

Speech-to-Process: Von Sprache zum Geschäftsprozess in Python

Finn Dohrn¹

Hochschule für angewandte Wissenschaften, Hamburg
Berliner Tor 5, 20099 Hamburg
`finn.dohrn@haw-hamburg.de`

Zusammenfassung Das *Business Process Modeling* (BPM) beschäftigt sich mit der Modellierung von Geschäftsprozessen. Domänenwissen ist die Grundlage für modellierte Prozesse und wird am häufigsten durch Prozessverantwortliche in einer Notation dokumentiert. Die *Business Process Modeling Notation* (BPMN) ist daher in der Regel nicht vom Fachbereich anwendbar. Es stellt sich daher die Frage, wie Data Science Methoden das Process Mining Verfahren unterstützen können, um BPM intuitiver betreiben zu können. Um diese Frage zu beantworten, wurde eine Literaturrecherche vorgenommen und das prototypische Framework *Speech2Process* mit einer grafischen Oberfläche in Python entwickelt, dass automatisch aufgenommene Sprache in ein BPMN Diagramm überführt. Dieses Framework kombiniert bekannte Methoden wie Speech-Recognition, Natural Language Processing (NLP) und Process Mining. Die Ergebnisse aus der Literaturrecherche und des Frameworks zeigen, dass die Kombination dieser Methoden eine Modellierung von BPMN Diagrammen ohne Vorwissen ermöglichen. Im Vergleich zur manuellen Modellierung geschieht dies durchschnittlich am schnellsten. Die Genauigkeit des erzeugten BPMN Diagramms hängt dabei stark von der Komplexität der Sprache ab. Das Framework kann als Unterstützung in der Prozessmodellierung in Firmen und in der Lehre eingesetzt werden, um Unternehmensprozesse intuitiver zu dokumentieren. Weitere Forschungen in den einzelnen genutzten Komponenten ist notwendig, um die Genauigkeit und damit die Verwendbarkeit des Frameworks zu erhöhen.

Keywords: BPM · Process Mining · Speech-Recognition · NLP · Python

1 Einleitung

Geschäftsprozesse sind das „organisatorische Rückgrat von Unternehmen“ [6]. Sie sind maßgeblich verantwortlich für den Unternehmenserfolg [6]. Unternehmen die mit Geschäftsprozessen organisiert sind, reduzieren Kosten und tragen dazu bei strategische Ziele schneller zu erreichen. Dabei beschreiben die Prozesse den Fluss von Materialien, Informationen, Operationen und Entscheidungen. Die Prozesse leiten sich aus den Tätigkeiten verschiedener Personen ab [7].

Das *Business Process Modeling* (BPM) beschäftigt sich mit der Modellierung solcher Geschäftsprozesse. Die modellierten Prozesse erleichtern die Kommunikation und den Wissensaustausch zwischen Domänenexperten und Prozessverantwortlichen. Hierzu werden die Aktivitäten, Verzweigungen oder Zusammenführungen von Aktivitäten in einer Notation dokumentiert. Bei der Modellierung spielt auch die verbrauchte Zeit, Ressourcen und resultierende Kosten eine Rolle. Die *Business Process Modeling Notation* (BPMN) ist eine grafische Darstellungsform und verwendet dafür verschiedene Flussobjekte, Verbindungen zwischen Objekten, Daten, Artefakte und Swimlanes für Gruppierungen. Symbole und Elemente in einem BPMN Diagramm verfolgen eine einheitliche standardisierte Sprache und Darstellung. Eine Prozess-Instanz beschreibt einen konkreten ausgeführten Prozess entsprechend der BPMN Definition [10].

1.1 Problemstellung

Es existieren bereits viele grafische Tools, um BPMN zu modellieren. Dazu zählen unter anderem *Camunda Modeler* oder die Online-Version *bpmn.io*. Um die Nutzung dieser Tools zu ermöglichen, wird das Wissen über BPMN und ihrer Semantik vorausgesetzt, sowie die Fähigkeit Unternehmensprozesse korrekt in diese Notation zu abstrahieren [9]. Während Prozessverantwortliche diese Fähigkeiten in der Regel mitbringen, fehlt ihnen das Verständnis über den zu modellierenden Prozess. Domänenexperten kennen die Notation nicht, bringen dafür das Branchenwissen mit, dass für den Inhalt eines Geschäftsprozesses entscheidend ist [14]. Die Modellierung von Prozessen über die BPMN ist für die Endanwender*innen immer ein zusätzlicher kognitiver Aufwand [17]. Neben den klassischen grafischen Tools gibt es andere innovative Methoden zur Erstellung eines BPMN Diagramms. Beispielsweise über eine im Jahr 2020 definierte Domänenspezifische Sprache (engl. *domain-specific language*, kurz DSL), dem *BPMN Sketch Miner*, der aus einem Text in Echtzeit ein BPMN Diagramm erzeugt [9]. Andere Forscher*innen haben 2017 Gestenerkennung mit Spracherkennung kombiniert, um eingeschränkten Anwender*innen eine Modellierung zu ermöglichen [12]. Beide Forschungen erfordern weiterhin ein Vorwissen in DSL und der Notation.

Als Teil des BPM Lifecycles ermöglicht Process Mining das (automatisierte) Entdecken, Überwachen und Optimieren von Geschäftsprozessen, in dem sie das notwendige Wissen aus bestehenden Daten wie Ereignisprotokollen (engl. *Event Logs*) extrahieren. Event Logs bestehen aus mindestens einer eindeutigen Kennung, einer Aktivität und einem Zeitstempel. Diese Ereignisprotokolle liegen in vielen IT-Systemen, beispielsweise in Form eines Systemlogs, bereits strukturiert vor. Ein Domänenwissen zur Erstellung eines Geschäftsprozesses ist damit nicht notwendig [2]. In vielen Unternehmen sind diese Informationen nicht immer verfügbar, zum Beispiel bei neu eingeführten Prozessen. Weiterhin kann es sein, dass Event Logs nicht in strukturierter Form vorliegen oder aufgrund ihrer Komplexität für das Process Mining unbrauchbar sind, beispielsweise bei Blockchain Transaktionen in Smart Contracts [13].

