

# Entwurf und Implementierung einer Werkzeugunterstützung zur sprachlichen Analyse und automatisierten Transformation von Projektlastenheften im Kontext der Automobilindustrie

An der Fachhochschule Dortmund

im Fachbereich Informatik

Studiengang Informatik

Vertiefung Praktische Informatik

erstellte Thesis

zur Erlangung des akademischen Grades

Bachelor of Science

B. Sc.

von Aaron Schul,

geboren am 24.06.1997

und Felix Ritter

geboren am 31.08.1997

Betreuung durch:

Prof. Dr. Sebastian Bab und Prof. Dr. Steffen Helke

Dortmund, 28.02.2019

## Zusammenfassung

Ein weitläufiges Problem in der Autoindustrie ist die effiziente Verarbeitung von Pflichtenheften. Einer der Gründe dafür ist, dass viele verschiedene Bereiche bei der Entwicklung der Fahrzeuge und sogar kleinster Einzelteile beteiligt sind. So müssen zu Beginn einer Produktentwicklung etwa Betriebswirtschaftler, Designer, Techniker und Ingenieure zusammen ein Dokument entwerfen, das die Produktmerkmale widerspiegelt. Darin müssen die Anforderungen an das gewünschte Produkt so genau beschrieben sein, sodass es anhand dieses Dokuments entwickelt werden kann. Als Experten ihrer jeweiligen Domäne weiß dabei jeder genau, was dazu nötig ist. Sobald jedoch Auswirkungen über die eigene Domäne hinausgehen, kann es schnell passieren, dass Widersprüche oder Abhängigkeiten entstehen. Diese werden später schnell übersehen, da solche Zuständigkeiten nicht eindeutig geklärt sind. Das Resultat ist dann ein Fehler in der Entwicklung, welcher von zeitlicher Verzögerung über zusätzliche Kosten bis hin zum Abbruch des Projekts führen kann. Methoden aus dem Natural Language Processing (NLP) können dabei helfen, die Überprüfung der Pflichtenhefte auf Konsistenz stark zu vereinfachen oder sogar teilweise zu automatisieren. Dafür kann beispielsweise der Text in den Lastenheften analysiert und dessen Inhalt innerhalb einer Wissensbasis, sog. Ontologie, abgespeichert werden. In dieser Arbeit wird daher die Entwicklung zweier NLP-basierter Werkzeuge zur Vereinfachung bzw. Lösung dieser Probleme vorgestellt werden. Zum einen ein *Requirements-to-Boilerplate-Converter*-Werkzeug (R2BC), welches dem Nutzer helfen soll, das Pflichtenheft eines Auftraggebers in die betriebsinternen Richtlinien und Standards zu überführen. Zum anderen wird der *Delta-Analyser* vorgestellt, welcher auf dem R2BC aufbaut, indem er automatisch zwei homogene Lastenheft vergleicht und dadurch im Kontext des gesamten Pflichtenhefts Widersprüche und Abhängigkeiten herausstellt. Ziel ist es dabei, jeweils einen Prototypen als „*proof of concept*“, also als Machbarkeitsstudie, zu implementieren. Nach dieser Vorstellung wird zusätzlich das weitere Potential der Programme erläutert, welches sich bei der Entwicklung der Prototypen gezeigt hat. Diese bietet Vorschläge zur weiteren bzw. vollständigen Implementierung der Programme.

## **Abstract**

A widespread problem in the auto industry is the efficient processing of specifications. One of the reasons for this is that many different areas are involved in the development of vehicles and even the smallest of parts. For example, at the beginning of product development, managers, designers, technicians, and engineers all need to design a single document together. The requirements for the desired product must be described in such detail that it can be developed using this document. As experts in their domain, everyone knows exactly what is needed. However, as soon as effects go beyond the own domain, it can quickly happen that contradictions or dependencies arise. These are quickly overlooked later, as such responsibilities are not clearly clarified. The result is then a mistake in the development, which can lead from time delay over additional costs up to the demolition of the project. Methods from Natural Language Processing (NLP) can help to simplify or even partially automate the review of functional specifications for consistency. For example, the text in the specifications can be analyzed and its contents stored within a knowledge base, so-called ontology. In this work, therefore, the development of two NLP-based tools to simplify or solve these problems will be presented. On the one hand, there is a program called Requirements-to-Boilerplate Converter (R2BC), which should help the user to transfer the specifications of a client into the company's internal guidelines and standards. On the other hand, the so-called delta analyzer is shown, which is based on the R2BC, in that it automatically compares two homogeneous specification sheets and thereby highlights contradictions and dependencies in the context of the entire specification. The goal here is to implement a prototype as a proof of concept of the program concepts from ZIC19. Due to the limited resources of this work, a list with further potential of the programs, which has been shown during the development of the prototypes, is additionally explained. This offers suggestions for further or complete implementation of the programs.

