Tampereen yliopisto

Janne Sarja

DISCOVERING KNOWLEDGE WORK TASKS FROM SEQUENTIAL EVENT DATA

Master's Thesis
Faculty of Management and Business
Examiner: Hongxiu Li
Examiner: Jukka Huhtamäki
December 2023

ABSTRACT

Janne Sarja: Discovering knowledge work tasks from sequential event data.
Master's Thesis
Tampere University
Information and Knowledge Management
December 2023

As the use of digital tools and systems increases in today's knowledge-work-focused organizations, improving the work performed in digital systems has gained more attention. Digitalization is a key driver for knowledge work productivity through factors like improved communication, task automation, and artificial intelligence assistants. Monitoring and analyzing the work performed in digital systems can offer insights into how to realize all the potential of digital solutions. The thesis aims to contribute to this topic by developing a new analytical solution for analyzing frequent tasks performed in digital systems. The thesis is implemented in collaboration with a case organization and aims to address an organizational problem defined by the case organization. The case organization collects event data from digital systems using background software installed on knowledge workers' computers. The research problem addressed in the thesis is how to identify frequently performed tasks from the collected data. Furthermore, the goal is to develop a methodology to analyze the identified tasks further and get insights into how to improve the work.

The research implementation started by conducting a literature review. The first objective of the literature review is to gain background knowledge on key topics related to tasks performed in digital environments. The second objective is to understand how to execute the data analysis with the main focus on discovering potential methods and algorithms to identify frequent patterns from sequence data. For the empirical research part, action research is selected as the research strategy. Action research uses an iterative approach to solve organizational issues through action and reflection. Action research focuses both on addressing the organizational issue and creating actionable theory. The research is carried out by first developing an Apriori-based algorithm to identify frequent task patterns from the data collected by the case organization. Next, multiple analysis and result representation techniques are developed to draw insights from the identified tasks. As per action research guidelines, the algorithm and results were iteratively improved. The feedback for improvements was collected from evaluation and reflection sessions organized with personnel from the case organization.

The literature review and empirical research helped to identify key areas where the task analysis results can be utilized to improve work efficiency. These key areas are task simplification, standardization, and automation. An analytical framework was formulated as a result of the research to generate an actionable theory. As a cornerstone to the framework, an approach with two algorithms is proposed to discover tasks from the data, where the first algorithm focuses on matching tasks with predefined structures, and the second algorithm aims to identify the most frequently performed tasks. For the result representation, a key factor is to have support for grouping the tasks so that the tasks with similar features are grouped together. Another key factor for supporting the analysis of results is the option to filter the tasks based on their features. Lastly, visualizing task flows on the digital system window level, displaying task volumes, and highlighting user activity-related metrics were identified to be key visualizations to support the insights generation.

Keywords: Digitalization, knowledge work, frequent tasks, data analysis, action research

TIIVISTELMÄ

Janne Sarja: Tietotyöhön liittyvien työtehtävien löytäminen sekvenssitapahtumadatasta.
Diplomityö
Tampereen yliopisto
Tietojohtaminen
Joulukuu 2023

Digitaalisten työkalujen ja järjestelmien käytön lisääntyminen tietotyöhön keskittyneissä organisaatioissa on kasvattanut kiinnostusta digitaalisen työn tehostamista kohtaan. Digitalisaatio voi parantaa tietotyön tehokkuutta esimerkiksi parempien kommunikaatiojärjestelmien käytön, tehtävien automatisoinnin sekä tekoälyn hyödyntämisen avulla. Digitaalisissa järjestelmissä suoritetun työn seuranta ja analysointi voi tarjota tärkeitä löydöksiä siitä, kuinka digitalisaation potentiaalia voidaan hyödyntää optimaalisesti. Tämä diplomityö keskittyy tähän aihealueeseen kehittämällä analytiikkaratkaisun digitaalisissa järjestelmissä suoritettavien toistuvien tehtävien analysointiin. Työ toteutetaan yhteistyössä kohdeorganisaation kanssa ja se keskittyy kohdeorganisaation määrittelemään ongelmaan. Kohdeorganisaatio kerää tietojärjestelmiin liittyvää tapahtumadataa tietotyöntekijöiden tietokoneille asennetun taustaohjelman avulla. Diplomityössä ratkaistava tutkimusongelma on, kuinka toistuvasti suoritettavia työtehtäviä voidaan tunnistaa kerätystä datasta. Lisäksi tavoitteena on kehittä ratkaisuja löydettyjen toistuvien työtehtävien analysointiin, joiden avulla voidaan tehdä johtopäätöksiä siitä, kuinka työntekoa voitaisiin tehostaa.

Tutkimuksen toteutus alkoi kirjallisuuskatsauksen suorittamisella. Kirjallisuuskatsauksen ensimmäinen tavoite oli kartoittaa taustatietoa digitaalisissa järjestelmissä suoritetuista työtehtävistä. Toinen kirjallisuuskatsauksen tavoite oli tutustua data-analyysin toteuttamiseen sekä löytää sopivia metodeja ja algoritmeja toistuvien kuvioiden löytämiseen sekvenssidatasta. Empiirisen tutkimusosuuden tutkimusstrategiaksi valittiin toimintatutkimus. Toimintatutkimus käyttää iteratiivista menetelmää, joka perustuu toimintaan ja pohdintaan organisaation asettaman ongelman ratkaisemiseksi. Toimintatutkimus keskittyy sekä käytännöllisten ongelmien ratkaisuun organisaatiossa että tieteellisen tiedon luomiseen. Tutkimus toteutettiin kehittämällä aluksi Apriori-menetelmään pohjautuva algoritmi toistuvien työtehtävien löytämiseksi kohdeorganisaation keräämästä datasta. Seuraavaksi kehitettiin useita analyysi- ja tulosten visualisointimenetelmiä helpottamaan johtopäätösten tekemistä tunnistetuista toistuvista työtehtävistä. Toimintatutkimuksen mukaisesti kehitettyä algoritmia ja tulosten analyysiä kehitettiin iteratiivisesti eteenpäin. Kehittämiseen käytettyä palautetta kerättiin kohdeorganisaatiosta järjestämällä tulosten arviointiin ja pohdintaan perustuvia sessioita kohdeorganisaation työntekijöiden kanssa.

Kirjallisuuskatsaus ja empiirinen tutkimus auttoivat tunnistamaan avaintekijöitä, joiden avulla toteutetun analyysin tuloksia voidaan hyödyntää työnteon tehostamisessa. Nämä avaintekijät ovat työtehtävien yksinkertaistaminen, standardisointi ja automatisointi. Tieteellisen tiedon tuottamista varten luotiin kehys, jolla tutkimuksen lopputuloksia pyritään havainnollistamaan. Kehyksen perustana toimii kahden algoritmin järjestelmä, jonka avulla pyritään löytämään työtehtäviä datasta. Ensimmäinen algoritmi keskittyy löytämään tehtäviä, joiden rakenne on määritelty etukäteen. Toinen algoritmi vastaa kaikista toistuvimpien työtehtävien löytämisestä. Tulosten esittämisen kannalta yksi avaintekijä on tehtävien ryhmittely niin, että tehtävät, joilla on samankaltaisia piirteitä kuuluvat samaan ryhmään. Toinen avaintekijä tulosten analysoinnin tukemiseksi on mahdollisuus suodattaa ja löytää tehtäviä haluttujen ominaisuuksien mukaan. Myös tehtäväpolkujen visualisointi tietojärjestelmien ikkunoiden tasolla, tehtävien suoritusmäärien ilmoittaminen, sekä käyttäjätoimintoihin perustuvien metriikoiden korostaminen ovat keskeisiä tekijöitä, jotka tukevat löydettyihin tehtäviin liittyvien johtopäätösten tekemistä.

Avainsanat: Digitalisaatio, tietotyö, toistuvat työtehtävät, data-analyysi, toimintatutkimus

PREFACE

Firstly, I would like to express my appreciation to Henri Wiik and Kustaa Kivelä for generously dedicating their time to this thesis despite the busy nature of their roles as start-up founders. Their insightful contributions and willingness to share their expertise have been instrumental in shaping the course of this research journey. Furthermore, I extend my thanks to Hongxiu Li and Jukka Huhtamäki for their exceptional guidance as supervisors. Their feedback and patience with the slow progress of the thesis have played a pivotal role in steering this thesis toward fruition.

Espoo, 8 December 2023

Janne Sarja

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

AI	Artificial Intelligence
BPM	Business Process Management
CRM	Customer Relationship Management
DFG	Directly-Follows Graph
ERP	Enterprise Resource Planning
ES	Enterprise System
IT	Information Technology
KDD	knowledge Discovery from Data
RPA	Robotic Process Automation
WFM	Workflow Management

1. INTRODUCTION

Drucker (1999) has stated that where the 20th century saw manufacturing productivity increase by fifty-fold, the focus in the 21st century should be to increase the productivity of knowledge workers. Nowadays, knowledge work is widely performed in digital environments (Wang et al. 2020). Digitalization has brought a wide range of digital systems to workplaces that aim to help employees perform their daily tasks more efficiently (Plesner et al. 2018; Ahmad & Van Looy 2020). Digitalization is seen as an essential driver to boost the productivity of knowledge work through factors like task automation and artificial intelligence (AI) assistants. Overall, the job descriptions in many companies have shifted to a direction where humans and technology work in partnership to utilize both parties' respective strengths optimally. (Holford 2019)

Digitalization has introduced many new technological solutions, like digital self-services, analytical systems, and enterprise resource planning (ERP) systems (Plesner et al. 2018; Madakam et al. 2019). The introduction of these new digital systems has also increased the demand for monitoring and enhancement capabilities. Digitalization has created more opportunities for business process automation, which has subsequently increased the demand for process discovery and monitoring (El-Gharib & Amyot 2023). To tackle this demand, ERP systems usually encompass ready-made capabilities to monitor and control business-related tasks in the digital space (Gurău 2020). Business process management (BPM) is a discipline developed to provide tools for mapping, improving, designing, administrating, and analyzing business processes (Benner & Tushman 2003; Weske 2012). Further on, process mining is a technological solution developed to help organizations understand as-is processes and discover routines that can be automated (El-Gharib & Amyot 2023).

The subject of this thesis is connected to the domain of monitoring and improving business-related activities within the digital space. The case company, Workfellow, is a start-up developing a work intelligence platform. The product collects information about how digital systems are used by extracting IT-system usage data directly from knowledge workers' computers. The collected data is analyzed to offer insights into work methods, system usage, and business process execution. The case company is continuously developing new ways to analyze the collected data and provide new insights to improve

digital work. A need to enhance task-level analysis capabilities in the case company led to the formulation of the thesis topic.

This thesis aims to design and implement new capabilities to analyze the data collected by the case company. The motivation for developing new capabilities is to enhance the analysis of recurring tasks and workflows within the case company's product. The primary goal established by the case company is to investigate a method for initially identifying key recurring tasks within the collected data and subsequently analyzing these identified recurring tasks. Based on this objective, two main research questions are formulated:

Q1: How to discover frequent patterns from the window navigation data to find often performed tasks and workflows?

Q2: How to formulate and represent the results to support the analysis of often performed tasks and workflows?

To answer the research questions, both a literature review and empirical research are performed. The first objective of the literature review is to gain an understanding of relevant insights and implications that can be formed from the task analysis results. This understanding is an essential prerequisite when designing how the results are represented. The second objective of the literature review is to understand potential methods to discover frequent patterns from a data set and gain a holistic understanding of how to perform the data analysis part. The empirical research focuses on iteratively improving the developed analysis based on feedback received from domain experts.

The rest of the thesis is organized into six sections. The second section discusses topics related to work in digital systems and covers some related research fields. The third section investigates the theoretical background of implementing the data analysis and discusses the options for finding frequent patterns from a data set. The fourth section describes the research design by connecting the choices made in the research with the existing research theory. The fifth section describes the process of developing the new analytical capabilities. The sixth section discusses the iterative changes made to the analytical approach during the action research based on the received feedback. Lastly, chapter 7 summarizes and concludes the findings of the research.

2. DIGITAL WORK

Even though the topic of the thesis is relatively data-focused, it is grounded in the practical needs of an organization. Hence, it is important to understand the practical implications of the thesis's results. The organization analyses data collected from the computers of knowledge workers. A key topic is to discover how finding repetitive patterns can improve knowledge work and get an overall understanding of the underlying industry and domain. Theoretical knowledge gained from this area will affect and help with the choices made later in this research. It will also support the result interpretation.

2.1 Digitalization of Knowledge Work

A key concept to understand is how work is performed in digital information systems. The workers performing this work in digital systems could be categorized broadly as knowledge workers in this research's context. Knowledge work is a widely researched topic, but multiple authors have stated that there seems to be no clear definition for the term knowledge work (Heidary Dahooie et al. 2011; Palvalin 2018). Coming up with a strict definition for the term knowledge work will not be too relevant to the thesis. The relevant deduction is that work performed in digital systems is commonly carried out by some form of knowledge workers. Understanding knowledge work and its productivity are critical concepts for building knowledge about digital work optimization.

The internet has enabled most knowledge work to be performed entirely digitally and remotely (Wang et al. 2020). The digitalization of workplaces has many implications for how the work is performed. Digitalization can be seen as the introduction of a wide range of technological solutions, like digital self-services and new analytical systems, to support the work (Plesner et al. 2018). As an increasing amount of work consists of digital components, Baiyere et al. (2023) suggest that categorizing all work with digital components as digital work would make the term digital work too broad. The digital work in this chapter focuses mostly on the work that is organized and shaped by digital technologies. This form of work can be categorized as digital-enabled work, which is seen as a shade of digital work.(Baiyere et al. 2023). Generally, the digitalization of work is seen to make knowledge work more efficient through factors like reducing the time it takes to perform a task, automating mundane work, and getting assistance from AI (Holford 2019).

The rapid technological improvements have also changed workplaces at a fast pace (Brahma et al. 2021). In the context of digital work, the introduction of digital technologies

is perceived as a tool for enabling or facilitating the execution of work (Baiyere et al. 2023). In some cases, the facilitating technology can simplify routine tasks but might also increase standardization, which can be seen to reduce employees' freedom. On the other hand, by completely automating routine tasks, technological solutions shift employees' focus on tasks with higher cognitive demands and more creativity. (Schwarzmüller et al. 2018) Standardization of processes with the help of digitalization can help organizations scale up their processes. Standardization of processes helps to adapt them across the organization and makes the business more stable and predictable when an organization is growing (Brahma et al. 2021).

Digitalization affects communication and information availability. For example, digital communication channels continuously increase available information (Schwarzmüller et al. 2018). The increase in available information can even result in information overload (Belabbes et al. 2023). Proper and easy management of task-relevant information is an important factor in task performance. Well-organized information can eliminate task switches occurring when switching between the core task activities and finding relevant information. (Kersten & Murphy 2015) Poorly implemented information management tools can often hamper knowledge workers' productivity (Simperl et al. 2010). To tackle the problem, for example, an ERP system can be used to help integrate and unify all the data collected from different sources under one system (Gurău 2020).

For knowledge workers, digitalization has also led to the need to adopt a new set of skills. Increased technologization of workplaces means that the workers need more IT competencies, and basic computer knowledge starts to be a required skill for any knowledge work (Schwarzmüller et al. 2018). Furthermore, the automatization of the processes has also led to role changes for many management-level knowledge workers, who are now also in charge of automation efficiency (Brahma et al. 2021).

The application and IT-system catalog for digitalized organizations could be huge. Spreadsheets, office applications, management information systems, and ERP systems are examples of applications that can be used to perform business processes in an organization (Madakam et al. 2019). Tasks performed with digital enterprise systems (ES) are especially interesting for the case company and its clients. ERP system is a popular form of ES. ERP systems are used to perform well-defined tasks and processes (Brocke et al. 2018). An ERP system usually includes monitoring, control, and management of these tasks, processes, and other business-related activities (Gurău 2020).

ERP systems function as coordinators for work performed by both people and machines. They also can have multiple integrations to other systems. (Brocke et al. 2018) An ERP

system also aims to improve collaboration between departments of an organization and is a significant factor when it comes to automation and efficiency improvement initiatives (Gurău 2020). Essentially, an ES functions as the backbone of an organization's operations (Brocke et al. 2018).

Digitalization is also not something that happens once, but it is a continuous process. For example, Industry 4.0 is a concept that is described to digitalize organizations further. Maximiliane and Uwe (2018) introduce the vision for Industry 4.0 to include even more comprehensive digitalization, where everything from production processes to downstream product services is linked through digitalized solutions. Industry 4.0 brings even more focus to the usage of big data, cloud computing, and AI in manufacturing and service processes (Brahma et al. 2021). Overall, many organizations place great emphasis on ramping up the usage of machines, robots, and AI to improve efficiency (Holford 2019).

2.2 Knowledge Work Productivity in the Digital Era

Standardization, automation, digital skills, changing job descriptions, and new digital systems have been so far reviewed more from the organizational level. The other side of the coin is also to understand the human side drivers when it comes to improving work and its productivity. Even though the usage of IT systems is often driven in organizations by cost-efficiency, it can also be seen as a valuable tool for employees when it comes to tasks that require complex decision-making (Braun et al. 2016). Moreover, technological advances have shifted the work in a direction where humans and technology work in a legitimate partnership, allowing respective strengths to be utilized optimally (Holford 2019).

When it comes to new digital process innovations, one of the main goals is to help humans perform their tasks in faster and more innovative ways (Ahmad & Van Looy 2020). Human-IT alignment is seen as an important factor for organizations. Human-IT alignment describes how well employees can apply data and IT tools within their daily work tasks. (Anon 2021) New IT-related tools like AI, the Internet of Things, and blockchain are seen to enhance organizations' business process capabilities significantly but also present a variety of new challenges with their adoption (Ahmad & Van Looy 2020). In many cases, humans need to work in collaboration with AI in order to leverage the advantages of both humans and AI (Rinta-Kahila et al. 2021). In many organizations, a great emphasis needs to be put on understanding the new complementarities between technology and human skills (Annunziata & Bourgeois 2018). On the human resource

level, introducing new technological innovations requires careful consideration of needed training, development, and cultural acceptance needs (Ahmad & Van Looy 2020).

It is found that digitalization itself is a crucial factor for knowledge workers' efficiency. For example, efficient and fast knowledge flows have been seen to enhance knowledge work efficiency by removing unnecessary delays when performing tasks. (Vuori et al. 2019) Palvalin (2018) found in their research that well-being at work, individual work practices, and the social environment significantly impact knowledge work productivity. From an organizational point of view, having an understanding of why tasks and processes are performed in a certain way by teams and individuals is an important basis for the continuous improvement of working methods (Kokina & Blanchette 2019).

There are multiple ways to address these performance factors in organizations. To improve efficiency and productivity, it is vital to make sure that knowledge workers are satisfied with their working circumstances (Palvalin 2018). Both Palvalin (2018) and Vuori et al. (2019) note that important drivers in knowledge work efficiency are autonomous working practices and self-management. Promoting asynchronous working is seen to be an essential factor as well (Vuori et al. 2019). Other highlighted focus points are management skills and the organization's work practices (Palvalin 2018).

2.3 Automation of Digital Work

Automation of business processes and tasks is a key topic for digital organizations. Typical requirements for automatable tasks are repetitiveness, high volume, structured data, rule-based nature, and involvement of multiple IT systems (Kokina & Blanchette 2019). Finding repetitive and high-volume patterns from the data set could lead to the discovery of tasks that have automation potential. As the research topic is highly related to automation requirements, task automation is an important field where the results from frequent pattern analysis can be utilized.

