Konzeption und Entwicklung eines Verfahrens zum Matching von Prozessmodellen

Divari Vasiliki

18. Juli 2011

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

In der industriellen Praxis hat sich mittlerweile eine Fokussierung auf Geschäftsprozesse etabliert. Geschäftsprozesse stellen hierbei eine Folge fachlich zusammenhängender Geschäftsaktivitäten dar, die oft einen Beitrag für die Wertschöpfung eines Unternehmens besitzen und organisations-übergreifend auch Kunden, Lieferanten und Partner einbinden können. Aufgrund kürzerer Produktzyklen und steigendem Konkurrenzdruck unter den Firmen wird es immer wichtiger, dass ein Unternehmen schneller als die Konkurrenz seine Geschäftsprozesse anpassen kann, um z.B. auf neue Kundenanforderungen zu reagieren. Ebenso wichtig ist es, dass bei der Entwicklung neuer Geschäftsprozesse die Erkenntnisse aus existierenden Prozesse so gut wie möglich genutzt und übertragen werden. Zu diesem Zweck erstellt das dBOP (Deep Business Optimization Platform) Projekt eine Plattform zur Verfügung, welche durch spezifische Datenintegrations- und Analysefähigkeiten sowie eine musterbasierte Optimierungsengine die Effizienz und die Effektivität des Analyseprozesses erhöht.

1.1 Motivation

Um die Intuition, dass die Ähnlichkeit zwei zu gewinnen Gegenstände sind mit ihrer Allgemeinheit verbunden, wir brauchen Messung der Allgemeinheit. Unsere Absicht ist, eine gesamte Definition der Ähnlichkeit zu erreichen. Ähnlichkeitsmaß kann dann abgeleitet werden von jenen Annahmen.

1.2 Überblick dBOP Plattform

In den letzten Jahren haben Unternehmen versucht, individuelle Business Funktionen für die Optimierung der gesamten Geschäftsprozesse zu finden. Die zunehmende Volatilität der wirtschaftlichen Rahmenbedingungen und der Wettbewerb zwischen den Unternehmen hat an Bedeutung in den letzten Jahren stark zugenommen. Die Fähigkeit eines Unternehmens Leistung zu erreichen, ist heutzutage eine der wichtigsten Bedürfnisse für den Wettbewerb. Die Auswahl der richtigen Prozesskonstruktion und die Anwendung des am besten geeigneten Optimierungsverfahrens sind von großer Bedeutung. Ein Unternehmen braucht eine gute Architektur, um die Herausforderung des Wettbewerbs anzunehmen und erfolgreich durchzuführen. Eine Plattform, die die Voraussetzungen erfüllt, ist der **dBOP** (**d**eep **B**usiness **O**ptimization **P**latform). Ziel der **dBOP** Plattform ist eine integrierte Umgebung, die sowohl eine halbautomatische als auch eine automatische Optimierung während der Prozesskonstruktion, -ausführung und -analyse unterstützt. [14].

Diese Plattform hat drei Schichten, die im folgenden Bild gezeigt werden (s. Abbildung 1.1):

1. Data Integration

2. Prozessanalyse

3. Prozessoptimierung

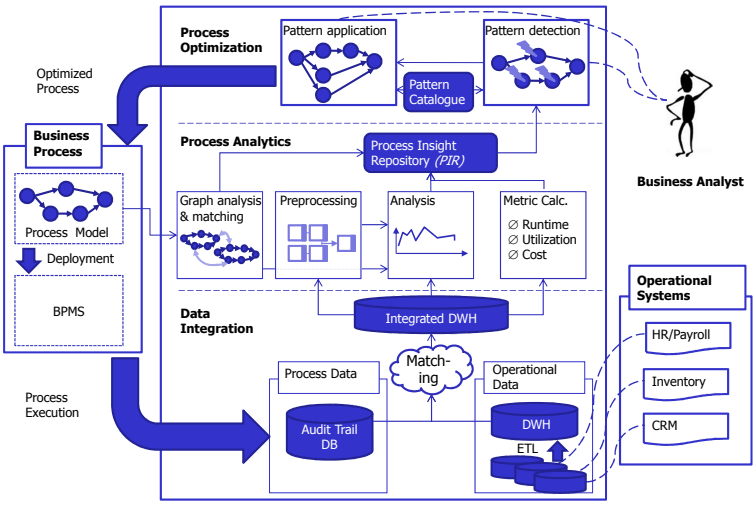

Abbildung 1.1: dBOP Plattform

1.2.1 Data Integration

Daten, die für den Prozess relevant sind, können aus einer Reihe von relevanten Quellen gefunden und innerhalb des Prozesses verteilt werden. Die am häufigsten verwendeten Datenquellen sind Audit Trail und Process Execution Daten. (s. Abbildung 1.2) [14], [16] [15].

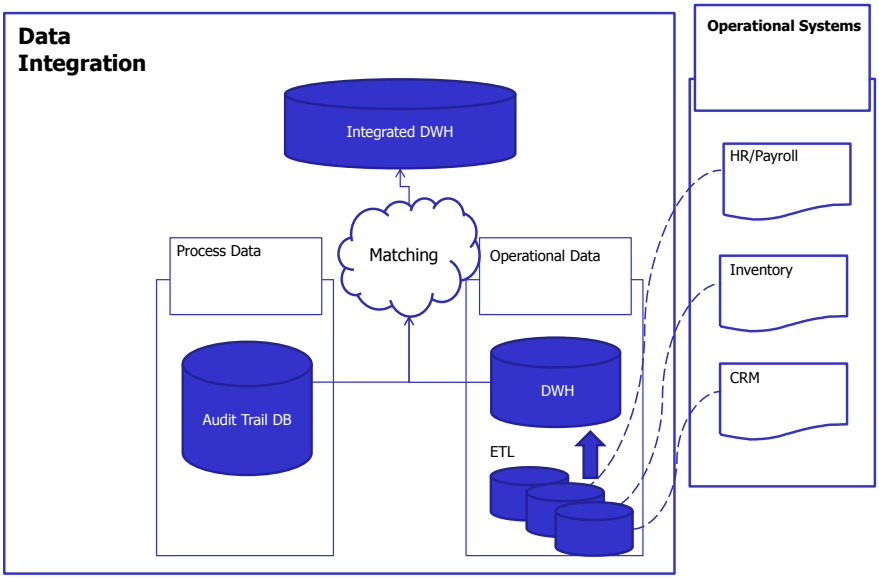

Abbildung 1.2: Daten - Integration

Audit ist das Verfahren zur Überprüfung und Verbesserung eines Prozesses in einer Organisation und zusätzlich zur Erfüllung von Anforderungen und Bewertung von Richtlinien. Es ist ein Datensatz, der zeigt, wer Zugriff zu einem Computer hat und welche Operationen man während eines bestimmten Zeitraums durchgeführt hat. Audit Trails sind sowohl für die Aufrechterhaltung der Sicherheit als auch für die Wiederherstellung von verlorenen Transaktionen nützlich [**?**]. Die meisten Abrechnungssysteme und Datenbank-Management-Systeme beinhalten eine Audit-Trail Komponente.
Andere Daten sind in der Regel in den operativen Datenquellen enthalten (z.B. eine Datei könnte in einem Prozess die ID des Mitarbeiters enthalten, aber nicht unbedingt seine Ausbildung, Berufserfahrung, usw). Die Daten werden in transparenten Tabellen gespeichert und stehen immer zur Verfügung.
Die Daten, die in den zwei Datenbanksystemen bereitgestellt sind, werden in das integrierte Data Warehouse gematched. Sie werden also von den zwei externen Datenbanken mit bestimmten manuellen und/oder automatischen Matching Methoden transformiert, um in die Umgebung des Data Warehouse integriert zu werden. Das integrierte Data Warehouse ist wiederum ein Datenbanksystem zur zentralen Datenhaltung und Zusammenführung von verschiedenen dezentralen Datenquellen [26].

Die erste Schicht beinhaltet relevante Daten, die verschiedene Datenformate haben und durch unterschiedliche Verfahren zum Einsatz gekommen sind.

1.2.2 Prozessanalyse

Die zweite Schicht ist die Schicht der Daten und der Prozessanalyse. Für die bedeutungsvollen Ergebnisse der Optimierung müssen aus der integrierten Schicht die Daten extrahiert werden (s. Abbildung 1.3) [14] .

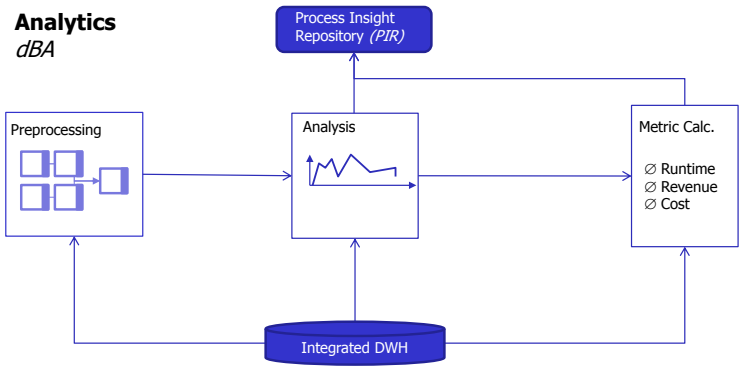

Abbildung 1.3: Prozessanalyse

Der Prozess ist ein Geschäftsprozess, der als *eine Kette von logischen verbundenen, wiederholenden Tätigkeiten* definiert werden kann [**?**]. Sie verwertet die Mittel der Organisation, um einen Gegenstand zum Zweck zu raffinieren und angegebene und messbare Ergebnisse oder Produkte für innere oder äußerliche Kunden zu erreichen. Diese Tätigkeiten schließen die Verarbeitung einer Anwendung und den planenden Prozess ein [17]. Prozess Analyse ist eine Annäherung, die den Prozessanalysten hilf, die Leistung ihrer Geschäftsvolumen zu verbessern. Es kann als ein Meilenstein in der andauernden Verbesserung betrachtet werden. [14] (s. Abbildung 1.3) [16].
Datenvorverarbeitung (engl. Preprocessing) beschreibt jede Art der Verarbeitung von Daten, um sie für eine andere Verarbeitung vorzubereiten, wie z.B. die Aktivitäten im Prozess. Es gibt eine Reihe von verschiedenen Werkzeugen und Methoden, die beim Preprocessing verwendet werden [**?**].
Zu diesem Zweck nutzen wir eine Reihe von speziellen Data Mining Techniken, wie die mehrdimensionale Assoziationsregel oder Klassifikationsbäume. Data Mining ist der Prozess, Daten von verschiedenen Perspektiven zu analysieren. Es beinhaltet nützliche Informationen zur Erhöhung von Einnahmen bzw. Verringerung von Kosten. Daten, die

Software abbauen, sind eines der mehreren Werkzeuge zur Analyse von Daten. Es erlaubt Benutzern, Daten von vielen verschiedenen Dimensionen zu analysieren, sie zu kategorisieren und die identifizierten Beziehungen zusammenzufassen. Technisch ist das der Prozess, um Korrelationen oder Muster unter Dutzenden von Feldern in großen relationalen Datenbanken zu finden. Die Ergebnisse der Analyse sind in dem Prozess *Insight* Lager (engl. Repository) gespeichert. Sie werden in der Hauptdataquelle verwendet. Diese Schicht enthält auch Prozess Matching Funktionen für Zeitoptimierung und statische Prozessanalysemethoden.

1.2.3 Prozessoptimierung

Die aktuelle Prozessoptimierung wird mit einem Katalog von formalisierten Optimierungsmustern durchgeführt [14] [16].

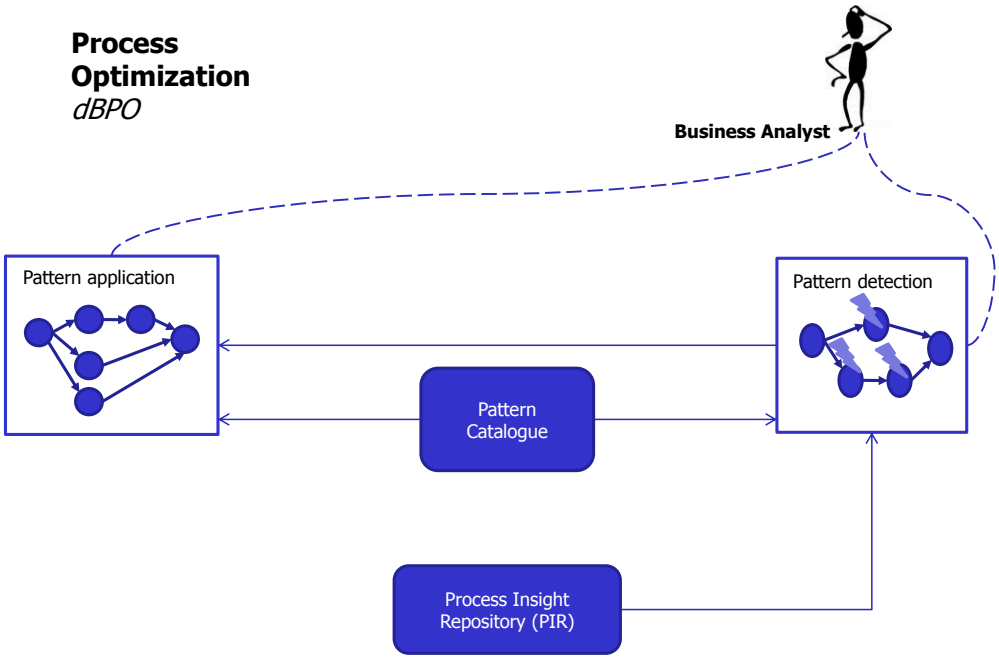

Abbildung 1.4: Prozessoptimierung

Pattern Detection heißt die automatische Identifikation von Zahlen, Zeichen, Formen und Mustern ohne aktive menschliche Teilnahme am Entscheidungsprozess.(s. Abbildung 1.4). *Pattern Catalogue* ist eine Sammlung von Techniken, die die Optimierung von Prozessen formalisiert und speichert, um automatisch angewendet zu werden. *Pattern*

Application ist die Anwendung für diese Techniken.
Die Techniken der Prozessoptimierung sind am häufigsten in der Business Literatur als Richtlinien beschrieben oder werden von *Best Practice* Fallstudien abgeleitet. Diese Beschreibungen sind meistens sehr üblich und abstrakt und bieten wenig bis gar keine Hinweise, wie man tatsächlich die beschriebenen Techniken ableitet. Andererseits hat die Forschung der Informatik einige Optimierungen in algorithmischer Form in die formalen Prozessmodelle beschrieben. Diese Algorithmen sind so präzise, dass sie automatisch ausgeführt werden und die Optimierung der Ergebnisse sichern.
Für die Plattform wählt die dBOP daher einen dritten Ansatz, der die Vorteile der beiden oben diskutierten miteinander zu verknüpfen sucht. Wir übersetzen das Optimierungsverfahren in einen gemeinsamen Formalismus und beschreiben die Anwendung als "Muster". Dieses Muster sind im Wesentlichen Heuristiken, also im Grunde willkürliche Prozessdomäne, die angewendet werden können.

1.3 Einführung Ähnlichkeitsmessung

Viele Unternehmen nutzen Werkzeuge für die Dokumentation ihrer Geschäftsprozesse. Aufgrund betrieblicher Vielfalt von großen Unternehmen, gibt es oft mehrere Tausend von Prozessmodellen, die in der Datenbank des Modellierungswerkzeugs gespeichert sind. Diese Modelle stellen einen wertvollen Bereich für Business Analyse und Unterstützung von System Design dar. In Organisationen mit hohem Business Process Management sind solche Prozessmodelle in eigenen Verzeichnissen (Speicherräume) im Prozessmodell zentral gespeichert und beschrieben [5], [21], [24], [23].
Aufgrund der großen Konkurrenz, ist es am wichtigsten, sich schneller und qualitativer als die Konkurrenzunternehmen anzupassen. Hier sollen Methoden gefunden werden, um die Ähnlichkeit zwischen Teilen im Prozess schnell zu berechnen. Außerdem soll - bevor ein neuer Prozess in einem Speicherraum (engl. Repository) gespeichert wird - zuerst kontrolliert werden, ob es zum ersten Mal gespeichert wird. So können Unternehmen einfacher entscheiden, welche Pakete den aktuellen Operationen am besten entsprechen. Ähnliche Modelle und die entsprechenden Geschäfte können dann in einen Prozess eingebunden werden.
Die Ähnlichkeit zwischen Paaren von Prozessmodellen berücksichtigt folgende Bestandteile eines Prozessmodells:

- die Aufgaben der Modellelemente (Aktivitäten)
- die Graphstruktur
- die Semantik der Ausführung

1.3.1 Definition

Ähnlichkeitsmessung ist die Auswahl und/oder Kombination von Verfahren zur Suche der Similarität zwischen ganzen Prozessen oder Teile davon. Man berücksichtigt bestimmte

Eigenschaften zwischen ihnen und sucht nach dem Grad der Ähnlichkeitsmetrik bzgl. dieser Eigenschaften. Dieses Verfahren bewertet diese Gleichartigkeit mittels einer Skala von 0 bis 1, wobei 0 keine Ähnlichkeit und 1 volle Ähnlichkeit bedeutet (s. Abschnitt 2.2.1), [5].

1.3.2 Eigenschaften der Ähnlichkeitsmessung und Definition der Ähnlichkeitsfunktion

Eine Ähnlichkeitsmessung hat folgende Eigenschaften :

- Eine Ähnlichkeitsmessung soll Merkmalen desselben Typs haben.
- Die Eigenschaften, die geprüft werden, sollen in allen Merkmalen beinhaltet sein.
- Die Anzahl der zu vergleichenden Merkmale ist nicht begrenzt. Man darf mehrere Merkmale miteinander vergleichen.

Die mathematische Funktion einer Ähnlichkeitsmessung heißt Ähnlichkeitsfunktion. Das Ergebnis des Vergleichs zwischen zwei Teilen eines Geschäftsprozesses schwankt zwischen Wert 0 und 1. Wenn sie identisch sind, haben sie eine Ähnlichkeit von 100% und wenn keine Ähnlichkeit besteht ist das Ergebnis 0%.
Die Funktion zur Bestimmung der Ähnlichkeit definiert man folgenderweise :
Seien a und b zwei Teile eines Prozesses.
Sei sim die Ähnlichkeitsfunktion :

$$sim(a, b) = \alpha \text{ , wobei } \alpha \in [0,1] \text{ (1= identisch, 0= keine Ähnlichkeit)}$$

1.3.3 Anwendung einer Aggregatfunktion

Die Ähnlichkeitsmessung basiert auf verschiedenen Metriken. Die korrekte Auswahl der richtigen Metriken ist subjektiv. Man sucht das Verfahren nach Prioritäten und Bedürfnissen aus. Manchmal besteht das Ergebnis aus einer Kombination mehrerer Ähnlichkeitsmaße.
Es muss eine Funktion gefunden werden, die die Kombination aller Metriken mit abhängigen vom Nutzer benutzten Gewichtungen, vorstellt. Das wird durch die sogenannte *Aggregatfunktion* repräsentiert. Eine Aggregatfunktion ist eine Funktion, die eine Berechnung auf einen Satz von Werten durchführt, und nicht auf einen einzigen Wert [**?**].
Man verwendet Aggregatfunktionen häufig in Datenbanken und Tabellenkalkulationen. Sie sind der Mittelwert oder die Summe einer Reihe von Zahlen [**?**]. Die Berechnung von einer Aggregatfunktion liefert einen einzelnen Wert aus mehreren Werten. Hier werden wir beim Vorstellen der Metriken eine gesamte Funktion definieren, die alle Verfahren umschließt.

1.4 Zielsetzung und Struktur der Arbeit

Ziel der Diplomarbeit ist die Konzeptionierung und Implementierung einer Komponente innerhalb der dBOP-Plattform zum Matching von Prozessmodellen. Durch diese Matching Komponente kann die Ähnlichkeit von Prozessen, Prozessfragmenten und einzelnen Aktivitäten von Prozessen gemessen werden. Diese Ähnlichkeitsmessung wiederum kommt zum Einen für die Umsetzung bestimmter Optimierungsmuster zum Einsatz (z.B. zur Eliminierung doppelter/redundanter Aktivitäten). Zum Anderen wird sie verwendet, um die Ausführungsergebnisse existierender Prozesse auf neue Prozessmodelle zu übertragen und so eine Optimierung derselbigen bereits zu Entwicklungskosten zu ermöglichen. Zu diesem Zweck soll im Rahmen der Diplomarbeit das Verfahren sowie die Metriken zur Ähnlichkeitsmessung definiert werden. Des Weiteren soll eine Komponente implementiert werden, welche das Ähnlichkeitsmessungsverfahren umsetzt. Die Komponente soll dabei in den existierenden dBOP-Editor integriert werden.
Der Aufbau der Arbeit gliedert sich wie folgt: Im aktuellen Kapitel erfolgt die Einführung in die Motivation und die allgemeinen Konzepte der Arbeit. Als nächstes stellt Kapitel 2 die formalen Grundlagen der Arbeit sowie verwandte Arbeiten vor. In Kapitel 3 werden die Teile eines Prozesses ausführlich beschrieben und die Arten von Ähnlichkeitsverfahren detailliert mit Beispielen präsentiert. Weiterhin in Kapitel 4 folgt die Beschreibung eines Prototyps zur Anwendung der Ähnlichkeitsmetriken. Als letztes werden Verbesserungsvorschläge und eine letzte Zusammenfassung präsentiert.

2 Grundlagen und verwandte Arbeiten

2.1 Definition und Eigenschaften eines Prozessmetamodells

Ein Prozess ist eine strukturierte Abfolge von Aktivitäten mit gewünschten und erwarteten spezifischen Leistungen. Es verfolgt ein oder mehrere Ziele und es existieren messbare Eingabe- und Ausgabedaten (s. Abschnitt 2.4.5), die sowohl materieller (z.B. Rohstoffe) als auch immaterieller Art (z.B. Informationen) sein können. Zur Erzeugung der Ergebnisse werden Ressourcen eingesetzt (z.B. menschliche Arbeit) [10].

