Institut für Rechnergestützte Ingenieursysteme

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit Nr. 254

Data Mining und Domain Ontology in der Produktentwicklung

Melek Bugdayci

Studiengang: Informatik

Prüfer/in: Univ.-Prof. Hon.-Prof. Dr. Dieter Roller

Betreuer/in: Dipl.-Inf. Akram Chamakh

Beginn am: 14. September 2015

Beendet am: 15. März 2016

CR-Nummer: H.2.8, I.2.4, I.2.7

Kurzfassung

Die Entwicklung von Produkten fordert von der Planung bis hin zur Herstellung Informationen um die Prozesse verwirklichen zu können. Das Wissen ist somit die wichtigste Ressource bei der Produktentwicklung und an diese zu gelangen eine grundlegende Aufgabe. Die Wissensrecherche kann allerdings lange Zeit in Anspruch nehmen, da Informationen verteilt auf textuelle Daten sind, die strukturiert oder unstrukturiert vorliegen. Diese Texte müssen analysiert und der Inhalt auf Relevanz geprüft werden. Jedoch ist die Zeit ein wichtiger Faktor bei der Wettbewerbsfähigkeit.

Um dieser Problematik entgegenzukommen wurde in dieser Arbeit eine Software entwickelt, die Texte mithilfe von Ontologien kategorisiert. Die Klassifizierung erfolgt mit Unterstützung von Machine Learning Verfahren. Durch Vorverarbeitung der Texte können bessere Ergebnisse der Klassifizierung erzielt werden. Weiterhin verhilft die Ontologie zur Erfassung der inhaltlichen Bedeutung der Texte. Sie dient als Repräsentation von Informationen auf semantischer Ebene. Dabei wurde eine domänenspezifische Basisontologie aufgebaut und integriert. Diese wird erweitert durch die Begriffe, die in der zu klassifizierenden Datei vorkommen.

Diese Arbeit soll dazu beitragen, bereits existierende und ähnliche Lösungen oder Lösungsideen für eine Problemstellung in der Produktentwicklung aufzufinden. Desweiteren wird recherchiert, inwiefern die Ontologie als Hilfe dienen kann.

Abstract

The development of products requires information for the realization from the planning up to manufacturing. So knowledge is the most important resource of product development and it is a fundamental task to earn it. However, research of knowledge can be a lot of time taking process because information is distributed among textual data which is structured or unstructured. These texts need to be analysed and the relevance of the content need to be examined. Nonetheless time is an important factor for competitiveness.

Concerning to this problem a software has been developed which categorizes texts with the aid of ontologies. The classification happens through the support of machine learning procedures. By applying certain preprocesses it is possible to get better results of the classification. Furthermore the ontology helps to understand the content of texts. On semantic level ontologies are a representation of information. Thereby a domain-specific basic ontology has been built which is expanded by the terms occuring in the text being classified.

The aim of this thesis is to find existing or similar solutions or solution ideas for a task in product development. Moreover it is investigated how the ontology can help refering to this.

Inhaltsverzeichnis

1 Einleitung **9**
1.1 Problemstellung . . . 9
1.2 Gliederung . . . 9

2 Lösungssuche in der Produktentwicklung **11**
2.1 Vorgehensmodelle . . . 11
2.1.1 Vorgehensmodelle für die Produktentwicklung . . . 11
2.1.2 Vorgehensmodelle für die Problemlösung . . . 12
2.1.3 Systemdenken . . . 14
2.1.4 Konstruktionsmodell . . . 15
2.2 Lösungssammlungen . . . 16
2.2.1 Sammlungen auf Wirkmodellebene . . . 16
2.2.2 Sammlungen auf Baumodellebene . . . 17
2.3 Probleme bei der Lösungssuche . . . 17

3 Ontologien und Semantik **21**
3.1 Informations- und Wissenmanagement . . . 22
3.2 Information Retrieval . . . 23
3.3 Semantische Technologien und Ontologien . . . 25
3.3.1 Das semiotische Dreieck . . . 25
3.3.2 Die semantische Treppe . . . 26
3.4 Einsatz von Ontologien . . . 27
3.4.1 Wissensrepräsentation in der Produktentwicklung . . . 28

4 Entwicklung einer Ontologie **31**
4.1 Vorgehensmodelle zur Entwicklung von Ontologien . . . 31
4.1.1 Vorgehensmodell zur Entwicklung von Ontologien in der Produktentwicklung 32
4.2 Basisontologie . . . 33
4.3 Erweiterung der Ontologie . . . 34

5 Related Works **37**
5.1 Text Categorization Using Automatically Acquired Domain Ontology . . . 37
5.2 Research on Domain Ontology Semi-automatic Construction Model towards Chinese Text . . . 40

6 Implementierung **45**
6.1 Verwendete Technologien . . . 45
6.1.1 Stanford CoreNLP . . . 45

6.1.2 WEKA . . . 45
6.1.3 BabelNet . . . 46
6.2 Vorgehensweise . . . 46
6.2.1 Ausgangslage . . . 46
6.2.2 Lösungsansatz . . . 47
6.3 Technische Umsetzung . . . 49
6.3.1 Oberfläche der Software . . . 49
6.3.2 Datenstruktur . . . 52
6.4 Evaluation . . . 54
6.4.1 Vorgehen . . . 54
6.4.2 Ergebnisse . . . 54
6.4.3 Erkenntnisse . . . 55

7 Zusammenfassung und Ausblick 57
7.1 Zusammenfassung . . . 57
7.2 Ausblick . . . 57

Literaturverzeichnis 59

Abbildungsverzeichnis

1.1 Gliederung der Bachelorarbeit 10

2.1 Vorgehensmodell für die Produktentwicklung [2] 12
2.2 Das Münchener Vorgehensmodell [6] 13
2.3 Problemabstraktion bei der Lösungssuche [3] 14
2.4 Der Modellraum des Konstruierens [9] 15
2.5 Informationsspeicher und Zugriffslogik einer Lösungssammlung 16
2.6 Probleme bei der Lösungssuche 18

3.1 Abgrenzung und Beziehung der Begriffe Zeichen, Daten, Information und Wissen[2] 22
3.2 Vorgehen beim Information Retrieval 23
3.3 Grafische Darstellung von Recall und Precision 24
3.4 Semantische Treppe [20] 26
3.5 Ebenen der Wissensrepräsentation im Semantic Web [27] 28

4.1 Vorgehensmodell für die Entwicklung einer Ontologie [32] 32
4.2 Vorgehensmodell für die Entwicklung einer Ontologie in der Produktentwicklung [30] 33
4.3 Basisontologie der Automobil-Domäne 34
4.4 Erweiterung der Ontologie durch Extraktion der Instanzen aus Textdokumenten [2] . 35

5.1 Ontologie-Struktur des InfoMap [35] 38
5.2 Framework des semi-automatischen Konstruktionsmodells für die Domain Ontologie[37] 41

6.1 Programablauf 47
6.2 Aufbau der ARFF-Datei 48
6.3 BabelNet Anfrage 49
6.4 Tab PreProcess 50
6.5 Tab Train Classifier 51
6.6 Tab Classify Requirements 51
6.7 Klassendiagramm 53

1 Einleitung

Das Niederschreiben und Dokumentieren ist notwendig für das Festhalten von Wissen. Somit können Informationen behalten oder auch weitergegeben werden, sodass andere davon Nutzen ziehen können. Das World Wide Web ist mittlerweile überfüllt von Dokumentationen verschiedenster Themen. Außerdem gibt es klassische Bibliotheken zunehmend in elektrischer Form. Somit steigt leider auch die Anzahl der unstrukturierten Daten und Ambiguitäten, welche die Informationssuche mühsam werden lassen[1].

1.1 Problemstellung

Die Wettbewerbsfähigkeit eines Unternehmens hängt an erster Stelle von den Produkten beziehungsweise von den Dienstleistungen ab. Dabei handelt es sich um die Schnelligkeit bei der Herstellung und die Qualität der Produkte. Hierfür sind wiederum Informationen notwendig.

Im Produktentwicklungsprozess ist der Austausch von Informationen ein wesentlicher Bestandteil und Erfolgsfaktor[2]. Vom großen Informationsangebot sollen relevante Informationen gefiltert werden, wobei man an Schwierigkeiten stößt. Vor allem auch deshalb, weil das Informationsangebot ständig wächst und es unstrukturiert vorliegt. Zudem werden verschiedene Begriffe bei der Beschreibung der Prozesse verwendet, sodass die Lösungssuche sehr mühevoll wird. Somit kostet der Ablauf Zeit und Geld. Dabei wird die Ontologie als Wissensrepräsentation zu Hilfe genommen, um existierende und relevante Lösungen zu finden.

1.2 Gliederung

In dieser Bachelorarbeit soll recherchiert werden, wie die Domain Ontologie in der Produktentwicklung als Lösungshilfe dienen kann. Dabei werden in den ersten Kapiteln die Grundlagen der Produktentwicklung und der Ontologie vorgestellt. Nachdem ein Überblick über die Grundladen verschaffen wurde, wird der Lösungsansatz bekannt gemacht. Dazu wird eine Basisontologie aufgebaut. Außerdem werden einige relevante Arbeiten präsentiert, um weitere Lösungsideen kennenzulernen. In den letzten Kapiteln wird letztendlich die Implementierung und die Ergebnisse vorgestellt. Zuletzt wird die Arbeit zusammengefasst und Verbesserungsvorschläge gemacht. Die genaue Gliederung ist wie folgt aufgebaut:

Kapitel 2 stellt zunächst den allgemeinen Ablauf einer Lösungssuche in der Produktentwicklung dar. Es werden die Vorgehensmodelle und die Klassifizierung und Strukturierung von Funktionen vorgestellt. Daraufhin wird auf die Problematik bei der Lösungssuche eingegangen.

Als Antwort auf das Problem wird in Kapitel 3 und 4 die Ontologie vorgestellt, wobei **Kapitel 3** die Grundlagen zur Ontologie vorstellt und **Kapitel 4** die Entwicklung einer Ontologie dargestellt. Es wird auf die verschiedenen Prozesse bei der Entwicklung und auf die Erweiterung einer Ontologie eingegangen.