Madakam et al. (2019) define automation as a technique of making a process or a system operate automatically. The manufacturing sector uses automation and robotics broadly these days (Madakam et al. 2019). Industrialization has already showcased how new technologies can revolutionize working practices by replacing traditional jobs with the use of machines. Similar extensive global-level consequences could be underway in the 21st century when computers substitute more conventional jobs. (Braun et al. 2016)

Robotic process automation (RPA) is one of the main trends in automating business processes with the use of software robots and AI (Madakam et al. 2019). Generally speaking, RPA is a set of software tools and platforms used to automate rule-based

digital processes (Lacity & Wilcocks 2016). RPA solutions usually target repetitive and clerical tasks performed as a part of office work with the aim of automating tasks previously performed by the human workforce (Willcocks 2020). Hence, the software can be thought of as a robot as it intends to replace human resources (Madakam et al. 2019). Furthermore, RPA can be deployed as software bots, with each having its own computer station. In essence, these digital workers with their own computer stations can be thought to mimic a human worker. (Kokina & Blanchette 2019) Lacity and Wilcocks (2016) describe RPA as software that can handle repetitive tasks on behalf of humans so that humans can focus more on unstructured and more interesting tasks. Furthermore, the RPA is seen to develop towards intelligent automation, where more complex tasks with unstructured data can be automated with the help of AI (Kokina & Blanchette 2019; Madakam et al. 2019).

RPA bots can interact with multiple business systems, perform routine tasks with binary decisions, and work autonomously (Kokina & Blanchette 2019). What makes RPA an intriguing option for organizations is that the bots can work around the clock, and more instances of the same bot are easy to scale up if additional processing capacity is needed (Lacity & Wilcocks 2016; Kokina & Blanchette 2019). Hence, some of the main drivers for automation in companies include financial benefits, improvements in speed and quality, 24-hour service availability, and increased regulatory compliance (Lacity & Wilcocks 2016). RPA will help make processes more efficient and, therefore, more profitable (Braun et al. 2016).

Automation can also end up creating better jobs for employees (Annunziata & Bourgeois 2018). Notably, RPA implementation is seen to lead to human-robot teams where robots work alongside humans and amplify and augment distinctive human strengths (Lacity & Wilcocks 2016). Automation can make daily tasks easier and faster and improve the perceived experience for the workers. Furthermore, automation reduces the monotonous, repetitive tasks that knowledge workers need to perform. (Madakam et al. 2019) Using RPA solutions free time for more creative work. Eliminating repetitive tasks from knowledge work is also seen to decrease task switching, which itself can be a significant productivity boost. (Kersten & Murphy 2015) Considering these factors, it is argued that RPA leads to higher employee satisfaction (Lacity & Wilcocks 2016).

The automation will shape the knowledge work and performed tasks. Humans still excel at performing unstructured parts of the work, whereas the structured parts of the process increasingly shift to be performed by robots (Lacity & Wilcocks 2016). Sampson (2021) argues that most of the jobs will be redesigned rather than completely replaced as a result of automation. New skills are required from workers as the content of knowledge

work changes. Hence, automation might result in skill mismatches and a need for additional training to keep up with the new tasks and responsibilities. (Annunziata & Bourgeois 2018) New responsibilities might involve more creative tasks, customer and human-facing interactions, and problem-solving-oriented tasks (Braun et al. 2016; Lacity & Wilcocks 2016; Sampson 2021). The automation and changing job descriptions of subordinates will also have effects on the jobs of managers and management practices in enterprises (Brocke et al. 2018).

It is important to note that automation happens on a task level rather than on a job level. A task can be defined as a standard episode performed by knowledge workers as part of their jobs. Furthermore, a task can be seen as a discrete piece of work that needs to be completed within defined time limits. (Sampson 2021) Tasks that are in the scope of RPA solutions are usually by nature repetitive, high volume, rule-based, performed digitally in multiple IT systems, and involve structured data (Kokina & Blanchette 2019).

Some basic examples of tasks that have been in the scope of RPA are copying data into a form, generating utility bills, paying health care insurance claims, accessing an ES to write to databases, and keeping employee records up to date (Lacity & Wilcocks 2016; Kokina & Blanchette 2019). When discovering a task from an automation perspective, it is important to focus on the information that is being consumed, created, or changed during the task in order to form the basis for understanding the task's context and the relationship between digital content that is being interacted with (Kersten & Murphy 2015). In summary, to discover and analyze tasks with RPA potential, it seems to be important to be able to capture the volume of the task, involved IT systems, understand task rules in terms of taken paths, and understand the involved data.

2.4 Improving Business Processes

From an RPA perspective, it is important to understand how business processes are executed and where is automation potential (El-Gharib & Amyot 2023). These days, many organizations also have a business process-centric mindset, and a key focus point for them is to monitor, control, and analyze business processes (van der Aalst et al. 2016; Weilkiens et al. 2016). As tasks can often relate to a business process, it is important to understand key concepts from the business process research field. This chapter focuses on reviewing topics related to improving business processes within the context of digital tools.

BPM is a discipline that focuses on business process automation and optimization (Ballard et al. 2006). Workflow management is about the automation of business processes

by focusing on a simple series of tasks called workflows (Wang et al. 2005; IBM). Process mining offers tools for modeling the actual state of a process and provides transparency into its execution (van der Aalst 2023). Task mining is a technology that provides insights into how tasks are performed by monitoring user activities and collecting interaction data from desktops (Dilmegani 2023). Figure 1 visualizes how these key topics relate to improving business processes.

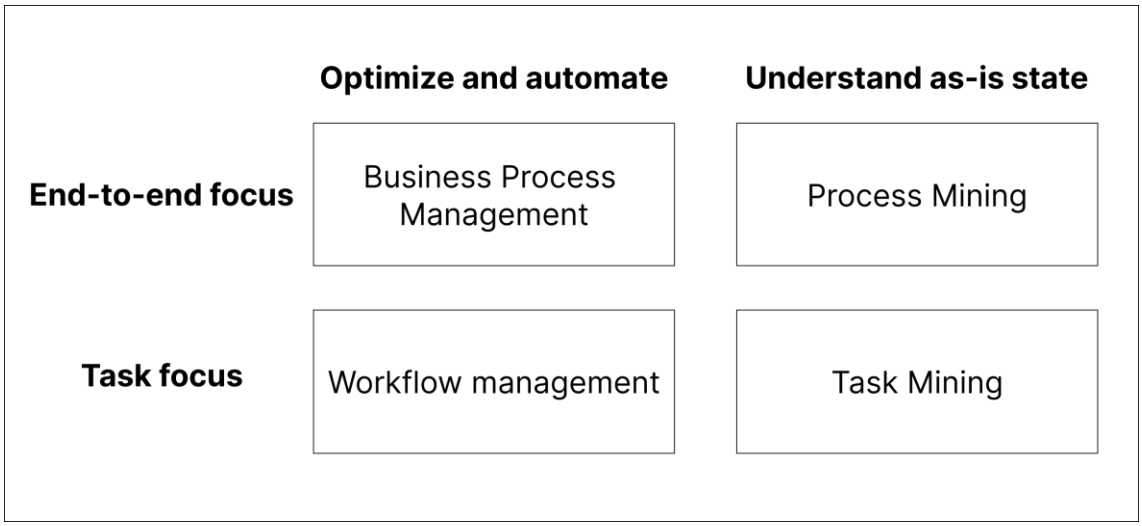

Figure 1. *Key research areas in the field of improving business processes.*

To maximize business profitability and gain competitive advantage, managing and optimizing business performance is a critical factor for organizations (Ballard et al. 2006). Business processes are key factors in realizing business goals (Weske 2012). Business processes can be defined as a set of defined activities performed as a response to an event (Ballard et al. 2006). A business process is usually performed in coordination with organizational and technical environments (Weske 2012). Having a process-centric mindset is common in most of today's organizations (van der Aalst et al. 2016). Business processes are key factors when it comes to optimizing business and how people and information systems play together (Weske 2012).

As business processes are key assets for organizations, they should be explicitly managed by measuring, monitoring, controlling, and analyzing them (Weilkiens et al. 2016). BPM is a term used to describe the discipline of managing business processes (Ballard et al. 2006). In BPM, organizations are viewed as a system of interlinked processes (Benner & Tushman 2003). BPM incorporates a wide variety of concepts like mapping, improving, designing, administrating, and analyzing business processes (Benner & Tushman 2003; Weske 2012). BPM aims to provide methods to build a foundation for mastering current and future challenges in management (vom Brocke et al. 2014).

BPM aims to optimize and automate business processes (Ballard et al. 2006). Some of the most common areas of BPM are the automation of production processes or administrative procedures (Bazan & Estevez 2022). A common outcome of BPM initiatives is the development of business applications to support the execution logic of the underlying business process (Wang & Wang 2006). Other notable results are, for example, identifying and eliminating redundancies and bottlenecks, increasing portability and decreasing costs by use of industry standards, and minimizing manual tasks (Ballard et al. 2006).

Being able to monitor and track the process execution throughout its value chain is important to support BPM (Ballard et al. 2006). Business processes consist of a number of activities (Weske 2012). BPM aims to collect data relating to these activities, like the start and completion times of activities and the input and output of the activities (Ballard et al. 2006). Being able to track and discover tasks performed as part of a business process can help monitor business processes and their activities.

Workflow management (WFM) is a discipline closely related to BPM. BPM has its roots in workflow management systems (van der Aalst et al. 2016). BPM is seen to be an extension of workflow management (van der Aalst 2013a). A workflow can be defined as a unit of work that is performed repeatedly in an organization of work (Schäl 1998). Workflows are by nature simple and repeatable, consisting of a simple series of tasks, while business processes are considered to be more complex and include multiple workflows, information systems, data points, and people (IBM). Workflows are usually described with definition languages, like Petri-net models, in order to support the design and definition of the flows (Schäl 1998).

Workflow management is about digitalizing and automating processes and managing the outcomes (SAP Insights). The aim of WFM is to automate business processes in order to improve speed and efficiency (Wang et al. 2005). The focus of WFM is on the automation of business processes, whereas BPM has a broader view into the topic of improving processes not only via automation but also focusing on process analysis, operations management, and the organization of work (van der Aalst et al. 2016). WFM can support BPM by offering support for modeling, simulating, automating, and monitoring processes (Wang et al. 2005).

Going beyond BPM and Workflow management, process mining is a rapidly growing approach for analyzing business processes with event logs (Vulpe et al. 2022). Where BPM focuses more on modeling processes in a To-Be format, process mining aims to describe As-Is processes and reality in a data-driven manner (Reinkemeyer 2020). Process mining combines data mining with business process modeling and analysis (van

der Aalst et al. 2012). Process mining includes multiple approaches to analyzing processes: process model discovery, conformance checking, performance analysis, predictive analysis, and automatic activities that address performance and compliance problems (van der Aalst 2023). The first commercial process mining tools date back to about 15 years ago. These days, process mining has been adopted for more extensive commercial use, as there are over 40 commercial process mining tools, and process mining technology is being utilized in thousands of organizations all over the globe. (van der Aalst & Carmona 2022)

Process mining helps with the identification of inefficiencies and effort drivers by providing a holistic picture of as-is processes (Reinkemeyer 2020). The focus of process mining is usually on operational processes consisting of repeating activities leading to the delivery of products or services (van der Aalst & Carmona 2022). Process mining is utilized in a wide range of industries like finance, logistics, production, customer service, and healthcare (van der Aalst 2023). Process mining aims to improve business efficiency by providing insights into bottlenecks, errors, and streamlining opportunities (Vulpe et al. 2022). Hence, process mining can be utilized as a supporting tool for digital transformations (Reinkemeyer 2020).

Process mining is based on event logs extracted from digital traces (Reinkemeyer 2020). These digital footprints are recorded when people, organizations, and devices interact with digital systems (van der Aalst 2013b). An event log can be defined as a collection of events that take place during the execution of a business process (Reinkemeyer 2020). Event data is extracted from IT-systems that are part of the business process execution. Usually, the event logs are scattered across multiple systems and databases, where they need to be converted into a unified format that supports process mining. (van der Aalst & Carmona 2022) Reference to a case ID, an activity, and a timestamp are the minimum requirements for an event log entry (Reinkemeyer 2020). Additionally, the event log can include extra data attributes like information about the resource, location, and cost (van der Aalst & Carmona 2022). A process model can be used to indicate the order of the activities for each case to describe the flow of the process (van der Aalst 2013b).

One of the primary use cases of process mining is process discovery. In process discovery, a process model is learned from the event logs (van der Aalst 2013b). The event data is turned into a process model by utilizing process discovery techniques (van der Aalst 2023). On many occasions, people think of processes as simple flows from point A to B, like the one visualized on the left side of Figure 2. However, the reality of processes is usually more complex, leading to visualizations outlined on the right side of

Figure 2. (Reinkemeyer 2020). Discovered process models support both the visualization of the most frequent mainstream flow and the more infrequent flows (van der Aalst 2023). Hence, process mining visualizations offer the option to drill down gradually from the mainstream variants to more infrequent and complex variants until all the observed variants are visualized (Reinkemeyer 2020).

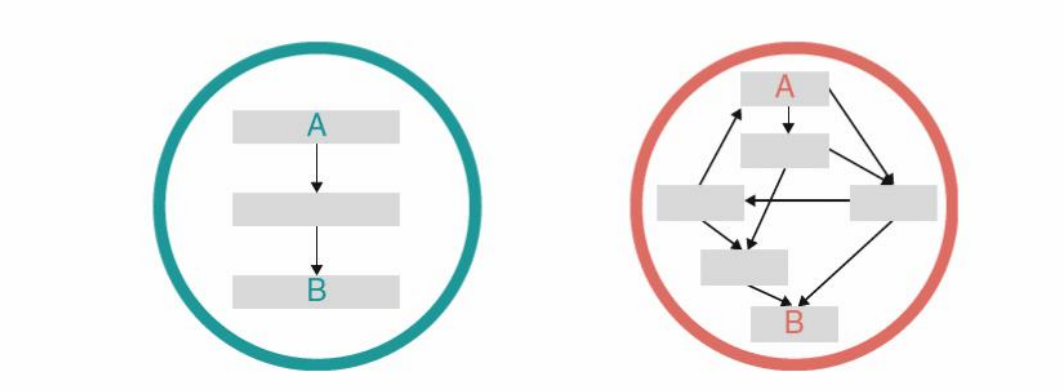

Figure 2. *Displaying different process variants (Reinkemeyer 2020)*

Process discovery techniques convert event logs to process models without using any prior information (van der Aalst et al. 2012). Moreover, the goal of the process models is to describe the actual state of the process and provide transparency into the execution (van der Aalst 2023). Describing a process as a control flow is at the core of any process modeling task (van der Aalst et al. 2012). Control flows can use different notations to visualize the process. Some examples of these notations are Directly-Follows Graphs (DFGs), Business Process Model and Notation (BPMN), and Petri nets. Process mining tools commonly use the DFG to visualize the process when loading an event log. Figure 3 demonstrates a DFG visualization. The DFG consists of nodes connected by directional arcs or arrows that represent directly-follow relationships between nodes. The nodes are activities of the process. (van der Aalst & Carmona 2022) Additionally, DFG visualizations can be enhanced by adding average or median times into the connecting arrows to represent the time between consecutive activities, as understanding the durations between activities can help identify process steps that take more time and pinpoint bottlenecks (Reinkemeyer 2020). The visualization also includes nodes to signal the start of the process and the end of the process (van der Aalst & Carmona 2022).

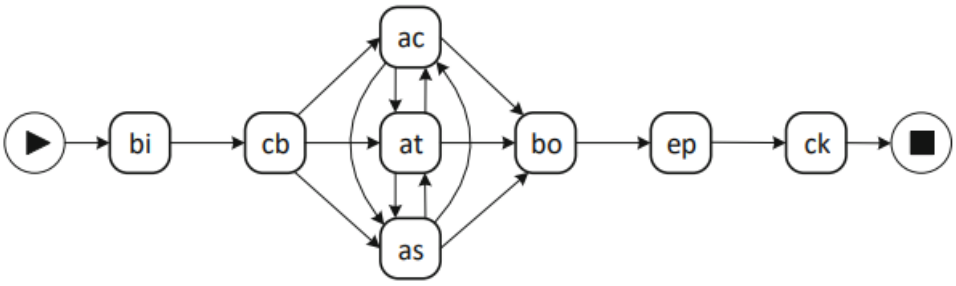

Figure 3. *Example of Directly-Follows Graph (van der Aalst & Carmona 2022)*

Process mining has realized only a fraction of its full potential in organizational settings (van der Aalst & Carmona 2022). At the same time, one of the continuous challenges in process mining is discovering how to provide meaningful insights from the event data (van der Aalst et al. 2012).

Task mining is another closely related industry term. Task mining is a technology that provides insights into how tasks are performed by monitoring user activities and collecting interaction data from desktops (Dilmegani 2023). The understanding gained from task mining can help organizations to automate and improve processes (Raiker 2020). The goal of task mining is to record what employees are actually doing during particular tasks and discover the most common actions through analysis of the interaction data (Dilmegani 2023). Task mining has similarities to process mining, but it focuses on user interaction data rather than logs (Raiker 2020).

Commonly, task mining solutions are integrated into process mining technologies to ground the task-level findings to the process scope. Task mining has not gained much attention as of yet but is expected to gain more popularity in the upcoming years. (Dilmegani 2023). Task mining solutions are seen to be in the early stages of enterprise adoption. However, the task mining markets are expected to experience rapid growth, with annual growth rate projections ranging from 75% to 85% (Everest Group 2022). Overall, task mining seems to be only an emerging industry term, and there has been a minimal amount of scientific research related to the topic.

3. PATTERN DISCOVERY

The research implementation will focus on performing data analysis. More specifically, the focus is on analyzing the sequence-based data to find interesting patterns. Sequence data mining is a field that focuses on mining patterns from sequences (Dong & Pei 2007). Pattern discovery focuses on finding patterns from a data set that satisfy defined criteria (Hand et al. 2001). This chapter focuses on key data analysis areas and relevant sequence data mining and pattern discovery concepts.

3.1 Data Format

Understanding more about similar data formats can help, for example, with data preprocessing techniques or with identifying suitable methods to analyze the data. They can also help with conceptualizing the topic and provide relevant terminology. Understanding key data features and attributes offers the necessary base knowledge for the analysis.

Logs are append-only sequences of some sort of records in chronological order (Kreps 2014). IT-system events and messages can be captured as log data (Gillespie 2020). The primary function of a log is to record what happened and when (Kreps 2014), hence providing details about a specific event (Gillespie 2020). Web logs contain sequences of user-page pairs that can include additional information, like time spent on a page (Dong & Pei 2007). A system trace is a very similar concept to weblogs, as they contain sequences of user's operations in one or more systems (Dong & Pei 2007). Another way to approach the collected data is to conceptualize it as a sequence. An especially interesting category of sequences is event sequences (Dong & Pei 2007). Event sequences are usually represented as sequences of event and occurrence time pairs. Event sequence can be used to describe how underlying actors behave and hence give insights from the collected data. Examples of event sequences are records of financial transactions, telecommunication alarm logs, and customer purchases. (Hand et al. 2001). Hand et al. (2001) also introduce the term time series, which seems to have features similar to event sequence. Time series entry can be represented using two variables: measurement timestamp and measurement value at that time (Hand et al. 2001).

