aalst et al., 2007), we apply the workaround indication rules to mark all workarounds as such using R . For example, workaround indicator ER1 states that any activity occurring after a patient has been discharged is a clear sign of a workaround. The rationale is that if a patient is not in the hospital anymore, no additional tests can be conducted on the patient. The application of this rule is therefore straightforward. As the activities in an event log are ordered chronologically, any event that occurs at any time after the discharge clearly indicates a workaround. In the ER we analyze seven workarounds; in the CCU three. For each workaround, Table 4.2 lists a short description, the indicator rule, and the related process data in the full HIS dataset.

Data analysis: The application of the workaround indicator rules allows us to conduct a synchronic analysis of process data (Berente et al., 2019) in which we analyze when the workarounds occur. After this initial analysis, we conduct a diachronic analysis in which we focus on how action patterns (in this study, workarounds) change over time (Barley, 1990; Berente et al., 2019; Pentland et al., 2021). Specifically, we investigate when and how the ratios of workarounds to normative behavior change. In R, we calculate the ratio of workarounds to normative process executions and present the results in graph form. We use visual inspection of the graphs to identify specific months of interest and analyze changes.

Table 4.2: Workaround Indicators

Code	Workaround indicator	Rule: The case is a workaround if ...	Slack	Set of cases	Number of cases	Mean case length
ER1	Events after Discharge	any event occurred after discharge. Normative otherwise.	-	All ER patients who were discharged	237,997	13
ER2	Orders before Physician Visit	an order was placed before the patient was seen by a physician. Normative otherwise. Exception: Also, normative if an (emergency) order is placed during Triage.	10 min	All ER orders where the patient was seen by a Physician	1,385,716	2
ER3	Follow-up test before Physician Visit	a follow-up test was ordered before the patient was seen by a physician. Normative otherwise.	10 min	All ER patients with at least one follow-up test where the patient was seen by a physician	22,327	2
ER4	Measurement, Order, Triage, or Treatment before nurse visit	a measurement, order, triage, or treatment was placed or executed before the patient was seen by a nurse. Normative otherwise.	10 min	All ER patients with any order, measurement, triage, or treatment event where the patient was seen by a nurse	227,516	17
ER5	Triage late in the trace	triage occurs after any other event. Normative otherwise.	15 min	All ER patients with a triage event	239,194	16
ER6	Few pain scores before Discharge	a patient was discharged with fewer than 2 pain scores. Normative otherwise.	-	All ER patients who were discharged	241,361	3
ER7	X-ray too close to Discharge	an X-ray was logged right before discharge. Normative otherwise.	20 min	All ER X-rays where the patient was also discharged	64,642	2
CCU1	Late Logging of Measurement	the logging of the measurement is long after the measurement time. Normative otherwise.	30 min	All CCU measurements	221,647	2
CCU2	Late Logging of Order	the logging of the order is long after the order time. Normative otherwise.	60 min	All CCU orders	489,623	2
CCU3	Late Discharge	the discharge is registered long after the previous event. Normative otherwise.	12 hrs	All CCU patients who were discharged	8,272	92

Step 4

In the last step of the study, we link incidents such as system updates, reorganizations, and new equipment implementations to changes in the ratios of workarounds to normative behavior.

Data collection. Since some significant changes in the process occurred more than ten years ago, experts are unlikely to remember incidents that may have contributed to the change. Therefore, in the last step of our study, we investigate them further using historical documents, such as presentation slides, meeting minutes, and implementation reports, stored in the UMCU database.

Data analysis. With historical documents, we use a diachronic analysis to identify the related incidents. The link between the identified incidents and the observed change allows us to further develop theory of the role of process-altering incidents in the emergence and evolution of workarounds.

4.4 Results

To investigate how workarounds change over time, we analyze the ratios of workarounds to normative process executions. After this, we observe four categories of how this ratio may evolve. Note that this category is not fixed for a single workaround, but it may change.

1. Stable: the ratio does not change strongly over time,
2. Gradual: Over time, the ratio changes slowly but steadily,
3. Temporary: At a specific point in time, the share of workarounds changes but after a while, it reverts to its original level, and
4. Sudden: At a specific point in time, the ratio completely changes.

For each moment the evolution category changes within a single workaround, we are able to identify an incident that facilitated this change by investigating historical documents.

A brief overview of our findings can be found in Figure 4.1. Following our research approach, we first discuss our findings regarding the evolution of the workarounds and the identification of specific months during which this evolution changed. After this, we delve deeper into these dates and discuss the underlying incidents.

4.4.1 Evolution

In the following, we present the evolution of the ratios of workarounds to normative process executions in graph form. In every graph, the grey bars represent the number of process executions. The red line depicts the ratio of workarounds to normative process executions. For readability, the ratio lines are presented using a sliding window of size 5; The value of every month is calculated by averaging the ratio of the month itself, as well as the two months before and after it. This smoothens most singular peaks and makes the graphs less jagged.

Stable

Figures 4.2 to 4.5 show processes where the ratio of workarounds to normative is generally stable. There may be some minor fluctuations, but these are to be expected in a hospital environment where the work itself is inherently dynamic.

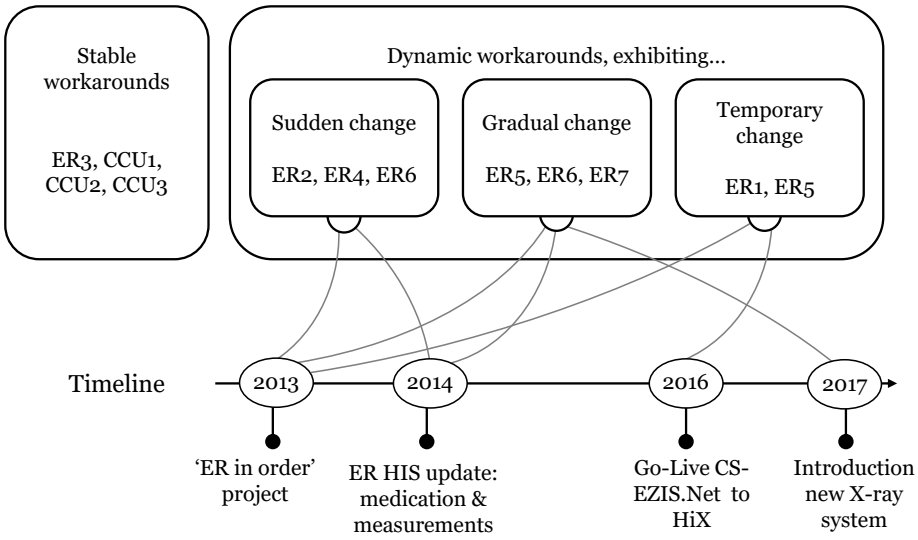


Figure 4.1: Categorization of workarounds with links to accompanying incidents

Gradual Change

Figure 4.6 only shows a gradual change in the workaround ratio between 2011 and 2017. The occurrence of the general process peaks at that time, but the occurrence of workarounds stabilizes.

Temporary Change

Figures 4.7 and 4.8 show a temporary increase in workarounds. In Figure 4.7, the overall process is generally stable, except for a temporary increase starting in 2016. In 2018, the ratio is back at a similar level as before the increase. In Figure 4.8, the general workaround version of the process is increasing continuously but the temporary increase shows a clear jump in the trend between 2013 and 2015. Without this peak, the trend would be strongly linear.

Sudden Change

Figures 4.9, 4.10, and 4.11 show very sudden changes in the workaround ratio. Apart from these changes, Figure 4.9 and Figure 4.10 are primarily stable. Figure 4.11 is very stable before the change but shows a continuous increase after it.

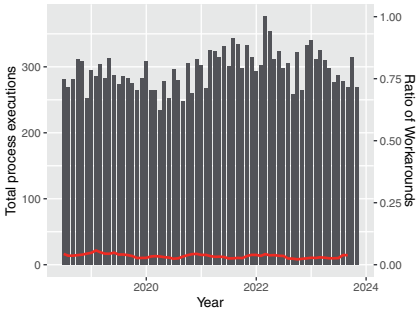


Figure 4.2: ER3 - Follow-up tests before Physician Visit. This process fluctuates slightly but is generally stable at around 3% workarounds.

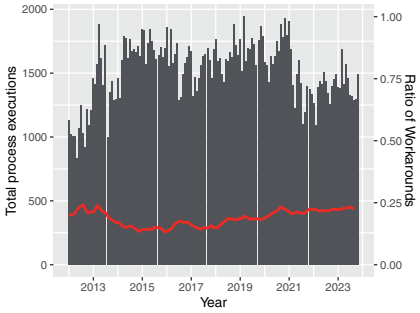


Figure 4.3: CCU1 - Late Logging of Measurement. The process is mostly stable around 20% workarounds.

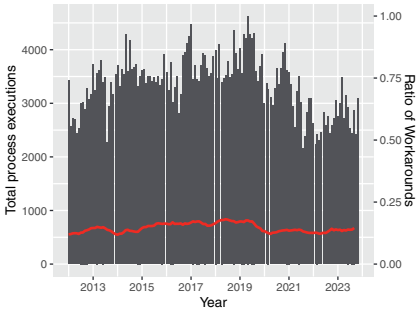


Figure 4.4: CCU2 - Late Logging of Order. This process is stable at 15% workarounds.

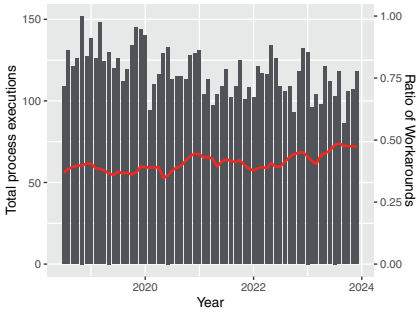


Figure 4.5: CCU3 - Late discharge. This process is stable at 40% workarounds.

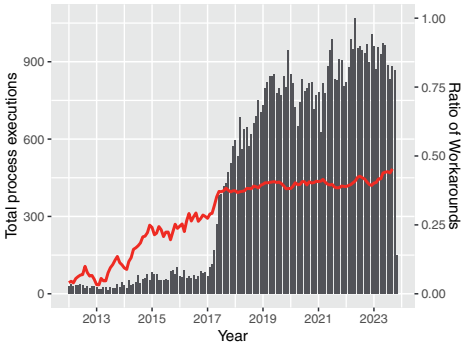


Figure 4.6: ER7 - X-ray close to discharge. In the first part, the ratio of workarounds continuously grows from 10% to 40% but it stabilizes in 3-2017 when the process itself becomes more much more common.

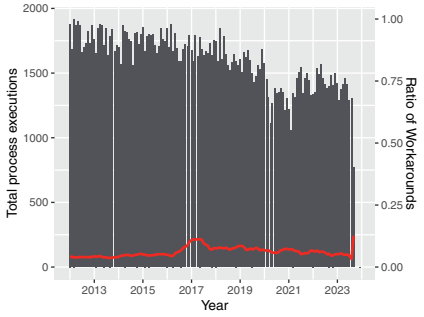


Figure 4.7: ER1 - Events after Discharge. There is a small temporary increase in workarounds starting in 4-2016. Before and after, the ratio of workarounds is stable at 5%.

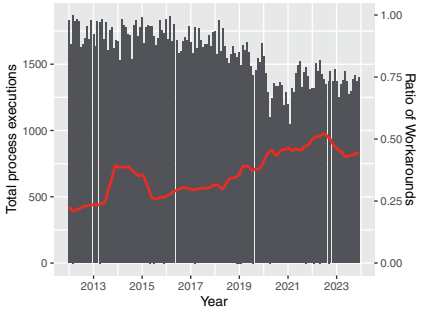


Figure 4.8: ER5 - Triage late in trace. There is a temporary increase starting at 7-2014. Without it, the ratio of workarounds linearly grows from 25% to 50%.

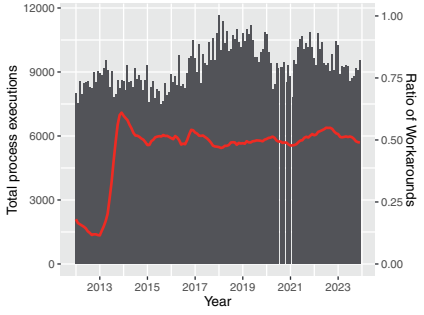


Figure 4.9: ER2 - Orders before physician visit. At the start, about 15% of the cases are a workaround. After 9-2013, this suddenly increases to 50%.

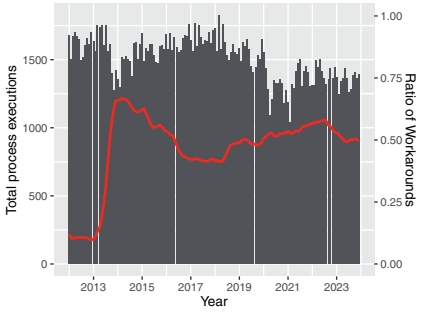


Figure 4.10: ER4 - Orders, Measurements, Triage, or Treatment before nurse visit. At the start, about 10% of the cases are workarounds. After 9-2013, this increases to 50%.

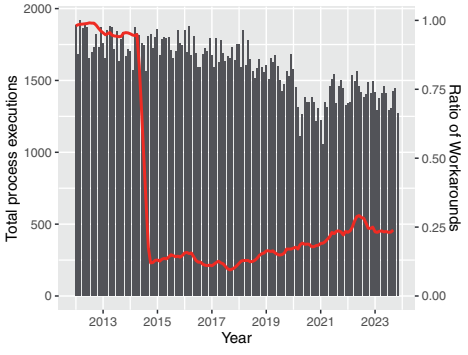


Figure 4.11: ER6 - Few pain scores before Discharge. At the start, the ratio of workarounds is stable at 100%. After 7-2014, this suddenly drops to 15% and gradually increases to 25%.

4.4.2 Incidents

As shown by the figures in the previous section, we identify four months that seem to strongly influence the use of workarounds: 9-2013, 7-2014, 4-2016, and 4-2017. As described before, we further investigated these months using historical documents: presentations, meeting minutes, and implementation reports. For each month, we find an incident that was likely the reason for the observed change, which we describe in more detail in the following.

September 2013 – Go-Live project "ER in order"

In the initial design of the HIS of the UMCU, most wards, including the ER, used the same structure to log all patient information. Due to the more dynamic work process in the ER, there was a misfit between the HIS and the work, especially in this ward. To improve the functioning of the ER, the specific ER part of the HIS was redesigned to better align with the work. This project is referred to as "ER in order" (in Dutch: "SEH op orde"). The "Go-Live" of the project was on September 24, 2013. While the project was initiated much earlier, it is not likely this would have influenced the daily processes: a large part of the work was behind the scenes where requirements were investigated and system updates were prepared. Instructions to users started a month before the Go-Live with key users and a week before with end users.

At the Go-Live of this project, we see a change in use at workaround ER2 (Figure 4.9), ER4 (Figure 4.10), and ER5 (Figure 4.8). The first two, related to the registration of physician and nurse visits before other events, show a large *sudden* increase in the ratio of workarounds to normative that is *stable* afterward. The last workaround, concerned with the timely logging of triage, shows a *temporary* increase until early 2015 in an otherwise *gradually* increasing ratio. Interestingly, the remaining ER processes do not show any significant change despite this project addressing most processes at the ER.

July 2014 – HIS update: ER medication and measurements tables

After the initial redesign of the ER system, multiple tweaks, fixes, and updates to the system were implemented. Such system updates do not only occur after a large-scale update but are common practice in such a large hospital. However, one large-scale update occurred after the previously discussed "ER in Order" project. This was logged as "Addition HIS medication in combination with measurements on ER". The new connection to the medication and measurement tables was implemented on July 8, 2014. Usually, updates note "Small details", "Adjustments", or "Review" as the primary changes in the script but this specific update was larger in scope. It connected two HIS tables to the general ER system in a new way, which also allowed enforcing new requirements related to the tables for specific activities. The effect of this update on the process can only be seen in workaround ER6 (Figure 4.11). This workaround is *stable* at 100% before the incident, during the incident we notice a *sudden* decrease in the occurrence of the workaround to 15%, after which the workaround increases *gradually* to 25%.

The indicator of this workaround is always the same: A patient is discharged with fewer than 2 pain scores being logged over their visit. However, we can clearly see a difference in how people work. In the initial stable state, if pain scores were deemed not relevant for a patient, they could be discharged without issues. Since the group of patients that visit the ER does not change much over time in this regard, the initial stability makes sense. After the HIS update, anyone who attempts to discharge a patient without there being at least two pain scores present receives an error message stating that these pain scores should still be measured, all but forcing people to enter additional pain scores before discharge. While it is still possible to perform this workaround in the new state of the system, it now requires

very specific knowledge on how to circumvent the error message. As this is not known to everyone, most people are forced to adhere to the guidelines, even if pain scores are irrelevant to the treatment of a specific patient.

April 2016 – Go-Live CS-EZIS.Net to HiX

At the start of the current dataset, the UMCU used the CS-EZIS.Net HIS throughout the hospital. The previous incidents describe updates to the existing system but in the long run, larger updates are required for the continuous effectiveness of any large system. The biggest change occurred in April 2016 when CS-EZIS.Net was replaced with HiX 6.0. While this new HIS was also developed by ChipSoft, it was a large update that had to be introduced to most wards of the hospital at the same time. Surprisingly, we find that most processes were overall unaffected. However, April 2016 coincides with a small increase in workarounds of type ER1 (Figure 4.7) in an otherwise *stable* ratio. While this increase was *temporary*, workers logged more events after discharge until more than a year later in the second half of 2017.