Dieser Arbeit liegen damit zwei Problemstellungen zugrunde. Zum einen neue Ansätze ohne BPMN Vorwissen zur Modellierung von Geschäftsprozessen und

die daraus resultierende Notwendigkeit einer simulierten Generierung von Event Logs, um Geschäftsprozesse mit Process Mining abzuleiten. Ein möglicher Ansatz wäre die Nutzung von Prozessbeschreibungen in natürlicher Sprache durch Audioaufnahmen. Es stellt sich daher folgende Forschungsfrage: *Wie können Data-Science Methoden das Process-Mining Verfahren unterstützen, um die Erstellung von Geschäftsprozessen intuitiver zu machen?*

1.2 Methodik

In der vorliegenden Arbeit werden zunächst die Ergebnisse einer Literaturrecherche über aktuelle Forschungsergebnisse zur Forschungsfrage vorangestellt. Weiterhin wird das prototypische *Speech2Process* Framework vorgestellt, dass die Ansätze der bisherigen Forschungsarbeiten der Literaturrecherche kombiniert, um einen weiteren intuitiven Ansatz für das BPM zu ermöglichen. Dafür wird das Grundkonzept und deren Komponenten genauer erläutert. Das Tool wurde in Python entwickelt und erzeugt aus aufgenommenen natürlicher Sprache ein Event-Log und anschließend durch Process Mining einen Geschäftsprozess als BPMN Diagramm.

Im Anschluss werden die Ergebnisse der Literaturrecherche und des entwickelten Framework diskutiert, um mögliche Herausforderungen und Chancen aufzuzeigen. Dabei werden auch beispielhafte manuelle Prozessmodellierungen gegenüber dem prototypische *Speech2Process* Framework gestellt. Dadurch werden erste Zwischenergebnisse eines erweiterten intuitiven Ansatzes vorgestellt, der natürliche Sprache verwendet, um Geschäftsprozesse zu modellieren. Zum Schluss werden die Erkenntnisse nochmals zusammengefasst und ein möglicher Ausblick über weitere Forschungspotenziale gegeben.

2 Literaturrecherche

Zunächst wird ausgewählte Literatur vorgestellt, die ähnliche Ansätze untersucht haben. Dafür wurden die Ergebnisse der Literaturdatenbank *Google Scholar* über die Suchbegriffe „process mining“, „natural language processing“ und „bpm“ eingegrenzt. Die Ergebnisse geeigneter wissenschaftlicher Arbeiten ab 2018 werden hier kurz vorgestellt und zusammengefasst.

2.1 NLP in BPM

Zunächst einmal haben Bordignon et al. 2018 eine systematische Literaturrecherche zum Thema NLP und BPM durchgeführt. Dafür haben sie 33 relevante Papern zwischen 2009 und 2016 nach BPM Lifecycle Phase und NLP Techniken gruppiert. *Natural Language Processing* (NLP) beschäftigt sich mit der Verarbeitung von natürlicher Sprache. Ziel ist es dabei die nützlichsten Informationen aus der natürlichen Sprache in strukturierte Daten zu überführen [3]. Das Paper hat sich primär auf die BPM Lifecycle Phasen *Identification*, *Discovery* und *Analysis* fokussiert [3].

Wie in der Tabelle 1 zu sehen, haben sich zweidrittel (69,7 %) der untersuchten Paper mindestens mit der Phase *Discovery* oder einer Kombination aus dieser beschäftigt. In dieser Phase werden Process Mining Algorithmen angewendet, um Geschäftsprozesse aus bestehenden Daten zu modellieren. In den untersuchten Arbeiten wird NLP hauptsächlich in der Prozessmodellierung für die Textverarbeitung (engl. *Text processing*) verwendet. Das heißt die Extraktion von Prozessen mithilfe von morphologischer, syntaktischer und semantischer Analysen. Weiterhin werden auch generierte Vorlagen als alternative Methode genutzt, um sinnvolle Prozessinformationen zu extrahieren. Bei diesem Ansatz wird versucht mit einer Menge von Regeln, basierend auf der Struktur der Sätze, Informationen aus Texten zu extrahieren. Darüber hinaus wird NLP auch häufig verwendet, um die Benennung von Entitäten (engl. *Entity identification*) zu untersuchen. Weiterhin zeigen die Ergebnisse, dass die meisten der vorgestellten Lösungen NLP-Tools wie *Stanford CoreNLP*, *Trigram* oder *NLTK* verwenden [3].

Tabelle 1. Anzahl der Paper gruppiert nach BPM Lifecycle und mehrfach verwendeter NLP Techniken, sowie die relative Häufigkeit der eindeutigen Phasen-Zuordnung [3].

NLP Technik	Discovery	Identification & Discovery	Discovery & Analysis	Anaylsis	Alle	Σ
Textverarbeitung	3	14	0	7	1	25
Generierung von Vorlagen	0	13	1	5	0	18
Identifizierung von Entitäten	0	12	1	2	0	15
Text-Strukturierung	2	8	0	2	0	12
Lexikonisierung	1	4	1	1	1	8
Ähnlichkeitsmessung	0	1	0	4	0	5
Relative Häufigkeit	12,2 %	48,5 %	3,0 %	30,3%	6 %	