Feature selection is a task that includes selecting the most useful features from all possible candidate features (Dong & Pei 2007). Han et al. (2012) define a feature to be a data field that is used to describe a characteristic or feature of a data object. The terms attribute, dimension, feature, and variable are usually used interchangeably and mean

the same thing in the literature. Usual feature types are nominal, binary, and numeric. Nominal attributes are names of things and usually represent some category, code, or state. A binary feature is a boolean value, meaning it has two categorical values: 1 and 0, or alternatively, true and false. A numeric feature is a quantitative attribute that measures quantity as an integer or real number value. With numeric features, normalization techniques can be utilized to scale data points to a smaller range, like 0.0 to 1.0. (Han et al. 2012)

3.2 Data Mining

As a result of the explosive growth of data and the imperative to convert it into valuable information, the topic of data mining has garnered significant attention (Han et al. 2012). The aim of data mining is to find relationships and summarize the data in novel ways to give understandable and useful insights (Hand et al. 2001). Data mining helps us to find interesting patterns and knowledge from large amounts of data. Data mining is a multidisciplinary field that is connected to statistics, machine learning, pattern recognition, artificial intelligence, high-performance computing, and data visualization. (Han et al. 2012)

A broader context to data mining is a process called knowledge discovery from data (KDD). The KDD process usually contains data mining as one of its steps (Hand et al. 2001). Han et al. (2012) present the KDD process to include seven steps:

1. Data cleaning
2. Data integration
3. Data selection
4. Data transformation
5. Data mining
6. Pattern evaluation
7. Knowledge presentation

Getting more familiar with the data before processing it is an integral part of the data mining process. Understanding data values and attributes is essential when dealing with often noisy and large real-world data sets. (Han et al. 2012) Graphical displays, like bar and pie charts, can be utilized to describe some fundamental statistical trends from the raw data (Han et al. 2012). Visual methods are valuable data exploration methods because they leverage humans' powerful eye-brain connections to detect structures.

Hence, visual and graphical methods can help humans to utilize their pattern-processing abilities better. (Hand et al. 2001)

Han et al. (2012) introduce reduction as one option to improve the performance and efficiency of data mining algorithms. Hand et al. (2001) also state that the scalability of the selected algorithm is an important issue. Sampling is an option to reduce the amount of computation needed to process the data. Sampling allows to pick only part of the data to be analyzed. If the selected sample is large enough, the frequency of a pattern should be approximately the same in a large sample as it is in the whole data set. (Hand et al. 2001) In a similar fashion, reduction techniques aim to obtain a reduced representation of the data, resulting in a data set with a much smaller volume but one that should give almost the same analytical results as the original data set would (Han et al. 2012).

With data mining projects, focusing on the insightfulness and usability of the results is a key factor for gaining valuable outcomes. Data mining algorithms can produce an extensive set of patterns. However, typically, most of these patterns are not interesting, and only a tiny fraction of the results are actually valuable for the given use case. There are sets of statistical objective measures that could be used to evaluate the generated patterns, but usually, they are insufficient until combined with subjective measures by the users of the results. (Han et al. 2012)

Lastly, result presentation is an important step of the data mining process. The presentation and visualization of data mining results help humans understand and use the results (Han et al. 2012). Visualization of results is important as it allows humans to understand and process the patterns (Hand et al. 2001). A key visualization tool to model process flows is control flow visualizations (van der Aalst et al. 2012).

3.3 Frequent Sequential Patterns

Mining patterns and knowledge from large sequence data sets is called sequence data mining. Web logs and customer purchase histories are examples of sequence data sets that capture human behavior. (Dong & Pei 2007) A subsequence that occurs often in sequence data is called a frequent sequential pattern (Han et al. 2012). Sequence data mining is a field that provides tools and techniques for finding knowledge from large sequence data sets (Dong & Pei 2007).

A pattern can be defined to be a local feature of the data that groups together, e.g., a set of records or variables (Hand et al. 2001). A sequence pattern is constructed from single-position patterns $\{c_1, c_2, \ldots, c_k\}$, with the definition of their positional distance (Dong & Pei 2007). In pattern discovery, the pattern's frequency in the data set is a key property

of the pattern. The frequency of a pattern can be determined by calculating the relative number of observations of the pattern in the whole data set. A pattern discovery task includes finding all the patterns from the data set that satisfy certain conditions, like a frequency threshold. A key aspect of pattern discovery is to find the proper balance between the expressivity of the patterns, comprehensibility, and computational complexity. Pattern discovery tasks also can contain consideration for the informativeness, novelty, and understandability of the pattern. (Hand et al. 2001)

Pattern discovery also splits into different sub-fields depending on the type of pattern. One commonly used pattern term is episode. An episode is simply a set of events that has restrictions for the order in which the event occurs (Mannila et al. 1997; Tatti 2014). In essence, an episode is a simple class of patterns that contain partially ordered events that occur together (Hand et al. 2001). Episode discovery is a core method in the field of time-series analysis (Casas-Garriga 2003). Multiple different algorithms have been developed for episode discovery (Achar et al. 2012). Another term used often in literature is itemset. A set that contains k items is called k-itemset. For example, the set $\{c_1, c_2\}$ has two items and is hence called a 2-itemset. Unlike episodes, the order of the events is not specified in an itemset. Also, for itemsets, the frequency can be calculated by counting how many of the transactions contain the itemset. (Han et al. 2012) Itemsets are simple patterns that can tell which variables occur often together (Hand et al. 2001). This makes frequent itemset discovery one of the most fundamental forms of frequent pattern discovery. Other typical pattern types are, for example, frequent subsequences and frequent substructures. (Han et al. 2012)

3.4 Pattern Discovery Algorithms

A data mining algorithm is a procedure that takes data as an input and produces models or patterns as an output. An algorithm should also consist of a finite set of rules, terminate after a finite number of steps, and always produce output after the steps. (Hand et al. 2001). Association rule mining is a popular and deeply studied orientation in the fields of data mining and knowledge discovery (Wang et al. 2009; Othman & Eljadi 2011). Association rules are used to describe correlations between different factors and to find useful relationships. A classic example of association rule mining is market basket analysis, where the aim is to find different associations between items that customers are purchasing often together. (Wang et al. 2009)

The association rule can be described as a rule $X \rightarrow Y$, where X and Y are a set of items and $X \subset I$, $Y \subset I$, $X \cap Y = \varnothing$. In this case, X is referenced as the antecedent of the association rule, and Y is referenced as the consequent of the association rule. (Othman & Eljadi

2011) For transaction set D, support S for association rule X→Y can be calculated by how many percentages of the transactions in set D contain both X and Y. Confidence C can be calculated by defining the percentage of cases where both X and Y are observed together compared to all the cases where the X is present. (Othman & Eljadi 2011; Zhan et al. 2019) In essence, confidence describes how related the variables are, and support describes how common the pattern is in the whole data set. If the association rule exceeds the minimum support and confidence thresholds, the association rule is found to be interesting (Zhan et al. 2019). Typically, to mine association rules, the support and confidence values need to be computed for all the rules and then compared to threshold values to come up with the set of rules that are the point of interest (Othman & Eljadi 2011). The Apriori algorithm was one of the earliest association rule approaches for finding frequent itemset, and after that, mining association rules have been broadly studied topic (Zhan et al. 2019). Frequent pattern (FP) growth and Apriori are some of the most traditional and significant algorithms in the field of association rule mining (Dongnan & Zhaopeng 2021).

The database is scanned once in the FP-growth approach to find frequent header list L1. The records are then represented in descending order based on their support to L1. From this representation, the frequent transactions are used to build the FP-tree. Patterns can then be mined recursively from the tree structure. (Wang et al. 2009) The Apriori algorithm relies on an iterative approach to find frequent patterns. The Apriori algorithm performs a level-wise search where frequent k-itemsets are used to discover frequent (k+1)-itemsets. (Han et al. 2012)

Besides association rule mining, for example, hidden Markov models and Kernel-based methods are alternative approaches to discover interesting patterns from sequence data (Dubey & Mishra 2011). Hidden Markov models have been a popular approach to classifying sequential patterns (Kumar et al. 2012). The Gaussian hidden Markov model is a well-known unsupervised machine learning method to handle time-series data. This approach is based on the assumption that at any point in time, there is a state that can be estimated from the data. (Uto et al. 2020) Kernel-based methods are applied to find both static and sequential patterns. When it comes to finding sequential patterns, the main challenge with kernels is to design suitable kernels that can handle patterns of varying lengths. To find patterns, the kernel method first performs nonlinear transformation from the input data to higher dimensional feature space. Then, an optimal linear solution is discovered for the kernel feature space to find the patterns. (Kumar et al. 2012)

Different algorithm options from the field were assessed to find a suitable solution for data analysis. The apriori-based approach was selected as the best match for the research's purpose. The simplicity of the Apriori method, as well as its popularity, were the main factors for choosing the Apriori-based approach. Having a relatively easy-to-implement algorithm gives more room to examine the value drivers from the business perspective rather than only from a data science perspective.

3.5 Apriori

Apriori algorithm is an algorithm that aims to find frequent itemsets (Han et al. 2012). Apriori-based methods are one of the earliest and most influential algorithms introduced to the sequential pattern mining problem (Kumar et al. 2012). The main principle of the Apriori algorithm is to discover frequent patterns with k-items and to expand the search to itemsets with k+1 items (Han et al. 2012). The Apriori algorithm is based on the assumption that if the sequence S is frequent in the data, all the subsets of S are also frequent (Hand et al. 2001). This proposition can hence be reversed the other way around; if sequence S is not frequent, all the super sequences of S are also infrequent (Kumar et al. 2012). This means that we do not need to define the frequency of sequence S if none of the subsets of S are frequent. Therefore, the Apriori algorithm can start by finding all frequent sequences with one item and work upwards from there. (Hand et al. 2001) Apriori property is defined as follows: All nonempty subsets of a frequent itemset must also be frequent (Han et al. 2012). The Apriori algorithm starts by scanning itemsets with the size of one and filters out non-frequent sets. Then, 2-itemsets containing items that passed the previous step are scanned and sets that fulfill the support threshold are again preserved. This process can be continued and repeated recursively for larger sets. Apriori algorithms help reduce the size of the scanned candidate sets as many of the possible combinations are eliminated early on in the process. (Han et al. 2012)

Dongnan and Zhaopeng (2021) mention that the selection of minimum support and confidence thresholds can be a drawback to the Apriori algorithm, as setting the thresholds requires subjective judgment. Also, the Apriori algorithm is not the most performant option when it comes to needed computational resources. Apriori can still generate a high number of candidates when dealing with a large data pool. This will lead to increased computational complexity and reduced processing capacity and computational speed. (Dongnan & Zhaopeng 2021)

4. RESEARCH DESIGN

This chapter explains choices made in the research design and aims to link them to research theory. The main focus is to explore the relationship between the thesis's business-driven background and scientific implications.

4.1 Philosophy

The main goal of the thesis is to produce a new analytical model or other similar artifact to enhance the case company's analytical capabilities. The focus is, therefore, very much linked to gaining direct practical value. From a research point of view, it is imperative to recognize the heavy emphasis on practical findings. Investigating how this affects the research philosophy is essential.

The research philosophy describes the underlying beliefs and assumptions when carrying out the research. Recognizing and understanding the philosophical orientation is important for the research coherency as the philosophy is linked, for example, to research strategy and data collection. (Saunders et al. 2016) The practical focus of this research hints towards a philosophical approach called pragmatism. Pragmatism emphasizes that the relevancy of a concept in research should be measured by its applicability to action (Kelemen & Rumens 2008). In pragmatism, research, theories, and results are observed on what are their practical implications in a specified context (Saunders et al. 2016).

4.2 Action Research

Saunders et al. (2016) highlight that action research strategy is a common choice with pragmatism. Action research focuses on action and reflection (McNiff 2013). The action part can resemble an intervention where existing practices are reconstructed and transformed. Reflection and combining it with research promotes learning for participants. (Somekh 2005). As the researcher performs the action, it also involves self-reflection (McNiff 2013). It is easy to see why action research complements the pragmatic philosophy well, as pragmatic philosophy stresses action-centric research (Kelemen & Rumens 2008) and highlights the role of the researcher's values in driving the research (Saunders et al. 2016).

The central emphasis of action research is on real organizational issues. Action research focuses both on addressing the organizational issue and creating actionable theory. Ac-

tion research also highlights collaboration with others through face-to-face dialogue, conversation, and joint action. (Coghlan 2018) The action research strategy consists of four stages: identifying issues, planning action, taking action, and evaluating action. (Saunders et al. 2016). The research process is also iterative, where the previously mentioned steps are performed multiple times, constructing a self-repeating cycle (Coghlan 2018). Due to the iterative nature of the action research, it is possible that even the original research question changes (Saunders et al. 2016).

4.3 Action Research Implementation

Several factors influenced the choice of action research as the research strategy for this study. The research setting prioritizes creating practical value from an organizational perspective. The thesis topic originates from an organizational need that would have been addressed within the organization regardless of the existence of this thesis project. These factors align well with the pragmatic philosophy and action research, which commonly focuses on real organizational issues (Coghlan 2018). The literature review on digital work and tasks revealed numerous potential ways to apply the research results to organizational settings. As the primary avenue for value creation is not predetermined at the outset of the research, the action research strategy provides flexibility to explore various paths during the iterative and researcher-driven implementation phase. When considering pattern discovery, it is important to acknowledge that the evaluation of patterns often involves some degree of subjective judgment by the users interpreting the results (Han et al. 2012). The informativeness, novelty, and understandability of a pattern serve as crucial criteria for its assessment (Hand et al. 2001). Thus, from the data analysis perspective, action research allows the result validation to prioritize the input from human experts rather than solely relying on statistical measures.

On a high level, the goal of the research is to generate better business insights from the collected data. Therefore, examining the data is the initial step in the research implementation, with the aim of comprehending the starting point within the research's context. Data is gathered by deploying background software on the computers used by customer company employees. The backbone of the case company's data collection is to collect only business-related data and avoid collecting any personal data. The customer company can decide what applications it wants to track, and data is collected only from these allow-listed applications. Applications falling outside of the data collection are not tracked, and no data is sent from them to preserve the user's privacy. The software communicates with an external API server to get data collection configurations. The tracked applications typically include business-critical IT systems like communication,

ERP, and customer relationship management (CRM) systems. The customer companies typically operate in industries like accounting, insurance, and procurement.

The empirical part of the research focuses on data analysis and result representation. Hence, the next step in the research involves implementing the algorithm and establishing the analysis pipeline as part of the action research process. The algorithm implementation focuses on developing an Apriori-based algorithm to discover frequent patterns from the data. The rest of the analysis pipeline includes data preparations, result persistence, and result representation. Implementation of the algorithm and analysis pipeline also starts the iterative action research process described in Figure 4. The planning action and taking action phases in the research cycle then include planning and executing improvements to the developed analysis pipeline.

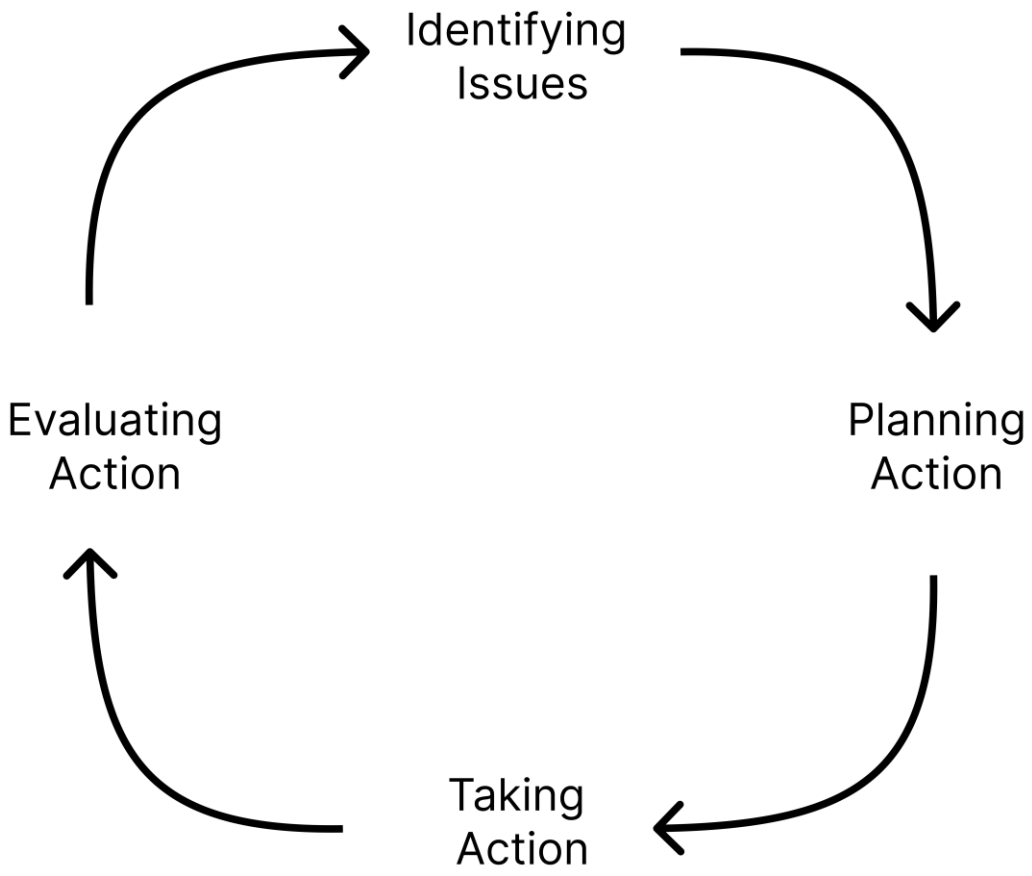

Figure 4. *Visualization of the action research cycle*

The next steps in the cycle are evaluating action and identifying issues. The result evaluation in each cycle is performed by discussing the results within the case organization. The two main persons involved in feedback discussions have multiple years of consultant experience in the fields of process mining, automation, and digital work. With their help,

the usefulness of the results is assessed. Based on the discussion, it is determined what worked and what did not in terms of value creation. The identifying issues step then includes a discussion on what to improve. After the discussion, the action research cycle resets back to the planning action state, where the improvements to identified issues are planned.

5. ANALYSIS AND RESULT EVALUATION

This chapter describes the action research process. The chapter starts with a description of the preparations and the initially developed analysis framework. Next, the research process is described in an iterative manner. Each iteration includes a discussion about the changes implemented for the analysis, result evaluation, and planning of improvements for the next iteration.

5.1 Data Preparations

The analysis phase starts by diving deeper into the existing data. It is important first to understand the data and then prepare the data accordingly before jumping to the algorithm implementation. Python was selected as the programming language for the project. Data preparation started by setting up the programming environment with Jupyter Notebooks. Jupyter Notebooks are designed to support scientific computing. The Notebooks suit well for interactive exploration as the code can be organized into multiple modifiable chunks that can produce rich outputs such as plots. (Kluyver et al. 2016)

The data set used in the thesis is collected from knowledge workers' computers. The data is collected by monitoring the operating system and application user interfaces. The data used in the thesis is pre-collected and stored by the case company in a database. One data record summarizes events performed during a window visit. A single record contains multiple fields. The fields contain information like the duration of the window visit, keyboard activities, and window categorization. From this perspective, the data set could be thought of as a system log as it includes sequences of user operations in one or more It-systems (Dong & Pei 2007).

The data also has timestamps that can be used to order the data in chronological order. Chronologically ordered sequences recording what happened and when can be conceptualized as log data (Kreps 2014). The most important data feature for finding frequent window sequence patterns is the window categorization information. This information describes the application type, application name, and active view or window name inside the application. For example, the application type could be a document, email application, or ERP system. The corresponding application names could be PDF, Outlook, and SAP. A window name could be, for example, Invoice, Inbox, or Create purchase order.

The case company has already implemented data separation to the collected data by introducing a term user session. A user session consists of sequential window visits performed by one user. During the data collection, the data is divided into user sessions by defining the end conditions for active sessions. When the end condition is reached, the currently active session is ended, and tracking of a new session is started. End conditions are computer shutdown, computer going to screensaver, or when the user is not interacting at all with the computer for an extended period of time. The user session-based data splitting can be used as the starting point when it comes to constructing the individual sequences from the data.