If we examine the workaround further and see what is logged after discharge, we do find a large difference. Two distinct main events are regularly logged after discharge and make up most of the events after discharge: A 'measurement' and a 'seen by physician' registration, which may be considered two distinct workarounds under one common indicator. The late 'measurement' registration is a common occurrence in many wards in the hospital. For example, we also investigated this workaround at the CCU with indicator CCU1 (Figure 4.3). When this workaround occurs, measurements are temporarily stored somewhere else, e.g., on paper, and logged at a more convenient or efficient time. This type of workaround generally occurs when nurses are pressed for time. Since this workaround affects patient data, the hospital is concerned about this, especially as the temporary logging elsewhere may not adhere to all privacy standards. The late 'seen by physician' registration can only occur in the ER as that is the only ward where the physician visits are explicitly logged. If they do not use the HIS themselves, physicians usually do not feel the need to log in to the system just to place a checkmark while a nurse is performing all other HIS-related tasks. While accurate information about when physicians see patients may be useful for analyses of ER processes, from a treatment perspective there is no incentive for the physician to log this check-in time.

If we look further into the distribution of the distinct workarounds in Figure 4.12, initially, only measurements were logged after discharge. This gradually moves towards the physician visit being logged after discharge. At the time of the incident, in April 2016, there was a large jump in the ratio, when the late physician check replaced the late measurements. So, instead of an overall stable process, we now note two variants of the process. One workaround is largely prevented with the system update, while another becomes much more common. In this case, the prevented workaround (i.e., a measurement being logged after discharge) is generally considered more problematic, so this tradeoff is most likely positive for the hospital.

The limited effect of the full system update can be explained by the type of system the UMCU opted for. There are two main versions of HiX: custom-built and standard content. The UMCU used the custom-built version. While this more expensive version required much more effort, all requirements could be fit to their current processes. Since they were already familiar with the general setup of the same vendor through their previous HIS, the effect of such an update proved to be limited. In line with this observation, we find no significant changes in our studied workarounds during the update to HiX 6.1 on March 16, 2018. The next large update would be in March 2024, when the UMCU updated to HiX 6.3. With this update, they also shifted from the custom-built to the less development-intensive standard content version. Since the current system cannot be fully customized to the processes, but processes should fit the system, the IT personnel expects more issues. Since the update is less than a month old at the time of writing, we do not have any data to investigate this expectation.

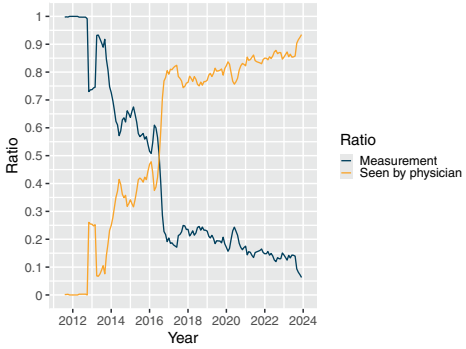


Figure 4.12: The ratio of the events executed after discharge. (ER1)

March 2017 – Introduction of new X-ray system

With workaround ER7 (Figure 4.6) we note a major change that does not coincide at all with any of the other workarounds: In March 2017, the number of times an X-ray is ordered jumps from around 100 executions to around 400 times per month, which then increases steadily to more than 900 times in 2020. If we specifically look at the ratio of workarounds, i.e., the cases in which an X-ray was logged right before discharge, we note that this initially occurs 10% of the time, which continuously grows to 30% in February 2017. One month later, when more X-rays are being logged in general, the ratio jumps to 40% but stays stable despite the continuous growth of the number of X-rays. While there were no exceptional incidents at the ER at that time, the X-ray system itself was replaced in March 2017. This makes it likely that the change in process in the ER was influenced by a change in the Radiology department.

4.5 Discussion

We have investigated ten different types of workarounds in two wards of the UMCU using all data available on these in the HIS. While our study does include interviews and unstructured historical data, the core of our approach is the quantitative analysis of process data. This approach allows us to study processes over an exceptionally long period, which would not be possible with qualitative methods such as interviews and observations. Where short-term studies have provided many insights into managing workarounds, they can only be used effectively to react to workarounds and improve processes, the additional long-term insights can be used to proactively address workarounds over multiple years. In this section, we discuss how workarounds evolve and how this corresponds with and diverges from existing literature. We also examine the limitations of our study.

4.5.1 Temporality of Workarounds

Stability versus change

We identified four temporal workaround categories: stable, sudden, temporary, and gradual workarounds. These findings partly correspond with earlier findings in the literature. Most studies into workarounds assume stability when examining them in detail (Ejnefjäll et al., 2023). While we do find that some workarounds are stable, other workarounds show a more complex, dynamic evolution. Research into the emergence of workarounds generally relates IS

updates to sudden changes (Ejnefjäll et al., 2023). Our identification of the incidents regarding the sudden changes in workarounds ER2, ER4, and ER6 are also IS-related. While the HIS update with ER6 is fully IS-related, the 'ER in order' project we found through ER2 and ER4 is broader than the IS in use as it also incorporates changes on a process design level. However, not all changes facilitated by IS updates are permanent. Workarounds ER1 and ER5 show that increases in workaround usage may only be temporary. This follows the observation that workarounds may disappear if the underlying obstacle that required the use of the workaround is solved (Malaurent and Karanasios, 2020). Although gradual workarounds are a part of theories surrounding the temporality of workarounds through unofficial modification of procedures (Alter, 2014) or diffusion through sharing (Safadi and Faraj, 2010), these theories state that after this relatively short period of a few weeks, workarounds can be officially adopted. Instead, we find that workarounds that change gradually (ER5, ER6, and ER7) may do so continuously for years.

On top of the new manifestations of workarounds we identified, another new insight pertains to the relations between these categories. In this study, we find that incidents may not only lead to sudden changes, but they may completely change the evolution of the workaround. For example, workaround ER6 (Figure 4.11) is stable for a few years, before the incident leads to a strong sudden decrease in workarounds, and it grows gradually afterward.

In Figure 4.13, we visualize the evolution of the workarounds that show any change. We see that most workarounds first have a relatively strong effect that is either temporary or permanent. After a few months, they then settle into either stability or gradual change, which can last for years. Workaround ER7 does not follow this and directly settles without an initial change. While we have no data supporting it, this seems to imply that a direct gradual change without a stronger initial change may also be possible. Half of these workarounds (ER1, ER5, and ER7) settle into a long-term category different from their initial state. This shows that while incidents such as HIS updates may have strong initial effects, the lasting long-term effects should not be ignored when studying how processes change.

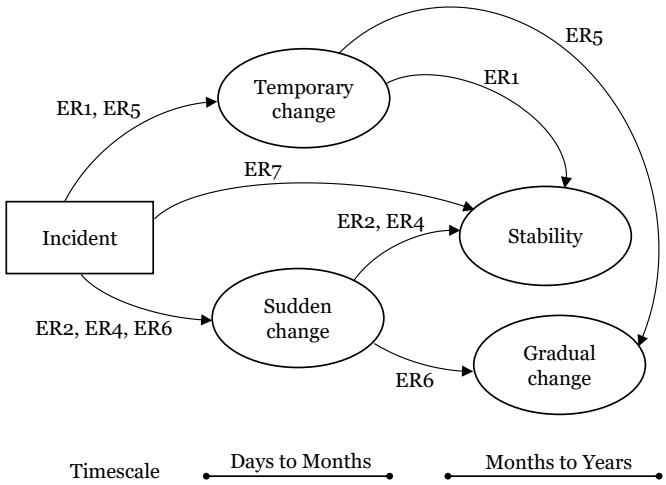


Figure 4.13: The change in the evolution category of workarounds.

Long-term Change

While details differ slightly, current theories concerning the temporality of workarounds start with the initial workaround as a temporary improvisation, which is then adopted in

the process by more people in a period of days to weeks, and finally accepted as formal improvements through planned changes (Safadi and Faraj, 2010; Alter, 2014; Weinzierl et al., 2022). These models take an organizational point of view: either a workaround exists, or it does not, regardless of whether a single worker uses the workaround or if most of them do. They also state a clear start and end point of a workaround's life cycle. It starts with an initial learning phase, usually initiated through an incident, and ends with the system being changed based on the workaround. In this organizational view, multiple workers are considered to change as a unit, following a dialectic view in terms of Van de Ven and Poole (1995).

Between the emergence and the system change, we find two options in existing theories: Either workarounds are adopted by groups (Alter, 2014; Weinzierl et al., 2022) or individuals share a workaround they know with peers (Safadi and Faraj, 2010). However, neither view fits with our data. If workarounds were consciously adopted by groups conform Alter (2014) and Weinzierl et al. (2022), we would expect large jumps in the ratio of workarounds to normative processes every time it is shared with a new group of colleagues. While ER7 shows some jumps during the growth phase, it overlaps with a period where the general process was not applied often, making the jaggedness in the line more likely related to this limited data availability. As the alternative adoption option, any individual with the workaround knowledge could share it conform the diffusion phase of Safadi and Faraj (2010). In that case, we would expect an exponential growth in the ratio, as a workaround could be shared more quickly every time a new actor uses it. This also does not fit as ER5 and ER6 show consistent linear growth, implying that the learning of the workaround is initiated through workers running into the obstacle before choosing the workaround. Both options follow a dialectic view: An obstacle to the normal procedure starts the development of an alternative variant, which is then adopted by multiple entities. While this view makes sense from an organizational point of view, the continuous growth we observe shows that the workaround is adopted more according to an individualistic, teleological model change (Van de Ven and Poole, 1995); Workers experience an obstacle individually and then adopt workarounds they may learn through colleagues.

4.5.2 Limitations

In this section, we reflect on the limitations of our study and how they may have influenced results. We first discuss the potential bias due to our case study with a single organization. Then we investigate the fit of the general indicators to specific workarounds. Finally, we explain why we did not identify any start and end points of the workarounds.

All workarounds we investigate originate from the same institution, which leads to a potential bias in our results. To mitigate this issue, we used an embedded case study with ten workaround types. All these types were discovered in previous work with a data-driven method where people, and potentially subjective bias, only were involved after the discovery. In addition, the workarounds originate from two very different wards. This makes it likely our results can be extended to other hospitals, although workarounds will likely evolve differently in smaller healthcare institutions or other domains.

If we view a workaround as a specific version of how a process can be executed, our expert-defined indicators may cover more than one version of non-prescribed behavior. As such, the general overview of the process does not paint the full picture of how processes change. We see this with indicator ER1. While the general overview indicates that the process is stable overall with a single temporary increase, if we examine it in more detail, we see that it consists of two distinct workarounds. We can place similar questions around indicators ER2 (Orders before Physician Visit), where the type of order may indicate alternative processes. The expert did include a general exception around this rule capturing emergency-related,

normative variants. Any further investigation into this would require specialized domain knowledge, which would be a clear direction for future work. Note that we are more confident in the remaining indicators. While ER3 and ER7 also concern the sequential order of events like ER1, they are very specific events. ER4, ER5, and CCU3 do compare a specific event to a group of general event types, but in these cases the workaround is clearly placed on the specific event, using the events as a general indication. Finally, ER6, CCU1, and CCU2 describe more specific workarounds. In all cases, the motivation of a specific workaround instance may differ, but the execution would be similar enough to consider them the same workaround.

According to current theories concerning the temporality of workarounds, a workaround has a clear starting point, after which its application grows over time before it may be adopted into the organization (Safadi and Faraj, 2010; Alter, 2014). While we find various incidents that increase the application of workarounds, we never find a very clear starting point. Despite our scope of 13 years, all workarounds have been present in some form since the start of the current information system. In addition, we also see no workarounds that were adopted as standard practice. This makes sense since we selected them during interviews with experts where we specifically asked for workarounds. Any official process that used to be a workaround before, would now be considered normative, so it would not come up during the interview. Future work could investigate this further by either specifically asking for adopted workarounds or by examining the current state of workarounds discovered in older studies. As an alternative route, we could shift the focus from workarounds within the full organization to workarounds performed by individual actors. This would allow us to determine when an individual first uses the workaround and examine the start and end in more detail.

4.6 Conclusion

In this study, we have examined how ten different types of workarounds in one large hospital evolved over 13 years using HIS data. Although most research into workarounds assumes they do not change much over time, we find that for most workarounds the occurrence of an incident drastically changes their application. Furthermore, even after the initial change, the use of a workaround may continue to grow for years instead of stabilizing. In addition, our data does not fit with current theories regarding how workarounds get shared in an organization. Instead of groups adopting a workaround or users sharing them with others, we find that the workarounds are adopted on a more individual level.

These new theoretical insights have direct implications for practice. First, workarounds need to be monitored and addressed continuously. Even if the frequency of workaround use is acceptable, this may change over time with or without incidents. Second, if workers decide to start using a workaround individually, prevention should also be targeted there. While informing about the dangers of a workaround may help, the individual need may be more important.

4.6.1 Future work

Repeating this study with different workarounds, preferably in a non-healthcare domain, should allow new insights. While the current study can contribute to existing theories into workarounds, taking a broader view would likely fill in even more gaps. For example, we do not know why some workarounds remain stable for a long time without any apparent control, while others continue to be used more and more.

A more detailed analysis of how workarounds evolve close to the incidents can further explain the initial changes in a process. While we can find specific incidents with a somewhat broad view of one month, this cannot further explain what is happening on a day-to-day basis.

In this work we split the evolution of workarounds into four categories and identify moments of interest based on a visual inspection, matching our observation to the dominant ideal types. Although this approach makes sense as a first step for investigating how workarounds evolve over long periods, the resulting categorization is somewhat subjective, especially since most workarounds show multiple categories in practice. In the future, this can be improved using more objective methods such as changepoint detection (Killick et al., 2012) for a more objective categorization.

Instead of our current focus on the ratio of workarounds to normative behavior, we can also switch our lens to include actors or even actor types. Instead of finding that workarounds spread according to certain patterns, this could further detail who instigates workarounds, potentially allowing insights into how people learn workarounds.

Chapter

5 Resource-Oriented Workaround Analysis: A Case Study

Abstract

Workarounds can show how workers solve local issues in designed processes and may therefore be used for general process improvements. Recent advances into workaround mining have shown data-driven approaches can both help discover workarounds and analyze them over long periods. However, earlier studies on workarounds have neglected the resource perspective. In this work, we use information about resource roles to find additional perspectives on workarounds. With a detailed analysis of a single workaround during a case study at an Emergency Room, we find clear differences in how various resource roles adopt it. We interview domain experts to further understand the motivations behind the use of the workaround. Following organizational theory, we find that the hierarchical power of their role is important in how resources approach workarounds, resulting in strongly different adoption speeds. We propose future work to enhance the research technique and suggest directions that may lead to new insights about workarounds.

5.1 Introduction

Workarounds are creative ways to handle unforeseen obstacles during work (Alter, 2014; Van der Waal et al., 2024c). Although workers typically have good intentions when using workarounds (Van der Waal et al., 2024b), there may be unforeseen negative consequences at a later time (Bartelheimer et al., 2025). As such, the management of workarounds is important for organizations. In a previous study, we investigated how workarounds change over a period of years (Van der Waal et al., 2024b). By measuring how often workarounds occurred compared to the normative process, we discovered that workarounds, and the underlying process, could change drastically after organizational and system changes. However, even when there are no discernible changes, workers may increase their use of the workaround, potentially changing an incidental workaround into a common part of the process.

Originally published as:
van der Waal, W. and Reijers, H. A. (2025). Resource-Oriented Workaround Analysis – A Case Study. In *Business Process Management Workshops*, pages 262–273

Our previous study also revealed that analyzing only the control-flow of the process was not always sufficient to understand the evolution of a workaround. For example, a workaround where activities were performed after discharge looked stable, with only a temporary increase in how often it was used after a Hospital Information System (HIS) update. However, when zooming in on the change, we found that initially nurses were involved but especially secretaries applied the workaround after.

In the field of Business Process Management (BPM), workarounds are commonly studied from a data perspective (Van der Waal et al., 2024c; Weinzierl et al., 2022). Where BPM is able to use historical data to study long term changes, there is little work studying the effect of resources, even though this perspective is considered important (Martin et al., 2021). On the other hand, with their high granularity through detailed interviews and observations, routine dynamics often studies differences between individuals (Nardelli, 2017; Tuominen et al., 2020) but struggles to study long term change. In this work, we aim to bridge the gap and investigate how workarounds change over time, considering resource roles.

In a case study at an Emergency Room (ER) we analyze one workaround in detail. By adding information about resource roles to an existing data analysis technique, we investigate changes in workarounds. Then, we conduct interviews to investigate the motivations and considerations for the workaround. By focusing on the resource perspective of workarounds during this case study, we gain a better understanding of how we can find differences between resource roles and how these influence their way of working. Practically, such differences between these roles can help tailor the management of workarounds better to roles.

In Section 5.2, we start with an overview of the current state of the art for the adoption of workarounds and how information about resource roles is currently used in BPM and routine dynamics. We then explain our technique in Section 5.3 and show the results of our case study in Section 5.4. We discuss our findings and limitations in Section 5.5 before concluding our study in Section 5.6.

5.2 Related Work

We start this section with an overview of how workarounds get adopted. As we are inspired by both BPM and routine dynamics, we will discuss how resource roles are used in these fields outside of workarounds. We will primarily use BPM terminology in this work. That is, we use *resource* to refer to a human resource or worker. However, to describe their designed job function, we use *role*, largely following one definition according to Organizational Role Theory (Biddle, 1986).

5.2.1 Workarounds

There has been limited research into the adoption of workarounds. In general, workarounds seem to originate as temporary improvisations, adopted by various resources, and, once one has grown enough, it is formally accepted or rejected (Alter, 2014; Safadi and Faraj, 2010; Weinzierl et al., 2022). There are various ideas about how they spread over time. One theory suggests that sharing is initiated by anyone who knows of the workaround (Safadi and Faraj, 2010), while another states that workarounds are shared through teams (Alter, 2014; Weinzierl et al., 2022). In both cases, once new people learn about the workaround, they will start using it. However, recent work has shown that sharing may not be the leading cause of the spread of workarounds; instead the use of a workaround may only grow if more workers feel the need to work around obstacles (Van der Waal et al., 2024b).