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>7</b>
1.1	Motivation . . . . .	7
1.2	Unternehmenskontext . . . . .	10
1.2.1	Einsatz bei HELLA . . . . .	10
1.2.2	Anforderungsmanagement im Unternehmen . . . . .	10
1.3	Hypothese . . . . .	11
1.4	Methodik . . . . .	12
1.5	Aufbau der Arbeit . . . . .	12
1.6	Autoren . . . . .	13
<b>2</b>	<b>Stand der Technik</b>	<b>14</b>
2.1	Rückblick auf die Projektarbeit und Grundlagen zu NLP . . . . .	14
2.2	Ontologien . . . . .	17
2.2.1	Definition . . . . .	17
2.3	Verwandte Arbeiten . . . . .	17
<b>3</b>	<b>Betriebliches Umfeld - Hella Use-Case</b>	<b>18</b>
3.1	Re-Prozesse bei Hella und allgemein in Firmen . . . . .	20
3.2	Betriebliche Anforderungen . . . . .	20
3.3	Ansatz und Konzept unserer Werkzeuge . . . . .	20
<b>4</b>	<b>R2B-Converter</b>	<b>21</b>
4.1	Architektur Klassen und Verteilung der Ressourcen . . . . .	21
4.2	Implementierung (bisschen Code, GUI, Listenarchitektur, Work-flow für User . . . . .	21
4.3	mögliche Erweiterungen . . . . .	21
4.4	Test . . . . .	21
4.4.1	Methodik . . . . .	21

4.4.2	Durchführung . . . . .	21
4.4.3	Ergebnisse . . . . .	21
<b>5</b>	<b>Delta-Analyse</b>	<b>22</b>
5.1	Architektur . . . . .	22
5.2	Implementierung . . . . .	22
5.3	mögliche Erweiterungen . . . . .	22
5.4	Test . . . . .	22
5.4.1	Methodik . . . . .	22
5.4.2	Durchführung . . . . .	22
5.4.3	Ergebnisse . . . . .	22
<b>6</b>	<b>Evaluation</b>	<b>23</b>
6.1	Auswertung der Testresultate . . . . .	23
6.2	Mehrwert? . . . . .	23
6.3	Ziel erreicht? Hypothese reviewen und schwafeln . . . . .	23
<b>7</b>	<b>Fazit</b>	<b>24</b>
7.1	Zusammenfassung . . . . .	24
7.2	Ausblick . . . . .	24

# Tabellenverzeichnis

# Abbildungsverzeichnis

# Abkürzungsverzeichnis

<b>R2BC</b>	Requirements-To-Boilerplate-Converter
<b>DA</b>	Delta-Analyser
<b>CRS</b>	Customer Requirement Specification
<b>OEM</b>	Original Equipment Manufacturer
<b>E-AE</b>	Electronics-Advanced Engineering
<b>PMT</b>	Projects, Methods and Tools



## **Danksagung**

An dieser Stelle möchten wir uns bei allen Personen bedanken, die uns bei unserer Bachelorarbeit und während des Studiums begleitet und unterstützt haben.

Der Dank gebührt unseren Familien und Freunden aus der Heimat sowie all jenen tollen Menschen, die wir durch das Studium an der FH Dortmund erst kennenlernen durften.

Wir danken auch allen Kollegen aus der Abteilung E-AE bei der Firma Hella in Lippstadt für ihr Interesse und die großartige Zusammenarbeit. Die einzigartigen Erfahrungen, die wir während unseres Praktikums und später während der Bachelorarbeit in der Firma sammeln konnten, haben uns wirklich motiviert. Besonderer Dank geht dabei an Konstantin Zichler, der weit mehr als nur Betreuer in der Firma, sondern auch Mentor und Ideengeber unserer Arbeit war.

Schlussendlich danken wir unseren Betreuern Prof. Dr. Sebastian Bab und Prof. Dr. Steffen Helke für die zahlreichen Tipps und Anregungen während der Bachelorarbeit.

# Kapitel 1

## Einführung

### 1.1 Motivation und thematische Grundlagen

In einer Vielzahl von Firmen stellt die Erhebung von Anforderungen den ersten Schritt bei der Entwicklung von neuen Produkten dar. Diese Anforderungen manifestieren sich später in der zu entwickelnden Hard- und Software. Spätere Produktmerkmale müssen dabei weit vor der tatsächlichen Entwicklung berücksichtigt werden und fließen in das Anforderungsdokument ein. Die Erhebung und Identifikation dieser Anforderungen wird als *Requirements-Engineering* bezeichnet [BAL10].

Dieser Prozess wird dabei sowohl bei kompletten Neuentwicklungen, als auch bei der Adaption eines bereits vorhandenen Produktes auf neue Projekte durchlaufen. Zur Rolle von Requirements-Engineering in Projekten siehe etwa [MW02] und [HL01].

Dem Betrieb liegen anschließend die Aufgaben und Ziele für die Entwicklung als Dokumente vor, auf deren Basis die Produktentwicklung beginnen kann. Die Entwicklung unterliegt dabei bestimmten Kriterien und Faktoren, die den unternehmerischen Erfolg beeinflussen. Neben betriebswirtschaftlichen Einflüssen wie der Einordnung des Produktes in der Wertschöpfungskette sind es dabei besonders technische Anforderungen an das Produkt, die definiert und während der Produktentwicklung eingehalten werden müssen.