Kapitel 5 stellt ähnliche Beiträge vor. Zum Einen wird *Text Categorization Using Automatically Acquired Domain Ontology* und zum Anderen *Research on Domain Ontology Semi-automatic Construction Model towards Chinese Text* vorgestellt.

In **Kapitel 6** wird der praktische Teil der Arbeit und die Ergebnisse der Implementierung vorgestellt.

Kapitel 7 fasst diese Arbeit zusammen und gibt einen Ausblick bezüglich der Domain Ontolgy in der Produktentwicklung.

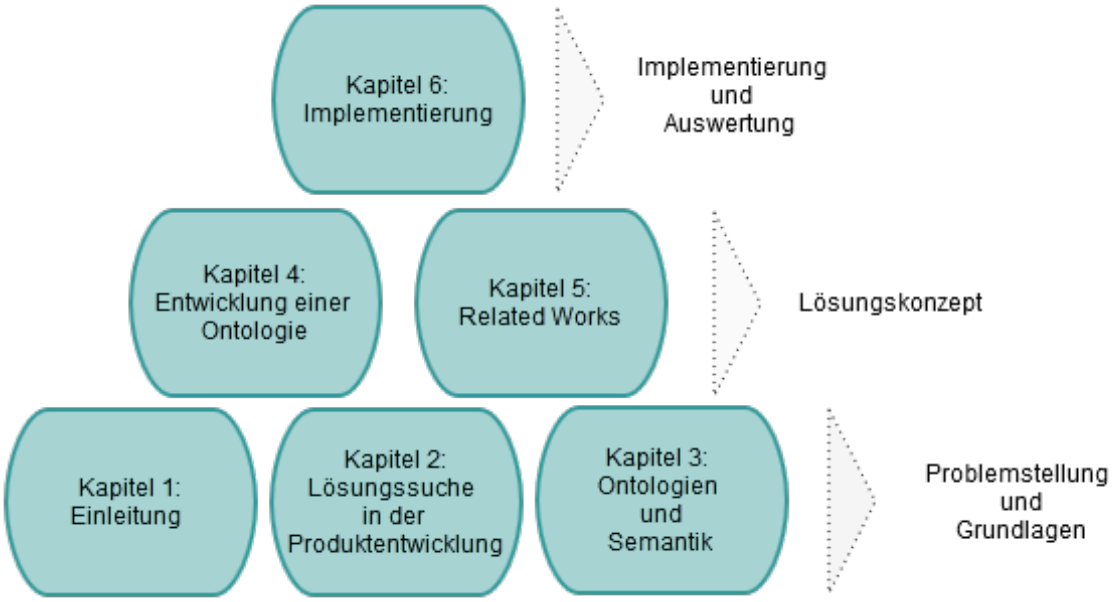

Abbildung 1.1: Gliederung der Bachelorarbeit

2 Lösungssuche in der Produktentwicklung

Die Produktentwicklung kann für kleine Produkte einfach, für andere große jedoch sehr umfangreich sein. Es ist ein systematisches und methodisches Vorgehen notwendig. Bei größeren Produkten sind außerdem mehrere Bearbeiter notwendig, die koordiniert werden müssen. Damit diese den Entwicklungsprozess überschauen können, wird der Vorgang in logisch abgrenzbare Abschnitte unterteilt. Vorgehensmodelle sind ein Hilfsmittel, mit der man Prozesse planen kann. Der Anwender erkennt, welchen Schritt er ausführt und welche folgen werden.

2.1 Vorgehensmodelle

Um die Prozesse der Produktentwicklung kennenzulernen werden in diesem Abschnitt verschiedene Vorgehensmodelle vorgestellt. Dabei wird zunächst ein Modell der Makro-Ebene eingeführt um eine grobe Vorstellung zu bekommen. Danach wird zur Spezifizierung ein Modell der Mikro-Ebene vorgestellt.. Zum ersten gehört das Modell der Produktentwicklung. Zur Mikro-Ebene zählen die Modelle der Problemlösung[2, 3]. Produktmodelle sind wesentliche Dokumentations- und Kommunikationsmittel. Sie vermitteln sowohl den Informationsraum, als auch die Zugriffsmöglichkeiten auf Lösungen.

2.1.1 Vorgehensmodelle für die Produktentwicklung

Die Produktentwicklung bietet Vorgehensmodelle um Aufgaben und Probleme zu lösen. Die Vorgehensschritte können wiederum in Phasen gruppiert werden, die aufeinander aufbauen, jedoch sich auch überlappen[5]. Zu diesen Phasen gehört das Planen, Konzipieren, Entwerfen und Ausarbeiten. In der **Planungsphase** werden die Anforderungen und Ziele festgelegt. Die **Phase des Konzipierens** schließt die Festlegung von Funktionen und die Lösungssuche ein. Die Aufgabe wird in Einzelfunktionen zerlegt, sodass sie jeweils als Teilaufgaben bearbeitet werden kann. Die funktionale Beschreibung ist dann der Ausgangspunkt für die Lösungssuche. Die fundamentalen Lösungen werden daraufhin in der **Entwurfs-** und **Ausarbeitungsphase** konkretisiert, indem das System in Module gegliedert, gestaltet und ausgearbeitet wird.

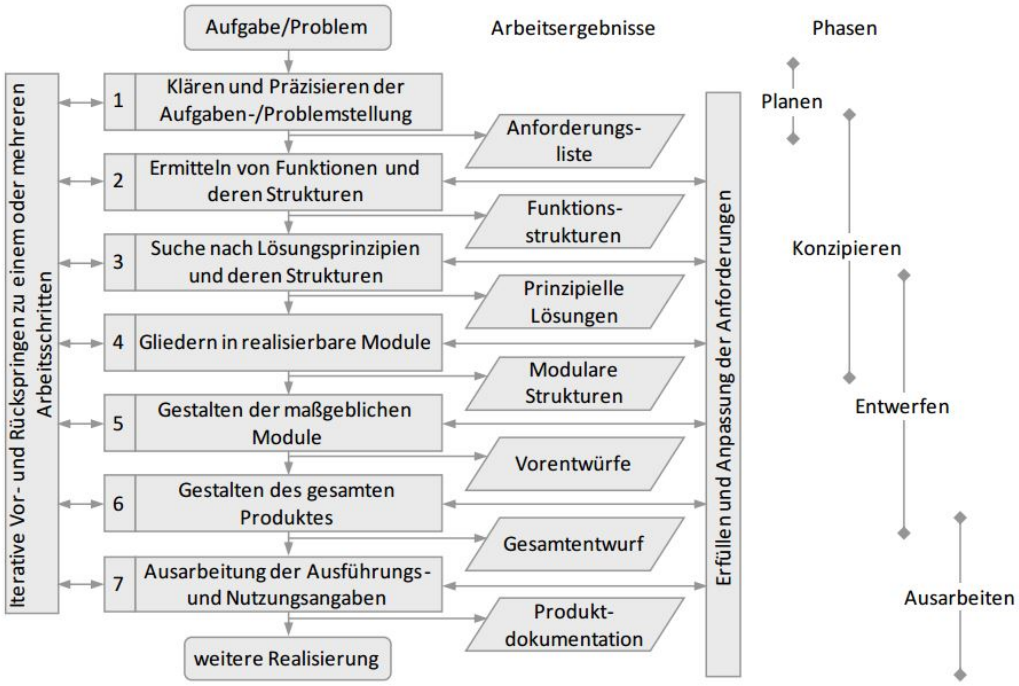

Abbildung 2.1: Vorgehensmodell für die Produktentwicklung [2]

Wie auch auf der Abbildung 2.1 zu erkennen ist, wird in dem Vorgehensmodell vom Abstrakten zum Konkreten vorangearbeitet[6]. Es ist auch zu sehen, dass das schrittweiße Vorgehen kein sequentielles Vorgehen ist, sondern es auch Überlappungen zwischen den Phasen gibt. Außerdem sind Vor-und Rücksprünge nicht zu vermeiden und es ist eine iterative Gestaltung notwendig.[2]

2.1.2 Vorgehensmodelle für die Problemlösung

Ein Problem liegt dann vor, wenn entweder das Ziel nicht eindeutig definiert ist oder die Mittel zur Lösung nicht verfügbar oder unbekannt sind [6]. Ferner gibt es drei Hauptpunkte zu verfolgen, die beim Überwältigen eines Problem als Leitfaden dienen[2].

1. **Zielermittlung**- das Problem wird analysiert und das Ziel konkretisiert
2. **Lösungssuche**- bei der Suche nach Lösungen werden nach bereits existierende Lösungen gesucht oder neue Lösungen entworfen
3. **Entschluss**- die gefundenen Lösungen werden bewertet und die geeignete Lösung ausgewählt

Die Vorgehensmodelle in der Problemlösung orientieren sich an diesen drei Punkten. Zudem können sie in weitere Einzelschritte aufgeteilt werden. Beispielhaft wird in dieser Arbeit das Münchener Vorgehensmodell vorgestellt. Diese enthält sieben Elemente, die auf Abbildung 2.2 zu sehen sind[3].

Auch in diesem Modell ist die Iteration ausschlaggebend. Das Münchener Vorgehensmodell ist aufgebaut aus sich überschneidenden Kreisen, die auf einen iterativen Durchlauf verweisen. Das

Standartvorgehen hat jedoch folgende Sequenz: Ziel planen, Ziel analysieren, Problem strukturieren, Lösungsideen ermitteln, Eigenschaften ermitteln, Entscheidungen herbeiführen, Zielerreichung absichern.