Einführung in die Prozessmodellierung

Prozessmodelle bereiten die Geschäftsprozesse einfacher und zwecksorientiert vor und sind gezielt auf die Analyse, Ausführung, Dokumentation und Kommunikation von Prozessen. Ein Modell hat die Aufgabe, Informationen eines bestimmten Teilaspektes möglichst formal und eindeutig zu beschreiben. Ein stark formalisiertes Artefakt wird als Modell bezeichnet. Die Informationsstuktur, die auf abstrakter Ebene für einen Artefakttyp festgelegt ist, ist dementsprechend das dem Artefakt zugrunde liegende Metamodell. Metamodell ist ein Modell, das man instanziiert um andere Modelle daraus zu erhalten. [10]
Es definiert eine Reihe von abstrakten Business Process Elementen für die Spezifikation von ausführbaren Geschäftsprozessen, die innerhalb eines Unternehmens geführt werden und kann zwischen unabhängigen Prozessen in verschiedenen Geschäftsbereichen oder Unternehmen zusammenarbeiten. Wenn ein Geschäftsprozess modelliert wird, werden alle möglichen Formen der Ausführung durch ein Prozessmodell beschrieben. Jeder laufende Prozess entspricht einer Instanz des Prozessmodells.
Weiterhin definieren Metamodelle Konzepte, Beziehungen und Semantik für den Austausch von Modellen zwischen verschiedenen Benutzermodellierungstools. Sie beschreiben die Eigenschaften und das Verhalten von Instanzen eines zugehörigen Datenmodells. Das Metamodell ist Teil einer 4-schichtigen Hierarchie. Diese Architektur stellt eine Hierarchie dar, die die einzelnen Modellarten definiert. Die abgebildete Realwelt (Daten-Schicht) bildet die unterste Schicht. Sie spezifiziert einen speziellen Fall eines Anwendungsbereichs. Diese Daten werden durch Modelle beschrieben (Modellen-Schicht). Die einzelnen Bausteine des Modells werden mit einem Metamodell beschrieben (Metamodellen-Schicht). Aussagen über Metamodelle gewinnt man mithilfe der Metametamodellen-Schicht. Sie definiert die Sprache zur Spezifikation des Metamodells [8], [10], [**?**].

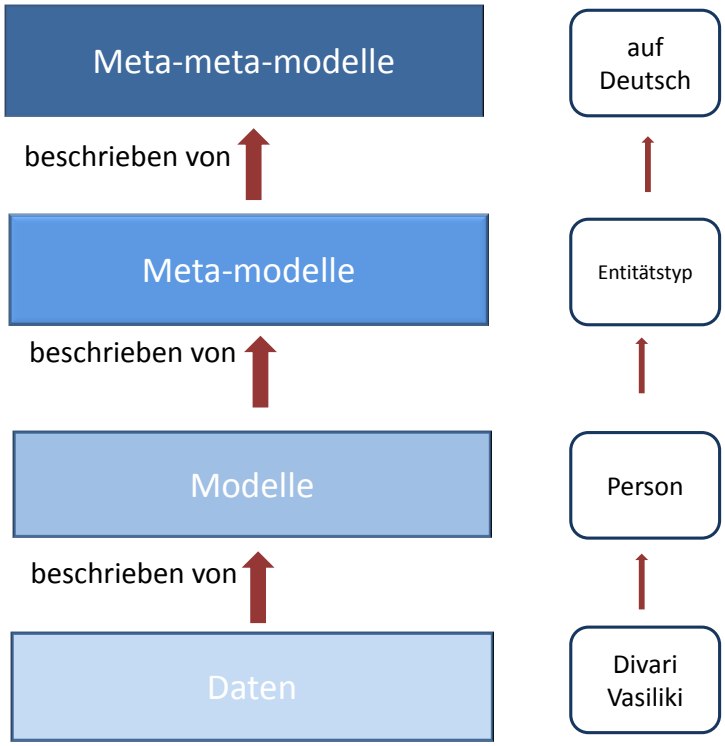

Abbildung 2.1: 4-Schichten Architektur

Die Konstrukte und die damit verbundenen Sprachen, die verwendet werden, um ein Datenmodell zu formulieren, sowie die genaue Beschreibung der Eigenschaften und das Verhalten der Instanzen einer damit verbundenen Daten, werden als Metamodell bezeichnet. Das Metamodell beschreibt die Eigenschaften der Instanzen.
Die Konstrukte und die Sprache werden von einem Metamodell als Syntax zur Verfügung gestellt und die Beschreibung der Eigenschaften und das Verhalten der Instanzen werden nach dieser Syntax ihrer Semantik gebaut.

Das Metamodell in dem Bereich von Geschäftsprozessen ist aus folgenden Gründen wichtig :

- Die Fähigkeit zur Integration von Prozessmodellen für Workflow Management Prozesse, zur möglichst hohen Automatisierung von Geschäftsprozessen und zur Zusammenarbeit zwischen den Geschäftseinheiten.
- Die Möglichkeit, in Geschäftsprozessen Spezifikationen zwischen Modellierungstools auszutauschen.
- Das Prozess-Metamodell bezieht sich auf die Abstraktion der grundlegenden Ele-

mente und auf die Regeln des Prozesses und es wird angewendet, um die Modellierung des Prozesses zu führen.

- Ein Metamodell von ausreichender formaler Mächtigkeit erlaubt ein Reasoning über die Prozesselemente.

2.1.1 Definition

In dieser Diplomarbeit wird ein bestimmtes Metamodell dargestellt, um die Konzeption der Prozessähnlichkeit präsentieren zu können. Dieses Prozess- und zusätzlich Ressource Metamodell wird verwendet, um das Schema eines Geschäftsprozesses zu beschreiben. Der Beispielprozess dieser Arbeit wird entsprechend konstruiert. Funktionen und Eigenschaften des Metamodells werden angesetzt, um die einzelnen Komponenten des Prozesses zu analysieren. Diese Teile des Metamodells werden später in unseren Beispielen angewendet und pro Teil intensiv untersucht.
Ein Prozessmodellgraph [14] ist ein Tupel (V, N, E, ι, C), Ω ist ein Set von Textlabel und Λ eine Funktion, bei der :

1. V eine endliche Menge von Prozess Data Elementen (auch genannt $Variablen$) ist
2. N eine endliche Menge von Aktivitäten ist
3. $E \subseteq N \cup N \cup C$ die Menge der Kontrollkonnektoren ist
4. $\iota \subseteq N \cup C \cup \{G\} \rightarrow \mathcal{P}(V)$ der Input Data Map, mit $\mathcal{P}(V)$ die Potenzmenge über V ist
5. $o \subseteq N \cup \{G\} \rightarrow \mathcal{P}(V)$ ist der Output Data Map
6. C die endliche Menge von Konditionen bzgl. Kontrollkonnektoren ist
7. $\Lambda : V \cup N \subseteq C \subseteq \{G\} \rightarrow \Omega$ ordnet Labels zu Prozesselemente zu
8. $\rho : N \rightarrow \wp R$ die Ressourcen Usage Map ist
9. $E_D \subseteq (N \cup N)$ die Menge der Daten Konnektoren, die bei Abhängigkeit zwischen ι und o ist
10. $E_R \subseteq N_A \times N_A$ die Menge der Ressource Konnektoren ist, die Ressource-Abhängigkeiten zwischen Aktivitäten zeigt.

Weiterhin definieren wir die Funktion AVGDUR : $N \rightarrow \mathbb{R}$ als die durchschnittliche Dauer der Ausführung einer Aktivität und FREQ : $N \rightarrow [0, 1]$ die Frequenz des Auftretens einer Aktivität in einem Prozess.

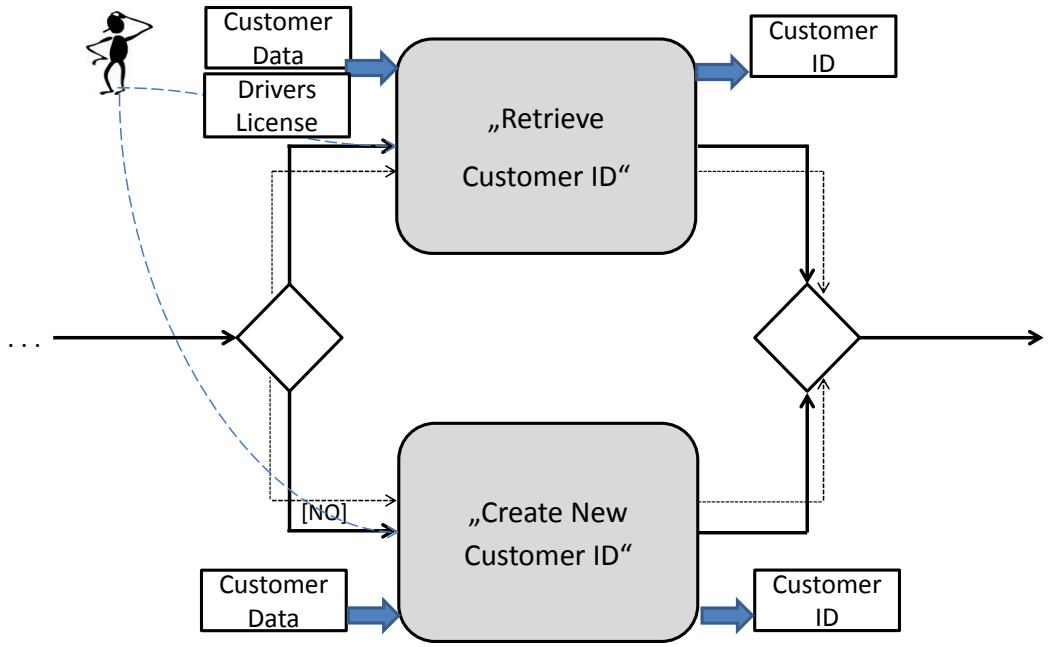

Abbildung 2.2: Beispiel Modell

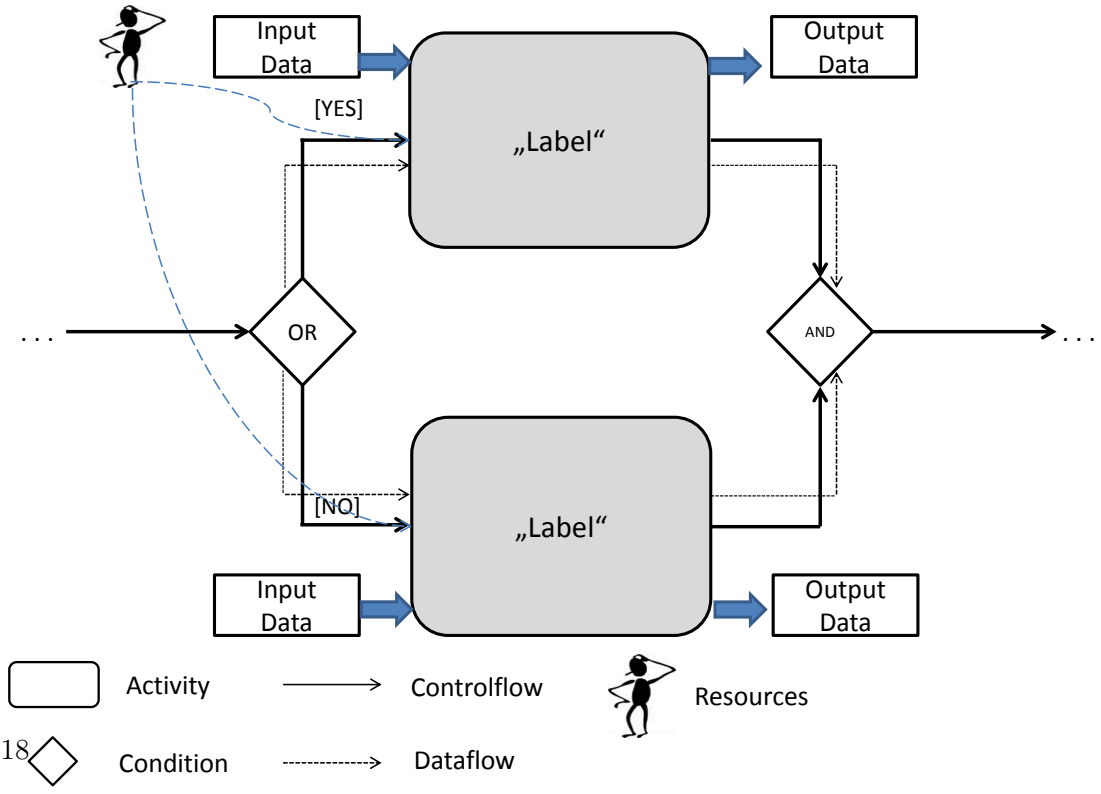

Abbildung 2.3: Beispiel Metamodell

2.1.2 Teile eines Metamodells

In diesem Kapitel werden die Teile eines Metamodells detailliert beschrieben :

Prozessdaten

In einem Prozessmodell beinhalten die Aktivitäten und die Konditionen Daten. Diese Daten enthalten Informationen, die als Eingabe in Aktivitäten benutzt werden, um diese Aktivitäten starten zu können. Die Konditionen benötigen auch Daten, um zu entscheiden, wann eine Aktivität beendet ist und fließen zwischen allen Aktivitäten. Metamodell bietet eine Menge V, wo alle Prozessdaten gesammelt werden und sich mit dem Prozessmodel verbindet. Eine Einheit v∈V heißt Datenelement oder Variable. Jedes Datenelement hat einen Namen und eine Struktur. Der Name erlaubt Zugriff auf sich, und die Struktur ordnet die Art der Zusammensetzung einer Menge aus Daten und die Menge der Relationen bzw. Operationen, welche die Daten miteinander verknüpfen.
Daten können technisch mit verschiedener Art und Weise gespeichert werden, z.B. entweder in Dateien oder in Datenbanken. Für das Metamodell hat diese Aktion die folgenden Konsequenzen: Auf einer relativ abstrakten Ebene des Modells genügt die Einführung eines Entitätstyps „Daten“.
Die Unterscheidung der Datenarten kann mittels Attributen erfolgen. Zum Beispiel können Attribute *Pfad* und *Dateiname* bestimmte Dateien näher beschreiben. Zusätzlich kann durch die Verwendung eines Attributs *Parameter* die Menge der Daten kontextabhängig eingeschränkt werden.

Aktivitäten

Ein Prozess besteht aus Aktivitäten, wird also in Aktivitäten zerlegt. Diese gelten als die kleinste Einheit eines Prozesses. Aktivität ist die Verteilung eines Projekts in eine Reihe von Aufgaben. Sie ist nötig, um die bisherigen Ergebnisse als vollständig zu bezeichnen. Ihre wichtigsten Merkmale sind die Dauer ihrer Ausführung, die davor definiert werden muss, die logische Beziehung mit dem Rest der Aktivitäten im Prozess, die Versorgung mit Daten (s. Abschnitt 2.4.5) und die Nutzung von Ressourcen. Zusammen modellieren sie die Logik der Geschäftsprozesse, die sogenannte Ablauflogik.
Die Aktivitäten werden durch Namen definiert. Sie entstehen entweder manuell oder automatisch mithilfe eines Computers. D.h sie können eine Aktion sein, die manuell bestimmt wird, oder eine vom Computer unterstützte automatisierte Operation. Eine Aktivität im Prozess nimmt menschliche und/oder maschinelle Unterstützung zur Verarbeitung der Informationen und Ausführung der Aktionen in Anspruch. Die Teilnehmer werden den Aktivitäten zugeordnet.
Diese kleinste Einheit des Prozesses enthält eine Reihe einzelner Aufgaben. Die Aktionen sind erst dann vollständig, nachdem sie abgeschlossen sind. Sie werden durch eine Reihe von spezifischen Eingaben im Prozess aktiviert. Diese Eingabedaten werden verwendet, um eine Operation zu erledigen und Daten auszugeben, die eventuell für andere Aktivitäten nötig sind. Diese gelten als offizielles Endergebnis des ganzen Prozesses. Die Aktivitäten können auch parallel laufen. Sie können weiter durch die Verwendung einer

Reihe von Tools und Techniken effizienter erreicht werden. Nötig sind die gegebenen Vorlagen für die Identifikation und Dokumentation der geplanten Tätigkeiten. Der Prozess mit allen Aktivitäten kann dann die Resultate auf der untersten Ebene (Arbeitspaket) in der Projektstrukturplanung identifizieren.

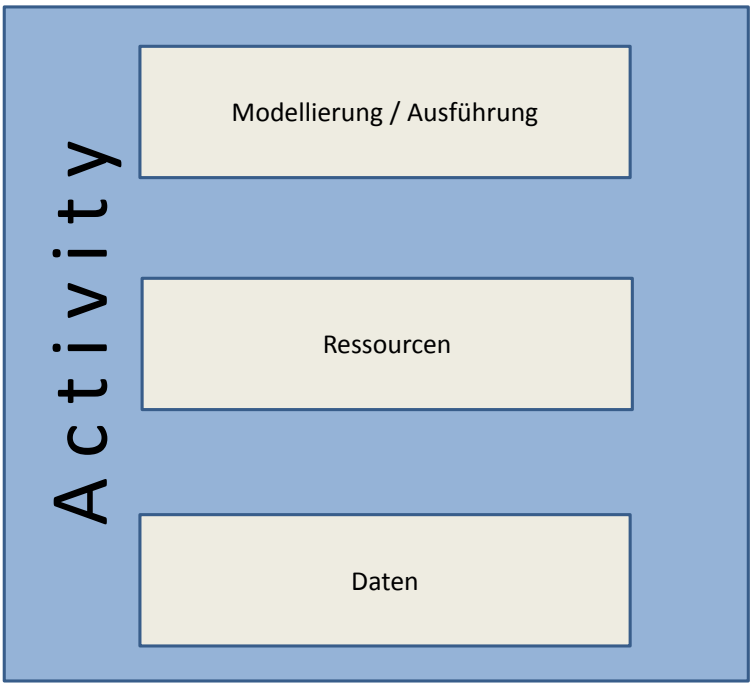

Abbildung 2.4: Struktur einer Aktivität

Datenkonnektoren

Eine Aktivität enthält Daten. Diese Daten sind in Speicherräume (engl. Container) gespeichert, genauso wie die Konditionen. Es gibt zwei Arten von Speicherräumen, die Eingabe- (engl. Inputs) und die Ausgabe-Speicherräume (engl. Outputs). Es soll spezifiziert werden, welche Daten zwischen den Aktivitäten getauscht werden. Das erfolgt durch den sogenannten Datenfluss. Das Metamodell beschreibt, welche Aktivitäten welche Daten von anderen Aktivitäten erwarten und wie die Datenelemente von einem *Input Container* von anderen *Output Containers* zusammengesetzt werden (Daten-*Mapping*). Der Weg einer Datennachricht von ihrer Entstehung bis zu ihrem Bestimmungsort, einschließlich aller Knoten, durch die er folgt, ist ein Datenfluss. Es wird für jede Aktivität ein Input Container und ein Output Container modelliert, die Informationen beinhalten,

wo die Daten drin stehen. Eingabe- und Ausgabedaten werden in Container gespeichert. Sie sind Daten, mit dem ein Prozess angestoßen wird [?], [?].

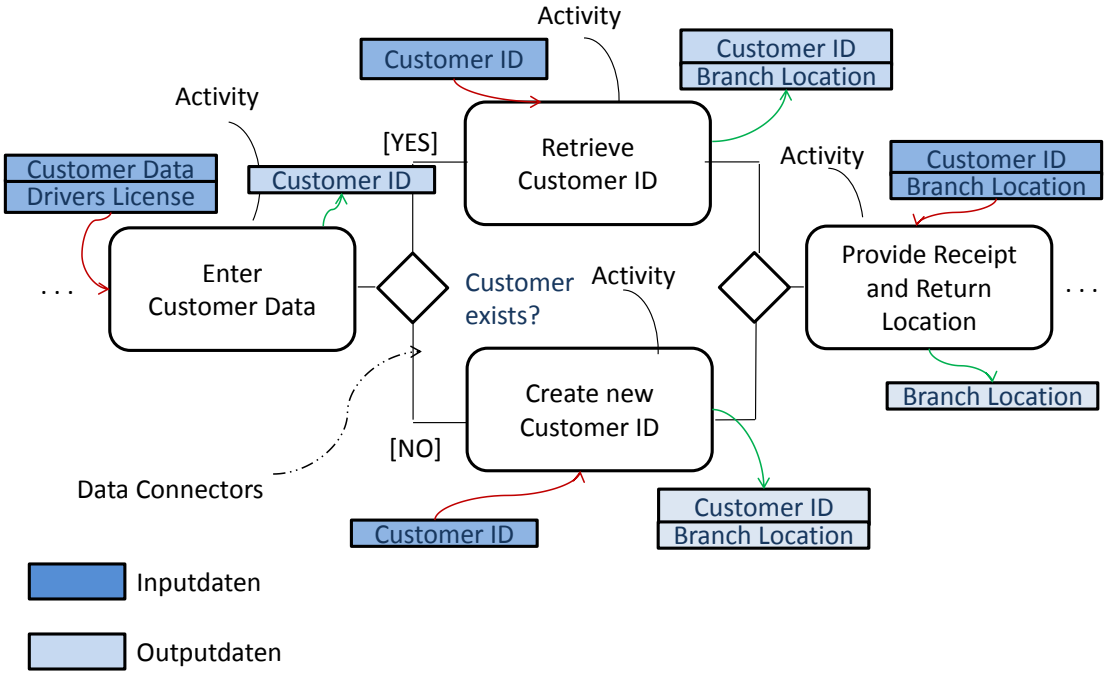

Abbildung 2.5: Datenfluss eines Prozesses

Kontrollkonnektoren

Ein Kontrollfluss entspricht dem Arbeitslauf eines Business Prozesses. Das Metamodell definiert die gültige Reihenfolge der Abarbeitung der Aktivitäten. Kontrollflüsse werden durch Kontrollkonnektoren modelliert. Es gibt Regeln in Form von booleschen Bedingungen, wo entschieden wird, welche Aktivität wann ausgeführt wird. Kontrollfluss wird sogar als Ablauflogik eines Workflows definiert. In einer Prozessmodellierung wird ein Graph dargestellt, der aus Knoten und Kanten besteht. Die Knoten der Graphen bezeichnet man als Aktivitäten. Im Workflow sind Aktivitäten Arbeitsschritte, die ablaufen müssen. Eine Kante zwischen zwei Aktivitäten modelliert den potenziellen Kontrollfluss. Eine Aktivität kann erst dann ausgeführt werden, wenn sie davor erfolgreich beendet wurde. Es gibt also eine Abhängigkeit zwischen Aktivitäten [?].