5.2.2 The resource perspective in BPM

If we take a broader look at the field of BPM, we note multiple calls for resource-oriented research. It is generally accepted that information about resources is relevant for BPM (Vom Brocke et al., 2016). To further understand process dynamics, their behavior is even considered crucial (Grisold et al., 2024). Further highlighting both the potential value and the limited research into resource-oriented research, a large Delphi study (Martin et al., 2021) notes "Analyzing business processes from the resource perspective" as an opportunity rated as extremely relevant by many international experts.

Two sub-fields of BPM take the resource perspective into account by design: Human Resource Mining and Resource Allocation. The former aims to leverage event logs to identify aspects of individual resources to suggest how teams of resources may work together better (Martin and Beerepoot, 2024). One of the more straightforward resource aspects in this field is their role within the organization (Cabanillas et al., 2020). Also, in the Resource Allocation field, the formal role is an important consideration (Bellaaj Elloumi et al., 2018; Pufahl et al., 2025).

While many conformance checking models include control-flow and data perspectives, the resource perspective is often suggested as future work (Alizadeh et al., 2018; Borrego and Barba, 2014; Dunzer et al., 2019). When this perspective is used, the role is often only used to approximate permissions (Mannhardt et al., 2016). For example, physicians and nurses may record patient tests, while secretaries are generally not allowed to do so. Instead of direct conformance, another angle is to investigate differences between resource roles. One such approach found that different roles can have vastly different ways to execute the same process, based on experience and training differences between them (De la Fuente et al., 2019).

5.2.3 The resource perspective in routine dynamics

Where BPM starts from the managerial side of the process, routine dynamics tends to approach the same topic from individuals. In this way, tensions between individuals are observed to change the underlying process (Nardelli, 2017; Tuominen et al., 2020). Further information, such as role and power dynamics about individuals in specific contexts can further help with understanding processes (Tuominen et al., 2020). Variances in resource roles often result in differences within the same process. On a process design level, this can be seen as organizations tend to only formally alter processes once resources with high power express that they consider it needs to change (Nigam et al., 2016).

However, even on a lower level these differences are clear. Given the same process, resource aspects such as gender and role strongly affect how and if workers vocalize experienced mistreatment through a designed "remedial voice mechanism" (Harlos, 2010). Besides the higher comfort expressing their frustrations, workers with more power also tend to be more open to altering their process executions creatively without fear of rejection by other, more powerful, colleagues (Danner-Schröder, 2020). Similarly, the notion of role-routine ecologies describes that in complex scenarios, similar routines often emerge within resources with the same role (Kremser and Blagoev, 2021).

To summarize, we find a broadly supported interest in the resource roles in both fields. Where BPM finds differences between roles, routine dynamics focuses on the similarities within them. In our work we will continue this focus to find how various resource roles behave when performing the same process.

5.3 Approach

To investigate how different resource roles use workarounds, we adopt a case study approach. Case studies are particularly suitable for how and why questions about complex, real-world phenomena (Yin, 2013). In our study, we aim to understand both the variation in workaround behavior across roles and the underlying reasons for such differences. This case study consists of two parts: a quantitative analysis of event data and a qualitative analysis of semi-structured interviews with domain experts. Our goal is to identify and understand differences in how resource roles engage with workarounds.

5.3.1 Context

The case study was conducted at the Emergency Room (ER) of the University Medical Center Utrecht (UMCU). This academic teaching hospital employs 12.000 people and provides care to over 220.000 patients annually. The ER consists of 19 treatment rooms and treats around 18.000 patients per year. The hospital uses HiX[1] as its Hospital Information System (HIS). This study was approved by the Medical Research Ethics Committee (MREC) NedMec (research protocol number 22/1055).

5.3.2 Data collection and preparation

For our data analysis, we use an event log enriched with resource role information. We classify each trace as either workaround or normative, based on prior, expert defined rules (Van der Waal et al., 2024b,c). To analyze the process, we view the process from three different resource-oriented angles to explore a single workaround.

Process Execution To determine how to measure the relevant resource for all traces, we find which event would be considered the workaround event. For example, if experts consider a late discharge of a patient a workaround, the discharge is labeled as the workaround event. We then generate an overview of all process executions, including both workarounds and normative traces, by calculating the proportion of traces in which each resource role performs the event. This enables us to find general changes in resource behavior.

Workaround Execution Next, we create a similar overview for traces marked as workarounds. This, again, allows us to find changes in resource behavior, but from a different perspective: Comparing this with the process execution plot reveals which resource roles approach workarounds differently. The plots also provide a complete overview of the roles relevant to investigate the workaround.

Workaround Ratio To sharpen our resource perspective, we calculate the ratio of workaround to normative process executions for each resource role. This allows us to determine precise changes in workaround behavior, extending the approach introduced in (Van der Waal et al., 2024b) by explicitly accounting for differences in resource roles.

[1] https://www.chipsoft.com/en

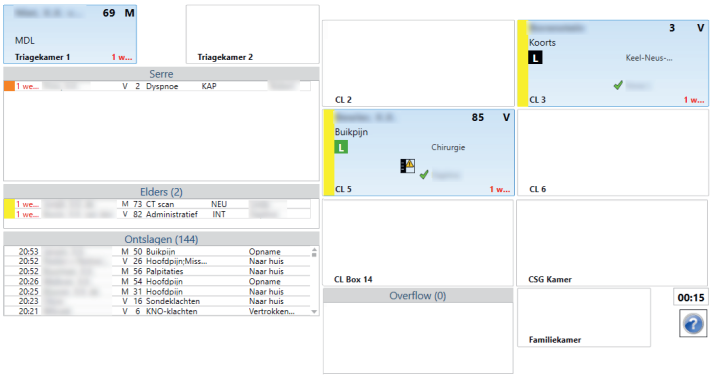


Figure 5.1: Screenshot of the process in HiX. The blue cards can be dragged to move patients to various rooms (e.g., "CL 2"), or discharge ("Ontslagen").

5.4 Case Study

5.4.1 Background

The workaround we investigate in this study concerns the registration of pain scores during visits to the ER. To ensure that treatment is effective, patients should be asked about their pain levels at least twice: once during triage at the start of the process and once again at discharge. This process is embedded in the HIS, which evaluates pain score registration before discharge.

This workaround was discovered in a previous study using the Semi-automated WORkaround Detection (SWORD) framework (Van der Waal et al., 2024c) and later further analyzed (Van der Waal et al., 2024b). In the present study we build on these findings by conducting a more detailed analysis of how the workaround changes over time, with a focus on the roles of the resources. We specifically selected this workaround because the HIS enforces the use of pain scores, which can only be circumvented manually. In other words, any deviation from the normative process must be a conscious choice, removing any accidental, non-workaround deviations from the data.

As illustrated in Figure 5.1, when discharging a patient, a user should drag a patient card from the room they are currently in to the Discharge field ("Ontslagen" in Dutch). This action starts the formal discharge procedure, which includes a check for pain score registration. If fewer than two pain scores have been registered during the ER visit, the system prompts to enter additional pain scores and prevents the actual discharge until this requirement is met. However, users can bypass this control by right-clicking the patient card to access a drop-down menu. From this, they can forcibly move the patient, circumventing any system checks. In doing so, they work around the pain score requirement.

5.4.2 Data Extraction

To follow patients, every ER visit is assigned a unique visit ID. We use this as a Case ID in our analysis. For each case, we also extract the pain score registrations and the discharge event. For the discharge event, we record the resource role of the employee that performed the discharge. We classify each trace as a workaround if fewer than two pain scores were registered. Otherwise, traces are labeled as normative.

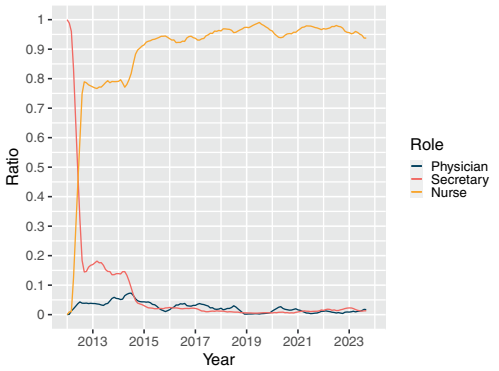


Figure 5.2: Process executions per resource role

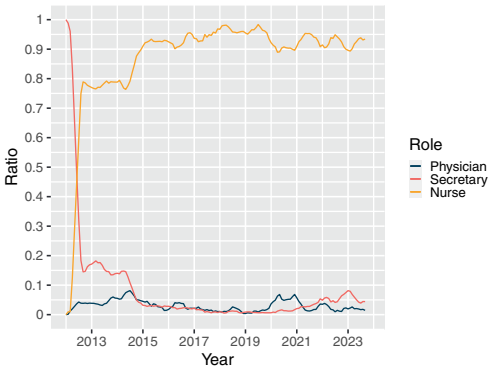


Figure 5.3: Workarounds per resource role

5.4.3 Data Results

Process Execution by Resource Role As shown in Figure 5.2, the discharge process is primarily executed by nurses. In 2013, approximately 80% of the patients are discharged by nurses, 15% by secretaries, and 5% by physicians. In July 2014, a sharp increase occurs: the proportion of discharges completed by nurses increases to over 95%. This moment of sudden change coincides with a previously discovered HIS update, which introduced a system constraint: users could only discharge patients if enough pain scores had been registered (Van der Waal et al., 2024b).

Workaround Execution by Resource Role Figure 5.3 shows the distribution of resources roles for workaround cases. Before the intervention, the role distribution ratios are similar to the overall process. However, notable differences emerge after the system update in July 2014. Then, nurses still perform most of the discharges, but their share of workarounds is usually lower than their share of the discharges. This difference is especially notable after 2020, where first physicians and later secretaries perform the workaround in more than 6% of the cases, despite neither of them performing more than 3% of the overall cases. This divergence indicates that secretaries and physicians have a higher tendency to use this workaround than nurses after the HIS update.

Workaround Ratio by Resource Role To gain a deeper understanding of role-specific behavior, we zoom in further on the resources by investigating the ratio of workarounds per role in Figure 5.4. Before the intervention, this ratio is close to 100% across all roles, as pain scores were rarely logged prior to being enforced by the system. Directly after the intervention in 2014, the workaround ratio sharply decreases to 15% and gradually lowers further to 10% in 2017 across all roles. We then see strong differences between the roles in how they adopt the workaround:

- Nurses only show gradual adoption of the workaround over five years up to 25%. This averages to 3%pt per year.
- Physicians start using the workaround commonly relatively early in 2018. Then they apply the workaround up to 75% of the time after four years. This averages to 16%pt per year.

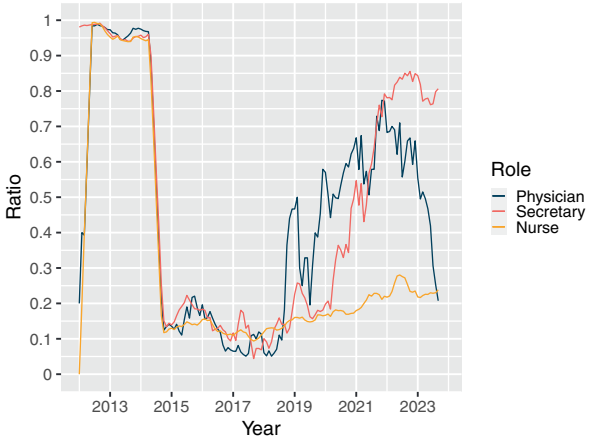


Figure 5.4: The ratio of workarounds to normative process executions for each role

- Secretaries do not strongly adopt the workaround until 2020, but they adopt it much faster than physicians and nurses. They ratio grows from 20% to 85% in two years. This averages to 32%pt per year.

Note that the workaround ratios of the physicians and secretaries fluctuate more sharply than those of the nurses. This can be explained as there are months where the former discharge few patients, resulting in ratios that change stronger. Despite this variability, the general changes in workaround ratio are clear.

5.4.4 Interviews

To understand the motivations behind the workaround behavior observed in the data, we interviewed two domain experts: one ER physician who has worked for 12 years at the UMCU and one nurse who has worked at the ER for 20 years, the latter also serving as the product owner of the HIS for the last six years. Both experts confirmed that they have performed the workaround themselves.

Discharge responsibilities and role dynamics The interviewees confirmed the general pattern shown in Figure 5.2: Nurses are typically responsible for discharging a patient. While both nurses and physicians have permission to perform all tasks related to the discharge, such as registering pain scores with patients, physicians often perform more specialized care. This makes nurses the more logical choice for these more general tasks that require a good overview of individual patients. As such, if a physician handles the discharge, the case is already exceptional.

While secretaries do not have permission to register measurements such as pain scores, they can discharge patients, typically in edge cases. For example, secretaries may discharge patients who leave the hospital against medical advice or they handle patients who were not properly discharged by anyone else. In this sense, secretaries often function as a "back-stop" and discharge remaining patients to make sure the log of patients at the ER is up to date.

Recognition of workarounds as alternative practices Both experts clearly distinguish between the normative process, where patients are discharged with pain scores during triage

and discharge, and the workaround, where patients are manually discharged with fewer than two pain scores. Besides these options, both experts noted that especially nurses use other strategies to satisfy the HIS requirements. Most notably, they may register placeholder pain scores, such as "n/a" (not available), when the patient is no longer present. While this option is designed for patients who are present but not capable of reporting pain, it is also used as a workaround. Alternatively, the nurse mentioned that some colleagues consistently register two pain scores during triage so they will not run into this issue at all later, already planning ahead and solving the issue beforehand.

Motivations for using workarounds Both interviewees report the same reasons for using any workaround: If the patient has left and it is found that there were not enough pain scores, no normative option (i.e., asking the patient) is available and people need to choose a workaround. This can happen regularly as the discharge procedure may get postponed while patients get sent home during a temporary peak in patients or if there is no computer present at the time.

Physicians often choose the manual discharge workaround as it is quick. The physician noted that they were explicitly instructed not to use this workaround, but they still did so regularly due to time pressure. In contrast, the nurse noted that most nurses are unaware of the manual option and typically resolve missing data through other means. The nurses who are aware of the workaround tend to use it if multiple forms are missing and manual discharge is more efficient than addressing each issue individually. For secretaries, the manual workaround is the only available option when discharging patients with missing pain scores, as they lack the permission to register additional measurements.

Perceived impact on patient care Neither expert could think of any direct negative impact on the quality of care resulting from this workaround. They are confident in the quality of care as patients are continuously monitored. They may not always log pain scores, but if patients state that their pain is manageable, nurses and physicians understand that no further treatment is necessary. In addition, patients are usually checked at least informally before they go home.

However, the physician did highlight a broader concern: while workarounds at discharge are typically harmless, similar behavior during intake could lead to more serious consequences. For example, skipping questions about antibiotic-resistant bacteria could pose significant risks if done improperly. This contrast underscores that not all workarounds carry the same level of risk, depending on the stage of the process and the nature of the information omitted.

5.5 Discussion

To interpret our findings, we further discuss our work first from a theoretical angle. After this, we consider the practical contributions and finish this section by discussing the limitations of our study.

Theoretical contributions We found differences in workaround adoption among the different resource roles. One explanation could be found in existing theory: Roles with higher power adjust their process with less fear of repercussions than those with low power (Dionysiou and Tsoukas, 2013). As such, secretaries would be hesitant to use the workaround, explaining their late adoption. However, as resources with less power are expected to align their work to those with more power (Danner-Schröder, 2020), it makes sense that secretaries use the workaround often, as they have no other options to discharge patients without sending work back to nurses or physicians.

The way nurses work strongly follows the temporal coordination theory in complex role-routine ecologies (Kremser and Blagoev, 2021). Following this theory, nurses could either try to follow the normative sequence or optimize timing. However, trying to follow both will lead to local conflicts, such as when a patient could leave without being formally discharged. In such cases, theory suggests that nurses would solve this through role-based prioritizing: They choose based on who benefits (Kremser and Blagoev, 2021). While sequence is important to management, timing is more important for the patient, who would go home earlier, and colleagues, who would have a less crowded ER.

Practical contributions Our theoretical findings have direct practical implications related to workaround management. The case study organization was already aware of this workaround, but our data analysis has shown that their attempts to minimize its usage have not worked over time. To manage workarounds more effectively it may be important to keep the resources roles in mind. Although physicians were explicitly told not to use it, they still do so as they feel comfortable deciding for themselves due to their high power. As such, threats of repercussions would have limited effect. Instead, discussing the effects on colleagues and the organization would likely have more effect (Nigam et al., 2016). Strict instruction could have an effect on low power roles like secretaries, although they would need alternative ways to align their process with their higher power colleagues.

In March 2024, the UMCU had a large HIS update when they moved from a Custom-Built system to the Standard-Content version of the same HIS. This new version standardizes many processes, including the intake and discharge of patients. Various questions, like pain scores, are not enforced anymore. For this workaround, the change is likely positive, given that registering pain scores is unlikely to negatively impact the process. However, during intake, questions about antibiotic-resistant bacteria may now also be skipped without a workaround, which may have potential negative effects. This emphasizes the need for ways to monitor workarounds, as their usage may change unpredictably.

Limitations The primary limitation of our work is its scope: We only inspected one workaround in a single organization and interviewed two domain experts. Investigating more workarounds and holding more interviews would broaden this study, making more generalizations possible.