2.2 Chancen und Herausforderungen

In einem Paper von Van der Aa et al. wurden 2018 die Herausforderungen und Chancen von *Natural Language Processing* (NLP) zur Anwendung auf BPM untersucht. Der vorgestellte Ansatz verfolgt die Idee, mithilfe der Sprachverarbeitung notwendige Komponenten für den Event Log zu extrahieren, aus dem ein BPMN erzeugt werden kann. Die Forscher*innen sehen die Chance, dass die Erstellzeit von Prozessen durch NLP beschleunigt werden kann, die 60 % der Zeit von Prozess Management Projekten in Anspruch nimmt. Als Herausforderung wird die Schwierigkeit genannt, den korrekten Kontext aus einem Satz für ein Event zu erkennen. Für die Wiederverwendbarkeit innerhalb großer Firmen ist auch eine Mehrsprachigkeit von Vorteil. Korrekte Annotationen im Text können das Rauschen von NLP-Tools reduzieren und die Präzision von Beschreibungen erhöhen. Die mögliche Anwendung von NLP-automatisch erzeugten BPMN

Diagrammen könnte in der Lehre, Fehlersuche, bei Verordnungen oder im Gesundheitswesen zum Tragen kommen. Beispielsweise können Studenten ihr Modell mit einem automatisch erzeugten BPMN Diagramm vergleichen, um die Notation besser zu verstehen [1].

Weitere NLP-Herausforderungen bei der Extraktion von Modellen aus natürlicher Sprache sind das Vorhandensein von anaphorischer Referenzen, also Verweise von Satzteilen auf andere Satzteile. Je nach Text wurden auch Schwierigkeiten bei der Erkennung von Satzbausteinen oder Referenzen zwischen Sätzen festgestellt. In den untersuchten wissenschaftlichen Publikationen der systematischen Literaturrecherche wurden zu diesen Herausforderungen auch Techniken und Werkzeuge vorgestellt aber nicht weiter benannt, die versuchen, diese Probleme zu lösen [3].

2.3 NLP as a Service

Freytag et al. haben 2021 ein „NLP as a Service“ in der Programmiersprache Java umgesetzt, um über eine technische Schnittstelle Prozess Modelle in natürliche Sprache („Process2Text“) und vice versa („Text2Process“) zu konvertieren. Diese Funktion kann damit in bestehende BPMN Tools eingefügt werden. „Text2Process“ verwendet die Stanford CoreNLP Bibliothek, um den beschriebenen Prozess in Textform zunächst zu parsen und ihn in die relevanten semantischen Elemente zu überführen. Daraus werden Akteure, Aktivitäten, Geschäftsobjekte und Ressourcen abgeleitet. Daraus wird im Anschluss ein Prozessmodellbild erzeugt. Beide Dienste („Process2Text“) und („Text2Process“) können als RESTful Webservice genutzt werden. Dadurch können viele wissenschaftliche Anwendungsfälle identifiziert werden. Eine mögliche Chance ist die Erweiterung der NLP, um auch andere Modellierungssprachen zu beherrschen, wie Ereignisgesteuerte Prozessketten (EPK) [8].

2.4 Conversational Interface for Process Mining

Barbieri et al. haben im März 2022 eine Gesprächsschnittstelle für Process Mining zur Abfrage von Prozessen und Prozessinstanzen implementiert. Die Implementierung kombiniert NLP Techniken mit einer abstrakten logischen Darstellung für Prozessabfragen. Die beschriebene Lösung wurde in die kommerzielle RESTful¹ Schnittstelle von *Everflow* integriert. Über eine Text Eingabemaske können natürlich-sprachliche Fragen über Prozesszustände (Kosten, Durchschnittliche Laufzeiten, Zusammenfassung und weitere) gestellt werden. Durch NLP werden die primären Textbausteine extrahiert und durch semantisches Parsen in eine Datenbankabfrage überführt, um das Ergebnis anschließend aus der Event Log Datenbank zu laden. Das Ergebnis wird dann in der Benutzeroberfläche angezeigt. Die Forscher*innen sehen die Umsetzung als Chance um Informationen zu Prozessinstanzen interaktiv zu erhalten. Herausforderungen gab es beim spezifischen regelbasierten semantischen Parsen, der zu spezifisch ist und auf neue/unbekannte Fragen falsch antwortet [4].

¹ REST = Representational State Transfer ist eine HTTP API für Webservices

3 Speech-to-Process Framework

Ziel des Frameworks *Speech2Process* ist die Erstellung eines BPMN Diagramms ausgehend von einer Audioaufnahme eines Prozessablaufs. Das Tool hat eine intuitive Benutzeroberfläche und verwendet Data Science Methoden und Process-Mining Algorithmen. Das entwickelte Framework ist Open-Source und auf GitHub² verfügbar.

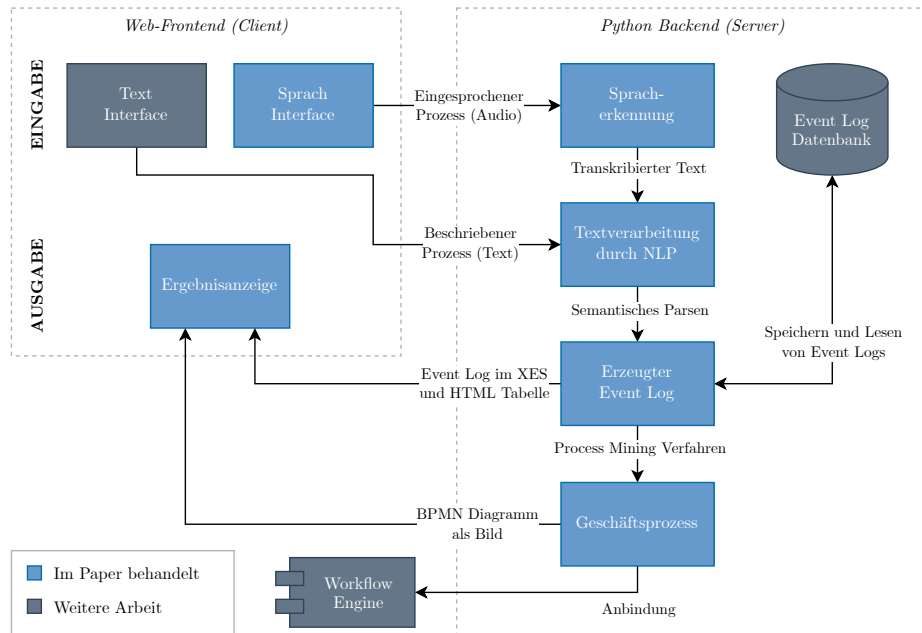


Abbildung 1. Grundkonzept des implementierten Frameworks mit bereits umgesetzten Komponenten und möglichen Erweiterungen.

