HAW HAMBURG

Bachelorarbeit

Sven Robin Gerbig

Aktueller Stand, Potenziale und Herausforderungen bei dem Einsatz von Process Discovery

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department Computer Science

Sven Robin Gerbig

Aktueller Stand, Potenziale und Herausforderungen bei dem Einsatz von Process Discovery

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Wirtschaftsinformatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Ulrike Steffens
Zweitgutachter: Prof. Dr. Rüdiger Weißbach

Eingereicht am: 13. Februar 2023

Sven Robin Gerbig

Thema der Arbeit

Aktueller Stand, Potenziale und Herausforderungen bei dem Einsatz von Process Discovery

Stichworte

Process Mining, Petrinetze, Process Science, Data Science, Potenziale und Herausforderungen, OCPM, RPA

Kurzzusammenfassung

Diese Thesis gibt einen umfassenden Überblick über den Bereich des Process Mining und untersucht insbesondere die aktuellen Potenziale und Herausforderungen von Process Discovery. Verschiedene Methoden und Techniken zur Prozessentdeckung werden analysiert, um eine aktuelle Übersicht zu schaffen, die es ermöglicht, Prozess Discovery effektiv und gewinnbringend einzusetzen. Der aktuelle Stand der Literatur wird untersucht, um eine Zusammenfassung der vorhandenen Potenziale und Herausforderungen von Process Discovery zu liefern und einen Überblick über die aktuellen Entwicklungen und Fortschritte im Bereich Process Mining zu geben. Anwendungsgebiete, Einsatzmöglichkeiten, Tools und Technologien werden betrachtet. Insgesamt gibt diese Thesis einen umfassenden Einblick in den Bereich des Process Mining und insbesondere in die Technik der Prozessentdeckung.

Sven Robin Gerbig

Title of Thesis

Current status, potentials and challenges in the use of Process Discovery

Keywords

Process Mining, Petrinets, Process Science, Data Science, potentials and challenges, OCPM, RPA

Abstract

The aim of this thesis is to provide a comprehensive overview of the field of Process Mining and to examine the current potentials and challenges of Process Discovery in particular. Various methods and techniques for process discovery are analyzed to create a current overview of the possibilities that allow for effective and profitable use of Process Discovery. The current state of literature is examined to provide a summary of the existing potentials and challenges of Process Discovery and to provide an overview of the current developments and advancements in the field of Process Mining. Applications, usage possibilities, tools, and technologies are considered. Overall, this thesis provides a comprehensive insight into the field of Process Mining and particularly into the technique of Process Discovery.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Relevanz in der Praxis

In der digitalen Ära sind Daten eine wichtige Ressource für Unternehmen jeglicher Größe. Sie werden genutzt, um Innovationen zu unterstützen, Geschäftsentscheidungen zu treffen und die Betriebseffektivität zu verbessern. Ähnlich wie Öl eine wichtige Ressource für die industrielle Wirtschaft ist, hat der Ausdruck „Daten sind das neue Öl“ an Bedeutung gewonnen, um den wachsenden Wert von Daten als Ressource zu unterstreichen (Nolin, 2019).

Der digitale Wandel ermöglicht es, Daten jederzeit und überall zu sammeln, was durch die fortschreitende Technologie und den immer billigeren Speicherplatz möglich wird. Dies führt dazu, dass Unternehmen große Mengen an unstrukturierten Daten haben, die es zu nutzen gilt. Die digitale Transformation hat dazu geführt, dass von allem, fast überall und jederzeit Daten erhoben werden. Jede Transaktion in einem Informationssystem führt dazu, dass Daten innerhalb eines Unternehmens erzeugt werden (van der Aalst, 2016). Die weltweite Datenmenge nimmt jährlich um etwa 30% zu und soll bis 2025 voraussichtlich auf 175 Zettabyte (175 Milliarden Terabyte) anwachsen (Lázaro u. a., 2022).

Process Mining und insbesondere Process Discovery, haben mit ihrem Ziel, Daten zu analysieren, um Geschäftsprozesse zu verstehen und zu verbessern, in dieser Situation viel Potenzial. Organisationen können durch die Auswertung von großen Datensätzen der tatsächlichen Durchführung eines Prozesses, Engpässe, Ineffizienzen und Optimierungspotenzial erkennen. Dadurch können Unternehmen die Betriebsabläufe verbessern, einen Wettbewerbsvorteil erlangen und ggf. Innovationen schaffen.

1.2 Gliederung der Bachelorthesis

Die Arbeit beschäftigt sich mit dem Thema Process Mining. Speziell mit dem Unterthema Process Discovery und den aktuellen Potenzialen und Herausforderungen. Die Thesis ist in zwei Teile gegliedert: einen grundlegenden Theorieteil und einen Hauptteil.

Im ersten Teil wird zunächst eine allgemeine Einordnung von Process Mining vorgenommen und anschließend auf die verschiedenen Arten von Process Mining eingegangen. Wobei sich diese Arbeit hauptsächlich mit dem Thema Process Discovery beschäftigt. Im Anschluss wird das wichtige Thema der Datengrundlage behandelt. Hier wird auf die Datenbeschaffung eingegangen, die darauf abzielt, die nötigen Daten für das Process Mining zu erfassen. Eventlogs, in denen alle relevanten Daten eines Geschäftsprozesses aufgezeichnet sind, werden eingeführt. Auch die Datenqualität wird angesprochen, da sie für die Genauigkeit und Zuverlässigkeit der Prozessanalyse von großer Bedeutung ist.

Als Nächstes werden Petrinetze als Form der Visualisierung von Prozessmodellen vorgestellt. Petrinetze sind grafische Darstellungen von Prozessen. Sie können verwendet werden, um den Ablauf von Prozessen zu veranschaulichen und zu analysieren. Danach wird der Alpha-Algorithmus vorgestellt, der das Konzept der Modellerstellung sehr gut verdeutlicht. Der Alpha-Algorithmus ist ein Verfahren, das auf Basis von Eventlogs Petrinetze erstellt. Allerdings gibt es auch Einschränkungen bei der Verwendung des Alpha-Algorithmus, die ebenfalls behandelt werden.

Im Kern der Arbeit werden die aktuellen Herausforderungen und Potenziale beim Einsatz von Process Discovery aufgezeigt. Der Aufbau der untersuchten Themen gliedert sich in mehrere Phasen. Zunächst wird das grundlegende Problem beleuchtet, anschließend erfolgt eine tiefgründige Theorieeinführung in das Problem sowie in den Lösungsansatz. In der finalen Phase wird das Augenmerk auf die Umsetzung der Lösung für die beschriebene Herausforderung gerichtet und existierende Softwarelösungen vorgestellt.

Zum Schluss dieser Thesis werden die wichtigsten Erkenntnisse aus dieser Arbeit zusammengefasst und ein Ausblick für zukünftige Arbeiten gegeben.

1.3 Ziel und Motivation der Bachelorarbeit

Die Motivation dieser Bachelorarbeit hat den Ursprung in der Teilnahme an dem Wahlpflichtkurs Process Intelligence von Prof. Dr. Steffens im fünften Semester. Erste Erfahrungen im Bereich Process Mining konnten dort gesammelt werden und seither wird dieses Thema sehr interessant empfunden. Daher soll diese Bachelorarbeit genutzt werden, um die Kenntnisse in diesem Gebiet zu vertiefen. Außerdem hat dieses Thema in den letzten Jahren in der Unternehmenswelt und in der Literatur immer mehr an Bedeutung gewonnen.

Die Zielsetzung dieser Thesis ist es, einen umfassenden Überblick über den Bereich des Process Mining zu vermitteln und darauf aufbauend die aktuellen Potenziale und Herausforderungen von Process Discovery näher zu untersuchen. Um dies zu erreichen, werden verschiedene Methoden und Techniken zur Prozessentdeckung analysiert. Dabei wird das Ziel verfolgt, die unterschiedlichen Potenziale aufzuzeigen und Handlungsalternativen für einen wirtschaftlichen Einsatz in der Praxis von Process Discovery abzuleiten.

Durch die Untersuchung des aktuellen Standes der Literatur soll diese Thesis eine Zusammenfassung der vorhandenen Potenziale und Herausforderungen von Process Discovery darstellen und einen Überblick über die aktuellen Entwicklungen und Fortschritte im Bereich Process Mining liefern. Dies beinhaltet auch die Betrachtung von verschiedenen Anwendungsgebieten und Einsatzmöglichkeiten von Process Discovery, sowie die Analyse von verschiedenen Tools und Technologien, die in diesem Bereich verwendet werden.

Insgesamt soll diese Thesis einen umfassenden Einblick in den Bereich des Process Mining und insbesondere in die Technik der Prozessentdeckung geben und den Leser auf die Möglichkeiten und Herausforderungen, die mit der Anwendung von Process Discovery einhergehen, aufmerksam machen.

2 Grundlagen Process Mining

2.1 Theoretische Grundlage von Process Mining

In diesem Kapitel werden die Grundlagen von Process Mining vorgestellt, um ein Fundament für die im Hauptteil vorgestellten Potenziale zu schaffen. Zunächst wird eine allgemeine Einordnung von Process Mining vorgenommen, danach wird auf die zentrale Datengrundlage eingegangen. Schließlich wird der Teilbereich Process Discovery behandelt, wo ein Discovery-Algorithmus näher präsentiert und die technische Repräsentation der Modelle vorgestellt werden. Auch die Stärken und Schwächen der verschiedenen Algorithmen werden betrachtet.

2.1.1 Einordnung Data Science

Angetrieben durch das stetige Wachstum der zur Verfügung stehenden Daten, hat sich eine neue Datenwissenschaft entwickelt. Eine interdisziplinäre Wissenschaft aus verschiedenen Bereichen. Im Kern geht es darum, aus Daten Wissen zu erlangen. In (van der Aalst, 2016, S.10) wird folgendes als allgemeine Definition von Data Science aufgeführt:

> "Data science is an interdisciplinary field aiming to turn data into real value. Data may be structured or unstructured, big or small, static or streaming. Value may be provided in the form of predictions, automated decisions,models learned from data, or any type of data visualization delivering insights. Data science includes data extraction, data preparation, data exploration, data transformation, storage and retrieval, computing infrastructures, various types of mining and learning, presentation of explanations and predictions, and the exploitation of results taking into account ethical, social, legal, and business aspects."

Aus der obigen Definition lassen sich für Data Science vier Hauptfragestellungen ableiten, die ein Data Scientist beantworten kann:

- (Reporting) Was ist passiert?
- (Diagnose) Warum ist es passiert?
- (Vorhersage) Was wird passieren?
- (Empfehlung) Was ist das Beste, was passieren kann?

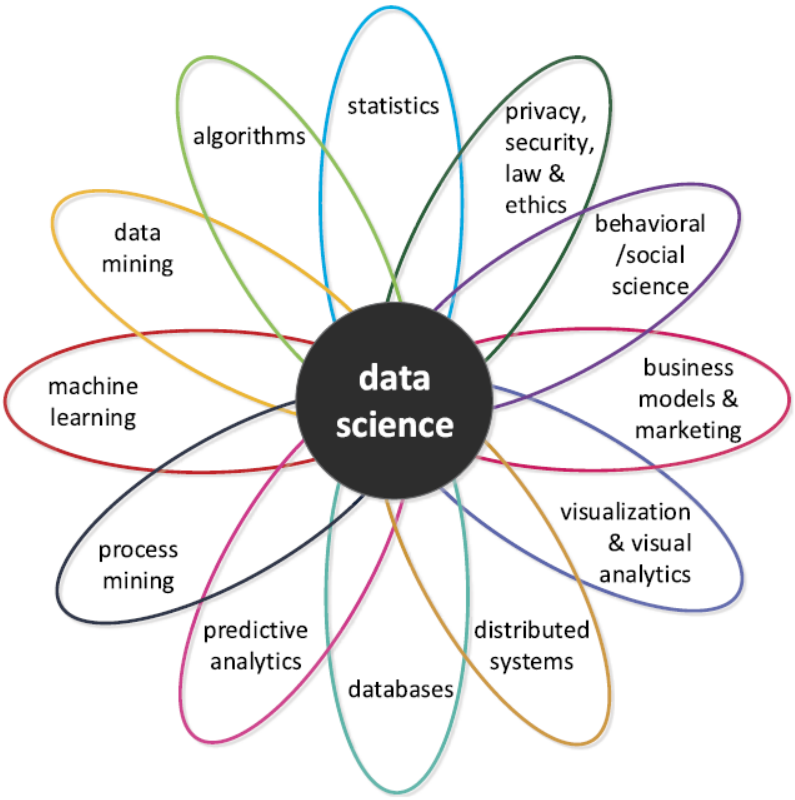

Abbildung 2.1: Die Bestandteile von Data Science (van der Aalst, 2016)

Wie in Abbildung 2.1 zu sehen, vereinigt Data Science viele verschiedene, sich überschneidende, Unterdisziplinen. Ebenfalls sind die Gebiete Process Mining, Machine Learning und Datenbanken Teil von Data Science. Process Mining fügt Machine Learning und Data Mining eine Prozessperspektive hinzu. Es geht darum, Eventdaten mit Prozessmodellen zu vergleichen. Diese Eventdaten sind mit expliziten Prozessmodellen wie Petrinetzen

oder BPMN-Modellen verbunden. Beispielsweise werden Prozessmodelle aus Eventdaten entdeckt oder Eventdaten werden auf Modelle repliziert, um die Einhaltung und Leistung zu analysieren (van der Aalst, 2016).

2.1.2 Einordnung Process Science und Process Mining

Die Einordnung von Process Mining in den Überbegriff Data Science, wie in Abbildung 2.1 zu sehen, wird nicht von allen geteilt. Genauer genommen wird die Wissenschaft, welche sich mit prozessgetriebenen, anstatt mit datengetriebenen Problemen beschäftigt, als Process Science bezeichnet. Van der Aalst beschreibt Process Science als breiter angelegte Disziplin, welche Wissen aus der Informationstechnologie und Managementwissenschaft vereint, um betriebliche Prozesse zu verbessern (van der Aalst, 2016). Die Abbildung 2.2 zeigt die Disziplinen von Process Science auf. Die Teilgebiete sind nicht starr voneinander getrennt, sondern haben ebenfalls Überschneidungen.

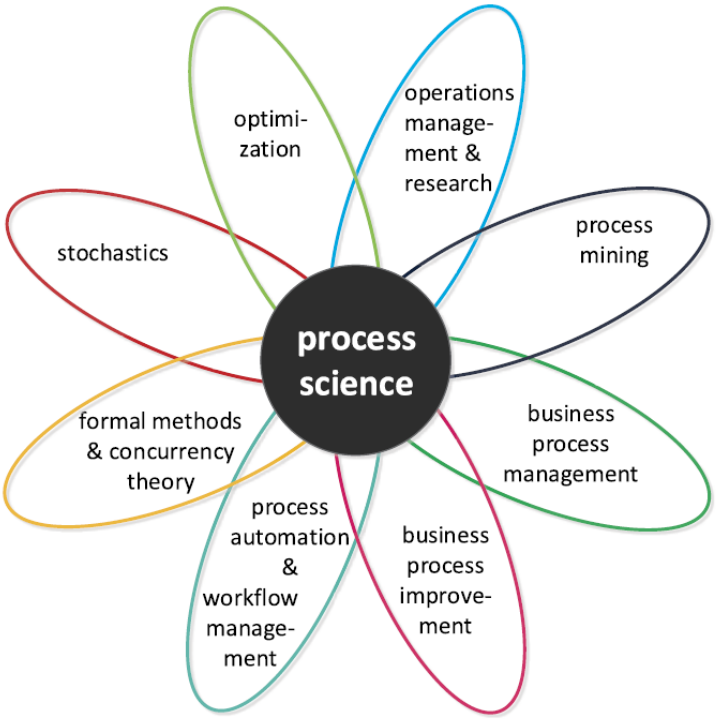

Abbildung 2.2: Die Bestandteile von Process Science (van der Aalst, 2016)

Process Mining kann nun als verbindendes Element zwischen Process Science und Data Science gesehen werden. Es nutzt die Ansätze aus beiden Wissenschaften, siehe Abbildung 2.3. Beim Process Mining geht es darum, beobachtbares Verhalten oder Eventdaten mit Prozessmodellen zu verbinden, die entweder von Menschen erstellt oder automatisch mittels Algorithmen erstellt wurden (van der Aalst, 2016).

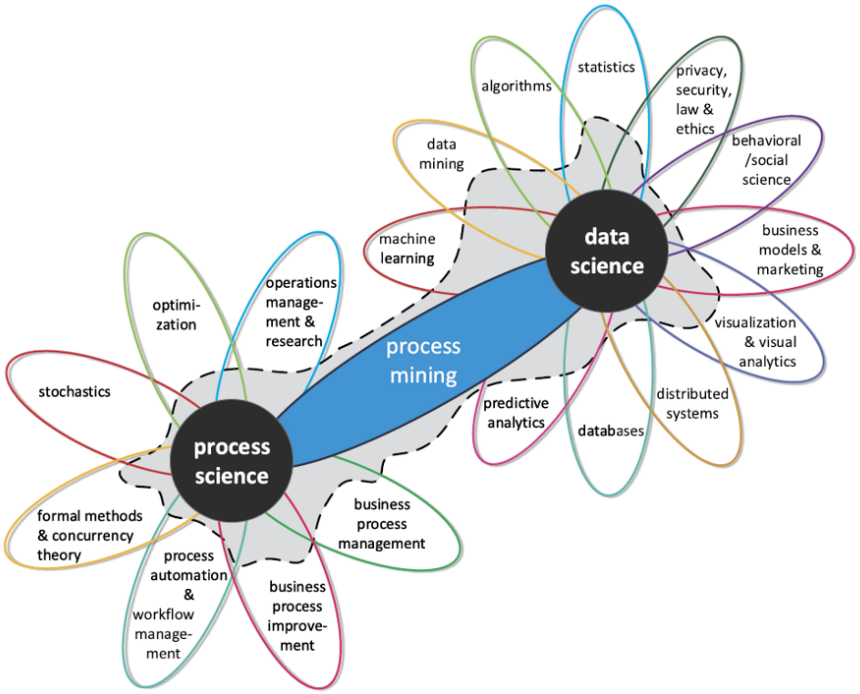

Abbildung 2.3: Process Mining als Verbindung zwischen Data Science und Process Science (van der Aalst, 2016)

2.1.3 Arten von Process Mining

Process Mining lässt sich generell in drei verschiedene Hauptarten aufteilen, wie in Abbildung 2.4 zu erkennen.