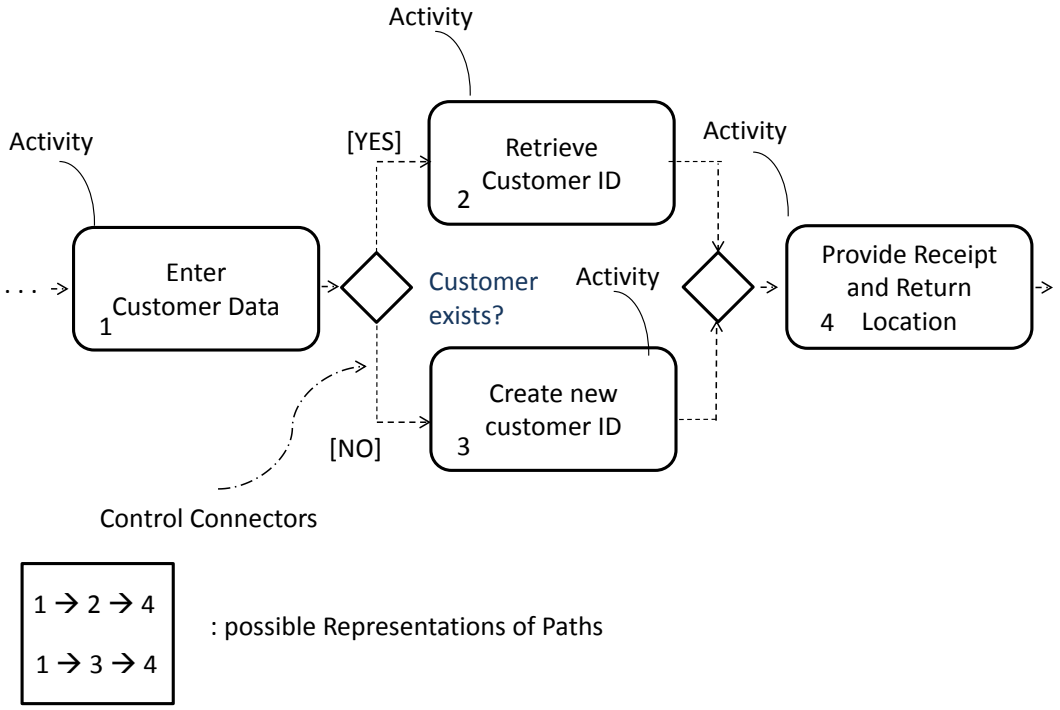

Abbildung 2.6: Kontrollfluss eines Prozesses

Konditionen

Eine Bedingung wird verwendet, um festzustellen, wann eine Aktivität im Prozess ausgeführt werden darf. Kondition gehört zu dem Kontrollfluss, der die ordnungsgemäße Folge der Abarbeitung von Aktivitäten definiert. (s. Abschnitt 2.1.2) Wir unterscheiden zwischen zwei Arten von Bedingungen : die UND-Bedingung und die ODER-Bedingung. Die UND-Bedingung stellt eine Kombination von zwei oder mehreren Bedingungen dar, wobei eine Übereinstimmung vorliegt, wenn alle Bedingungen wahr sind. Bei der ODER-Bedingung liegt eine Übereinstimmung vor, wenn eine der Bedingungen wahr ist.

Ressourcen

Ressourcen sind Mittel, die Aufgabenträgern zur Erfüllung ihrer Aufgaben zur Verfügung stehen, z.B. Arbeitskraft (Menschen) oder Werkzeuge (Maschinen). Jede Ressource kann einer oder mehreren Verantwortlichkeiten zugeordnet werden. Dadurch werden Zugriffsrelationen, d.h. Berechtigungen einzelner Mitarbeiter für die Nutzung bestimmter Ressourcen, definiert. Ressourcen können sowohl ein Mitarbeiter, als auch eine Organisationseinheit oder eine Stelle sein. Sie können hierarchisch angeordnet sein und so

anderen Ressourcen über- bzw. untergeordnet sein. Für den Vorgang des Metamodells soll dieser Ansatz etwas weiter verfeinert werden. Dabei geht es darum, für das Modell Entitätstypen und deren Beziehungen zueinander zu identifizieren.

Prozesshierarchie

Ein Prozess besteht aus einer oder mehreren Prozesseinheiten (Prozesselementen). Eine Prozesshierarchie besteht aus einer Reihe von Prozessen und deren Logik. Auf jeder Ebene können Prozesse unterschiedliche Arten von Funktionen repräsentieren.

2.1.3 Aspekte des Metamodells

Jeder Anwendungsfall benötigt spezifische Lösungen. Viele verschiedene Modellierungssprachen erschweren die Kommunikation zwischen Modellen. Höhere Metamodelle schaffen eine gemeinsame Basis für verschiedene Sprachen und erleichtern damit die Kommunikation zwischen Modellen. Eng damit verbunden ist die Verwendung von Metamodellen als ein Schema für semantische Daten, die in einem Repository gespeichert werden.
Ein Metamodell beschreibt die Regeln und Zwänge der Metatypen und Metabeziehungen. Es ist ein Modell einer (Modellierungs-) Sprache. Ein Metamodell stellt kein Modell oder eine Reihe von Modellen dar, sondern die abstrakte Syntax einer Modellierungssprache.

2.1.4 Anforderungen im Rahmen der Modellierung

Verringerung von Kosten und Laufzeit sowie Vergrößerung von Flexibilität und Qualität sind wichtige Outputs der Modellierungsphase. Die Laufzeit definiert wie lange ein Prozess dauert. Es bestehen zwei Sorten von Kosten. Einerseits existieren fixe Kosten, die für die Infrastuktur der Firma bestimmt sind und andererseits die variablen, die in Koorelation im Prozess stehen. Zusätzlich existieren die operationalen Kosten, die für die Ausführung einzelner Arbeitsschritte nötig sind.
Man unterscheidet zwischen externer und interner Qualität. Kundenzufriedenheit (extern) und Mitarbeiterzufriedenheit (intern). Flexibilität verleiht die Möglichkeit zur schnellen Änderungsreaktion. Wenn man eine der folgenden Charakteristiken verbessern will, besteht der Nachteil, dass sich die anderen Charakteristiken unter Wert verschlechtern. Will man z.B in der Produktion die Kosten verringern, wird vermutlich die Qualität reduziert. Das Modell, das hilft, die negative Dynamik in solchen Beziehungen zu erkennen, heißt Teufelskreis. (s. Abbildung 2.7).

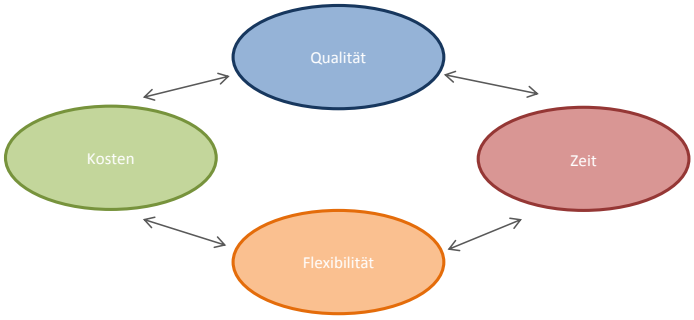

Abbildung 2.7: Teufelskreis

Die Modellierung von Geschäftsprozessen wird grafisch dargestellt. Dies hilft zur Verbesserung einiger Kategorien. Die erste Anforderung im Rahmen der Modellierung ist die Analyse der Geschäftstätigkeit. Ein Prozess kann als eine logische Reihe von Transaktionen betrachtet werden. Diese Transaktionen wandeln die Eingaben zu erforderlichen Ergebnissen um. Der Prozess ist ein "Business Process", also *eine Kette von logischen verbundenden Tätigkeiten*, der die Ressourcen zum Zweck der Erreichung messbarer Ergebnisse oder Produkte nutzt. Prozessanalyse ist ein Ansatz, der die Leistung der Geschäftstätigkeit unterstützt. Es kann ein Meilenstein in der kontinuierlichen Verbesserung werden.

2.2 Ähnlichkeitsfunktionen

Wenn wir nun Teile eines Prozesses miteinander vergleichen wollen, um eine etwaige Gemeinsamkeit festzustellen, müssen wir Kriterien definieren und formulieren, die uns zu sagen gestatten, wann wir zwei Merkmale als gleich oder verschieden feststellen und beurteilen können. Aus diesem Grund definiert man eine sogenannte Ähnlichkeitsfunktion (engl. Similarity Function), die ein Maß für die Ähnlichkeit zweier Merkmale angibt: Je größer der Wert der Ähnlichkeitsfunktion ist, desto ähnlicher sind sie.
Häufig bestimmt man die Ähnlichkeitsfunktion so, dass 1 für die maximale Ähnlichkeit steht (beide Merkmale sind identisch) und 0 für die kleinste Ähnlichkeit (haben nichts gemeinsam). Eine andere Möglichkeit Merkmale zu vergleichen ist eine Unähnlichkeitsfunktion, d.h. Abstands- oder Distanzfunktion (engl. Distance Function) anzugeben, die den Unterschied zwischen den zu vergleichenden Charakteristiken anzeigt. Je nach Art der Merkmale kann die eine oder andere Methode einfacher sein. Um ähnliche Geschäftsprozesse, wie Prozesse ähnlicher Geschäftseinheiten oder ähnliche Organisationen zu erschaffen, müssen die Ähnlichkeiten und Unterschiede zwischen diesen Geschäftsprozessen erkannt werden und beseitigt werden [21].

2.2.1 Eigenschaften einer Ähnlichkeitsfunktion

Jede Metrik zur Suche der Ähnlichkeit hat folgende Eigenschaften:

- Die *Identitätseigenschaft* hat eine Aktivität, wenn sie volle Ähnlichkeit bzw. null Unterschied mit sich selbst hat (reflexiv).

$$sim\ (a,\ a) = 1$$

$$sim\ (a,\ b) \geq 0$$

- Die zwei zu vergleichenden Teile sind symmetrisch zueinander (*Symmetrie*). Zwei Aktivitäten sind immer gleich, unabhängig davon, welche als Ausgangsmuster oder Vergleichsmuster berücksichtigt wird.

$$sim\ (a,\ b) = SIM\ (b,\ a)$$

Es werden in folgenden Kapiteln unterschiedliche Metriken zur Berechnung dieser Ähnlichkeit vorgestellt. Sie basieren auf verschiedene Eigenschaften der vergleichenden Objekte und werden detailliert definiert und durch Beispiele verständlicher gemacht.

2.2.2 Ähnlichkeitsfunktion für Prozessmodelle

Wir haben die Möglichkeit, diese Messung mithilfe der Ähnlichkeitsfunktion an ganzen Prozessen anzuwenden. Um zwei Prozesse nach Ähnlichkeit zu vergleichen, müssen wir davor schon die einzelnen Komponenten oder Teile von Prozessen vergleichen können. (s. Abschnitt 2.5) Wir können dann diese Metriken problemlos in den gesamten Geschäftsprozessen anwenden.
Zur Exaktheit kombinieren wir alle Metriken zu einer gesamten Metrik, die eine Mischung prozentualer Summe von Metriken sich präsentiert. Wir werden uns näher in Abschnitt **??** mit diesem Thema beschäftigen.

2.3 Verwandte Arbeiten

Veschiedene Ähnlichkeitsverfahren für Geschäftsprozesse geben dem Benutzer den Auswahl, nach Wunsch und Notwendigkeit, unterschiedliche Komponente von Prozessen zu vergleichen. Wissenschaftliche Fachliteraturen zur Verwendung von Ähnlichkeitsverfahren in Prozesse sind vorhanden. Im Feld vom Prozess Matching, haben bis jetzt viele Leute Gedanken gemacht. Um die Ähnlichkeitsbeziehung zwischen zwei Merkmale zu berechnen, soll entweder nach Ähnlichkeit oder nach Unterschiede gesucht. Im Prinzip sind diese zwei Definitionen stark verbunden. Eine ganze Reihe von Ähnlichkeitsverfahren wurde definiert. Die meisten beziehen sich auf der Analyse-Phase der Aktivitäten.

Das bekannteste Ähnlichkeitsverfahren ist dafür das Label Ähnlichkeitsverfahren (s. Abschnitt 2.5.3). Die Empfehlung von Untersuchungen der semantischen Ähnlichkeitsmetrik ist zahlreic, wie z.B in [6], [11] [20], da diese Metrik leicht und effizient, und schnell Ergebnisse anzeigt. In [5] werden die Verfahren definiert, Formel präsentiert und Beispiele in einem bestimmten Prozessbeispiel durchgeführt. Mit syntaktischer und semantischer Ähnlichkeit beschäftigen sich auch die Publikationen von [6], [11], [20], [11], wobei die definierten Formeln der Metriken, analog sind. Einige Maßnahmen, die später erwähnt werden, haben eine Allgemeinheit: Sie alle hängen von einer hierarchischen Struktur, wie z.B die semantische Ähnlichkeitsmessung mittels WordNet Anwendung, wie z.B bei [25], [22]. Ressource Ähnlichkeitsmetrik wurde im [19] analysiert. Weitergehend existiert gegenwärtig das kontextuelle Ähnlichkeitsverfahren. Leider, gibt es aber nur wenige fundamentale Arbeiten auf diesem Gebiet ([9]).
Die Berechnung von ganzen Geschäftsprozesse, die die Form eines Graphens haben, ist noch zu verbessern und wird meistens theoretisch durch Standardverfahren über Distanz-Messung(s. Kap **??** zwischen solcher Geschäftsprozesse. Außer Analyse-Bereich, einige Autoren haben sich mit Ähnlichkeitsverfahren über die Ausführung in einem Prozess. [14], [16], [5], [4], [26].

Die folgende Tabelle wird uns helfen ,ein Überblick über die Arten von Ähnlichkeitsverfahren und die Markierung der Bereiche auf die Forschung verschaffen.

Sim -> Quelle:	Label	Kontextuell	Daten	Ressourcen	Execution	Behavioural	Histogramm	Strukturell
[1]	x							
[6]	x							x
[8]	x							
[13]	x	x						x
[14]	x							
[18]	x							x
[19]	x	x				x		x
[22]	x	x						
[32]	x							
[33]	x							
Diese Arbeit	x	x	x	x	x		x	x

Abbildung 2.1: Verwandte Arbeiten

2.4 Arten von Metriken

Das Ziel dieser Diplomarbeit ist die Suche nach Ähnlichkeit von Geschäftsprozessen. Der Vergleich zweier Prozesse erfolgt nach vergleichsweiser Beurteilung von kleineren Teilen der Prozesse, nämlich konkrete Aktivitäten und Subprozesse.

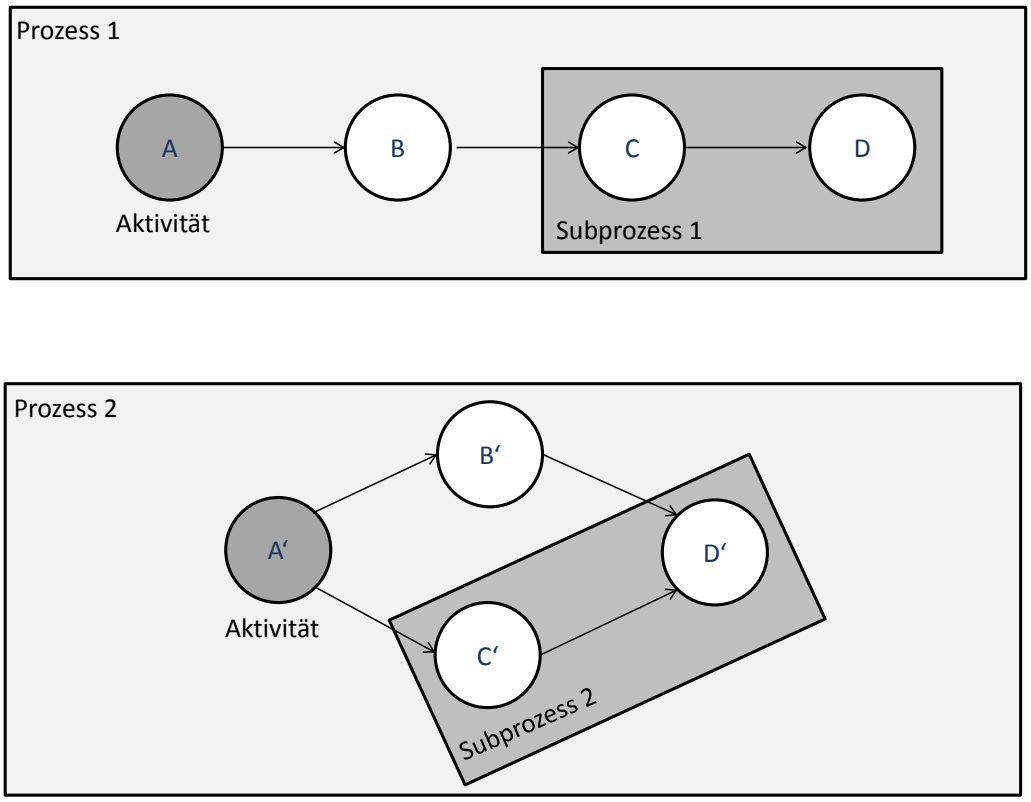

Abbildung 2.1: Teile eines Prozesses

Am Anfang definieren wir einen Beispielprozess. Wir suchen zwei Aktivitäten aus und stellen einen Vergleich an mit allen wichtigen Informationen, die für das angewendete Verfahren nötig sind. Weiterhinaus wird ein zweiter Prozess in späterer Phase definiert, der mit dem Hauptprozess verglichen wird, um die Anwendung von Ähnlichkeitsverfahren zwischen Subprozesse und Prozesse anzeigen zu können.

Alle Eigenschaften und Komponenten dieses Prozesses werden in der folgenden Abbildung angezeigt. Sie dienen zur Anwendung verschiedener Arten von Metriken und führen zu den Ergebnissen der Berechnung.

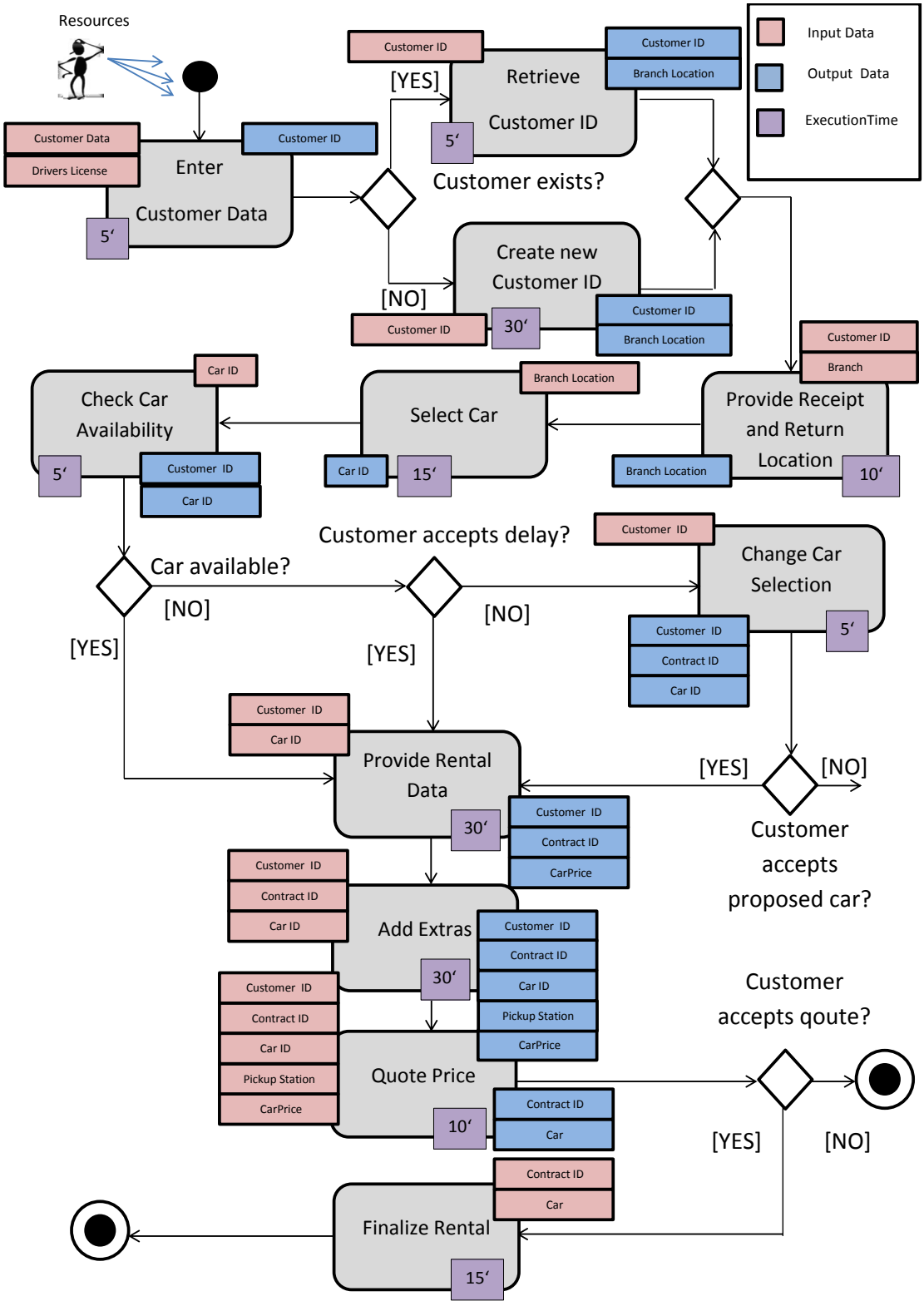

Abbildung 2.2: Prozessmodell (Details)

2.4.1 Aktivität

Aktivitäten sind Bestandteil eines Prozesses. Sie definieren schrittweise die Arbeit, die geleistet wird, um ein Ziel zu erreichen (s. Abschnitte 2.4.1 und 2.1).
Das Metamodell enthält :

1. die Spezifizierung der Eingabe- und Ausgabedaten einer Aktivität
2. das Tool um die Arbeit zu erfüllen
3. die qualifierten Personen
4. die Methoden (Konditionen) zur Feststellung, ob die Arbeit beendet ist oder nicht.