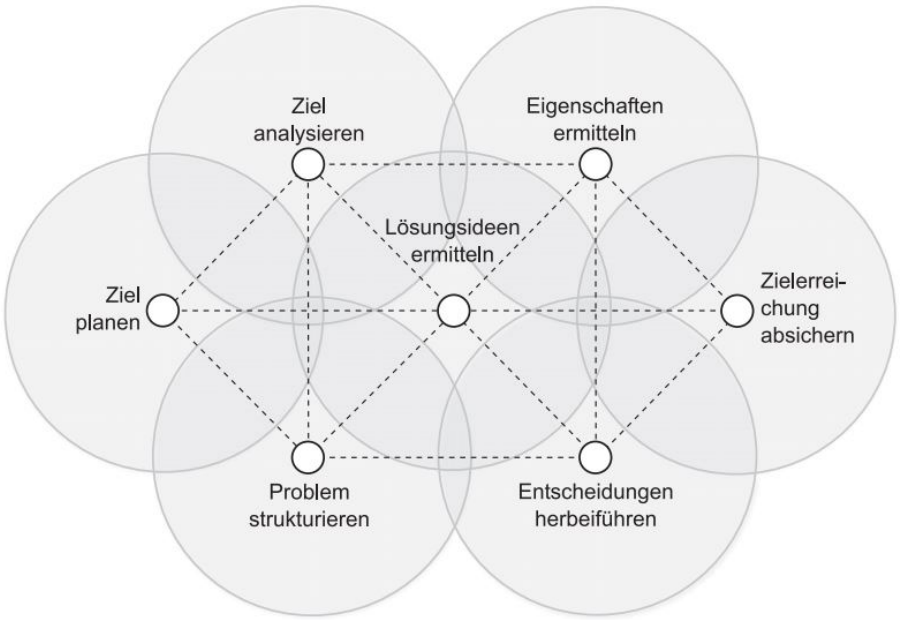

Abbildung 2.2: Das Münchener Vorgehensmodell [6]

Im Gegensatz zu linearen Darstellungen sind in dem dargestellten Netzwerk Rücksprünge erlaubt, die in Situationen in der Produktentwicklung benötigt werden, mit der nicht gerechnet wird oder weitergearbeitet werden kann. Zum Bespiel kann in der Entscheidungsebene ein Rücksprung erzeugt werden, wenn die gefundenen Lösungen alle nicht gut geeignet sind. In diesem Fall kann in der Ebene der Lösungssuche nach weiteren Lösungen gesucht werden.

Bei der Problemlösung ist die Problemabstraktion wichtig. Vor allem in der Ebene der Lösungssuche ist sie von schwerwiegender Bedeutung. Die Suche der bereits existierenden Lösungen kann vereinfacht werden, indem das Problem zuvor auf einer abstrakten Ebene beschrieben wird. Danach ist die Analyse und Anwendung der Lösung möglich.

2.1.3 Systemdenken

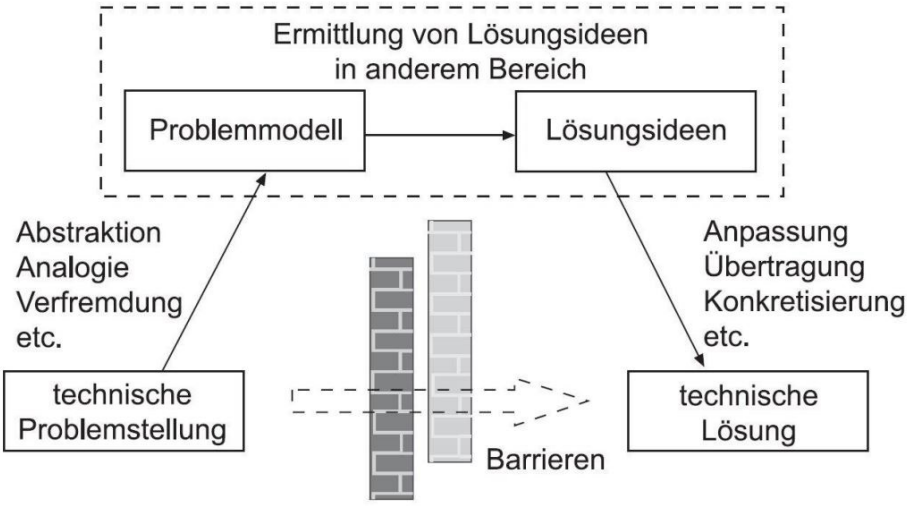

Abbildung 2.3: Problemabstraktion bei der Lösungssuche [3]

Technische Systeme sind neben den Vorgehensmodellen als Teil der Lösungssuche in der Produktentwicklung zu betrachten. Systeme unterscheiden sich unter drei Arten. Diese Arten sind Sach-, Handlungs- und Zielsysteme. Sachsysteme sind technische Gebilde wie zum Beispiel Maschinen. Sie werden auch als technisches System bezeichnet und stellen das eigentliche Objekt der Systemtechnik dar. Sachsysteme sind charakterisiert durch ihre Zweckorientierung und ihrer Strukturierung. Für die Entwicklung und Gestaltung von Sachsystemen sind Handylungssysteme zuständig. Dabei geht es um die Organisation der technischen Arbeit wie beispielsweise Unternehmen. Somit ist das Handlungssystem das Subjekt der Systemtechnik. Mit den Vorgaben des Handlungssystems und der Umgebung versucht der Ingenieur oder Entwickler seine Handlungen entsprechend dem Zielsystem auszurichten.

Das Systemdenken bietet die Darstellung von technischen Systemen. Eine verbreitete Ansichtsweise im Systemdenken ist die Betrachtung des Systems als Black Box[7]. Dabei wird ein Input zu einem Output überführt. Bei der Lösungssuche werden die technischen Systeme gesucht, die auf einen technischen Prozess so einwirken, dass die Überführung des Inputs in den Output ermöglicht wird[8]. Auch hier ist die Abstraktion von Vorteil, da die technische Problemstellung modelliert und somit in weiteren Bereichen nach Lösungen gesucht werden kann. Daraufhin kann die gefundene Lösung konkretisiert und auf den speziellen Fall angewendet werden (siehe Abbildung 2.3).

2.1.4 Konstruktionsmodell

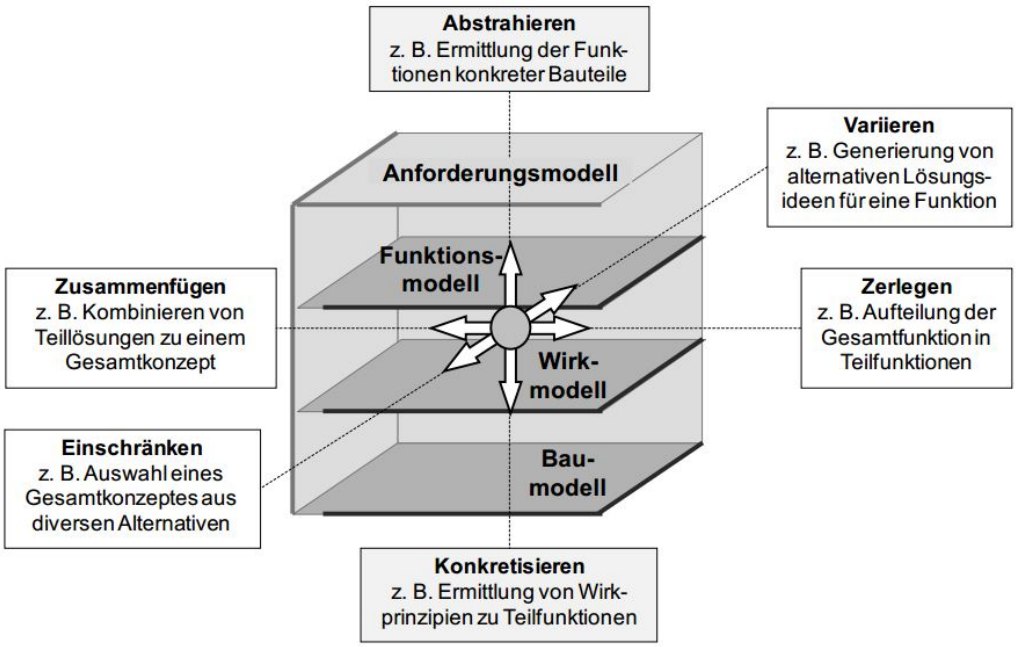

Abbildung 2.4: Der Modellraum des Konstruierens [9]

Für die Entwicklung eines Produktes vom Abstrakten zum Konkreten existieren ebenfalls Modelle, die zu den Konstruktionsmodellen zählen. Diese Beschreiben auf verschiedenen Konkretisierungsstufen das System. Hierzu wird ein Modell vorgestellt, das im bereits vorgestellten Münchener Vorgehensmodell eingeordnet wird. Das Modell besteht aus drei Ebenen der Produktkonkretisierung: Funktionsebene, Wirkmodellebene und Baumodellebene. Es ist möglich gegensätzlich in die Richtungen Abstrahieren vs. Konkretisieren, Zerlegen vs. Zusammenfügen und Einschränken vs. Varieren der Lösungsmenge zu navigieren. Das **Anforderungsmodell**, was auf der Abbildung 2.4 zu sehen ist, ist die Hüllkurve des Würfels, weil Anforderungen im ganzen Verlauf der Produktkonkretisierung wichtig sind und sich auf alle Ebenen beziehen. Auf der obersten Ebene, die **Funktionsmodellebene**, wird das Produkt als Funktionsmodell dargestellt. Die mittlere Ebene, die **Wirkmodellebene**, skizziert das System in seinen Wirkprinzipien und Wirkstrukturen und gibt somit die prinzipielle Lösung eines Problems wieder. Sie enthält die physikalischen Effekte in Verbindung mit geometrischen und stofflichen Merkmalen, die das Prinzip der Lösung zeigen, weshalb die Lösungen auf dieser Ebene als Prinziplösungen bezeichnet werden. Die **Baumodellebene** fasst das Produkt in seiner räumlichen Ausgestaltung. Sie repräsentiert die Bauelemente und die Montage an ein Produkt. Die Zusammenstellung der einzelnen Bauteile werden durch die Baustruktur festgelegt. Die Baustruktur wiederum ist gemeinsam mit der Definition der Schnittstellen und Form der Bauteile im Baukonzept definiert.[9]

2.2 Lösungssammlungen

Bei der Suche nach Lösungen existieren Sammlungen, die als Hilfsmittel dienen. Einige davon sollen hier beispielhaft vorgestellt werden um bei der Lösungssuche in der Produktentwicklung zu helfen. Ein Sammlung kann eine einfache Liste oder auch ein komplexes Rechnerwerkzeug sein. Sie ermöglichen den Zugriff auf bestehende Lösungen und verfügen notwendigerweise einen Informationsspeicher und eine Zugriffslogik (siehe Abbildung 2.5). Der Speicher ist charakterisiert durch die Lösungsanzahl, die inhaltliche Beschreibung, die Struktur und Ordnung und den Konkretisierungsgrad der Lösung. Die Zugriffslogik definiert verschiedene Zugriffsmechanismen auf die Lösung und berücksichtigt die Abstraktion oder begriffliche Aspekte.