- Process Discovery Techniken beschäftigten sich mit der Erstellung eines Prozessmodells auf Basis eines Eventlogs.
- Process Conformance vergleicht ein bestehendes Prozessmodell (SOLL-Modell) mit einem Eventlog (IST-Prozess) desselben Prozesses, um beispielsweise die Einhaltung von vorgegebenen Arbeitsabläufen zu überprüfen.

- Process Enhancement kann vorhandene Prozessmodelle mit gewonnenen Informationen aus Eventlogs anpassen. Ziel ist es, eine Optimierung des Prozessmodells zu erlangen, um beispielsweise mögliche Engpässe zu beseitigen.

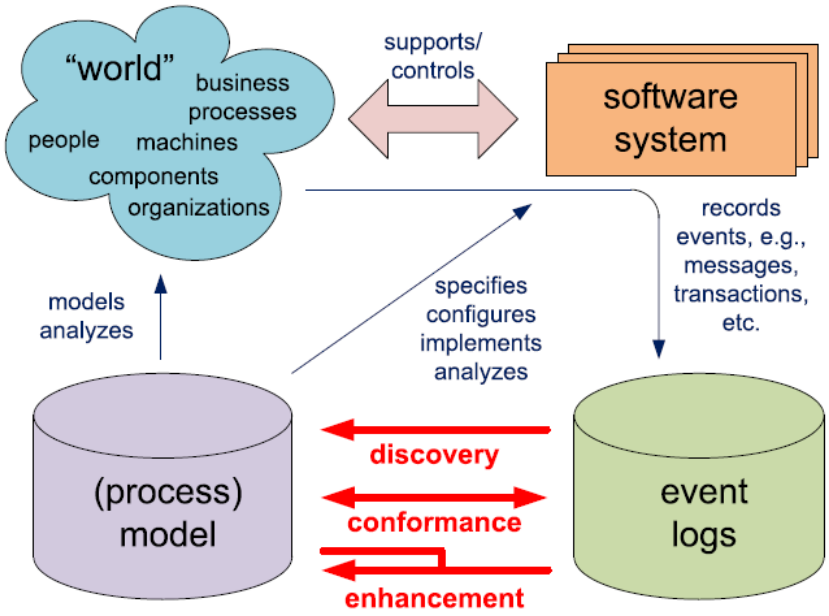

Abbildung 2.4: Schematische Übersicht von Process Mining (van der Aalst, 2016)

In dieser Arbeit wird hauptsächlich der Bereich Process Discovery untersucht. Die Anwendung von Process Mining kann aus verschiedenen Perspektiven betrachtet werden. Je nachdem, welche Fragen beantwortet werden sollen. In (van der Aalst, 2016) werden vier verschiedene Perspektiven beschrieben:

- Die Kontrollperspektive konzentriert sich auf den Kontrollfluss, d.h. die Reihenfolge der Aktivitäten. Ziel ist es, eine Charakterisierung der möglichen Pfade zu erhalten. Oftmals ist dies der Anwendungsbereich, der zuerst mit Process Mining assoziiert wird.

- Die Organisationsperspektive beschäftigt sich mit den Ressourceninformationen und mit der Zuordnung der Akteure (z.B. Menschen, Systeme, Rollen und Abteilungen). Das Ziel ist es, die Zuordnung von Rollen oder organisatorischen Einheiten für die Mitarbeiter zu bestimmen und die Beziehungen zwischen Ressourcen in Form eines sozialen Netzwerks darzustellen.

- Die Fallperspektive betrachtet die Eigenschaften einzelner Fälle, um beispielsweise ein bestimmtes Verhalten aufzudecken.
- Die Zeitperspektive befasst sich mit dem Zeitpunkt, der Zeitdauer und der Häufigkeit von Ereignissen. Ziel hierbei ist es, Engpässe zu entdecken oder die Auslastung von Ressourcen zu tracken.

2.2 Datengrundlage

Process Mining nutzt Daten aus den Systemen, die im Unternehmen zur Abwicklung von Geschäftsprozessen verwendet werden. Dies können Daten aus verschiedenen Quellen wie ERP-Systeme, Workflow-Management-Systeme oder E-Mail-Systeme sein. Die genaue Art der Datengrundlage hängt von der spezifischen Anwendung von Process Mining und dem verwendeten Tool ab. Process Mining hat wie jeder datengesteuerte Ansatz das Problem, dass die Datenqualität über Erfolg und Misserfolg entscheidet (Batini und Scannapieco, 2016).

2.2.1 Datenbeschaffung

Die Datenbeschaffung ist eines der wichtigsten Schritte im Process Mining. Die relevanten Daten sind in der Regel auf etliche Datenquellen verteilt. Daten können unstrukturiert in Excel Dokumenten, PDF Dokumenten oder E-Mails vorliegen. Eine typische SAP Implementation enthält mehr als 10.000 Tabellen, somit sind die Daten auch innerhalb einer Datenbank breit gestreut. So werden heute ca. 80% der Gesamtzeit für die Datenbeschaffung aufgewendet und lediglich 20% für die Datenanalyse.

Ein strukturiertes Verfahren für die Extrahierung und Bereitstellung von Daten bietet der Extract-Transform-Load (ETL) Prozess. ETL ist die Abkürzung für Extract-Transform-Load und bezeichnet einen Prozess zur Datenaufbereitung. In einem ETL-Prozess werden Daten aus verschiedenen Quellen extrahiert, in einem bestimmten Format transformiert und anschließend in ein Data Warehouse geladen. Der ETL-Prozess ist ein wichtiger Schritt bei der Integration von Daten aus verschiedenen Systemen und Datenbanken. Ein ETL-Prozess besteht aus drei Hauptphasen:

Die **Extraktionsphase**: In dieser Phase werden die Daten aus den verschiedenen Quellen extrahiert und in einem geeigneten Format bereitgestellt. Dabei werden die Daten aus den verschiedenen Systemen und Datenbanken identifiziert und abgefragt.

Die **Transformationsphase**: In dieser Phase werden die extrahierten Daten verarbeitet und in ein geeignetes Format für die weitere Verwendung transformiert. Dabei können Schritte wie das Bereinigen und Anreichern der Daten, das Verknüpfen von Datensätzen oder das Konvertieren von Daten in ein anderes Format durchgeführt werden.

Die **Ladephase**: In dieser Phase werden die transformierten Daten z.B. in ein Data Warehouse geladen, von wo aus sie für weitere Analysen und Verarbeitungen zur Verfügung stehen (Ponniah, 2011).

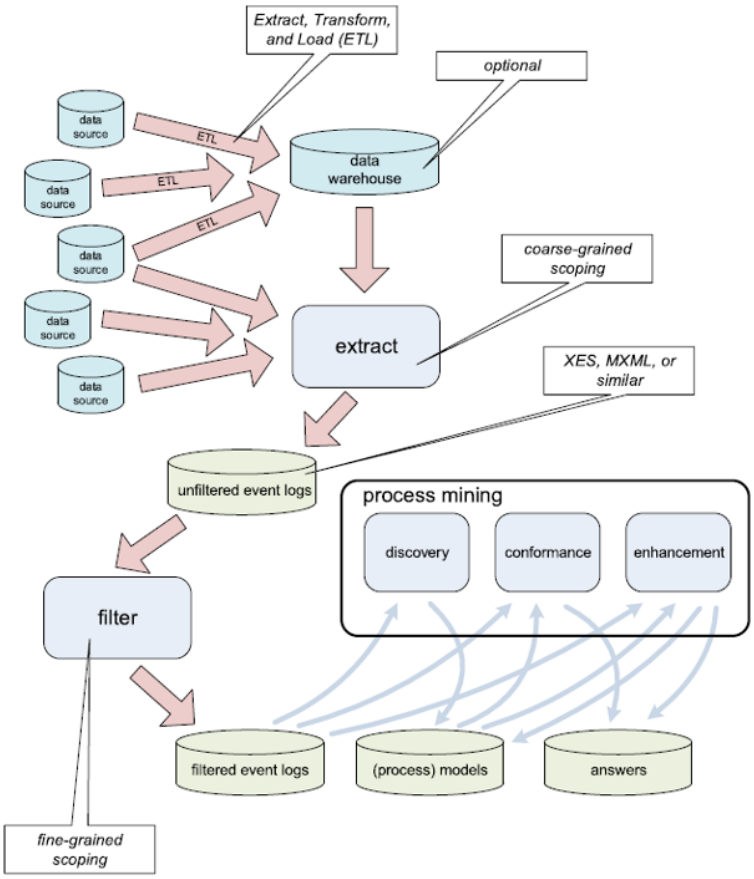

Abbildung 2.5: Schematische Übersicht von der Datenbeschaffung für Process Mining (van der Aalst, 2016)

Der schematische Ablauf der Datenbeschaffung ist in der Abbildung 2.5 verdeutlicht. Der Einsatz eines Data Warehouse ist in der Datenbeschaffung nicht zwingend notwendig. Es ist lediglich ein Ansatz, der es erleichtert, Daten aus verschiedenen Quellen zu vereinheitlichen. Vielmehr ist es von Relevanz, die Daten in einen Eventlog umwandeln zu können.

2.2.2 Eventlogs

In 2010 wurde durch die *IEEE Task Force on Process Mining* das eXtensible Event Stream (XES) Format für die Speicherung von Eventdaten eingeführt.
XES ist ein offener Standard zur Speicherung von Informationen über Geschäftsprozesse. XES ist ein XML-basiertes Format, das die Eigenschaften und Attribute von Ereignissen in Geschäftsprozessen speichert. XES wird häufig zur Speicherung von Daten aus Process Mining Anwendungen verwendet und trägt zur Interoperabilität von Tools und Anwendungen in diesem Bereich bei (Günther, 2009).

Die grundlegende Struktur sowie die Assoziationen und Kardinalitäten eines Eventlogs werden in Abbildung 2.6 näher aufgeführt. Der Aufbau des Modells lässt sich durch zehn verschiedene Eigenschaften beschreiben.

1. (a) Jeder Prozess kann eine beliebige Anzahl von Aktivitäten haben, aber jede Aktivität gehört genau einem Prozess.
2. (b) Jeder Fall gehört genau einem Prozess.
3. (c) Jede Aktivitätsinstanz bezieht sich genau auf eine Aktivität.
4. (d) Jede Aktivitätsinstanz gehört genau einem Fall; es können mehrere Aktivitätsinstanzen für jede Aktivität/Fall-Kombination vorliegen.
5. (e) Jedes Ereignis bezieht sich genau auf einen Fall.
6. (f) Jedes Ereignis entspricht einer Aktivitätsinstanz; für die gleiche Aktivitätsinstanz können mehrere Ereignisse vorliegen.
7. (g) Jedes Fallattribut bezieht sich auf einen Fall und hat einen Namen und einen Wert, z.B. (Geburtsdatum, 25.10.1996).
8. (h) Jedes Ereignisattribut bezieht sich auf ein Ereignis und ist durch einen Namen und einen entsprechenden Wert charakterisiert, z.B. (Kosten, 300€)

9. (i) Es gibt verschiedene Unterklassen von Fallattributen, z.B. die Beschreibung eines Falls, Fallidentifikator, Startzeit des Falls usw..

10. (j-n) Es gibt verschiedene Unterklassen von Ereignisattributen, z.B. die Zeit des Ereignisses, die Position in der Spur oder der Transaktionstyp. Mindestens benötigt das Ereignis drei Attribute, den Fall, die Aktivität und ein Zeitattribut.

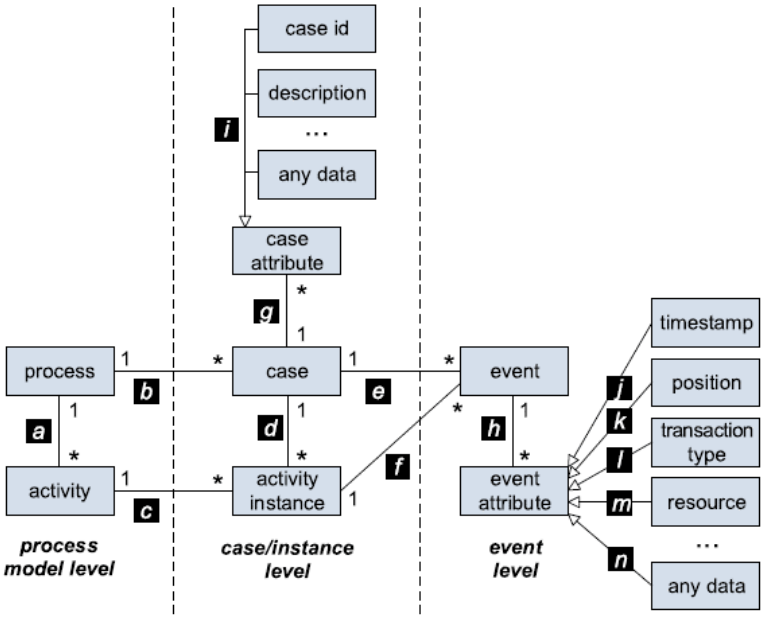

Abbildung 2.6: Klassendiagramm Eventlog (van der Aalst, 2016)

Die Tabelle 2.1 enthält Daten aus einem Beispiel Eventlog. Diese enthält sechs Spalten, die jeweils Informationen über einen bestimmten Aspekt des Eventlogs enthalten. In den Spalten werden Informationen über Case id, Event id, Timestamp, Activity, Resource und Cost dargestellt, wobei jede Zeile ein Ereignis im Eventlog darstellt. Bei großen Datenmengen kann eine solche Ansicht eines Eventlogs schnell unübersichtlich werden. Aus diesem Grund gibt es eine simplifizierte Darstellung mittels Simple Eventlogs.
Ein Simple Eventlog ist eine spezielle Form eines Eventlogs, der nur die Sequenzen von Aktivitäten enthält, ohne zusätzliche Attribute wie Zeitstempel oder Ressourceninformationen. Ein Simple Eventlog besteht aus einem Multi Set von Traces, wobei ein Trace eine Sequenz von Aktivitäten ist. Ein Beispiel für ein Simple Eventlog, welcher aus 6

Cases und 23 Events besteht, wäre:

$$[(a, b, c, d)^3, (a, c, b, d)^2, (a, e, d)^1]$$

Simple Eventlogs können bei großen Mengen von Cases nützlich sein, da sie übersichtlicher sind als komplexere Eventlogs (van der Aalst, 2016).

Tabelle 2.1: Auszug eines Beispiel Eventlogs (van der Aalst, 2016)

Case id	Event id	Timestamp	Activity	Resource	Cost	...
1	35654423	30-12-2010:11.02	register request	Pete	50	...
1	35654424	31-12-2010:10.06	examine thoroughly	Sue	400	...
1	35654425	05-01-2011:15.12	check ticket	Mike	100	...
1	35654426	06-01-2011:11.18	decide	Sara	200	...
1	35654427	07-01-2011:14.24	reject request	Pete	200	...
2	35654483	30-12-2010:11.32	register request	Mike	50	...
2	35654485	30-12-2010:12.12	check ticket	Mike	100	...
2	35654487	30-12-2010:14.16	examine casually	Pete	400	...
2	35654488	05-01-2011:11.22	decide	Sara	200	...
2	35654489	08-01-2011:12.05	pay compensation	Ellen	200	...
3	35654521	30-12-2010:14.32	register request	Pete	50	...
3	35654522	30-12-2010:15.06	examine casually	Mike	400	...
3	35654524	30-12-2010:16.34	check ticket	Ellen	100	...
3	35654525	06-01-2011:09.18	decide	Sara	200	...
3	35654526	06-01-2011:12.18	reinitiate request	Sara	200	...
3	35654527	06-01-2011:13.06	examine thoroughly	Sean	400	...
3	35654530	08-01-2011:11.43	check ticket	Pete	100	...
3	35654531	09-01-2011:09.55	decide	Sara	200	...
3	35654533	15-01-2011:10.45	pay compensation	Ellen	200	...
4	35654641	06-01-2011:15.02	register request	Pete	50	...
4	35654643	07-01-2011:12.06	check ticket	Mike	100	...
4	35654644	08-01-2011:14.43	examine thoroughly	Sean	400	...
4	35654645	09-01-2011:12.02	decide	Sara	200	...
4	35654647	12-01-2011:15.44	reject request	Ellen	200	...
...	...	...	...	...	...	...