In this study, we only used existing visual inspections to find differences in the workaround ratios for the various resource roles. In this case, the differences between roles are large enough to discover these without further statistical proof. However, measures such as cross-correlation (Yoo and Han, 2009) or change point detection (Aminikhanghahi and Cook, 2017), would provide a more rigorous way to determine such differences.

5.6 Conclusion

In this resource role-oriented study, we performed a case study at an ER. We looked into the discharge process of patients which should have at least two pain scores. We find that besides the designed differences in how often various roles discharge patients, physicians and secretaries adopt the workaround more quickly than nurses. The findings are in line with IS literature, stating that roles with more hierarchical power tend to alter their personal process quickly without fear of repercussions, while roles with low power may change their process to support those with higher power.

While the resource perspective has already led to novel insights, we can extend this work further in various ways: First, we could discover more about this specific workaround with a further study into the alternative pathways for this process. Batch detection (Van der Waal

et al., 2024a) could discover multiple pain scores registered during triage, while rule-based detection (Van der Waal et al., 2024c) could find "not available" pain scores. In addition, a new analysis with more recent data could show new insights into how the process changed, especially given that the system requirements that enforced the number of pain scores is not present anymore.

Second, while we focused on resource roles, similar techniques could be used to find differences between individual resources. This could provide new insights into how workarounds are shared between individuals and when they are used. Combining this with additional resource characteristics, such as quality or experience (Zhao et al., 2015), could help organizations manage workarounds further.

Finally, in this study, we use expert-defined rules to classify traces as workarounds or normative. While this approach manages to identify instances of a specific path, interviews showed that alternative workarounds can be missing. By repeating the cycle of data-analysis and interviews, we could get a more complete view of workaround behavior within a single process.

Chapter 6

Operating the SWORD: Finding Workarounds for Process Improvement

Abstract

Objective: Workarounds can reveal important opportunities for process improvement once they are identified. In this study, we aim to quantitatively detect workarounds that are relevant for improving healthcare processes.
Methods: We present the Semi-automated WORkaround Detection (SWORD) framework. Based on a set of 22 patterns, this framework uses existing Healthcare Information System (HIS) data to iteratively discover workarounds through four recurring steps: 1) data processing, 2) quantitative analysis using the SWORD patterns, 3) classification and evaluation of the analysis by domain experts, 4) data refinement using insights from the domain experts. We demonstrate this method in a case study conducted at the Gastroenterology ward of a large teaching hospital.
Results: In our case study, we discovered five workarounds: three in the first round, and two additional ones in the second. Despite previous qualitative process optimization projects, the process manager was unaware of two of the workarounds and considered them concerning and actionable.
Conclusion: The publicly available SWORD framework supports the identification of workarounds using routinely collected HIS data and can help healthcare practitioners uncover concrete opportunities for real-world process improvement.

6.1 Introduction

As organizations grow, they introduce formalized procedures to instruct employees on how to work. However, such procedures rarely cover all subtleties that are inherent in human work, such as patient admission protocols, medication administration guidelines, or documentation requirements in information systems. As a result, employees sometimes need to work around formal procedures to handle unforeseen circumstances, so-called workarounds. Workarounds have been extensively studied in healthcare, where they are often deployed to, for example, handle the high workload (Beerepoot et al., 2019b) or to solve the mismatch between the Healthcare Information System (HIS) and the actual process (Safadi and Faraj, 2010; Cresswell

Submitted manuscript

et al., 2016). As workarounds point to issues in the day-to-day work, they can be directly used to improve patient care (Van der Waal and Reijers, 2025; Beerepoot and Van de Weerd, 2018).

The insights that may be obtained from workarounds are twofold and depend on the consequences of the particular workaround. 1) With some workarounds, users may ignore or forget important details that are covered in the procedure. Such workarounds should be actively prevented, as they may, for example, induce risks to patients (Beerepoot and Van de Weerd, 2018). 2) Other workarounds may not have any clear drawbacks, but instead compensate for shortcomings in the prescribed procedure and enable employees to perform their work more effectively. Insights from these should be leveraged to improve the official procedure and improve the way other people work too (Cresswell et al., 2016; Beerepoot and Van de Weerd, 2018; Bartelheimer et al., 2023). Regardless, workarounds provide a unique perspective on difficulties, bottlenecks, or other issues in procedures that can be used to improve them.

Although workarounds have been known to be useful for decades (Safadi and Faraj, 2010; Petrides et al., 2004), the actual discovery of workarounds is still a challenge. While they may be discovered qualitatively through interviews and observations aimed at finding discrepancies between the procedure and the execution, such methods are limited as people may not always remember all the details from their work, they may not recognize that they use a workaround at all, or they may actively hide their way of working if they suspect people may disapprove (Beerepoot and Van de Weerd, 2018; Weinzierl et al., 2022).

As one solution to these issues, recent studies have focused on a more data-driven, quantitative approach using data from HISs as a more objective way to determine how people work. As such data never reflects the full process, the Semi-automated WORkaround Detection (SWORD) framework has explored the combination of quantitative data analysis with qualitative interviews to use objective data to lead in-depth interviews to allow completely new insights (Van der Waal et al., 2024a). The SWORD framework has previously been used to discover workarounds over data from full wards. While this led to new workarounds, this broad setup effectively discovered general workarounds in the general way of working, instead of workarounds as a solution for a specific problem. As such, they are difficult to use for any targeted process improvement. This is a major impediment to more general use. As such, our aim is to overcome this so that SWORD is an effective instrument for healthcare organizations to improve their processes.

In this study, we focus on process analysis with the aim of finding opportunities for *targeted* process improvement. We show how to discover workarounds iteratively in four recurring steps: 1) data processing to transform HIS data into an event log, 2) detailed, SWORD-based data analysis to generate fine-grained, process-specific insights, 3) short interviews with domain experts to classify candidates as true workarounds or flag non-workarounds; and 4) data refinement, using these insights to remove deviations unrelated to the target process and thereby improve event-log quality for the next iteration. While the first three steps are requirements of using the SWORD framework, the fourth step directly focuses the subsequent iteration towards improvement opportunities of the specific process.

We demonstrate the operationalized SWORD framework in a real-life case study targeted towards the discovery of actionable workarounds, which can be directly used to improve the underlying process. Specifically, we perform a case study at the Gastroenterology ward of the Jeroen Bosch Ziekenhuis (JBZ), a large Dutch teaching hospital, to find workarounds in their colonoscopy procedure. As this process is performed very often, it has been improved by the hospital recently with traditional methods. Even so, we discovered five workarounds with our data-driven approach, two of which were considered to require further action by the domain experts.

This demonstrates that with the SWORD framework, we can find workarounds that can be used to improve an established process even if it is 1) specific, 2) routinized, and 3) previously optimized through traditional (non-process science oriented) methods. The first two points

show that the SWORD framework not only applies to general, explorative studies but also in very focused contexts. The last one shows the added value of workaround mining to existing approaches.

In Section 6.2, we first give a brief overview of the current state of the art of workaround mining. Then, in Section 6.3, we describe the way the SWORD framework can be used for workaround discovery in a targeted process improvement initiative. We demonstrate how to use the framework in Section 6.4 with a real-life case study, the results of which we discuss in Section 6.5 before concluding in Section 6.6.

6.2 Related Work

In this chapter, we adopt the widely cited definition of workarounds of Alter (2014, p. 1044): "A workaround is a goal-driven adaptation, improvisation, or other change to one or more aspects of an existing work system to overcome, bypass, or minimize the impact of obstacles, exceptions, anomalies, mishaps, established practices, management expectations, or structural constraints that are perceived as preventing that work system or its participants from achieving a desired level of efficiency, effectiveness, or other organizational or personal goals."

Traditionally, workarounds have been detected through qualitative measures. Within the healthcare sector, especially in the Information Systems field, many have been discovered with interviews or observations (Alter, 2014; Ejnefjäll and Ågerfalk, 2019). Building on examples and insights from these discoveries, more quantitative approaches have been developed (Beerepoot et al., 2021). Initially, rule-based approaches have been used to quantify the occurrence of known workarounds (Outmazgin and Soffer, 2014). When workarounds are expressed in explicit rules, all instances of them may be found in a single event log, regardless of the duration of the measurements. However, as this design does not allow flexibility, only new instances of previously detected workarounds can be found.

To discover new workarounds, we find three quantitative approaches. First, specifically for time-related workarounds, deep learning may be used (Weinzierl et al., 2022). However, this has only been attempted in partially synthetic data. In addition, such deep learning requires a huge number of traces to be labeled by hand to train the model, which requires too much manual effort from experts to be usable in practice (Weinzierl et al., 2022; Van der Waal et al., 2024c).

Second, as a process mining approach, we can detect workarounds by comparing a representation of the designed process to data generated during the actual process to find discrepancies between the theoretical, prescribed process and the practically executed process (Wijnhoven et al., 2023). While this approach has been shown to work in practice, it always requires the user to first determine the prescribed process. As this requires specialized domain knowledge from the start, the scalability of this method is still limited.

Finally, the SWORD framework can be used to discover workarounds (Van der Waal et al., 2024a). This framework consists of 22 patterns that describe how workaround traces may differ from normative ones. By applying each pattern to an event log, the most strongly deviating traces are marked as workaround candidates. Like all previous approaches, these candidates still need to be evaluated by experts to determine if a trace is a workaround or a rare occurrence for any other reason. However, domain knowledge, and thus expensive expert involvement, is only necessary during short, focused interviews near the end of a project, improving scalability. While the SWORD framework has been shown to be useful for discovering workarounds in a complete hospital (Van der Waal et al., 2022), the emergency room (Van der Waal et al., 2024a,c,b), the cerebrovascular nursing ward (Van der Waal et al., 2024a), and the Cardiovascular Care Unit (Van der Waal et al., 2024b), all these studies analyzed the data of all events that occurred in the hospital or department without a focus on

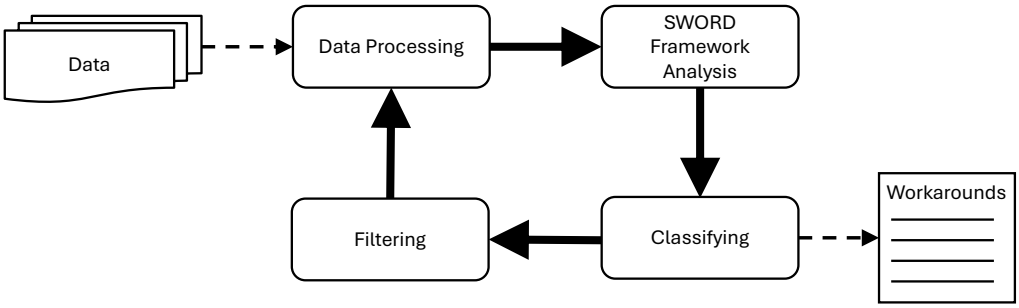


Figure 6.1: The SWORD framework cycle to extract workarounds from data.

a particular process. In this study, we explore this gap and investigate the usability of the SWORD framework for targeted process improvement.

6.3 SWORD Framework

In this section, we will describe how to use the SWORD framework to discover workarounds in a practical healthcare setting. After extracting data from the HIS, there are four steps, as outlined in Figure 6.1: processing the data into an event log, applying the SWORD framework to discover workaround candidates, classifying said candidates as workarounds or non-workarounds with semi-structured interviews, and finally creating new filters to improve data quality to repeat the cycle. In the remainder of this section, we will describe these four steps in more detail.

6.3.1 Data Processing

The data processing step transforms raw HIS data into an event log. As shown by the example in Table 6.1, an event log contains at least events with specific timestamps for each separate trace. For a more detailed view of each trace, the log can be enriched with additional attributes, such as timestamps for the ending or log time of each activity, the actors or resources involved, or more detailed information such as measurement values (Eck et al., 2015). The exact process of constructing an event log depends strongly on the underlying database structure, the extraction tools, and the ICT support available. As this is beyond the scope of this work, we refer to (Stein Dani et al., 2022) for a detailed overview of steps and considerations when creating event logs.

6.3.2 SWORD Framework Analysis

To analyze the event log, we apply the Semi-automated WORkaround Detection (SWORD) framework (Van der Waal et al., 2024a). The SWORD framework leverages a set of predefined patterns in event logs to identify traces with workaround-related deviations. Note that at this point, deviating traces cannot be considered workarounds without domain expert involvement. As such, any traces flagged by the SWORD framework should only be treated as workaround candidates in this step.

To quantify the deviations, most patterns are rated as z-scores: the number of standard deviations from the mean. With this, we create a distribution covering all traces for each

Table 6.1: Part of a trace from the event log.

CaseID	Activity	Sub-activity	Timestamp
PROCEDURE:1039373937	Admission	Colonoscopy	2021-xx-xx 06:56:00
PROCEDURE:1039373937	Observation	Resp freq.	2021-xx-xx 07:17:00
PROCEDURE:1039373937	Form	Checklist	2021-xx-xx 07:22:00
PROCEDURE:1039373937	Order	Transfer	2021-xx-xx 07:24:00
PROCEDURE:1039373937	Observation	VAS, Painscore	2021-xx-xx 07:24:00
PROCEDURE:1039373937	Procedure	Local anesthetics	2021-xx-xx 08:30:00
PROCEDURE:1039373937	Medication (Request)	FENTANYL	2021-xx-xx 08:43:00
PROCEDURE:1039373937	Medication (Administer)	FENTANYL	2021-xx-xx 08:43:00
PROCEDURE:1039373937	Medication (Administer)	MIDAZOLAM	2021-xx-xx 08:43:00
PROCEDURE:1039373937	Form	Transfer	2021-xx-xx 09:02:00
...	...	...	...
PROCEDURE:1039373937	Discharge	Home	2021-xx-xx 10:42:00

separate pattern. This allows us to compare how strongly a trace deviates from others on the various aspects of the patterns. For example, Table 6.2 shows that the ten traces that deviate most on "Time between activities out of bounds" do not show such strong deviations on the "Time period out of bounds" pattern. This indicates that patients who often have to wait are not necessarily treated at times out of the norm.

While most patterns are expressed as a single ranking, several patterns require additional context to be correctly interpreted. This is especially true for six out of the 22 patterns:

- Activity frequency out of bounds
- Occurrence of activity outside of time period
- Delay between start of trace and activity is out of bounds
- Time between activities out of bounds
- Duration of activity out of bounds
- Delay between event and logging is out of bounds

To add context to these time-oriented patterns, we determine the original ranking for each pattern per activity. In this manner, we not only check if any activity occurs, but also if each activity occurs a certain number of times.

As shown in Table 6.2, for the "Max time between activities" pattern, this allows us to distinguish delays after filling out a standardized Form or Order, which indicate a patient waiting for a long time, from delays following Medication Administration, which are often clinically acceptable.

6.3.3 Classifying

While the SWORD framework flags the strongest deviations in the event log, a strong deviation does not necessarily imply a workaround. Such deviations may also result from unintended mistakes, errors in the system, or complex patients that require rare types of care. While these deviations do indicate potential workarounds, we require specialized domain knowledge to

CaseID	Max time period	Max time between activities	Form	Order	Medication (Administer)	Medication (Request)	Report
1051674462	1.41	**29.08**	**29.08**	2.35	-	-	-0.17
1072625069	2.50	**24.48**	0.28	0.08	**24.48**	-0.37	-0.12
1041974319	0.21	**23.25**	**23.25**	-0.07	-	1.66	0.23
1055637025	2.10	**22.90**	0.62	**22.90**	-	-0.32	-0.22
1075144397	1.97	**22.55**	0.52	-0.32	**22.55**	0.08	0.03
1040318314	2.23	**22.06**	1.61	**22.06**	-	0.67	-
1040148402	1.79	**22.06**	0.08	**22.06**	-	-0.27	0.03
1043519877	1.30	**22.01**	0.18	0.67	-	**22.01**	-0.22
1039553342	2.33	**22.01**	0.18	**22.01**	0.18	-0.37	-0.07
1045258268	1.74	**21.96**	0.28	**21.96**	-	-	1.46

Table 6.2: The ten traces with the highest deviations on the "Time between activities" pattern. The first two columns show the original pattern. The remaining columns split the scores further per activity in the event log. As shown, even if the maximum deviations are of a similar level, the cause for the high pattern scores differ between "Form", "Order", "Medication (Administer)", and "Medication (Request)" events.

evaluate what is happening in a specific trace. We therefore review strongly deviating traces with domain experts to determine whether a trace is a workaround or anything else.

We do this by evaluating the traces that are indicated as the most likely workaround candidates by the SWORD framework. Per pattern, we select the strongest deviating traces and present them to domain experts together with the scores of the SWORD framework, which indicate why the trace is different from the norm. Experts can then classify the trace based on very specific data. For example, a high score on "Activity frequency out of bounds" can point to a trace where a patient is admitted twice to the hospital instead of once. Experts can explain if such a second admittance is a local solution to, for example, a patient registration problem or a normal but rare occurrence for other reasons.

6.3.4 Filtering

If a workaround candidate is not considered a workaround by an expert, it still provides information about the base process. A marked trace that is not a workaround may be rare because of two reasons. 1) Its occurrence may be rare, but part of the studied process, in which case it helps the researchers understand the edge cases of the process. While interesting, more useful is option 2): It is possible that the trace describes a rare type of patient that happens to follow the same process. For example, when scheduling MRI tests, most patients will be planned well in advance. However, some patients will require the MRI urgently, following an emergency procedure. These patients will naturally follow a different path which the SWORD framework will detect. This second type of non-workaround can be filtered out to improve the focus on the process to investigate. In this example, removing the emergency cases to better focus on the routine MRI tests.