Eine Aktivität ist eine Abstraktion einer Arbeit, die innerhalb eines Prozesses durchgeführt werden muss. Sie bietet alle Informationen, um festzustellen, wer was zu tun hat, mit welchen Daten und mit welcher Art von Werkzeugen. Ein Ressource ist meistens eine Person, manchmal auch eine Maschine, die einen Teil der Arbeit -auch automatisch- verwirklichen kann. Das Metamodell stellt eine Reihe von Agenten dar, die geeignet sind, um die Arbeit auszuführen. Die meisten Aktivitäten dauern meistens einen langen Zeitraum. Es soll möglich sein, die Aktivitäten zu unterbrechen und zu einem späteren Zeitpunkt fortzusetzen.

2.4.2 Subprozess

Ein Subprozess bildet einen Teil des Prozesses (s. Abschnitte 2.4.3 und 2.1)). Er wird definiert als der Teil eines laufenden Geschäftsprozesses, der eine einzige Aufgabe oder eine bestimmte Teilmenge davon hat. Alle verbundenen Subprozesse zusammen bilden den Geschäftsprozess.

2.4.3 Prozess

Viele Firmen suchen ein besseres Verständnis ihrer geschäftlichen und technischen Prozesse (s. Abschnitt 2.1). Das hilft den neuen Mitarbeitern zu verstehen, wie hier der Ablauf ist. Dies ist eine klare Grundlage für den Nachweis kontinuierlicher Prozessverbesserung. Die Art und Weise wie ein Geschäft funktioniert, wird durch Geschäftsprozesse dargestellt.
Prozess ist die Menge aller Informationen, Hilfsmittel, Menschen, Ressourcen und Daten zur erfolgreichen Herstellung eines Produktes oder zur Gewinnung einer Dienstleistung in einem Unternehmen. Er ist eine Reihe von Aktionen, Maßnahmen, Änderungen, Funktionen und Operationen, die durchgeführt werden, zur erfolgreichen Erschaffung von Produkten und/oder Dienste. Ein Geschäftsprozess besteht aus einer Menge von Aktivitäten und deren Beziehungen und Voraussetzungen zum Einstieg, Bearbeitung und zur Beilegung des Prozesses. Informationen über die einzelnen Aktivitäten werden konkret verteilt. Die zuständigen und qualifizierten Teilnehmer sollen in der Lage sein, die Arbeitsschritte auszuführen und sind dem Prozess zugeordnet. Die Darstellung eines Geschäftsprozesses

in einer Form z.B. Modellierung, wird durch ein Workflow-Management-System unterstützt.
Ein Weg, um Prozesse klarer zu verstehen, ist eine grafische Darstellung. Sobald die grafische Darstellung korrekt definiert ist, können zusätzliche Details hinzugefügt werden, um den Prozess noch weiter zu verfeinern.
Die Reihenfolge aller Methoden, die auf jeder Stufe eine oder mehrere Ressourcen verbrauchen (Mitarbeiter, Zeit, Maschinen, Geld) und Eingaben nötig haben (Daten, Material, Teile, etc.), bauen die Aufbauorganisation eines Geschäftsprozesses. Diese Ausgänge dienen dann als Eingabe für die nächste Stufe, bis ein bekanntes Ziel oder Ergebnis erreicht wird. Der Prozess hilft zur Erfassung von innovativen und häufig verwendeten Aktivitäten und zur Bestimmung der Grenzlinie der Aktionen, wie z.B. Rollen und Verantwortlichkeiten. Schließlich spielt Prozess eine wichtige Rolle zur Vermeidung redundanter oder verpasster Arbeit. Durch eine richtige und vollständige Definition von Geschäftsfunktionen können Änderungen rapide in Anwendungen reflektieren, denn das Produkt kann schneller und kostengünstiger erstellt werden.

2.4.4 Ressourcen

Die Ressourcen sind wichtiger Bestandteil der Aktivitäten im Prozess. Eine Ressource ist entweder eine Person oder ein Material, mit dem ein Ziel erreicht werden kann. Sie dient zur Unterstützung eines Prozesses. Die Mitarbeiter und allgemein der Betrieb einer Organisation bilden die Ressourcen, also die Quelle im Geschäftsprozess. Die finanziellen und materiellen Sachmittel einer Organisation stellen auch Ressourcen dar. Sie entscheiden, wer die Arbeitsschritte ausführen kann und sie dienen zur Modellierung der Aufbauorganisation, der Struktur, der Rollen und allgemein der Hierarchie in einem Unternehmen.
Business Ressourcen sind alles, was ein Unternehmen betreibt und damit Geschäfte machen kann. Jedes Unternehmen benötigt Ressourcen zur Herstellung von Gütern oder Dienstleistungen für ihre Kunden. Sie haben eine bemerkenswert wirtschaftliche Bedeutung.
Die wichtigsten Ressourcen für jedes Unternehmen sund ihre Arbeitskräfte, also die Leute mit zahlreichen Fähigkeiten, Anstregungen und Wissen (Humankapital). Die Mitarbeiter haben die Fähigkeit, Rohstoffe zu wertvollen Produkten und Kenntnisse in Dienstleistungen zu verwandeln. Menschliche Arbeitskräfte sind auch die organisatorische Funktion, die Probleme wie Entschädigung, Einstellung, Leistungsmanagement und Ausbildung verbinden. Zur menschlichen Arbeitskraft gehört eine einzelne Person oder ein Angestellter innerhalb der Organisation.

2.4.5 Daten

Daten sind Informationen, die durch einen Geschäftsprozess ständig fließen (s. Abschnitt 2.1.2). Sie werden in Speicherräume gelagert. Der Datenfluss ist verantwortilich für alle diese Schritte bei der Ausführung eines Prozesses. Datenfluss bestimmt wie sich die Daten versorgen (s. 2.1.2).

2.5 Ähnlichkeitsmessverfahren zwischen Prozessmodellelementen (Aktivitäten)

In diesem Abschnitt stellen wir Verfahren zur Messung der Ähnlichkeit zweier Aktivitäten vor. Es folgt ein grafischer Überblick über alle vorgestellten Metriken :

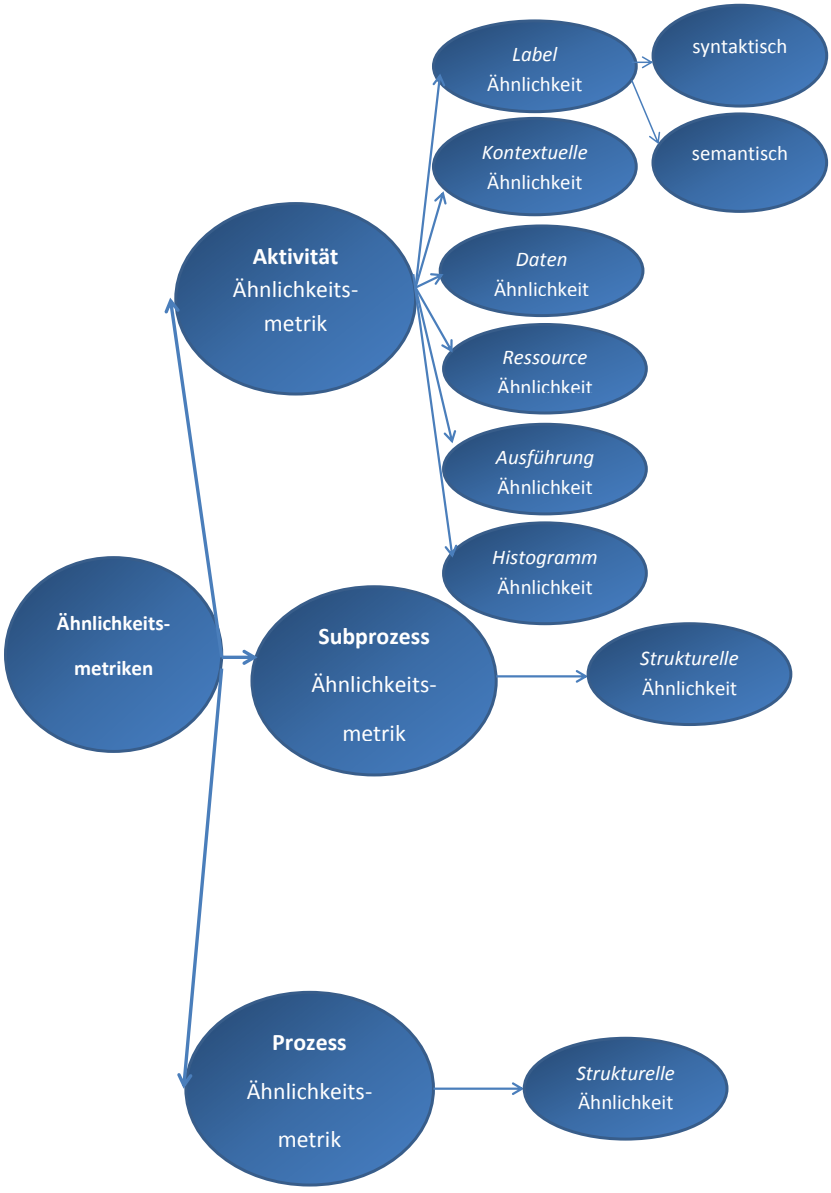

Abbildung 2.1: Ähnlichkeitsmetriken

Aktivitäten sind die kleinsten Einheiten in einem Geschäftsprozess (s. Abschnitt 2.4.1). Wir wählen zwei Aktivitäten innerhalb des gegebenen Prozesses, vergleichen sie nach einer bestimmten Metrik und zeigen am Ende das Resultat auf.

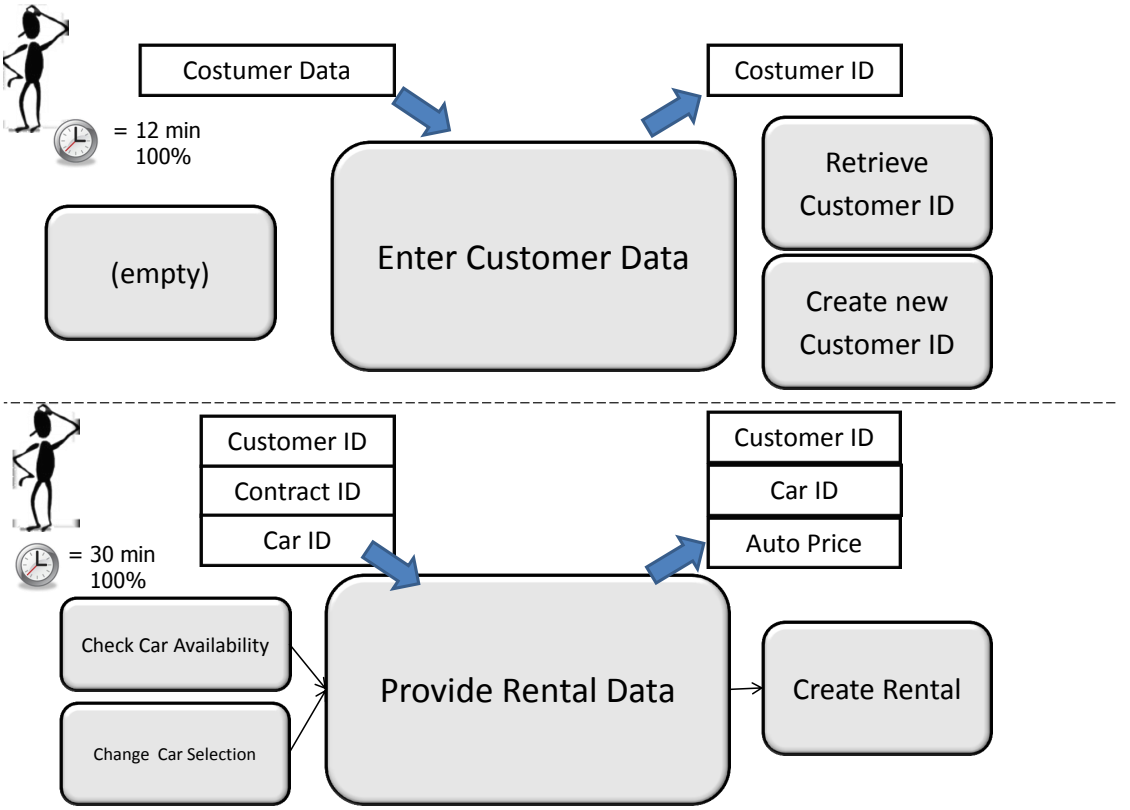

Abbildung 2.2: Beispiel : Aktivitätsmetriken

2.5.1 Syntaktische Ähnlichkeitsmetrik

Definition

Die erste Methode zwei Aktivitäten zu vergleichen heißt syntaktische Ähnlichkeitsmetrik. Das Verfahren beginnt mit einem optimalen Matching zwischen Aktivitäten durch Berechnung ihrer Labels. Ein Label kennzeichnet eine Aktivität. Es ist ein Name, der einer Aktivität gegeben wird, um sie als Teil einer besonderen Gruppe (Prozess) zu kategorisieren.
Wir wählen zwei Aktivitäten von dem Prozess in Abschnitt 2.2 aus [7]. Dieses Paar von Aktivitäten wird weiter bei allen Aktivitätsmetriken verwendet, um die unterschiedlichen Ergebnisse der Metriken zu erkennen. Die Ähnlichkeitsmetrik basiert auf die *Levenshstein Distanz* (*StringEditDistance*) und auf die *Stop Words* Eliminierung.

Signatur und Formel

Es seien a und b die gewählten Aktivitäten. Der Inhalt der Aktivitäten sind Strings, die das Label der Aktivität präsentieren. Weiterhin sei $|a|$ die Länge des ersten Strings und $|b|$ die Länge des zweiten. Die Signatur des Verfahrens wird folgenderweise definiert :

$$sim_{syn}\ (a \times b) \rightarrow [0, 1]$$

String Edit Distance (SED)(s. Abschnitt 2.5.1) ist die minimale Anzahl von Operationen, um a zu b umzuwandeln. Ihr Abstand wird definiert als :

$$SED(|a|, |b|)\ \text{(SED : String Edit-Distance).}$$

Die mathematische Formel für die syntaktische Ähnlichkeitsmetrik zweier Aktivitäten lautet:

$$sim_{syn}\ (a,b) = 1 - \frac{SED(a,b)}{max(|a|,|b|)}$$

Levenshstein Distanz

Levenshtein Distanz ist die minimale Anzahl der Einfügen-, Löschen- und Ersetzen-Operationen, zur Änderung einer bestimmten Zeichenkette und zum Erhalt einer anderen (String Abgleich). Sie präsentiert die minimale Anzahl der editierten Operationen, die erforderlich sind, um eine Folge von Buchstaben in andere zu drehen. Der Anfangsstring heißt Quelle und das Endstring heißt Ziel [3] [6], [**?**], [18], [**?**].

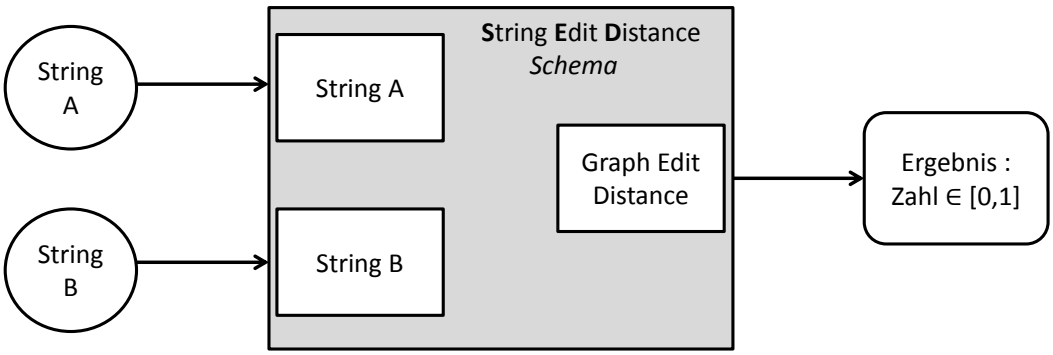

Abbildung 2.3: Schema : String Edit Distance

Beispielsweise haben wir zwei Strings s und t definiert. Wenn s $Test$ ist und t $Test$ ist, dann SED (s, t) = 0, weil keine Transformationen erforderlich sind. Die Strings sind bereits identisch. Wenn aber s $Test$ ist und t $Zelt$ ist, dann SED (s, t) = 2, weil ein Austausch (Änderung T zu Z und S zu L) ausreicht, um s in t umzugestalten. [**?**]

Jede Operation hat einen Aufwand, der durch eine Funktion berechnet wird und ist definiert als: (1- Ähnlichkeit der Knoten). Je mehr Operationen erforderlich sind, desto weniger ähnlich sind die zwei Aktivitäten.

Levenshtein Entfernung wird nach dem russischen Wissenschaftler Wladimir Levenshtein benannt, der sich den Algorithmus 1965 erfand [**?**]. Der Levenshtein Entfernungsalgorithmus wird meistens in folgenden Bereichen verwendet:

- Rechtschreibprüfung [**?**]
- Spracherkennung [12]
- DNA-Analyse [1]

Der Algorithmus funktioniert, wie folgt : Definiert sind zwei Strings S1(1..i) und S2(1..j), wobei i und j die Länge der Strings sind mit Leerzeichen mitgezählt. Es werden von links nach rechts Paare von Buchstaben verglichen. Falls zwei Buchstaben identisch sind, wird zu dem nächsten Paar weitergegangen. Falls das Paar unterschiedlich ist, wird der zweite durch den ersten umgesetzt (Ersetzen-Operation) und letztens, falls kein anderer Buchstabe bei dem zweiten Wort enthalten ist, werden die übrigen Buchstaben eingefügt (Einfügen-Operation). Umgekehrt, falls kein anderer Buchstabe bei dem ersten Wort enthalten ist, wird im zweiten String der Rest der Buchstaben gelöscht (Löschen-Operation).

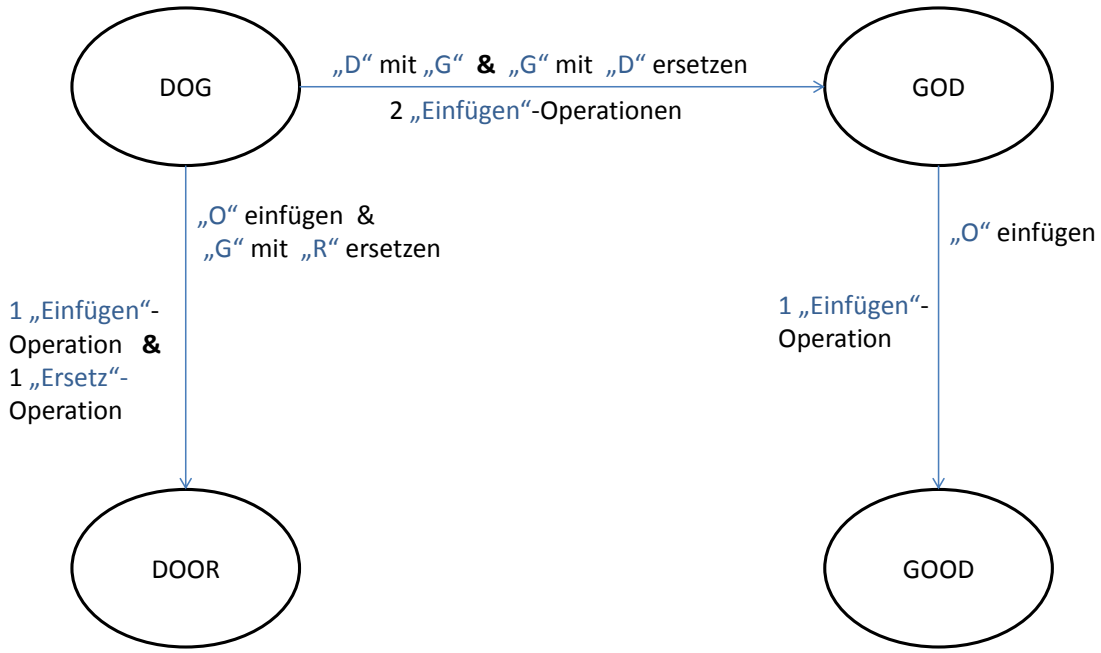

Abbildung 2.4: Beispiel : String-Edit Distance

In der Praxis wird die Levenshstein Distanz zur Bestimmung der Ähnlichkeit von Zeichenketten oder bei der Duplikaterkennung eingesetzt. Die Messung berücksichtigt die Anzahl der vorgenommenen Änderungen einer Zeichenkette zu der anderen und zum Ergebnis normiert sie die Anzahl der Operationen gegen die Länge der längsten der beiden Strings. (s. Formel 2.5.1)

Einsatz von Stoppworte

Der Einsatz von Stoppworte ist ein linguistisches Konzept. Ein Stoppwort (engl. Stop Word) ist ein häufig verwendetes Wort (wie das englische Wort *the*), das in einem Satz durch diesen Mechanismus ignoriert wird. Es ist ein Wort ohne eigentlichen Informationsgehalt und ist sehr häufig im Sprachgebrauch. Sie sind sprachliche/ grammatische Hilfsfunktionen. Stoppworte haben wenig Relevanz in einer Suchabfrage und werden von dieser vor der Entdeckung relevanter Suchergebnisse entfernt [**?**], [**?**].
Bei der Berechnung werden sie entweder vor oder nach Verarbeitung einer natürlichen Sprache gefiltert. Jede Gruppe von Wörtern kann als Stoppworte für einen bestimmten Zweck ausgewählt werden. HTML Strings oder Nummern zählen auch dazu. Für einige Suchmaschinen sind die häufigsten, kurzen Wörter wie z.B : Präpositionen wie *in, auf, von* und Artikel wie *der, die, das*, usw. Andere Suchmaschinen entfernen einige der häufigsten Wörter einschließlich lexikalische Wörter wie *wollen und können*, um die Leistung zu verbessern. Da in unseren Beispielen die Aktivitäten auf English definiert sind, werden wir nach englischen Stoppworte suchen und sie entfernen.

Beispiel

Es seien a = *Enter Customer Data* und b = *Provide Rental Data* die Aktivitäten, die mithilfe von syntaktischer Ähnlichkeitsmetrik verglichen werden. Die minimale Anzahl von Operationen, um das erste Label zu dem zweiten umzuwandeln ist 14. (10 Ersetzen-Operationen und 4 Einfügen-Operationen (s. Abschnitt 2.5.1) .