Feature selection will be an essential step for the data preparation. The feature selection will reflect how items in sequences are named. The initial design is to convert selected features to the categorization of the event. The sequence item name will essentially summarize all the selected features. The window categorization information will be at the core of the sequence item naming. The window categorization information includes information in different granularity levels, e.g., the view name inside the application is a more specific categorization than the more generic application type. Using and mixing different granularity levels is an interesting feature selection task.

Other features can also be added to the categorization. Adding information about user activities could be meaningful. For example, the visit's duration could be used to create categorization into short, medium, or long visits. Other simple activity categorization rules could also be generated based on user interactions. Typing, clicking, and clipboard activities could be used to generate human-readable categorizations about how the application was used.

Feature selection adds an interesting dimension to the item name generation, as item names can be constructed differently for different windows based on their use cases. Selecting the most useful features is at the core of feature selection (Dong & Pei 2007). Being careful with the feature selection is important, as adding too many features could add unnecessary complexity to the analysis and results. Hence, focusing on only the window categorization in the beginning should be a good starting point. Adding extra information could be considered if the need for more granular information is detected during the action research phase. The iterative approach enables the accumulation of more business understanding from the data over time, which can be reflected in the data preparation and data analysis.

Initially, data collected from one of the case company's customers is used. Potentially, data from other companies could be used later to validate if the results are applicable to

other customers as well. The data used in the analysis was decided to be restricted to a 2-month time period. The data was generated by tracking about 40 knowledge workers daily during the 2-month data collection period. The company where the data originates has allowed the data to be used in product development. However, it is also agreed that the data would be anonymized to a level where the company cannot be recognized, for example, based on the applications that are used. For this reason, some anonymization of the data is required so that it can be used in the thesis.

The focus of anonymization is to anonymize the application and view-level information. Application names were anonymized based on their use case. The new application names generalize the application without giving away exact application information. After anonymizing the application names, the time spent on each application was calculated. The results are visualized in Figure 5. The primary system for performing the knowledge work is the financial management application, which clearly has the most visits to its windows.

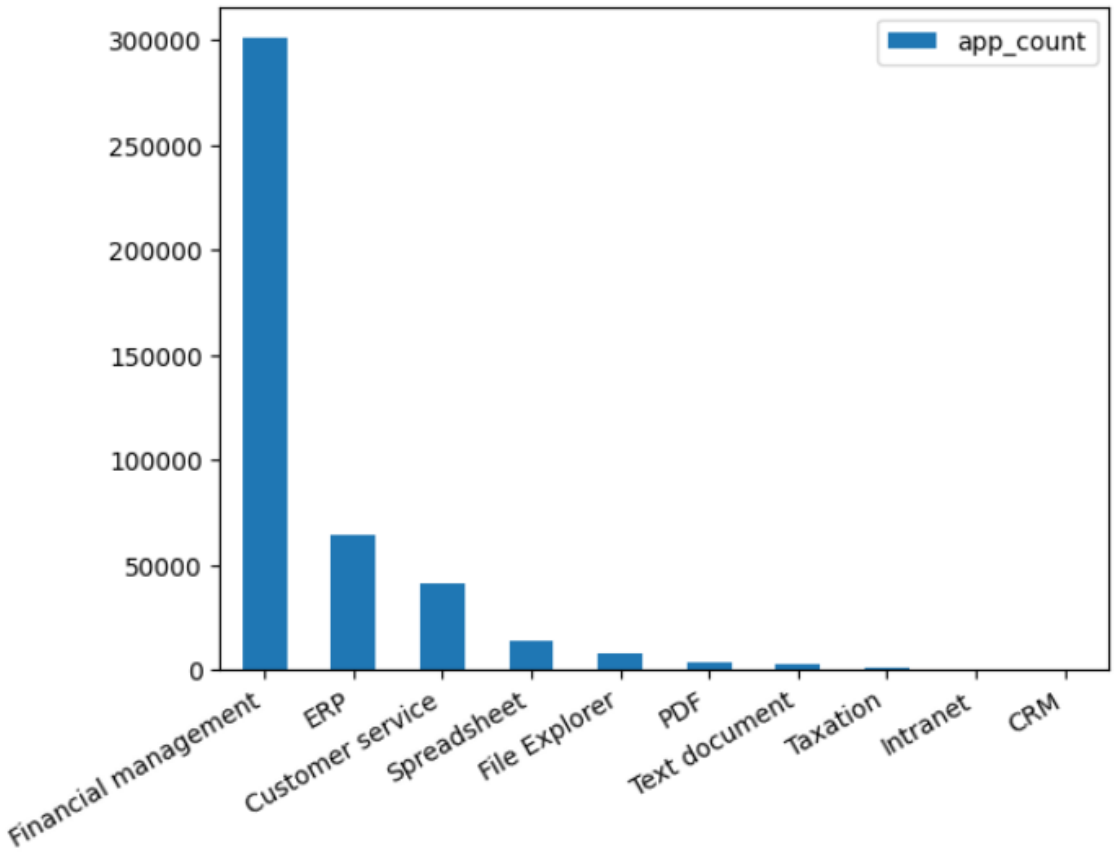

Figure 5. *Visits to application windows after the anonymization.*

The next step in the anonymization is to anonymize window names inside the applications, as they might have references to the original application names. As the input space for window names is much larger than applications, categorizing them manually would

have been very time-consuming and potentially still give out too much information about the application itself. Hence, for each application, windows were just given sequence numbers starting from number 1. The visit counts of the 15 most used windows are visualized in Figure 6. The total unique window amount is 253.

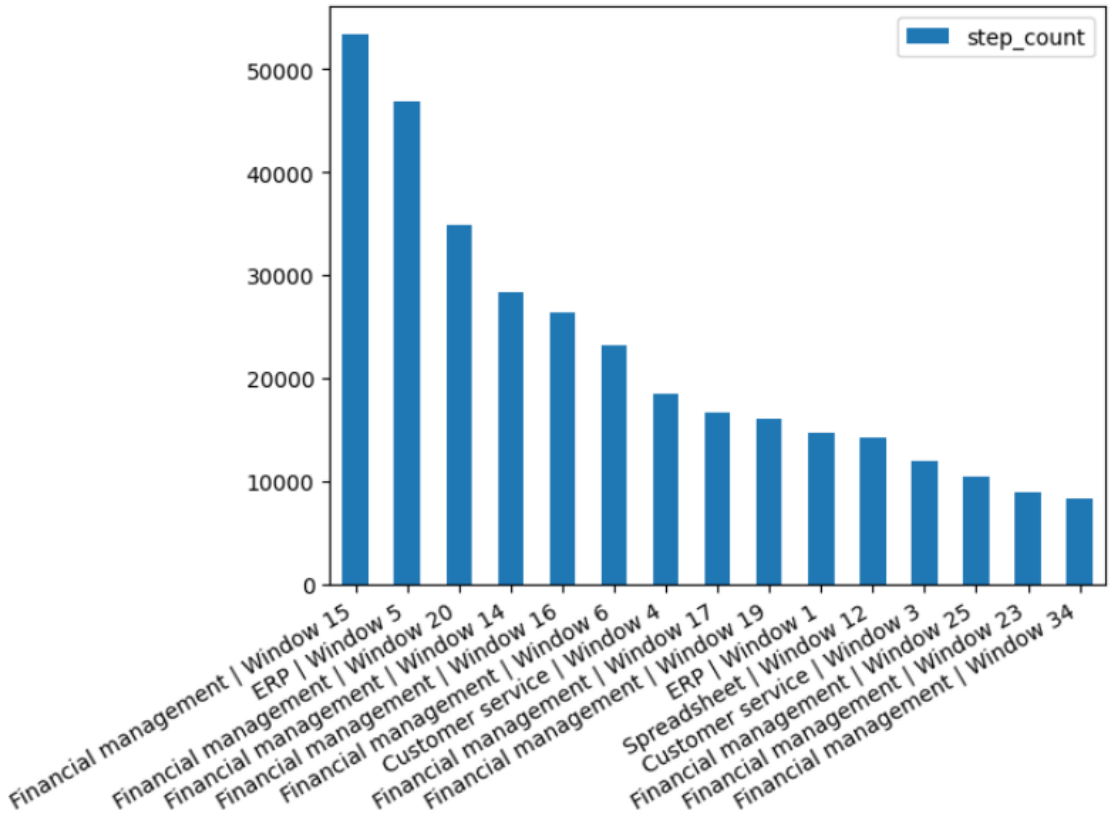

Figure 6. *Window visits count for 15 of the most used windows after anonymization.*

As described earlier, the data is pre-assigned to sequences based on the user session. There might still be a need for some adjustments to data sequences. Size- and duration-related metrics were calculated to discover how the data is distributed to sequences. Metrics for user session-based sequences are displayed in Table 1. The median sequence contains about 448 items and has a duration of about 3.4 hours. So, many of the sequences can include multiple different repeating sub-sequence patterns. Also, there are some shorter sequences that have only a couple of items. These sequences can be cleaned out of the data as short sequences should not contain meaningful patterns.

Metric	Value
Total sequences	1401
Items per sequence, median	447.76
Sequence duration, median	3.40 hours
Total sequences with less than four items	75

Table 1. *User session-based sequence metrics*

The case company collects data only from business-related application visits that are defined to be in the scope of data collection. This means that there might be some discontinuity inside the sequences. This appears in the data as unaccounted time between 2 consecutive window visits. For example, there might be a 1-minute difference between the end time of a window visit and the start time of the next window visit. Including these untracked visits to sequences could offer some interesting insights into the results. For example, if a quick visit to an untracked window frequently happens between 2 tracked business application windows, this could be a sign that a shadow business system is used as part of a business process that the management is not aware of. On the other hand, these unaccounted snippets of time could add extra noisiness to the data. There is no way of knowing if the untracked time span included only one untracked window or multiple windows. For example, longer time spans could consist of numerous untracked window visits, and the time spent in these untracked windows might not be related to the task that was being performed before. Hence, it might make sense to split the user session into multiple different sequences based on these discontinuation points where a more extended period is spent in untracked systems. The quicker visits to untracked systems could be included in the data as their own sequence items. The distribution of visit times to tracked windows is presented in Figure 7. As seen from visualizations, the majority of the window visits are relatively short.

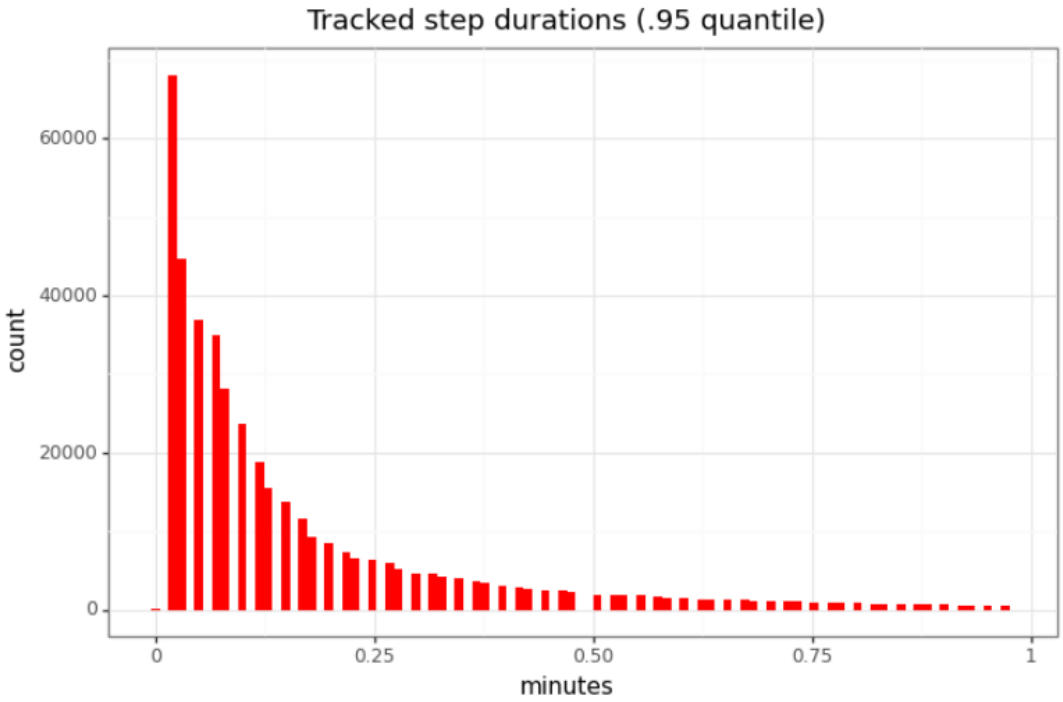

Figure 7. *Window visit duration distribution.*

The duration distribution of untracked system usage between 2 tracked systems is displayed in Figure 8. As can be seen from the figure, there are both longer and shorter visits to untracked systems. Based on these metrics, the initial threshold for untracked time removal was set to 2 minutes. If there is an untracked time span that is less than this threshold, a new untracked window item is added to the sequence. Suppose the time span is longer than the threshold; the current sequence is divided into two sequences. The first sequence ends with the last item before the untracked time span, and the second sequence starts after the untracked time span. This seems like a good generalization for now, as quicker visits have a higher chance of being part of the ongoing task. The longer visits, on the other hand, have a higher probability of causing noisiness and, hence, will be removed. Based on the results, this threshold could be fine-tuned during the action research cycles.

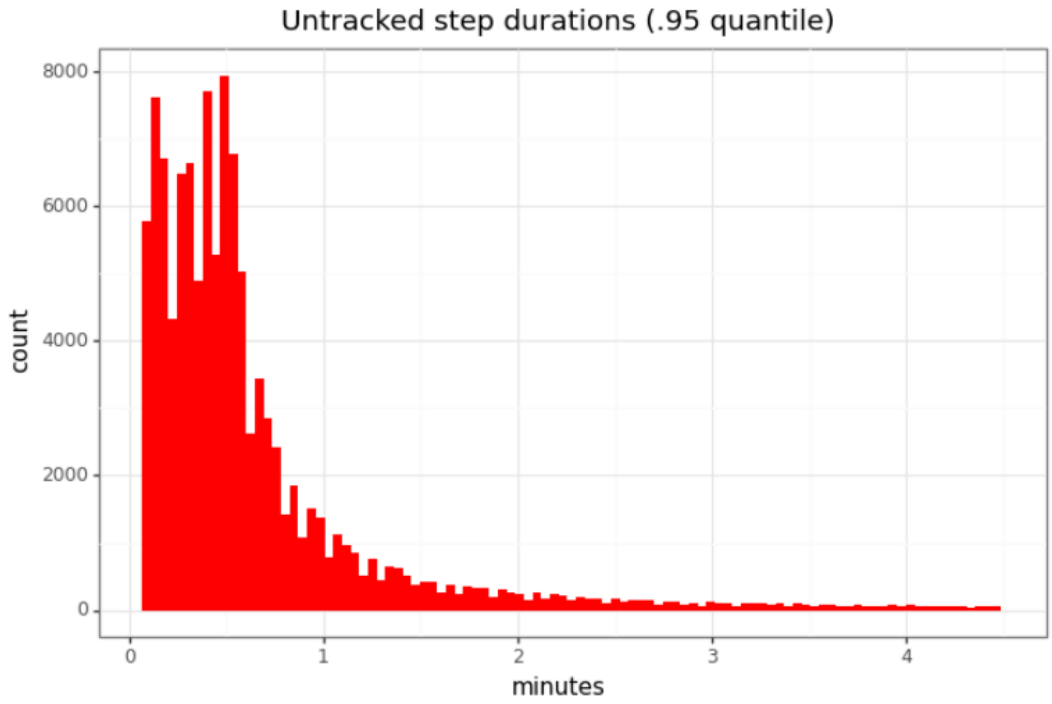

Figure 8. *Untracked time span durations distribution*

The data preparation included feature selection, anonymization, and sequence generation, as described in this chapter. The prepared data is saved to a new collection for later use. The next step is to implement the algorithm that is used to analyze the prepared data.

5.2 Cycle 1

5.2.1 Performing The Analysis

The analysis phase of the first action research cycle started by implementing the core pattern discovery algorithm. The algorithm was selected to be developed from the ground up. A sliding window approach and apriori-based thinking were the guiding principles in the development of the algorithm. The first iteration of the algorithm focuses on finding episodes where the order of the events matters.

A sliding window is used to generate subsequences from a parent sequence. For each window iteration, the window size is kept constant, and the window's start point is advanced forward one data point at a time. The window is advanced until the window's last data point is the last data point of the parent sequence. The sliding window's size starts at one, and after all the sequences are processed, the window size is increased by one. This procedure of scanning the data and increasing the size of the window by one is

repeated until there are no sub-sequences found from the data that would satisfy the support count criteria.

The algorithm implements support count checks for the found episodes. The implemented parameters that control processing are pattern_min_length and min_support_count. Pattern minimum length describes the minimum size for an interesting episode. The minimum support count describes the minimum requirement for how often an episode needs to appear in the whole data set to be considered interesting.

Lastly, some unit tests were put together to test that the algorithm produces the expected results. The main weakness of the algorithm is that it is almost a brute-force algorithm. The same data point will be processed multiple times for each sliding window size. For E.g., with a sliding window of size 4, one data point can be part of 4 result windows as it can be in the 1st, 2nd, 3rd, and 4th position of a window.

After implementing the data pattern discovery algorithm, the end-to-end data processing pipeline was built. The pipeline implements the steps defined in the KDD process and will include everything from cleaning the raw data to visualizing the results. From the implementation perspective, the processing pipeline comprises four main stages: data fetching and transformation, pattern detection, pattern matching, and result visualization.

The data fetching and transformation stage handles the fetching of thesis-related data from the database. This was implemented by fetching the data from the selected customer. The data was limited to two months of collected data. Next, the necessary data-cleaning steps are executed. Lastly, the data transformation includes data anonymization and data conversion to the format expected by the task discovery algorithm.

The second stage consists of executing the task discovery algorithm. The transformed data was given as input to the algorithm. The minimum pattern length and support count for the algorithm are defined in this stage. The output of the algorithm is a list of patterns that match the given search criteria.

After generating the list of the most frequent sub-sequences in the second stage, the third stage will match the patterns to the original data set. Each episode of window steps in the collected data that match any of the generated patterns is converted to an instance of a repeating task. The repeating task instance includes usage statistics from each step, for example, mouse click amounts and information about copy-paste events within the task. The matching algorithm starts from the discovered patterns that have the highest number of steps and works down to shorter patterns.

For example, let us define that the longest frequent pattern that fulfilled the minimum support count requirement has a length of 8 steps. The matching algorithm will start with

a sliding window of size eight and search for all the instances where the sliding window matches any of the discovered patterns with a length of 8. When a match for a pattern is found from the sliding window, all the steps in the sliding window are removed from the data set to prevent the same window visit from being associated with multiple task instances. The next step is to shorten the sliding window to the length of seven and perform the pattern matching for patterns with seven steps. This is repeated until the minimum pattern length is reached. In short, the pattern-matching algorithm aims to ensure that a window visit belongs only to one task instance and that the discovered patterns with the highest number of steps have the highest priority when assigning window visits to repeating task instances.

Lastly, visualization of the results is a focal part of the data pipeline as it allows humans to understand and process the patterns (Hand et al. 2001). DFG graphs are widely used in process mining visualizations. DFG visualization method was selected for the visualization of the found patterns and tasks. The DFG consists of nodes connected by directional arcs or arrows that represent directly-follow relationships between nodes (van der Aalst & Carmona 2022). On a task level, each node in the DFG visualization represents a window. While analyzing the discovered tasks, a central metric for windows is the duration spent on them. When representing tasks with a DFG, these time metrics could be added to the nodes. The case company has an existing DFG visualization tool implemented for process mining. This tool will be used as a base for the result visualization and can be modified further to support the task analysis needs.

A problem raised during the visualization stage implementation is how the different task instances should be grouped under various main tasks. There should be some groups for different types of task instances, as having multiple unrelated task flows in the same visualization will increase the difficulty of the result interpretation. The initial approach is to group all the task instances with the same set of unique window names under one high-level task.