Wie in Abbildung 1 gezeigt handelt es sich um eine Client-Server-Architektur. Das Frontend kommuniziert über eine RESTful Schnittstelle mit dem Backend. In der Weboberfläche werden zunächst mehrere Prozessschritte aufgenommen. Anschließend werden die Aufnahmen mit Speech Recognition in einen Gesamttext transkribiert. Aus dem Text werden mit NLP, ähnlich wie in [4] und [8] die wichtigsten Textbausteine extrahiert und durch das semantische Parsen ein Event Log erzeugt. Mit Process Mining Algorithmen wird der Event Log anschließend zu einem BPMN Diagramm überführt. Event Log und BPMN Diagramm werden dann in der Benutzeroberfläche im Ergebnisbereich angezeigt.

² <https://github.com/bitnulleins/speech2process>

3.1 Intuitive Benutzeroberfläche

Damit die Verwendung des Tools in vielen Unternehmenssituationen von Domänenexperten angewandt werden kann, wurde eine intuitive Benutzeroberfläche für das Web implementiert. Wie in Abbildung 2 schemantisch gezeigt, beschreiben die Anwender*innen auf der linken Seite in zwei Schritten einzelne Prozessschritte und Generieren dann den Geschäftsprozess aus dem Gesprochenen. Die einzelnen Aufnahmen können dabei nochmal kontrolliert werden. Nach dem Generieren werden die Ergebnisse in der Ergebnisanzeige angezeigt.

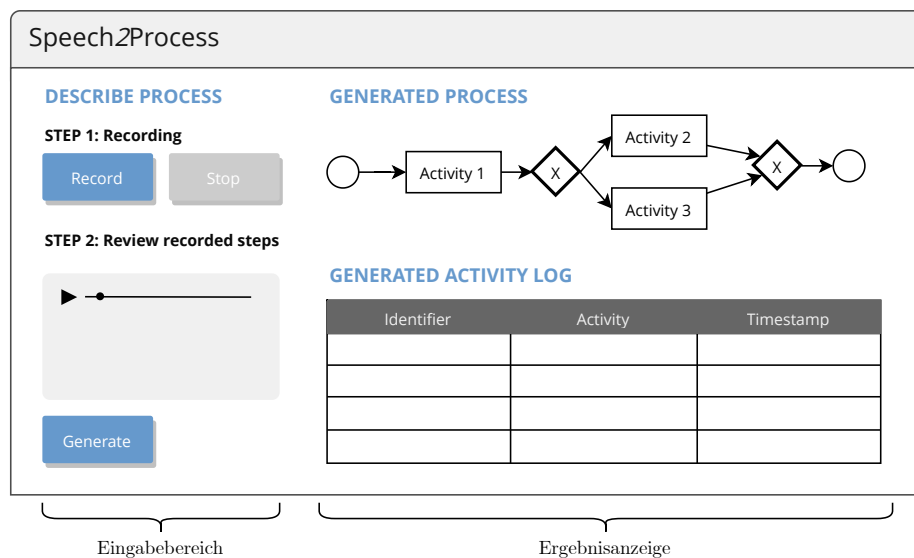


Abbildung 2. Konzept der Web-Benutzeroberfläche des Frameworks zur intuitiven Verwendung aufgeteilt in Eingabebereich und Ergebnisanzeige.

Die Weboberfläche ist responsive und daher auch für mobile Endgeräte verwendbar. Das ermöglicht die Nutzung unterwegs, um während der Prozessausführung im Unternehmen den Prozess für die Modellierung zu beschreiben. Ziel ist es, wann immer möglich, den kognitiven Aufwand der Endanwender*innen für das BPM zu reduzieren [17].

3.2 Vorverarbeitung der Prozessbeschreibung

Die einzelnen Prozessbeschreibungen werden als einzelne Sätze verarbeitet und anschließend zu einem Gesamttext zusammengeführt, um den Gesamtprozess darzustellen.

Speech Recognition Zunächst wird die Bibliothek `SpeechRecognition` verwendet, um aus den Audioaufnahmen den Text zu extrahieren [16]. Dabei bietet die Bibliothek die Möglichkeit mit unterschiedlichen Anbietern (unter anderem Microsoft, Google Cloud oder die hier verwendete kostenlose Google Speech Recognition) und deren Schnittstellen eine Transkribierung von Audioaufnahmen zu Text. Damit ist es möglich je nach Anwendungsfall eine geeignete Spracherkennung auszuwählen, zum Beispiel für Mehrsprachigkeit wie in [1] gefordert.

Natural Language Processing Anschließend werden die erzeugten Sätze mit der NLP-Bibliothek `spaCy` verarbeitet [15]. `spaCy` basiert dabei auf ein Sprachspezifisch vortrainiertes Modell, das für die Verarbeitung geladen werden muss. Daher ist diese Bibliothek für mehrsprachige Anwendungsfälle geeignet [1]. Die einzelnen Sätze werden nun durch die `spaCy`-Pipeline geführt [4]:

1. **Tokenization:** Teilt den ganzen Satz, basierend auf Leerzeichen und Interpunktionen in einzelne Satzbausteine (Tokens).
2. **Part-of-Speech (POS):** Markiert die Texte nach einer morphologischen Analyse mit Tags wie PRON (Pronomen), ADJ (Adjektiv) und VERB.
3. **Dependency Parsing:** Markiert grammatikalische Strukturinformationen und Abhängigkeitsbeziehungen zwischen den Tokens.
4. **Lemmatization:** Gibt die Grundform von Wörtern zurück (zum Beispiel gemalt → malen).
5. **Entity Recognition:** Markiert Entitäten wie Personen oder Organisationen in dem Text.