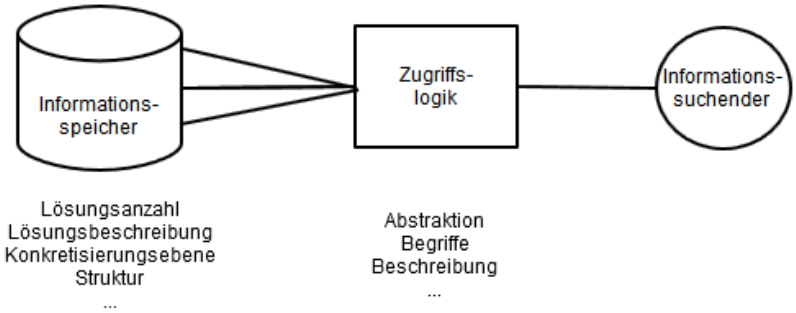

Abbildung 2.5: Informationsspeicher und Zugriffslogik einer Lösungssammlung

Nachfolgend sollen einige Sammlungen zur Lösungssuche untersucht und danach strukturiert und geordnet werden. Dabei handelt sich um Konstruktionskataloge und Lösungssammlungen auf verschiedenen Konkretisierungsebenen, Patentsammlungen, geometrieorientierte Bausteine und Elemente wie Normteile, CAD-Features und allgemeine Vertriebsmedien, besonders Lösungen von Unternehmen im Internet, da dieser in den letzten Jahren verstärkt genutzt wird. Ausgehend vom Münchener Produktkonkretisierungsmodell ergeben sich grundlegende Kriterien zur Einordnung der Sammlungen. Zu diesen Kriterien zählt die Konkretisierungsebene, der Zerlegungsgrad und der Vielfalt. Die genannten Sammlungen sollen nun bezüglich der Lösungsinhalte und der Zugrifflogik in das Konkretisierungsmodell eingeordnet werden.

2.2.1 Sammlungen auf Wirkmodellebene

Auf Wirkmodellebene sind sowohl papierbasierte Sammlungen als auch Internetportale oder Datenbanken vorhanden. Da es bei der Produktentwicklung mehrere Bereiche gibt auf die zu achten ist, existieren in der Hinsicht verschiedene Quellen. Einer der Bereiche ist die Physik. Während die physikalische Effektesammlungein papierbasierter Informant ist, ist beispielsweise *www.creax.com* ein Internetportal, das den Zugang zu Informationen über phyisikalische und chemische Effekte gestattet

[9]. Vergleichsweise gibt es auch Sammlungen von biologischen Prinzipien, die für die Lösungssuche in der Produktentwicklung eingesetzt werden können [10].
Zudem enthalten Konstruktionskataloge Lösungen für spezifische Aufgaben gewisser Anwendungsgebiete oder Produktgruppen. Demgegenüber sind Lösungen für Verfahren in Patentsammlungen zu finden. *freepatentsonline.com* ist eine geeignete Seite dafür. Bei Patentsammlungen ist jedoch zu beachten, dass sie rechtlich geschützt sind und nur mit Einwilligung des Patentsinhabers benutzt werden dürfen.

2.2.2 Sammlungen auf Baumodellebene

Die Lösungssammlungen auf Baumodellebene sind meistens separate Elemente oder Bausteine, wie zum Beispiel Schrauben, Dichtungen etc. Auch auf dieser Ebene gibt es verschiedene Quellen. Hierzu zählen Unternehmen, die Produkte und Lösungen in Form von Katalogen, Messen oder über das Internet anbieten. Darin gibt es virtuelle Marktplätze, die in horizontale und vertikale Marktplätze unterschieden werden[11]. Vertikale Marktplätze stellen Produkte und Lösungen von zahlreichen Unternehmen zur Verfügung. Horizontale Marktplätze liefern genaue Produktbeschreibungen mit Kennwerten. Oft sind auch Anwendungsgebiete der Produkte angegeben, die zur Einschätzung für den Einsatz in der persönlichen Aufgabe verhelfen. Außerdem sind im CAD-Bereich Form-Features zu finden, die in unterschiedlichen Bereichen zu finden sind. Konstruktionsbezogene Features bezeichnen die funktionalen Attribute von Flächen. Fertigungsbezogene Features unterstützen die Automatisierung und Optimierung der Planung von Fertigungsprozessen und qualitätsbezogene Features beschreiben die Qualitätsmerkmale. Die Form-Features können Elemente auf Geometrie- und Technologiebasis eingesetzt werden. Für diesen Einsatz bieten sich CAD-Bibliotheken in Form von Katalogen.

2.3 Probleme bei der Lösungssuche

Es ist leider nicht immer möglich eine Antwort auf eine Aufgabe in der Produktentwicklung zu finden. Hin und wieder gibt es Hindernisse, die das schnelle oder effiziente Finden einer Lösung verhindern. Vor allem kommt es zu Problemen bei einem großen Lösungsfeld, wie das Internet. Die Probleme werden in zwei Kategorien unterteilt (siehe auch Abbildung 2.6) [2]:

1. **Fehlende Informationen** zu den Lösungen. Es wird zwar eine Lösung gefunden, jedoch enthält diese nicht genügende Informationen, ist nicht genug detailliert oder es ist kein vollständiger Zugriff gewährleistet.
2. **Ungeeignete Such- und Zugriffsmechanismen**. Das gefundene Ergebnis ist ungeeignet für die Aufgabe.

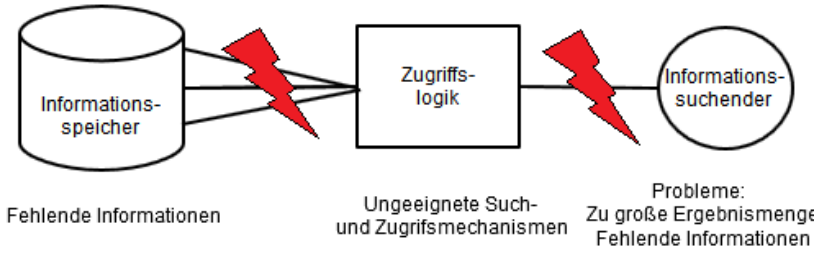

Abbildung 2.6: Probleme bei der Lösungssuche

Zu Gruppe 1: Oftmals ist die Behinderung für den Zugriff auf Informationen im Internet der Schutz des unternehmensinternen Know-Hows. Die Unternehemn geben jeweils unterschiedlich viele Informationen preis, denn sie wollen das Internet als Vertriebs- und Marketinginstrument nutzen. Ein weiteres Hinderniss ist die Verteilung der Informationen über Abteilungen oder sogar Unternehmen. Der effiziente und schnelle Zugriff aud die Information ist kaum möglich, weil es keinen zentralen Speicherort gibt. Dies führt unter anderem zu einem weiteren Problem. Da die Informationen nicht nur in einer Datenbank und damit in strukturierter Form vorliegt, kommt es zur Heterogenität der Daten. Größtenteils sind die Informationen unstrukturiert verteilt auf beispeilsweise Emails und weiteren Textdokumenten. Zumeist wird bei der Dokumentation in der Produktentwicklung auf die räumliche Anordnung, Baugruppen oder Anforderungen geachtet und es liegen keine abstrakten Beschreibungen vor. Somit ist die Lösung schwer zu erreichen und es kann keine Verbindung vom Abstrakten zu speziellen Anwendungsfällen gesetzt werden.

Zu Gruppe 2: Vor allem bei großen Datenbeständen spielen Such- und Zugriffsmechanismen eine wichtige Rolle. Dabei liefert eine wortbasierte oder begriffsorientierte Suche nur mühsam gewünschte Informationen, wenn falsche Begriffe für die Suche eingesetzt werden. Hierzu kann es kommen, wenn der Suchende nicht oder nicht genau weiß, was er sucht. Dann kann es zu einem Abbbruch der Suche kommen. Um die Gründe für einen Abbruch einer Lösungssuche besser zu identifizieren wurde eine Studie mit 41 befragten durchgeführt. Dabei haben sich folgende Ergebnisse zeigen lassen: lange Antwortzeit des Systems wie auch der hoher Zeitaufwand und lange Suchdauer, um den gewünschten Lösungsinhalt zu finden; die Funktionalität der Suchsysteme, darunter fehlender Suchkomfort und Suchfunktionen, schwierige Navigation und Übersicht auf den Seiten; negative Suchergebnisse, wie zum Beispiel fehlendes Suchwort, Suche endet in Shops oder Werbung im Display[12].

Die Suche nach bereits existierenden und relevanten Lösungen ist ein grundlegender Punkt in der Produktentwicklung. In diesem Kapitel wurde die Abstraktion als Richtlinie zum effizienteren Finden der Lösung genommen, da dieser das etappenweise Arbeiten und das Vorgehen vom Abstrakten zum Konkreten unterstützt. Trotz den vielfätigen Hilfsmitteln und Sammlungen gelingt die Lösungssuche

leider nicht problemlos. Das kann auf fehlende Information oder auf Such- und Zugriffsmechanismen zurückgeführt werden. Diese Mechanismen basieren auf Begriffe oder Strukturen, weshalb die Kenntnis darüber für das Auffinden der Lösung erforderlich ist. Zu diesem Zweck ist die Semantik von Begriffen eine Möglichkeit, um die unterschiedlichen Sichten und Fachbegriffen zu überwinden. Es wird eine Zugriffslogik benötigt, die folgende Anforderungen erfüllt[2]:

- Zugriff auf Lösungen über die Abstraktion
- breites Lösungsfeld und erschließen ähnlicher Lösungen

In den nächsten Kapiteln dieser Arbeit wird ein Lösungsansatz zur Überwindung der Probleme vorgestellt.