After implementing the end-to-end data pipeline, the data pipeline was executed for the first time. Simple settings were selected for the analysis. Each step was named based on the application and window name, and the unknown windows were included in the data. As described before, the user sessions were divided into two if the unknown window duration was longer than 2 minutes. The data was anonymized in the same way as in the data discovery phase. Also, another data set was later prepared without the anonymization to support the case company’s internal results review.

To operate the pattern discovery algorithm, a minimum number of steps and support count must be defined. For the initial data analysis, a minimum of 4 steps and a support count of 200 were chosen. The pattern discovery algorithm was executed to extract patterns that met these criteria. The extracted patterns were subsequently saved to the database for further analysis of the most repetitive tasks.

For the result evaluation, the visualizations were prepared. The case company's current process mining features were slightly modified to display the task discovery data. Also, a small presentation was prepared about how the task discovery algorithm was developed. The purpose of the presentation is to give more background information to the participants of the result evaluation process.

The initial analysis of the tasks revealed that a significant proportion of the most common task patterns were indicative of repetitive switching between two windows. Furthermore, it was observed that in a considerable number of instances, one of the windows involved in the repetitive switching was an untracked window. In Figure 9, an example task variant is presented in which the user navigates between an ERP window and an untracked step. This finding may suggest that specific IT systems, which appear to be relevant for some day-to-day tasks, were not included in the data collection for some reason.

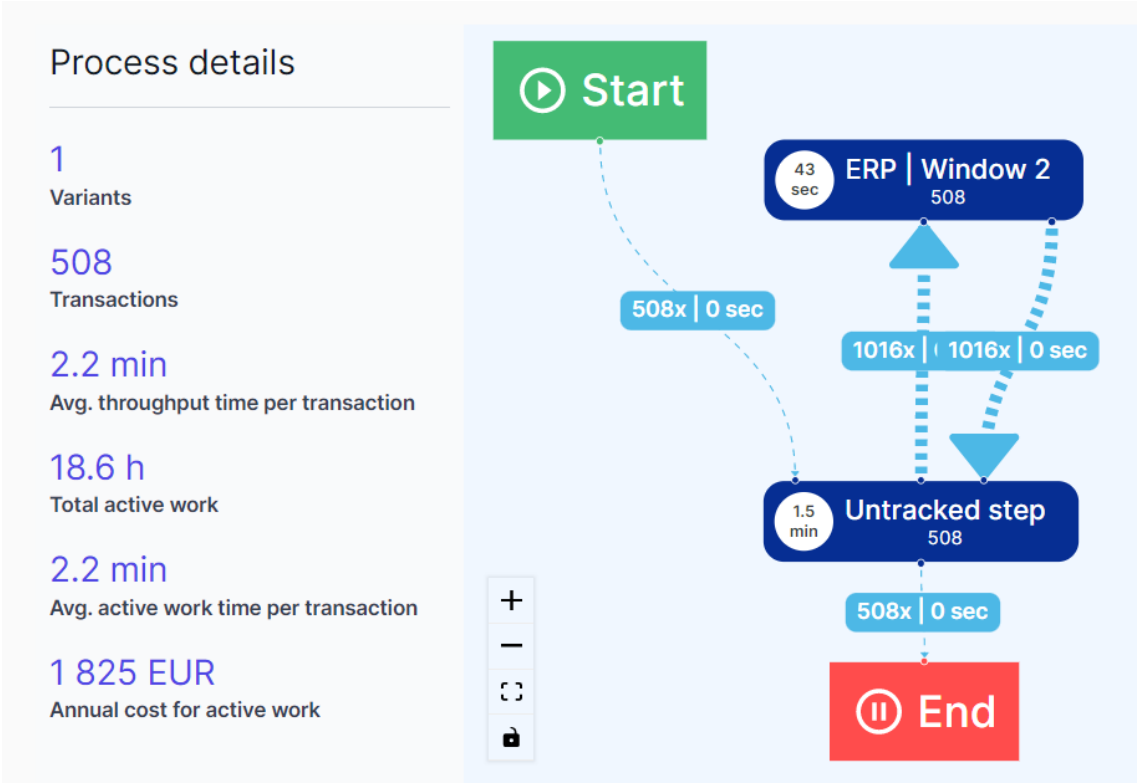

Figure 9. *Variant with visits to an ERP window and untracked windows*

In some instances, a repetitive task was limited to only a single window. This is not unexpected, as it is possible to switch between windows that are categorized with the same name. For example, this may occur when switching between invoices that are processed in similarly categorized windows or when alternating between two similar Excel files. Figure 10 shows a task variant that serves as an exemplar of this type of behavior.

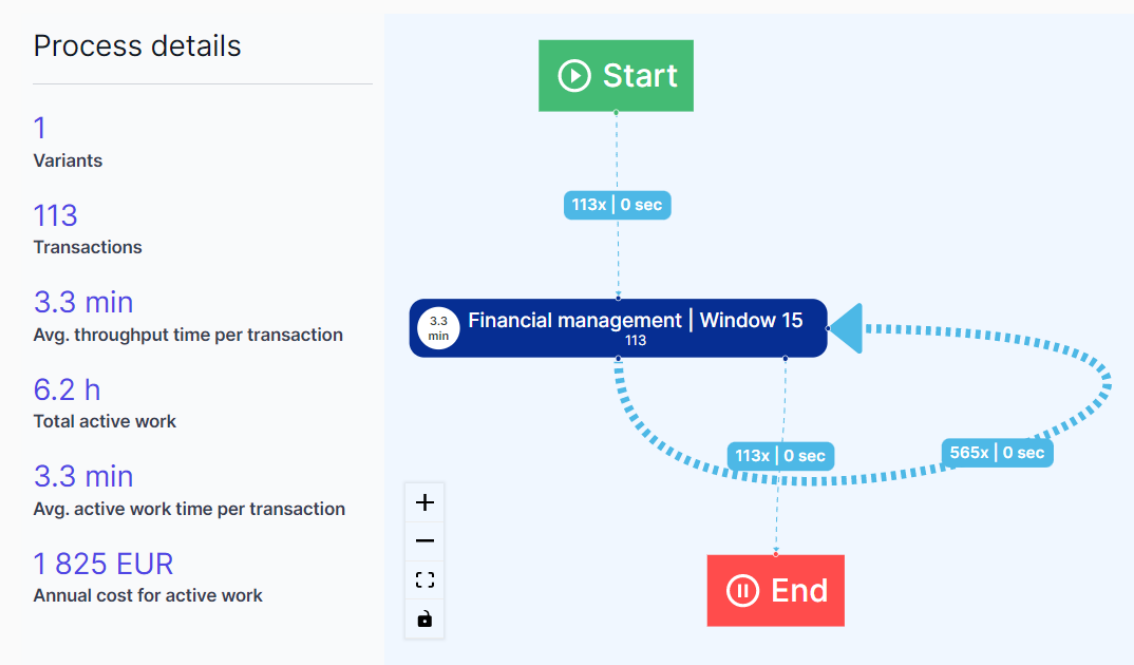

Figure 10. *A variant where similarly categorized windows are visited repeatedly.*

These findings already highlight that there may be at least two distinct subtypes of repetitive tasks present. The first subtype is characterized by the employee taking multiple steps in core-system windows, potentially indicating a more structured task flow to achieve a specific objective. Figure 11 illustrates a task flow variant that exemplifies this type of task, as it includes multiple windows and applications within the flow. The second repetitive task subtype is characterized by tasks with more back-and-forth navigations. This latter subtype could describe tasks such as comparing two documents or manually transferring data between 2 windows. These findings suggest that there may be variations in the nature of repetitive tasks, which can offer an interesting research branch about the implications of these variations.

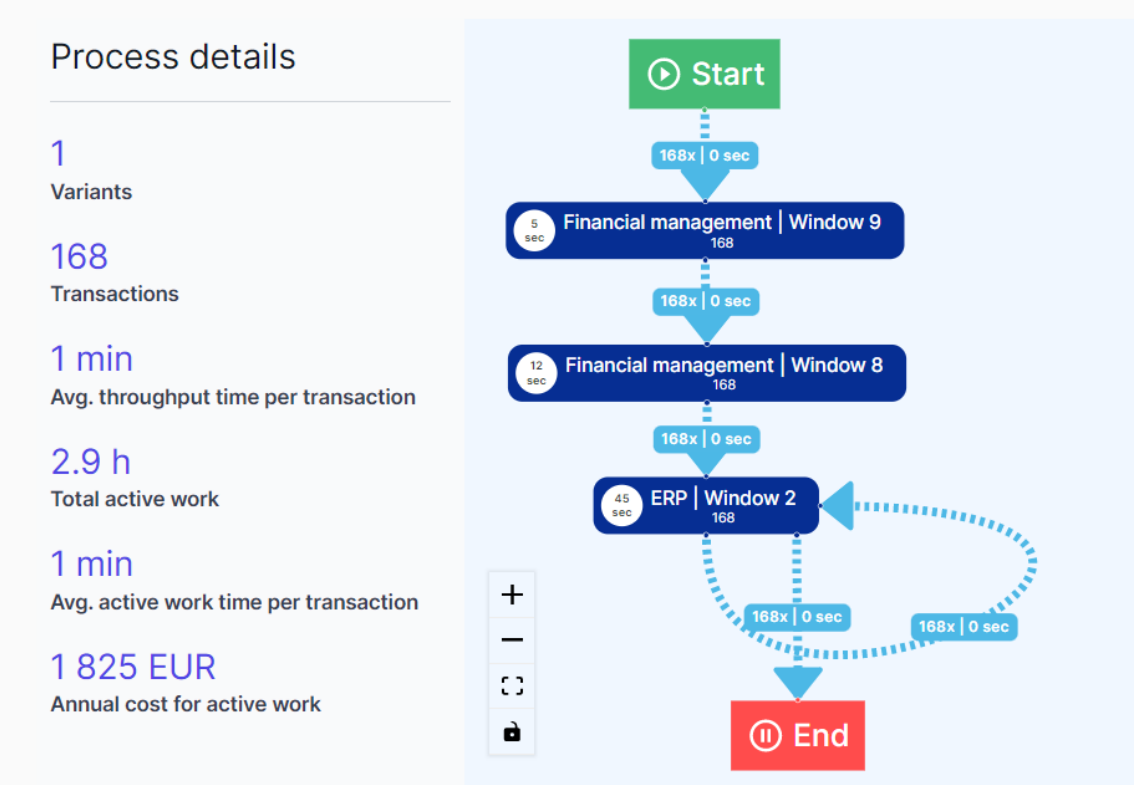

Figure 11. *Variant with multiple unique windows and applications*

Upon being familiarized with the initial findings, the main talking points were prepared for the meeting with domain experts. Based on the initial findings and ideas generated during the implementation phase, two key focus areas were identified. The first area focuses on the differentiation of the two task subtypes discussed earlier, with the aim of enhancing the analysis of their implications. The second area of focus is on how to improve the grouping of various tasks. The aim is to enhance the understandability of results when it comes to differentiating tasks that relate to different objectives. The result evaluation is not limited to these topics, as the aim is to get as much feedback as possible from individuals who would be interpreting the data and making concrete improvement suggestions.

5.2.2 Result Evaluation

The review session was conducted through an in-depth discussion about the outcomes of the first cycle with the two domain experts. During the results review session, the first discussion was centered on the repetitive visit patterns between one or two windows. It was determined that a significant proportion of tasks were characterized by such repetitive navigation patterns. Both repetitive and more diverse tasks were seen to be interesting results, as they can both yield valuable insights upon further analysis. As a result of the discussion, it was agreed to split the data into two categories in the representation

of the results to support the analysis of the results better. The following working names were given for the categories: repetitive tasks and flow tasks.

An observation was made that a relatively substantial proportion of tasks included untracked windows. It was noted that the customer company had not included emailing applications in the data collection, which could account for a significant portion of unaccounted time. It is also possible that other missing systems are contributing to this phenomenon. The outcome of the discussion was that showcasing untracked applications can provide valuable insights into potential additional systems that were not included in the data collection configurations for one reason or another. Unexpected task patterns and systems can also highlight knowledge work tasks that the managers were not expecting. These could be, for example, tasks that were introduced by the rapid digitalization, which the managers were not able to keep track of. Discovering these tasks can help managers understand how their subordinates perform their work and how to support them in their jobs better.

One of the main areas of discussion was the development of an efficient method for grouping different task instances under high-level tasks. By categorizing the tasks, the analyst can quickly understand on a high level where the most frequent tasks occur across different processes and IT systems. This was deemed as an important prerequisite to make the results easier to interpret and analyze. Finding tasks relating to high-level tasks would also be an important prerequisite for finding standardization potential. It was suggested that each task instance should be linked to a window that is seen to be the most relevant to the task. This window will be referred to as the task's base window. The base window will then be used to group the task instances in the result representation.

As a result of the discussion, it was agreed that the following logic should be incorporated into the task grouping algorithm. If there is a window that clearly appears in the task instance more frequently than any other window, that window should be designated as the base window. If there is no window that is clearly dominant in the task, the largest window should be selected as the base window. The largest window is defined as the window that has the highest number of visits when examining the entire dataset. The hypothesis is that by associating the task with the window that is most popular, the analysis will yield the most meaningful results. Uncategorized steps will be excluded from the base window determination process.

A point that was also raised during the review was the inclusion of team information within the high-level tasks. The knowledge of which teams are performing a given task

can be crucial in understanding the nature of the task. For example, this can immediately help distinguish between tasks that are specific to a particular team versus those that are organization-wide, which may have different implications.

5.3 Cycle 2

5.3.1 Performing the Analysis

A new algorithm for splitting and naming high-level tasks was implemented based on feedback received in the previous iteration. The proposed algorithm consists of two stages. The first stage attempts to identify a dominant step in the task sequence by analyzing the frequency of visits for each step in the sequence. If a dominant step is not found, the second stage falls back to selecting the step with the highest total number of visits in the dataset.

To determine the dominant window, a novel approach was implemented that calculates the frequencies of the most and the second most frequent steps in the task sequence. If the most frequent step's share is 20 or more percentage points larger compared to the second most frequent step, the most frequent step is selected as the name of the high-level task. For example, let us look at a case where a task sequence has five items. Among these items, one item appears three times, while the other two items appear once each. Using the algorithm, we can determine that the most frequent item's frequency is 60%, while the second most frequent item's frequency is 20%. The difference between these items is 40 percentage points, which exceeds the threshold of 20 percentage points. Thus, the most frequent window is selected as the dominant window in this case.

In cases where no task name is found using the first method, the algorithm uses the second method of selecting the step with the highest number of occurrences. To implement this second method, functionality was added to calculate the number of occurrences for each step in the whole dataset beforehand. This approach provides a reliable secondary method for selecting high-level task names, even if the task sequence spans over multiple application windows.

To distinguish between repetitive tasks and flow tasks, the "is-repetitive-task" property was added to each high-level task. This property was used in the visualizations to split the data into these two categories. A simple algorithm was implemented to determine if a task was repetitive. The algorithm checked if the task steps consisted of only one or two unique step names. If this condition was fulfilled, the task was marked as a repetitive task. In contrast, if the task consisted of more than two unique steps, it was interpreted as a flow task.

Minor adjustments were made to the configuration of the task discovery algorithm to match the other changes. The minimum pattern length remained unchanged at 4, while the minimum support count for task instances was lowered to 50. This decision was made to provide more details for each high-level task, as lower cut-off points for different task variants can reveal less frequent but still possibly interesting variants. However, it is important to note that decreasing the minimum support count can result in reduced performance of the analysis, as more data must be processed and stored in the database. The trade-off between increased detail and decreased performance is a factor to be taken into account, but for now, the expressiveness of the task discovery was prioritized.

To support the algorithm changes, the database model that stores each task instance was updated to include team ID and "is-repetitive-task" fields. These new properties were utilized in the visualizations to present more detailed and organized results. The team information is stored as IDs in the database. The team ID field was transformed into actual team names as a preparation for the visualization. For each high-level task, a distribution of how much time each team used for that task was visualized. Figure 12 illustrates the team distribution visualization when listing the results, demonstrating how the analysis can quickly get a high-level understanding of what teams are working on the high-level task.

Figure 12. *List of high-level tasks with team distribution*

To split the analysis of task sequences, the result visualization incorporated the "is-repetitive-task" information to categorize high-level tasks into repetitive and flow tasks. This was visualized by dividing the task result list into two tabs, allowing analysts to easily switch between them to find the type of task that they are interested in. This change provided a clear and intuitive way to differentiate between different types of tasks.

The new high-level task categorization provided an interesting take on the analysis of task sequences. The obtained results seemed to highlight task hubs better. To illustrate this, Figure 13 shows the top five variants for a high-level task where the base window was “ERP | Window 2”. While the flow visualization can become quite complex with multiple variants and numerous back-and-forth navigation patterns, it can still provide a useful overview of the windows that are most frequently connected to tasks performed around “ERP | Window 2”.

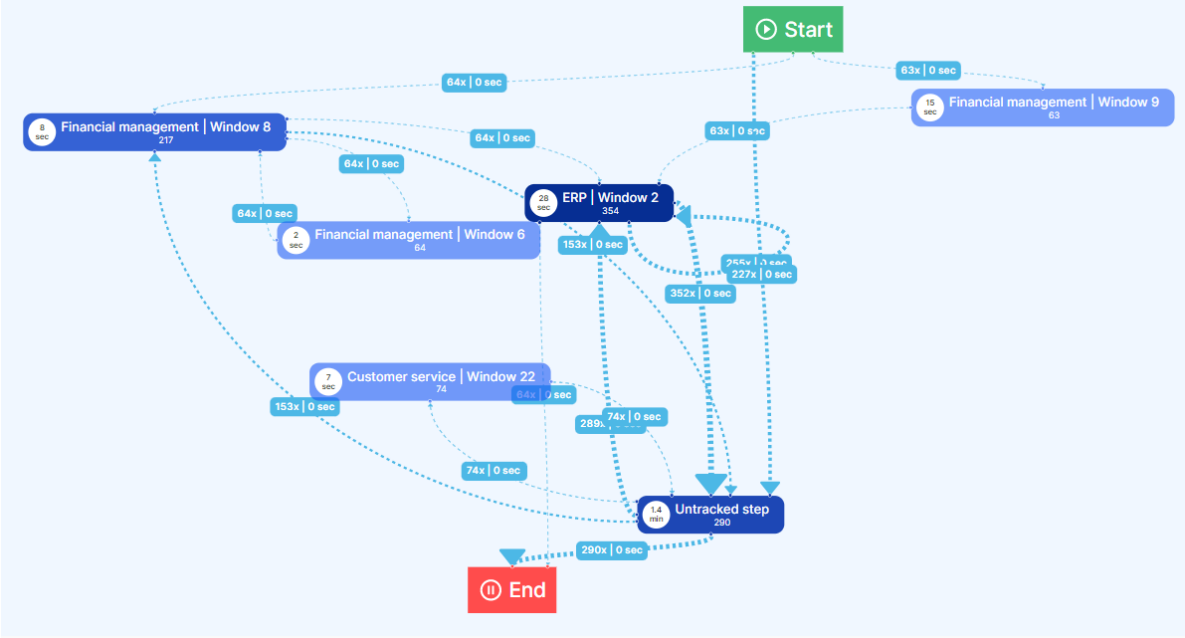

Figure 13. *Top 5 variants of “ERP | Window 2” tasks*

The improved visualizations and task categorization were significant steps forward in the task analysis process. The results were delivered to the domain experts in preparation for the next evaluation and feedback session.