#### Anforderungen im betrieblichen Entwicklungsprozess

Das Resultat wird als sogenanntes Pflichtenheft, in Unternehmen häufig als CRS (*Customer Requirement Specification*) bezeichnet, mithilfe dessen nun

die Entwicklung eines Produktes begonnen werden kann [DGE19]. Ein später hergestelltes Produkt sollte möglichst allen beschriebenen Anforderungen entsprechen. Da Betriebe jedoch nur selten alle einzelnen Komponenten des späteren Produktes selbst herstellen können, sind sie auf Zulieferer angewiesen, die einzelne Komponenten entwickeln und produzieren. Dieser Hersteller fertiger Komponenten aus Einzelteilen wird als *OEM* (*Original Equipment Manufacturer*).

In diesem Fall wird dem Zulieferer ein Pflichtenheft übergeben, dass allen Anforderungen des Kunden genügt. Das bedeutet jedoch, dass auch die Produktentwicklung anhand des Pflichtenheftes an den Zulieferer ausgelagert wird. Dem Zulieferer werden damit Möglichkeiten geboten und Verantwortung übertragen, das Produkt weiter zu definieren. Einerseits können etwa interne Richtlinien und Erfahrungen auf dem Gebiet der Produktentwicklung aus früheren Projekten integriert werden, andererseits aber auch weitere Optimierungen sowie Merkmale, die noch nicht im Pflichtenheft dokumentiert sind. Somit agiert der Zulieferer nicht nur als produzierende Fabrik, sondern nimmt am Entwicklungsprozess teil.

Die Anforderungen an ein Produkt, etwa technische Rahmenbedingungen und Qualitätsanforderungen, setzen sich dabei aus einer Auflistung von funktionalen und nicht-funktionalen Anforderungen zusammen [BAL10]. Häufig enthält das Pflichtenheft auch weitere Informationen und abseits von Text Abbildungen, die einzelne Anforderungen ergänzen.

### **Umstände der Anforderungsdefinition**

Die Anforderungen werden dabei von vielen verschiedenen Domänenexperten beim OEM formuliert und in das Pflichtenheft für die Entwicklung eingetragen. Verschiedenste Akteure aus einem Unternehmen sind dabei an der Festlegung der Anforderungen an ein Entwicklungsprojekt bzw. Produkt beteiligt. Beteiligte sind etwa Produktdesigner, Ingenieure und Systemtechniker, die an verschiedenen Stellen im Lastenheft Anforderungen an eine Komponente festlegen und an entsprechenden Stellen vermerken. Diese Beteiligten sind in der Regel auf ihren Bereich spezialisiert und nicht interdisziplinär, da häufig an verschiedenen Orten und an spezifischen Aspekten eines Produktes in mehreren Projektteams entwickelt wird.

Projektteams aus einem Bereich tauschen sich häufig nicht untereinander aus und legen unabhängig von anderen Beteiligten Anforderungen fest. Demzufolge sammeln sich im Lastenheft schon bei der Entstehung beim OEM

verschiedenste Merkmale einer Komponente, die aber nicht im Bezug zueinander stehen und sich im schlimmsten Fall gegenseitig ausschließen. Ferner gibt es sprachliche Eigenheiten der Autoren und sprachliche Fehler, die innerhalb des Teams nicht auffallen und später nicht mehr korrigiert werden. Auch gibt es unternehmensinterne Richtlinien für die Formulierung von Anforderungen beim OEM, die das Verständnis auf Seite des Zulieferers erschweren können. Christian Allmann, Lydia Winkler und Thorsten Kölzow haben in [AWK06] solche Herausforderungen für das Requirements-Engineering zwischen OEMs und Zulieferern identifiziert.

Durch die entstehende große fachliche Breite und Tiefe der Spezifikationen im Pflichtenheft, können mitunter Dokumente mit einem Umfang von von mehreren tausend Seiten entstehen. Dies bedeutet einen hohen Aufwand bei der Transformation und Bündelung der Anforderungen in ein tatsächliches Produkt beim Zulieferer.

### Prozessoptimierung durch Werkzeuge

Bei der Untersuchung von Projektlastenheften muss also nach Zusammenhängen und Bezügen zwischen mehreren Anforderungen gesucht werden, damit die Korrektheit des späteren Produktes gewährleistet ist. Die Analyse von Zusammenhängen zwischen Anforderungen stellt dabei aus Gründen der Effizienz ein Problem dar, wenn jeder Projektbeteiligte von Hand die für ihn relevanten Anforderungen aus dem Lastenheft extrahieren muss. Auch müssen die Lastenhefte an die Formulierungen und Ausdrucksweisen für Requirements-Management im Unternehmen angepasst werden.

Bislang gibt es jedoch kaum Werkzeugunterstützung, die effiziente Möglichkeiten zur automatisierten Überarbeitung und Anpassung einzelner Anforderungen aus dem Dokument bietet. Ansätze aus dem *Natural-Language-Processing*, kurz NLP, stellen gleichzeitig vielversprechende Forschungsfelder in der Informatik dar, die eine solche automatisierte Verarbeitung auf Basis von Sprachanalyse ermöglichen. Diese werden in Kap. 2 ausführlich erläutert. Syntax und Semantik der einzelnen Sätze und Zusammenhänge in Texten können auf Basis aktueller Trends wie Machine-Learning und dynamischer Programmierung zunehmend besser abgebildet werden.