2.2.3 Datenqualität von Eventdaten

Die Qualität der Daten in Eventlogs ist wichtig, da sie die Genauigkeit und Zuverlässigkeit der auf den Eventlogs basierenden Analysen und Entscheidungen beeinflusst. Ein Eventlog mit fehlerhaften oder unvollständigen Daten kann dazu führen, dass die Ergebnisse der Analysen falsch sind oder wichtige Informationen fehlen, was zu ungenauen Entscheidungen führen kann. Aus diesem Grund ist es wichtig, dass die Daten in Eventlogs möglichst genau und vollständig sind, um sicherzustellen, dass die Analysen und Entscheidungen, die auf den Eventlogs basieren, zuverlässig und nützlich sind. Aus diesem Grund wurden 12 Logging Guidelines definiert, welche die Datenqualität sichern sollen (van der Aalst, 2015). Der Inhalt der genannten Guideline for Logging besagt, dass Eventlogs sorgfältig organisiert und verwaltet werden sollten, um ihre Genauigkeit und Nutzbarkeit zu gewährleisten. Dabei sollten Referenzen und Attribute klar definiert sein und die Eventlogs sollten stabil und präzise sein. Regelmäßige Prüfungen sollten durchgeführt werden und die Vergleichbarkeit von Event Protokollen im Laufe der Zeit und bei unterschiedlichen Gruppen von Fällen oder Prozessvarianten sollte sichergestellt werden. Ereignisse sollten nicht aggregiert oder entfernt werden und die Nachvollziehbarkeit und der Datenschutz sollten gewahrt bleiben.

2.3 Process Discovery Algorithmen

Es wurde bis jetzt herausgearbeitet, dass Process Mining ursprünglich als Teilbereich von Data Science betrachtet wurde. Mit der weiteren Entwicklung wurde Process Mining in das neue Forschungsgebiet Process Science eingebettet. Die Grundelemente von Process Mining wurden vorgestellt und die Datengrundlage näher betrachtet.

Im Rahmen dieses Abschnitts wird die Verwendung von Process Discovery Algorithmen und deren Anwendung bei der Erstellung von Prozessmodellen betrachtet. Ein wichtiger Aspekt bei der Prozessmodellierung ist die Verwendung von Petrinetzen als formale Basis. Petrinetze sind ein wichtiges Werkzeug zur Modellierung von parallelen und sequentiellen Prozessen und werden häufig in der Process Discovery eingesetzt. Im Rahmen dieses Abschnitts wird erläutert, wie Petrinetze zur Modellierung von Prozessen verwendet werden können und wie sie als Grundlage für Algorithmen der Prozessentdeckung dienen können.

Process Discovery ist eine der anspruchsvollsten Aufgaben im Process Mining. Basierend auf einem Eventlog wird ein Prozessmodell erstellt, welches das in dem Eventlog beobachtete Verhalten erfasst. Dieses Kapitel stellt das Thema mit dem eher naiven α-Algorithmus vor. Dieser Algorithmus illustriert einige der allgemeinen Ideen, die von vielen Process Mining Algorithmen verwendet werden, und hilft dabei, das Konzept der Process Discovery zu verstehen.

2.3.1 Petrinetze

Ein Petrinetz ist ein formales Modellierungswerkzeug, das zur Darstellung und Analyse von parallelen und sequentiellen Prozessen verwendet wird. Es wurde von Carl Adam Petri in den 1960er Jahren entwickelt und besteht aus zwei Hauptkomponenten: Stellen, die als Orte für Ressourcen in einem Prozess dienen und die Zustände beschreiben. Transitionen beschreiben mögliche Übergänge zwischen den durch die Stellen repräsentierten Zuständen.

Ein Petrinetz ist formal definiert als ein Triplet $N = (P, T, F)$, wobei P eine endliche Menge von Stellen ist, T eine endliche Menge von Übergängen, $P \cap T = \emptyset$ und F eine Teilmenge von $(P \times T) \cup (T \times P)$ ist, die gerichtete Kanten darstellt. Diese Definition beschreibt die grundlegenden Komponenten eines Petrinetzes und die Beziehungen zwischen ihnen. Ein markiertes Petrinetz ist ein Petrinetz, das zusätzlich eine spezielle Markierung hat, die anzeigt, welche Stellen im Netz aktiv sind (van der Aalst, 2016).

Stellen werden als Kreise dargestellt und enthalten Marken, die den Zustand des Netzes definieren. Transitionen, die durch Rechtecke dargestellt werden, repräsentieren Ereignisse, die im modellierten System auftreten können. Um eine Transition auszulösen, müssen alle Vorgänger mindestens eine Marke besitzen. Wenn eine Transition ausgelöst wird, wird eine Marke von den Vorgängern genommen und eine Marke den Nachfolgern hinzugefügt. Dieser Vorgang wird als „Feuern“ einer Transition bezeichnet. Besitzt eine Transition keine Eingabestelle, so kann sie jederzeit feuern. Die Stellen begrenzen die möglichen Zustände des Netzes, da die Marken in den entsprechenden Stellen den Zustand definieren (Priese und Wimmel, 2008) (Reisig, 2010).

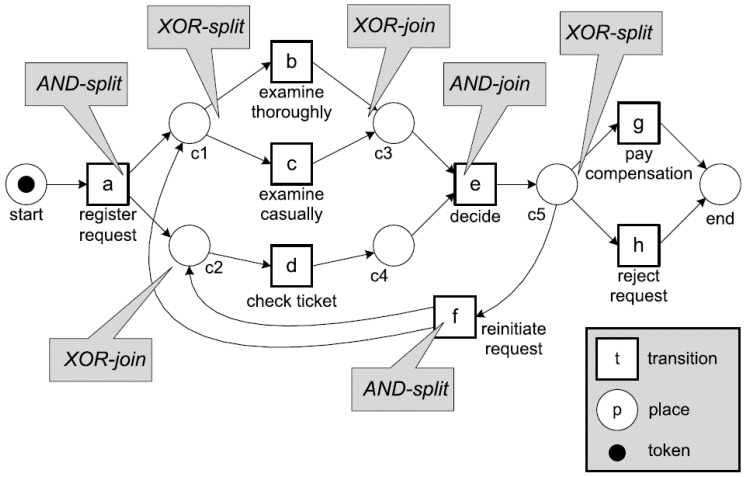

Abbildung 2.7: Markiertes Petrinetz (van der Aalst, 2016)

In der Abbildung 2.7 ist ein Beispiel für ein markiertes Petri Netz zu sehen. Das gezeigte Petri Netz kann formal wie folgt definiert werden:

- M_0 = {start}
- P = {start, c1, c2, c3,c4, c5, end}
- T = {a, b, c, d, e, f, g,h}
- F = {(start, a), (a, c1), (a, c2), (c1, b),(c1, c), (c2, d), (b, c3), (c, c3), (d, c4),(c3, e), (c4, e), (e, c5), (c5,f), (f, c1),(f, c2), (c5, g), (c5,h), (g, end), (h, end)}

2.3.2 α-Algorithmus

Der α-Algorithmus war einer der ersten Process Discovery Algorithmen, der mit Parallelität umgehen konnte. Dennoch hat der Algorithmus einige Schwächen, die später in Abschnitt 2.3.3 näher erläutert werden. Der α-Algorithmus ist eine einfache Technik, die in andere Methoden integriert wurde. Trotzdem wird er häufig als Ausgangspunkt für Diskussionen über Herausforderungen in Bezug auf die Prozessentdeckung und die Entwicklung anderer Algorithmen verwendet. Das Konzept hinter dem α-Algorithmus besteht darin, Eventlogs auf gewisse Muster zu durchsuchen.
Van der Aalst beschreibt die Mustererkennung in (van der Aalst, 2016, S.168) wie folgt:

"Wenn zum Beispiel auf die Aktivität a von b gefolgt wird, aber b nie von a gefolgt wird, dann wird angenommen, dass eine kausale Abhängigkeit zwischen a und b besteht. Um diese Abhängigkeit widerzuspiegeln, sollte das entsprechende Petrinetz eine Stelle haben, die a mit b verbindet. Wir unterscheiden vier protokollbasierte Ordnungsbeziehungen, die darauf abzielen, relevante Muster im Protokoll zu erfassen."

Definition 6.3 (Log-based ordering relations) Let L be an event log over $\mathscr{A}$, i.e., $L \in \mathbb{B}(\mathscr{A}^*)$. Let $a, b \in \mathscr{A}$.

- $a >_L b$ if and only if there is a trace $\sigma = \langle t_1, t_2, t_3, \ldots, t_n \rangle$ and $i \in \{1, \ldots, n-1\}$ such that $\sigma \in L$ and $t_i = a$ and $t_{i+1} = b$;
- $a \rightarrow_L b$ if and only if $a >_L b$ and $b \not>_L a$;
- $a \#_L b$ if and only if $a \not>_L b$ and $b \not>_L a$; and
- $a \|_L b$ if and only if $a >_L b$ and $b >_L a$.

Abbildung 2.8: Definition der Eventlog basierten Ordnungsrelationen (van der Aalst, 2016)

In der Definition 2.8 sind vier Ordnungsrelationen aufgezeigt und können sprachlich wie folgt beschrieben werden.

1. Ordnungsrelation: $>L_n$, beinhaltet alle Paare von Aktivitäten, die direkt aufeinander folgen.

2. Ordnungsrelation: $\rightarrow L_n$, beinhaltet alle Paare in einer Kausalitätsrelation. Sei x$\rightarrow$y, gdw. x$>$y und nicht y$>$x.

3. Ordnungsrelation: $\#L_n$, beinhaltet alle Paare, die keine direkte Verbindung untereinander haben. Sei x#y, gdw. nicht x$>$y und nicht y$>$x und x$\neq$y

4. Ordnungsrelation: $||L_n$, beinhaltet alle Paare, die in Parallelitätsrelation zueinander stehen. Sei x||y, gdw. x$>$y und y$>$x.

Als Input für den α-Algorithmus dient ein Simple Eventlog, welcher bereits in dem Abschnitt 2.2.2 näher erläutert wurde. Gegeben sei wieder der folgende Simple Eventlog:
$L_1 = [(a, b, c, d)^3, (a, c, b, d)^2, (a, e, d)^1]$
Dann haben die oben definierten Ordnungsrelationen für diesen Eventlog die folgende Ausprägung.
$> L_1 = \{(a, b), (a, c), (a, e), (b, c), (c, b), (b, d), (c, d), (e, d)\}$

$\rightarrow L_1 = \{(a,b),(a,c),(a,e),(b,d),(c,d),(e,d)\}$
$\#L_1 = \{(a,a),(a,d),(b,b),(b,e),(c,c),(c,e),(d,a),(d,d),(e,b),(e,c),(e,e)\}$
$||L_1 = \{(b,c),(c,b)\}$

Die Ordnungsrelationen werden für jedes Paar von Aktivitäten abgeleitet. Zur Visualisierung eignet sich eine Footprint-Matrix, welche die Relationen übersichtlich darstellt. Im Fall von L1 würde sie, wie in Abbildung 2.9 dargestellt, aussehen.

	a	b	c	d	e
a	$\#_{L_1}$	$\rightarrow_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$	$\rightarrow_{L_1}$
b	$\leftarrow_{L_1}$	$\#_{L_1}$	$\|\|_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$
c	$\leftarrow_{L_1}$	$\|\|_{L_1}$	$\#_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$
d	$\#_{L_1}$	$\leftarrow_{L_1}$	$\leftarrow_{L_1}$	$\#_{L_1}$	$\leftarrow_{L_1}$
e	$\leftarrow_{L_1}$	$\#_{L_1}$	$\#_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$

Abbildung 2.9: Footprint-Matrix zur Darstellung der Ordnungsrelationen (van der Aalst, 2016)

Aus den vier Eventlog basierten Ordnungsrelationen, bzw. aus der Footprint-Matrix lassen sich nun Muster ableiten. In Abbildung 2.10 werden die typischen Muster aufgezeigt. Allerdings zeigt die Abbildung nur einfache Muster und soll als grundlegende Idee dienen. Beispielsweise wird das XOR-Split Muster modelliert, wenn in der Footprint-Matrix die Footprints a→b, a→c und b#c gegeben sind.

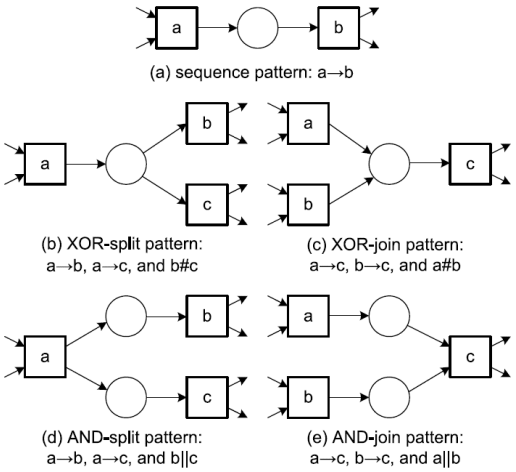

Abbildung 2.10: Typische Muster und Footprints aus einem Eventlog (van der Aalst, 2016)

Definition 6.4 (α-algorithm) Let L be an event log over $T \subseteq \mathscr{A}$. $\alpha(L)$ is defined as follows:

1. $T_L = \{t \in T \mid \exists_{\sigma \in L}\ t \in \sigma\}$,
2. $T_I = \{t \in T \mid \exists_{\sigma \in L}\ t = first(\sigma)\}$,
3. $T_O = \{t \in T \mid \exists_{\sigma \in L}\ t = last(\sigma)\}$,
4. $X_L = \{(A, B) \mid A \subseteq T_L \wedge A \neq \emptyset \wedge B \subseteq T_L \wedge B \neq \emptyset \wedge \forall_{a \in A} \forall_{b \in B}\ a \rightarrow_L b \wedge \forall_{a_1, a_2 \in A}\ a_1 \#_L a_2 \wedge \forall_{b_1, b_2 \in B}\ b_1 \#_L b_2\}$,
5. $Y_L = \{(A, B) \in X_L \mid \forall_{(A', B') \in X_L} A \subseteq A' \wedge B \subseteq B' \Longrightarrow (A, B) = (A', B')\}$,
6. $P_L = \{p_{(A,B)} \mid (A, B) \in Y_L\} \cup \{i_L, o_L\}$,
7. $F_L = \{(a, p_{(A,B)}) \mid (A, B) \in Y_L \wedge a \in A\} \cup \{(p_{(A,B)}, b) \mid (A, B) \in Y_L \wedge b \in B\} \cup \{(i_L, t) \mid t \in T_I\} \cup \{(t, o_L) \mid t \in T_O\}$, and
8. $\alpha(L) = (P_L, T_L, F_L)$.

Abbildung 2.11: Definition des α-Algorithmus (van der Aalst, 2016)

Der α-Algorithmus lässt sich in 8 Schritte aufteilen. Die Hauptaktivitäten des Algorithmus finden in den Schritten 4 und 5 statt. Die einzelnen Schritte lassen sich vereinfacht, wie folgt erklären:

1. Ermittlung aller Transitionen. Welche Events sind im Eventlog vorhanden.
2. Ermittlung der Start-Transitionen.
3. Ermittlung der End-Transitionen.

4. Verallgemeinerte Kausalitätsrelationen, Kandidaten für Stellen finden, um Paare von kausal aufeinanderfolgenden Transitionen zu verbinden.
5. Maximale Kausalitätsrelationen, erzeugt eine Übersichtlichkeit im Petrinetz. Minimale Anzahl von Stellen, die benötigt werden, um alle Paare von kausal aufeinanderfolgenden Transitionen zu verbinden.
6. Stellen zwischen die Paarmengen erzeugen und die Anfangs- und Endstelle generieren.
7. Stellen und Transitionen mit Kanten verbinden.
8. Petrinetz als Output.

Eine ausführliche und mathematische Definitionsbeschreibung findet man auf den Seiten $171 - 174$ (van der Aalst, 2016).

2.3.3 Schwächen des α-Algorithmus

Schleifenkonstrukte

Der Standard α-Algorithmus, der im obigen Abschnitt vorgestellt wurde, hat einige Einschränkungen. Eine Einschränkung besteht darin, dass er Schwierigkeiten hat, mit Schleifenkonstrukten zu arbeiten, die weniger als drei Transitionen enthalten. Unterschieden wird in Schleifen der Länge 1 und Schleifen der Länge 2. Ersteres ist in der Abbildung 2.12 dargestellt. Das obere Modell wurde inkorrekt durch den Algorithmus erzeugt, das untere zeigt allerdings das korrekte Modell für Schleifen der Größe 1. Diese Schleifen mit nur einer Transition im Eventlog haben eine bestimmte Form, die es schwierig macht, eine Stelle zu erstellen, die B als Input und ebenfalls als Output besitzt. Dies ist aufgrund der logischen Einschränkungen, die mit dem Erstellen von Stellen verbunden sind, nicht möglich (Medeiros u. a., 2003).