5.3.2 Result Evaluation

The first discussed topic during the result evaluation was filtering the results. The domain experts would like to have more options to find specific types of tasks from the results. They were looking for more filtering options to help analysts narrow down the results and find specific types of tasks related to different use cases and improvement initiatives. During the review session, various filtering options were suggested, such as filtering tasks based on teams, connected business processes, or specific step combinations. Overall, these filtering options could make it easier to focus on particular areas of interest and analyze the data in a more targeted manner.

In addition to the benefits mentioned earlier, filtering the results also provides an opportunity to connect the feature to other product features. By allowing the navigation from

other views to the task listing with inherited filters, an analyst could easily access and explore the tasks related to the context they are currently working on. For instance, an analyst examining a business system window could quickly navigate to a view that displays all the tasks associated with that window by applying relevant filters. This capability would enhance the usability of the feature and enable analysts to gain deeper insights into the task flows.

As filtering options were seen as a valuable addition to the task discovery tool, the following filters were agreed to be added: team filter, application/window-related filters, business process filter, and task pattern-based filters. These filters and their possible use cases are covered individually more in-depth in the following paragraphs. The new functionality would include defining one or more of these filters for the database query. By defining filters for the query, the user would get all the matching results grouped under one high-level task, thereby replacing the default high-level task grouping. This new functionality provides analysts with the flexibility to use the task discovery feature either in a more general mode to understand different high-level tasks or by defining filters to generate their own classification criteria for the high-level task.

The feedback from the domain experts included some interesting use case definitions where the filters could generate value for the analyst users. The use case descriptions helped to identify the needed filter. The discussion on use cases also demonstrates how the developed approach to task discovery can offer versatility in a wide range of customer industries and contexts.

The first discussed use case would be to gain a deeper understanding of task flows in scenarios where the majority of visits concentrate on one primary IT system but could also be supplemented by additional supporting systems. This approach was described to offer useful insights into conformance within main business systems. Analyzing these flows can provide valuable insights into how workers carry out their tasks within the main business system, as well as add an understanding of when additional supplementary systems are used. This approach also seems to support the standardization and RPA initiatives discussed in the first chapter. Understanding different task variants in the main system can show how standardized the tasks are within this main system. From an RPA perspective, understanding the volume of the task, involved IT systems, and task rules in terms of taken paths are key requirements for discovering RPA potential. An example scenario could be where a worker needs to access a document application while working within an ERP system to acquire additional information for the task's context. To find these flows, a filter was discussed that would filter task flows based on the proportion of

work completed within the main system versus other supporting systems. The functionality would be to find all the task flows where a given business system's windows account for 50% or more of all visits performed in a task.

One discussed functionality was to identify tasks that initiate from specific windows, which could provide insights into potential automation targets. For example, by tracking task flows that begin from a ticketing system, analysts can identify common paths taken to progress or resolve a ticket. Analyzing these flows could, for example, aid with identifying tasks that can be automated to improve efficiency. It was agreed that adding a filter for the start window of a task allows for the identification and analysis of task flows that begin from specific systems. A filter for the end window was also agreed on. Finding tasks that end in a specific window can discover paths that have similar use cases as described with the start window filter.

Another important use case for the task filtering methodology is to gain deeper insights into the data transfer between applications in the form of copy-paste activities. The data flow perspective is important as poorly implemented information flows will affect the task performance (Simperl et al. 2010; Kersten & Murphy 2015). By understanding the task context around a data transfer, it is possible to identify areas where the current flow can be improved or eliminated. For example, if data is copied to an ERP system from another application, it may be useful to analyze the context of the transaction and determine why the data needs to be transferred between applications. To achieve this, the implementation could involve adding data transfer information to each task, which would then allow filtering by certain types of data transfers.

The last discussed topic related to task filtering is the ability to gain insights into the task flows occurring around business process steps. For instance, discovering process steps where document applications were used to support the main business processes can provide valuable insights for process analysis. Overall, adding task-level analysis for business process steps can provide additional task-level details to enrich insights gained from process mining. Adding filters to identify tasks where a particular window or application appears together with the detected business process step was seen as an essential addition to filtering capabilities.

In addition to the significant filtering improvements discussed earlier, other minor enhancements were discussed as well. One key suggestion was to provide more detailed information in the initial task listing, which would allow analyst users to quickly identify interesting tasks without the need for an in-depth review of each high-level task. To

achieve this, each application and its usage time for each high-level task could be visualized already in the listing view. The approach to visualization could be similar to how team times are currently visualized. Additionally, the listing view could be enhanced with a sneak-peek feature, allowing users to quickly review the top 5 variants for each high-level task without needing to open the analysis view. These enhancements would then have the potential to significantly streamline the task analysis process, providing the key information in a concise format.

During the review session, three new improvements were suggested for the task mining algorithm. Firstly, it was noted that many patterns begin or end with an untracked window. To address this, the algorithm will be updated to disregard patterns that feature an untracked window as either the first or last window in the sequence. Secondly, having the same window repeating multiple times in a flow was seen to cause some unnecessary noise in the results. If multiple consecutive windows are categorized with precisely the same name, these windows will now be compressed into one step. Lastly, it was agreed that a more dynamic support threshold, which adjusts according to the length of the pattern, will be tested. By prioritizing longer patterns with lower thresholds, the hope is that the algorithm can provide more comprehensive insights into longer task flows. Overall, these small adjustments are expected to enhance the accuracy and relevance of the task analysis results.

5.4 Cycle 3

5.4.1 Performing the Analysis

The third iteration started with implementing new capabilities for the pattern discovery algorithm. One of the major changes implemented was the fine-tuning of how the minimum support count operates. The objective was to lower the threshold for longer tasks, enabling the algorithm to capture more comprehensive patterns. The new algorithm introduces a linear reduction of the minimum support count as the task lengths increase. The implementation works so that the minimum support count gradually decreases with the pattern length until it reaches 33% of the initial support count, which will be the absolute minimum threshold for all the longer patterns.

In order to enhance the filtering and result representation capabilities, additional changes were made to incorporate data copy-paste activities and related business process information into the discovered task instances. A copy-paste event was associated with a task instance when a data copy activity was identified in one of the task instance windows, and the corresponding paste activity was detected in another window belonging to the same task instance. The collected data set also includes information about related

business processes. The collected business process information was now added for each window in the task instance. With this implementation, each task instance window connected to a business process is enriched with details such as process name, transaction ID, and interpreted process step name. These enhancements to the pattern discovery algorithm will offer more context and granularity to the user activities.

Other changes were made in the task discovery algorithm according to the received feedback as well. Firstly, consecutive windows with the same name were compressed into a single window visit. Additionally, a restriction was implemented to ensure that task patterns could not start or end with an untraced window. These changes aim to simplify the results by removing extra variance and avoiding the inclusion of incomplete information.

Following the implementation of the necessary changes for the task discovery algorithm, the focus shifted towards implementing new data fetching and filtering capabilities. Specifically, the main focus was to introduce new filtering capabilities for database queries. The new filters included search criteria for task length, start and end windows, included windows, data transfers, and process connections. With the introduction of the new filters, the task grouping and listing logic were updated. If any filters are applied for the queries, all the matching tasks are now grouped under one high-level task, departing from the original functionality where multiple high-level tasks are listed based on their categorization. Also, the newly introduced information regarding data flows and connected business processes was included to be fetched in the data queries.

In terms of result representation, several new functionalities were introduced to the task discovery feature’s user interface. A notable addition was the introduction of a dedicated interface for filters, providing users with a clear overview of active filters and the ability to set new ones. Figure 14 showcases the implemented filtering menu. Furthermore, it can be seen from the figure that when active filters are applied, the listing view has been updated to showcase a singular result in which all task instances matching the defined filtering criteria are grouped.

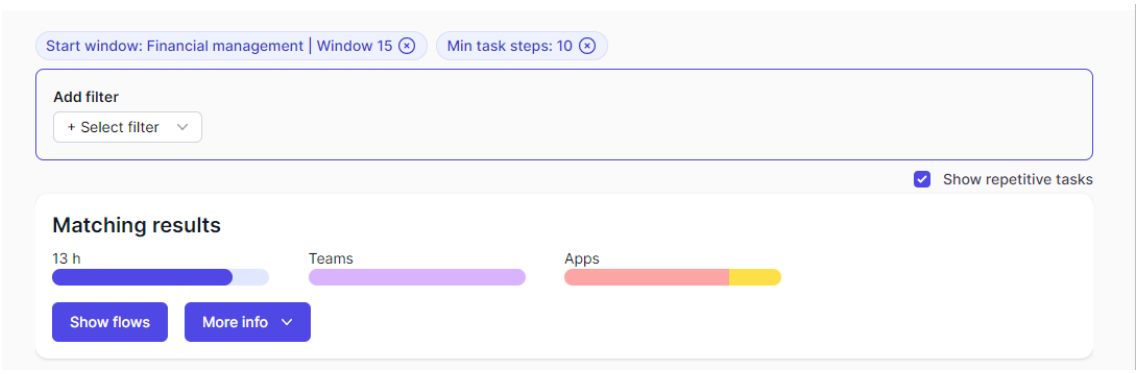

Figure 14. *New filtering menu and result display*

The user interface for separating the repetitive flows and other flows was changed. Now, all the results are shown in the same list, and a toggle feature was introduced, enabling users to selectively include or exclude repetitive flows from the list based on their analysis requirements. Repetitive flows were distinctly marked within the listing view, allowing users to differentiate them from other flows easily.

A new list visualization was implemented for listed high-level tasks. Mainly, users can now see the top 5 window flows and the largest copy-paste flows for each high-level task already in the listing view. Moreover, the listing view now presents the total application usage times for all high-level tasks, enabling users to quickly have an overview of what applications the high-level task consists of. Figure 15 demonstrates the changes to the enhanced list visualization with the inclusion of top window flows, the largest data flows, and total application usage times.

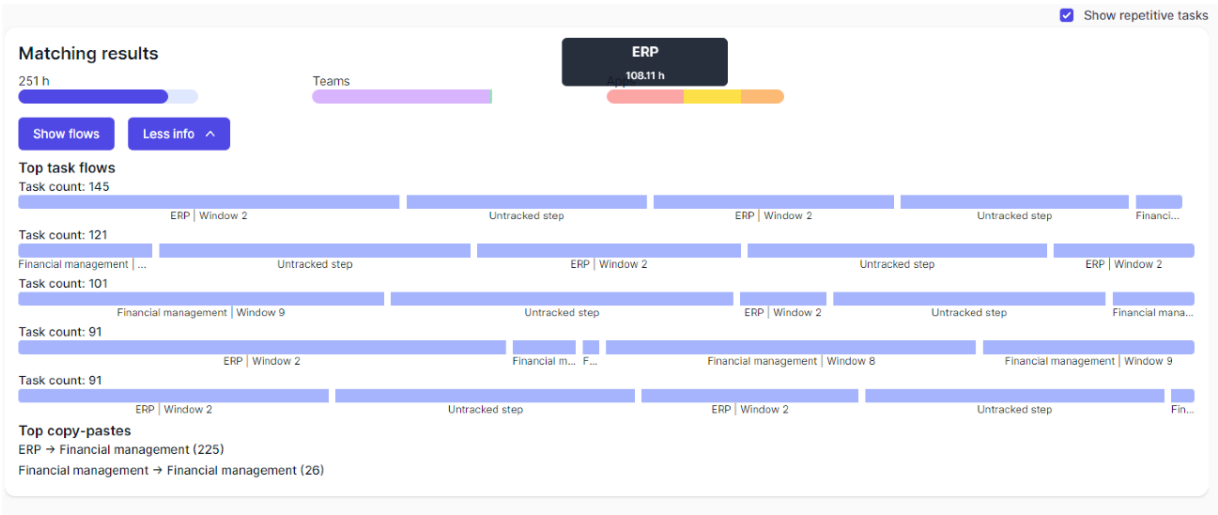

Figure 15. *Application usage, tasks, and copy-pastes in the listing view*

The DFG visualization for task variants was enhanced. The visualization of data transfers and business process connections was incorporated into the view. Each window in the

visualization now displays the number of copy-and-paste activities performed within the selected task instances. Additionally, the visualization includes information on the number of times a business process is associated with each window, highlighting the frequency of business process involvement in task execution. Figure 16 exemplifies these changes to the task flow visualization.

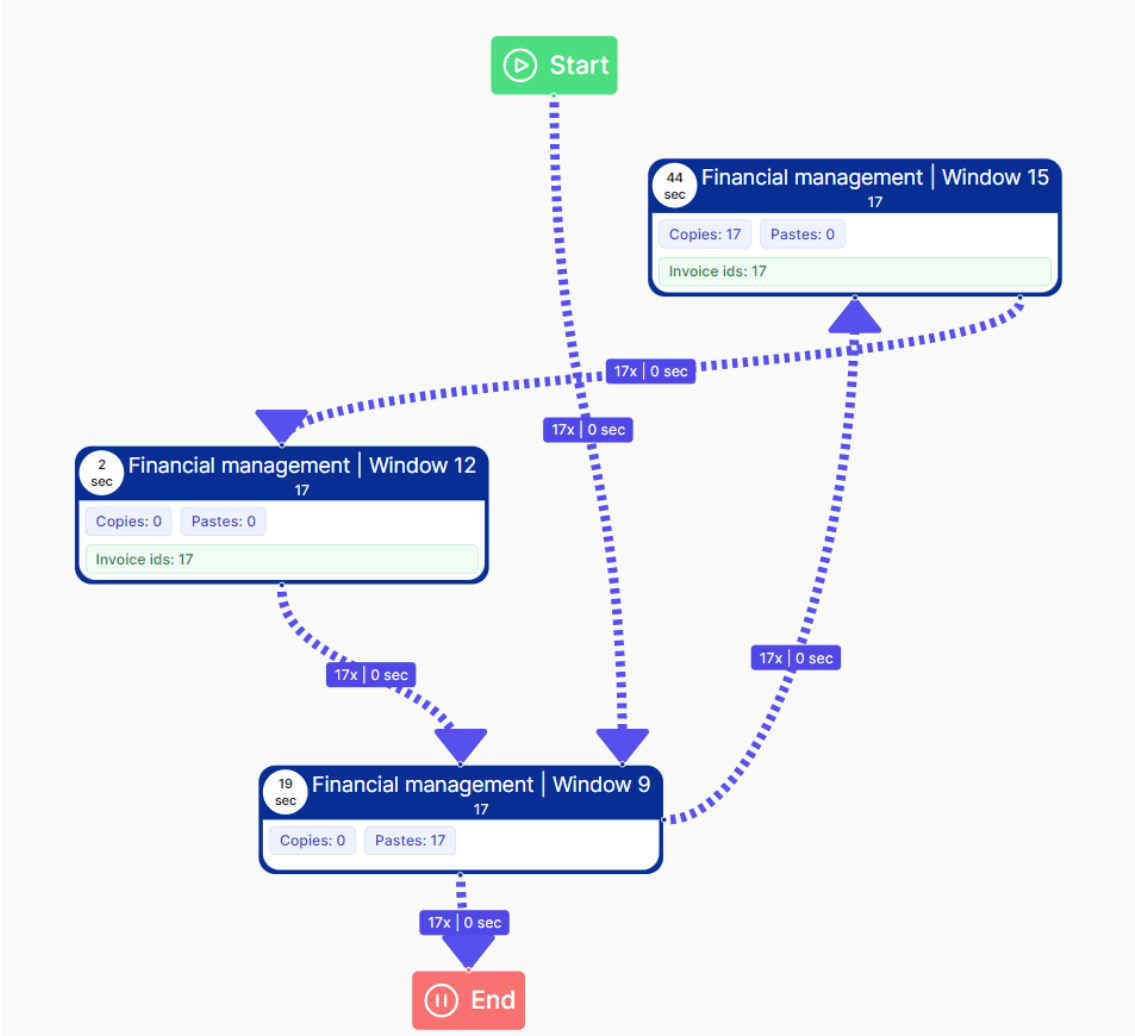

Figure 16. *Task flow visualization with copy-paste and business process data*

In conclusion, the new filtering capabilities and the analysis of data flows and process information have been the main drivers to improve the result interpretation during this implementation round. The next step is to evaluate how effective the filtering improvements are for the result interpretations and how interesting extra context the data flow and process analysis can provide for the analysts.

5.4.2 Result Evaluation

During the evaluation session, potential applications for the new filtering capabilities were explored and discussed. Firstly, the data copy filters were tested by using a set of filters

to selectively identify tasks involving data transfers from the ERP system to the financial management system. Some interesting use cases were discovered where the task consisted of repeatedly switching between ERP and financial management system windows. These findings indicate a recurring pattern of multiple distinct data points being systematically copied from the ERP system to the financial management system.

The business process filters were also assessed during the evaluation session. For instance, a filter was used to identify tasks related to the invoice payment process within the financial management system. One of the observed patterns showcased instances of the worker's seamless navigation between the invoice approval and payment-related windows. Intriguingly, on each occasion of this task variant, the task would then proceed to a distinct reporting window within the financial system, where data was consistently copied from the previously visited payment window. Furthermore, by using the minimum task length filter, the analysis unveiled longer task flows where the previously described task was executed multiple times in a row. This most likely indicates that the worker followed the same sequence while processing multiple distinct invoices consecutively. The described findings demonstrate the insights that can be derived from the introduction of the new filters and the visual representation of copy-paste flows.

The utilization of filters is usually interesting when the analyst possesses a specific contextual understanding derived from their domain expertise or previous insights gained from the collected data. The discussion steered towards there usually being some specific contexts being defined already prior to the task analysis, such as examining the events preceding and following the invoice processing step. This discussion tied in with the examples shared in the previous session, which showcased concrete use cases for when the filters are especially useful. Notably, there are instances where predefined rules can be established prior to task discovery analysis to identify specific business-related tasks. For instance, custom search criteria could be defined to capture tasks after creating a new ticket in a ticketing system window or to capture the activities preceding and succeeding the user's engagement with a business process-related window.

As a result of the discussion, it was deduced that a two-phased analysis could be employed for the task discovery process. The first phase includes matching predefined task patterns to uncover the execution details of tasks linked to established business processes. Although the outlines of a task might be known beforehand, it can have various unknown variations and side steps that merit additional analysis, such as identifying bottlenecks or generating automation specifications, which justifies this sort of approach for task discovery.

The second phase encompasses exploring activities that fall outside of predefined task patterns, leveraging the existing task discovery method to identify tasks in a more organic manner. The 2nd phase holds the potential to unveil some previously unknown shadow processes and shed light on the less visible but repeating day-to-day operations. Overall, the two-phased analysis presents an interesting direction for task splitting, where the ratio of tasks related to known business processes compared to other activities inherent to the daily knowledge work can be analyzed.

To implement the predefined task capturing, it was agreed that some basic window-based matching rules for tasks would be implemented. To start with, two matching methods will be implemented to assess the new functionality. The first method entails capturing an x-number of windows preceding and succeeding a matched business process step. The second matching algorithm involves matching a sequence starting and ending with specific windows, with an option to specify the maximum number of steps between the defined start and end points.

Furthermore, the newly introduced copy-paste visualizations were reviewed. The data flow in the visualizations proved to be interesting to analyze. It offered substantial contextual information regarding users' actions and revealed the precise locations of manual data transfer bottlenecks from each task’s perspective. Copy-paste analysis prompted a discussion on how to enhance the user activity storytelling capabilities of the flow visualization.

In a similar manner to copy-pastes, the inclusion of text input information in the visualization would enhance context regarding user actions within the visited window. Moreover, incorporating additional high-level categorizations of activities can significantly improve the analysis of results. These categorizations can include distinctions such as whether a window visit entails intensive and interactive tasks involving typing or mouse clicks or if the visit is by nature more static.