Eine beispielhafte transkribierte Prozessbeschreibung eines Prozessschritts inklusive der Verarbeitung der natürlichen Sprache mit Tokenization, POS und Dependency Parsing wird in der Abbildung 3 demonstriert. Die `spaCy`-Pipeline wird auch im Paper [4] genutzt, um relevante Bausteine der Fragen zu extrahieren.

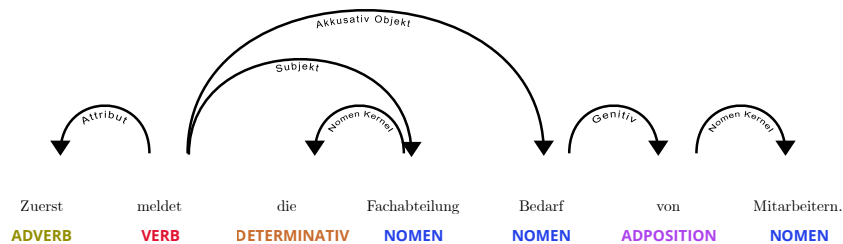


Abbildung 3. Verarbeitung des transkribierten Satzes durch NLP Bibliothek `spaCy`.

3.3 Semantisches Parsen

Das semantische Parsen zielt darauf ab, die Bedeutung eines unstrukturierten Satzes in natürlicher Sprache zu verstehen und sie auf eine Maschinenverständliche Repräsentation zu bringen. Aus dem strukturierten Zwischenergebnis wird dann der Event Log erstellt [4].

Ein Event Log L ist eine Multimenge aus Pfaden (engl. *Traces*). Ein Trace besitzt eine eindeutige ID („Fall ID“) und ist eine zeitliche Folge von einzelnen Aktivitäten A_n mit Zeitstempeln und anderen optionalen Attributen wie „Resource“. Gleiche Traces können innerhalb eines Event-Logs mehrfach auftreten, trotz unterschiedlicher ID [2]. Definieren lässt sich die Menge folgendermaßen:

$$L = [\langle A_1, \dots, A_n \rangle^k, \dots]$$

Als Ausgangslage für das semantische Parsen liegt die transkribierte Prozessbeschreibung vor. Jeder Satz stellt eine Aktivität in einem Trace dar. Die **Fall ID** für alle Aktivitäten für den Trace beginnt, da es sich hier um eine einzige beschriebene Prozessinstanz handelt, initial mit 1. Für jeden weiteren Trace wird die Fall ID inkrementiert. **Zeitstempel** werden einfachheitshalber sequentiell von einem beliebigen Ausgangsdatum pro Aktivität hochgezählt, damit die Reihenfolge der Aktivitäten berücksichtigt wird. Mögliche Konjunktionen (POSTag: CCONJ) im Satz werden genutzt, um Sätze in mehrere Einzelsätze aufzuteilen. Damit die eigentliche Aktivität und mögliche Ressourcen aus den einzelnen Sätzen extrahieren werden können, werden anschließend zwei Schritte benötigt: Regelabgleiche und Event Log Simulation.

Rule-Matching Für die Regelabgleiche (engl. *Rule-Matchings*) werden die linguistischen Eigenschaften der Sätze aus Kapitel 3.2 verwendet. Ein ähnlicher Ansatz über die Part-of-Speech Repräsentation wurde 2022 auch in der Arbeit von Barbieri et al. verfolgt [4]. Ziel des Rule-Matching ist es mit bestimmten Regeln den Aktivitätsnamen und eine mögliche Ressource gezielt aus den einzelnen Prozessschritten zu extrahieren. Damit werden auch mögliche nicht relevante Teile der Beschreibung aus dem Gesprochenen wie Verzögerungslaute (zum Beispiel „ehm“ oder „hmm“) nicht weiter verarbeitet.

Um die **Aktivität** aus einem Satz abzuleiten wird dem Objekt das Verb in der Grundform angehängt. Das Objekt lässt sich durch die Beziehung Akkusativ Objekt aus dem Satz (zum Beispiel: „Bedarf“ aus Abbildung 3) identifizieren. Falls es zu dem Objekt noch Nomen Kernel Elemente oder Genetiv Beziehungen gibt, werden Adpositionen und das dazugehörige Nomen zusätzlich angehängt, um den Kontext der Aktivität eindeutiger anzugeben (zum Beispiel „von Mitarbeitern“). Das Verb wird als Grundform aus der Lemmatization angehängt (zum Beispiel: „meldet“ → „melden“). Dadurch entsteht die Aktivität „Bedarf von Mitarbeitern melden“. Die **Ressource** lässt sich mit der Abhängigkeitsbeziehung „Subjekt“ identifizieren und als Ressource für die Aktivität speichern, zum Beispiel aus Abbildung 3 „Fachabteilung“. Die Ressource wird als optionales Attribut an das Event Log angehängt.