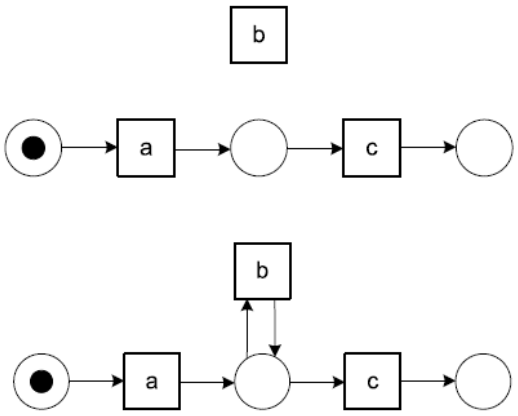

Abbildung 2.12: Schwäche mit Schleifen der Größe 1 (van der Aalst, 2016)

Durch einen Pre- und Postprocessing Ansatz lässt sich dieses Problem lösen. Zuerst werden Schleifen mit nur einer Transition identifiziert und aufgezeichnet. Danach werden diese aus dem Eventlog entfernt und der α-Algorithmus wird angewendet. In der Postprocessing Phase werden die Transitionen an die richtigen Stellen angeschlossen. (Medeiros u. a., 2003)

Noise und Incompleteness

Noise bezieht sich auf ungewöhnliche oder seltene Ereignisse im Eventlog, die nicht repräsentativ für den typischen Verlauf des Prozesses sind. Diese Ereignisse können dazu führen, dass die von Process Mining Techniken erzeugten Prozessmodelle ungenau oder verzerrt sind und somit weniger aussagekräftig sind. Aus diesem Grund ist es wichtig, Noise in den Daten zu minimieren oder zu eliminieren. Noise kann auf außergewöhnliches Verhalten oder Logging Fehler zurückzuführen sein und wird manchmal als Ausreißer oder Anomalie bezeichnet.
Unvollständigkeit bezieht sich darauf, dass der Eventlog zu wenige Ereignisse enthält, um bestimmte Steuerflussstrukturen zu entdecken. Beides kann dazu führen, dass Prozessmodelle ungenau oder verzerrt sind (van der Aalst, 2016)(van der Aalst u. a., 2012). Die Bedeutung von Unvollständigkeit lässt sich anhand eines Beispiels sehr gut verdeutlichen. Gegeben sei ein Prozess mit 10 parallelen Aktivitäten und ein zugehöriger Eventlog mit 10.000 Cases. Die Anzahl an möglichen Traces beläuft sich auf $10! = 3.628.800$. Daher ist es nicht möglich, dass der Eventlog mit 10.000 Cases alle Möglichkeiten abdeckt.

Selbst wenn es im Eventlog Millionen von Traces gibt, ist es unwahrscheinlich, dass alle möglichen Variationen der gleichzeitigen Aktivitäten vorhanden sind (van der Aalst u. a., 2012). Durch die Verwendung von local completeness benötigt der α-Algorithmus, mit den oben aufgeführten 10 Aktivitäten, lediglich $10 \times (10 - 1) = 90$ Cases, anstelle von 3.628.800 Cases, um das Model zu konstruieren. Van der Aalst beschreibt local completeness in (van der Aalst, 2016) wie folgt:

> The α-algorithm uses a local completeness notion based on >L, i.e., if there are two activities a and b, and a can be directly followed by b, then this should be observed at least once in the log.

Ausblick - Fortgeschrittene Algorithmen

Der Alpha-Algorithmus mag ein bekannter Ansatz sein, um Prozesse zu entdecken, jedoch ist er nicht in der Lage, alle Herausforderungen in diesem Bereich zu meistern. Dazu gehören die Unfähigkeit, Parallelität darzustellen, mit Schleifenkonstrukten umzugehen, stille Aktionen darzustellen, doppelte Aktionen darzustellen und die Unfähigkeit OR-Split/Joins zu modellieren. Fortgeschrittene Algorithmen mit unterschiedlichen Eigenschaften sind erforderlich, um mit Problemen wie Noise, Incompleteness und der Darstellung von Prozessmodellen besser umzugehen. Unter den fortgeschrittenen Algorithmen, die diese Herausforderungen besser meistern können, sind Heuristic-Mining, Genetic Mining und Inductive Mining zu nennen. Dies ist lediglich ein Überblick darüber, dass es neben dem Alpha-Algorithmus noch andere Algorithmen gibt. Für detailliertere Beschreibungen über die Vorgehensweise und die Auswahl der Algorithmen empfiehlt sich ein Blick in die Literatur auf Seite 195ff. von (van der Aalst, 2016).

2.3.4 Qualitätsbewertung von Prozessmodellen

Bis zu diesem Punkt wurde die Modellgenerierung ausführlich beschrieben und die Schwächen des α-Algorithmus untersucht. In diesem Abschnitt wird dargelegt, auf welche Punkte es bei einer Bewertung der Qualität generierter Prozessmodelle ankommt.

Es gibt vier Kriterien, die verwendet werden können, um die Qualität von generierten Prozessmodellen zu beurteilen: Fitness, Generalisierung, Präzision und Einfachheit. Die Fitness bezieht sich auf die Fähigkeit des Modells, das beobachtete Verhalten zu erklären.

Die Generalisierung bezieht sich darauf, dass das Modell auch auf neue Daten angewendet werden können sollte und nicht nur auf die Daten, die zur Erstellung des Modells verwendet wurden (um zu vermeiden, dass das Modell „overfitted“ ist). Präzision bezieht sich darauf, dass das Modell nicht zu allgemein sein sollte und in der Lage sein sollte, spezifische Details zu erfassen (um zu vermeiden, dass das Modell „underfitted“ ist). Schließlich sollte das Prozessmodell so einfach wie möglich sein, um die Verständlichkeit und Nachvollziehbarkeit zu erhöhen (gemäß dem Occam's Razor-Prinzip). Die Abbildung 2.13 zeigt die vier Qualitätsdimensionen. Eine der größten Herausforderungen bei der Beurteilung der Qualitätskriterien besteht darin, eine gute Balance zwischen Overfitting und Underfitting zu finden. Alle vier Kriterien können quantifiziert werden, wobei die Quantifizierung im Teilbereich Prozess Conformance stattfindet (Buijs u. a., 2012).

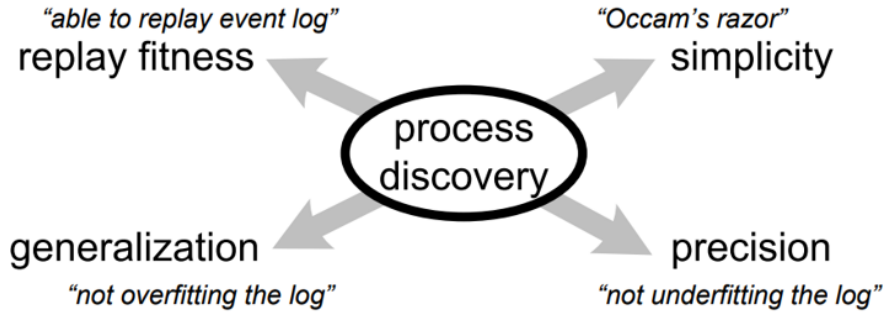

Abbildung 2.13: Qualitätskriterien von Prozessmodellen (Buijs u. a., 2012)

3 Potenziale und Herausforderungen von Process Discovery

Bis jetzt wurden die theoretischen Grundlagen von Process Mining und Process Discovery erklärt, wobei die grundlegenden Konzepte dazu beitragen, die in diesem Kapitel aufgeführten Potenziale und Herausforderungen zu verstehen und zu bewerten.

Dieses Kapitel befasst sich mit den aktuellen Herausforderungen und Möglichkeiten von Process Discovery. Um die aktuellen Themen aufzunehmen, wird hierbei eine Literaturrecherche durchgeführt. Die in diesem Kapitel untersuchten Themen werden jeweils entlang derselben grundlegenden Struktur erarbeitet. Anfangs wird das zugrundeliegende Problem genau untersucht. Darauf folgend wird eine gründliche Einführung in das Problem und die Lösungsmethoden gegeben. Schließlich wird die Anwendung der Lösung auf die beschriebenen Herausforderungen untersucht und bereits bestehende Softwarelösungen werden vorgestellt.

3.1 Objektzentriertes Process Mining

3.1.1 Problemstellung

Die meisten Techniken zur Entdeckung von Prozessmodellen aus Ereignisdaten gehen davon aus, dass es nur eine eindeutige Identifikation für jedes Ereignis gibt. In der Realität gibt es jedoch oft viele verschiedene Beziehungen zwischen verschiedenen Objekten in einem Prozess, wie zum Beispiel Bestellungen, Artikeln, Paketen, Kunden und Produkten. Diese Beziehungen können komplex sein und es kann zum Beispiel vorkommen, dass ein Artikel in mehreren Bestellungen enthalten ist, oder dass ein Paket, Artikel aus mehreren Bestellungen enthält. Dies macht die Analyse von Prozessen aus Ereignisdaten schwieriger, da diese Beziehungen berücksichtigt werden müssen (Aalst und Berti, 2020)(van der Aalst, 2019)(Adams und Van Der Aalst, 2021).

Im Gegensatz zum normalen Process Mining, bei dem der Prozess in seiner Gesamtheit betrachtet wird, wird beim objektzentrierten Process Mining der Fokus auf den Veränderungen der Objekte und ihrem Fluss durch den Prozess gelegt. Dieser Ansatz kann insbesondere dann nützlich sein, wenn die Prozessschritte nicht gut definiert sind oder sich ändern, da er eine flexiblere und dynamischere Sichtweise des Prozesses ermöglicht. Die Anwendung von objektzentrierten Process Mining kann dazu beitragen, detaillierteres Prozessverständnis, besseres Kundenverständnis oder genauere Prozesskonformität zu erlangen (van der Aalst, 2019) (Aalst und Berti, 2020).

3.1.2 Objektzentrierte Eventdaten

Die Bedeutung und der Aufbau von traditionellen Eventdaten wurde bereits in den Grundlagen näher erläutert. Bei den klassischen Eventdaten gibt es zwei Annahmen, die getroffen werden.

- Es gibt nur einen Objekttyp.
- Jedes Event referenziert nur auf einen einzigen Case.

Wir gehen davon aus, dass es viele Objekttypen gibt und dass sich ein Event auf beliebig viele Objekte beziehen kann. Der Eventlog kann unter Konvergenz (ein Event ist mit zahlreichen Cases verbunden) oder unter Divergenz (unabhängige, wiederholte Ausführung einer Gruppe von Aktivitäten innerhalb eines Cases) leiden. Das Problem ist in der Abbildung 3.1 noch einmal verdeutlicht. Dieses kann zu einer Wiederholung von Events und somit unweigerlich zu verfälschten Ergebnissen führen. Schleifen, die eigentlich keine Schleifen sind, können entstehen. Die passenden Process Mining Werkzeuge können den Anwender bei diesen Problemen allerdings unterstützen (van der Aalst, 2019).

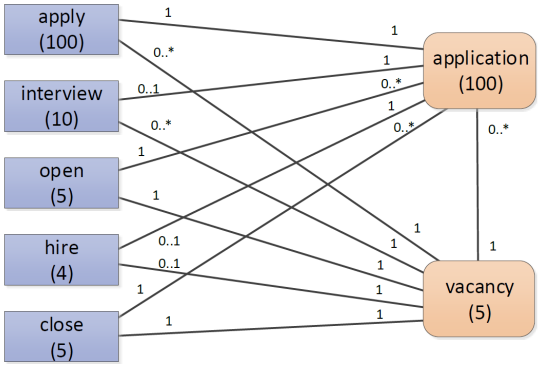

Abbildung 3.1: Ein Beispiel zur Erklärung von Konvergenz- und Divergenzproblemen. Es gibt fünf Aktivitäten (links) und zwei mögliche Objekttypen (rechts) (van der Aalst, 2019)

Konvergenz und Divergenz sind Probleme, die bei der Anwendung von traditionellen Process Mining auftreten können. Sie entstehen, wenn man versucht, Prozesse mit unterschiedlichen Objekttypen (z.B. Bewerbungen oder Stellenangebote) abzubilden.

Wie in Abbildung 3.1 zu sehen, kann es bei einem Einstellungsprozess beispielsweise sein, dass es 100 Bewerbungen und 5 Stellenangebote gibt. Wenn man nun den Objekttyp auf Bewerbungen legt und die Öffnung und Schließung der jeweiligen Stelle mit einbeziehen möchte, müsste man 100 Öffnungs- und Schließungsereignisse in der Prozessabbildung haben, anstatt nur 5. Dies wird als Konvergenz bezeichnet, da ein einziges Ereignis (Öffnen oder Schließen einer Stelle) in mehrere Fälle (Bewerbungen) aufgeteilt wird. Dadurch werden die Ereignisse repliziert und die Ergebnisse der Prozessabbildung entsprechen nicht mehr der tatsächlichen Anzahl von Ereignissen.

Alternativ kann man den Objekttyp auf Stellenangebote legen und die Bewerbungen und Interviews der jeweiligen Bewerber mit einbeziehen. In diesem Fall würde man viele Bewerbungs- und Interviewereignisse innerhalb eines einzigen Falls (Stellenangebot) sehen. Da man dann nicht mehr zwischen den einzelnen Bewerbern unterscheiden kann, gehen wichtige Informationen über die Reihenfolge der Ereignisse verloren und es entstehen Schleifen im Prozessmodell, die in der Realität nicht existieren. Dies wird als Divergenz bezeichnet. Beide Probleme können dazu führen, dass die Prozessabbildung nicht mehr den tatsächlichen Abläufen entspricht und somit ungenau oder sogar falsch ist (van der Aalst, 2019).

3.1.3 Objektzentrierte Eventlogs

Die Problematiken mit realen, objektzentrierten Daten wurden nun aufgezeigt. Es gibt einige Lösungsansätze, z.B. die durch eine Entkopplung bei der Datenextraktion die Daten in einen klassischen Eventlog umwandeln (van Eck u. a., 2015).
Ein einheitliches Dateiformat für die Speicherung und den Datenaustausch von objektzentrierten Eventlogs wurde im Jahr 2021 eingeführt. Das OCEL Format kann die bereits oben erwähnten Probleme der Konvergenz und Divergenz bewältigen, die bei der Extraktion der Daten entstehen.

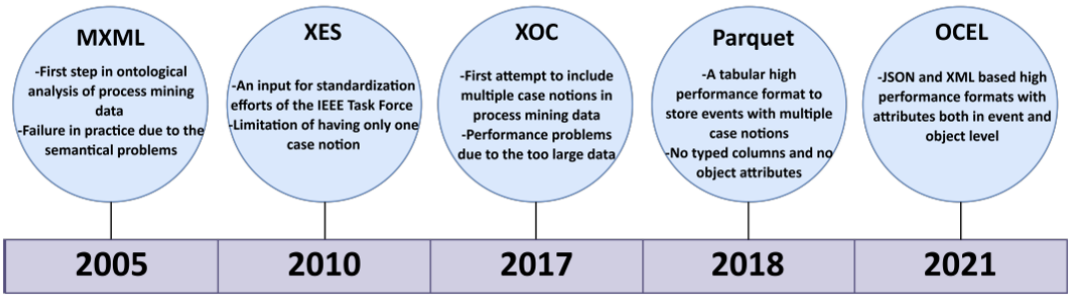

Abbildung 3.2: Ein Zeitstrahl der Standards für die Speicherung von Eventdaten (Ghahfarokhi u. a., 2021)

In (Ghahfarokhi u. a., 2021) wird das neue Format vorgestellt. Ein OCEL besteht aus den zwei Kernelementen Event und Objekt. Ein Event besitzt die Eigenschaften Identifikator, Aktivität, Zeitstempel, ggf. zusätzliche Informationen und eine Referenz zu einem oder mehreren Objekten. Der Zeitstempel ist die Grundlage für die Ordnung der Events. Im Vergleich zu klassischen Eventlogs haben die Events nicht einen einzigen Objekttypen, sondern referenzieren hier stattdessen auf eine Reihe von Objekten. Das Objekt kann ebenfalls diverse Eigenschaften haben, diese beziehen sich auf einen bestimmten Typ und werden durch einen Objektbezeichner identifiziert. In der Tabelle 3.1 und der Tabelle 3.2 ist der Aufbau informell aufgezeigt.

Tabelle 3.1: Informelle Darstellung von Events in einem OCEL

id	activity	timestamp	order	item	package	customer	resource	price
e_1	place order	2020-07-09 08:20:01.527+01:00	$\{o_1\}$	$\{i_1, i_2, i_3\}$	$\emptyset$	$\{c_1\}$	Alessandro	200.0
e_2	confirm order	2020-07-10 09:23:01.527+01:00	$\{o_1\}$	$\emptyset$	$\emptyset$	$\emptyset$	Anahita	302.0
e_3	check availability	2020-07-10 17:10:08.527+01:00	$\{o_1\}$	$\{i_1\}$	$\emptyset$	$\emptyset$	Gyunam	125.0
...	...	...	...	...	...	...	...	...

Tabelle 3.2: Informelle Darstellung von Objekten in einem OCEL

id	type	product	color	age	job
i_1	item	iPod	silver		
c_1	customer			young	teacher
...	...	...	...	...	...

Das vollständige Metamodell des OCEL Formats ist im Anhang A.1 aufgeführt.

3.1.4 Möglichkeiten der Darstellung von OCEL

In 3.1.3 wurden die Grundlagen rund um objektzentrierte Eventlogs erläutert. In (Aalst und Berti, 2020) werden die objektzentrierten Petrinetze vorgestellt, die dazu dienen, die OCEL abzubilden. Hierbei handelt es sich um eine Subklasse der gefärbten Petrinetze. Im Gegensatz zu klassischen Petrinetzen sind hier Tokens Instanzen von Objekten und die Farbe dieser spiegelt den Objekttyp wider. Stellen sind ebenfalls typisiert und können nur Tokens des jeweiligen Objekttyps enthalten. Transitionen werden auf Grundlage von Objekttypen eingefärbt, auf die sie sich beziehen und können mehrere Tokens einer Stelle konsumieren oder mehrere Tokens für eine Stelle produzieren. Dieses wird gekennzeichnet mit einer doppelten Kante zwischen der Transition und der Stelle.

In dieser Arbeit wird der generische Process Discovery Ansatz von Berti und Van der Aalst für objektzentrierte Petrinetze untersucht. Es wird ein Beispiel Eventlog betrachtet und gezeigt, wie es mittels eines OCPN dargestellt werden kann.
Das Vorgehen, welches in sieben Schritte aufgeteilt ist, wird in vereinfachter Form anhand des Kontextes dieser Arbeit erläutert.