3 Ontologien und Semantik

Wissen ist Macht (Francis Bacon) -

ist ein bekanntes Sprichwort, welches sich auch im Geschäftsleben bewährt. Denn Wissen hat als Wettbewerbsfaktor einen enormen Einfluss auf den Erfolg von Unternehmen. Sei es im Marketing über aktuelle Trends und Verkaufsanzahl, in der Produktion über Qualitätsuntersuchungen, Lieferantenkontakte oder in der Produktentwicklung über die Marktanalyse oder Produktionsanalyse. Informationen werden in allen Phasen der Produktentstehung benötigt[2]. Dabei ist es nicht entscheident einen Informationsflut zu haben, sondern die richtige Information zum richtigen Zeitpunkt zu haben. Deshalb ist das Management von Wissen sehr wichtig und der Übergang vom Suchen zum Finden muss durch geeignete Methoden behandelt werden. Eine besondere Rolle hat dabei die Ontologie, die als Repräsentation von Informationen auf semantischer Ebene dient [13].

In diesem Kapitel sollen Grundlagen und Werkzeuge zu Ontologien und Semantik behandelt werden.

Begriffserläuterung

Bevor die Grundlagen zu Informationsmanagement präsentiert werden, werden zunächst die Begriffe Daten, Information und Wissen definiert und voneinander abgegrenzt. Die Beziehung der Begriffe zueinander kann als zunehmende Anreicherung semantischer Informationen angesehen werden (siehe Abbildung 3.1). Die syntaktische Anordnung von Zeichen ergeben Daten, die etwas wahrnehmbares darstellen wie beispielsweise eine Telefonnummer. Wenn den Daten durch einen Kontext eine Bedeutung zugewiesen wird, wird daraus eine Information. In dem gegebenen Beispiel ist die Telefonnummer von Max Mustermann. Eine Verknüpfung von Information und Erfahrung führt zu Wissen. Das heißt, dass der Wissende bzw. der Wissensträger die Information verstehen kann. In dem Beispiel wiederum ist dem Wissensträger bekannt wer Max Mustermann ist, andernfalls ist Max Mustermann eine unbekannte Person und die Telefonnummer ist nur eine Information. Auf der Wissensebene kann der Wissende sein Handeln aus dem Wissen ableiten und zur Problemlösung einsetzen[2].

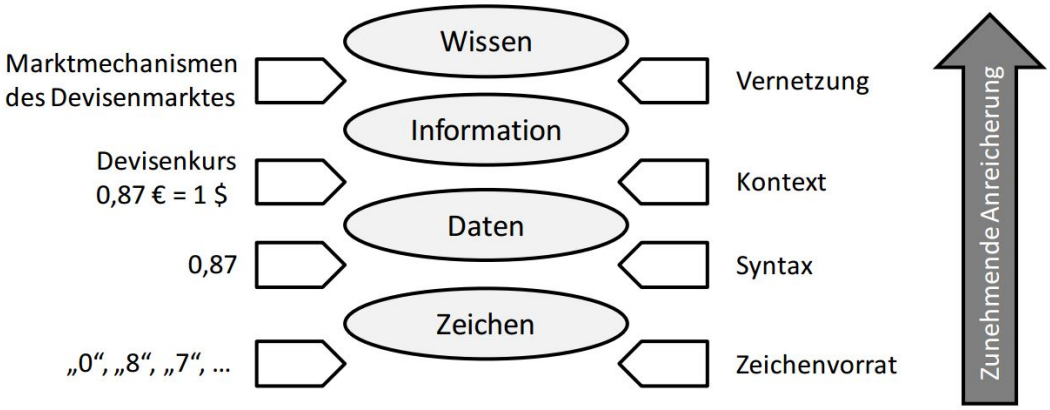

Abbildung 3.1: Abgrenzung und Beziehung der Begriffe Zeichen, Daten, Information und Wissen[2]

3.1 Informations- und Wissenmanagement

Das Informationsmanagement beschäftigt sich mit der Speicherung, Verwaltung und Weitergabe von Informationen. Außerdem umfasst es Vorgehensweisen, die zur Erfassung von Informationen dient und diese automatisch strukturiert und bereitstellt[14]. Informationssysteme sind dabei Mensch-Maschine-Systeme, die zur Deckung des Informationsbedarfs dienen. Datenbanken oder Produktmanagementsysteme sind solche Informationssysteme[2].

Im Vergleich dazu dient das Wissensmanagement zur Lösungssuche durch die kognitiven Fähigkeiten des Menschen wie zum Beispiel die Wahrnehmung oder das Denken und schafft einen Mehrwert. Die Informationen können somit mit weiteren Kontexten verbunden und verknüpft werden. Wissensbasierte Systeme bilden eine Anwendungsdomäne als informationstechnisches Modell nach, dadurch dass sie die beim Menschen bekannten Grundsätze der kognitiven Modellierung nutzen. Dafür ist zunächst eine Konzeptualisierung zu definieren, die die Festlegung der Einheiten und deren Merkmale beinhaltet. Einheiten sind zum Beispiel konkrete Objekte und die Relationen unterhalb diesen Objekten. Daraus können Schlussfolgerungen abgeleitet werden. Daraufhin wird die Konzeptualisierung formalisiert und implementiert [15].

3.2 Information Retrieval

Eines der Probleme beim Zugriff auf Informationen ist das unstrukturierte Vorliegen der Informationen. Als Antwort können Information Retrieval Systeme gesehen werden, die selbst bei unstrukturierten Inhalten das Auffinden von Informationen ermöglichen. Unter Betrachtung der Bedeutung von Informationen werden durch semantische Technologien die Verarbeitung von Informationen unterstützt.

Information Retrieval beschäftigt sich- wie auch der Name schon verrät- mit dem wieder Auffinden von Informationen. Dabei können diese in unstrukturierter Form wie Briefe, Artikel oder Bücher darliegen. Die Aufgabe des Information Retrieval ist die Repräsentation von Daten, sodass man auf Bedeutung von der Information zugreifen kann und Information filtern kann. So soll der Anfragende möglichst gut bei der Lösungssuche unterstützt werden.

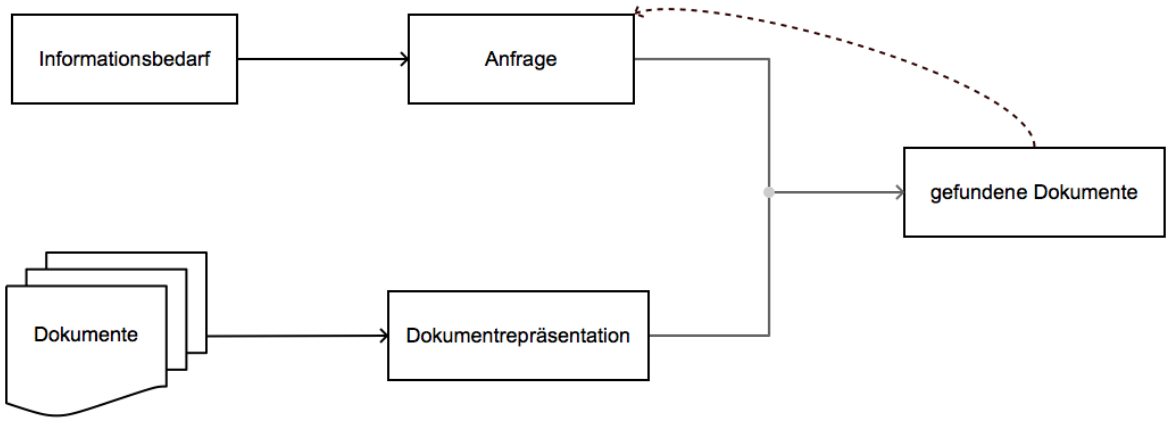

Abbildung 3.2: Vorgehen beim Information Retrieval

Das Vorgehen beim Information Retrieval ist in Abbildung 3.2 zu sehen. Infolge einer Anfrage werden relevante Dokumente gesucht. Dazu wird die Anfrage und die Dokumente in eine Repräsentation überführt, die maschinell leichter verarbeitet werden kann. Durch Verknüpfung von Anfrage und Dokumentrepräsentation wird das Ergebnis geliefert. Ist das Ergebnis nicht zufriedenstellend, kann ein Rückkopplungsprozess stattfinden. Das kann beispielsweise der Benutzer manuell durch das Modifizieren der Anfrage durchführen[16]. Im Mittelpunkt eines Information Retrieval Verfahrens steht das Überprüfen der Relevanz zwischen der Suchanfrage und dem Ergebnis.

Zur Bewertung der Ergebnisse werden die Variablen *Recall* und *Precision* verwendet. Diese sind wie folgt definiert:

- **Recall** liefert den Anteil der relevanten Dokumente im Ergebnis im Verhältnis zu den relevanten Dokumenten insgesamt. Gewünscht ist ein möglichst hoher Recall-Wert.

$$Recall = \frac{RelevanteDokumenteinderAntwortmenge}{RelevanteDokumenteinsgesamt}$$

- **Precision** gibt die Quote der relevanten Dokumente in der Antwortmenge unter allen gefundenen Dokumenten an. Auch hier ist ein hoher Wert erwünscht, denn je kleiner der Wert, desto mehr irrelevante Dokumente sind im Ergebnis enthalten.

$$Precision = \frac{RelevanteDokumenteinderAntwortmenge}{GefundeneDokumenteinsgesamt}$$

Die Qualität der Information Retriecal Systeme lassen sich durch die angegebenen Formeln einschätzen. Eine grafische Darstellung der einzelnen Mengen, die zur Berechnung von Recall und Precision herangezogen werden, ist in Abbildung 3.3 reflektiert.