5.5 Cycle 4

5.5.1 Performing the Analysis

The task discovery analysis was enhanced with the option to define custom patterns to match predefined tasks. The discovery algorithm and logic remain the same when it comes to automatically discovering frequent patterns from the raw data. The new matching feature was introduced as an extension to the task-matching part of the analysis, where discovered patterns are applied to the data. If predefined tasks are configured,

they are matched from the data first. After the predefined tasks are matched, the discovered patterns are then matched from the remaining data.

New algorithms were introduced to match predefined tasks. The first developed algorithm matches patterns that start and end with a specified window. The user can also customize the matching by defining the minimum and maximum number of steps that there can be between the matched windows. The second algorithm for matching predefined tasks includes matching a window and then saving steps before and after the window. A user can define how many windows are defined before and after the matched window. In both algorithms, window matching can be defined with application, window, and connected business process name constraints where one or multiple of these fields can be used to generate the matching criteria for a window. In the future, more algorithms could be added to extend the predefined task matching, but for now, the two implemented algorithms were deemed to be enough to experiment with the predefined tasks.

The implementation of predefined task matching also included a definition of templates that are used to define the predefined task matching criteria. These templates include the needed parameters for the functions so that the matching can be customized for each unique use case. This implementation allows the predefined tasks to be flexibly defined for each customer and their task-matching needs. Additionally, new categorizations were introduced for the tasks. Previously, the task discovery analysis split the tasks into flow tasks and repetitive tasks. Now, a third categorization was added to mark the predefined tasks as their own category.

The other addition to the analysis capabilities was to introduce more granularity into the activities performed during a window visit. Currently, the copy-paste flows within tasks can be thought of as user activities. Other user activities within a window should be highlighted to give more insights into the tasks. Hence, two new metrics were introduced for each window: the number of key presses to inform about how much typing was performed and the number of copy-paste activities within the window to inform how much data needed to be reorganized within the window. These metrics help highlight how users interact with data or create it in the task's context.

The second new addition for the activity improvements was to add a general categorization for the work performed within a task's window. The case company already has a functionality to categorize visits to a window based on user activity. These categories are manual text entry, text creation, data moving, peeking, and browsing. Showing the distribution of activities both in the task and individual window levels can give more context

to understanding the task's nature. Activities can provide more detailed insights into repetitive tasks by potentially uncovering if the repetition is due to comparison, data moving, or data insertion. Understanding the activities can also help to understand on a higher level the potential to automate or other ways to improve the task.

With the new changes introduced to the analysis, the next step was to include the new capabilities and data in the result representation. The listing view was enhanced with the predefined task category. As displayed in Figure 17, a user can now select with checkboxes which task types are shown in the listing. The task type is represented with a colored badge on the right side of the high-level task container.

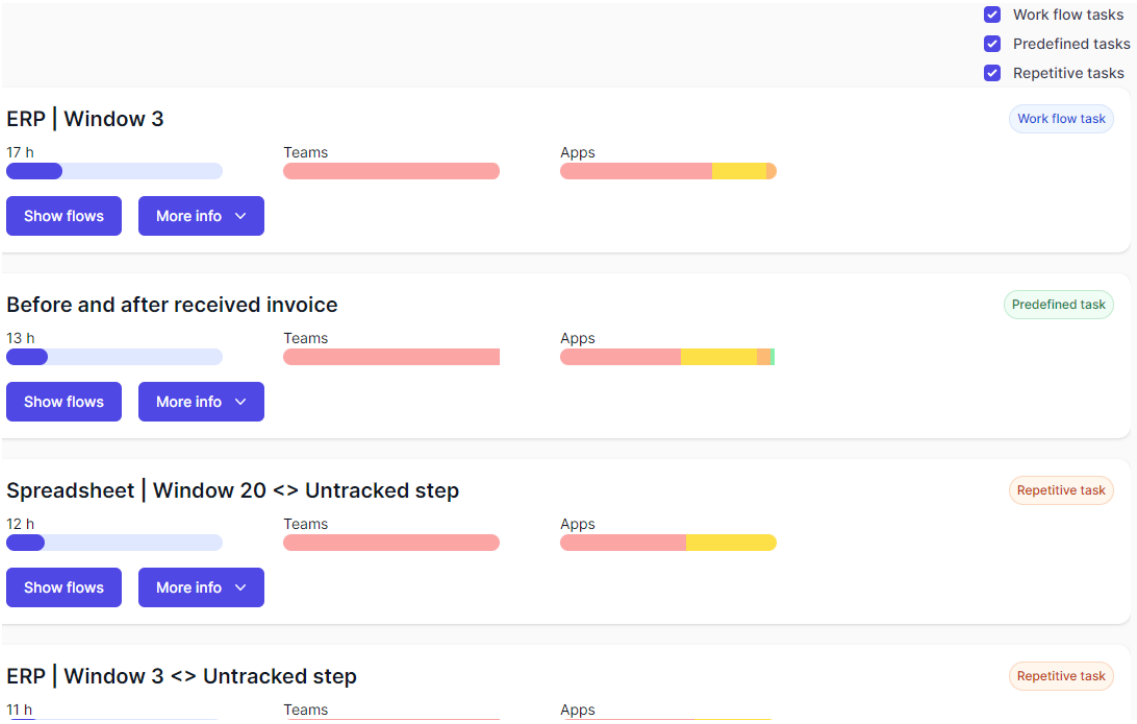

Figure 17. *Displaying different task types in the task listing view*

The activity categorization was introduced as part of the additional information for each high-level task in the listing view, as showcased in Figure 18. The splits are calculated by adding up the activity times from each individual window visit within the high-level task. Displaying the activity splits helps with highlighting the high-level tasks with more manual activities like typing and copy-pasting from more browsing-based and information ingestion-based tasks.

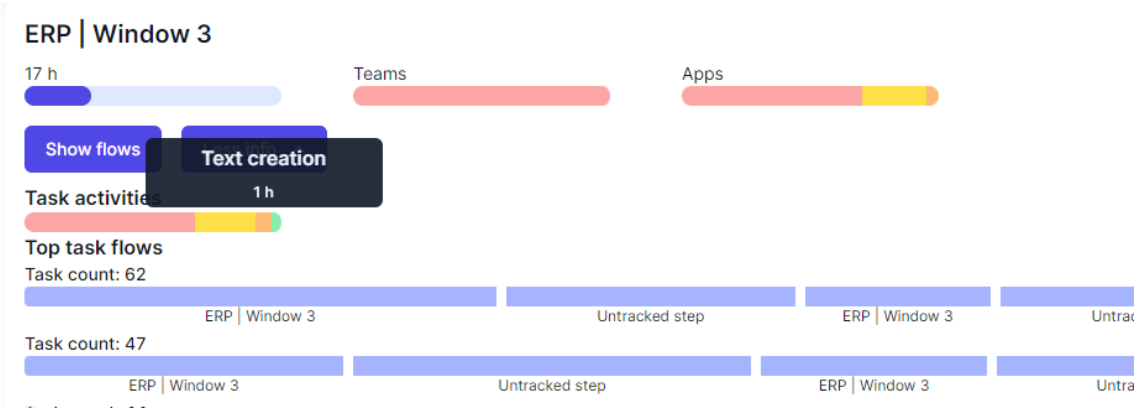

Figure 18. *The distribution of time used with different activities visualized.*

Updates were also made to the flow visualization. First, the new metrics and activity categorizations were aggregated into each step's data. All the activity metrics are visualized in a modal window when a user clicks a step in the flow view. The new metric visualizations are showcased in Figure 19.

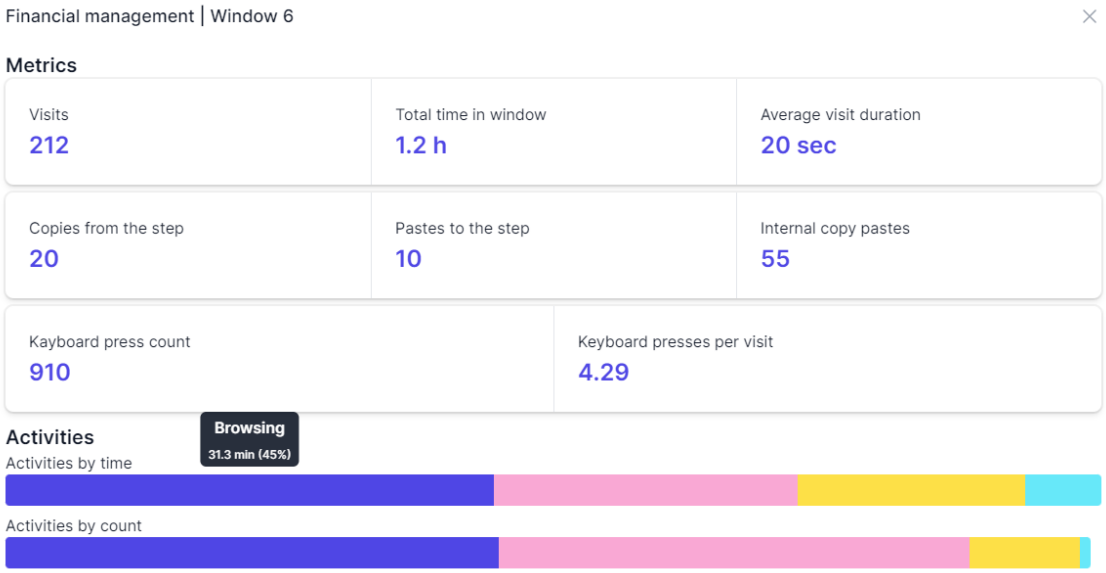

Figure 19. *Metrics and activities are displayed for a window in task flow.*

Finally, a new visualization option was implemented to highlight individual user activities in a more granular way compared to the current flow visualization. This new visualization option is available only when one task variant is selected. The visualization creates a straight-line visualization of each window visit in the task variant. Each window visit gets its own node in the visualization, even if the same window is visited multiple times. This ensures that activities in each separate visit can be displayed with as granular details as possible. For each node in the visualization, a detailed description of activity data is added in written format, as displayed in Figure 20.

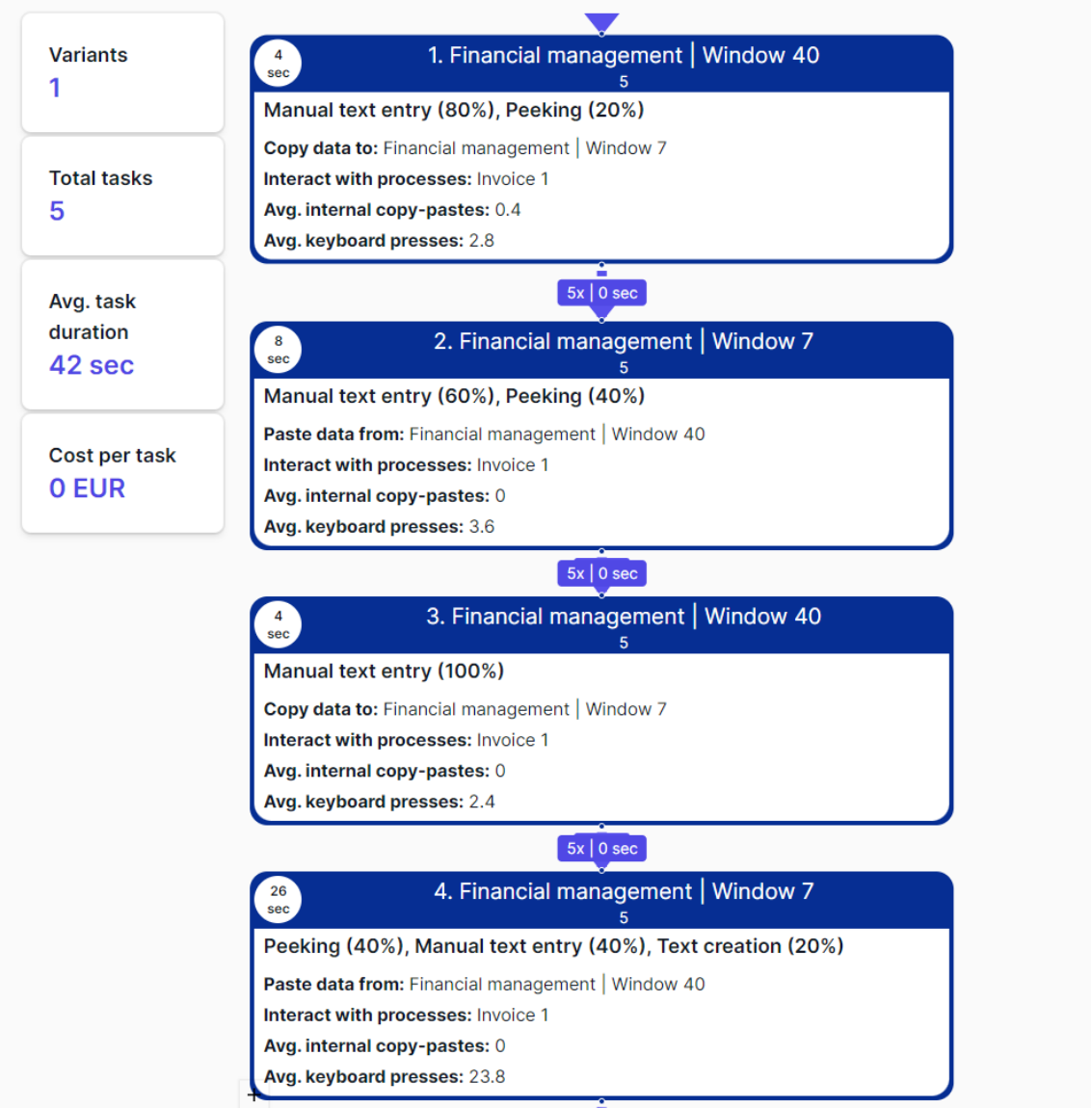

Figure 20. *New detailed task flow visualization.*

The updated visualizations aim to help analysts understand more deeply what happens in singular tasks or task variants on a step-by-step level. The activity data can help understand tasks better on a high level as there is more information about how users act during the task. On the other hand, the activity information can also add more context on a granular level, as the user activities can be tracked for individual tasks and steps.

5.5.2 Result Evaluation

In earlier iterations, only one data set from one of the case company's customers was analyzed. At this point, the data was analyzed for other customer companies to provide a broader scope of data for the experts to investigate. As the iteration period in the action research seemed to have reached its end, some more formal questions were formulated

for the experts to give their final thoughts and findings on the usability of the new task analysis tool. The following questions were given to them beforehand:

1. Can you find and describe auto-generated flows or repetitive tasks that you found interesting while using the tool?
2. Can you give some examples of cases where the filtering gave more insights when having a specific scope in mind (e.g., what happens during copy-paste from application A to B)?
3. Have you found use cases where it would be insightful to define a predefined task?
4. Is the current flow representation insightful from an automation specification point of view? What could be improved still?

The experts were invited to investigate the data from any of the customer companies where the new analysis was performed. In the final meeting, the before-mentioned questions were discussed, as well as any other additional feedback. Based on the meeting, the final results and conclusions were formulated.

For question number one, the main findings focused on repetitive flows. It was found that, in many cases, there are repetitive flows focusing on communication apps where the core business apps are rarely present. This was found to be an interesting insight that can describe how the daily tasks in knowledge work don’t always revolve around core business applications. For the second question about filtering capabilities, the main focus was clearly on the copy-paste filters. The main use cases revolved around moving data between two different applications. Understanding the task flow around data moving can give insights into data-related inefficiencies.

Relating to question number three, the predefined tasks were seen as important for understanding how much variance there is around a specific context. A good use case would be to define a predefined task for tracking what happens around a specific business process-related action. For question number four, the key observations were that the task volume and understanding user activities in windows are useful features for automation projects. A key improvement area was that, currently, the tasks are separated into different variants based on the window flows. From an automation perspective, it would be useful if the variant separation could also be done on the user activity level, enabling the analysis of different activity variations. Another improvement topic was to gain more knowledge on how to best represent and export the data so that it could be used directly in automation projects.

Some concrete examples found from the data were also discussed. An example of a manual data creation flow was observed in a task pattern where a main business application and an Excel file were repeatedly visited. From activity information, you could see that a user first inputted text into the main business application, then switched to an Excel file where they pasted information from the business application, and then continued creating even more text in the Excel file. This was seen as a potential symptom of having to maintain business information in multiple applications and formats. Discovering repetitive work, using copy-paste filters, and understanding data-related activities on a task level were seen as some of the most beneficial features developed during the thesis to discover elimination potential.

A good example of a standardization case was found in a customer's invoicing process. The process had an order-matching step performed in SAP. Looking at the three most common task variants around the step showed that one of them focused on work performed in the SAP Purchase Order window, one in Microsoft Teams, and one in the Order document in Microsoft Excel. This highlighted the fact that there can be multiple paths to perform a similar task depending on the data source.

Another interesting finding was that for many companies within the scope of data review, the tasks seemed to roughly split into two categories. Firstly, there appeared to be tasks that were related to the core business activities and systems. The second task category focused on communication applications where core business applications are rarely involved. The tasks performed in these communication applications seemed more discontinuous in nature, and drawing conclusions from these tasks seemed harder. However, in many cases, a significant share of the work time was spent on communication tasks, which itself can be an interesting finding for managers.

The topic of automation was also touched on during the last review. It was noted that the automation perspective gained the least focus in the final analysis. This seemed initially unexpected as automation seemed to be a very promising topic for task discovery during the literature review. During the discussion, it was noted the lack of automation findings could have been partially influenced by the selected data set. The data set used throughout the research was collected from a company that has some automation programs in place. This could have led to other aspects of the work being highlighted in the results, as the most obvious parts of the job are already automated.

6. CONCLUSION

The research started with two research questions. The first focused on finding frequent patterns from the data to find recurring tasks. The second question focused on developing tools to analyze the tasks. A literature review was conducted to gain critical background knowledge about the key topics related to tasks performed in digital environments. The review also included exploring subjects, such as data analysis and identifying frequent patterns from sequence data. The next step was to perform an action research phase. During this phase, an Apriori-based data analysis was implemented to detect patterns from sequence data. Additional analysis and result representation approaches were then developed during the iterative action research process. The results were discussed and evaluated in each action research cycle with the case company's experts.

Based on the literature review and action research, key analysis factors for task discovery solutions were identified. Three high-level themes were identified to have been present both during the literature review and empirical research. These three themes are task simplification, standardization, and automation. Identifying and understanding these themes during the research process allowed to iteratively improve the analytical model to provide key insights.

The task simplification perspective focuses on the tasks that could be streamlined by removing unnecessary activities or execution logic. Digital systems have the potential to simplify routine tasks and enable tasks to be performed faster and in more innovative ways (Schwarzmüller et al. 2018; Ahmad & Van Looy 2020). Proper and easy management of task-relevant information is a key factor in performing tasks efficiently, and the implementation of Information management tools plays a key role in how well the information is managed (Simperl et al. 2010; Kersten & Murphy 2015). One of the main findings during the empiric research was the detection of task flows that highlighted inefficient patterns around information channels. A concrete example of this working pattern is tasks where a worker needs to input or synchronize business-related information into multiple systems manually. The key features for highlighting inefficiently performed tasks were the detection of task patterns with a repetitive nature and options to search and highlight copy-paste flows within tasks.

The standardization perspective was found to focus on tasks that have a high variance when it comes to the task execution paths. Standardizing processes and tasks can help organizations scale up their processes, adapt them across the organization, and make

the business more stable and predictable (Brahma et al. 2021). Additionally, standardization can be an important prerequisite for automation and help to unify working practices towards the most efficient working methods. The most useful features developed during the research, from a standardization perspective, were filters on a business process step and predefined task definition. With business process step filters, one could focus on a particular task within a business process and understand how different the user flows around the focused task are. Similarly, with predefined tasks, a user could loosely define the outlines of a particular task and get insights into the different execution paths.