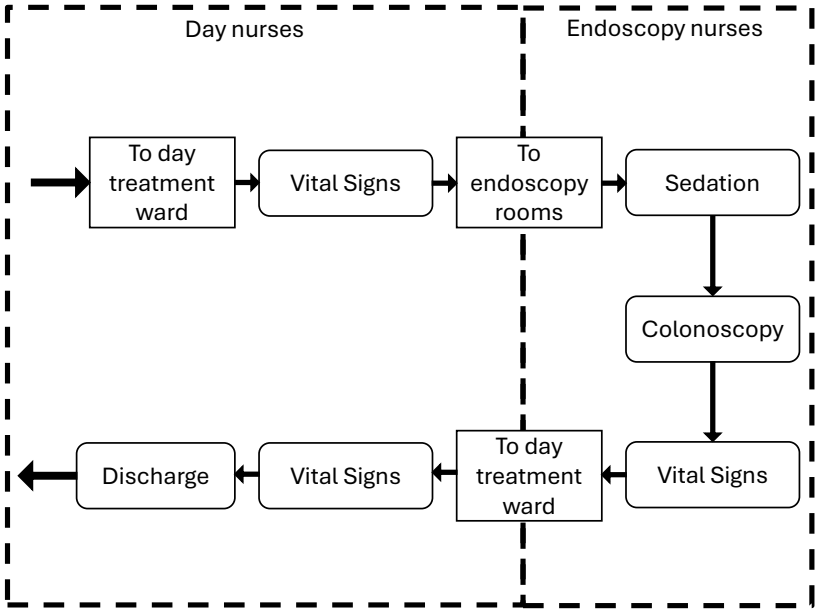


Figure 6.2: The simplified patient pathway for a typical colonoscopy.

6.4 Case Study

To demonstrate our method, we apply it to a real-life case study with the Jeroen Bosch Ziekenhuis (JBZ). The JBZ is a large teaching hospital in the Netherlands, offering both emergency and advanced care to all types of patients. Annually, the hospital admits about 55,000 patients and handles around 500,000 patient visits to outpatient departments. More specifically, our case study targets the colonoscopy process of their Gastroenterology ward. As it is performed around 5,000 times per year, even small improvements to this process add up quickly, making it an ideal process to improve for making a meaningful impact from a hospital perspective. In addition, this procedure requires demanding preparation and tends to be stressful for patients, so improvements are also important for the quality and comfort of care.

As shown in Figure 6.2, most patients who are scheduled for a colonoscopy need to check in with the day treatment ward at least one hour before the procedure. At this ward, day nurses verify whether the patient has completed all required preparations and measure their vital signs to establish a baseline. If everything is in order, nurses perform the final preparations, such as placing an IV cannula, and escort the patient to the endoscopy area at the planned time. Here, endoscopy nurses will take over and bring the patient to the assigned procedure room. Immediately before the procedure, the patient typically receives sedation (Midazolam and Fentanyl). During the colonoscopy, vital signs are measured continuously to check the condition of the patient. After the procedure, patients are transferred to the waiting room, and day nurses are called to bring them back to the day treatment ward. On returning, vital signs are measured again to check for discrepancies. To monitor side effects of sedation, patients have to stay at this ward for one hour after their sedation is initiated. Finally, nurses fill out a final discharge checklist and remove the IV needle before the patient can leave the hospital.

6.4.1 Data extraction

The JBZ primarily records patient data in HiX[1], their Healthcare Information System (HIS). As this system is primarily designed to support the day-to-day clinical work instead of research, the focus is on reporting the current patient status, rather than the process a patient followed over time. To obtain research-ready data, we use CTcue[2], a tool that supports the extraction of pseudonymized data from the backend databases of various HISs using more standardized queries. For the initial data extraction from the database, we select all patients with activity code 034686, which indicates a colonoscopy, from 1-1-2021 to 31-12-2024.

For all selected patients, we extract the other care events that occurred on the same day as their colonoscopy from all available tables: Admission, Allergy, Appointment, Care Plan, Consent, Diagnosis, Episode of Care, Form, Measurement, Medication Administration, Medication Request, Observation, Order, PDMS, Procedure, and Report. For each event, we register the procedure ID, the (pseudonymized) patient ID, the source table as the activity, the timestamp, the logging time, and the actor involved. For context, we also register extra information depending on the activity. For example, for Medication Administration, we record the medication and dosage, and for Observation, we save the type, such as pain score or heart rate. This resulting event log contains 780,455 events, covering 18,519 colonoscopy procedures across 16,260 patients. This event log serves as the input for the subsequent data processing and analysis steps described in this section. Figure 6.3 shows an overview of the data processing and analysis steps.

6.4.2 Round 1

Data Processing

To improve the data quality of our event log, we apply four filters: 1) We exclude individual events without a usable timestamp, 2) We exclude full traces without an admission event, 3) We exclude individual events with both a reported data and logging date other than the admission event, and 4) We exclude traces without a reported colonoscopy. This results in an event log with 644,465 events covering 17,172 colonoscopy procedures over 15,320 unique patients usable for further analysis.

SWORD Framework Analysis

Given the data available in the event log, we can apply eight patterns from the SWORD framework, listed in Table 6.3. Applying these results in a table containing z-scores describing the types of deviations for each trace, like Table 6.2. For a detailed overview, patterns in italics are adjusted to take all unique activities into account, allowing us to distinguish between, for example, short delays after admission and intended longer delays after medication administration conform to Section 6.3. This results in one column for each of the 15 activities. The remaining patterns all result in one column per pattern, for a total of 64 rankings.

Classifying

To classify if deviations indicated by the SWORD framework analysis relate to workarounds or not, we conducted two rounds of interviews for different perspectives on the process. Both rounds took place in the JBZ and were recorded. The first round was jointly with two nurses

[1] https://www.chipsoft.com/en/solutions/electronic-health-record-hix/
[2] https://www.iqvia.com/locations/emea/iqvia-connected-healthcare-platform/iqvia-patient-finder-solution

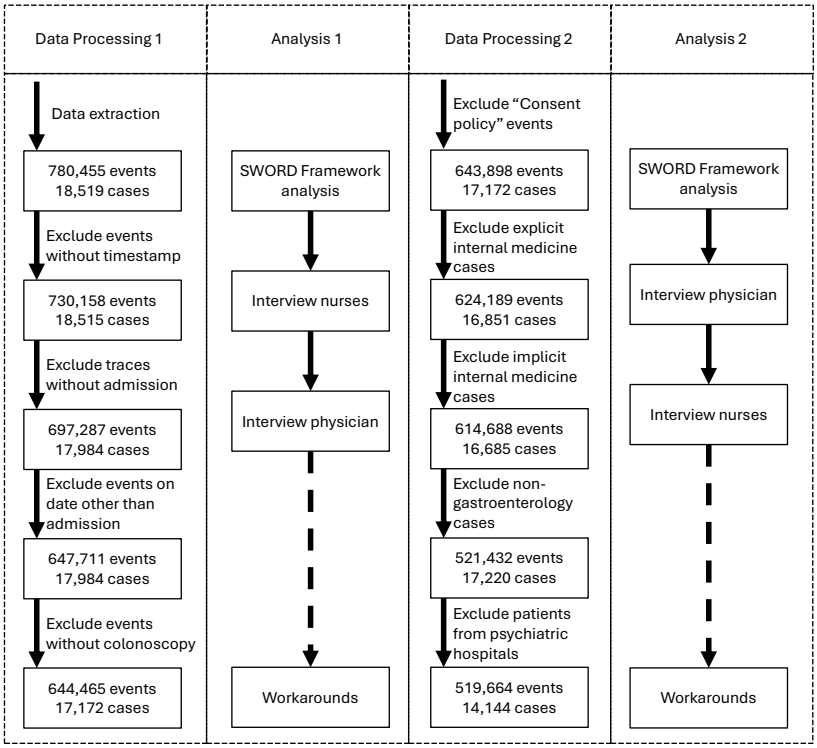


Figure 6.3: An overview of the two processing and analysis rounds in the case study.

and one coordinator and took 54 minutes. The second round with one physician took 59 minutes.

Both rounds are structured in the same way: We start with a short introduction of what we consider workarounds, then we briefly describe the data processing, and finish the introduction by showing how to read a trace from an event log. We then move on to discuss the notable traces in detail to determine if they consider a trace to be a workaround or any other type of deviation. For a balanced view, we discuss both the highest-scoring and, if patterns also return negative scores, the lowest-scoring trace for each pattern. For example, the "admission" event usually occurs once per trace. The "Activity frequency" pattern returns positive scores if a patient is admitted more than once, with higher scores for more admissions. Negative scores may occur, too, if the patient was never admitted according to the event log.

As discussing even only two traces for the 64 rankings would take too much time, we limit the number of traces in two ways: 1) If a trace scores high on multiple patterns at the same time, we discuss it only once but indicate the multiple concerns with the trace. For example, traces that have more activities score high on "Activity Frequency" and usually also on "Duration of trace". 2) For the patterns that are applied to the unique activities, we discuss the separate activities if they score high compared to the others. So, for the activity frequency, we discuss the diagnosis, admission, observation, and discharge as they have z-scores higher than 20 or lower than -10. While these boundaries are arbitrary, we selected them to keep the length of the interviews limited. In total, we discussed 19 traces in both interviews.

Table 6.3: The patterns from the SWORD framework applicable to this case study. Patterns in italics were adjusted to take all unique activities into account.

Type	Pattern
Control-flow	Occurrence of directly repeating activity
	Activity frequency out of bounds
Resource	Number of resources out of bounds
Time	*Occurrence of activity outside of time period*
	Delay between start of trace and activity is out of bounds
	Time between activities out of bounds
	Duration of trace out of bounds
	Delay between event and logging is out of bounds

Workarounds

After discussing all cases, we identified three workarounds, each discovered through time-based patterns:

WA1 **Backdating diagnoses to a default year.** In some cases, a diagnosis event is recorded in 1900. The physician explained that this is a common way to satisfy the system's requirement to enter a diagnosis date, even if the underlying condition was discovered much earlier but is not recorded in the system yet. Using 1900 signals that the diagnosis predates the current admission, and not because of recent tests.

WA2 **Closing orders from previous admissions.** In some cases, an order that started a few years ago is updated at the start of a hospital admission. Nurses sometimes need to do this for certain orders related to intravenous (IV) needles: when the type of IV requires multiple checks per day, the system automatically generates recurring orders until the IV is removed. To make sure the current process is not polluted with such automated checks, nurses close all open orders during admission.

WA3 **Keeping default administration times.** For medication logging, the system requires an administration time. In a few cases, the administration time was substantially later than the logged time. This means that, according to the system, nurses would, for example, register at 21h that they would administer medication at 22h instead of registering it during the event. Nurses explained that the system gives a few standard time options that can be manually adjusted. As this takes time, nurses often keep the default time for medication that is not time-critical, such as common painkillers.

Filtering

Besides discovering workarounds, the interviews also led to a better understanding of the rare events in the colonoscopy process. With this, we could create six filters to separate these workarounds candidates from the core process. The first filter describes an activity that turned out to be unrelated to the process; the remaining four all cover patients with a colonoscopy in a non-standard, but normative way. 1) The "Consent Policy" activity was performed by patients themselves in the online portal. While it was often logged on the same day as the colonoscopy, it holds no direct relation to it. As such, we removed all events with this activity. 2) Many traces covered patients who were hospitalized for a longer period and

also underwent a colonoscopy procedure at some point. As we are interested in patients who visit the hospital only for the colonoscopy, we filtered them out when another ward was explicitly stated as admission data. 3) Patients who were administered medication outside of working hours on the day ward were filtered out for the same reason. Although another ward was not stated explicitly, the medication times indicate that these patients were hospitalized. 4) Some patients undergo a colonoscopy through specializations other than Gastroenterology. These patients follow more complex pathways and are therefore filtered out. 5) A few patients are hospitalized at different care facilities, for example, for psychiatric care, but do get a colonoscopy at the JBZ. These patients get more guidance and therefore do not follow a typical path. As such, patients who are admitted from anywhere other than home are removed from the data set.

6.4.3 Round 2

Data Processing

As the analysis in Round 1 indicated that multiple workaround candidates should be ignored as they were either rare, acceptable, or unrelated to the core process, we can use the resulting filters to better focus our data analysis. With these, we end up with an event log with 519.664 events covering 14.144 colonoscopy procedures over 12.938 unique patients. Figure 6.3 shows a detailed overview of all the impacts of the separate filters.

SWORD Framework Analysis

In addition to applying the six filters to refine the event log, we also discovered that the definitions of two patterns needed to be further specified to fit this case. For the "Duration of trace out of bounds" pattern, we ran into a common issue: Often, there is one activity related to a hospitalization executed either very early or very late (Van der Waal et al., 2024a,c). For example, the registration of a physician meeting a patient may be logged multiple days after the patient has visited the emergency room. This is a common workaround to solve registration problems, which is strongly related to workarounds WA1 and WA2. These all result in this pattern not considering the true duration of the hospitalization. To address this, we apply a correction: rather than determining duration from the first and last recorded events, we compute it using the "Admission" and "Discharge" events, which experts indicated are highly reliable. That is, these events are consistently logged at the time they occur.

In addition, the "Delay between start of trace and activity is out of bounds" pattern measures the delay based on the first activity. Similarly to the previous point, the time of this first activity is not a reasonable point to align traces. As alternatives, we use the more reliable "Admission" event instead.

Classifying

The interview setup was the same as in the first round. After a brief introduction, we reviewed the traces with the strongest deviations per pattern. To avoid redundancy, we excluded overlapping traces that had already been evaluated in Round 1. In total, we discussed seven new traces. As before, we conducted and recorded two interview sessions in the JBZ: the first with the same physician as in the first round, which took 37 minutes, and the second with two nurses who had not participated previously, taking 45 minutes.

Workarounds

Despite the smaller set of traces reviewed, in Round 2, we discovered two new workarounds:

WA4 **Completing checklists before discharge.** When patients return to the day treatment ward after their colonoscopy, nurses should immediately test their vital signs and fill out a checklist. In most cases, patients then remain under observation for 15 minutes before they can be discharged. During discharge, nurses follow an additional list to make sure all checks are performed. However, the second checklist is often already filled out while recording the vital signs to save time. This can result in important checks effectively being skipped. For example, the check to take out a patient's IV is often already ticked off before it is taken out. In rare cases, often during shift changes, a nurse may forget to take out the IV before sending a patient home. To mitigate, most nurses ask patients if someone took their IV out if they see them leaving the ward, even if they are not the responsible nurse.

WA5 **Discharging patients earlier than expected.** According to the prescribed process, the time between admission and discharge of patients visiting the day treatment ward should be at least two hours. In some cases, a trace may be finished much quicker. However, the system does not allow the discharge of patients from the day treatment ward before this minimum duration. To still discharge such patients, nurses may change the location of the patient from "day treatment" to "not billable". While this workaround has no effect on patient care, there may obviously be financial effects for the hospital if the financial administration of the JBZ does not catch this.

Filtering

Not all discussed traces were classified as workarounds. However, the remaining candidates were considered to be part of the core process. As this meant that no new filters could be created, our demonstration stops at Round 2.

6.4.4 Evaluation

To evaluate our findings, we discussed the identified workarounds with one physician (one of the co-authors) and one research nurse, neither of whom was involved in the previous classification interviews. Workarounds WA1, WA2, and WA3 from the first round were all considered to be pragmatic solutions for HIS limitations. While not ideal, the hospital is dependent on the HIS vendor for addressing these issues. Since these workarounds were also commonly known and the potential effects of these workarounds were limited, the experts did not see a reason to spend much effort addressing them further.

In contrast, the experts were concerned about workarounds WA4 and WA5 from the second round. Neither workaround was known by the physician, even though physicians are responsible for the overall process. Workaround WA4 may have a direct effect on the quality of patient care, which is a severe concern for a hospital. As workaround WA5 may have financial implications, it is also very important to manage any effects from this so the hospital does not miss out on resources. Given the importance of both aspects for the overall process, the expert noted that further action to manage these workarounds was essential.

6.5 Discussion

With this study, we show how to use the SWORD framework to quantitatively analyze HIS data to uncover actionable workarounds with limited domain knowledge. While the framework was originally used to discover workarounds in broad setups over full wards (Van der Waal

et al., 2024c,a) or even a complete hospital (Van der Waal et al., 2022), the operationalized framework targets a specific process.

In our case study, we demonstrate that this additional focus allows the discovery of previously unknown, actionable workarounds, despite earlier attempts to understand and optimize the process using qualitative methods. This shows that with the adjustments, the SWORD framework clearly demonstrates the added value of data-driven methods alongside established qualitative approaches to manage and improve work processes.

6.5.1 Workarounds in Healthcare

Workarounds have been studied across the medical domain, reflecting their relevance for improving clinical practice. Within this broader context, Nursing Science stands out as a field that is particularly concerned with the improvement of processes. As workarounds are clear indicators of process issues, they have attracted a strong interest in this field. Many studies concerned with workaround discovery find that administrative demands and use of technology are the most common causes for workarounds (McCord et al., 2022). These workarounds occur across diverse (international) medical settings, ranging from acute inpatient care to outpatient clinics and long-term care (Fraczkowski et al., 2020). However, despite this strong interest, the methodological quality of existing studies is often limited, with many lacking theoretical grounding (McCord et al., 2022).

Regardless of the specific medical sector, most workaround studies rely on qualitative methods such as interviews, observations, and focus groups (Fraczkowski et al., 2020; McCord et al., 2022). Studies that adopt a more quantitative approach primarily use questionnaires (Johannessen et al., 2015; Tucker, 2016; Heron and Bruk-Lee, 2020), which still suffer from the limitations of self-reporting, such as recall bias and social desirability. Only a few studies have used HIS data for more data-driven analyses, and these typically focus on a single known workaround in detail (Koppel et al., 2008; Van Onzenoort et al., 2008; Miller et al., 2011; Watson et al., 2014; Hardmeier et al., 2014; Vitoux et al., 2015). As a result, existing research offers limited support for systematically detecting previously unknown workarounds or assessing their prevalence.