Event Log Simulation Zusätzlich werden noch die Gateways und Prozessschrittwiederholungen aus der gesprochenen Prozessbeschreibung simuliert. Hergestellte Beziehungen zwischen Event Logs und BPMN Diagrammen haben Kalenkova et al. 2017 in einer Arbeit ausführlich untersucht. Gateways und Prozesswiederholungen tragen dazu bei komplexe nicht lineare Prozesse abzubilden [11]. Durch Simulation von künstlich erzeugten Prozessinstanzen wird der Event Log durch neue Traces erweitert. Als Beispiel wird ein Event Log aus der Formel 1 mit einem Trace bestehend aus drei Aktivitäten angenommen.

$$L = [\langle A_1, A_2, A_3 \rangle^1] \quad (1)$$

Falls eine untergeordnete Konjunktion (POS-Tag: **SCONJ**) die konditional ist (zum Beispiel „wenn“, „falls“ und so weiter) im Satz gefunden wird, wird ein **Exclusive-Gateway** erzeugt. Jeder weiterer folgender Satz der eine untergeordnete konditionale Konjunktion enthält, bekommen eine Kopie der bis dahin bestehenden Aktivitäten A_1 bis A_n als neues Trace kopiert. Die jeweilige extrahierte Aktivität A_{n+1} und das Subjekt der untergeordneten Konjunktion wird den jeweils Traces angehängt (Formel 2.1):

$$L_{neu} = [\langle A_1, A_2, A_3, A_4 \rangle^1, \langle A_1, A_2, A_3, A_5 \rangle^1, \dots] \quad (2.1)$$

Durch die Kopien sind die Traces im Event Log identisch und unterscheiden sich erst ab einer Stelle. Process Mining Verfahren erzeugen aus einen solchen Event Log ein Exclusive Gateway und damit ein Entweder-Oder Prozessverhalten. Diese Simulationen lassen sich beliebig auf weitere syntaktische und semantischen Satzarten erweitern. Beispielsweise auf temporal untergeordnete Konjunktionen wie „während“ oder „solange“, um **Parallele Gateways** zu erzeugen. Dafür werden die parallel laufenden Aktivitäten (A_4 bis A_5 in Formel 2.2) als permutierte Menge jeweils in neue Traces kopiert:

$$L_{neu} = [\langle A_1, A_2, A_3, A_4, A_5 \dots \rangle^1, \langle A_1, A_2, A_3, A_5, A_4 \dots \rangle^1, \dots] \quad (2.2)$$

Prozesswiederholungen lassen sich erkennen, indem die extrahierte Aktivitätsbeschreibung bereits im Event Log vorkommt. Alle zu wiederholenden Aktivitäten W_n werden als Trace-Untermenge zwischen den nicht wiederholenden Aktivitäten A_1 bis A_n und dem neuen Prozessstrang B_1 und B_n kopiert:

$$L_{neu} = [\langle A_1, \dots, W_n, \dots, A_n, B_1, \dots, W_n, \dots, B_n \rangle^1] \quad (2.3)$$

3.4 Process Mining aus Event Log

Das erzeugte Event Log aus dem semantischen Parsing kann mit der Python Bibliothek `pm4py` in viele Ergebnisse überführt werden. Zum einen kann ein standardisiertes XML basiertes XES³-Dokument erzeugt werden, das in Workflow

³ XML-based standard for event logs = XML Format für Event Logs

Engines übertragen werden kann. Zum anderen lassen sich grafische Prozessdiagramme wie BPMN generieren. Dafür wird das Event Log zunächst in ein Petri-netz und dann in ein BPMN Diagramm überführt [5]. In der Implementierung des *Speech2Process* Frameworks kam dafür der *Inductive Miner* Algorithmus zum Einsatz, der vom *pm4py* Framework angeboten wird.

4 Ergebnisse und Diskussion

Die Ergebnisse der Literaturrecherche und des implementierten *Speech2Process* Frameworks zeigen, dass natürliche Sprache genutzt werden kann, um die Kommunikation zwischen Prozessen und Prozessanwender*innen zu vereinfachen. Dafür kommen auch Data Science Methoden wie NLP zum Einsatz [1,8,4], um die unstrukturierte natürliche Sprache in Datenbankabfragen [4] oder Event Logs und BPMN Diagramme zu überführen, wie in Abbildung 4.

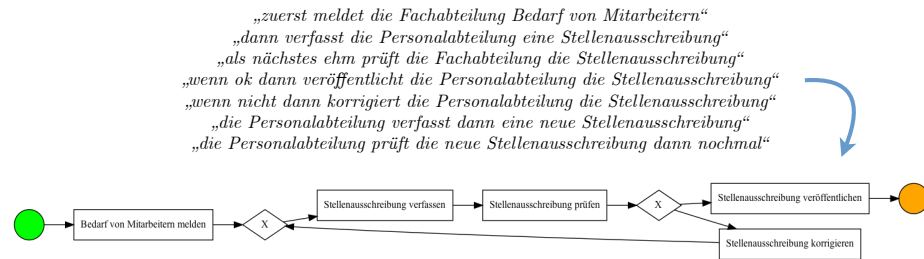


Abbildung 4. Ergebnis des *Speech2Process* Frameworks: Transkribierte Prozessbeschreibung und das erzeugte BPMN Diagramm.

Bei drei beispielhafte aufgenommene Prozessbeschreibungen war die Erstellung des BPMN Diagramms über das Framework im Vergleich zu einer manuellen Modellierung mit Vorwissen im Schnitt 28 Sekunden schneller. Das lässt sich auf die Einfachheit des Ansatzes zurückführen, da hier keine zusätzliche kognitive Leistung von BPMN erbracht werden muss [17]. Zudem wird kein Vorwissen in der Notation benötigt, wodurch der Prozess nur eingesprochen werden muss. Eine Abfrage von Prozesszuständen und Prozessperformance ist damit auch mit Fragen in natürlicher Sprache möglich, wie in [4].