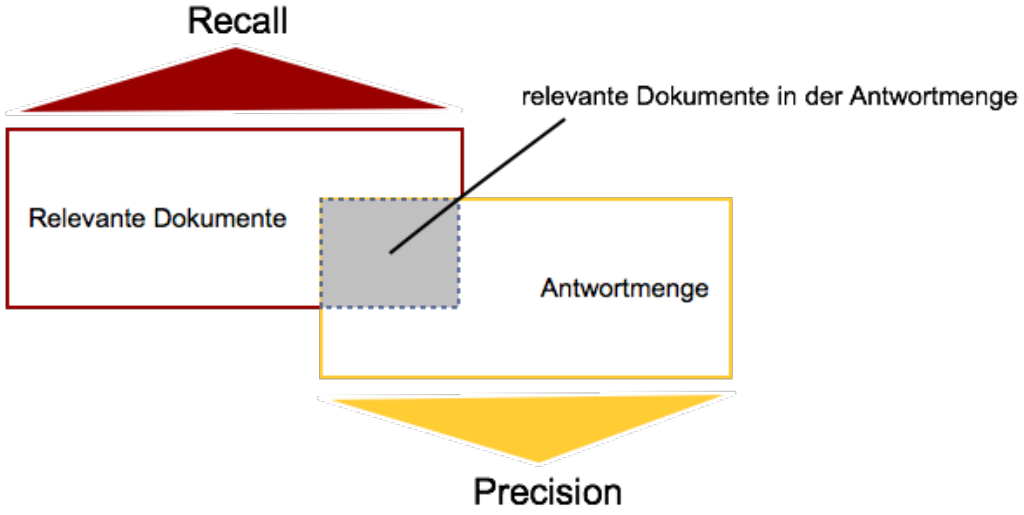

Abbildung 3.3: Grafische Darstellung von Recall und Precision

Information Retrieval Systeme greifen auf textuelle Datenbestände, weshalb Methoden zur Verarbeitung der natürlichen Sprache notwendig sind. Das Zurückführen der Wörter auf eine einheitliche Wortform ist eines der wichtigen Anforderungen um unterschiedliche Flexionsformen, Derivationsformen oder Komposita zu behandeln. Natural Language Processing (NLP) bewilligen die Normalisierung von Texten und ermöglichen eine strukturierte Weiterverarbeitung [2]. Desweiteren existieren weitere Bearbeitungsschritte, um die Semantik der Dokumente zu adressieren. Wiederum steht die Abstraktion im Vordergrund. Wesentliche Bereiche sind die Stoppworteliminierung, die Stammformreduktion,

die Mehrwortgruppenbehandlung und die Betrachtung von Begriffsbeziegungen.

Bei der Stoppworteliminierung sollen Wörter, die keinen Sinn tragen entfernt werden. Das sind beispielsweise Wörter wie und, oder, im etc.. Sie werden als Stoppwörter bezeichnet. Durch die Eliminierung wird der Speicherbedarf reduziert und es treten die charakteristischen Begriffe stärker hervor. Die Stammformreduktion führt die Wörter zu ihrer Grundform zurück. Zum Beispiel werden die Begriffe „lauter“ und „Hauses“ auf „laut“ und „Haus“ zurückgeführt. Die Mehrwortgruppenbehandlung hilft zusammengesetzte Wörter zu verstehen. Synonyme und sinnverwandte Wörter sollen durch die Betrachtung von Begriffbeziehungen entdeckt werden. Dabei helfen Thesauri, in denen inhaltliche Beziehungen zwischen Begriffen verwaltet werden [16].

Information Retrieval Systeme werden in Verbindung mit der Suche im Internet als Web Information Retrieval und generell als Suchmaschinen bezeichnet. Sie unterstützen den Nutzer bei der Suche nach relevanten Internetseiten. Der bekanntester Vertreter ist *Google* und gilt mittlerweile als eine Art Synonym für die Web-Suche [17].

3.3 Semantische Technologien und Ontologien

Der Wissensaustausch und der Wissenskonsum sind das primäre Element im Einsatz von semantischen Technologien. Semantische Technologien stützen auf den Gebrauch von semantischen Informationen, die in einem semantischen Netz gespeichert werden. Unter einem semantischen Netz wird ein formales Modell von Begriffen und deren Relationen verstanden. Zahlreiche informationstechnische Anwendungen verwenden diese Art der Wissensrepräsentation, wie im Bereich des Natural Language Processing[18].

3.3.1 Das semiotische Dreieck

Für die Definition von Semantik sollen die Elemente des semiotischen Dreiecks verglichen werden. Dabei ist die Semiotik die Lehre von Zeichen. Sie beschäftigt sich mit der Repräsentation von Informationen in Zeichen und deren Benutzer Im Dreieck werden die Begriffe Syntax, Semantik und Pragmatik voneinander abgegrenzt. Dabei ist zwischen dem Zeichen, der Bedeutung von Zeichen und der Realität, die das Zeichen symbolisiert, zu unterscheiden.

Element	**Beziehung**
Syntax	Zeichen $\leftrightarrow$ Zeichen
Semantik	Zeichen $\leftrightarrow$ Bedeutung
Pragmatik	Zeichen $\leftrightarrow$ Benutzer und Situation

Tabelle 3.1: Elemente und Beziehungen des semiotischen Dreiecks[18]

Die Syntax unterscheidet sich von der Semantik dadurch, dass sie nur die formalen und strukturellen Beziehungen der Zeichen ohne die Bedeutung beschreibt. Die Semantik hingegen untersucht die

Sprachzeichen und deren Designate. Die Pragmatik wendet sich der Entstehung, den Anwendungen und den Auswirkungen von Sprachzeichen zu, d.h. sie bezieht sich auf den Zeichenbenutzer. So versucht die Pragmatik die Semantik eines Begriffes mit dem Kontext der Interpretierenden zu verknüpfen (siehe Tabelle 3.1) [2, 18, 19].

3.3.2 Die semantische Treppe

Es gibt verschiedene Modelle zur Abbildung der Semantik von Begriffen, die sich in einer sogenannten semantischen Treppe darstellen lassen. Wie auch auf Abbildung 3.4 zu sehen ist, nimmt die semantische Reichhaltigkeit mit jeder Stufe der Treppe zu.

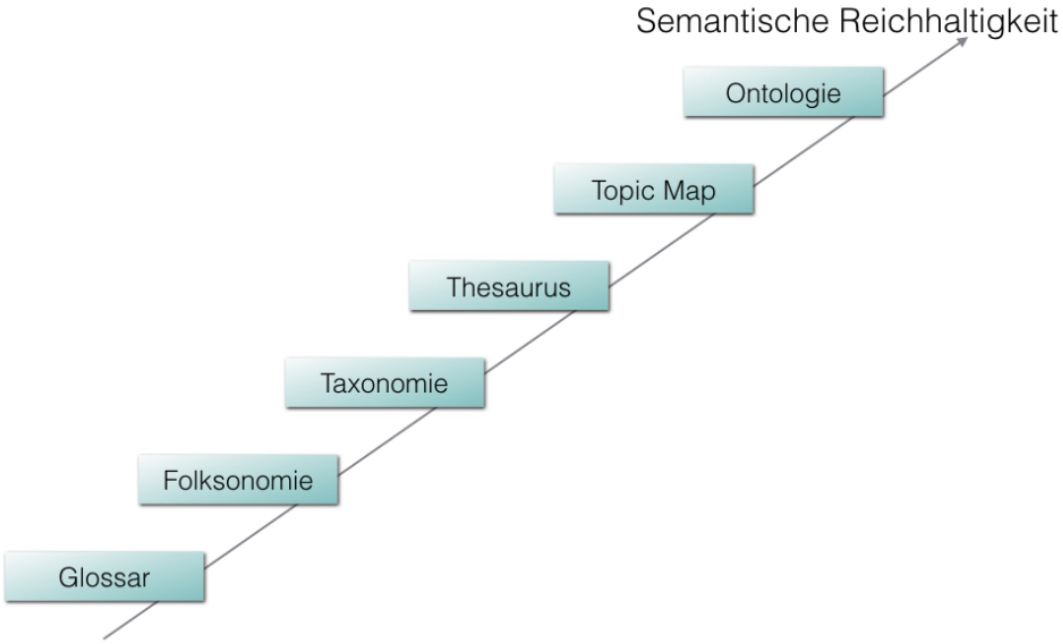

Abbildung 3.4: Semantische Treppe [20]

Auf der nierdrigsten semantischen Stufe ist das Glossar, welches eine wörterbuchartige Auflistung mit Erklärungen von Begriffen ist. Die nächste Stufe, die Folksonomie, handelt um das gemeinsame Indexieren von digitalen Informationen. Zugeordnet werden die Schlagworte, die als „tags" bezeichnet werden, durch die Nutzer, wobei es keine festen Regeln gibt. Eine Beispielplattform dafür ist *BibSonomy*[1].

Semantisch reichhaltiger sind Klassifizierungen und Taxonomien, die über die Begriffe hinaus auch deren hierarchische Strukturierung ausweisen[2]. Eine Taxonomie ist dabei eine Klassifikation, in der Objekte oder Themen geordnet und zusammengefasst werden. Hierarchische Systeme lassen sich durch den Detaillierungsgrad differenzieren und können als Baumstruktur dargestellt werden.

[1] http://www.bibsonomy.org/

Thesauri enthalten weitere Zusatzinformationen zu den Begriffen und beschreiben die Beziehungen und Zusammenhänge zwischen diesen wie beispielsweise mehrdeutige Wörter, Synonyme oder Deskriptoren [DIN 1463].

Topic Maps sind ein XML-basiertes Datenformat die ebenfalls Zusatzinformationen enthält. Die Begriffe werden Themen zugeordnet und nach verschiedenen Aspekten strukturiert. Verglichen mit der Ontologie ist die Vererbung von Eigenschaften jedoch nicht möglich und kann somit solche Zusammenhänge nicht darstellen[20].

Das am semantisch reichhaltigste Modell ist die Ontologie. Für die Begriffsbestimmung wird meist die Definition von Gruber herangezogen. Diese lautet:

> *An ontology is an explicit specification of a conceptualization*[21].

Dabei versteht man unter der Konzeptionalisierung die Abbildung der relevanten Einheiten einer Anwendungsdomäne als Konzepte, die festgelegt werden müssen[2]. Die Ontologie beschreibt nicht das Verständnis einer einzelnen Person, sondern die Realität[22]. Im Gegensatz zu Taxonomien ist die zwanghafte eindeutige Klassifizierung notwendig. Ein Objekt wird bezüglich des Umfanges eingeordnet. Dadurch kann aus unterschiedlichen Ausgangspunkten auf eine Information zugegriffen werden. Somit erhöht sich auch die Vollständigkeit der relevanten gefundenen Informationen, da diese nicht nur einer Klasse zugeordnet werden, sondern bezüglich der Bedeutungskontext und der Relationen betrachtet werden [20].