Our work contributes to this literature in two ways. First, the SWORD framework is firmly grounded in both earlier published workaround cases and 'The Theory of Workarounds' (Alter, 2014), which addresses one recommendation by (McCord et al., 2022). With this case study, we bridge the gap between established, but strongly theoretical work and the pragmatic need to discover workarounds in practice. Second, we have demonstrated that our data-driven method can uncover workarounds not discovered in earlier qualitative work, thereby complementing the current qualitative state of the art. Together, these contributions move workaround research towards more systematic, theory-informed, and scalable analysis.

6.5.2 Process Mining in Healthcare

In a broader sense than workaround mining, HIS data is a great source of information, as hospitals are often required to register the vast majority of care-related activities in some way. However, extracting actionable insights from this data is notoriously challenging in the healthcare sector (Rojas et al., 2016; Munoz-Gama et al., 2022). One common issue is the relatively low data quality. We address this issue on a conceptual level with the SWORD framework by comparing individual traces to the larger body of HIS data, where both originate from the same source. This makes SWORD robust against low data quality, as long as that quality is consistent. For example, if an important activity is systematically missing in the data, it affects all traces similarly and does not distort deviation scoring. Even if such data

quality is inconsistent, the framework will highlight these differences, potentially identifying workarounds that lead to inconsistent data issues.

A second challenge for process mining in healthcare concerns the way algorithms are developed. Many process mining algorithms are tailor-made for specific studies. While that makes sense for exploratory research, they are often very difficult to transfer across hospital environments (Rojas et al., 2016). As the general applicability of the SWORD framework has been demonstrated in previous studies (Van der Waal et al., 2024a,c), albeit in less applied scenarios, the present case study shows the transferability of the SWORD framework.

6.5.3 Limitations

We recognize two main concerns with the setup of this study. First, we test the applicability of the SWORD framework only in a single case study. Although not directly related to actionable workarounds, this issue is partly mitigated with results from previous studies that discovered workarounds both within healthcare (Van der Waal et al., 2024a,c) and outside (Outmazgin et al., 2023; Löhr et al., 2024). In addition, we consider the studied case to be relatively complex to study, as it has been optimized earlier. As such, simple insights would already have been discovered, and workarounds would only be considered actionable if they were not known, despite the earlier attention on the process. Other, unoptimized, cases would likely show similar or even better results.

Second, this study is completely data-driven. While this has advantages, one clear issue is that not all activities are registered in the HIS. As such, we are only able to study the process as logged in the system, not necessarily the full process as performed. This is a fundamental limitation for SWORD and for process mining more broadly. Our findings, therefore, should not be interpreted as a replacement for interviews and observations, but as a complementary lens that can direct attention to atypical traces, support prioritization, and accelerate the discovery of issues that warrant deeper qualitative investigation.

6.6 Conclusion

The goal of this study was to detect workarounds that support process improvement using HIS data in a well-defined clinical process, while minimizing the effort required from domain experts. To this end, we operationalized the SWORD framework to obtain fine-grained insights into the colonoscopy process and complemented these in multiple rounds of expert interviews. The workarounds discovered in the first round were relatively generic and therefore less actionable. However, in the second round, we identified workarounds that were more strongly related to the colonoscopy process and carried clearer operational implications. While nurses were practicing these workarounds regularly, and in one case already attempting to mitigate their effects, the physicians responsible for the overall process quality were unaware of them. As such, these workarounds provide excellent input for targeted interventions to improve both the quality and governance of the process. This demonstrates the applicability of the SWORD framework as a method to find actionable workarounds in the medical field.

To further improve workaround mining, we suggest two main directions. First, while this study demonstrates the potential to discover actionable workarounds, the method can be explored further. In particular, where we finish our case study with two iterations, continuing may lead to even more workarounds. As such, future case studies should investigate this further. In addition to finding more workarounds that are useful for process improvement within the case, this could also lead to a more complete view of how common workarounds are.

Second, there are various theories on how workarounds may be leveraged to improve the process (Beerepoot and Van de Weerd, 2018; Outmazgin et al., 2023). A study attempting this from start to end would likely lead to more insights into which workarounds are actionable. In the future, this knowledge could be used to further direct the data side of the analysis.

Chapter 7 Conclusion

In this final chapter, we discuss the impact of this work on both research and domain practice. For this, we follow the types of contributions scientific work can have according to Ågerfalk and Karlsson (2020): Artefactual, Empirical, and Theoretical. After these contributions, we will consider the limitations and provide an outlook on the future of workaround mining.

7.1 Artefactual Contributions and Implications

The core contribution of this dissertation is the SWORD framework, which allows data-driven workaround detection. While we make use of the framework or its output in every chapter, we initially introduce it in Chapter 2. The patterns in the base framework already allow the discovery of workarounds with limited required expert effort compared to interviews and observations. We further decrease the effort with active learning in Chapter 3. By choosing the traces about which the underlying algorithm is most unsure, we ensure the expert evaluations are the most valuable for further workaround detection.

In addition to limiting the required domain expert effort, the SWORD framework provides a quantitative basis that can be used to start a conversation with experts. The targeted questions allow experts to focus on the interesting parts of the process. As some workarounds can be rare, these questions may point to parts of their work they would otherwise not consider interesting to talk about during interviews. Also, it limits the possibility of workers hiding their behavior for whatever reason. This is also seen by Outmazgin et al. (2023) as they state that the quantitative results of the SWORD framework help employees open up to talk about problems they encounter in their work, enabling them to discuss parts of their work that could otherwise be missed. The limited requirements of expert involvement are underlined by the low total time spent on interviews across all studies: less than eight hours in total. Case studies based on interviews and observations tend to require between 20 and 30 hours of expert involvement (Beerepoot et al., 2019b; Assbrock et al., 2024), which shows that with the SWORD framework, workarounds can be discovered much faster.

In our studies, we have verified the applicability of the SWORD framework in the healthcare sector only. However, researchers may use it to support workaround detection in any field, potentially with our publicly available R implementation (Van der Waal et al., 2024d). This is done by Outmazgin et al. (2023) to study the financial processes in a public academic college. Here, our framework is first used to discover workarounds, which are then studied further with additional models to enable process improvement. Alternatively, as the SWORD framework is designed to be extensible, additional patterns may be added to target new workaround perspectives. For example, Löhr et al. (2024) make such an extension in an industrial context with the LongSWORD framework, which adds two patterns to specifically focus on discovering workarounds that occur between cases.

7.2 Empirical Contributions and Implications

The artefactual contribution in the form of the SWORD framework has led to empirical contributions: the discovery and description of multiple new workarounds. For domain practice, these can be used to improve processes. Over the course of this dissertation, we found 28 workarounds in four different healthcare cases: the emergency room, cardiovascular nursing ward, and critical care unit of the University Medical Center Utrecht, and the gastroenterology ward of the Jeroen Bosch Hospital. While not all are directly useful for process improvement, we specifically discover actionable workarounds in Chapter 6. Prior to our study, the hospital investigated its colonoscopy process through a qualitative project focused on decreasing financial and time waste. Despite their insights, we find two completely new workarounds that were unknown to the physicians responsible for the quality of the process, demonstrating the complementary value of our data-driven approach to established methods. The workarounds were also considered concerning as they had a clear effect on the quality of care or the finances of the hospital.

In Chapter 3, we not only discover workarounds, but we construct rules that we use to quantify their occurrence in the complete data set. This allows us to determine how often specific workarounds are used, which is not possible with qualitative methods. Especially when trying to study them over long periods, systematically logged data is required. With this rule-based approach of discovering workarounds, we were able to monitor workarounds over much longer periods than previously possible. This enabled us to investigate the precise ratio of workarounds to normative process executions and, more importantly, how that changed over time. By studying how ten workarounds changed over thirteen years in Chapter 4, we could link four software and hardware updates to large changes in how workarounds spread through the hospital. While some updates immediately decreased the number of workarounds used, others increased it suddenly or gradually. In Chapter 5, we explored the differences between how actor roles adopt a workaround. Here, we found strong differences: Physicians adopt workarounds early and quickly, secretaries adopt them late, but very fast, and nurses learn them slowly over time, but have a broader selection of workarounds to use. In addition to the empirical value, these quantifications open completely new opportunities to generalize to theory (Ågerfalk and Karlsson, 2020). Although existing theories around workarounds were all based on interviews and observations, the SWORD framework enables us to test them using longitudinal process data.

7.3 Theoretical Contributions and Implications

With the rule-based approach of discovering workarounds in Chapter 3 that allows us to monitor workarounds on a multi-year scale, we were able to study workarounds over much longer periods than previously possible. The insights in Chapter 4 enabled us to empirically evaluate existing theories about how workarounds change over time. Multiple options have been proposed in IS literature: Often, the temporality of workarounds is considered stable (Ejnefjäll et al., 2023), sometimes even in the form of institutionalized workarounds (Azad and King, 2012; Choudrie and Zamani, 2016; Davison et al., 2019). While we find that they can indeed be stable over time, a long period of stability may be followed by a sudden change coinciding with system updates. For workarounds that do change over time, Alter (2014) theorizes that workarounds may spread through organizations through group adoption after a period of bricolage, while Safadi and Faraj (2010) suggest that workers will share workarounds with others as they discover or learn them themselves. Using our empirical data, we find that neither theory fits well. The former would present itself as sudden bursts in the ratio when groups decide to adopt a workaround, followed by plateaus, while

the latter would show an exponential growth as more workers who learn the workaround would mean more people who would actively share it. Instead, we found that the ratio of many workarounds shows linear growth, which suggests that individual motivation is often much more important for workers to adopt strategies that go against designed protocols.

We further investigated this individual adoption in Chapter 5. We found that the role an actor has in the organization strongly affects how fast they adopt a workaround, as physicians adopt workarounds much earlier than secretaries. Within the routine dynamics field, this can be explained through differences in power between the roles: Physicians have high power in the organization and therefore do not fear potential repercussions from deviating from the prescribed process (Dionysiou and Tsoukas, 2013). Secretaries, on the other hand, have low power, which makes them initially more hesitant. As support staff, however, they are more motivated to align their work with the physicians (Danner-Schröder, 2020). Confirming such theories empirically is valuable in itself, but it also directs the potential management of workarounds. In particular, when individuals hold substantial power within an organization, dissuading workarounds through rules and repercussions is likely to be less effective than for those with lower power.

7.4 Limitations

While the work shows multiple advances in workaround research, the potential of our approach is not limitless. In this section, we first discuss the conceptual limits of automatic workaround mining. Then we consider the boundaries of using event data for workaround detection. Finally, we discuss the generalizability of our work.

The definition of a workaround requires an intended deviation from the prescribed process (Alter, 2014; Ejnefjäll and Ågerfalk, 2019). That is, subjective motivation is an essential part of whether an observed deviation is a workaround. As motivation is rarely part of process data, quantitative methods can never detect workarounds with certainty (Bartelheimer et al., 2025). In our work, we solve this issue by using quantitative techniques at the start to find workarounds candidates, followed by qualitative interviews at the end to confirm or reject them as a real workaround. For the detection of individual workarounds, such as in Chapters 2 and 6, this is always sufficient. However, the same approach cannot be used for large-scale detection, like in Chapters 3 to 5. While each workaround on which the large-scale classification is based was indeed confirmed as a workaround, experts cannot evaluate this for the millions of individual instances. Therefore, it is very well possible that some workers intended to work around an obstacle in some of the instances, while the deviations were accidental in others. As these particular chapters study only the general way of working, such an approximation is sufficient for these purposes. However, care should be taken during more detailed studies using these methods.

In addition, not all workarounds may leave traces in event data (Bartelheimer et al., 2025). To detect them using the SWORD framework, workarounds must be visible in the data in some way, otherwise they can never be detected in this manner. While we find that the absence of expected or usually occurring data can also be used to find workarounds in Chapter 2, not all data can or should be logged. In these cases, our quantitative methods cannot find the workarounds. For these types of workarounds, qualitative methods would still work, showing that quantitative and qualitative detection of workarounds are complementary to each other.

Even though we use case studies over multiple wards and hospitals, we do not test our techniques and methods outside of this healthcare setting in this dissertation, which limits their generalizability. By design, this is mitigated somewhat as many of the patterns in the framework are based on retail. Therefore, it is conceivable that workaround discovery could be supported with the SWORD framework in this domain. Furthermore, additional work

shows that the SWORD framework is useful in other domains that are strongly different from hospitals, such as industry (Löhr et al., 2024) or academic finances (Outmazgin et al., 2023). Still, more studies into these domains could verify this on the one hand, while also investigating how workarounds may differ between domains in terms of evolution and change, as theoretical insights have not been tested outside of hospitals.

7.5 Outlook on Workaround Mining

Where the previous chapters show how we can find workarounds for both theoretical insights and practical process management, they also raise many more questions and goals. In this section, we will discuss what future opportunities there are for workaround mining. We first suggest two options to improve the SWORD framework itself, followed by two theoretical research directions, and one way to make workaround mining more accessible for domain practice.

The core of the SWORD framework can be further developed. As we have found in the case studies in this dissertation, the currently included patterns provide a basis for workaround detection, but more patterns can be added as we learn more about workarounds. We show in Chapter 2 that the discovery of new workarounds can be used to define new patterns. In addition, Löhr et al. (2024) add two new patterns based on the theoretical insight that one workaround may inadvertently lead to another.

We initially only distinguish traces between workarounds and non-workarounds in Chapter 2. However, not every workaround can be managed in the same way. Workarounds that are performed throughout a larger organization, such as nurses often registering patient data hours after the measurements because of limited time in almost every ward, may require impossibly large changes in the whole hospital. We address this in Chapter 6, where we create an additional distinction: actionable and non-actionable workarounds. However, in literature, we find further distinctions in classes of workarounds that require conceptually different approaches to manage. For example, Friedman et al. (2014) talk about temporary versus routinized, avoidable versus unavoidable, and deliberate versus unplanned workarounds, and Bartelheimer et al. (2025) suggest a difference between phantom, transparent, and offline workarounds. While such distinctions are not trivial to make, they increase the potential impact of workaround discovery on practical process improvement by guiding and prioritizing the way they can be managed.

Chapter 4 shows that workarounds may emerge during various incidents, such as system updates. However, it is unclear why some updates had a dramatic effect on one process, but not on another. While we were mostly concerned with the question of where workarounds originated and when they changed, turning the question around and asking what effects incidents have on processes with a workaround lens may give insights into such differences. This, in turn, may even lead to predicting where and when workarounds may start or change, allowing organizations to manage them proactively.

While the SWORD framework shows that limited expert involvement is sufficient to discover workarounds, a deeper understanding of workarounds on a theoretical level is difficult to achieve with the low number of interviews. For further insights, traditional methods such as in-depth interviews or focus groups can be used to obtain new insights into the motivations of professionals after workaround discovery using the SWORD framework. While this initially supports organizations in managing the individual workarounds, this information can also lead to further understanding of why workarounds start or change.

One direction we have not followed in this work is the opportunity to study the different ways individuals may approach the same process. While we consider specific actor roles in Chapter 6, we do not explore whether individuals learn and adopt workarounds at

the same time or speed. Such insights could provide further understanding of how people approach work and allow for more nuance in the balance between evolution through bricolage and official acceptance (Alter, 2014), spread by individuals (Safadi and Faraj, 2010), or our suggested individual adoption. In addition, it could show the starting points of workarounds better and therefore learn more about emergence events in different ways.

The motivation of both the SWORD framework in Chapter 2 and the active learning addition in Chapter 3 concerns limiting the effort experts need to spend to discover workarounds. All case studies show this effort is indeed now very low. One step necessary to unlock the full potential of the SWORD framework for workaround detection in practice is reducing the dependency on academic resources. That is, process managers should be able to find workarounds in their own processes without a full academic case study. An open-source tool already allows this (Van der Waal et al., 2024d), but further development with, for example, a dashboard plugin to the HIS would even make it accessible to domain experts with limited affinity to data.

Bibliography

van der Aalst, W. M. P., van Dongen, B. F., Herbst, J., Maruster, L., Schimm, G., and Weijters, A. J. M. M. (2003). Workflow mining: A survey of issues and approaches. *Data & Knowledge Engineering*, 47(2):237–267.

van der Aalst, W. M. P., Reijers, H. A., Weijters, A. J. M. M., van Dongen, B. F., Alves de Medeiros, A. K., Song, M., and Verbeek, H. M. W. (2007). Business process mining: An industrial application. *Information Systems*, 32(5):713–732.

Adair, J. G. (1984). The Hawthorne Effect: A Reconsideration of the Methodological Artifact. *Journal of Applied Psychology*, 69(2):334–345.

Alizadeh, M., Lu, X., Fahland, D., Zannone, N., and van der Aalst, W. M. P. (2018). Linking data and process perspectives for conformance analysis. *Computers & Security*, 73:172–193.

Alman, A., Ciccio, C. D., Haas, D., Maggi, F. M., and Nolte, A. (2020). Rule Mining with RuM. In *2nd International Conference on Process Mining*, pages 121–128.

Alter, S. (2014). Theory of Workarounds. *Communications of the Association for Information Systems*, 34.

Alvarez, R. (2008). Examining technology, structure and identity during an enterprise system implementation. *Information Systems Journal*, 18:203–224.

Aminikhanghahi, S. and Cook, D. J. (2017). A survey of methods for time series change point detection. *Knowledge and Information Systems*, 51(2):339–367.