## 1.2 Unternehmenskontext der Arbeit

Besonders betroffen von dem geteilten Entwicklungsprozess zwischen OEMs und Zulieferern ist die Automobilindustrie. Eine Vielzahl von Zulieferern beliefert die Automobilhersteller mit Komponenten für ihre Fahrzeuge. Einzelne Komponenten bestehen dabei häufig auch aus mehreren Einzelteilen, die wiederum von mehreren verschiedenen Zulieferern geliefert werden. [AWK06]

### 1.2.1 Einsatz bei HELLA

Einer dieser Zulieferer ist die HELLA GmbH & Co. KGaA (im folgenden HELLA genannt), die sich auf die Entwicklung und Produktion von Licht- und Elektronikkomponenten in Hard- und Software spezialisiert hat. HELLA ist ein international operierender deutscher Automobilzulieferer mit Hauptsitz in Lippstadt und zählt zu den Top 100 Automobilzulieferern weltweit [LV10]. Für weitere Information siehe [HE19] sowie die Firmenwebsite.

Zum Verfassen dieser Arbeit waren die Autoren fünf Monate bei der Firma beschäftigt. Dies beinhaltete zwei Monate Arbeit als Praktikanten und die verbleibenden drei Monate als Bacheloranden. Die Hauptaufgabe dort war die Mitarbeit im Bereich Requirements-Engineering für die Abteilung der Elektronik-Vorentwicklung *E-AE* (*Electronics-Advances Engineering*).

Die Abteilung beschäftigt sich mit dem Projektmanagement der Vorentwicklungsprojekte und beschäftigt sich, wie zuvor beschrieben, mit dem Austausch von Informationen der Firma HELLA mit diversen OEMs bezüglich gemeinsamer Projekte. Die genaue Darstellung dieser Prozesse erfolgt in Kap. 3.1. . Somit kommen dort auch die Lastenhefte der OEMs an, die dann an die Projektteams weitergeleitet werden.

### 1.2.2 Anforderungsmanagement im Unternehmen

Die Gewichtung einzelner Anforderungen in einem größeren Systemkontext fällt dort schwer, da nun Projektteams beim Zulieferer, die an der Entstehung des Lastenheftes nicht beteiligt waren, dieses verstehen sollen und auf ihre Teilkomponente anwenden müssen. Zusätzlich zu den teilweise widersprüchlichen Anforderungen aus dem Pflichtenheft kommen weitere Anforderungen des Zulieferers, wie etwa die Gewinnmarge und unternehmensinterne Richtlinien hinzu. Schlussendlich soll jedoch ein Produkt entwickelt werden, dass möglichst alle Anforderungen berücksichtigt. [MW02]

Zu diesem Zweck können bei Zulieferern spezielle Abteilungen existieren, die Projektteams bei der Identifikation und Gewichtung von Anforderungen unterstützen. Diese Abteilungen sind auf die Auswertung der Projektlastenhefte spezialisiert und erstellen Modelle zu Merkmalen des späteren Produktes. Die Experten stellen dabei die Eindeutigkeit und Verständlichkeit sicher und identifizieren relevante Anforderungen für die Mitglieder des Projektteams. Somit kann ein eigenes Anforderungsdokument für das Projektteam beim Zulieferer erstellt werden. Falls nicht vorhanden, muss das Team selbst über die Bewertung von Anforderungen entscheiden und Rücksprache mit dem OEM halten.

Bei HELLA existiert eine dementsprechende Abteilung, genannt *Projects, Methods and Tools*, kurz *PMT*. Im Rahmen unserer Beschäftigung konnte in enger Zusammenarbeit mit PMT genau ermittelt werden, wie Lastenhefte momentan verarbeitet werden, welche Verbesserungspotentiale dabei bestehen und wie diese genutzt werden können.

## 1.3 Hypothese

Hypothese dieser Arbeit ist, dass sich mithilfe von NLP Lastenhefte effizient automatisiert verarbeiten lassen. Damit kann die Arbeit von Requirements-Engineers, aber auch von Beteiligten an der Entwicklung beim Verständnis der Anforderungen, erleichtert werden. Dabei ist die Auswertung der Syntax und Semantik für ein tieferes Verständnis von Textzusammenhängen, also von verschiedenen Anforderungen, relevant.

Die sprachliche Analyse kann Requirements erkennen und kategorisieren, sowie in Beziehung zueinander setzen. Insbesondere lassen sich die beschriebenen Probleme bei der Überprüfung der Pflichtenhefte auf Konsistenz lösen oder zumindest vereinfachen. Dies kann mithilfe der in dieser Arbeit beschriebenen Programme geschehen, welche konzeptuell dargestellt und prototypisch implementiert werden.

Weiterhin soll geprüft werden, wie stark ein Mehrwert bei der Nutzung solcher Programme für ein Firma ausfallen kann. Dies beinhaltet ersparte Arbeitszeit, Kosten und vermiedene Fehler.

## 1.4 Methodik

Verschiedene Methoden aus dem Bereich Natural Language-Processing werden zunächst anhand des Rückblicks auf die Projektarbeit theoretisch und anhand praktischer Beispiele erläutert. Der betriebliche Kontext der Firma Hella wird durch genaue Darstellung der dortigen Verfahren beleuchtet.

Mit diesen Grundlagen wird dann die Unterstützung der dortigen Prozesse im Bereich Requirements-Engineering auf Basis von natürlichsprachlicher Textanalyse evaluiert. Die gewonnenen Erkenntnisse aus der Zusammenarbeit mit der Abteilung PMT wurden in Anforderungen an die zu entwickelnden Werkzeuge zusammengefasst, die als Basis für unsere Softwareentwicklung dienten. Dadurch entstand eine genaue Vorstellung der zu entwickelnden Software, welche mit Hilfe der Programmkonzepte von Konstantin Zichler ZIC19 entworfen und implementiert werden konnte.