1. Gegeben sei ein OCEL. Identifiziere die Objekttypen. Für jeden Objekttyp wird ein flattened Eventlog erzeugt.
2. Für jeden flattened Eventlog wird mit klassischen Process Discovery Techniken ein Petrinetz erzeugt.
3. Führe die Petrinetze zu einem Petrinetz zusammen.
4. Weise den Stellen im fusionierten Petrinetz Objekttypen zu.
5. Identifiziere die variablen Kanten, die Kanten bei denen mehrere Tokens konsumiert oder produziert werden.
6. Kombiniere die letzten drei Schritte, um ein OCPN zu erhalten.
7. Gebe das OCPN zurück.

Eine detailliertere und mathematische Beschreibung der oben aufgeführten Punkte ist in (Aalst und Berti, 2020) beschrieben. Ebenso wird dort beispielhaft ein OCEL vorgestellt, siehe 3.3. Dieser Eventlog spiegelt ein einfaches Beispiel wider, da es lediglich eine one-to-many Beziehung zwischen Order und Item gibt. In realen Prozessen gibt es unweigerlich mehrere many-to-many Beziehungen und lassen das OCPN deutlich komplexer werden.

activity	timestamp	order	item
...	...	...	...
place order	25-11-2019:09.35	{99001}	{88124, 88125, 88126}
pick item	25-11-2019:10.35	∅	{88126}
place order	25-11-2019:11.35	{99002}	{88127, 88128}
pick item	26-11-2019:010.25	∅	{88124}
send invoice	27-11-2019:08.12	{99001}	∅
send invoice	28-11-2019:09.35	{99002}	∅
pick item	29-11-2019:09.35	∅	{88127}
send reminder	29-11-2019:10.35	{99002}	∅
pick item	29-11-2019:11.15	∅	{88128}
ship item	29-11-2019:12.35	∅	{88124}
pick item	29-11-2019:13.30	∅	{88125}
send reminder	29-11-2019:14.35	{99001}	∅
ship item	29-11-2019:15.15	∅	{88125}
send reminder	29-11-2019:16.15	{99002}	∅
ship item	29-11-2019:17.45	∅	{88126}
ship item	29-11-2019:18.00	∅	{88128}
send reminder	30-11-2019:09.35	{99002}	∅
ship item	30-11-2019:10.05	∅	{88127}
pay order	30-11-2019:11.45	{99002}	∅
pay order	30-11-2019:12.55	{99001}	∅
mark as completed	01-12-2019:09.35	{99001}	{88124, 88125, 88126}
place order	02-12-2019:10.40	{99003}	{88129}
mark as completed	04-12-2019:11.05	{99002}	{88127, 88128}
place order	06-12-2019:14.18	{99004}	{88130, 88131, 88132, 88133, 88134}
...	...	...	...

Abbildung 3.3: Ein Teil eines Eventlogs. Jede Zeile entspricht einem Ereignis, welches mit den Objekten Order oder Item in Verbindung stehen kann (Aalst und Berti, 2020)

In dem OCPN erhalten wir zwei Arten von Stellen. Die grünen Stellen beziehen sich auf die Order und die blauen auf die Items. Ebenso sind die Transitionen nach ihrem Objekttyp eingefärbt. Der Konsum und die Produktion von mehreren Tokens ist durch einen Kantendoppelpfeil gekennzeichnet. Nach dem aufgezeigten Vorgehen von Berti und Van der Aalst entsteht nun das in Abbildung 3.4 gezeigte OCPN, welches durch die klassischen Process Discovery Techniken nicht erstellt werden kann.

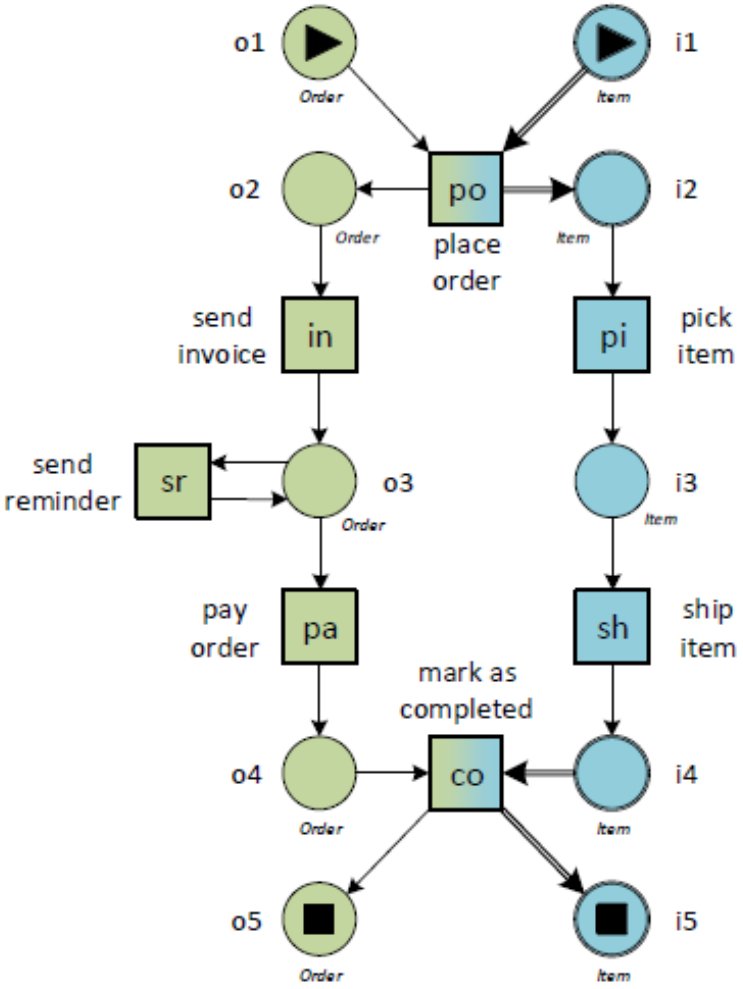

Abbildung 3.4: Ein OCPN mit zwei Objekttypen (Aalst und Berti, 2020)

3.1.5 Softwarelösungen für Objektzentriertes Process Discovery

Das Open-Source Software PM4Py[1], welches für die Modellierung in (Aalst und Berti, 2020) genutzt wird und in (Berti u. a., 2019) vorgestellt wird, ist eine neuartige Process Mining Bibliothek. Basierend auf Python und mithilfe der State of the Art Bibliotheken Pandas, Numpy, Scipy und Scikit-Learn ist PM4Py in der Lage, mit OCEL sehr performant umzugehen und dieses zu visualisieren.

[1]Downloadbar unter www.pm4py.org oder https://github.com/Javert899/pm4py-mdl

Ein weiteres frei zugängliches Tool wird in der Publikation (Park und Aalst, 2022) eingeführt. ProPPA[2] verfolgt einen ähnlichen Ansatz wie PM4Py.
Zur Darstellung der Funktionalität von PM4Py wird beispielhaft ein OCEL[3] importiert. Dieser umfasst 20.237 Events mit 5 Objekttypen. Nach dem oben aufgeführten generischen Prozess Discovery Ansatz und einer Betrachtung von lediglich drei von fünf Objekttypen wird das in 3.7 gezeigte objektzentrierte Petrinetz erstellt.
In der Abbildung 3.5 zu sehen sind 2000 eindeutige Orders (grüner Kreis), die eins zu eins in der Aktivität „place order“ auftreten. In diesen 2000 Orders sind 8159 items (roter Kreis) enthalten. PM4Py zeigt noch weitere statistische Details an, wie etwa, dass durchschnittlich 4,08 items in einer order enthalten sind.

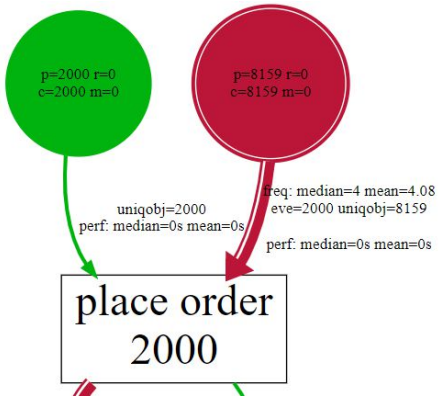

Abbildung 3.5: Ausschnitt von 3.7 zeigt die Aktivität Place Order. (Aalst und Berti, 2020)

Die Abbildung 3.6 zeigt die Aktivität „failed delivery“. Die Aktivität ist in dem Eventlog 391 Mal aufgetreten, in der 261 Packages mindestens eine „failed delivery“ hatten und dabei 1565 items involviert waren.

[2]Downloadbar unter https://github.com/gyunamister/ProPPa.git
[3]Eventlog zu finden unter https://raw.githubusercontent.com/Javert899/pm4py-mdl/master/example_logs/mdl/mdl-running-example.mdl

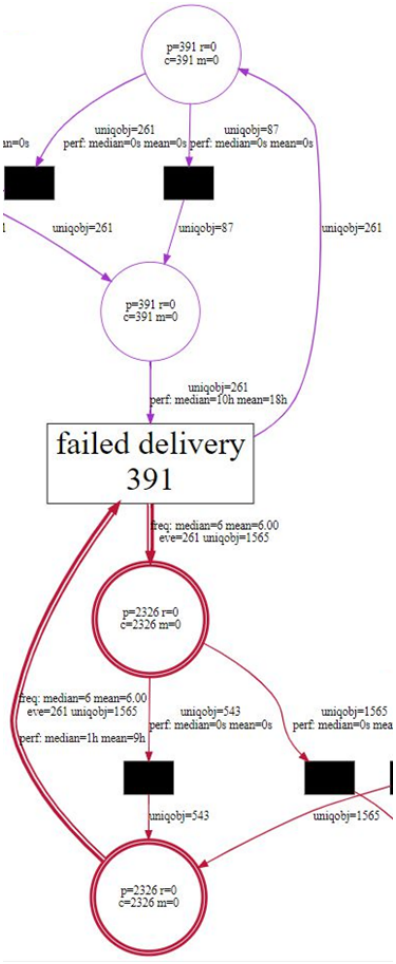

Abbildung 3.6: Ausschnitt von 3.7 zeigt die Aktivität failed delivery. (Aalst und Berti, 2020)

Die Abbildungen spiegeln sehr gut die oben aufgeführten Punkte der Komplexität und Beziehungsrelationen wider. Eine solche Darstellung wäre mit den klassischen Process Mining Werkzeugen nicht möglich. Abschließend kann festgehalten werden, dass durch die Anwendung von objektzentrierten Process Mining die Möglichkeit besteht, Prozessmodelle zu erstellen, die in ihrer Detailgenauigkeit und Realitätsnähe den klassischen Modellen überlegen sind und somit als bessere Grundlage für weitere Analysen und Entscheidungen dienen können.

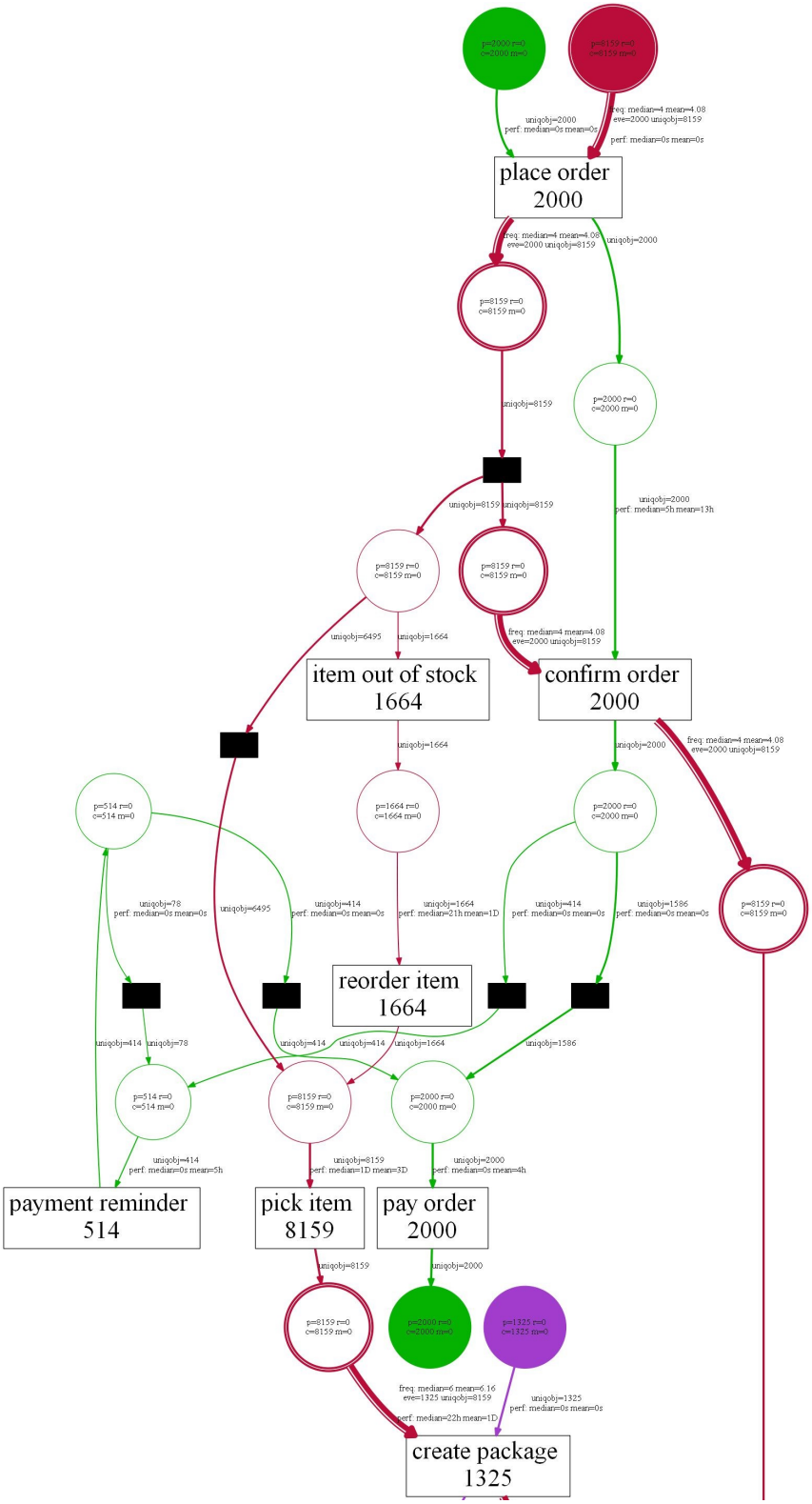

Abbildung 3.7: Ausschnitt eines objektzentrierten Petrinetzes. Generiert mit PM4Py (Aalst und Berti, 2020)

3.2 Nutzung von Fachwissen bei der datengesteuerten Process Discovery

Bisher wurde festgestellt, dass objektzentriertes Process Mining dazu beitragen kann, eine objektorientierte Perspektive auf Prozessmodelle anzufügen, um somit detailreichere und realitätsnähere Modelle zu erzeugen. In diesem Abschnitt wird dies erweitert, indem nicht nur Eventdaten als einzige Informationsquelle betrachtet werden, sondern dass auch Fachwissen der Prozessbeteiligten als zusätzliche Eingabe mit einfließen kann, um genauere Prozessmodelle zu erhalten.

3.2.1 Problemstellung

Eventdaten sind das Fundament für eine erfolgreiche Verbesserung von Prozessen im Process Mining. Sie entstehen bei der Durchführung von Prozessen und werden anschließend gespeichert. Die Qualität dieser Daten ist von größter Bedeutung, denn bestehende Algorithmen können aufgrund von mangelhafter Datenqualität unvollständige Prozessmodelle erzeugen, die das tatsächliche Verhalten nicht präzise wiedergeben. Zwar können die Eventdaten automatisch gefiltert werden, um störende Elemente zu entfernen, doch dabei gehen oft wertvolle und korrekte Daten verloren. Der Ansatz der Nutzung von Fachwissen bei der datengesteuerten Process Discovery zielt genau auf dieses Problem ab. Durch das Vorwissen über den Prozess als zusätzlichen Input neben den Eventdaten, kann eine neue Art von Entdeckungsalgorithmus entstehen.

In der Veröffentlichung von Schuster et al. (Schuster u. a., 2022), werden mehrere Algorithmen vorgestellt, die auf der Nutzung von Fachwissen basieren und von den Autoren in eine Taxonomie eingeteilt werden. In diesem Abschnitt werden die wichtigsten Erkenntnisse aus der Untersuchung von Schuster et al. vorgestellt, in der die Algorithmen klassifiziert und miteinander verglichen werden. Es werden dabei die Stärken und Schwächen der Algorithmen beleuchtet, sowie die Möglichkeiten und Grenzen der Nutzung von Fachwissen in der Prozessentdeckung aufgezeigt.