Assbrock, A., Löhr, B., Bartelheimer, C., and Beverungen, D. (2024). Workarounds as Catalysts for Process Innovation: A Multiple Case Study on Pitfalls and Protocols. In *International Conference of Information Systems*.

Azad, B. and King, N. (2008). Enacting computer workaround practices within a medication dispensing system. *European Journal of Information Systems*, 17(3):264–278.

Azad, B. and King, N. (2012). Institutionalized computer workaround practices in a Mediterranean country: an examination of two organizations. *European Journal of Information Systems*, 21:358–372.

Barley, S. R. (1990). Images of Imaging: Notes on Doing Longitudinal Field Work. *Organization Science*, 1(3):220–247.

Bartelheimer, C., Löhr, B., Reineke, M., Aßbrock, A., and Beverungen, D. (2025). Workarounds as a Cause of Mismatches in Business Processes—: Insights from a Multiple Case Study. *Business & Information Systems Engineering*, 67:339–356.

Bartelheimer, C., Wolf, V., and Beverungen, D. (2023). Workarounds as generative mechanisms for bottom-up process innovation—Insights from a multiple case study. *Information Systems Journal*, 33:1085–1150.

Batista, E. and Solanas, A. (2018). Process Mining in Healthcare: A Systematic Review. In *9th International Conference on Information, Intelligence, Systems and Applications*.

Becker, M. C. (2004). Organizational routines: a review of the literature. *Industrial and Corporate Change*, 13(4):643–678.

Beerepoot, I., Koorn, J., van de Weerd, I., van den Hooff, B., Leopold, H., and Reijers, H. A. (2019a). Working Around Health Information Systems: The Role of Power. In *Fortieth International Conference on Information Systems*.

Beerepoot, I., Lu, X., van de Weerd, I., and Reijers, H. A. (2021). Seeing the Signs of Workarounds: A Mixed-Methods Approach to the Detection of Nurses' Process Deviations. In *Proceedings of the Annual Hawaii International Conference on System Sciences*.

Beerepoot, I., Ouali, A., van de Weerd, I., and Reijers, H. A. (2019b). Working Around Health Information Systems: To Accept Or Not To Accept? In *European Conference on Information Systems*.

Beerepoot, I. and van de Weerd, I. (2018). Prevent, Redesign, Adopt or Ignore: Improving Healthcare using Knowledge of Workarounds. In *Proceedings of the 26th European Conference on Information Systems*.

Bellaaj Elloumi, F., Sellami, M., and Bhiri, S. (2018). Avoiding resource misallocations in business processes. *Concurrency and Computation: Practice and Experience*, 32(15).

Berente, N., Seidel, S., and Safadi, H. (2019). Research Commentary—Data-Driven Computationally Intensive Theory Development. *Information Systems Research*, 30(1):50–64.

Bezemer, T., de Groot, M. C. H., Blasse, E., ten Berg, M. J., Kappen, T. H., Bredenoord, A. L., van Solinge, W. W., Hoefer, I. E., and Haitjema, S. (2019). A Human(e) Factor in Clinical Decision Support Systems. *Journal of Medical Internet Research*, 21(3):e11732.

Biddle, B. J. (1986). Recent Developments in Role Theory. *Annual Review of Sociology*, 12(1):67–92.

Boell, S. K. and Cecez-Kecmanovic, D. (2015). What is an Information System? In *Proceedings of the 48th Hawaii International Conference on System Sciences*, pages 4959–4968.

Bolt, A., de Leoni, M., and van der Aalst, W. M. P. (2018). Process variant comparison: Using event logs to detect differences in behavior and business rules. *Information Systems*, 74:53–66.

Borrego, D. and Barba, I. (2014). Conformance checking and diagnosis for declarative business process models in data-aware scenarios. *Expert Systems with Applications*, 41(11):5340–5352.

Bose, R. P. J. C. and van der Aalst, W. M. P. (2013). Discovering signature patterns from event logs. In *IEEE Symposium on Computational Intelligence and Data Mining*.

Boudreau, M. C. and Robey, D. (2005). Enacting Integrated Information Technology: A Human Agency Perspective. *Organization Science*, 16:3–18.

vom Brocke, J., Zelt, S., and Schmiedel, T. (2016). On the role of context in business process management. *International Journal of Information Management*, 36(3):486–495.

Burattin, A., Maggi, F. M., and Sperduti, A. (2016). Conformance checking based on multi-perspective declarative process models. *Expert Systems with Applications*, 65:194–211.

Cabanillas, C., Ackermann, L., Schönig, S., Sturm, C., and Mendling, J. (2020). The RALph miner for automated discovery and verification of resource-aware process models. *Software and Systems Modeling*, 19(6):1415–1441.

Choudrie, J. and Zamani, E. D. (2016). Understanding Individual User Resistance and Workarounds of Enterprise Social Networks: The Case of Service Ltd. *Journal of Information Technology*, 31(2):130–151.

Cloutier, C. and Langley, A. (2020). What Makes a Process Theoretical Contribution? *Organization Theory*, 1:1–32.

Cresswell, K. M., Mozaffar, H., Lee, L., Williams, R., and Sheikh, A. (2016). Workarounds to hospital electronic prescribing systems: a qualitative study in English hospitals. *BMJ Quality & Safety*, 26(7):542–551.

Danner-Schröder, A. (2020). Without actors, there is no action: How interpersonal interactions help to explain routine dynamics. *Review of Managerial Science*, 15(7):1913–1936.

Davison, R. M., Wong, L. H., Alter, S., and Ou, C. (2019). Adopted Globally but unusable Locally: What Workarounds Reveal about Adoption, Resistance, Compliance and non-Compliance. In *European Conference on Information Systems*.

Davison, R. M., Wong, L. H. M., Ou, C. X. J., and Alter, S. (2021). The coordination of workarounds: Insights from responses to misfits between local realities and a mandated global enterprise system. *Information & Management*, 58:103530.

Debono, D., Greenfield, D., Lipworth, W., Carter, D. J., Black, D., Hinchcliff, R., Carland, J. E., and Braithwaite, J. (2025). "I know I shouldn't but ..." the inevitable tension of using workarounds to be a "good nurse". *Frontiers in Health Services*, 5.

Debono, D. S., Greenfield, D., Travaglia, J. F., Long, J. C., Black, D., Johnson, J., and Braithwaite, J. (2013). Nurses' workarounds in acute healthcare settings: a scoping review. *BMC Health Services Research*, 13(1).

Di Ciccio, C., Marrella, A., and Russo, A. (2012). Knowledge-intensive Processes: An Overview of Contemporary Approaches. *1st International Workshop on Knowledge-Intensive Business Processes*, pages 33–47.

Di Francescomarino, C., Ghidini, C., Maggi, F. M., and Milani, F. (2018). Predictive Process Monitoring Methods: Which One Suits Me Best? In *Business Process Management*, pages 462–479.

Dionysiou, D. D. and Tsoukas, H. (2013). Understanding the (Re)Creation of Routines from Within: A Symbolic Interactionist Perspective. *Academy of Management Review*, 38(2):181–205.

Doctor, E., Hennrich, J., Eymann, T., and Buck, C. (2025). Understanding Antecedents of Nurses' and Physicians' Workaround Behavior Regarding Hospital Information Systems: Qualitative Interview Study. *Journal of Medical Internet Research*, 27:e51781.

Dunzer, S., Stierle, M., Matzner, M., and Baier, S. (2019). Conformance checking: A state-of-the-art literature review. In *Proceedings of the 11th International Conference on Subject-Oriented Business Process Management*. ACM Press.

Eck, M. L. v., Lu, X., Leemans, S. J. J., and van der Aalst, W. M. P. (2015). PM^2: A Process Mining Project Methodology. In *Advanced Information Systems Engineering*, pages 297–313. Springer International Publishing.

Ejnefjäll, T. and Ågerfalk, P. J. (2019). Conceptualizing Workarounds: Meanings and Manifestations in Information Systems Research. *Communications of the Association for Information Systems*, 45:340–363.

Ejnefjäll, T., Ågerfalk, P. J., and Hedrén, A. (2023). WORKAROUNDS IN INFORMATION SYSTEMS RESEARCH: A FIVE-YEAR UPDATE. In *European Conference on Information Systems*, page 211.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*.

Feldman, M. S. and Pentland, B. T. (2003). Reconceptualizing Organizational Routines as a Source of Flexibility and Change. *Administrative Science Quarterly*, 48(1):94–118.

Feldman, M. S. and Pentland, B. T. (2022). Routine dynamics: Toward a critical conversation. *Strategic Organization*, 20(4):846–859.

Ferneley, E. H. and Sobreperez, P. (2006). Resist, comply or workaround? an examination of different facets of user engagement with information systems. *European Journal of Information Systems*, 15(4):345–356.

Field, A. (2013). *Discovering statistics using IBM SPSS*. Sage.

Fraczkowski, D., Matson, J., and Lopez, K. D. (2020). Nurse workarounds in the electronic health record: An integrative review. *Journal of the American Medical Informatics Association*, 27(7):1149–1165.

Friedman, A., Crosson, J. C., Howard, J., Clark, E. C., Pellerano, M., Karsh, B.-T., Crabtree, B., Jaén, C. R., and Cohen, D. J. (2014). A typology of electronic health record workarounds in small-to-medium size primary care practices. *Journal of the American Medical Informatics Association*, 21(e1):e78–e83.

de la Fuente, R., Fuentes, R., Munoz-Gama, J., Riquelme, A., Altermatt, F. R., Pedemonte, J., Corvetto, M., and Sepúlveda, M. (2019). Control-flow analysis of procedural skills competencies in medical training through process mining. *Postgraduate Medical Journal*, 96(1135):250–256.

Goff, M., Hodgson, D., Bailey, S., Bresnen, M., Elvey, R., and Checkland, K. (2021). Ambiguous workarounds in policy piloting in the NHS: Tensions, trade-offs and legacies of organisational change projects. *New Technology, Work and Employment*, 36(1):17–43.

Grabski, S. V., Leech, S. A., and Schmidt, P. J. (2011). A Review of ERP Research: A Future Agenda for Accounting Information Systems. *Journal of Information Systems*, 25(1):37–78.

Grisold, T., van der Aa, H., Franzoi, S., Hartl, S., Mendling, J., and vom Brocke, J. (2024). A Context Framework for Sense-making of Process Mining Results. In *International Conference on Process Mining*, pages 57–64.

Günther, C. W., Rozinat, A., and van der Aalst, W. M. P. (2010). Activity Mining by Global Trace Segmentation. In *Business Process Management Workshops*, pages 128–139.

Handel, M. J. and Poltrock, S. (2011). Working around official applications: experiences from a large engineering project. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, pages 309–312.

Hardmeier, A., Tsourounis, C., Moore, M., Abbott, W. E., and Guglielmo, J. B. (2014). Pediatric Medication Administration Errors and Workflow Following Implementation of a Bar Code Medication Administration System. *Journal for Healthcare Quality*, 36(4):54–63.

Harlos, K. (2010). If you build a remedial voice mechanism, will they come? Determinants of voicing interpersonal mistreatment at work. *Human Relations*, 63(3):311–329.

Haux, R. (2006). Health information systems - past, present, future. *International Journal of Medical Informatics*, 75(3-4):268–281.

Heron, L. and Bruk-Lee, V. (2020). When empowered nurses are under stress: Understanding the impact on attitudes and behaviours. *Stress and Health*, 36(2):147–159.

Hofstede, G. (2011). Dimensionalizing Cultures: The Hofstede Model in Context. *Online Readings in Psychology and Culture*, 2(1).

Holden, J. D. (2001). Hawthorne effects and research into professional practice. *Journal of Evaluation in Clinical Practice*, 7(1):65–70.

van den Hooff, B. and Hafkamp, L. (2017). Dealing with Dissonance: Misfits between an EHR System and Medical Work Practices. *International Conference on Information Systems*.

Ignatiadis, I. and Nandhakumar, J. (2009). The effect of ERP system workarounds on organizational control: An interpretivist case study. *Scandinavian Journal of Information Systems*, 21(2):3.

Ilie, V. (2013). Psychological Reactance and User Workarounds. a Study in the Context of Electronic Medical Records Implementations. In *Proceedings of the 21st European Conference on Information Systems*.

Johannessen, H. A., Gravseth, H. M., and Sterud, T. (2015). Psychosocial factors at work and occupational injuries: A prospective study of the general working population in Norway. *American Journal of Industrial Medicine*, 58(5):561–567.

Killick, R., Fearnhead, P., and Eckley, I. A. (2012). Optimal Detection of Changepoints With a Linear Computational Cost. *Journal of the American Statistical Association*, 107(500):1590–1598.

Kleinbaum, D. G. and Klein, M. (2010). *Logistic Regression*. Springer New York.

Kobayashi, M., Fussell, S. R., Xiao, Y., and Seagull, F. J. (2005). Work coordination, workflow, and workarounds in a medical context. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*.

Koppel, R., Wetterneck, T., Telles, J. L., and Karsh, B.-T. (2008). Workarounds to Barcode Medication Administration Systems: Their Occurrences, Causes, and Threats to Patient Safety. *Journal of the American Medical Informatics Association*, 15(4):408–423.

Kovach, K. A. and Cathcart, C. E. (1999). Human Resource Information Systems (HRIS): Providing Business with Rapid Data Access, Information Exchange and Strategic Advantage. *Public Personnel Management*, 28(2):275–282.

Kremser, W. and Blagoev, B. (2021). The Dynamics of Prioritizing: How Actors Temporally Pattern Complex Role–Routine Ecologies. *Administrative Science Quarterly*, 66(2):339–379.

Lazovik, A., Aiello, M., and Papazoglou, M. (2004). Associating assertions with business processes and monitoring their execution. In *Proceedings of the 2nd International Conference on Service Oriented Computing*.

de Leoni, M. and van der Aalst, W. M. P. (2013). Aligning Event Logs and Process Models for Multi-perspective Conformance Checking: An Approach Based on Integer Linear Programming. In *Lecture Notes in Computer Science*, volume 8094, pages 113–129. Springer Berlin Heidelberg.

Lewis, D. D. (1998). Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. In *Machine Learning: ECML-98*, pages 4–15. Springer Berlin Heidelberg.

Lewis, D. D. and Catlett, J. (1994). Heterogeneous Uncertainty Sampling for Supervised Learning. In *Machine Learning Proceedings*, pages 148–156.

Lo, D., Cheng, H., Han, J., Khoo, S.-C., and Sun, C. (2009). Classification of software behaviors for failure detection. In *Proceedings of the 15th International Conference on Knowledge Discovery and Data Mining*.

Lu, X., Fahland, D., van den Biggelaar, F. J. H. M., and van der Aalst, W. M. P. (2016). Detecting Deviating Behaviors Without Models. In *Business Process Management Workshops*, pages 126–139.

Löhr, B., Bartelheimer, C., Köhne, F., Nordlohne, S., Alile, D., and Latten, A. (2024). Forging the LongSWORD: Exaptation and Enhancement of the SWORD Framework for Workaround Detection. In *Business Process Management Workshops*, pages 319–331.

Mahbub, K. and Spanoudakis, G. (2004). A framework for requirents monitoring of service based systems. In *Proceedings of the 2nd International Conference on Service Oriented Computing*, pages 84–93.

Mahendrawathi, E. R., van der Waal, W., Beerepoot, I., Putra, M. A. R. R., and Propitadewa, H. (2024). Understanding Student Behavior Using Active Window Tracking and Process Mining. In *Process Mining Workshops*, pages 117–128.

Malaurent, J. and Karanasios, S. (2020). Learning from workaround practices: The challenge of enterprise system implementationsin multinational companies. *Information Systems Journal*, 30(4):639–663.

Mannhardt, F., de Leoni, M., Reijers, H. A., and van der Aalst, W. M. P. (2016). Balanced multi-perspective checking of process conformance. *Computing*, 98(4):407–437.

Martin, N. and Beerepoot, I. (2024). Unveiling Use Cases for Human Resource Mining: A Framework of Past and Future Research. *Business & Information Systems Engineering*, 67:531–550.

Martin, N., Fischer, D. A., Kerpedzhiev, G. D., Goel, K., Leemans, S. J. J., Röglinger, M., van der Aalst, W. M. P., Dumas, M., La Rosa, M., and Wynn, M. T. (2021). Opportunities and Challenges for Process Mining in Organizations: Results of a Delphi Study. *Business & Information Systems Engineering*, 63(5):511–527.

McCord, J. L., Lippincott, C. R., Abreu, E., and Schmer, C. (2022). A Systematic Review of Nursing Practice Workarounds. *Dimensions of Critical Care Nursing*, 41(6):347–356.

Miller, D. F., Fortier, C. R., and Garrison, K. L. (2011). Bar Code Medication Administration Technology: Characterization of High-Alert Medication Triggers and Clinician Workarounds. *Annals of Pharmacotherapy*, 45(2):162–168.

Munoz-Gama, J., Martin, N., Fernandez-Llatas, C., Johnson, O. A., Sepúlveda, M., Helm, E., Galvez-Yanjari, V., Rojas, E., Martinez-Millana, A., Aloini, D., Amantea, I. A., Andrews, R., Arias, M., Beerepoot, I., Benevento, E., Burattin, A., Capurro, D., Carmona, J., Comuzzi, M., Dalmas, B., de la Fuente, R., Di Francescomarino, C., Di Ciccio, C., Gatta, R., Ghidini, C., Gonzalez-Lopez, F., Ibanez-Sanchez, G., Klasky, H. B., Kurniati, A. P., Lu, X., Mannhardt, F., Mans, R., Marcos, M., de Carvalho, R. M., Pegoraro, M., Poon, S. K., Pufahl, L., Reijers, H. A., Remy, S., Rinderle-Ma, S., Sacchi, L., Seoane, F., Song, M., Stefanini, A., Sulis, E., ter Hofstede, A. H. M., Toussaint, P. J., Traver, V., Valero-Ramon, Z., van de Weerd, I., van der Aalst, W. M. P., Vanwersch, R., Weske, M., Wynn, M. T., and Zerbato, F. (2022). Process mining for healthcare: Characteristics and challenges. *Journal of Biomedical Informatics*, 127:103994.