Das Ergebnis stellten zwei Software-Prototypen auf *java*-Basis dar, deren Struktur und Funktionsweisen wir anhand von Modellen und Codebeispielen darstellen. Die Prototypen wurden mit originalen Pflichtenheften von verschiedenen OEMs und der HELLA getestet und evaluiert.

## 1.5 Aufbau der Arbeit

In der folgenden Ausarbeitung wird zunächst in Kapitel 2 ein Überblick über den Stand der Technik gegeben. Dies beinhaltet einerseits einen Rückblick auf die zuvor verfasste Projektarbeit zu NLP und einer Erläuterung der Grundlagen von Ontologien und andererseits einen Überblick von Arbeiten, die sich der gleichen oder ähnlichen Problemstellungen gewidmet haben.

Kapitel 3 beschäftigt sich für eine bessere Einordnung der beiden Programme mit dem betrieblichen Umfeld bei der HELLA. Hierzu wird zunächst, vor dem Hintergrund des momentanen Vorgehens, der Anwendungsfall der Programme genauer beschrieben. Somit lassen sich die betrieblichen Anforderungen an die Werkzeuge bestimmen, welche dann von den vorgestellten Konzepten abgedeckt werden sollen.

In Kapitel 4 wird das erste Werkzeug - der R2B-Converter - ausführlich beschrieben. Dies beinhaltet die Darstellung dessen Architektur, eine Vorstellung der Implementierung, eine Auflistung möglicher Erweiterungen und eine genaue Beschreibung des Testverfahrens der Software. Zu den Tests werden dabei präzise die Methodik, die Durchführung und die Ergebnisse

vorgestellt, welche in dem späteren Evaluations-Kapitel aufgegriffen und eingeordnet werden.

Das zweite Werkzeug - der Delta-Analyser - wird in Kapitel 5 behandelt. Dabei werden ähnlich zu Kapitel 4 die Architektur, die Implementierung, mögliche Erweiterungen und durchgeführte Tests beschrieben. Es wird genau dargelegt wie die Testergebnisse erzielt wurden, die später zusammen mit den Ergebnissen von Kapitel 4 evaluiert werden, um das Resultat der beiden Werkzeuge bewerten zu können.

Kapitel 6 stellt das zuvor erwähnte Evaluations-Kapitel dar. Hier findet zunächst die Auswertung der erhobenen Testergebnisse statt. Anschließend wird anhand dessen dargestellt, ob und inwiefern das Nutzen der Werkzeuge einen Mehrwert für die Firma erwirtschaften kann und somit im betrieblichen Umfeld eingesetzt werden sollte. Somit lässt sich die Korrektheit der in 1.2 gestellten Hypothese überprüfen.

Die Arbeit wird von Kapitel 7 abgeschlossen. Dieses beinhaltet eine Zusammenfassung der geleisteten Arbeit und der Ergebnisse. Außerdem wird zuletzt noch ein Ausblick für die Zukunft der Werkzeuge und die Lösung des Problems erläutert.

## 1.6 Autoren



# Kapitel 2

## Stand der Technik

In diesem Kapitel wird zunächst ein Rückblick auf die zuvor von den Autoren verfasste Projektarbeit und die darin enthaltenen Grundlagen von NLP gegeben. Aufgrund ihrer besonderen Relevanz für die Programme werden anschließend zusätzliche Ontologien als Teil von NLP genauer beleuchtet. Außerdem wird mit Hilfe von verwandten Arbeiten ein Überblick über die Problematik und verschiedene Lösungsansätze geboten.

### 2.1 Rückblick auf die Projektarbeit und Grundlagen zu NLP

In [RS18] sind die Grundlagen von NLP im Rahmen einer Projektarbeit zusammengefasst. In diesem Abschnitt werden die für die Arbeit relevanten Aspekte kurz wiederholt.

#### Definition und Ziel

Natural Language Processing ist zu verstehen als die automatisierte oder halb-automatisierte Verarbeitung von natürlicher Sprache mit Hilfe eines Computers. In NLP sind verschiedenste wissenschaftliche Bereiche verknüpft. Hauptsächlich handelt es sich dabei um Linguistik und Informatik, jedoch gibt es auch viele Verbindungen zur Psychologie, Philosophie Mathematik und Logik[COP04].

Ziel von NLP ist üblicherweise die Extraktion von Informationen aus einem Text durch kleinschrittige Textanalyse. Diese Informationen werden

dem Text direkt über sogenannte Annotationen angehängt oder anders abgespeichert, etwa in Ontologien, welche in einem späteren Abschnitt genauer erläutert werden.

### Beispiele von Forschung und Entwicklung

Die wissenschaftliche Erforschung von NLP entstand Mitte des 20. Jahrhunderts, als der Linguist Noam Chomsky zeigte, dass sich der englischen Sprache formalisieren und somit automatisieren lassen [CHO57].

Frühe Projekte wie der Sprachcomputer ELIZA aus den 1960er Jahren waren somit bereits in der Lage eine natürlichsprachige Frage als Texteingabe entgegenzunehmen und üblicherweise eine plausible Antwort zu liefern.