Die syntaktische Ähnlichkeitsmetrik zwischen *Enter Customer Data* und *Provide Rental Data* lautet :

$$sim_{syn}\ (a, b) = 1 - \tfrac{14}{19} = \mathbf{0.26315}$$

Stoppworte sind hier nicht vorhanden.

2.5.2 Semantische Ähnlichkeitsmetrik

Definition

Semantik heißt der Zweig der Linguistik, der sich mit der Natur, der Struktur, der Entwicklung und den Änderungen der Bedeutungen von Sprachformen oder mit der Kontextbedeutung beschäftigt. Sie ist die Theorie und/oder Wissenschaft der Bedeutung von Syntax. Syntax beschäftigt sich mit der Struktur und Semantik der Bedeutung von *Zeichen*. Zeichen können in diesem Fall Wörter, Phrasen oder Symbole sein.
In vielen Gebieten der künstlichen Intelligenz, wie Verarbeitung der natürlichen Sprache und Informationsgewinnung, sind semantische Konzepte notwendig. Der abstrakte semantische Zusammenhang zwischen Wörtern ist ein grundsätzliches Problem in vielen Anwendungen der linguistischen Datenverarbeitung und künstlichen Intelligenz. Hier

wird ein neues Maß auf die semantische Ontologie-Datenbank *WordNet* vorgeschlagen, der Konzepte mit semantischen Beziehungen verbindet und Konzepte als hochdimensionale Vektoren organisiert [6].
Wenn man in unserem Beispiel durch syntaktisches Ähnlichkeitsverfahren zwei Aktivitäten vergleicht, besteht die Gefahr, dass obwohl die Wörter syntaktisch geringe Ähnlichkeit aufweisen, eventuell bedeutungsmäßig sehr ähnlich zueinander sind. Sie werden also mit geringer Ähnlichkeit bewertet, obwohl sie nah zueinander stehen, weil ihre Bedeutung sehr ähnlich oder sogar fast identisch ist. Ein kleines Beispiel für zwei Strings, die syntaktisch komplett anders sind und semantisch gleich sind die Wörter *automobile* und *car*. Aus diesem Grund definieren wir ein Verfahren, das den Zusammenhang der Bedeutung zweier Aktivitäten berechnet, die sogenannte semantische Ähnlichkeitsmetrik.
Semantische Ähnlichkeitsmetrik spielt eine wichtige Rolle in dem *Information Retrieval*(IR), also zur inhaltlichen Suche nach Texten oder allgemeinen "Dokumenten". Einige der beliebtesten semantischen Ähnlichkeitsansätze sind auf WordNet Datenbank implementiert und evaluiert. Wordnet wird direkt im nächsten Abschnitt präsentiert und mit Details erklärt.

Signatur und Formel

Es seien a und b die gewählten Aktivitäten. Der Inhalt der Aktivitäten sind Strings, die das Label der Aktivität präsentieren. Weiterhin sei $|a|$ die Länge des ersten Strings und $|b|$ die Länge des zweiten Strings. Die Signatur des Verfahrens wird folgenderweise definiert :

$$sim_{sem}\ (a \times b) \rightarrow [0, 1]$$

Wir nehmen an, dass ein genaues Match über ein Match auf Synonymen bevorzugt wird. Entsprechend werden Wörter, die identisch sind, einem Faktor 1 gegeben, während Wörter, die Synonyme sind, dem Faktor 0.75 gegeben werden.

Die mathematische Formel für die semantische Ähnlichkeitsmetrik zweier Aktivitäten lautet:

$$sim_{sem}\ (a, b) = \frac{1*|a \cap b| + 0.75 * \sum_{(syn(a,b))}}{max(|a|, |b|)},$$

wobei |a ∩ b| die identischen Wörter und (syn(a,b)) die Synonyme innerhalb der Labels der zwei Aktivitäten sind.

Vorgehensweise

Damit man zwei Aktivitäten vergleichen kann, soll zuerst eine Reihenfolge von Zwischenschritten gestaltet werden. Diese Schritte erleichtern den Prozess des Verfahrens und garantieren am Ende richtige Ergebnisse. Sie bereiten eine Form von Labels, die schneller, leichter und effizienter nach Ähnlichkeit durchgesucht werden können. Die Schritte werden in den folgenden Abschnitten ausführlich definiert.

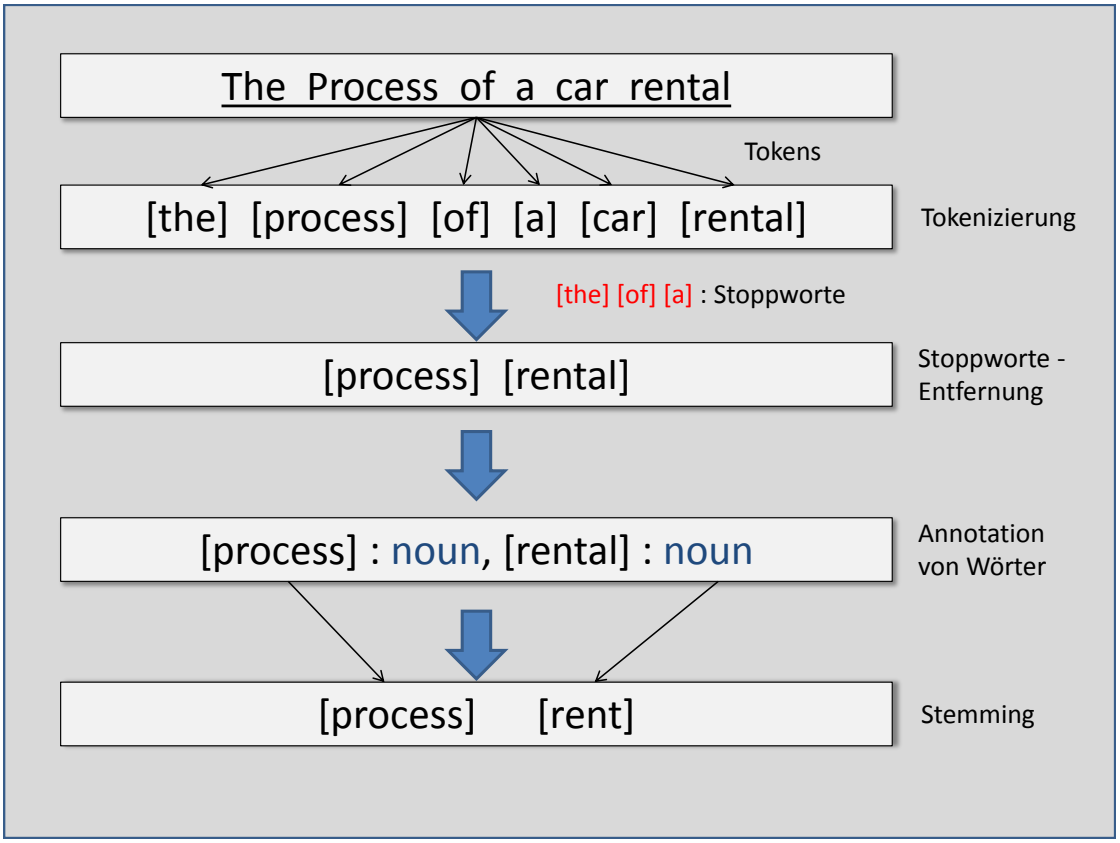

Abbildung 2.5: Grafische Darstellung der Vorarbeitung in der semantischen Ähnlichkeitsmetrik

Methode der Tokenisierung und Einsatz von Stop Words

Bevor man weiter fortschreitet, muss ein Text in Wörter und Sätze segmentiert sein. Dieser Prozess wird Tokenization genannt. Tokenization teilt eine Wörterfolge in Sätze **??**. Nicht nur Wörter werden als Tokens betrachtet, sondern auch Zahlen, Satzzeichen, Parenthesen und Anführungszeichen. In alphabetischen Sprachen werden Wörter gewöhnlich durch Leerzeichen getrennt und enthalten Satzzeichen oder Parenthesen. Eine einfache Tokenizationregel kann deshalb wie folgt festgesetzt werden: *Spalten Sie die Wörterfolge* an Leerzeichenpositionen und abgeschnittenen Satzzeichen und Parenthesen, um die Folge von Tokens zu erhalten. [20]
Bestimmte Zeichen, wie Satzzeichen und SStop Words"(s. Kapitel 2.5.1), werden meistens entfernt. Z.B der Satz "Quotation to be created from inquiry“ wird nach der Tokenisierung [[quotation][created][inquiry]]. ("to","beünd "fromßind StoppWorte und werden entfernt.)

Annotation von Wörtern(Tagging)

Als nächstes folgt das sogenannte Tagging. In der Liguistik ist die grammatische Begriffserklärung der Wörter (engl. Part of Speech oder POS) der Prozess, der die Wörter in Gruppen kategorisiert. [20]
Diese Aufgabe ist um die richtige Wortart (POS (s. Abschnitt 2.5.2) - wie Substantiv, Verb, Pronomen, Adverb ...) im Satz zu identifizieren. Der Algorithmus nimmt einen Satz als Eingabe und legt einen bestimmten Tag-Set fest (eine endliche Liste von POS-Tags). Die Aufgabe ist ein einziges beste POS-Tag für jedes Wort zu finden. Es gibt zwei Arten von Tagger: Die erste misst syntaktische Rollen für jedes Wort (Subjekt, Objekt, ..), und die zweite misst nur funktionale Rollen (Substantiv, Verb, ...). Es gibt eine Menge Arbeit, die auf POS-Tagging vorgenommen worden ist.

Part Of Speech (POS)	Definition	Wörter (Beispiel)
Verb	Drückt die Wortart, die Existenz, Handlung, oder Ereignis in den meisten Sprachen	kaufen, mieten
Nomen	Die Wortart, die verwendet wird, um Person, Platz, Ding, Qualität, oder Handlung zu nennen.	Auto , Kunde
Adjektiv	Die Wortart, die ein Substantiv modifiziert und qualifizierend spezifiziert	schnelles , teure
Adverb	Die Wortart, die in der Regel die Umstände von Tätigkeiten, Geschehnissen, Ereignissen, Eigenschaften oder Verhältnissen genauer beschreibt	schnell , teuer
Pronomen	Die Wortart, die Substantive oder nominale Wortverbindungen auswechselt und Personen oder Dinge benennt	sie, sich
Präposition	Drücken Verhältnisse bzw. Beziehungen zwischen Personen, Gegenständen und/oder Sachverhalten aus	Mit, bei, nach, an

Abbildung 2.6: Part Of Speech (POS)

In unserem Beispiel werden wir nur zwischen Verb und Nomen unterscheiden, weil die Beispielprozesse nur aus solchen Wortarten bestehen.

Stemming Verfahren und Algorithmus

Der dritte Schritt enthält das Stemming Verfahren. Die Reduktion von morphologischen Varianten eines Wortes auf eine Grundform oder einen gemeinsamen Wortstamm heißt Lemmatisierung (engl. Stemming). Stemming ist im *Informational Retrieval* der Bereich, der sich mit der Suche nach Informationen und Metadaten in Dokumenten beschäftigt.
Einige Vorteile des Stemmings in dem *Information Retrieval* sind die besseren Suchergebnisse durch Bündelung unterschiedlicher morphologischer Varianten eines Suchbegriffs, die bessere Performanz (Clustering-Verfahren), die grosse Effizienz durch Einsatz von Stemmern, weniger Redundanz und geringerer Speicherbedarf fürs Vokabular (Reduzierung um bis zu 50 %)
Ein stammender Algorithmus ermöglicht die Verringerung jedes eingegebenen Wortes auf Englisch zu ihrer grundlegenden Wurzel oder Stamm (z.B. *walking* zu *walk*), so dass Variationen über ein Wort als gleichwertig bei der Suche betrachtet werden. Zum Stemming gibt es verschiedene Algorithmen zur automatischen Zurückführung von Wörtern auf ihren Wortstamm. Der bekannteste Algorithmus, der auf die englische Sprache angewendet wird, ist der Porter-Stemmer Algorithmus. [20]

WordNet

WordNet ist eine lexikalische Datenbank der englischen Sprache. Im Allgemeinen teilt sie die englischen Wörter in Gruppen von Synonymen, die sich Synsets nennen. Sie bietet kurze, allgemeine Definitionen, und zeichnet die verschiedenen semantischen Relationen zwischen diesen Synonym-Sets. Zweck ist eine Kombination aus Wörtern zu finden, die mehr intuitiv nutzbar sind und automatische Textanalyse und künstliche Intelligenzanwendungen unterstützen.
Synsets werden mittels begrifflich-semantischer und lexikalischer Beziehungen verkettet. Das resultierende Netz bedeutungsvoller, zusammenhängender Wörter und Konzepte kann mit dem Browser befahren werden. WordNet ist auch frei und öffentlich verfügbar für Download. Die Struktur von WordNet macht es zu einem nützlichen Werkzeug für die linguistische Datenverarbeitung und Verarbeitung der natürlichen Sprache [22], [20], [25]. Die Java API,die für WordNet eine Applikation ist, die Java-Anwendungen mit der Fähigkeit versorgt, Daten von der WordNet-Datenbank wiederzubekommen, heißt Jaws(s. Abbildung 3.1). Es ist eine einfache und schnelle Applikation, die sowohl mit den 2.1 als auch mit 3.0 Versionen der WordNet Datenbankdateien vereinbar ist und mit Java 1.4 verwendet werden kann.

Die Hauptbeziehung unter Wörtern in WordNet ist Synonymie. Synonyme von Wörtern, die dasselbe Konzept anzeigen und in vielen Zusammenhängen austauschbar sind, werden in nicht eingeordnete Sätze (Synsets) gruppiert. Jeder von 117.000 Synsets von WordNet wird zu anderen Synsets mittels einer kleinen Anzahl von Begriffsbeziehungen verbunden. Zusätzlich enthält ein Synset eine kurze Definition und in den meisten Fällen, ein oder kürzere Sätze, die den Gebrauch der Synset-Mitglieder illustrieren. Wortformen

mit mehreren verschiedenen Bedeutungen werden in vielen verschiedenen Synsets vertreten. So ist jede Form eines bedeutenden Paares in WordNet einzigartig.
Die am häufigsten verschlüsselte Beziehung unter Synsets ist die über- und untergeordnete Beziehung (auch Hyperonymie, Hyponymie oder Beziehung von ISA genannt). Es verbindet allgemeinere Synsets wie *Möbel* zu immer spezifischeren wie *Bett*. Alle Substantiv-Hierarchien steigen schließlich zum Wurzelknoten (Entität). Hyponymie-Beziehung ist transitiv: Wenn ein Sessel eine Art Stuhl ist, und wenn ein Stuhl eine Art Möbel ist, dann ist ein Sessel eine Art Möbel.
In WordNet gehört jeder Synset zu einem der vier Part-of-Speech (POS) (s. Abschnitt 2.5.2) Kategorien : Substantiv, Verb, Adjektiv und Adverb). In unserem Beispiel werden wir nur zwischen *Substantiv* und *Verb* unterscheiden.

Synonyme, Hyponyme und Hypernyme

Der Hauptteil von semantischer Ähnlichkeitsmetrik ist das Auffinden von Synonymen, Hyponymen und Hypernymen. Ein Wort kann vielartig im Verhältnis mit einem anderen Wort sein. Wir werden hier vier verschiedene solche Beziehungen betrachten. Zwei Wörter können identisch sein. Das bedeutet, dass sie in allen ihren Eigenschaften ununterscheidbar sind. Sie können auch Synonyme sein. Ein Wort ist Synonym eines anderen Wortes, wenn sie die gleiche oder ähnliche Bedeutung haben (Bedeutungsähnlichkeit, Sinnverwandschaft), z.B *Klassifizierung* und *Verifikation* (s. Abbildung2.7).

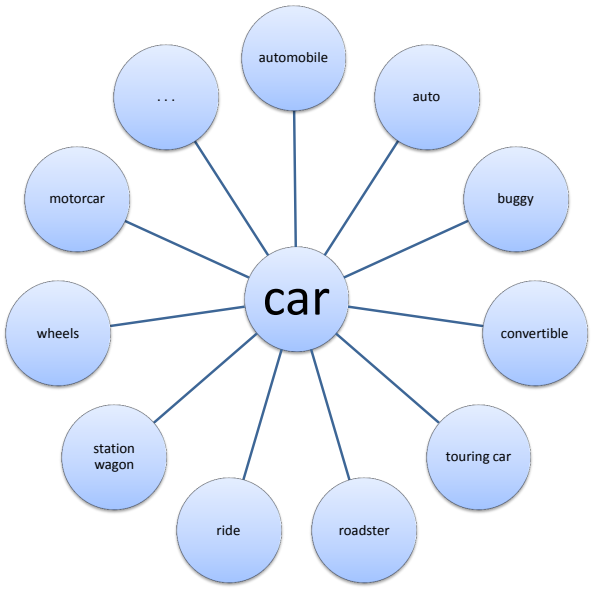

Abbildung 2.7: Beispiel: Synonyme des Wortes „Auto“

Wir erweitern dieses Ähnlichkeitsverfahren auf noch zwei Kategorien von semantischen Verhältnissen, dem Hyponym und dem Hypernym. Hyponyme und Hypernyme zusammen präsentieren die dritte Kategorie.
Ein Hyponym ist ein Wort, das konzeptionell unter die Definition eines anderen Wortes liegt. In der Sprachwissenschaft (Linguistik) verwendet man einen bestimmten Begriff, um ein Mitglied einer Klasse zu bezeichnen. Zum Beispiel ist *Apfel* ein Hyponym von *Obst* und *Hund* ist ein Hyponym von *Tier*. Obwohl Hyponym und Hypernym eng von Bedeutung dem Synonym verwandt sind, sind sie zwei verschiedene semantische Kategorien. *Tulip* ist z.B. ein Hyponym der *Blüte*, aber nicht ein Synonym. Hyponyme sind eine Reihe von verwandten Wörtern, deren Bedeutung konkrete Beispiele für ein allgemeineres Wort sind (z.B. *rot*, *weiß*, *blau*, etc. sind Hyponyme der *Farbe*). Hyponymie ist also die Beziehung zwischen einem allgemeinen Begriff wie *Polygon* und spezifische Instanzen davon, wie *Dreieck*. Hyponymie ist eine semantische Beziehung. Das Gegenteil eines Hyponyms ist ein Hypernym. [**?**]
Ein Hypernym ist ein umgangssprachlicher Begriff für ein übergeordnetes Wort, dessen Bedeutung auch die Bedeutungen der anderen Wörter umfasst. In der Sprachwissenschaft ist ein Hyponym ein Wort oder eine Phrase, deren semantischer Bereich innerhalb eines anderen Wortes enthalten ist. Hypernym ist die semantische Beziehung, in dem ein Wort der Überbegriff für ein anderes ist. So ist *Obst* ein Hypernym von $Apfel$, während *Apfel* ein Hyponym vom *Obst* ist [**?**].
Die Beziehung zwischen einem Wort und seinem Hyponym oder Hypernym ist eine IS-Beziehung. Zum Beispiel wird *Farbe Rot* verwendet, um die Beziehung zwischen Hyponym *rot* und *Farbe* zu beschreiben. Beispielsweise bezeichnet $Fahrzeug$ alle Dinge, die getrennt von den Worten Zug, $Wagen$, $Flugzeug$, *Auto* bezeichnet werden und ist somit ein Überbegriff für jedes dieser Wörter. Umgekehrt sind die Wörter : Zug, $Wagen$ etc. Hyponyme des $Fahrzeugs$. *Es ist ein Verhältnis von Hyponymie wenn ein Wort stets durch ein zweites Wort ersetzt werden kann, aber nicht umgekehrt, ohne Veränderung der Bedeutung* (s. Abbildung 2.8).

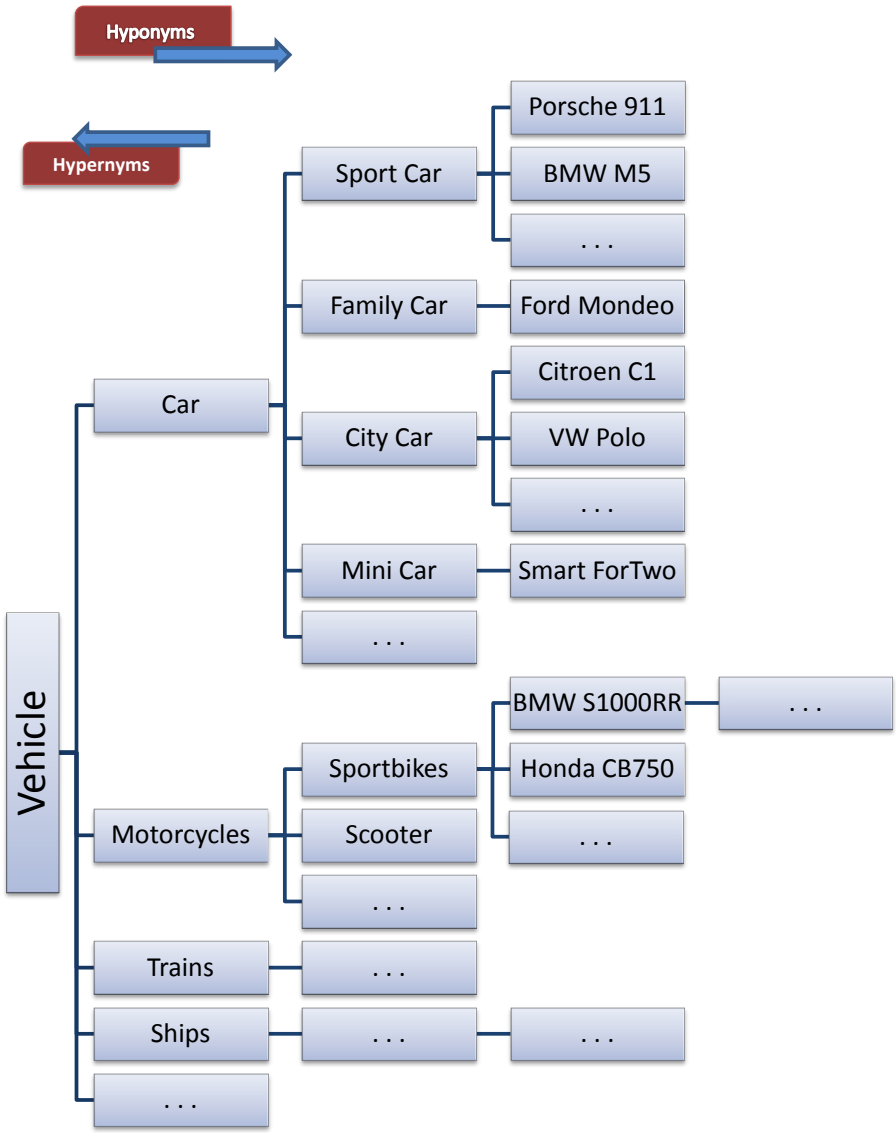

Abbildung 2.8: Beispiel : Hyponyme-Hypernyme des Wortes Auto

Algorithmus (Erklärung)

Nach allen vorherigen Schritten, beginnt dann der Algorithmus. Wir haben zwei Listen von Wörtern. Diese Listen sind in einer Grundform reduziert (s. Abschnitt 2.5.2) und für jedes Wort innerhalb der zwei Listen wurde eine funktionale Rolle festgelegt (s. 2.5.2). Als nächstes finden wir die Synonyme, Hyponyme und Hypernyme der Wörter der ersten Liste. Wir vergleichen die Ergebnisse mit den Wörtern der zweiten Liste. Falls ein Synonym, Hypernym oder Hyponym identisch ist mit einem Wort der zweiten Liste, werden sie als *similar* betrachtet.