Weiterhin hat der Einsatz von simulierten Event Logs den Vorteil eine Prozessbeschreibung bereits mit einer einzigen Prozessbeschreibung möglichst vollständig modellieren zu können. Eine Prozessbeschreibung entspricht dabei einer Prozessinstanz, die durch das semantische Parsen künstlich erweitert wird. Vor allem neue Prozesse lassen sich damit initial schnell ohne Vorwissen modellieren. Das bestätigt die Annahme von Van der Aa et al., dass NLP die Erstellungzeit von Prozessen durch NLP verkürzen kann [1]. Der Nachteil dieser Methode liegt in der Komplexität und Ungenauigkeit. Die Herausforderung der Forschung und des

prototypischen Frameworks liegen darin den korrekten Kontext aus Prozessbeschreibungen zu extrahieren [1]. Die Simulation von Event Logs aus Kapitel 3.3 ist nicht trivial. Der Erfolg hängt von der Einfachheit der Sprache ab, wie die Ergebnisse der Tabelle 2 zeigen. Umso komplexer die Sprache (unter anderem die Länge der Sätze oder anaphorische Referenzen [3]) und damit Prozessbeschreibungen werden, umso ungenauer werden die Ergebnisse, wie die Transkribierung und das anschließende NLP. Eine mögliche Lösung dafür wäre es mehrere Prozessbeschreibungen über die Oberfläche für einen einzigen Prozess aufzunehmen und zu speichern. Transkribiert und durch NLP sowie Rule-Matching übersetzt würde der Event Log mit vielen Prozessinstanzen gefüllt werden. Eine Event Log Simulation wäre damit nicht notwendig. Process Mining Algorithmen würden aus dem nicht-simulierten Event Log ein Geschäftsprozess generieren, der möglichst alle beschriebenen Prozesssituationen abdeckt. Das ist weniger komplex, dauert aber durch die mehrfachen Einzelaufnahmen länger. Dadurch ergibt sich aber die Chance das Tool im BPM Lifecycle zu nutzen, um den Prozess durch regelmäßig neue Prozessbeschreibung neu modellieren zu können. Das kann vor allem bei Anpassungen von Prozessen hilfreich sein.

Tabelle 2. Vergleich von Erstellzeit und Genauigkeit manuell modellierter Modelle mit Modellen aus dem *Speech2Process* Framework von drei beispielhaften Prozessen.

Prozess	Anzahl Schritte	Durchn. Satzlänge	Erstellzeit (manuell)	Erstellzeit (Framework)	Conformance Checking ⁴
1	5	10,2 Wörter	43 Sekunden	36 Sekunden	0,68
2	7	7,5 Wörter	106 Sekunden	53 Sekunden	0,92
3	4	9,6 Wörter	58 Sekunden	32 Sekunden	0,83

Herausforderungen gibt es auch in der Spracherkennung. Die verwendete Speech-Recognition erkennt keine Interpunktionen und hat auch Schwierigkeiten mit der Groß- und Kleinschreibung bei deutschen Wörtern. Dadurch kommt es in der NLP Komponente zu falschen POS. Potenzial gibt es daher im Ausbau der Spracherkennung und der Sprachverarbeitung mit NLP. Eine optimierte Spracherkennung würde Interpunktionen erkennen. Eine erweiterte NLP könnte noch weitere Gateways oder andere Komponenten eines BPMN wie Annotationen für Geschäftsprozesse ermöglichen. Annotationen können die Präzision des BPM erhöhen [1]. Außerdem können Annotationen genutzt werden, um beispielsweise Kanten des Exclusive-Gateway zu beschriften, die bisher nicht berücksichtigt wurden. Es können auch beispielsweise Adjektive extrahiert werden, um die Zufriedenheit eines Prozesses oder einzelnen Prozessschritts zu erfassen.

⁴ Conformance Checking manuelles BPMN und erzeugter Event Log: Durchschnittliche Replay Fitness durch Token-basiertes Replay

5 Zusammenfassung

Ziel der Arbeit war es zu untersuchen, wie Data-Science Methoden das Process Mining unterstützen können, um die Erstellung von Geschäftsprozessen ohne Vorwissen zu ermöglichen. Denn für die Modellierung von Geschäftsprozessen wird entweder das Wissen über die BPM Notation vorausgesetzt oder es müssen Event Logs für das Process Mining vorhanden sein.

Um die Forschungsfrage zu beantworten wurde eine Literaturrecherche vorgenommen, um eine Übersicht über aktuelle ähnliche Ansätze mit Nutzung von NLP für BPM zugeben. Van der Aa et al. sehen die NLP im BPM als beschleunigten Ansatz als Chance - zugleich die Komplexität der Sätze als Herausforderung [1]. Auch konkrete Umsetzungen als „NLP as a Service“ oder eines „Conversational Interface“ wurden aktuell untersucht und umgesetzt [8,4].

Zusätzlich wurde im Rahmen dieser Arbeit das Framework *Speech2Process* entwickelt, dass die Idee verfolgt Sprache in Geschäftsprozesse zu überführen. Dabei sind die Ansätze aus der Literaturrecherche mit eingeflossen. Dafür wird eine intuitive Weboberfläche genutzt, um einzelne Prozessbeschreibungen in natürliche Sprache aufzunehmen. Der transkribierte Text wird im Anschluss mit NLP syntaktisch und semantisch verarbeitet, um ein Event Log für das Process Mining zu erzeugen. Daraus konnte dann erfolgreich ein BPMN Diagramm erzeugt werden, das in Abbildung 4 im Kapitel 4 demonstriert wird.

5.1 Fazit

Die Ergebnisse aus der Literaturrecherche und des Frameworks zeigen, dass es möglich ist, mit Data-Science Methoden wie NLP die Prozessmodellierung zu ermöglichen und zu vereinfachen. Mögliche Herausforderungen sind dabei die Komplexität der Sätze. Die Event Log Simulation aus 3.3 ist zu spezifisch und generalisiert nicht genug. Einfluss auf die Genauigkeit des Ergebnis hat vor allem die Satzkomplexität. Chancen der Geschwindigkeitszunahme und Herausforderungen der Satzkomplexität von Van der Aa konnten damit bestätigt werden [1]. Eine prototypische Entwicklung konnte sich als Proof-of-Concept bewähren und hat einfache Geschäftsprozesse basierend auf Sprachaufnahmen erzeugt und im XES- und BPMN-Format als Ergebnis zurückgeliefert.