Heutzutage gibt es eine Vielzahl von Spracherkennungssoftware auf Smartphones oder Smart-Devices wie ALEXA, welche in der Lage sind gesprochene Kommandos zu erkennen und über das Internet an Server zu leiten, welche in der Lage sind diese Kommandos schnell und effizient zu verarbeiten [HAO14].

### Herausforderungen von NLP

Durch die Verbindung von Linguistik und Informatik stellt sich NLP generell der Herausforderung sowohl inhärente Probleme der Sprache, als auch Probleme der Automatisierung lösen zu müssen.

Zu den sprachlichen Problemen gehören Aspekte wie Mehrdeutigkeit, Sarkasmus und Umgangssprache bzw. Redewendungen. Diese lassen sich oft nicht einzig und allein durch den gegebenen Text erklären, sondern beruhen auf Kontext und Situation.

Vor allem bei der Erfassung von Semantik, also dem Inhalt der Texte, können komplexe Probleme auftreten. So gibt es beispielsweise Mehrdeutigkeiten, welche selbst für den Menschen nicht einfach zuzuordnen sind. So kann „Die Betrachtung des Studenten“ etwa bedeuten, dass der Student etwas betrachtet. Genauso kann es jedoch sein, dass jemand anderes den Studenten betrachtet.

Eine andere häufige Fehlerquelle ist das korrekte Zuordnen von sprachlichen Bezügen. Betrachtet man die Frage „Verkaufen Sie Handys und Computer von Samsung“, so ist diese eindeutig grammatikalisch korrekt. Dennoch ist es uneindeutig, ob sich die Frage auf nach der Marke der Handys richtet, oder ob dies lediglich auf die Computer bezogen ist.

Noch komplexer wird es, wenn sich diese Bezüge über verschiedenen Sätze bzw. Teilsätze erstrecken. So ist der Satz „Tom schenkt Tim ein Fahrrad, weil er nett ist“ ebenfalls auf zwei verschiedenen Arten zu deuten. Einerseits kann es sein, das Tom das Fahrrad verschenkt, weil Tom ein netter Mensch ist, andererseits kann es sein, das Tom das Fahrrada verschenkt, weil Tim nett zu ihm ist.

Weiterhin können auch bei der korrekten Erfassung des Sprachinhalts weitere Probleme die Kommunikation stark beeinflussen. So kann etwa die Intention der Sprache dem Inhalt widersprechen, wie es bei Sarkasmus üblicherweise der Fall ist.

Bei der Implementierung ergeben sich dann Probleme, trotz der sprachlichen Besonderheiten konsistente Regelmäßigkeiten als Grundlage zur Automatisierung zu finden.

### **Linguistische Analyse**

Um einen Text möglichst Fehlerfrei verarbeiten zu können bedarf es also einem kleinschrittigen und präzisen Verfahren. Dieses bezeichnet man als linguistische Analyse.

Sie ist aufgeteilt in vier wichtige Schritte, welche aufeinander aufbauen und an Komplexität zunehmen. Im folgenden sind diese Schritte aufgeführt und kurz erläutert [RS18]:

1. Morphologische Analyse - Die Zerlegung von Wörtern bezüglich ihrer Struktur. Die Komposition aus Präfix, Wortstamm und Suffix von Wörtern wird erkannt, um Fall, Tempus, Numerus etc. des Wortes zu bestimmen. Im Englischen zeigt so die Endung -ed bei Verben die Vergangenheitsform an. Dabei ist zu beachten, dass Mehrdeutigkeiten entstehen können.
2. Syntax - Aufbau von Sätzen durch einzelne Wörter. Jede Sprache besitzt syntaktische Regelungen bezüglich der auftauchenden Wörter; im Deutschen folgt etwa auf einen Artikel irgendwann ein Substantiv oder bestimmte Signalwörter geben Satztyp und Tempus an. Auf Basis dieses Wissens können formale und strukturelle Analysen der Textbestandteile erfolgen.
3. Semantiken - Die Identifikation der Bedeutung von einzelnen Sätzen und Wörtern. Die Semantik eines Textabschnittes wird häufig als „Lo-

gik“ bezeichnet und fragt nach dessen Bedeutung bzw. Thema. Semantische Deutungen können anhand von Kompositionen einzelner Wörter und Sätze auf Basis der semantischen Analyse erkannt werden.

4. Kontext - Darstellung von satzübergreifenden Zusammenhängen syntaktischer und semantischer Natur. Mehrere Sätze können über Konstrukte wie etwa Pronomen verbunden sein, wenn sich das Pronomen des einen Satzes auf ein Subjekt des anderen bezieht.

Diese Aufteilung lässt sich jedoch in noch weitere Teilschritte für die Spracherkennung und -generierung als NLP-Architektur unterteilen. Dies ist in [RS18] genauer erläutert.

## 2.2 Ontologien

Dieser Abschnitt befasst sich genauer mit dem Begriff der Ontologien im Kontext der Informatik.

### 2.2.1 Definition

Ontologien lassen sich allgemein definieren als „formale, schematische Abbildungen eines Wissensbereichs, bestehend aus einem Vokabular und Regeln zu seiner Zusammensetzung“[WE13].