3.2.2 Arten von Nutzung von Fachwissen der datengesteuerten Process Discovery

Grundlegend wird angenommen, dass Eventdaten die objektivste Darstellung eines Prozesses ist. Allerdings haben Eventdaten oft Qualitätsprobleme wie z.B. unvollständiges Prozessverhalten aufgrund von noch laufenden Prozessinstanzen oder dass tatsächlich auftretendes Prozessverhalten nicht protokolliert wird (Martin u. a., 2021). Klassische Process Discovery Algorithmen funktionieren voll automatisch. Als Input nehmen sie einen Eventlog, optional können noch Konfigurationsparameter angegeben werden und zurückgegeben wird das fertige Prozessmodell. Es besteht keine Notwendigkeit für eine Interaktion, die von menschlichem Fachwissen angetrieben ist. Genau an dieser Stelle setzt der neue Ansatz an. Jedes Wissen über den Prozess abseits der Eventdaten wird hier als Fachwissen definiert. Dies können z.B. Prozessdokumentationen oder Prozesswissen der Prozessteilnehmer sein. Probleme mit unvollständigem Prozessverhalten oder spezifische Muster wie langfristigen Abhängigkeiten oder Prozessmodelle mit doppelten Aktivitätsbezeichnungen können so gelöst werden.
Bei aktuell existierenden Process Discovery Ansätzen wird unterschieden in vollautomatische und inkrementelle Algorithmen. Die Abbildung 3.8 zeigt die verschiedenen Ansätze und deren Eigenschaften. Beispielsweise kann Fachwissen bereits in der Vorverarbeitungsphase genutzt werden, um in dem Eventlog bestimmtes Prozessverhalten zu kennzeichnen und ggf. negatives Prozessverhalten zu entfernen. Ebenfalls können teilweise vordefinierte Prozessmodelle dem Algorithmus als zusätzlicher Input gegeben werden. In dem inkrementellen Ansatz kann der Fachnutzer während der Erzeugung des Modells aktiv Feedback über das erzeugte Prozessmodell geben und die nächsten Schritte beeinflussen. Nach Erzeugung des Modells kann in der Nachbearbeitungsphase das Modell bearbeitet werden, um letztendlich ein auf Daten und Fachwissen basiertes Prozessmodell zu erhalten (Schuster u. a., 2022).

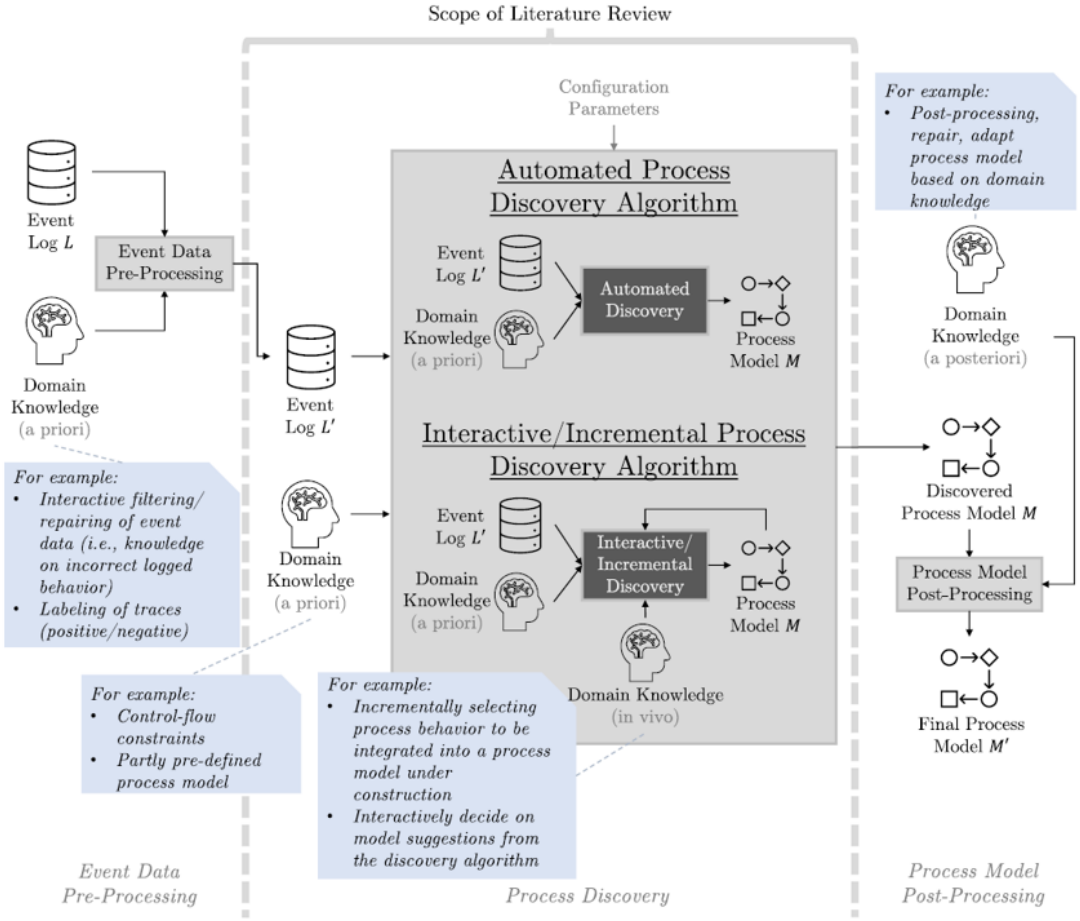

Abbildung 3.8: Übersicht der Ansätze von der Nutzung von Fachwissen für datengesteuertes Process Discovery (Schuster u. a., 2022)

3.2.3 Unterscheidungsmerkmale der Ansätze

In der Literaturrecherche von Schuster wurden 13 verschiedene Ansätze untersucht. Um die Ansätze kategorisierbar und vergleichbar zu machen, wurden die Unterscheidungsmerkmale mit ihren Eigenschaften herausgearbeitet. Es wurden acht wesentliche Unterscheidungsmerkmale identifiziert, die in der Abbildung 3.9 mit ihren Eigenschaften aufgeführt sind. Dies ermöglicht es, die Ansätze anhand dieser Merkmale zu unterscheiden und miteinander zu vergleichen.

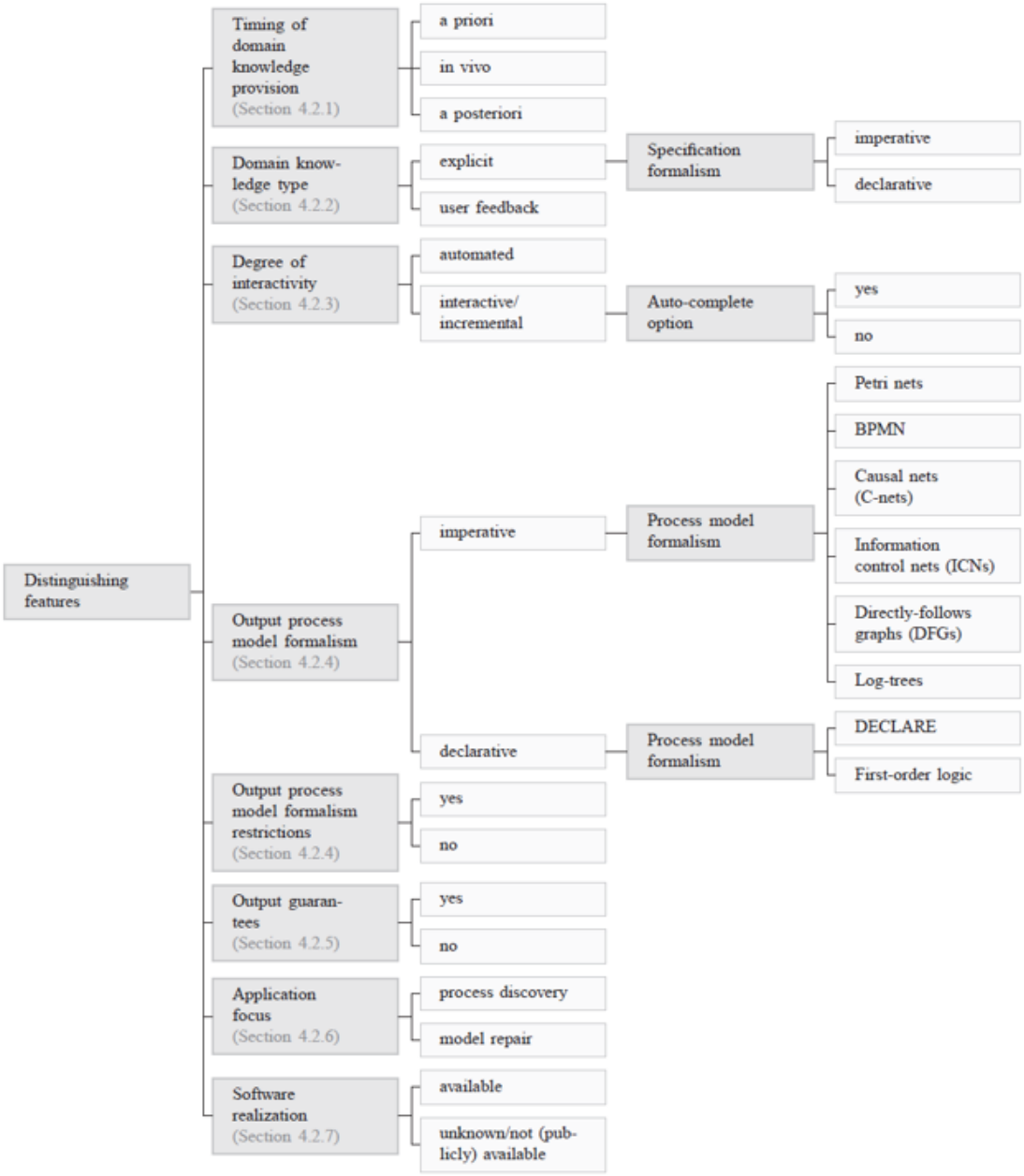

Abbildung 3.9: Übersicht über die ermittelten Unterscheidungsmerkmale (grau gefüllte Kästchen) und ihre Eigenschaften (hellgrau gefüllte Kästchen) (Schuster u. a., 2022)

Wie bereits in Sektion 3.2.2 teilweise beschrieben, wird unterschieden in, wann wird das Fachwissen genutzt, wie wird das Fachwissen übergeben, wie oft wird das Fachwissen

übergeben, wie wird das fertige Modell ausgegeben, können die übergebenen Eigenschaften garantiert werden, liegt der Fokus auf Process Discovery oder auf Modell Reparatur und gibt es eine Software Realisierung für den Ansatz.

Der aktuellste von den vorgestellten Verfahren ist der Ansatz A13 (Schuster u. a., 2020). Ein Benutzer wählt hier inkrementell eine Trace-Variante aus, die vom aktuellen Prozessmodell noch nicht unterstützt wird. Die ausgewählte Trace-Variante wird in den inkrementellen Discovery-Ansatz eingespeist. Der Ansatz verwendet Prozessbäume als Prozessmodellformalismus und garantiert, dass die inkrementell ausgewählten Trace-Varianten in das resultierende Modell passen. Er bietet auch eine Autovervollständigungsoption, indem er einfach das gesamte Verhalten eines Ereignisprotokolls zum Modell hinzufügt (Schuster u. a., 2022).

3.2.4 Softwarelösungen

Von den 13 untersuchten Ansätzen wurden acht davon mittels Software umgesetzt. Sechs davon sind als Plugin für das Open Source Programm ProM[4] verfügbar. Ein weiterer Ansatz wurde in das Tool Apromore[5] integriert. Der oben beschriebene Ansatz A13 wurde in einer Standalone Softwarelösung Cortado[6] implementiert und wird in dem Paper (Schuster u. a., 2021) näher beschrieben. Cortado nutzt für die Kernfunktionalitäten das bereits vorgestellte PM4Py. Um einen Überblick und das Potenzial von Cortado zu bekommen, wurde versucht zur Darstellung des Tools ein realen Eventlog von der BPI Challenge 2019 (van Dongen, 2019) zu importieren. Der Eventlog umfasste 251.734 Cases mit 1.595.923 Events. Der Import konnte nicht vervollständigt werden, da das Endgerät nach zwei Minuten aufgrund von fehlendem Arbeitsspeicher abstürzte. Daher wurde ein simplerer Eventlog importiert (van Dongen, 2011), der lediglich 1143 Cases beinhaltete.

[4] https://www.promtools.org/
[5] https://apromore.com/
[6] https://cortado.fit.fraunhofer.de/

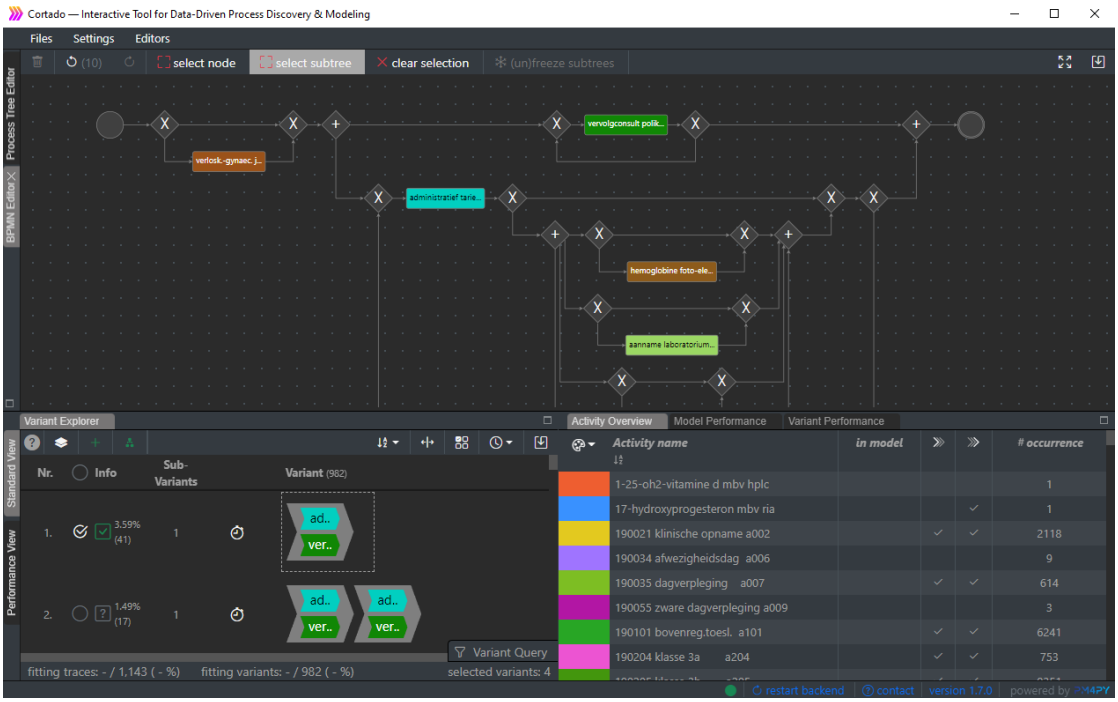

Abbildung 3.10: Grafische Benutzeroberfläche von Cortado, eigene Darstellung

Die Abbildung 3.10 zeigt die grafische Nutzeroberfläche von Cortado, am Beispiel des oben genannten Eventlogs. Im unteren linken Bereich können die verschiedenen Prozessvarianten ausgewählt werden. Das Vorschaufenster zeigt das zugehörige BPMN Diagramm zu den ausgewählten Varianten. In der Vorschau kann zwischen dem BPMN Diagramm, Prozessbaum und einem Varianten-Editor gewechselt werden. Die Performance von Cortado, insbesondere die enorme Arbeitsspeicherintensität während des initialen Imports des Eventlogs und der Modellerzeugung, führt zu dem Schluss, dass mit handelsüblicher Computerhardware die Benutzung mit großen Eventlogs nur eingeschränkt möglich ist.

3.2.5 Ausblick

Schuster, van Zelst und van der Aalst führen zehn Hauptaufgaben für zukünftige Arbeiten in diesem Bereich auf (Schuster u. a., 2022). Die Aufgaben wurden aus den Schwächen der untersuchten dreizehn Ansätze abgeleitet. Nachfolgend sind die zehn Hauptaufgaben beschrieben.

1. In der Vergangenheit wurden Ansätze verwendet, die entweder auf Fachwissen oder Nutzerfeedback basierten. Es wird empfohlen, sich nicht auf eine Art von Ansatz zu beschränken, sondern den Nutzen aus beiden zu ziehen, indem man die verschiedenen Ansätze kombiniert, um die Stärken der einzelnen Ansätze zu nutzen.

2. Es wird vorgeschlagen, verschiedene Arten von Interaktionsmodellen zur Verfügung zu stellen, damit inkrementelle und vollautomatische Verfahren kombiniert werden können.

3. Die hohe Komplexität des Alignment Conformance Checking führt bei den Ansätzen oft zu langen Rechenzeiten. Um diese Herausforderung zu bewältigen, sollten in zukünftigen Verfahren schnellere Conformance-Checking-Algorithmen eingesetzt werden.

4. Viele Ansätze beschränkten ihre Ausgabe auf einfache Modellarten wie ein Directly-Follows Graph (DFG) oder Log-Bäume. Für künftige Verfahren sollte die Zielklasse der Ausgabe und damit mögliche Einschränkungen sowie die Minimierung der Repräsentationsverzerrung beachtet werden.

5. Für einen industriellen Einsatz müssen die erzeugten Modelle schnell aufnehmbar sein. Log-Bäume oder Petrinetze sind für den einfachen Anwender jedoch schwer nachvollziehbar.

6. Die Kernaufgabe ist es, die Visualisierung auf der Nutzeroberfläche möglichst einfach und nachvollziehbar zu gestalten.

7. Eine weitere Aufgabe ist es, den Nutzer dabei zu unterstützen, das Fachwissen möglichst effizient und einfach zu übergeben.

8. Außerdem geht aus der Arbeit hervor, dass eine Priorisierung des Fachwissens von Vorteil ist.

9. Schlussendlich ist es von großer Bedeutung zukünftige Ansätze mittels Software umzusetzen.

10. Es ist wichtig alternative Perspektiven zu schaffen, indem der Fokus nicht nur auf den Kontrollfluss des Prozesses gelegt wird, sondern die weiteren Informationen in einem Eventlog wie Ressourcen und Zeitdaten besser genutzt werden.