3.4 Einsatz von Ontologien

Verschiedene Bereiche und Zwecke ziehen Nutzen aus den Vorteilen der Ontologie. Hierbei steht die Wissensrepräsentation im Vordergrund. In Information Retrieval Systemen werden Ontologien als Wissensmodell verwendet um relevante Informationen aufzufinden [23].Mittlerweile existieren auch zahlreiche Forschungprojekte wie *SEKT*[2] oder *THESEUS*[3], die sich mit der Entwicklung und Optimierung von bestehenden Werkzeugen und Methoden, aber auch mit der Anwendung der Technologien beschäftigen. Das Forschungsprojekt THESEUS entwickelte beispielsweise ein eigenes Modell und eine Ontologie im Bereich der Medizintechnik, um Patienteninformationen, die für aktuelle Diagnosefälle relevant sind, anzubieten [24]. Weitere Anwendungsbereiche der Ontologie sind in der Bioinformatik[25] und vor allem im Semantic Web. Dabei ist das Semantic Web wie folgt definiert:

> *The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation.*[26].

[2] Semantic Knowledge Technologies, konzentriert sich auf den semantik-basierten Zugriff auf Informationen in Netzwerken

[3] Das Forschungsprogramm THESEUS wurde vom Bundesministerium für Wirtschaft und Technologie gestartet, um neue internetbasierte Wissensinfrastruktur zu aufzubauen

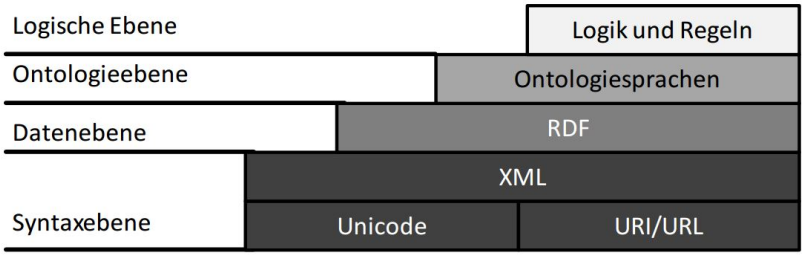

Abbildung 3.5: Ebenen der Wissensrepräsentation im Semantic Web [27]

Die Architektur des Semantic Web gründet auf bestehende Standards wie *Uniform resours identifiers (URIs)* um auf Objekte zu referenzieren und *Unicode* um Zeichensätze zu verschlüsseln. Es werden auch bereits existierende Web Technologien wie *Extensible Markup Language (XML)* für die Syntax verwendet[27]. Auf der Datenebene wird der Zusammenhang von Objekten auf semantischer Ebene festgelegt durch beispielsweise das *Resource Description Framework RDF*. Komplexere Zusammenhänge können durch Ontologiesprachen ausgedrückt werden. Eine standardisierte Sprache ist die *Ontologie Web Language*. Darüber hinaus können weitere logische Folgerungen aus der Logik und festgelegte Regeln erschließen werden. In Abbildung 3.5 sind die verschiedenen Ebenen der Wissensrepräsentationen im Semantic Web zu sehen. Sie stufen sich durch die Ausdrucksstärke der Sprachlogik voneinander ab [2].

3.4.1 Wissensrepräsentation in der Produktentwicklung

Für die Wissensdarstellung in der Produktentwicklung existieren ebenfalls semantische Repräsentationen von Produkt und Entwicklungsprozessen. Bei der Lösungssuche mithilfe von Ontologien wird zusätzlich zu der hierarchischen Struktur von Begriffen auch deren Bedeutung genutzt. Nachfolgend werden beispielhaft einige Forschungsprojekte veranschaulicht, die technische Systeme mit Ontologien modellieren.

S. Hami-Nobari und *L. Blessing* stellen in der internationalen Konferenz für konstruktive Entwicklung die Ausstellung von Produkten durch die Ontologie mit *ist-ein* und *besteht-aus* Beziehungen bzw. mit Taxonomien und Baugruppenstrukturen. Diese objektorientierte Beschreibung der Abhängigkeiten zwischen den Baugruppen reduziert die Anzahl der definierten Abhängigkeiten , weil sie so nur ein Mal definiert werden müssen und danach wiederverwendet werden können[2, 28]. Desweiteren definieren *N. Lohse, G. Valtchanov, S. Ratchev, M. Onori* und *J. Barata* eine Ontologie, um Produktinformationen mit Prozessen, um genauer zu sein mit Montagevorgängen, zu verknüpfen [29].

Zudem gibt es mehrere Beiträge, die eine Ontologie für die Produktentwicklung entwickeln, welche genutzt werden um Textdokumente, die in der Entwicklungspraxis verfasst werden, zu indexieren und auch Schwerpunkt dieser Arbeit ist. Eines dieser Ontologien ist *Engineering Design Integrated Taxonomy (EDIT)* und wurde von *S. Ahmed, S. Kim* und *K. M. Wallance* entwickelt[30]. Die Dimension der Ontologie *EDIT* bezieht das Produkt, den Entwicklungsprozess, die Entscheidungslogik und die Funktion ein. Vergleichsweise entwickelte *J. Ponn* und *U. Lindemann* eine Ontologie, worin die

Branche, die Funktion, die Anforderungen an das Produkt, das Wirkprinzip, die Produkteigenschaften und die Unternehmen, die die Produkte herstellen zu den Konzepten der Ontologie gehören [9].

4 Entwicklung einer Ontologie

Ontologien dienen zum besseren Verständnis zwischen Mensch und Maschine. Als Modell eines Anwendungsbereiches zeigen sie die semantischen Beziehungen innerhalb von Begriffen bezüglich eines festgelegten Sachverhaltes. Leider ist das manuelle Erstellen und Warten von Ontologien ab einer gewissen Datenmenge und Komplexität bedenkenlos nicht mehr realisierbar. Deshalb sind automatische oder semi-automatische Technologien notwendig, die den Menschen unterstützen [20]. In diesem Kapitel sollen Methoden zur Entwicklung von Ontologien und die Grundstruktur einer Ontologie vorgestellt werden.

4.1 Vorgehensmodelle zur Entwicklung von Ontologien

Die Entwicklung einer Ontologie beginnt mit der Planung, worauf die grundlegenden Phasen **Spezifizierung, Konzeptualisierung** und **Implementierung** folgen. Bei der Durchführung der Phasen ist kein streng sequentieller Ablauf zwingend. Durchaus sind Rücksprünge erlaubt und manchmal sogar notwendig. Die **Spezifizierung** soll den Hauptziel, den Zweck, die Detailgenauigkeit und den Anwendungsbereich der Ontologie zusammenstellen. Die **Konzeptualisierung** organisiert und strukturiert das erworbene Wissen, indem es externe Repräsentationen benutzt, die unabhängig von der Implementierungssprache und -Umgebung sind. Zuletzt soll bei der **Implementierung** das konzeptuelle Modell zu einem Implementierungsmodell überführt werden [31].

Aus diesem Grundmodell können weitere Prozesse abgeleitet werden. Ein allgemeines Vorgehensmodell für die Ontologieentwicklung ist auf Abbildung 4.1 [32] zu sehen. Der erste Baustein ist die **Machbarkeitsstudie**, in der Probleme und mögliche Lösungen identifiziert werden. Außerdem wird die Durchführbarkeit bezüglich den Kosten, der Technik und der Planung geprüft. Nach diesem Schritt beginnt die eigentliche Entwicklung. An dieser Stelle, beim **Kick-Off** werden die Anforderungen geklärt und schriftlich festgehalten. Zugleich werden Quellen erforscht und eine erste semi-formale Ontologie erstellt. Darin sind relevante Begriffe so wie Konzepte der Domäne enthalten. Daraufhin kann die semi-formale Ontologie **verfeinert** werden. Hierbei wird zwischen drei Methoden unterschieden. Bei der top-down-Methode werden die Konzepte und Relationen der Ontologie schrittweise detailliert. Entgegenstellend werden bei der bottom-up-Methode aus bestehenden Textquellen die Begriffe als Instanzen herausgefiltert und mit diesen Klassen und Relationen gebildet. Zudem gibt es eine Methode- die middle-out-Methode- die zwischen den beiden Methoden steht und die wichtigsten Konzepte spezifiziert.

Die Formalisierung der Ontologie erfolgt durch den Aufbau einer Taxonomie, woraufhin weitere Relationen gebildet und Konzepte instanziiert werden. Sie schließt auch die Implementierung mit ein. Es sollte dennoch beachtet werden, dass die Verfeinerung der Ontologie regelmäßig evaluiert

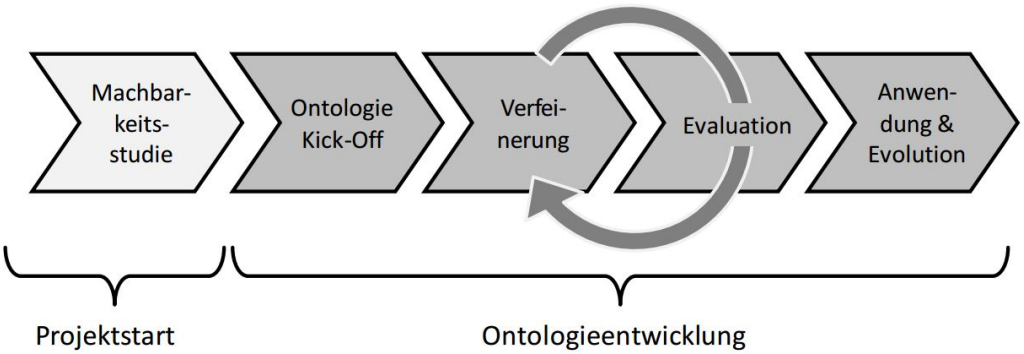

Abbildung 4.1: Vorgehensmodell für die Entwicklung einer Ontologie [32]

wird. Die **Evaluation** an sich stellt sich aus einer technologie-, nutzer- und ontologieorientierten Bewertung zusammen. Die Übereinstimmung der Syntax mit den Standards gehört zur Bewertung der Technologie, wie auch die Skalierbarkeit und weitere technische Eigenschaften. Die nutzerorientierte Bewertung wird als sehr wichtig eingestuft. Dabei sollten die Anforderungen der Nutzer erfüllt werden. Die Evaluation der Ontologie für sich urteilt die formalen Eigenschaften und optimiert sie wenn nötig. Als letzter Punkt ist die **Evolution** ein organisatorischer Prozess, die strikte Regeln in Bezug auf die Aktualisierung und Wartung der Ontologie festsetzt [2, 32].