Zunächst bedeutet dies, dass eine Ontologie stets auf nur Wissen aus einer bestimmten Domäne zusammenfasst und nicht versucht das gesamte bzw. sehr allgemeines Wissen abzubilden.

Des weiteren Besteht eine Unterteilung in ein Vokabular und in eine Menge von Regeln.

Einer der Aspekte von Ontologien, welche sie für die Informatik so geeignet machen ist, dass sie in maschinenlesbaren Ontologiesprachen verfasst werden. Somit können Computersysteme diese automatisch auslesen und ggf. interpretieren und daraus schlussfolgern können.

Da Terminologie und bestimmte Formalitäten zwischen Ontologiesprachen variieren können, wird im Folgenden der Bezug auf die weit verbreitete Ontologiesprache „Web Ontology Language“ (OWL) angenommen.

### Aufbau

Die Grundlage für die Struktur einer Ontologie stellt der hierarchische Aufbau von Begriffen (*concepts*). Diese Repräsentieren eine Art Klassenstruktur und werden daher oft auch als Klassen bezeichnet. Sie können in aus der Informatik bekannten Ansätzen wie Ober- und Unterklassen angeordnet werden. Begriffe sind üblicherweise sehr generell gehalten und könnten etwa die Klasse *Auto* abbilden.

In vielen Fällen lässt sich ein Begriff in verschiedene Typen aufteilen. Ein Auto kann beispielsweise nach verschiedenen Automarken typisiert werden. Diese Typen werden dann ebenfalls durch Klassen verkörpert und bilden in dem hierarchischen Aufbau eine Unterklasse des jeweiligen Begriffs.

Die sogenannten Instanzen (*instances*) verkörpern konkrete Objekte dieser Klassen. Ein Instanz wäre dann etwa ein bestimmtes Modell, einer bestimmten Marke von Autos.

Diese Hierarchie ist ggf. beliebig erweiterbar. So könnte man für eine bessere Klassifizierung z.B. die Marke eines Autos zunächst in eine Produktreihe oder nach einer Produkteigenschaft wie *Combi* typisieren und erst davon abhängig die verschiedenen Instanzen zuordnen.

Relationen beschreiben Beziehungen zwischen Begriffen und werden oft auch als Eigenschaften verwendet und bezeichnet. Ähnlich zu vielen Programmiersprachen können diese Relationen und Eigenschaften eines Begriffs an die Instanzen des Begriffs vererbt werden. Grundsätzlich ist dabei auch eine Mehrfachvererbung möglich.

## 2.3 Verwandte Arbeiten

# Kapitel 3

## Betriebliches Umfeld - Hella Use-Case

Im betrieblichen Umfeld liegen zu Beginn jeden Entwicklungsprojektes für neue Produkte die Aufgaben und Ziele für die Entwicklung als Dokumente vor. Forschungsergebnisse finden Anwendung in der Vorentwicklungsphase, in der die Eignung der Erkenntnisse für neue Produkte eines Unternehmens evaluiert wird. Die Produktentwicklung unterliegt dabei bestimmten Kriterien und Faktoren, die den unternehmerischen Erfolg beeinflussen. Neben betriebswirtschaftlichen Einflüssen wie der Einordnung des Produktes in der Wertschöpfungskette sind es dabei besonders technische Anforderungen an das Produkt, die definiert und während der Produktentwicklung eingehalten werden müssen. Verschiedenste Akteure aus einem Unternehmen sind dabei an der Festlegung der Anforderungen an ein Entwicklungsprojekt bzw. Produkt beteiligt.

In der Automobilindustrie betrifft dieser Ablauf zumeist die Entwicklung neuer Fahrzeugkomponenten, heutzutage meist elektronische und mechanische Bausteine. Diese Bausteine werden dabei nicht sämtlich vom Fahrzeughersteller (OEM) selbst, sondern durch eine Vielzahl von Zulieferern produziert und entwickelt. Die Produktspezifikationen liegen meist digital als Texte, Tabellen und Grafiken vor und werden an den Zulieferer übermittelt. Nach dem Entwicklungsprozess steht dann die (Serien-)entwicklung und -fertigung des Produktes für das Ausrollen in großen Stückzahlen an den Hersteller, der das zugelieferte Produkt dann in seinen Produkten verwendet. Um dies zu erreichen, müssen während des gesamten Prozesses die Anforderungen, die das Systemumfeld des Fahrzeugherstellers hat, berücksichtigt und

eingehalten werden.

Die Anforderungen an das Produkt, etwa technische Rahmenbedingungen, werden dabei von vielen verschiedenen Domänenexperten beim OEM formuliert und in das sogenannte Pflichtenheft für die Entwicklung eingetragen. Beteiligte sind etwa Produktdesigner, Ingenieure und Systemtechniker, die an verschiedenen Stellen im Lastenheft Anforderungen an eine Komponente festlegen. Diese Beteiligten sind in der Regel auf ihren Bereich spezialisiert und nicht interdisziplinär, zudem gibt es sprachliche Eigenheiten der Autoren und unternehmensinterne Richtlinien für die Formulierung, die das Verständnis erschweren können. Demzufolge sammeln sich im Lastenheft verschiedenste Merkmale einer Komponente, die aber nicht im Bezug zueinander stehen und sich im schlimmsten Fall gegenseitig ausschließen.