3.3 Robotic Process Automation und Process Mining

3.3.1 Problemstellung

Die Technologie RPA ermöglicht es Unternehmen, einige wiederkehrende Aufgaben wie Dateneingabe und -verarbeitung zu automatisieren. Während RPA-Systeme eine Vielzahl von Aufgaben automatisieren können, können sie eben nicht die Aufgaben auswählen, die automatisiert werden sollten. Daher müssen Analysten z.B. Interviews, Prozessdurchläufe und Beobachtungen von Mitarbeitern durchführen, um potenzielle Routinen zu finden. Diese Techniken beinhalten das Gespräch mit Mitarbeitern oder das Beobachten ihrer Arbeit, um Routinen zu finden, die möglicherweise automatisiert werden können. Auch wenn diese Techniken nützlich sein können, um potenzielle Routinen zu finden, können sie sehr zeitaufwendig sein, vor allem in Organisationen mit vielen Aufgaben. Da es für Analysten schwierig sein kann, alle Szenarien vollständig zu analysieren, kann dies ein Skalierungsproblem darstellen (Leno u. a., 2021).

Genau an dieser Stelle kann Process Mining unterstützend zur Verfügung stehen. Process Mining ist eine Technik, die bei der Entscheidung hilft, welche Prozesse mithilfe von RPA automatisiert werden sollten. Es unterstützt dabei, geeignete Kandidaten für die Automatisierung zu finden. Process Mining kann auch nach der Einführung von RPA genutzt werden, um Prozesse und Systeme, die sowohl RPA als auch menschliche Interaktionen oder andere Arten von Automatisierung nutzen, zu überwachen und zu verwalten (Aalst, 2022) (van der Aalst u. a., 2018).

Zum Beispiel kann Process Mining auch verwendet werden, um Engpässe oder Bereiche zu identifizieren, in denen häufig Fehler in einem Prozess auftreten. RPA kann dann verwendet werden, um diese Aufgaben zu automatisieren und die Fehlerwahrscheinlichkeit zu reduzieren (Aalst, 2022) (van der Aalst u. a., 2018).

3.3.2 Einführung RPA

RPA ist ein Ansatz für die Prozessautomatisierung. RPA steht als Oberbegriff für alle Anwendungen, die auf der Nutzeroberfläche agieren und somit die menschliche Interaktion mit einem oder mehreren Systemen imitieren. Die Technologie stellt einen „Outside-In“ Ansatz dar, der die zugrunde liegenden Informationssysteme nicht verändert (van der

Aalst u. a., 2018).
RPA garantiert stets eine Genauigkeit sowie eine konsistente Abarbeitung der Aktivitäten, bei erstmaliger Ausführung. RPA kommt zum Einsatz, wenn das Informationssystem keine technische Schnittstelle anbietet. Allerdings eignen sich nicht alle Prozesse für eine Automatisierung. Nur Prozesse, die standardisiert, wiederholend und skalierbar sind, eigenen sich für RPA. Je komplexer und variantenreicher ein Prozess ist, desto weniger eignet sich dieser. Ziel ist es, die Prozessdurchlaufzeit zu verringern und somit Kosten zu sparen. Außerdem werden Mitarbeiter von repetitiven und ermüdenden Arbeiten befreit (Geyer-Klingeberg u. a., 2018). Abbildung 3.11 zeigt die Einordnung der verschiedenen Automatisierungslösungen innerhalb eines Unternehmens. Die Pareto Verteilung trifft auf Prozesse in einem Unternehmen zu. Häufig können 80% der Geschäftsvorfälle mit 20% der Case Typen erklärt werden. Diese eignen sich für die klassische Automatisierung. Ein Teil der restlichen 20% Geschäftsvorfälle eignen sich für RPA. Bei der Identifizierung dieser setzt die Sektion 3.3.3 an.

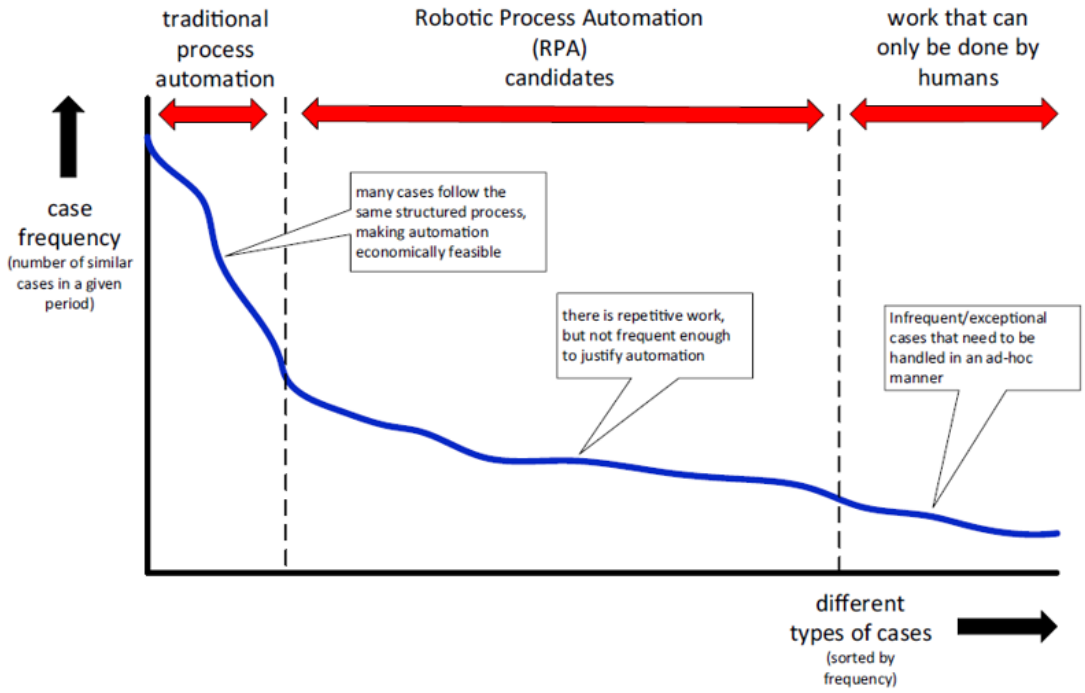

Abbildung 3.11: Einordnung von RPA (van der Aalst u. a., 2018)

3.3.3 Use Case

In (Geyer-Klingeberg u. a., 2018) stellen die Autoren die Symbiose von Process Mining und RPA dar, indem durch Prozess Discovery RPA fähige Prozesse entdeckt werden können. Das Vorgehen wird in drei verschiedene Phasen aufgeteilt und mittels einer Fallstudie für einen SAP Purchase-to-Pay (P2P) Prozess vorgestellt.

Bewertung des RPA Potenzial. Nachdem der Prozess mittels Prozess Discovery abgebildet wurde, muss eine Bewertung vorgenommen werden, inwieweit der Prozess den RPA Kriterien entspricht. Da Teile des Prozesses bereits mit den bestehenden Informationssystemen automatisiert sind, müssen die Automatisierungsraten der einzelnen Aktivitäten verglichen werden.

Entwicklung der RPA Anwendung. Es werden nun verschiedene Prototypen von Robotern entwickelt und eingesetzt. Nach mehrmaliger Ausführung muss nun der effektivste Ansatz ausgewählt werden. Die einzelnen Roboter werden miteinander und mit dem IST-Zustand verglichen, indem die Prozessdurchlaufzeit mittels Process Mining evaluiert wird.

Evaluierung der RPA Vorteile auf Dauer. Nachdem nun der effektivste Roboter im Einsatz ist, kann mithilfe von Process Mining der Einfluss sowie der Kostenvorteil überwacht werden.

3.3.4 Fallstudie

In der Fallstudie von (Geyer-Klingeberg u. a., 2018) wird das Vorgehen anhand eines realen Prozesses verdeutlicht. Zur Anwendung kommt die Process Mining Software von Celonis[7]. Die oben beschrieben Phasen sind hier wiederzufinden. In der ersten Phase wird als weiterer Input neben dem Eventlog, noch ein Nutzer-Aktivitäten-Log aus SAP importiert. In diesem werden die Ausführungen der Aktivitäten in Dialognutzer und Systemnutzer unterteilt. Somit lassen sich die Automatisierungsraten der jeweiligen Aktivitäten bestimmen. Die Rate berechnet sich wie folgt:

$$Automatisierungsrate = \frac{\text{Systemnutzeraktivitäten dieser Aktivität}}{\text{Gesamtzahl der Instanzen dieser Aktivität}}$$

[7] https://www.celonis.com/

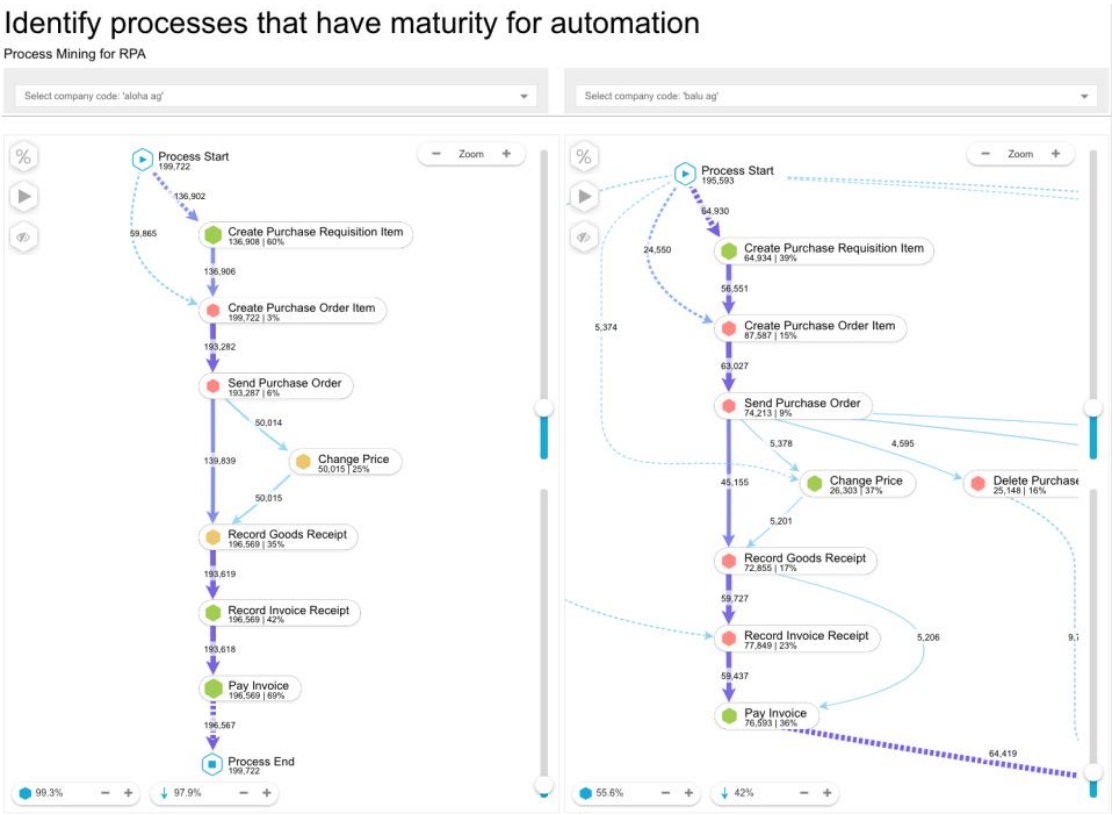

Abbildung 3.12: Ausschnitt aus Celonis für die Identifizierung von RPA fähigen Prozessen (Geyer-Klingeberg u. a., 2018)

Abbildung 3.12 zeigt einen Ausschnitt aus Celonis für zwei Firmen des Konzerns. Die Automatisierungsrate ist den jeweiligen Aktivitäten zugewiesen und eine Benchmarking-Analyse wurde durchgeführt, um den Prozessreifegrad zu vergleichen. Aus der Analyse geht hervor, dass sich der linke Prozess anhand der RPA Kriterien besser für eine RPA Implementierung eignet, da dort u.a. 99,3% der Aktivitäten abgebildet sind.
Darüber hinaus bietet Celonis eine detaillierte Zusammenfassung über potenzielle Automatisierungsmöglichkeiten für spezifische Materialgruppen oder Aktivitäten, sowie den geschäftlichen Nutzen, der aus einer Automatisierung entsteht. Änderungen der Prozessleistungsindikatoren bei steigender Prozessautomatisierung können direkt erkannt werden, siehe in Abbildung 3.13.

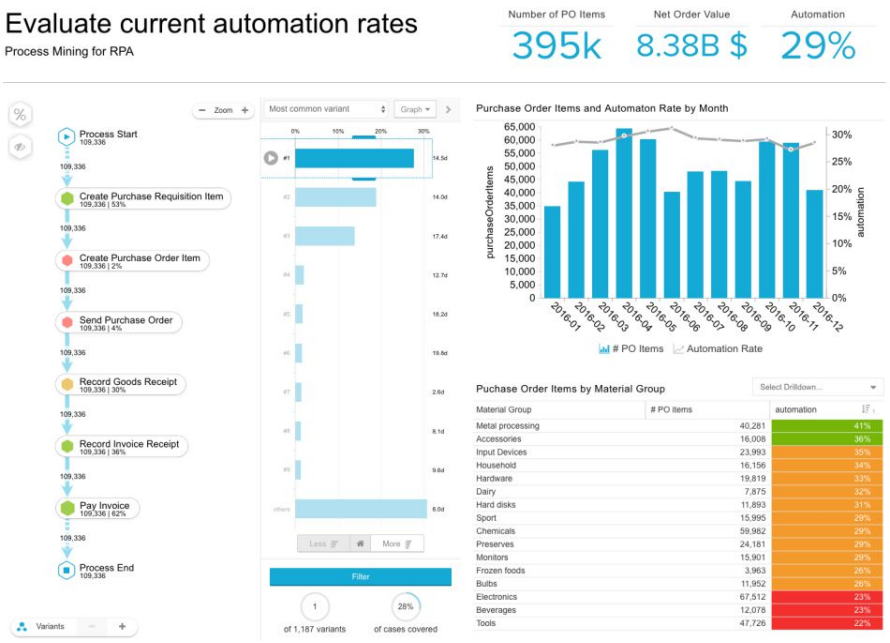

Abbildung 3.13: Zusammenfassung des Automatisierungspotenzials in Celonis (Geyer-Klingeberg u. a., 2018)

In der zweiten Phase werden nun verschiedene Roboter für unterschiedliche Prozessvarianten entwickelt und erprobt. Im Gegensatz zu RPA Software wie UIPath[8] oder Power Automate[9], die lediglich die Performance einzelner Roboter tracken, dient die Process Mining Software als Aufseher über alle Implementationen. Nach ausreichender Analyse wird nun der effektivste Roboter ausgewählt und eingesetzt. Zuletzt werden die Ergebnisse stetig überwacht und die Roboter werden ggf. angepasst. Jetzt können weitere Automatisierungspotenziale erneut mithilfe der drei Phasen identifiziert werden.

Zusammenfassend kommt Geyer-Klingenberg zu dem Ergebnis, dass eine erfolgreiche Symbiose zwischen Process Mining und RPA zustande kommt, wenn folgende fünf Punkte beachtet werden:

1. Finde und wähle eine geeignete Aktivität für RPA.
2. Standardisiere bevor du automatisierst.
3. Priorisiere Aktivitäten, in denen du den meisten Einfluss erzeugen kannst.

[8] https://www.uipath.com/de
[9] https://powerautomate.microsoft.com/de-de/

4. Richte eine Einheit im Unternehmen ein, die sich um das Projektmanagement für RPA kümmert.

5. Überprüfe ständig die Ergebnisse und passe die RPA ggf. an.

Ein weiterer bedeutender Schritt, bevor die erste geeignete RPA Aktivität ausgewählt werden kann, fehlt. In dem Paper wird nicht auf die Auswahl einer geeigneten RPA Software eingegangen. Die Softwarelösungen unterscheiden sich deutlich in ihrer Bedienbarkeit und Funktionalität. Die Entscheidung über die Auswahl der Software bezogen auf die Fähigkeiten der angestellten Entwickler oder Teammitglieder, sowie auf die Funktionalität der Software, könnte als erster Punkt in die Aufzählung mit aufgenommen werden.

In der Welt der digitalen Transformation ist die Automatisierung von Geschäftsprozessen von zentraler Bedeutung. RPA stellt eine vorübergehende Lösung dar, solange nicht alle Informationssysteme über eine technische Schnittstelle verfügen, die eine gezielte Automatisierung ermöglicht. Indem Process Mining die Aktivitäten entdeckt, die am besten für eine Automatisierung geeignet sind, kann es dazu beitragen, diese Herausforderung zu meistern.

4 Fazit

4.1 Zusammenfassung

In dieser Bachelorarbeit wurden drei Herausforderungen, die beim Einsatz von Process Discovery auftreten können, aufgedeckt und Lösungsansätze vorgestellt. Nach einer ausführlichen Erläuterung der theoretischen Grundlagen von Process Mining und Process Discovery im Kapitel 2, widmete sich das Kapitel 3 der Untersuchung der Potenziale und Herausforderungen von Process Discovery und beantwortete die Hauptfragestellung der Arbeit.

Ein wichtiger Aspekt des Process Mining ist die Anwendung effizienter Algorithmen, die es ermöglichen, Prozesse aus verschiedenen Blickwinkeln zu analysieren. Im realen Unternehmensumfeld sind die Eventdaten jedoch deutlich komplexer als die Theorie beschreibt. Die Eventdaten haben normalerweise Beziehungen zu mehreren Objekten und nicht nur zu einem, wie es im klassischen Process Mining angenommen wird. Hier setzt das objektzentrierte Process Mining an, um die Beziehungen von Events zu Objekttypen, wie Bestellungen und Artikel, darzustellen. Ein neuer Standard für die Speicherung von objektzentrierten Daten wurde vorgestellt, genannt OCEL. Dieser Ansatz ermöglicht eine realitätsnähere und detailliertere Prozessentdeckung und somit eine tiefgreifendere Analyse der daraus generierten Prozessmodelle.