Beispiel

Es seien $a =$ *Enter Customer Data* und $b =$*Provide Rental Data* die Labels der Aktivitäten, die mithilfe von semantischer Ähnlichkeitsmetrik verglichen werden. Zuerst werden die zwei Labels in Wörter zerlegt (s. Abschnitt 2.5.2) : ([Enter][Customer][Data]) ist die erste Menge und ([Provide][Rental][Data]) die zweite. Stoppworte sind nicht vorhanden. Danach werden die richtigen Wortarten identifiziert. *Enter* und *Provide* sind Verben und *Customer*, *Rental* und *Data* sind Nomen. Nach dem Stemming Verfahren (s. Kap 2.5.2), werden die Wörter unveränderbar bleiben. Nachher beginnt der Hauptteil des Algorithmus. Man prüft zuerst, ob Wörter in beiden Mengen identisch sind. In unserem Beispiel ist es das Wort *Data*. Danach definiert man Synonyme für die Menge der Wörter in dem ersten Label und prüft, ob eins davon in der zweiten Menge beinhaltet ist. Das Wort *Enter* ist Synonym des Wortes *Provide*. Wir definieren einen Parameter α. Man kann die Notwendigkeit eines Synonyms (bzw. Hyponyms oder Hypernyms) in dem Ähnlichkeitsverfahren beurteilen. In unserem Beispiel ist ($\alpha = 0.75$). Die semantische Ähnlichkeitsmetrik zwischen *Enter Customer Data* und *Provide Rental Data* ist : $SIM_{sem}\ (a, b) = \frac{1*1+0.75(0+1)}{3} = \mathbf{0{,}91667}$

2.5.3 Label Ähnlichkeitsmetrik

Die zwei vordefinierten Verfahren gehören zu einem größeren Verfahren, dem sogenannten Label Ähnlichkeitsverfahren, das die Labels der Aktivitäten vergleicht. Man erhält das Ergebnis, nachdem man eine optimale Ähnlichkeit zwischen den Knoten, die verglichen werden, gefunden hat. Das Label, das die Ähnlichkeit vergleicht, ist die Summe der Labels-Ähnlichkeit von verglichenen Paaren von Knoten. Man erhält hier ein Ergebnis zwischen 0 und 1. Wir fassen die zwei Metriken zusammen (s. Abb. 2.9) [5].

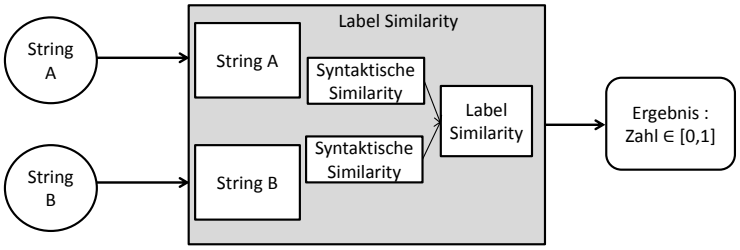

Abbildung 2.9: Schema : Label Similarity

Signatur und Formel

Es seien a und b die gewählten Aktivitäten. Der Inhalt der Aktivitäten sind Strings, die das Label der Aktivität präsentieren. Weiterhin sei $|a|$ die Länge des ersten Strings und $|b|$ die Länge des zweiten Strings. Die Signatur des Verfahrens wird folgenderweise definiert :

$$sim_{label}\ (a \times b) \rightarrow [0, 1]$$

Die mathematische Formel für die semantische Ähnlichkeitsmetrik zweier Aktivitäten lautet

$$sim_{label}\ (a, b) = \alpha * sim_{syn}(a, b) + \beta * SIM_{sem}(a, b),$$

wobei α und β zwei vom Benutzer definierte Konstanten sind und α+β=1.
(= Man kann in den Formeln Gewichtungen einfügen. Nach Wunsch und Bedürfnis kann der Anwender unterschiedliche Faktoren dazufügen, um ein Ähnlichkeitsverfahren verstärkter zu bewerten als ein anderes. Die Begründung dafür ist, dass bei der syntaktischen Ähnlichkeit 2.5.1 die Gefahr besteht, falsche Ergebnisse zu erhalten.)

Beispiel

Es seien a = *Enter Customer Data* und b =*Provide Rental Data* die Aktivitäten, die mithilfe von Label Ähnlichkeitsmetrik verglichen werden. Wir kombinieren die zwei davor definierten Ähnlichkeitsverfahren zu einem. Die Parameter α und β haben die Werte : α

= 0.3 und $\beta = 0.7$.
Die Label Ähnlichkeitsmetrik zwischen *Enter Customer Data* und *Provide Rental Data* ist : $sim_{label}\ (a, b) = 0.3 * 0.26315 + 0.7 * 0,91\overline{6} =$ **0.715945**

2.5.4 Kontextuelle Ähnlichkeitsmetrik

Ein weiteres Verfahren zwei Aktivitäten zu vergleichen, ist die kontextuelle Ähnlichkeitsmetrik. Als Kontext bezeichnet man eine geschriebene Behauptung oder Aussage, die vor und/oder nach einem bestimmten Wort stattfindet und in der Regel seine Bedeutung oder Wirkung beeinflusst. Die Grundidee in dem Verfahren sagt, dass wenn in einem Prozess die Vorgänger- und die Nachfolgeraktivitäten ähnlich sind, dann sind es auch die gewählten zwei Aktivitäten. Die Aktivitäten werden mithilfe der Label Ähnlichkeitsmetrik zum Vergleich herangezogen [5], [9].

Definition und Formel

Es seien a und b die gewählten Aktivitäten. Der Inhalt der Aktivitäten sind Strings, die das Label der Aktivität präsentieren.

Weiterhin seien va_i (mit i= 0,...,n) und vb_j (mit j = 0,..,m) die Mengen von Vorgängeraktivitäten der zwei Aktivitäten (n, m $\in \mathbb{N}$). Man definiert mittels kartesischen Produkts Paare zwischen Vorgänger- und Nachfolgeraktivitäten von a und b.

$(va_i \times vb_j)$: Kartesisches Produkt für Vorgängeraktivitäten
$(na_i \times nb_j)$: Kartesisches Produkt für Nachfolgeraktivitäten,
mit (i= 0,...,n), (j = 0,..,m) (und n, m $\in \mathbb{N}$

Diese Paare werden nach Label Ähnlichkeitsverfahren verglichen (s. Abschnitt 2.5.3. Das Paar mit der höchsten Ähnlichkeit wird als Ergebnis der Ähnlichkeitsmessung gewählt. Das Ähnlichkeitsverfahren für Nachfolgeraktivitäten wird gleichartig gemessen.

Die maximale kontextuelle Ähnlichkeit für die Vorgänger wird definiert :

$$sim_{contexv}(a,b) := \max\ (sim_{vlabel}(va_i \times vb_j)$$

Die maximale kontextuelle Ähnlichkeit für die Nachfolger wird definiert :

$$sim_{contexn}(a,b) := \max\ (sim_{nlabel}(na_i \times nb_j)$$

Die Signatur des Verfahrens wird folgenderweise definiert :

$$sim_{contx}\ (a \times b) \rightarrow [0, 1]$$

Die mathematische Formel für die gesamte kontextuelle Ähnlichkeitsmetrik zweier Aktivitäten lautet:

$$sim_{contx}\ (a,b) = \frac{sim_{contexv}(a,b)+sim_{contexn}(a,b)}{2}$$

Vorgehensweise

Zwei Aktivitäten sind ähnlich, wenn sowohl die Vorgänger- als auch die Nachfolgeraktivitäten ähnlich sind. Jedes zu vergleichende Paar von Aktivitäten wird mithilfe des syntaktischen und des semantischen Ähnlichkeitsverfahrens nach Ähnlichkeit durchsucht (s. Abschnitte 2.5.3, 2.5.1 und 2.5.2).

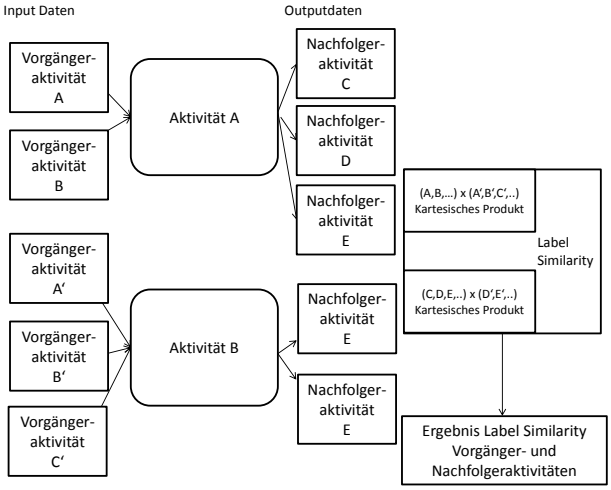

Abbildung 2.10: Schema : Kontextuelle Similarity

Beispiel

Es seien a = *Enter Customer Data* und b =*Provide Rental Data* die Aktivitäten, die mithilfe von kontextueller Ähnlichkeitsmetrik verglichen werden. Weiterhin seien *Change Car Selection* und *Check Car Availability* die Vorgänger der zweiten Aktivität. Die erste Aktivität hat keine Vorgängeraktivitäten. Die Aktivitäten *Retrieve Customer ID* und *Create new Customer ID* sind die Vorgängeraktivitäten von a, und *Add extras* die von b. Wir vergleichen die Vorgängeraktivitäten der zwei Aktivitäten mithilfe des Verfahrens von Label Ähnlichkeitsmetrik (s. Abschnitt 2.5.3). Da a keine Vorgängeraktivitäten hat, ist die Ähnlichkeit gleich **0**. Durch Anwendung der Label Ähnlichkeitsmetrik für die Outputaktivitäten erhält man als Ergebnis **0**, da sie überhaupt nicht ähnlich sind (weder syntaktisch noch semantisch). Aus diesem Grund ist : sim_{contx} $(a,b) = \mathbf{0}$

2.5.5 Daten Ähnlichkeitsmetrik (Input/ Output Mappings)

Ein Geschäftsprozess (engl. Business Prozess) ist eine Sammlung von Aktivitäten. Er unterstreicht, wie die Arbeit innerhalb einer Organisation getan ist. Ein Prozess ist also eine bestimmte Reihenfolge der Tätigkeiten über Zeit und Ort, mit einem Anfang und einem Ende.
Jede Aktivität benötigt Eingabe- um Ausgabedaten zu produzieren (s. Kap 2.4.5). Die produzierten Outputs sind die benötigten Inputs der nächsten Aktivitäten. Sie enthalten

wichtige Informationen und sind nötig für den Kontrollfluss, also die Aussagen und anderen Konstrukte, die die Reihenfolge in der die Operationen ausgeführt werden definieren (s. Abschnitt 2.1.2) oder Datenfluss des Prozesses, also Daten, die zwischen Aktivitäten getauscht werden (s. Abschnitt 2.1.2). Output ist das Ergebnis nach der Ausführung einer Aktivität, das für die nächste Aktivität auch von Bedeutung ist.(s. 2.4.1)

Definition und Formel

Es seien a und b die gewählten Aktivitäten. Der Inhalt der Aktivitäten sind Strings, die das Label der Aktivität präsentieren.

Weiterhin sei $varIn_a$ (mit i= 0,...,m) und $varI_b$ (mit j = 0,..,n) die Mengen der Eingabedaten (Variablen) der zwei Aktivitäten :
Weiterhin seien $varOut_a$ (mit i= 0,...,k)und $varOut_b$ (mit j = 0,..,l) die Mengen von Ausgabedaten der zwei Aktivitäten (k, l $\in \mathbb{N}$). Man definiert mittels kartesischen Produkts Paare zwischen Eingabedaten und Ausgabedaten von a und b.

$$(varIn_i \times varIn_j) : \text{Kartesisches Produkt für Eingabedaten}$$
$$(varOut_i \times varOut_j) : \text{Kartesisches Produkt für Ausgabedaten,}$$
$$\text{mit (i= 0,...,n), (j = 0,..,m) und n, m} \in \mathbb{N}$$

Diese Paare werden nach Label Ähnlichkeitsverfahren verglichen. Man nimmt das Ergebnis für das Paar mit der höchsten Ähnlichkeit. Das Ähnlichkeitsverfahren für Ausgabedaten wird gleichartig gemessen.
Die maximale Datenähnlichkeit für die Eingabedaten der zwei Aktivitäten lautet :

$$sim_{data_{in}}(a,b) := \max\ (sim_{vlabel}(varIn_i \times varIn_j)$$

Die maximale Ähnlichkeit für die Ausgabedaten der zwei Aktivitäten lautet

$$sim_{data_{out}}(a,b) := \max\ (sim_{nlabel}(outna_i \times outnb_j)$$

und sei na_i (mit i= 0,...,n)und nb_j (mit j = 0,..,m) , wobei n, m $\in \mathbb{N}$

Die Signatur des Verfahrens wird folgenderweise definiert :

$$sim_{data}\ (a \times b) \rightarrow [0,1]$$

Die mathematische Formel für die Datenähnlichkeit zweier Aktivitäten lautet:

$$sim_{data}\ (a,b) = \frac{sim_{data_{in}}(a,b)+sim_{data_{out}}(a,b)}{2}$$

Vorgehensweise

Eine Liste mit den Eingabedaten der zwei zu vergleichenen Aktivitäten wird aufgestellt. Wir suchen Synonyme des ersten Inputs und vergleichen sie mit dem Input der zweiten Aktivität per Permutation (d.h alle möglichen Kombinationen von Paaren der zwei Aktivitäten). Dieselbe Vorgehensweise gilt auch für die Ausgabedaten.

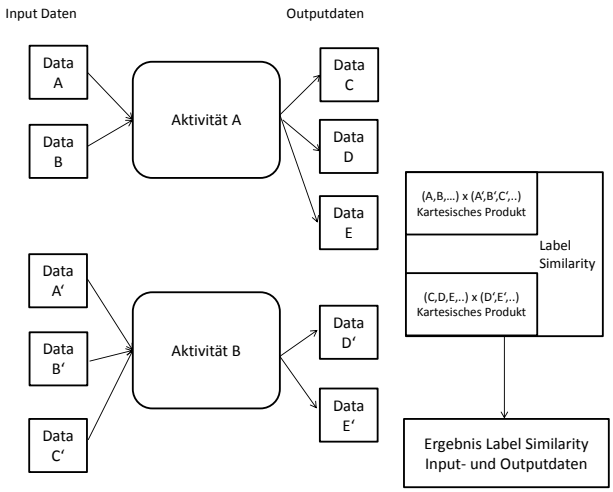

Abbildung 2.11: Schema : Data Similarity

Es seien $a =$ *Enter Customer Data* und $b =$*Provide Rental Data* die Aktivitäten, die mithilfe von Daten Ähnlichkeit verglichen werden. Weiterhin seien *Customer Data* und *Driver License* die Eingabedaten der ersten Aktivität und *Customer ID* und *Car ID* die Eingabedaten der zweiten Aktivität. Nach Anwendung der Label Ähnlichkeitsmetrik (s. Abschnitt 2.5.3) ist das Paar *Customer Data* und Customer ID das Paar der Eingabedaten mit dem größten Ähnlichkeitsgrad. Es besteht eine syntaktische Ähnlichkeitsmetrik von $1 - \frac{4}{13} = \mathbf{0.6923}$ und eine semantische Ähnlichkeitsmetrik von $\frac{(1*1)+(0.75*0)}{2} = \mathbf{0.5}$ (s. Abschnitte 2.5.1 und 2.5.2). Insgesamt ist die Label Ähnlichkeitsmetrik gleich **0.5961**. Es seien noch *Customer ID* die Ausgabedata der ersten Aktivität und *Customer ID*, *Contract ID* und *Car Price* die Ausgabedaten der zweiten Aktivität. Das Paar von Eingabedaten, das am meisten Ähnlichkeiten hat, ist das Paar *Customer ID* und *Customer ID*, die identisch sind und deswegen eine Ähnlichkeit von **1** haben.
Die Datenähnlichkeit der zwei Aktivitäten berechnet man folgenderweise : $\frac{0.5961+1}{2} =$ **0.798**

2.5.6 Ressource Ähnlichkeitsmetrik

Die benötigten Ressourcen führen zur Realisierung eines Geschäftsprozesses. Solche Dokumente dienen dazu, die Vorgänge und Abläufe im Unternehmen anzulegen, Verantwortlichkeiten abzugrenzen und die am Geschäftsprozess beteiligten Personen über ihre

Aufgaben zu informieren (s. Abschnitt 2.4.4). Zwei Aktivitäten sind dann ähnlich, wenn ihre Ressourcen ähnlich sind.

Bedeutung und Arten von Ressourcen

Ressourcen sind das Mittel zur Erbringung einer definierten Leistung oder Erfüllung einer definierten Aufgabe. Sie legen formale, fachliche, räumliche, technische, finanzielle und zeitliche Ausstattungen als Rahmenbedingungen fest.
Die Ressourcen sind eine Anzahl von Mitteln zur Optimierung der Arbeit im Prozess. Sie sind in unteren Kategorien unterteilt : Programme (Anwendungen), die entweder automatisch oder manuell Informationen verarbeiten, Informationen in Form von Daten, Hardware, Betriebssysteme, Datenbank-Management Systeme und Netzwerke, die den Prozess unterstützen, und die Menschen, die für alle Bereiche wichtig sind, z.B Planung, Organisation, Implementierung, Beschaffung, Evaluierung, usw.

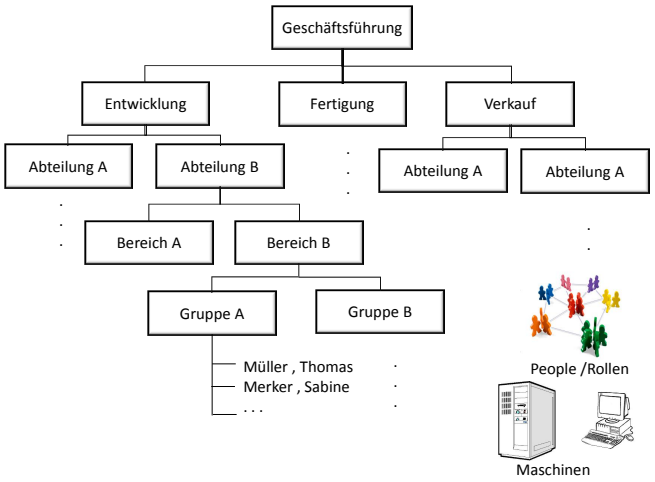

Abbildung 2.12: Ressourcenbaum

Definition und Formel

Es seien a und b die gewählten Aktivitäten. Der Inhalt der Aktivitäten sind Strings, die das Label der Aktivitäten präsentieren.
Die Signatur des Verfahrens wird folgenderweise definiert:

$$sim_{res}\ (a \times b) \rightarrow [0, 1]$$

Bei Ressource Metrik werden wir drei Ähnlichkeitsverfahren benutzen, da wir drei Bereiche nach Ähnlichkeit messen wollen. Zuerst werden wir mithilfe der Label Ähnlichkeitsmetrik die Labels der zwei Aktivitäten messen. Die Ressourcen in zwei Prozessen haben Typen. Die Formel ist:

$$sim_{resTyp}(a,b) = \begin{cases} 1, & \text{falls } sim_{resTyp}(a) = sim_{resTyp}(b) \\ 0, & \text{sonst} \end{cases}$$

Die Ressourcen, also die Menschen und die Maschinen, die wichtig sind zur Bearbeitung einer Aktivität, haben verschiedene Rollen. Die Rollen sind Teil der Ähnlichkeitsmetrik [19]. Die Rollen in einem Unternehmen sind in Form von Organigramme (s. Abbildung 2.12). Die Ähnlichkeit der Rollen berechnet man folgenderweise:

$$sim_{resRol}(a,b) = \frac{dist(R_1,R_2)}{NF},$$

wobei NF ein definierter NF (Normalfaktor) ist.

Der Gebrauch von Ressourcen (engl. Utilization) zeigt das Verhältnis der Verfügbarkeitszeit (ausgedrückt gewöhnlich als ein Prozentsatz) eines Systems. Die Ähnlichkeit der Rollen berechnet man folgenderweise:
Es sei

- h_{op} die operative Zeit des Gebrauchs einer Ressource
- h_{av} : die verfügbare Zeit des Gebrauchs einer Ressource

Es sei weiterhin

$util_{R1} = \frac{h_{op}}{h_{av}}$ der Gebrauch einer Ressource. $util_{R1} = \frac{h_{op}}{h_{av}} * 100$ der prozentuale Gebrauch einer Ressource.