The automation perspective focuses on repetitive tasks that could be executed by computers, enabling humans to focus on more complex tasks. During the literature review, the key requirements for automatable tasks were found to be repetitiveness, high volume, structured data, rule-based nature, and involvement of multiple IT systems (Kokina & Blanchette 2019). Additionally, in automation projects, it is important to understand the information that is being consumed, created, or changed during the task (Kersten & Murphy 2015). Similar findings were made based on the domain expert's feedback during the development of the analytical approach. The key features found to be beneficial for automation initiatives were understanding the volume of performed tasks and understanding performed user activities in IT-system windows. Useful user activities to highlight in visualizations were, for example, copy-paste activities and text creation events.

From the action research perspective, these themes factored into solving the organizational problem of improving the case organization's task analysis. The other objective of action research is the creation of an actionable theory. The task analysis method, developed based on themes discovered during the research, can be formulated into an analytical framework. The presented framework aims to contribute to academic knowledge and answer the research questions raised at the beginning of the research. Figure 21 visualizes the key components of the analytical framework.

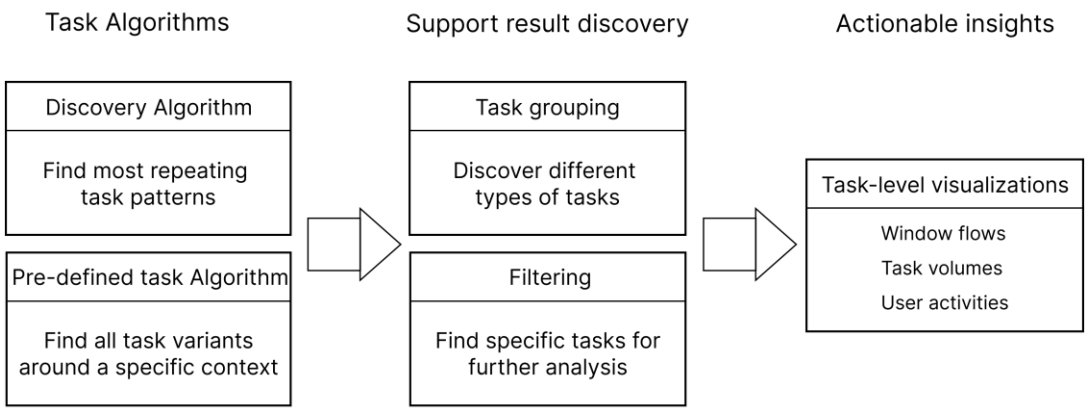

Figure 21. *The developed analytical framework.*

To answer the first research question about how to find frequent patterns from the data to find recurring tasks, a model with two parallel task analysis algorithms is proposed. The first algorithm focuses on matching the most frequent patterns from the data to find the most repeating tasks. In this research, an Apriori-based algorithm was used to find the most frequent episodes from sequence data. The second algorithm allows the capture of tasks based on predefined rules to record all the different task variants around a specific task context.

To answer the second research question about how to formulate and represent the results to support the analysis of often performed tasks and workflows, three key result representation techniques were identified during the research. The first two techniques focus on how to pre-process the discovered tasks to support the analysis and discovery of the tasks on a high level. Automatic task grouping was seen as a key requirement as it allows the analysts to quickly discover different types of tasks and helps them to identify tasks that are related to the same context. The second technique to support efficient result discovery is filtering. Filters help analysts to find specific types of tasks quickly for further analysis. The third identified result representation technique relates to showing actionable insight on the task variant level. Key data points to visualize are task flows on the IT system window level, task volumes, and user activity-related metrics and categorization.

The action research approach allowed a lot of freedom to iteratively understand the underlying problems better. It also provided some freedom to explore research directions that seemed the most interesting. This has been stated to be natural for action research. In some cases, even the original research questions can change (Saunders et al. 2016). Even though the thesis didn't take such a drastic turn, the focus changed to a slightly

different direction than what was initially thought. During the planning phase, the most natural direction seemed to be to improve the capabilities of the discovery algorithm. However, during the research phase, the main focus shifted to result representation and search capabilities on the results, and less emphasis was put on further developing the actual algorithm producing the results.

The current findings also highlight further research and implementation paths. A further research path indicated by the domain experts was to gain an understanding of how to add capabilities to convert the results into concrete instructions or specifications for an automation project. Another improvement area discussed within the case company was how to connect the task dimension with other insights and features within the existing dashboard product.

The research ended up focusing more on the result representation rather than on the algorithm development. This leaves a lot of future research opportunities to understand better how to analyze the data to extract the most useful patterns. The current implementation focuses on finding episodes where the order of events matters. This can, for example, create additional restrictions on finding different variants of the same task. Experimenting with itemset-based discovery, where the restrictions for the order of the events in a sequence are looser, could be an interesting research direction. Also, the results suggest this to be an interesting direction. For example, many tasks seem to be consisting of jumping between two windows. In this case, a more relevant finding could be to gain an understanding that these two windows are tightly related rather than focusing on exactly which order or how many times these windows are visited within a task instance.

REFERENCES

[1] Achar, A., Laxman, S. & Sastry, P. S. (2012). A unified view of the apriori-based algorithms for frequent episode discovery, Knowledge and Information Systems, Vol. 31(2), pp. 223–250.

[2] Ahmad, T. & Van Looy, A. (2020). Business Process Management and Digital Innovations: A Systematic Literature Review, Sustainability (Basel, Switzerland), Vol. 12(17), p. 6827.

[3] Annunziata, M. & Bourgeois, H. (2018). The future of work: how G20 countries can leverage digital-industrial innovations into stronger high- quality jobs growth, Economics, Vol. 12(42), pp. 1–23.

[4] Anon (2021). Creating multiple value streams from big data: Danish insights on aligning automation processes, Strategic Direction, Vol. 37(7), pp. 40–42.

[5] Baiyere, A., Schneider, S. & Stein, M.-K. (2023). Digital Work: A Conceptual Clarification, Proceedings of the 56th Hawaii International Conference on System Sciences, Honolulu: Hawaii International Conference on System Sciences (HICSS), pp. 4588–4597.

[6] Ballard, C., Abdel-Hamid, A., Frankus, R., Hasegawa, F., Larrechart, J., Leo, P. & Julio, J. (2006). Improving business performance insight-- with business intelligence and business process management, IBM Redbooks.

[7] Bazan, P. & Estevez, E. (2022). Industry 4.0 and business process management: state of the art and new challenges, Business Process Management Journal, Vol. 28(1), pp. 62–80.

[8] Belabbes, M. A., Ruthven, I., Moshfeghi, Y. & Rasmussen Pennington, D. (2023). Information overload: a concept analysis, Journal of Documentation, Vol. 79(1), pp. 144–159.

[9] Benner, M. J. & Tushman, M. L. (2003). Exploitation, Exploration, and Process Management: The Productivity Dilemma Revisited, The Academy of Management Review, Vol. 28(2), pp. 238–256.

[10] Brahma, M., Tripathi, S. S. & Sahay, A. (2021). Developing curriculum for industry 4.0: digital workplaces, Higher Education, Skills and Work - Based Learning, Vol. 11(1), pp. 144–163.

[11] Braun, A., Zweck, A. & Holtmannspötter, D. (2016). The ambiguity of intelligent algorithms: job killer or supporting assistant, European Journal of Futures Research, Vol. 4(1), pp. 1–8.

[12] Brocke, J. vom, Maaß, W., Buxmann, P., Maedche, A., Leimeister, J. M. & Pecht, G. (2018). Future Work and Enterprise Systems, Business & Information Systems Engineering, Vol. 60(4), pp. 357–366.

[13] Casas-Garriga, G. (2003). Discovering Unbounded Episodes in Sequential Data, Knowledge Discovery in Databases, Heidelberg: Springer Berlin Heidelberg. pp. 83–94.

[14] Coghlan, D. (2018). Conducting action research for business and management students, 1st ed., London: SAGE Publications Ltd.

[15] Dilmegani, C. (2023). Task Mining in ’23: What it is & How it works in 5 Steps, 9 May 2023. Available at: https://research.aimultiple.com/task-mining/ (Accessed 18 July 2023)

[16] Dong, G. & Pei, J. (2007). Sequence Data Mining, 1st ed. 2007., New York, NY: Springer US: Imprint: Springer.

[17] Dongnan, S. & Zhaopeng, Z. (2021). Parallel Design of Apriori Algorithm Based on the Method of “Determine Infrequent Items & Remove Infrequent Itemsets,” IOP Conference Series: Earth and Environmental Science, Vol. 634(1), p. 012065.

[18] Drucker, P. F. (1999). Knowledge-Worker Productivity: The Biggest Challenge, California Management Review, Vol. 41(2), pp. 79–94.

[19] Dubey, S. & Mishra, N. (2011). Web Page Prediction using Hybrid Model, International Journal on Computer Science and Engineering, (3), pp. 2170–2176.

[20] El-Gharib, N. M. & Amyot, D. (2023). Robotic process automation using process mining — A systematic literature review, Data & Knowledge Engineering, Vol. 148, p. 102229.

[21] Everest Group (2022). Task Mining Software Market Posts 85%-95% Growth Rate, Remains One Of The Fastest Growing Markets In The Intelligent Automation Space—Everest Group - Everest Group, 16 November 2022. Available at: https://www.everestgrp.com/press-releases/task-mining-software-market.html (Accessed 15 September 2023)

[22] Gillespie, M. (2020). Understanding Log Analytics at Scale, 1st edition, O’Reilly Media, Inc.

[23] Gurău, M. I. (2020). The Role of Accounting and Accountant in the Modern Economy, Global Economic Observer, Vol. 8(2), pp. 119–124.

[24] Han, J., Kamber, M. & Pei, J. (2012). Data mining: concepts and techniques. 3rd ed, Burlington, MA: Elsevier.

[25] Hand, D. J., Mannila, H. & Smyth, P. (2001). Principles of data mining, Cambridge, Mass: MIT Press.

[26] Heidary Dahooie, J., Afrazeh, A. & Mohammad Moathar Hosseini, S. (2011). An activity-based framework for quantification of knowledge work, Journal of Knowledge Management, Vol. 15(3), pp. 422–444.

[27] Holford, W. D. (2019). The future of human creative knowledge work within the digital economy, Futures: The Journal of Policy, Planning and Futures Studies, Vol. 105, pp. 143–154.

[28] IBM. What is a Workflow? Available at: https://www.ibm.com/topics/workflow (Accessed 16 July 2023)

[29] Kelemen, M. & Rumens, N. (2008). An Introduction to Critical Management Research, SAGE Publications, Ltd.

[30] Kersten, M. & Murphy, G. C. (2015). Reducing Friction for Knowledge Workers with Task Context, AI Magazine, Vol. 36(2), pp. 33–41.

[31] Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C. & Team, J. D. (2016). Jupyter Notebooks – a publishing format for reproducible computational workflows, Positioning and Power in Academic Publishing: Players, Agents and Agendas, IOS Press, pp. 87–90.

[32] Kokina, J. & Blanchette, S. (2019). Early evidence of digital labor in accounting: Innovation with Robotic Process Automation, International Journal of Accounting Information Systems, Vol. 35, p. 100431.

[33] Kreps, J. (2014). I [heart symbol] logs, First edition, Sebastopol, CA: O'Reilly Media.

[34] Kumar, P., Radha Krishna, P. & Raju, S. B. (Eds.) (2012). Pattern discovery using sequence data mining: applications and studies, Hershey, PA: Information Science Reference.

[35] Lacity, M. C. & Wilcocks, L. P. (2016). A new approach to automating services, MIT Sloan Management Review, Vol. 58(1), p. 41.

[36] Madakam, S., Holmukhe, R. M. & Jaiswal, D. K. (2019). The Future Digital Work Force: Robotic Process Automation (rpa), Journal of Information Systems and Technology Management: JISTEM, Vol. 16, pp. 1–17.

[37] Mannila, H., Toivonen, H. & Inkeri Verkamo, A. (1997). Discovery of Frequent Episodes in Event Sequences, Data Mining and Knowledge Discovery, Vol. 1(3), pp. 259–289.

[38] Maximiliane, W. & Uwe, W. (2018). Industry 4.0 – organizing routines or innovations?, VINE Journal of Information and Knowledge Management Systems, Vol. 48(2), pp. 238–254.

[39] McNiff, J. (2013). Action research: principles and practice, 3rd ed., Milton Park, Abingdon, Oxon: Routledge.

[40] Othman, Z. A. & Eljadi, E. E. (2011). Network anomaly detection tools based on association rules, Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, pp. 1–7.

[41] Palvalin, M. (2018). What matters for knowledge work productivity? Employee Relations, Vol. 41(1), pp. 209–227.

[42] Plesner, U., Justesen, L. & Glerup, C. (2018). The transformation of work in digitized public sector organizations, Journal of Organizational Change Management, Vol. 31(5), pp. 1176–1190.

[43] Raiker, R. (2020). Task Mining Extends ABBYY Process Intelligence with Desktop Analytics. 13 October 2020. Available at: https://www.abbyy.com

/blog/task-mining-extends-abbyy-process-intelligence-with-desktop-analytics/ (Accessed 18 July 2023)

[44] Reinkemeyer, L. (2020). Process mining in action: principles, use cases and outlook, Cham, Switzerland: Springer.

[45] Rinta-Kahila, T., Penttinen, E. & Lyytinen, K. (2021). Organizational transformation with intelligent automation: Case Nokia Software, Journal of Information Technology Teaching Cases, Vol. 11(2), pp. 101–109.

[46] Sampson, S. E. (2021). A Strategic Framework for Task Automation in Professional Services, Journal of Service Research: JSR, Vol. 24(1), pp. 122–140.

[47] SAP Insights. Workflow management: A beginner's guide to workflow automation and business process management. Available at: https://www.sap.com/products/technology-platform/workflow-management.html (Accessed 16 July 2023)

[48] Saunders, M., Lewis, P. & Thornhill, A. (2016). Research methods for business students, 7th edition, New York: Pearson Education.

[49] Schäl, T. (1998). Workflow management systems for process organisations, Second edition., Berlin; Springer.

[50] Schwarzmüller, T., Brosi, P., Duman, D. & Welpe, I. M. (2018). How Does the Digital Transformation Affect Organizations? Key Themes of Change in Work Design and Leadership, Management Revue, Vol. 29(2), pp. 114–138.

[51] Simperl, E., Thurlow, I., Warren, P., Dengler, F., Davies, J., Grobelnik, M., Mladenic, D., Gomez-Perez, J. M. & Moreno, C. R. (2010). Overcoming Information Overload in the Enterprise: The Active Approach, IEEE Internet Computing, Vol. 14(6), pp. 39–46.

[52] Somekh, B. (2005). Action Research A Methodology for Change and Development, Maidenhead: McGraw-Hill Education.

[53] Tatti, N. (2014). Discovering episodes with compact minimal windows, Data Mining and Knowledge Discovery, Vol. 28(4), pp. 1046–1077.

[54] Uto, M., Miyazawa, Y., Kato, Y., Nakajima, K. & Kuwata, H. (2020). Time- and Learner-Dependent Hidden Markov Model for Writing Process Analysis Using Keystroke Log Data, International Journal of Artificial Intelligence in Education, Vol. 30(2), pp. 271–298.

[55] van der Aalst, W., Adriansyah, A., de Medeiros, A. K. A., Arcieri, F., Baier, T., Blickle, T., Bose, J. C., van den Brand, P., Brandtjen, R., Buijs, J., Burattin, A., Carmona, J., Castellanos, M., Claes, J., Cook, J., Costantini, N., Curbera, F., Damiani, E., de Leoni, M., Delias, P., van Dongen, B. F., Dumas, M., Dustdar, S., Fahland, D., Ferreira, D. R., Gaaloul, W., van Geffen, F., Goel, S., Günther, C., Guzzo, A., Harmon, P., ter Hofstede, A., Hoogland, J., Ingvaldsen, J. E., Kato, K., Kuhn, R., Kumar, A., La Rosa, M., Maggi, F., Malerba, D., Mans, R. S., Manuel, A., McCreesh, M., Mello, P., Mendling, J., Montali, M., Motahari-Nezhad, H. R., zur Muehlen, M., Munoz-Gama, J., Pontieri, L., Ribeiro, J., Rozinat, A., Seguel Pérez, H., Seguel Pérez, R.,

Sepúlveda, M., Sinur, J., Soffer, P., Song, M., Sperduti, A., Stilo, G., Stoel, C., Swenson, K., Talamo, M., Tan, W., Turner, C., Vanthienen, J., Varvaressos, G., Verbeek, E., Verdonk, M., Vigo, R., Wang, J., Weber, B., Weidlich, M., Weijters, T., Wen, L., Westergaard, M. & Wynn, M. (2012). Process Mining Manifesto, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 169–194.

[56] van der Aalst, W. M. P. (2013a). Business Process Management: A Comprehensive Survey, ISRN Software Engineering, Vol. 2013, pp. 1–37.

[57] van der Aalst, W. M. P. (2013b). Decomposing Petri nets for process mining: a generic approach, Distributed and Parallel Databases: An International Journal, Vol. 31(4), pp. 471–507.

[58] van der Aalst, W. M. P. (2023). Object-Centric Process Mining: Unraveling the Fabric of Real Processes, Mathematics (Basel), Vol. 11(12), p. 2691.

[59] van der Aalst, W. M. P. & Carmona, J. (2022). Process Mining Handbook, 1st ed. 2022. [Online], Cham: Springer Nature.

[60] van der Aalst, W. M. P., La Rosa, M. & Santoro, F. M. (2016). Business process management: don't forget to improve the process, Business & Information Systems Engineering, Vol. 58(1), pp. 1–6.

[61] vom Brocke, J., Schmiedel, T., Recker, J., Trkman, P., Mertens, W. & Viaene, S. (2014). Ten principles of good business process management, Business Process Management Journal, Vol. 20(4), pp. 530–548.

[62] Vulpe, M.-I., Stancu, S., Elena, D., Pernici, A. & Constantin, I. (2022). Robotic Process Automation through Process Mining, Manager (Bucureşti), (36), pp. 7–16.

[63] Vuori, V., Helander, N. & Okkonen, J. (2019). Digitalization in knowledge work: the dream of enhanced performance, Cognition, Technology & Work, Vol. 21(2), pp. 237–252.

[64] Wang, B., Schlagwein, D., Cecez-Kecmanovic, D. & Cahalane, M. C. (2020). Editorial: Beyond the Factory Paradigm: Digital Nomadism and the Digital Future(s) of Knowledge Work Post-COVID-19, Journal of the Association for Information Systems, Vol. 21(6), p. 10.

[65] Wang, H., Zhang, G., Chen, H. & Jiang, X. (2009). Mining Association Rules for Intrusion Detection, Fourth International Conference on Frontier of Computer Science and Technology, [Online], 2009 IEEE. pp. 644–648.

[66] Wang, M. & Wang, H. (2006). From process logic to business logic—A cognitive approach to business process management, Information & Management, Vol. 43(2), pp. 179–193.

[67] Wang, M., Wang, H. & Xu, D. (2005). The design of intelligent workflow monitoring with agent technology, Knowledge-Based Systems, Vol. 18(6), pp. 257–266.

[68] Weilkiens, T., Weiss, C., Grass, A. & Nena Duggen, K. (2016). Basic Principles of Business Process Management, OCEB 2 Certification Guide, United States: Elsevier Science & Technology.

[69] Weske, M. (2012). Business Process Management Concepts, Languages, Architectures, 2nd ed. 2012., Berlin, Heidelberg: Springer Berlin Heidelberg.

[70] Willcocks, L. (2020). Robo-Apocalypse cancelled? Reframing the automation and future of work debate, Journal of Information Technology, Vol. 35(4), pp. 286–302.

[71] Zhan, F., Zhu, X., Zhang, L., Wang, X., Wang, L. & Liu, C. (2019). Summary of Association Rules, IOP Conference Series: Earth and Environmental Science, Vol. 252, p. 032219.