Data-Science Methoden wie NLP und Speech-Recognition können das Process Mining für das BPM unterstützen. Ein einfaches interaktives Interface ermöglicht es Prozessanwendern mit natürlicher Sprache ein Prozess zu modellieren oder sich über Prozesse zu informieren, wie in [4]. Der transkribierte Text, kombiniert mit den NLP Komponenten, ermöglicht eine künstliche Erzeugung eines Event Logs, um darauf Process Mining Algorithmen anzuwenden. Ein möglicher Einsatz eines solchen Tools oder des Conversational Interface von Barbieri et al. kann die Lehre [1] sein, um die BPM Notation besser zu verstehen. Ein weiterer Ansatz kann die Modellierung von neu eingeführten Prozessen sein. Die Domänenexperten haben damit die Chance über natürliche Sprache und ohne Prozesswissen eigenständig Prozesse zu modellieren oder sich über aktuelle Prozessinstanzen und Prozessperformance Kennzahlen zu informieren [4].

Die ersten Ergebnisse des implementierten Frameworks sind gut. Der implementierte Prototyp ist schnell aber ungenau. Er funktioniert besonders gut mit kurzen einfachen Sätzen, die viel Wortredundanz enthalten. Besonders die Event Log Simulationen wird, je nach Satzkomplexität, nicht immer (korrekt) erkannt. Um komplexe Prozessbeschreibungen zu ermöglichen, kann der Ansatz verfolgt werden, mehrere Prozessbeschreibungen für einen Prozess aufzunehmen und zu speichern. Die stetige Ergänzung des Event Logs verbessert sukzessiv das Modell. Denkbar wäre daher ein kombinierter Ansatz: Die initiale BPM über Event Log Simulation und das Process Redesign (als Teil des BPM Lifecycle) über weiter aufgenommene Prozessbeschreibungen.

5.2 Ausblick

Eine weitere Forschung in den Ausbau des Frameworks und deren Komponenten kann den Nutzen im BPM vielseitig erweitern. Mögliche Erweiterungen sind in der Abbildung 1 dargestellt. Dies kann eine Anbindung an eine Workflow Engine oder auch das Abspeichern mehrerer Prozessbeschreibungen pro Prozess sein. Durch ständig iterative Updates wäre die Integration des BPM Lifecycles gegeben. Weiterhin bietet das NLP und die Grammatik einer Sprache noch weiteres Forschungspotenzial, um mehr komplexe Inhalte semantisch korrekt zu extrahieren. Dies sollte weiter erforscht werden. Speziell trainierte NLP Modelle für Prozessbeschreibungen könnten die Präzision des semantischen Parsings steigern.

Literatur

1. van der Aa, H., Carmona, J., Leopold, H., Mendling, J., Padró, L.: Challenges and opportunities of applying natural language processing in business process management pp. 2791–2801 (Aug 2018), <https://aclanthology.org/C18-1236>
2. van der Aalst, W.: Process mining **3**(2), 1–17. <https://doi.org/10.1145/2229156.2229157>
3. de Almeida Bordignon, A.C., Thom, L.H., Silva, T.S., Dani, V.S., Fantinato, M., Ferreira, R.C.B.: Natural language processing in business process identification and modeling. In: Proceedings of the XIV Brazilian Symposium on Information Systems - SBSI'18. ACM Press. <https://doi.org/10.1145/3229345.3229373>
4. Barbieri, L., Madeira, E.R.M., Stroeh, K., van der Aalst, W.M.P.: Towards a Natural Language Conversational Interface for Process Mining, vol. Eindhoven, pp. 268–280 (2022). doi:10.1007/978. RWTH Aachen University. <https://doi.org/10.18154/RWTH-2022-04971>
5. Berti, A., van Zelst, S.J., van der Aalst, W.: Process mining for python (pm4py): Bridging the gap between process- and data science (May 2019)
6. Eva Best, M.W.: Geschäftsprozesse optimieren. Gabler, Betriebswirt.-Vlg (2009)
7. Fischer, H., Fleischmann, A., Obermeier, S.: Geschäftsprozesse realisieren. Vieweg+Teubner Verlag (2007)
8. Freytag, T., Leger, N., Kanzler, B., Semling, D.M.: Nlp as a service: An api to convert between process models and natural language text. In: BPM (2021)

9. Ivanchikj, A., Serbout, S., Pautasso, C.: From text to visual BPMN process models. In: Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems. ACM (oct 2020). <https://doi.org/10.1145/3365438.3410990>
10. Jochen Göpfert, H.L.: Geschäftsprozessmodellierung mit BPMN 2.0. de Gruyter Oldenbourg (2014)
11. Kalenkova, A.A., van der Aalst, W.M.P., Lomazova, I.A., Rubin, V.A.: Process mining using BPMN: relating event logs and process models **16**(4), 1019–1048. <https://doi.org/10.1007/s10270-015-0502-0>
12. Keller, T., grünert, d.: Modeling business processes using gesture and speech recognition. PROCEEDINGS OF THE 15th INTERNATIONAL CONFERENCE e-Society 2017 (04 2017)
13. Koschmider, A., Duchmann, F.: Validation of smart contracts using process mining. In: Central-European Workshop on Services and their Composition (ZEUS 2019). WS-CEUR (Februar 2019), (to appear)
14. Pufahl, L., Zerbato, F., Weber, B., Weber, I.: BPMN in healthcare: Challenges and best practices **107**, 102013. <https://doi.org/10.1016/j.is.2022.102013>
15. Vasiliev, Y.: Natural Language Processing with Python and spaCy. Random House LCC US (May 2020), https://www.ebook.de/de/product/38366269/yuli_vasiliev_natural_language_processing_with_python_and_spacy.html
16. Zhang, A.: Speech recognition (version 3.8), https://github.com/Uberi/speech_recognition#readme
17. Zimoch, M., Pryss, R., Probst, T., Schlee, W., Reichert, M.: The repercussions of business process modeling notations on mental load and mental effort. In: Business Process Management Workshops, pp. 133–145. Springer International Publishing. https://doi.org/10.1007/978-3-030-11641-5_11