In Abschnitt 3.2 wurde ein weiteres interessantes Gebiet vorgestellt, das sich mit dem Einfluss von Fachwissen bei der datengesteuerten Prozessentdeckung beschäftigt. Im Gegensatz zu traditionellen Ansätzen, bei denen der Eventlog als einzige Datenquelle verwendet wird, kann das vorhandene Fachwissen, das bei den Prozessbeteiligten vorliegt oder dem Unternehmen zur Verfügung steht, momentan nicht genutzt werden. Es wurden verschiedene Arten der Nutzung des Fachwissens untersucht und Möglichkeiten aufgezeigt, wie diese während der Prozessentdeckung verwendet werden können. Obwohl dies ein relativ

junger Forschungsbereich ist, in dem noch viel Arbeit geleistet werden muss, um einen industriellen Einsatz zu ermöglichen, wurde ein Ausblick für zukünftige Arbeiten gegeben, der zeigt, wie dies erreicht werden kann.

In Abschnitt 3.3 wurden die Synergien zwischen der RPA und dem Process Mining vorgestellt. Durch die Prozessentdeckung werden geeignete Aktivitäten und Prozesse identifiziert, die sich für eine Automatisierung mit RPA eignen. Ein Beispiel wurde verwendet, um eine Methode zur Auswahl geeigneter Kandidaten zu erläutern und zu zeigen, wie Process Discovery dazu beitragen kann, das Automatisierungspotenzial eines Unternehmens zu erhöhen und somit sowohl wirtschaftliche als auch prozessuale Vorteile zu erzielen.

4.2 Ausblick

Process Mining ist ein schnell wachsender Bereich, der in den letzten Jahren immer mehr an Bedeutung gewonnen hat. In Zukunft wird erwartet, dass die Verwendung von künstlicher Intelligenz und maschinellem Lernen in Process Mining-Anwendungen zunehmen wird. Dies ermöglicht es, Prozesse automatisch zu analysieren und zu optimieren, was zu einer höheren Effizienz und Produktivität führen kann.

Objektzentriertes Process Mining ist ein relativ neues Forschungsgebiet, das sich mit der Analyse von Prozessen auf Objektebene beschäftigt. Es ermöglicht ein tieferes Verständnis von Prozessen und ihrer Beziehungen zu Objekten und ermöglicht es, Prozesse auf eine objektbasierte Weise zu optimieren. Es erfordert auch spezialisierte Tools und Methoden, die noch nicht vollständig entwickelt sind. Daher gibt es noch viele Herausforderungen und Möglichkeiten für zukünftige Forschungen in diesem Bereich.

Ebenso lässt sich sagen, dass die Nutzung von fachlichem Wissen in der datengesteuerten Prozessentdeckung ein Thema ist, bei dem noch viel Forschungsbedarf besteht. Es handelt sich hierbei um einen wichtigen Anwendungsfall, der das Potenzial hat, die Effizienz und Genauigkeit der Prozessentdeckung in verschiedenen Branchen erheblich zu verbessern. Der in den aktuellen Ansätzen der datengesteuerten Prozessentdeckung noch nicht berücksichtigt wird.

Literaturverzeichnis

[Aalst 2022] Aalst, Wil: Process Mining and RPA: How To Pick Your Automation Battles? (2022), 01

[Aalst und Berti 2020] Aalst, Wil ; Berti, Alessandro: *Discovering Object-Centric Petri Nets*. 10 2020

[van der Aalst 2015] Aalst, Wil M. P. van der: *Extracting Event Data from Databases to Unleash Process Mining*. S. 105–128. In: Brocke, Jan vom (Hrsg.) ; Schmiedel, Theresa (Hrsg.): *BPM - Driving Innovation in a Digital World*. Cham : Springer International Publishing, 2015. – URL https://doi.org/10.1007/978-3-319-14430-6_8. – ISBN 978-3-319-14430-6

[van der Aalst 2016] Aalst, Wil M. P. van der: *Process Mining: Data Science in Action*. 2. Heidelberg : Springer, 2016. – ISBN 978-3-662-49850-7

[van der Aalst 2019] Aalst, Wil M. P. van der: Object-Centric Process Mining: Dealing with Divergence and Convergence in Event Data. In: Ölveczky, Peter C. (Hrsg.) ; Salaün, Gwen (Hrsg.): *Software Engineering and Formal Methods*. Cham : Springer International Publishing, 2019, S. 3–25. – ISBN 978-3-030-30446-1

[van der Aalst u. a. 2018] Aalst, Wil M. van der ; Bichler, Martin ; Heinzl, Armin: Robotic Process Automation. In: *Business amp; Information Systems Engineering* 60 (2018), Nr. 4, S. 269–272

[van der Aalst u. a. 2012] Aalst, Wil van der ; Adriansyah, Arya ; Medeiros, Ana Karla A. de ; Arcieri, Franco ; Baier, Thomas ; Blickle, Tobias ; Bose, Jagadeesh C. ; Brand, Peter van den ; Brandtjen, Ronald ; Buijs, Joos ; Burattin, Andrea ; Carmona, Josep ; Castellanos, Malu ; Claes, Jan ; Cook, Jonathan ; Costantini, Nicola ; Curbera, Francisco ; Damiani, Ernesto ; Leoni, Massimiliano de ; Delias, Pavlos ; Dongen, Boudewijn F. van ; Dumas, Marlon ; Dustdar, Schahram ; Fahland, Dirk ; Ferreira, Diogo R. ; Gaaloul, Walid ; Geffen, Frank

van ; Goel, Sukriti ; Günther, Christian ; Guzzo, Antonella ; Harmon, Paul ; Hofstede, Arthur ter ; Hoogland, John ; Ingvaldsen, Jon E. ; Kato, Koki ; Kuhn, Rudolf ; Kumar, Akhil ; La Rosa, Marcello ; Maggi, Fabrizio ; Malerba, Donato ; Mans, Ronny S. ; Manuel, Alberto ; McCreesh, Martin ; Mello, Paola ; Mendling, Jan ; Montali, Marco ; Motahari-Nezhad, Hamid R. ; Muehlen, Michael zur ; Munoz-Gama, Jorge ; Pontieri, Luigi ; Ribeiro, Joel ; Rozinat, Anne ; Seguel Pérez, Hugo ; Seguel Pérez, Ricardo ; Sepúlveda, Marcos ; Sinur, Jim ; Soffer, Pnina ; Song, Minseok ; Sperduti, Alessandro ; Stilo, Giovanni ; Stoel, Casper ; Swenson, Keith ; Talamo, Maurizio ; Tan, Wei ; Turner, Chris ; Vanthienen, Jan ; Varvaressos, George ; Verbeek, Eric ; Verdonk, Marc ; Vigo, Roberto ; Wang, Jianmin ; Weber, Barbara ; Weidlich, Matthias ; Weijters, Ton ; Wen, Lijie ; Westergaard, Michael ; Wynn, Moe: Process Mining Manifesto. In: Daniel, Florian (Hrsg.) ; Barkaoui, Kamel (Hrsg.) ; Dustdar, Schahram (Hrsg.): *Business Process Management Workshops*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2012, S. 169–194. – ISBN 978-3-642-28108-2

[Adams und Van Der Aalst 2021] Adams, Jan N. ; Van Der Aalst, Wil M.: Precision and Fitness in Object-Centric Process Mining. In: *2021 3rd International Conference on Process Mining (ICPM)*, 2021, S. 128–135

[Batini und Scannapieco 2016] Batini, Carlo ; Scannapieco, Monica: *Data and Information Quality - Dimensions, Principles and Techniques*. Berlin, Heidelberg : Springer, 2016. – ISBN 978-3-319-24106-7

[Berti u. a. 2019] Berti, Alessandro ; Zelst, Sebastiaan van ; Aalst, Wil: Process Mining for Python (PM4Py): Bridging the Gap Between Process- and Data Science. (2019), 05

[Buijs u. a. 2012] Buijs, Joos ; Dongen, Boudewijn ; Aalst, Wil: On the Role of Fitness, Precision, Generalization and Simplicity in Process Discovery, 09 2012, S. 305–322. – ISBN 978-3-642-33605-8

[van Dongen 2011] Dongen, Boudewijn van: Real-life event logs - Hospital log. (2011), 3. – URL https://data.4tu.nl/articles/dataset/Real-life_event_logs_-_Hospital_log/12716513

[van Dongen 2019] Dongen, Boudewijn van: BPI Challenge 2019. (2019), 1. – URL https://data.4tu.nl/articles/dataset/BPI_Challenge_2019/12715853

[van Eck u. a. 2015] Eck, Maikel L. van ; Lu, Xixi ; Leemans, Sander J. J. ; Aalst, Wil M. P. van der: PM^2: A Process Mining Project Methodology. In: Zdravkovic, Jelena (Hrsg.) ; Kirikova, Marite (Hrsg.) ; Johannesson, Paul (Hrsg.): *Advanced Information Systems Engineering*. Cham : Springer International Publishing, 2015, S. 297–313. – ISBN 978-3-319-19069-3

[Geyer-Klingeberg u. a. 2018] Geyer-Klingeberg, Jerome ; Nakladal, Janina ; Baldauf, Fabian ; Veit, Fabian: Process Mining and Robotic Process Automation: A Perfect Match, 07 2018

[Ghahfarokhi u. a. 2021] Ghahfarokhi, Anahita F. ; Park, Gyunam ; Berti, Alessandro ; Aalst, Wil M. P. van der: OCEL: A Standard for Object-Centric Event Logs. In: Bellatreche, Ladjel (Hrsg.) ; Dumas, Marlon (Hrsg.) ; Karras, Panagiotis (Hrsg.) ; Matulevičius, Raimundas (Hrsg.) ; Awad, Ahmed (Hrsg.) ; Weidlich, Matthias (Hrsg.) ; Ivanović, Mirjana (Hrsg.) ; Hartig, Olaf (Hrsg.): *New Trends in Database and Information Systems*. Cham : Springer International Publishing, 2021, S. 169–175. – ISBN 978-3-030-85082-1

[Günther 2009] Günther: *XES Standard Definition*. http://www.xes-standard.org/_media/xes/xes_standard_proposal.pdf. 2009. – Zugriff: 2022-12-13

[Lázaro u. a. 2022] Lázaro, Oscar ; Alonso, Jesús ; Ohlsson, Philip ; Tijsma, Bas ; Lekse, Dominika ; Volckaert, Bruno ; Kerkhove, Sarah ; Nielandt, Joachim ; Masera, Davide ; Patrimia, Gaetano ; Pittaro, Pietro ; Mulè, Giuseppe ; Pellegrini, Edoardo ; Köchling, Daniel ; Naskos, Thanasis ; Metaxa, Ifigeneia ; Lessmann, Salome ; Enzberg, Sebastian von: *Next-Generation Big Data-Driven Factory 4.0 Operations and Optimization: The Boost 4.0 Experience*. S. 345–371. In: Curry, Edward (Hrsg.) ; Auer, Sören (Hrsg.) ; Berre, Arne J. (Hrsg.) ; Metzger, Andreas (Hrsg.) ; Perez, Maria S. (Hrsg.) ; Zillner, Sonja (Hrsg.): *Technologies and Applications for Big Data Value*. Cham : Springer International Publishing, 2022. – URL https://doi.org/10.1007/978-3-030-78307-5_16. – ISBN 978-3-030-78307-5

[Leno u. a. 2021] Leno, Volodymyr ; Polyvyanyy, Artem ; Dumas, Marlon ; La Rosa, Marcello ; Maggi, Fabrizio M.: Robotic Process Mining: Vision and Challenges. In: *Business Information Systems Engineering* 63 (2021), Nr. 3

[Martin u. a. 2021] Martin, Niels ; Fischer, Dominik A. ; Kerpedzhiev, Georgi D. ; Goel, Kanika ; Leemans, Sander J. J. ; Röglinger, Maximilian ; Aalst, Wil M. P. van der ; Dumas, Marlon ; La Rosa, Marcello ; Wynn, Moe T.: Opportunities and Challenges for Process Mining in Organizations: Results of a Delphi Study. In: *Business Information Systems Engineering* 63 (2021), Nr. 5

[Medeiros u. a. 2003] Medeiros, Ana ; Aalst, Wil ; Weijters, A.: Workflow Mining: Current Status and Future Directions, 11 2003, S. 389–406. – ISBN 978-3-540-20498-5

[Nolin 2019] Nolin, Jan: Data as oil, infrastructure or asset? Three metaphors of data as economic value. In: *Journal of Information, Communication and Ethics in Society* ahead-of-print (2019), 11

[Park und Aalst 2022] Park, Gyunam ; Aalst, Wil: Monitoring Constraints in Business Processes Using Object-Centric Constraint Graphs. (2022), 10

[Ponniah 2011] Ponniah, Paulraj: *Data warehousing fundamentals for IT professionals*. John Wiley & Sons, 2011

[Priese und Wimmel 2008] Priese, Lutz ; Wimmel, Harro: *Petri-Netze*. Springer, 2008

[Reisig 2010] Reisig, Wolfgang: *Petrinetze - Modellierungstechnik, Analysemethoden, Fallstudien*. Berlin Heidelberg New York : Springer-Verlag, 2010. – ISBN 978-3-834-89708-4

[Schuster u. a. 2022] Schuster, Daniel ; van Zelst, Sebastiaan J. ; van der Aalst, Wil M.: Utilizing domain knowledge in data-driven process discovery: A literature review. In: *Computers in Industry* 137 (2022), S. 103612. – URL https://www.sciencedirect.com/science/article/pii/S0166361522000070. – ISSN 0166-3615

[Schuster u. a. 2020] Schuster, Daniel ; Zelst, Sebastiaan J. van ; Aalst, Wil M. P. van der: Incremental Discovery of Hierarchical Process Models. In: Dalpiaz, Fabiano (Hrsg.) ; Zdravkovic, Jelena (Hrsg.) ; Loucopoulos, Pericles (Hrsg.): *Research Challenges in Information Science*. Cham : Springer International Publishing, 2020, S. 417–433. – ISBN 978-3-030-50316-1

[Schuster u. a. 2021] Schuster, Daniel ; Zelst, Sebastiaan J. van ; Aalst, Wil M. P. van der: Cortado—An Interactive Tool for Data-Driven Process Discovery and Modeling. In: *Application and Theory of Petri Nets and Concurrency*. Springer Internatio-

nal Publishing, 2021, S. 465–475. – URL https://doi.org/10.1007%2F978-3-030-76983-3_23

A Anhang

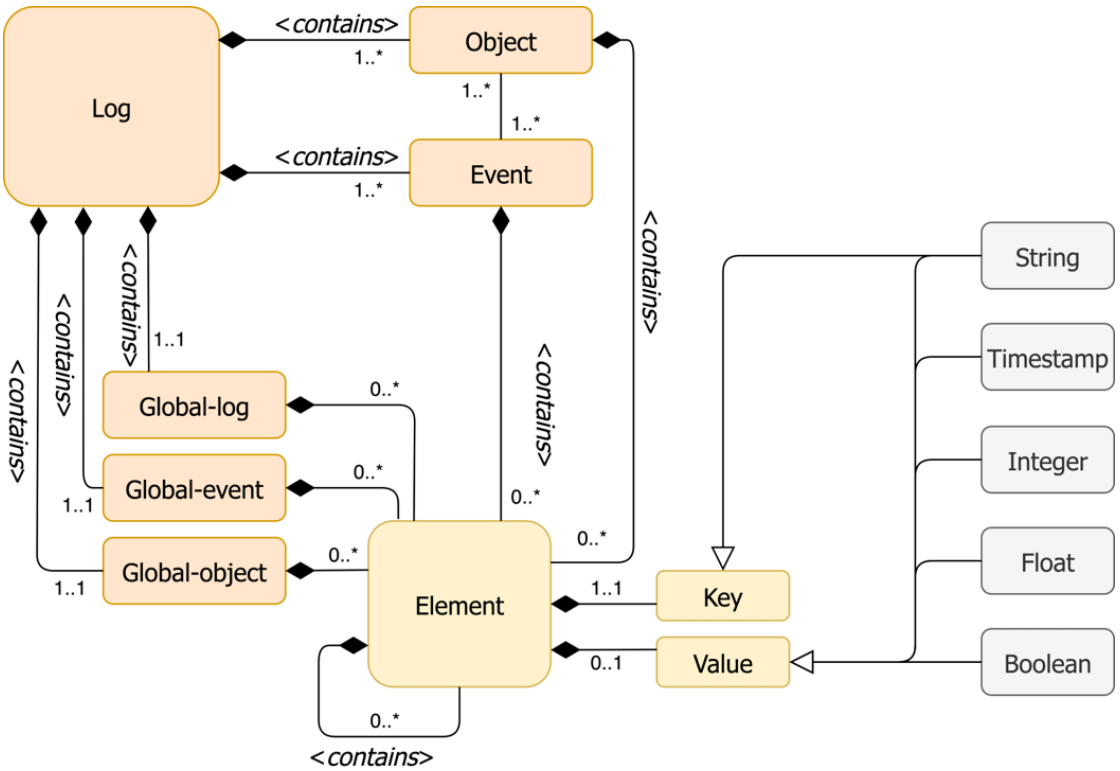

Abbildung A.1: Ein UML Klassendiagramm für das Metamodell des OCEL Formats. (Ghahfarokhi u. a., 2021)

Erklärung zur selbstständigen Bearbeitung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort	Datum	Unterschrift im Original