Durch diese fachliche Breite und Tiefe der Spezifikationen im Pflichtenheft, aber auch durch den Umfang des Lastenheftes von mehreren tausend Seiten, kommt es häufig insbesondere zu Verständnisproblemen auf Seite des Zulieferers. Die Gewichtung einzelner Anforderungen in einem größeren Systemkontext fällt dort schwer, da nun Projektteammitglieder, die an der Entstehung des Lastenheftes nicht beteiligt waren, dieses verstehen und ein Produkt entwickeln sollen, dass möglichst alle Anforderungen berücksichtigt. In Texten muss also nach Zusammenhängen und Bezügen zwischen mehreren Anforderungen gesucht werden, damit die Korrektheit des späteren Produktes gewährleistet ist.

Die Analyse von Zusammenhängen zwischen Anforderungen stellt dabei aus Gründen der Effizienz ein Problem dar, wenn jeder Beteiligte von Hand die für ihn relevanten Anforderungen aus dem Lastenheft extrahieren muss. Auch müssen die Lastenhefte an die Formulierungen und Ausdrucksweisen für Requirements-Management im Unternehmen angepasst werden. Bislang gibt es jedoch kaum Werkzeugunterstützung, die effiziente Möglichkeiten zur automatisierten Überarbeitung und Anpassung einzelner Anforderungen aus dem Dokument bietet. Ansätze aus dem *Natural-Language-Processing* (NLP) stellen gleichzeitig vielversprechende Forschungsfelder in der Informatik dar, die eine solche automatisierte Verarbeitung auf Basis von Sprachanalyse ermöglichen. Syntax und Semantik der einzelnen Sätze und Zusammenhänge in Texten können auf Basis aktueller Trends wie Machine-Learning und dynamischer Programmierung zunehmend besser abgebildet werden.

### **3.1 Re-Prozesse bei Hella und allgemein in Firmen**

### **3.2 Betriebliche Anforderungen**

### **3.3 Ansatz und Konzept unserer Werkzeuge**



# Kapitel 4

## R2B-Converter

- 4.1 Architektur Klassen und Verteilung der Ressourcen
- 4.2 Implementierung (bisschen Code, GUI, Listenarchitektur, Workflow für User)
- 4.3 mögliche Erweiterungen
- 4.4 Test
  - 4.4.1 Methodik
  - 4.4.2 Durchführung
  - 4.4.3 Ergebnisse

# Kapitel 5

## Delta-Analyse

### 5.1 Architektur

### 5.2 Implementierung

### 5.3 mögliche Erweiterungen

### 5.4 Test

#### 5.4.1 Methodik

#### 5.4.2 Durchführung

#### 5.4.3 Ergebnisse

# Kapitel 6

## Evaluation

6.1 Auswertung der Testresultate

6.2 Mehrwert?

6.3 Ziel erreicht? Hypothese reviewen und schwafeln

# Kapitel 7

## Fazit

### 7.1 Zusammenfassung

### 7.2 Ausblick

# Literaturverzeichnis

- [AWK06] Christian Allmann, Lydia Winkler, Thorsten Kölzow. „The requirements engineering gap in the OEM-supplier relationship.“, Journal of Universal Knowledge Management 1.2 (2006): 103-111.
- [BAL10] Helmut Balzert. „Lehrbuch der softwaretechnik: Basiskonzepte und requirements engineering.“, Springer-Verlag, 2010.
- [CHO57] Noam Chomsky, „Syntactic Structures“ , Mouton & Co., Feb. 1957.
- [COP04] Ann Copestake, University Of Cambridge, „Natural Language Processing“, 2004.
- [DGE19] Deutsche Gesellschaft für EMV-Technologie e.V. „CRS Customer Requirements Specification“, aus [https://www.demvt.de/publish/view-full.cfm?ObjectID=ba9a0878\\_e081\\_515d\\_74a3411df6771be8](https://www.demvt.de/publish/view-full.cfm?ObjectID=ba9a0878_e081_515d_74a3411df6771be8), abgerufen am 09.01.19.
- [HE19] Firma HELLA. „Unternehmensinformationen in Kürze.“, aus <https://www.hella.com/hella-com/de/HELLA-im-Ueberblick-723.html>, abgerufen am 09.01.19.
- [HAO14] Hao Wu, et al. „ILLINOISCLOUDNLP: Text Analytics Services in the Cloud.“ LREC. 2014.
- [HL01] Hubert F. Hofmann, Franz Lehner. „Requirements engineering as a success factor in software projects.“IEEE software 4 (2001): 58-66.
- [LV10] F. Langenscheidt, B. Venohr. „Lexikon der deutschen Weltmarktführer: Die Königsklasse deutscher Unternehmen in Wort und Bild.“, Deutsche Standards–Gabal Verlag Google Scholar (2010).

- [MW02] Matthias Weber, Joachim Weisbrod. „Requirements engineering in automotive development-experiences and challenges.“, Requirements Engineering, 2002. Proceedings. IEEE Joint International Conference on. IEEE, 2002.
- [RS18] Felix Ritter, Aaron Schul. „Analyse aktueller NLP-Methoden und -Werkzeuge am Beispiel von GATE, Projektarbeit Fachhochschule Dortmund 2018 “
- [WE13] Weller, Katrin. Öntologien.”(2013): 207-218.