Die Formel zum Gebrauch der Ähnlichkeitsmetrik zwischen zwei Ressourcen ergibt sich folgendermaßen:

$$sim_{res_{util}}(a,b) = 1 - |util_{R1} - util_{R2}|$$

Beispiel

Es seien $a =$ *Enter Customer Data* und $b =$*Provide Rental Data* die Aktivitäten, die mithilfe von Label Ähnlichkeitsmetrik verglichen werden (s. Abschnitt 2.5.3. Es sei weiterhin $h_{op} = 8$ Stunden die operative Zeit des Gebrauchs der ersten Ressource im Prozess und $h_{av} = 8$ Stunden die verfügbare Zeit des Gebrauchs der ersten Ressource. Für eine zweite Ressource sei $h_{op} = 8$ Stunden ihre operative Zeit und $h_{av} = 24$ Stunden ihre verfügbare Zeit. Es sei dann : $sim_{res_util}(a,b) = 1 - |\frac{24}{24} + \frac{8}{24}| = 1 - |1 - 0.\overline{3}| = \mathbf{0.333}$

Berechnung der Label Ähnlichkeitsmetrik der Ressourcen

Wir vergleichen die Ressourcen auf mehreren verschiedenen Ebenen. Zuerst beurteilen wir die Ressourcen vergleichsweise nach Label Ähnlichkeitsmetrik, d.h nach syntaktischer und semantischer Ähnlichkeitsmetrik.

Berechnung der Ressourcen bzgl. der Rolle im Prozess

Damit die Aktivitäten in einem Business Prozess flexibel, effizient, effektiv, fehlerfrei und in optimaler Zeit durchgeführt werden, sollte man die Rollen definieren.

Gebrauch der Ressourcen im Prozess

Die Verbesserung des Managements von Ressourcen führt zur Erhöhung der Wirtschaftlichkeit durch die Optimierung der Auslastung und durch die Optimierung der Zeit. Es erreicht auch die Loyalität der Mitarbeiter, die Wettbewerbsvorteile als Folgerung hat. Durch die effektive Nutzung der Ressourcen kann man die Kosten für die Bereitstellung drastisch reduzieren und gleichzeitig den Kundendienst erhöhen.

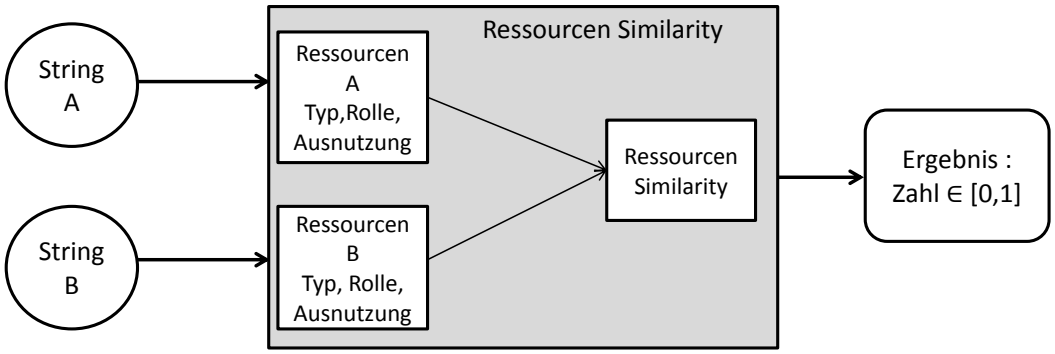

Abbildung 2.13: Schema : Ressourcen Similarity

2.5.7 Execution Ähnlichkeitsmetrik

Nach der Modellierungsphase in einem Business Prozess erfolgt die Ausführungsphase. In dieser Phase werden die Aktivitäten zur Abarbeitung des Prozesses ausgeführt. Falls zwei vergleichende Aktivitäten (ungefähr) dieselbe Ausführungsdauer und/oder Frequenz haben, kann dies ein Hinweis auf Ähnlichkeit sein.

Definition und Formel

Es seien a und b die gewählten Aktivitäten. Der Inhalt der Aktivitäten sind Strings, die das Label der Aktivitäten präsentieren.
Die Signatur des Verfahrens wird folgenderweise definiert :

$$sim_{exe}\ (a \times b) \rightarrow [0,1]$$

Bei Execution Metrik werden wir zwei Ähnlichkeitsmetrikverfahren benutzen, da wir zwei Kategorien nach Ähnlichkeit messen wollen. Die erste Kategorie ist die Dauer, die im Process Metamodell (s. Abschnitt 2.1) schon definiert ist. Die Formel lautet :

$$dur(a) = end_a - begin_a$$
$$dur(b) = end_b - begin_b$$

Signatur der Dauer ist :

$$dur : G \cup N_A * COp \rightarrow \mathbb{R}_{\geq 0}$$

Die Formel lautet :

$$dur(a) = \frac{F_{op}(a_i)}{F_{Op}(a_i)}$$

Signatur von Frequenz ist :

$$freq : N \cup E_C \cup E_D \cup E_R \cup G * FO_p \rightarrow \mathbb{R},$$

wobei
N : die Menge der Aktivitäten
E_C : die Menge der Kontrollkonnektoren 2.1.2
E_D : die Menge der Datenkonnektoren 2.1.2
E_R : die Menge der Ressourcenkonnektoren
G : Graph
FO_p : Frequenzoperator (absolut, shared)

Vorgehensweise

Dauer der Ausführung eines Prozesses

Dauer der Ausführung der Aktivität ist die Zeit, die man braucht, bis eine Aktivität mit den benötigten Eingaben und den dazugehörigen Ressourcen genau das, was sie berechnen soll, erfolgreich ausführt und sich ein korrektes Ergebnis ergibt.

Frequenz der Ausführung der Aktivitäten eines Prozesses

Frequenz der Ausführung der Aktivitäten zeigt, wie oft eine Aktivität durchgeführt werden soll, bis ein Business Prozess erfolgreich endet.

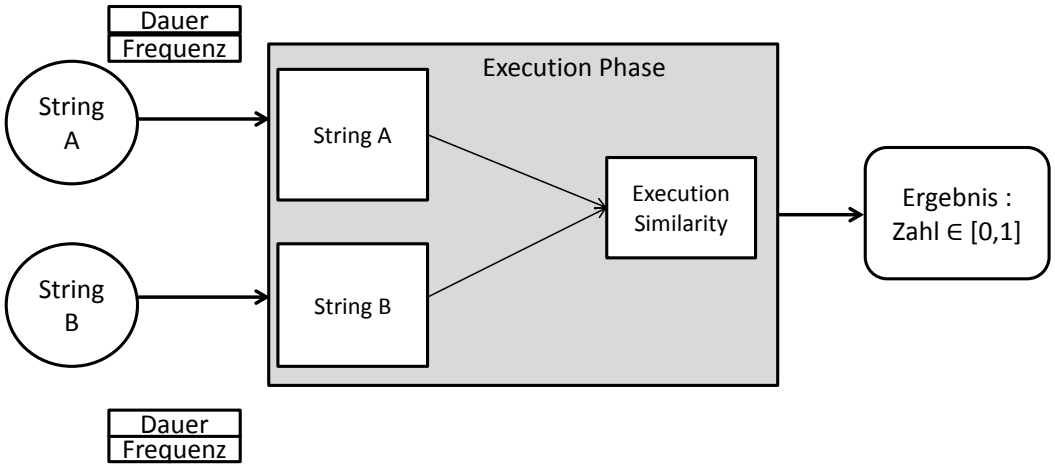

Abbildung 2.14: Schema : Execution Similarity

Beispiel

Es seien $a =$ *Enter Customer Data* und $b =$*Provide Rental Data* die Aktivitäten, die wir mithilfe von Label Ähnlichkeitsmetrik verglichen werden. Sei weiterhin $dur_1 = 12$ Min die Dauer der ersten Aktivität und $dur_2 = 30$ Min.

2.5.8 Histogramm Matching

Histogramm

Histogramm ist ein Mittel zur grafischen Darstellung einer Verteilungsfunktion von Werten eines Merkmals. Eine große Reihenfolge von Zahlen ist schwer zu durchschauen, sodass mithilfe eines Histogramms diese Zahlenfolge übersichtlicher wird. Die gemessenen Häufigkeiten an numerischen Einheiten werden in Form von Rechtecken dargestellt.

Die Fläche der Rechtecke ist proportional zu den Häufigkeiten. [2]

In einem Histogramm kann man die Häufigkeitsverteilung der Werte einer intervallskalierten Variable darstellen. Ein Histogramm ist eine grafische Darstellung der Klassenhäufigkeiten von Daten. Die Werte werden in Gruppen zusammengefaßt. Durch das Einfügen einer Normalverteilungskurve in ein Histogramm, kann man die empirische Häufigkeitverteilung mit der Normalverteilung vergleichen und damit grafisch überprüfen, ob die Annahme einer Normalverteilung der Werte plausibel erscheint. D.h., mithilfe eines Histogramms kann man sich einen visuellen Eindruck über die Häufigkeitsverteilung großer Datenmengen machen. [?] [?]
Ein Histogramm hat meistens folgende diskrete Form :

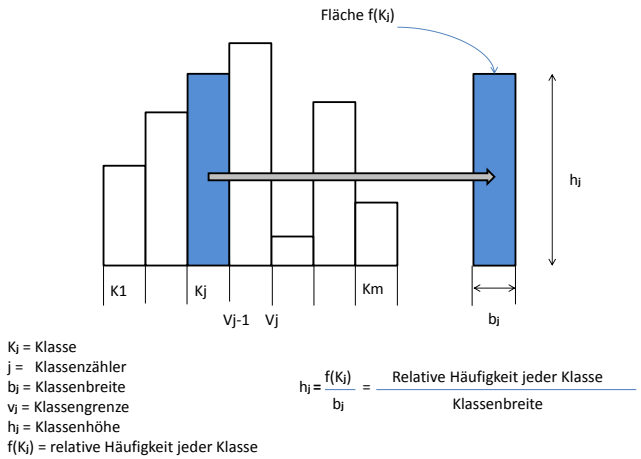

Abbildung 2.15: Histogramm(1)

- Klasse K : In bestimmten Fällen bietet es sich an, benachbarte Messwerte einer Verteilung zu Klassen zusammenzufassen, um die Übersichtlichkeit und graphische Darstellung zu erleichtern. Damit versteht man unter einer Klasse K die Menge sämtlicher Messwerte, die innerhalb festgelegter Grenzen liegt.

- Klassenzähler j : berechnet die Anzahl der Spalten in einem Histogramm.

- Klassenbreite b_j : berechnet sich aus der Differenz zweier aufeinanderfolgender Klassenmitten. Die Klassenbreite oder auch Klassenweite drückt aus, wie viele der untersuchten Messwerte einer Verteilung in einer Klasse zusammengefasst sind. Je nach unterschiedlicher Klassenbreite kann sich die Darstellung der Häufigkeitsverteilung stark verändern. In den meisten Fällen ist die Breite aller Klassen in einem Histogramm gleich. Unterscheidet sich die Breite, verändert sich die Berechnung.

Die Häufigkeiten sollen korrigiert werden, sodass am Ende die Fläche wiederum 1 ergibt.

- Klassengrenze v_j : entspricht dem größten bzw. kleinsten Messwert einer Klasse.

- relative Häufigkeit $f(K_i)$: Anteil der Elemente einer Gesamtheit, die zu einer bestimmten Klasse gehören. Die Summe der relativen Häufigkeiten ist 1.

Oft wird ein Histogramm in einer kontinuierlichen Darstellung präsentiert :

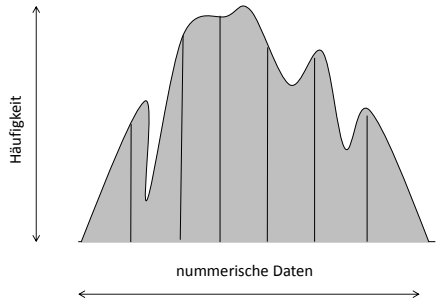

Abbildung 2.16: Histogramm(2)

Einführung in das Histogramm Matching

Zu vergleichen sind zwei Aktivitäten zweier Prozesse. Sie bekommen mehrere verschiedene Eingabedaten um ausgeführt zu werden (s. Abschnitt 2.4.5). Wir vergleichen Eingaben der zwei Aktivitäten mithilfe der Methode von Data Ähnlichkeitsmetrik (s. Abschnitt 2.5.5). Es seien z.B. zwei zu vergleichende Aktivitäten A und B. Aktivität A bekommt die Eingabedaten x und y und die Aktivität B die Eingabedaten c und d. Nach Anwendung des Datenähnlichkeitsverfahrens folgt daraus, dass Data x ähnlich mit Data c und y mit d ist. Wir konstruieren ein Histogramm für die Aktivität A mit der Häufigkeit, mit der die Eingabedaten x und y vorkommen. Dasselbe machen wir mit Aktivität B.

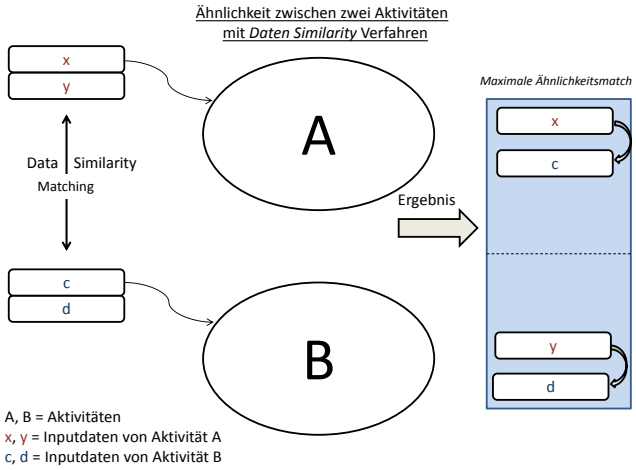

Abbildung 2.17: Beispielprozess

Eine Aktivität in einem Prozess kann mehrmals vorkommen, also mehrmals Eingabedaten bekommen. Wir benutzen das Verfahren der Histogramm Ähnlichkeitsmetrik genau dann, wenn die Daten natürlichen oder reelen Zahlen entsprechen. Anstatt die Werte einzeln zu betrachten, gruppieren wir die Werte der Zahlen in kleinen Mengen mit derselben Klassenbreite, die vorerwähnten Klassen (s. Abschnitt 2.15). Wir konstruieren zwei Histogramme jeweils für jedes Input Data des Paares.

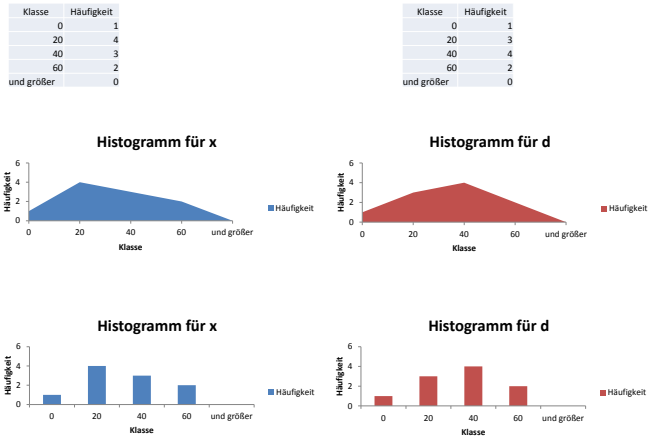

Abbildung 2.18: Erstellung von Histogrammen

Vorgehensweise

Es seien A und B die gewählten Aktivitäten. Der Inhalt der Aktivitäten sind Strings, die das Label der Aktivitäten präsentieren. Sie bekommen jeweils Eingabedaten, jedes Mal, wenn sie betroffen sind.
Die Signatur des Histogramm Ähnlichkeitsverfahrens wird folgenderweise definiert :

$$SIM_{histogr}\ (A \times B) \rightarrow [0,1]$$

Nach Anwendung von Data Ähnlichkeitsmetrik werden Paare von ähnlichen Daten gestellt. Wir benutzen das erste ähnliche Paar um die Folgeweise des Verfahrens zu erklären. Man verbindet die zwei Histogramme und berechnet die gemeinsame Fläche. Histogramme sind Abbildungen von einer Definitionsmenge in eine Wertemenge:

$$H : \mathbb{D} \rightarrow \mathbb{W}$$

Es sei

$$f(K_1) = \sum_{i=1}^{N} f(K_i)$$

die relative Häufigkeit von x und es sei

$$f(K_2) = \sum_{j=1}^{M} f(K_j)$$

die relative Häufigkeit von c.
Man berechnet den Schnitt der zwei Histogramme bzw. die gemeinsame Fläche der Rechtecke und erhält die Ähnlichkeitsmetrik.
Die Formel :

$$SIM_{histogr}(A \times B) = f_{gem}(K) = f(K_1) \cap f(K_2)$$

In dem Beispiel in Abbildung 2.18 bekommen wir durch den Schnitt ihres Ergebnisshistogramms die Ähnlichkeitsmetrik.

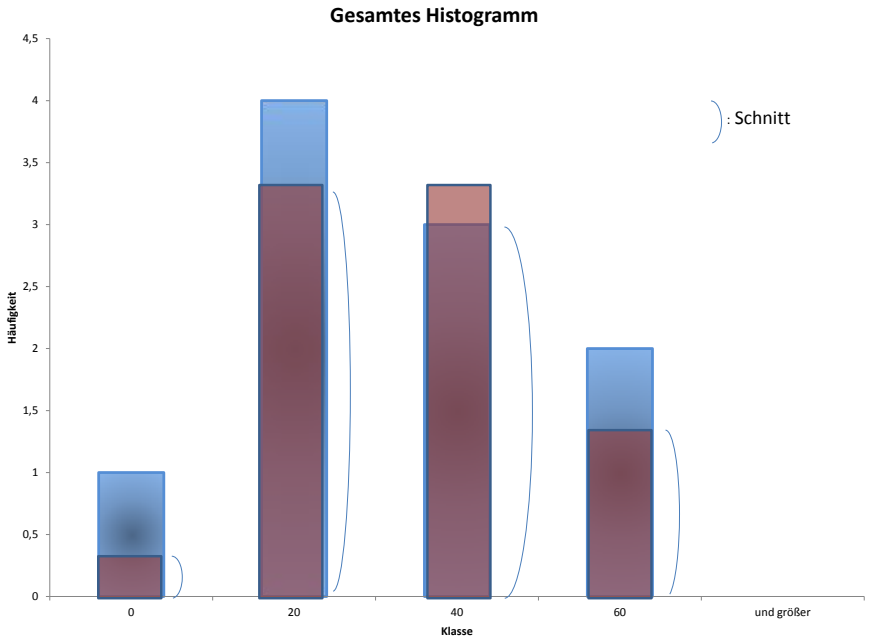

Abbildung 2.19: Matching zweier Histogramme

2.6 Ähnlichkeitsverfahren zwischen Subprozessen bzw. Prozessen

2.6.1 Strukturelle Ähnlichkeitsmetrik

Im Bereich der Ähnlichkeitsmetrik zwischen zwei Subprozessen sollen wir berechnen, wie ähnlich sie zueinander sind. Um es herauszufinden, wenden wir die sogenannte strukturelle Ähnlichkeitsmetrik an. Struktur bedeutet die Art und Weise, wie die Elemente (Aktivitäten) eines Prozesses aufeinander bezogen sind, d.h durch Kanten verbunden sind, so dass der Prozess entsteht und sich erhält. Je weniger die Schritte, von dem ersten zu dem zweiten Subprozess zu kommen, sind, desto ähnlicher sind die Subprozesse zueinander [9].
Wir betrachten einen Subprozess als einen Graph. Graph Matching ist der Prozess, eine Ähnlichkeit zwischen Knoten und Kanten zweier Graphen zu finden, die einige Bedingungen erfüllen. Diese garantieren, dass gleiche Subgraphen des einen auf gleichen Substrukturen des anderen Graphen abgebildet werden.
Wir unterscheiden zwischen zwei Arten von Prozessvergleich (engl. Graph Matching): den exakten Graphen Matching und den unexakten. Die präzise Berechnung des exakten Graph Matching ist eine Methode, die auf Subgraphisomorphismus basiert. Zwei Graphen heißen zueinander isomorph, wenn es einen Isomorphismus zwischen ihnen gibt. Isomorphismus ist eine Abbildung zwischen zwei Strukturen, durch die die Teile einer Struktur

auf *bedeutungsgleiche* Teile einer anderen Struktur umkehrbar eindeutig (bijektiv) abgebildet werden. Subgraphisomorphismus ist ein Graphisomorphismus zwischen einem Graph und einem knoteninduzierten Subgraph eines zweiten Graphes. Die zwei Subgraphen sind isomorph, wenn sie die gleiche Anzahl von Knoten und Kanten haben und ein 1:1 Mapping zwischen den Knoten und den Kanten der zwei Graphen existiert. Das Problem des maximal ähnlichen Subgraphen Isomorphismus ist NP-hard [**?**].
Die Voraussetzung, dass eine bedeutende Anzahl von Knoten und Kanten in zwei Graphen identisch sein muss, ist in Anwendungen auf Graphen, die aus wirklichen Daten herausgezogen sind, nicht realistisch. Die unexakte Methode von Graph Matching bietet eine grosse Reihe von Modellen für strukturelle Ähnlichkeit, da ein Matching zwischen Graphen unterschiedlicher Größe stattfinden kann.

Graph Edit Distance

Die bekannteste Methode ein Matching zu berechnen, ist die Methode von Graph Edit Distance. Die minimale Anzahl der Operationen (Einfügen, Löschen, Ersetzen) um den ersten Prozess zum zweiten zu transformieren heißt Graph Edit Distance. Diese Anzahl ergibt sich aus der Differenz der zwei Subprozesse. Graph Edit Distance basiert auf das Konzept von Levenshtein Distanz (s. abschnitt 2.5.1). Paare von Prozess-Graphen werden verglichen. Wir versuchen ein optimales Verfahren zu definieren, die minimalen Kosten bei der Umgestaltung eines Graphens zu einem anderem zu berechnen [4],[13].