Nardelli, G. (2017). Innovation dialectics: an extended process perspective on innovation in services. *The Service Industries Journal*, 37(1):31–56.

Nguyen, H., Dumas, M., La Rosa, M., Maggi, F. M., and Suriadi, S. (2016). Business process deviance mining: review and evaluation. *ArXiv e-prints*.

Nigam, A., Huising, R., and Golden, B. (2016). Explaining the Selection of Routines for Change during Organizational Search. *Administrative Science Quarterly*, 61(4):551–583.

van Onzenoort, H. A., van de Plas, A., Kessels, A. G., Veldhorst-Janssen, N. M., van der Kuy, P.-H. M., and Neef, C. (2008). Factors influencing bar-code verification by nurses during medication administration in a Dutch hospital. *American Journal of Health-System Pharmacy*, 65(7):644–648.

Osuna, E., Freund, R., and Girosi, F. (1997). An Improved Training Algorithm for Support Vector Machines. In *Neural Networks for Signal Processing VII. Proceedings of the 1997 IEEE Signal Processing Society Workshop*, pages 276–285.

Outmazgin, N., der Waal, W. van., Beerepoot, I., Hadar, I., de Weerd, I. van., and Soffer, P. (2023). From Automatic Workaround Detection to Process Improvement: A Case Study. In *Business Process Management Forum*, pages 372–390.

Outmazgin, N. and Soffer, P. (2014). A process mining-based analysis of business process work-arounds. *Software & Systems Modeling*, 15(2):309–323.

Outmazgin, N., Soffer, P., and Hadar, I. (2020). Workarounds in Business Processes: A Goal-Based Analysis. In *Advanced Information Systems Engineering*, pages 368–383.

Pentland, B., Vaast, E., and Wolf, J. R. (2021). Theorizing Process Dynamics with Directed Graphs: A Diachronic Analysis of Digital Trace Data. *MIS Quarterly*, 45(2):967–984.

Pentland, B. T. and Feldman, M. S. (2008). Designing routines: On the folly of designing artifacts, while hoping for patterns of action. *Information and Organization*, 18(4):235–250.

Petrides, L. A., McClelland, S. I., and Nodine, T. R. (2004). Costs and benefits of the workaround: inventive solution or costly alternative. *International Journal of Educational Management*, 18(2):100–108.

Powers, D. M. W. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *ArXiv e-prints*.

Pufahl, L., Stiehle, F., Ihde, S., Weske, M., and Weber, I. (2025). Resource allocation in business process executions—A systematic literature study. *Information Systems*, 132:102541.

Ras, G., van Gerven, M., and Haselager, P. (2018). Explanation Methods in Deep Learning: Users, Values, Concerns and Challenges. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*, pages 19–36. Springer International Publishing.

Rebuge, Á. and Ferreira, D. R. (2012). Business process analysis in healthcare environments: A methodology based on process mining. *Information Systems*, 37(2):99–116.

Reineke, M., Löhr, B., Aßbrock, A., Bartelheimer, C., and Beverungen, D. (2025). From Temporary Fixes to Informed Decisions—Design Echelons for Evaluating Workarounds. In *Business Process Management*, pages 541–557.

Röder, N., Wiesche, M., and Schermann, M. (2014). A Situational Perspective on Workarounds in IT-Enabled Business Processes: a Multiple Case Study. In *Proceedings of the 22nd European Conference on Information Systems*.

Rojas, E., Munoz-Gama, J., Sepúlveda, M., and Capurro, D. (2016). Process mining in healthcare: A literature review. *Journal of Biomedical Informatics*, 61:224–236.

Rozinat, A. and van der Aalst, W. M. P. (2008). Conformance checking of processes based on monitoring real behavior. *Information Systems*, 33(1):64–95.

Safadi, H. and Faraj, S. (2010). The role of workarounds during an opensource electronic medical record system implementation. In *International Conference on Information Systems*.

Schein, A. I. and Ungar, L. H. (2007). Active learning for logistic regression: an evaluation. *Machine Learning*, 68(3):235–265.

Settles, B. (2009). Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Soffer, P., Outmazgin, N., Hadar, I., and Tzafrir, S. (2023). Why Work Around the Process? analyzing Workarounds Through the Lens of the Theory of Planned Behavior. *Business & Information Systems Engineering*, 65(4):369–389.

Song, M., Günther, C. W., and van der Aalst, W. M. P. (2009). Trace Clustering in Process Mining. In *Business Process Management Workshops*, pages 109–120.

Sordo, M. and Zeng, Q. (2005). On Sample Size and Classification Accuracy: A Performance Comparison. In Oliveira, J. et al., editors, *Biological and Medical Data Analysis*, volume 3745 of *LNBI*, page 193–201.

Stein Dani, V., Leopold, H., van der Werf, J. M. E. M., Lu, X., Beerepoot, I., Koorn, J. J., and Reijers, H. A. (2022). Towards Understanding the Role of the Human in Event Log Extraction. In *Business Process Management Workshops*, pages 86–98.

Swennen, M., Martin, N., Janssenswillen, G., Jans, M., Depaire, B., Caris, A., and Vanhoof, K. (2016). Capturing Resource Behaviour From Event Logs. In *Proceedings of the 6th International Symposium on Data-driven Process Discovery and Analysis*, pages 130–134.

Symons, V. J. (1991). Impacts of information systems: four perspectives. *Information and Software Technology*, 33(3):181–190.

Taymouri, F., La Rosa, M., Dumas, M., and Maggi, F. M. (2021). Business process variant analysis: Survey and classification. *Knowledge-Based Systems*, 211:106557.

Tucker, A. L. (2016). The Impact of Workaround Difficulty on Frontline Employees' Response to Operational Failures: A Laboratory Experiment on Medication Administration. *Management Science*, 62(4):1124–1144.

Tuominen, T., Edvardsson, B., and Reynoso, J. (2020). Institutional change and routine dynamics in service ecosystems. *Journal of Services Marketing*, 34(4):575–586.

van Alphen, I., Meulenbeld, A., van de Weerd, I., Reijers, H. A., and van der Waal, W. (2026). Workarounds Beware: Comparing Machine Learning Algorithms with Active Learning. In *Business Process Management Workshops*. In print.

Veldman, J., Lu, X., van der Waal, W., Dees, M., and van de Weerd, I. (2023). Generating Process Anomalies with Markov Chains: A Pattern-Driven Approach. In *Process Mining Workshops*, pages 123–135.

van de Ven, A. H. and Poole, M. S. (1995). Explaining development and change in organizations. *The Academy of Management Review*, 20(3):510–540.

Verbeek, H. M. W. (2020). Process discovery contest 2020.

Vitoux, R. R., Lehr, J., and Chang, H. (2015). Eliminating Clinical Workarounds through Improved Smart Pump Drug Library Use. *Biomedical Instrumentation & Technology*, 49(s4):23–28.

Vogelsmeier, A. A., Halbesleben, J. R. B., and Scott-Cawiezell, J. R. (2008). Technology Implementation and Workarounds in the Nursing Home. *Journal of the American Medical Informatics Association*, 15(1):114–119.

van der Waal, W., Beerepoot, I., van de Weerd, I., and Reijers, H. A. (2022). The SWORD is Mightier Than the Interview: A Framework for Semi-automatic WORkaround Detection. In *Business Process Management*, pages 91–106.

van der Waal, W. and Reijers, H. A. (2025). Resource-Oriented Workaround Analysis – A Case Study. In *Business Process Management Workshops*, pages 262–273.

van der Waal, W., van de Weerd, I., Beerepoot, I., Lu, X., Kappen, T., Haitjema, S., and Reijers, H. A. (2024a). Putting the SWORD to the Test: Finding Workarounds with Process Mining. *Business & Information Systems Engineering*, 67(2):171–190.

van der Waal, W., van de Weerd, I., Beerepoot, I., and Reijers, H. A. (2024b). The Emergence and Evolution of Workarounds: A Study of Stability and Change. In *International Conference on Information Systems*.

van der Waal, W., van de Weerd, I., Haitjema, S., Kappen, T., and Reijers, H. A. (2024c). Whetting the SWORD: Detecting Workarounds by Using Active Learning and Logistic Regression. In *Proceedings of the 57th Hawaii International Conference on System Sciences*, pages 3687–3696.

van der Waal, W., van de Weerd, I., and Reijers, H. A. (2024d). SWORD Extractor Pulling Workarounds out the Eventlog. In *ICPM Tool Demonstration Track*.

Warrens, M. J. (2015). Five ways to look at Cohen's kappa. *Journal of Psychology & Psychotherapy*, 5(4):1.

Watson, A., Skipper, C., Steury, R., Walsh, H., and Levin, A. (2014). Inpatient Nursing Care and Early Warning Scores: A Workflow Mismatch. *Journal of Nursing Care Quality*, 29(3):215–222.

van de Weerd, I., Vollers, P., Beerepoot, I., and Fantinato, M. (2019). Workarounds in retail work systems: prevent, redesign, adopt or ignore? In *Proceedings of the 27th European Conference on Information Systems*.

Weinzierl, S., Wolf, V., Pauli, T., Beverungen, D., and Matzner, M. (2020). Detecting Workarounds in Business Processes-a Deep Learning method for Analyzing Event Logs. In *Proceedings of the 28th European Conference on Information Systems*.

Weinzierl, S., Wolf, V., Pauli, T., Beverungen, D., and Matzner, M. (2022). Detecting temporal workarounds in business processes – A deep-learning-based method for analysing event log data. *Journal of Business Analytics*, 5(1):76–100.

Westphal, J., Lancaster, R., and Park, D. (2013). Work-Arounds Observed by Fourth-Year Nursing Students. *Western Journal of Nursing Research*, 36(8):1002–1018.

Wijnhoven, F., Hoffmann, P., Bemthuis, R., and Boksebeld, J. (2023). Using process mining for workarounds analysis in context: Learning from a small and medium-sized company case. *International Journal of Information Management Data Insights*, 3(1):100163.

Wolf, V. and Beverungen, D. (2019). Conceptualizing the impact of workarounds: an organizational routines perspective. In *Proceedings of the 27th European Conference on Information Systems*.

Wong, L. H. M., Hurbean, L., Davison, R. M., Ou, C. X. J., and Muntean, M. (2022). Working around inadequate information systems in the workplace: An empirical study in Romania. *International Journal of Information Management*, 64:102471.

Wynn, M. T., Poppe, E., Xu, J., ter Hofstede, A. H. M., Brown, R., Pini, A., and van der Aalst, W. M. P. (2017). ProcessProfiler3D: A visualisation framework for log-based process performance comparison. *Decision Support Systems*, 100:93–108.

Yin, R. K. (2013). *Case Study Research. Design and Methods*. SAGE Publications, fifth edition.

Yoo, J.-C. and Han, T. H. (2009). Fast Normalized Cross-Correlation. *Circuits, Systems and Signal Processing*, 28(6):819–843.

Zhao, W., Liu, H., Dai, W., and Ma, J. (2015). An entropy-based clustering ensemble method to support resource allocation in business process management. *Knowledge and Information Systems*, 48(2):305–330.

Zhou, X., Ackerman, M., and Zheng, K. (2011). CPOE Workarounds, Boundary Objects, and Assemblages. In *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems*, pages 3353–3362.

Ågerfalk, P. J. and Karlsson, F. (2020). Artefactual and empirical contributions in information systems research. *European Journal of Information Systems*, 29(2):109–113.

List of Figures

1.1 The structure of this thesis 6

2.1 The phases of our research approach 12
2.2 The construction of the patterns in the SWORD framework 14
2.3 Workaround with multiple resources 22
2.4 Workaround with single resource 22
2.5 Timestamps of all orders in the ER 23
2.6 Workaround with delayed registration 23
2.7 The required steps to apply the SWORD framework 26
2.8 Part of the ER interface that keeps track of patients 30

3.1 A schematic overview of how we create, train, and evaluate a model 39
3.2 The approximate precision and recall of the model 44

4.1 Categorization of workarounds 56
4.2 ER3 - Follow-up tests before Physician Visit 57
4.3 CCU1 - Late Logging of Measurement 57
4.4 CCU2 - Late Logging of Order 57
4.5 CCU3 - Late discharge 57
4.6 ER7 - X-ray close to discharge 57
4.7 ER1 - Events after Discharge 58
4.8 ER5 - Triage late in trace 58
4.9 ER2 - Orders before physician visit 58
4.10 ER4 - Orders, Measurements, Triage, or Treatment before nurse visit 58
4.11 ER6 - Few pain scores before Discharge 58
4.12 The ratio of the events executed after discharge 61
4.13 The change in the evolution category of workarounds. 62

5.1 Screenshot of the process in HiX 71
5.2 Process executions per resource role 72
5.3 Workarounds per resource role 72
5.4 The ratio of workarounds to normative process executions for each role 73

6.1 The SWORD framework cycle to extract workarounds from data. 80
6.2 The simplified patient pathway for a typical colonoscopy. 83
6.3 An overview of the two processing and analysis rounds in the case study. . . . 85

List of Tables

1.1 The hospitals and wards per chapter. 4

2.1 An example of a complete workaround snapshot 13
2.2 Explanation and examples of the key terms in this chapter 15
2.3 Workaround detection patterns 16
2.4 Workaround detection patterns with the required data 19
2.5 Ranking of the traces using Z-scores 25
2.6 Detected workarounds per pattern 28

3.1 Two example patterns for each perspective in the SWORD framework 40
3.2 The patterns scores and the score assigned by the Logistic Regression model. . 41
3.3 The workaround types, explanations and rules discovered at the ER 43
3.4 The approximate recall of the final model per workaround type. 45

4.1 Data Usage Overview 53
4.2 Workaround Indicators 54

6.1 Part of a trace from the event log. 81
6.2 The ten traces with the highest deviations 82
6.3 The patterns from the SWORD framework applicable to this case study 86

Publication List

Included in this dissertation:

- van der Waal, W., van de Weerd, I., Beerepoot, I., Lu, X., Kappen, T., Haitjema, S., and Reijers, H. A. (2024a). Putting the SWORD to the Test: Finding Workarounds with Process Mining. *Business & Information Systems Engineering*, 67(2):171–190
- van der Waal, W., van de Weerd, I., Beerepoot, I., and Reijers, H. A. (2024b). The Emergence and Evolution of Workarounds: A Study of Stability and Change. In *International Conference on Information Systems*
- van der Waal, W., van de Weerd, I., Haitjema, S., Kappen, T., and Reijers, H. A. (2024c). Whetting the SWORD: Detecting Workarounds by Using Active Learning and Logistic Regression. In *Proceedings of the 57th Hawaii International Conference on System Sciences*, pages 3687–3696
- van der Waal, W. and Reijers, H. A. (2025). Resource-Oriented Workaround Analysis – A Case Study. In *Business Process Management Workshops*, pages 262–273

Additional publications:

- van der Waal, W., Beerepoot, I., van de Weerd, I., and Reijers, H. A. (2022). The SWORD is Mightier Than the Interview: A Framework for Semi-automatic WORkaround Detection. In *Business Process Management*, pages 91–106
- Outmazgin, N., der Waal, W. van., Beerepoot, I., Hadar, I., de Weerd, I. van., and Soffer, P. (2023). From Automatic Workaround Detection to Process Improvement: A Case Study. In *Business Process Management Forum*, pages 372–390
- Veldman, J., Lu, X., van der Waal, W., Dees, M., and van de Weerd, I. (2023). Generating Process Anomalies with Markov Chains: A Pattern-Driven Approach. In *Process Mining Workshops*, pages 123–135
- Mahendrawathi, E. R., van der Waal, W., Beerepoot, I., Putra, M. A. R. R., and Propitadewa, H. (2024). Understanding Student Behavior Using Active Window Tracking and Process Mining. In *Process Mining Workshops*, pages 117–128
- van der Waal, W., van de Weerd, I., and Reijers, H. A. (2024d). SWORD Extractor Pulling Workarounds out the Eventlog. In *ICPM Tool Demonstration Track*
- van Alphen, I., Meulenbeld, A., van de Weerd, I., Reijers, H. A., and van der Waal, W. (2026). Workarounds Beware: Comparing Machine Learning Algorithms with Active Learning. In *Business Process Management Workshops*. In print

Curriculum Vitae

Wouter van der Waal was born on 17 May 1992 in Pijnacker, the Netherlands. He earned a Bachelor's degree in Artificial Intelligence from Radboud University Nijmegen in 2015. He then moved to Utrecht University and obtained his Master's degree in Game and Media Technology in 2020.

In 2021, he began his PhD in Utrecht University's Process Science group, focusing on workaround mining. He conducted his research primarily in hospitals, collaborating closely with the University Medical Center Utrecht and the Jeroen Bosch Hospital in 's-Hertogenbosch. He published his work in the Business & Information Systems Engineering journal and presented his findings at major conferences, including the International Conference on Business Process Management, the Hawaii International Conference on System Sciences, and the International Conference on Information Systems. His work has been recognized with several awards, including the runner-up best student paper award at the 2022 International Conference on Business Process Management and a best paper in track award at the 2024 International Conference on Information Systems.

Beyond research, Wouter has contributed to education by supervising multiple Bachelor's and Master's theses and teaching courses such as Process Modeling, Strategic Management of Organizations and ICT, and Academic Skills. Among Information Science students, he is best known for teaching statistics in the course Scientific Research Methods.