Die Grundidee von Graph Edit Distance soll die Unähnlichkeit von zwei Graphen durch den minimalen Betrag der Operationen (Einfügen, Löschen, Editieren) definieren. Für jedes Graphenpaar besteht eine Sequenz von editierten Operationen (engl. Substitution) oder editierten Pfaden (engl. Edit Path), die einen Graphen in den anderen umgestalten. Ein gültiger solcher Pfad kann immer durch das Entfernen aller Knoten und Kanten vom ersten Graphen und dann Einfügen aller Knoten und Kanten des zweiten Graphen gebaut werden. Jedoch sagt diese Prozedur nichts darüber aus, ob die zwei Graphen strukturell ähnlich sind. Ersetzungen von Knoten und Kanten können als positives Matching betrachtet werden, sodass Editierungen den gemeinsamen Teil von zwei Graphen identifizieren. Einfügen- und Löschen-Operationen finden auf Knoten und Kanten statt. Knoten und Kanten des anderen Graphen können nicht mit ihnen verglichen werden. Dieser editierte Pfad kann als ein Modell verstanden werden, das beschreibt, welche Knoten und Kanten eines Graphen mit Knoten und Kanten eines anderen Graphen erfolgreich verglichen werden können.
Jede elementare Operation hat Kosten, die durch eine Kostenfunktion gegeben werden. Ein Graph Edit Distance Algorithmus erschafft mögliche Kombinationen derjeniger Operationen mit den minimalen Gesamtkosten. Wir ziehen elementare Transformationsoperationen in Betracht:

- Knotenersatz: Knoten eines Graphen wird durch einen anderen Knoten ausgewechselt.
- Knoteneinfügung/Entfernung: Ein Knoten wird darin eingefügt oder von einem Graphen gelöscht.

- Kanten-Einfügung/Löschen: Eine Kante wird darin eingefügt oder von einem Graphen entfernt.

Wir nehmen an, dass die Kosten eines Knotenersatzes gleich (1- die Ähnlichkeit der Knoten) ist. Die Ähnlichkeit von Knoten ist durch die Ähnlichkeit der Knotenlabel festgesetzt (s. Abschnitt 2.5.3.

Definition und Formel

Es seien $SubPr_1$ (V_1, N_1, E_1, ι, C_1) und $SubPr_2$ (V_2, N_2, E_2, ι, C_2) zwei Subprozesse. Es sei, weiterhin Ω_1 und Ω_2 zwei Sets von Textlabel und Λ_1 und Λ_2 Funktionen zur Ordnung von Labels zu Prozesselementen. M_1 und M_2 sind die Mappings von Aktivitäten, Daten, Konnektoren, Variablen und Ressourcen der zwei Prozesse. Die Graph Edit Distance zweier Subprozesse is definiert durch:

$$d(SubPr_1, SubPr_2) = \min \sum_{i=1}^{k} c(e_i),$$

wobei e die Operationen (Einfügen-, Löschen-, Ersetzen-Operationen) sind, die einen Graphen in den anderen umgestalten und c die Kostenfunktion dafür ist.

Edit Distance Algorithmus

Ein best-first Suchalgorithmus für die Berechnung der Ähnlichkeit ist der folgende: Der Algorithmus wird nur in Bezug auf den Knoten die Operationen formuliert und nicht auf die Operationen der Kanten. Die Knoten werden in der Ordnung (u_1, u_2,...) abgearbeitet. In jedem Schritt wird der folgende unverarbeitete Knoten u_{k+1} des ersten Graphen ausgewählt und versuchsweise durch alle unverarbeiteten Knoten des zweiten Graphen eingesetzt. Wenn alle Knoten des ersten Graphen bearbeitet worden sind, werden die restlichen Knoten des zweiten in den Graphen in einem Einzelschritt eingefügt bzw. noch übrige im zweiten Graph gelöscht. Das Verfahren wählt immer den optimalen Pfad und endet, sobald ein ganzer Pfad editiert worden ist. Im exakten Algorithmus ist jeder Knoten des Graphen erlaubt, mit jedem Knoten eines anderen Graphen verglichen zu werden. Diese Flexibilität macht ihm besonders passend für verrauschte Daten, aber vergrößert andererseits die Komplexität im Gegensatz zum einfacheren Graphen. Der Zeitaufwand des Algorithmus ist exponential zu den Knoten der beiden Graphen.

Approximierter Edit Distance Algorithmus (Greedy Algorithmus)

Greedy Algorithmus

Ein Algorithmus ist eine Schritt-für-Schritt Prozedur um ein Problem zu lösen. Die Idee hinter einem Greedy-Algorithmus ist, ein einheitliches Verfahren durchzuführen und immer wieder zu wiederholen, bis es zu Ende ist und sehen, welche Ergebnisse es produzieren wird. Greedy-Algorithmen sind einfach und unkompliziert. Sie sind spontan und treffen Enscheidungen auf der Grundlage von Informationen ,ohne sich Gedanken über

die Wirkung dieser Entscheidungen in der Zukunft zu machen. Sie sind einfach zu implementieren und meistens recht effizient. Greedy-Algorithmen werden verwendet, um Optimierungsprobleme zu lösen.
Der Graph Edit Distance Algorithmus in vorherigen Kapitel ist hauptsächlich uneffizient, weil die Zahl dessen zu bewertenden exponential wächst, um Graphen geradlinig anzubauen. Dafür wird eine andere Methode vorgeschlagen : Die Lösung des Algorithmus ist, ein suboptimaler editierter Pfad p zwischen den zwei Eingangsgraphen auf eine schrittweise Weise zu bauen. Am Anfang wird es eine Substitution ausgewählt. In manchen Fällen kann es möglich sein, einen Knoten aus dem ersten Graphen und einen Knoten vom zweiten Graphen abzuleiten. Sonst kann ein versuchsweiser zusammenpassender Graph-Prozess verwendet werden, um mehrere viel versprechende Kandidaten zu erzeugen. Im folgenden Schritt werden die durch die zwei Knoten der Substitution definierten Nachbarschaftsubgraphen verglichen. Das Ergebnis des Nachbarschaft-Matchings, eines editieren Pfads zwischen den zwei Nachbarschaftsuubgraphen, wird dann zur Lösung des editieren Pfades p. Dieser ist ein gieriger Algorithmus, d.h. Operationen werden nur hinzugefügt, aber nie von der Lösung p entfernt. Schließlich werden alle Knotenersetzungen im editieren Pfad, die sich aus der Nachbarschaft-Maching zusammenpasst, zur First-In-First-Out (FIFO) Warteschlange hinzugefügt. Dieselben zusammenpassenden Schritte werden dann für alle weitere Schritte durchgeführt. Die Nachbarschaft, die zusammenpasst, wird in Bezug auf eingesetzte Knoten darin ausgeführt. Schließlich, wenn alle Knotenersetzungen bearbeitet sind, werden die restlichen Knoten vom den ersten Graphen gelöscht und die restlichen Knoten vom zweiten Graphen werden eingefügt. Der editierter Pfad ist die approximative Lösung.

2.6.2 Prozesse

Genauso wie bei Subprozessen, wird für die Berechnung der Ähnlichkeit zwischen zwei Prozessen, die strukturelle Ähnlichkeitsmetrik angewendet. Je weniger die Schritte, von dem ersten zu dem zweiten Prozess zu kommen, sind, desto ähnlicher sind die Prozesse zueinander. Die Berechnung der Ähnlichkeit findet mithilfe vom Greedy Algorithmus (s. AbschnittGreedy) [9].

Definition und Formel

Es seien Pr_1 (V_1, N_1, E_1, ι, C_1) und Pr_2 (V_2, N_2, E_2, ι, C_2) zwei Subprozesse. Es sei, weiterhin Ω_1 und Ω_2 zwei Sets von Textlabel und Λ_1 und Λ_2 Funktionen zur Ordnung von Labels zu Prozesselementen. M_1 und M_2 sind die Mappings von Aktivitäten, Daten, Konnektoren, Variablen und Ressourcen der zwei Prozesse. Die Graph Edit Distance zweier Prozesse is definiert durch:

$$d(Pr_1, Pr_2) = \min \sum_{i=1}^{k} c(e_i),$$

wobei die Vergleichsoperation zwischen Aktivitäten der zwei Prozessen wird durch die folgende Aggregatfunktion stattgefunden:

$$sim_{aggregate}(a,b) = \frac{\alpha * sim_{label}(a,b) + \beta * sim_{contx(a,b)} + \gamma * sim_{data(a,b)} + \delta * sim_{res(a,b)} + \epsilon * sim_exe(a,b)}{\alpha + \beta + \gamma + \delta + \epsilon}$$

3 Prototyp und Evaluierung

3.1 Aufbau des Prototyps

3.1.1 Prototyp Implementierung

Der dBOP Editor besteht aus drei Schichten. (s. Bild unten)

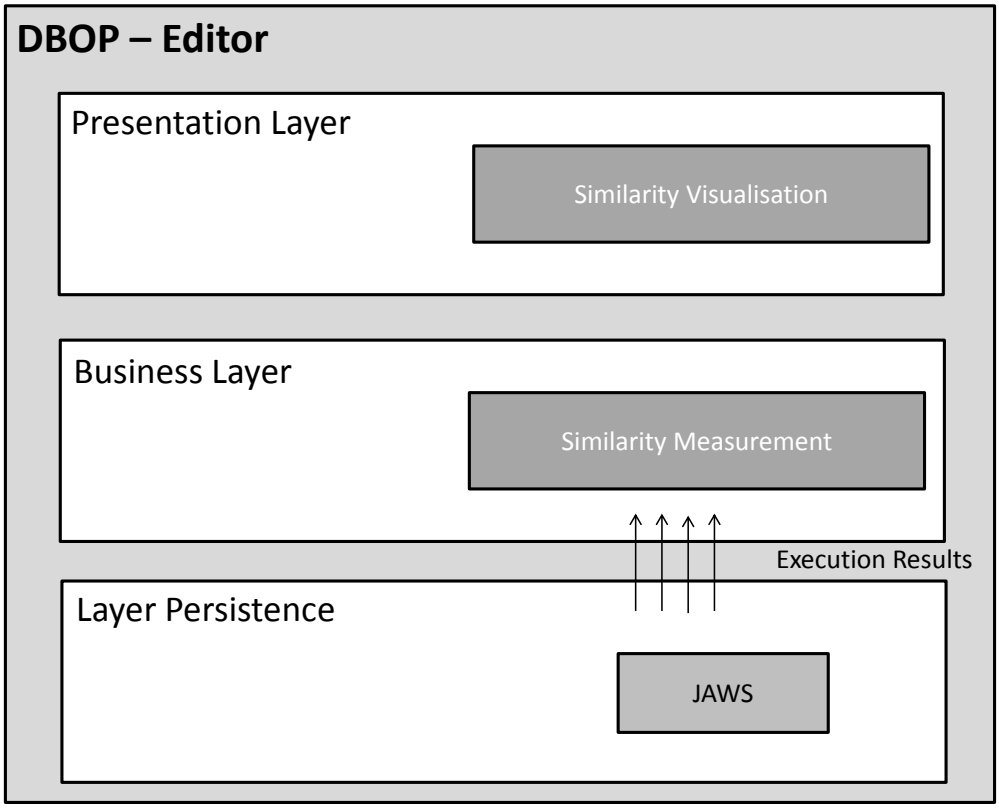

Abbildung 3.1: Prototyp Implementierung

3.1.2 Auszeichnung und Evaluierung des Prototyps

Die Auswertung der Ergebnisse in allen verschiedenen Metriken findet im dBOP Editor statt. Man definiert einen Prozess und fügt alle nötigen Informationen ein.

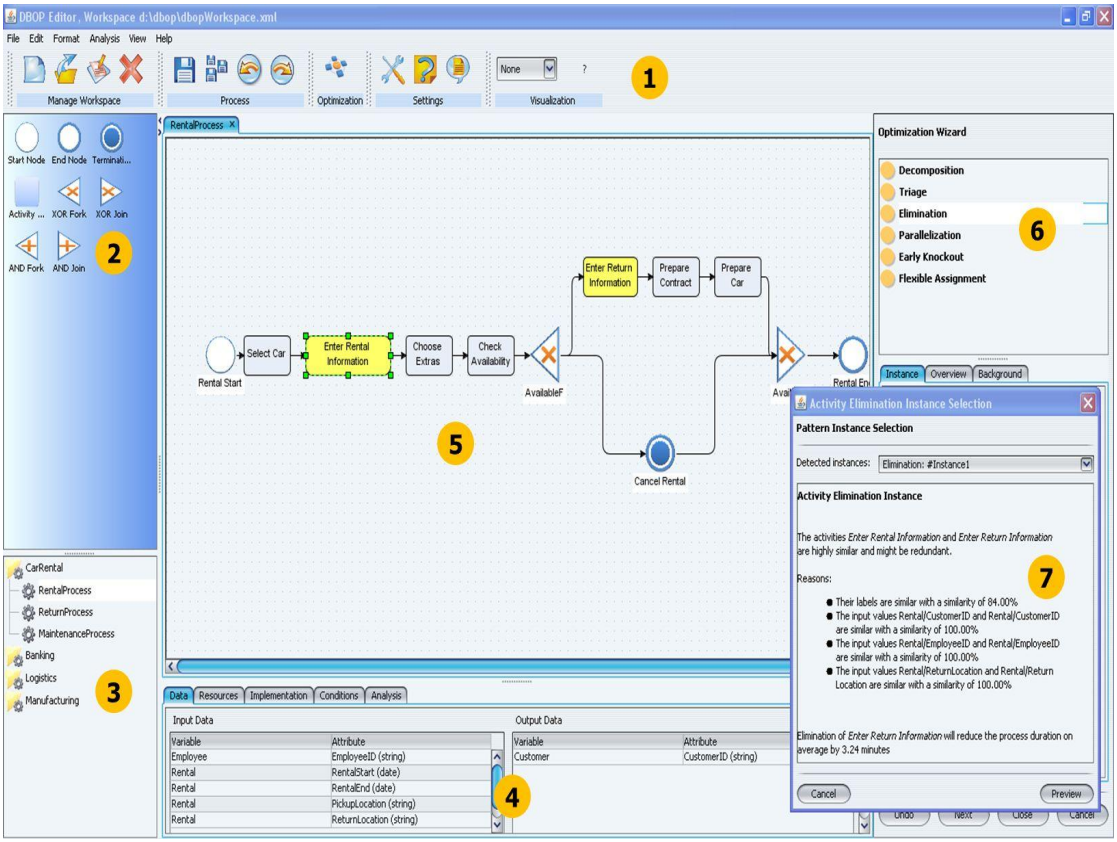

Abbildung 3.2: dBOP Editor

Der Hauptbestandteil der Integrationsschicht ist der Matching Editor, welcher die zusammenpassende beschriebene Annäherung durchführt. Für die Speicherung des einheitlichen Prozesses und die betrieblichen Daten, verwendet unser Prototyp grundsätzlich DB2 Unternehmen. Im Mittelpunkt der Analytik ist die Schicht der dBOP Analyzator. Er verbindet Datenaufbereitung, metrische Berechnung und Daten, die Techniken abbauen, um die notwendigen Eingänge für die Optimierungsschicht zur Verfügung zu stellen. Es ist eine kundenspezifische Durchführung.
In dem Bild ist ein Beispiel vorhanden. Die Nummerierung der Bereiche im Editor wird uns helfen, die Schrittweise eines Ähnlichkeitsmetrikverfahrens besser zu verstehen. Man kann einen neuen Prozess kreieren (s. 1 im Bild). Ein Prozess hat Start- und Endknoten, Terminierungsknoten, Aktivitäten und XOR-Fork, XOR Join, AND Fork und AND Join als Arten von Konditionen. Diese sind die Komponenten, mit deren Hilfe ein Prozess in dBOP Editor kreiert werden kann (s. 2 im Bild). Im Bereich 3 sieht man eine Liste der verfügbaren definierten Prozesse im Editor. Bereich 5 ist der eigentliche Bereich, wo ein Prozess modelliert wird. Man schiebt die einzelnen Komponente vom Bereich 2 in den Bereich des Editors. Es werden nur die Aktivitäten und die Verbindungen zwischen ihnen durch Bedingungen angemalt. Die anderen wichtigen Informationen, wie Daten (Eingabe-

und Ausgabedaten), Ressourcen, Implementierung, Analyse, Dauer und Frequenz der Ausführung der Aktivitäten werden im Bereich 4 definiert. Durch diese Informationen -abhängig davon, welche Art von Metrik verwendet wird- werden die Ergebnisse der Ähnlichkeit berechnet. Zum Beispiel definiert und wählt man selber bei der Einfügung von Daten, ob sie als Eingabe- oder Ausgabedaten betrachtet werden sollen und fügt die entsprechenden Attribute zu jedem Data ein. Man wählt in der Menüleiste *Analysis* z.B *Node Ähnlichkeitsmetrik*. Man wählt die zwei zu vergleichenden Knoten und drückt auf *Calculate*. Die zwei Knoten werden nach allen schon definierten Metriken berechnet. Eine gesamte Berechnung aller Metriken wird am Ende angezeigt.

4 Abschluss und Ausblick

Das Problem der Ähnlichkeitsmessung ist allgemein nicht sonderlich neu, nur die Form der Anwendung.

4.1 Zusammenfassung und Erweiterungsmöglichkeiten

Geschäftsumgebungen werden immer komplizierter und dynamischer und Informationstechnologie einschließlich des Netzwerkanschlusses entwickelt sich schnell. Dafür ist der optimale und effiziente Grad der Ähnlichkeit zwischen Prozessmodellen für das Management, den Wiedergebrauch, und die Analyse von Geschäftsprozessmodellen von grosser Bedeutung. Die Motivation für diese Diplomarbeit war es, Möglichkeiten herauszuarbeiten, wie man effizienter zwei Prozesse vergleichen kann. Am Anfang wurden unterschiedlichen Methoden zur Messung der Ähnlichkeit zwischen Teile eines Prozesses. Diese hat später geholfen, eine Aggregatfunktion zu formulieren, die mit Berücksichtigung aller Komponente eines Prozesses eine gesamte Formel für die Ähnlichkeit zu definieren. Die Idee dafür war einen Konzept zu entwickeln, wobei alle Bestandteile eines Prozesses in Anspruch genommen werden. Der Anwender, der die Ähnlichkeit berechnen möchte, kann selber sich entscheiden welche Ähnlichkeitsverfahren ihn am wichtigsten zu berücksichtigen, sind. Um die Qualifikation die Metrik zu verwenden, um eine Sammlung von Geschäftsprozessmodellen zu suchen, muss Arbeit noch getan werden. Bis jetzt haben viele Leute sich darauf konzentriert, die Metrik, aber nicht effizienten Algorithmen zu entwickeln. Eine Erweiterungsmöglichkeit wäre die Implementierung von Metriken, die schon mathematisch definiert wirden sind und sie in der Aggregatfunktion miteinzuziehen.

Literaturverzeichnis

[1] *Game Engige Toolset Development.* Wihlidal, Graham, 2006.

[2] F. Brosius. Spss 8. *International Thomson Publishing*, 1998.

[3] Pradeep Fienberg Stephen E. Cohen, William W. Ravikumar. A comparison of string distance metrics for name-matching tasks. *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03)*, 2003.

[4] M. Dijkman, R. ; Dumas and L. García-Bañuelos. Graph matching algorithms for business process model similarity search. *In Pproc of BPM 2009*, 2009.

[5] M. ; van Dongen B. ; Käärik R. andMendling J. Dijkman, R. ; Dumas. Similarity of business process models : Metrics and evaluation. *Working Paper 269, Beta Research School*, September Eindhoven, 2009.

[6] A. Ehrig, M. ; Koschmider and A. Oberweis. Measuring similarity between semantik business process models. *In Proc. of APCCM*, pages 71–80, 2007.

[7] Tolga Ergin. Evaluation of automated business process optimization. Master's thesis, 2011.

[8] G. Génova. What is a metamodel : the omg's metamodeling infrastructure. *Modeling and metamodeling in Model Driven Development*, Warsaw, 2009.

[9] J. Jeh, G. ;and Widom. *SimRank: a Measure of Structural-Context Similarity.* In KDD, 2002.

[10] F. Leymann and D. Roller. *Production Workflow: Concepts and Techniques.* Prentice Hall, Upper Saddle River, 2000.

[11] A. Maguitman and G..F. Menczer. Algorithmic detection of semantic similarity. 2005.

[12] S. Neugebauer, M. ; Schulz. Verfahren und vorrichtung zur erzeugung einer trefferliste bei einer automatischen spracherkennung. 2010.

[13] Michel Neuhaus. Aufsatz: 3. graph edit distance. *Bridging the gap between graph edit distance and Kernel machines*, pages 21–56.

[14] F. Niedermann and B. Mitschang. Formalized patterns for business process optimization. Technical report, University of Stuttgart, 2011.

[15] S. ; Mitschang B Niedermann, F. ; Radeschütz. Design-time process optimization through optimization patterns and process model matching. In *Proceedings of the 12th IEEE Conference on Commerce and Enterprise Computing*, 2010.

[16] Sylvia Mitschang Bernhard Niedermann, Florian Radeschütz. Design-time process optimization through optimization and process model matching. *12th Conference on Commerce and Enterprise Computing*, 2010.

[17] UCF (University of Central Florida). What is process analysis. 1999.

[18] P. N. Ristad, E. S. und Yianilos. Learning string-edit distance. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 20*, pages 522–532, 1998.

[19] A. ;and Dumke R. Rud, D. ; Schmietendorf. Resource metrics for service-oriented infrastructures. *In Proceedings of the Workshop on Software Engineering Methods for Service Oriented Architecture*, pages 90–98, 2007.

[20] T. Simpson and T. Dao. Wordnet-based semantic similarity measurement (capturing the semantic similarity between two short sentences based on the wordnet dictionary). 2010.

[21] R. van Dongen, B. ; Dijkman and J. Mendling. Measuring similarity between business process models. *In Proc. of CAise 2008*, 5074 of LNCS:450–464, 2008.

[22] S. Wan and Angryk R. A. Measuring semantic similarity using wordnet-based context vectors. *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference*, 2007.

[23] M. Weidlich, M ; Weske and J. Mendling. Change propagation in process models using behavioural profiles. *IEEE International Conference on Services Computing*, 2009.

[24] R. Yan, Z. ; Dijkman and P. Grefen. Business process model repositories - framework and survey. *Information Sciences*, 2009.

[25] D. Yang and D. M.W. Powers. Measuring semantic similarity in the taxonomy of wordnet. *28th Australasian Computer Science Conference*, pages 315–322, 2005.

[26] Wang A.K.H. ; Feng J. Zeng, Z. ; Tung and L. Zhou. Comparing stars: On approximating graph edit distance. *Proceedings of the VLDB Endowment 2(1)*, pages 25–36, 2009.