Neben diesen allgemeinen Prozessen gibt es bei großen oder zusammengesetzten Ontologien weitere Aspekte, die berücksichtigt werden sollten. Das Modulasierieren der großen Ontologie erleichtert den Umgang und bringt Vorteile bezüglich Wissensverarbeitung mit sich [33]. Bei zusammengesetzten Ontologien wird die Ontologie von verschiedenen Personen zu einer Anwendungsdomäne verknüpft [2].

4.1.1 Vorgehensmodell zur Entwicklung von Ontologien in der Produktentwicklung

In Kapitel 3 (Seite 28) wurde EDIT kurz als Beispielontologie der Produktentwicklung eingeführt. An dieser Stelle wird es nochmals aufgegriffen und als Modell zur Entwicklung von Ontologien in der Produktentwicklung vorgestellt. EDIT teilt das Verfahren in zwei Hauptabschnitte, wobei im ersten Abschnitt die Ontologie auf die Wissenserfahrung beruhend entwickelt wird und im zweiten Abschnitt durch computerbasierte Verfahren erweitert und verfeinert wird [2, 30].

In Abbildung 4.2 sind die einzelnen Schritte und die zwei Abschnitte zu sehen. Als ersten Schritt werden die **Hauptkonzepte der Taxonomie identifiziert**. Dazu wird die Anwendung bestimmt, wofür die Ontologie entwickelt wird. Es werden auch die Hauptkonzepte, die das Anwendungsgebiert beschreiben, festgelegt. In diesem Fall ist es zum Beispiel die Produktentwicklung. Nach diesem Schritt wird nach **bestehenden Taxonomien** gesucht. Ist der Befund nicht zufriedenstellend, so müssen neue **Taxonomien entworfen** werden. Dafür werden Klassen und streng hierarchisch strukturiert Unterklassen aufgebaut. Danach werden Instanzen aus dem Anwendungsgebiert integriert. Unterdessen wird die Vollständigkeit der Hauptkonzepte und der Taxonomien bezüglich der

spezifischen Anwendung überprüft. Zeitgleich zu diesen Entwicklungsschritten werden Interviews, Literaturrecherchen und Dokumentanalysen durchgeführt und die Informationen bei der Entwicklung verwendet.

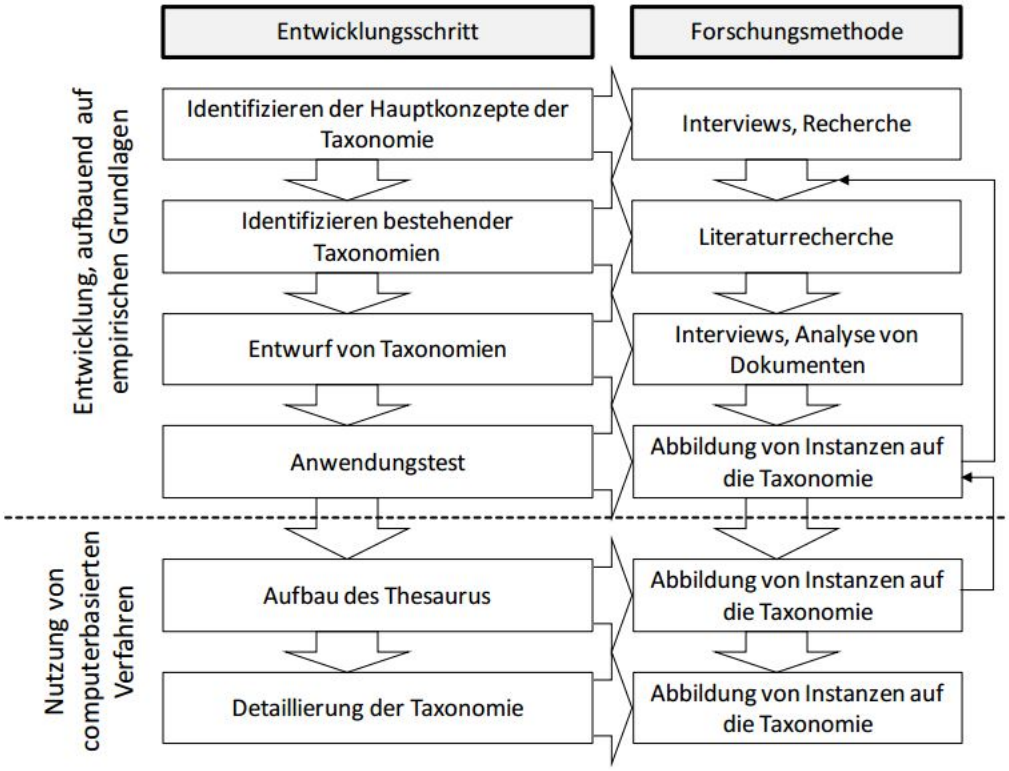

Abbildung 4.2: Vorgehensmodell für die Entwicklung einer Ontologie in der Produktentwicklung [30]

Im zweiten Abschnitt der Entwicklung wird ein **Thesaurus der Begriffe** für den computerbasierten Verfahren aufgebaut. Die Konzepte und Instanzen der Ontologie werden mit einem Thesaurus hauptsächlich automatisch verknüpft. Jedoch ist eine Kontrolle notwendig. Diese Verknüpfung hilft zum Verständnis von zweideutigen Wörtern und ist als Vorbereitung zur automatischen Ergänzung der Instanzen der Ontologie zu sehen. Im letzen Schritt erfolgt die eigentliche Befüllung und **Detaillierung der Ontologie**, welcher eine ständige Pflege und Aktualisierung der Ontologie einschließt.

4.2 Basisontologie

Zur Entwicklung der Ontologie werden die im vorherigen Abschnitt vorgestellten Vorgehensmodelle zu Hilfe genommen. Die Machbarkeitsstudie und Anforderungen wurden bereits in vorausgehenden Kapiteln angesprochen und werden hier nicht nochmals erläutert. Eine semi-formale Ontologie wird im Folgenden erstellt und die Grundstruktur bekannt gemacht. Bei der Erstellung der Grundstruktur, welches den Kern der Ontologie bildet, sollen die Hauptkonzepte und Relationen festgelegt werden. In

dieser Bachelorarbeit wurde beispielhaft eine Ontologie der Automobil-Domäne aufgebaut, welches als Netz in Abbildung 4.3 zu sehen ist. Es wurden lediglich die wichtigsten Konzepte bezüglich des Automobils verwendet. Diese Ontologie wird bei der Implementierung eingesetzt.

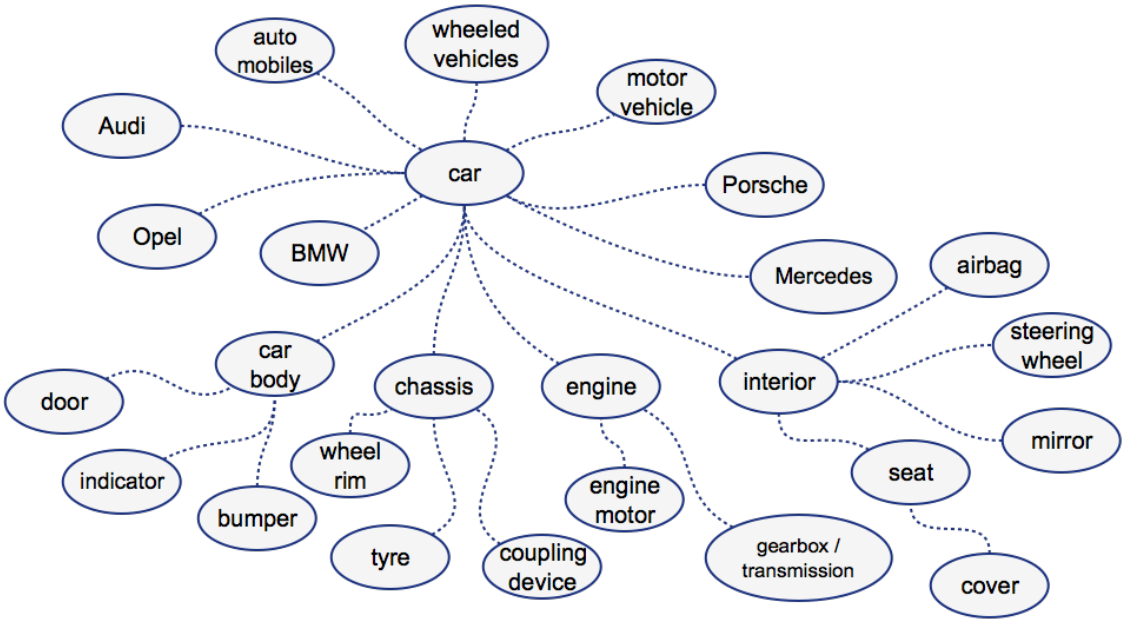

Abbildung 4.3: Basisontologie der Automobil-Domäne

4.3 Erweiterung der Ontologie

Nachdem die Grundstruktur der Ontologie aufgebaut wurde ist die Verfeinerung der nächste Schritt. Dieser Schritt bezieht sich auf eine bestimmte Anwendungsdomäne. Die Wissensextraktion aus Texten ist bedeutsam, weil Lösungen meistens mit textuellen Dokumenten beschrieben werden. Deshalb wird darauf näher eingegangen. Abbildung 4.4 beschreibt den Vorgang, wie aus textuellen Beschreibungen eine Ontologie gewonnen werden kann. Dazu ist eine Textannotation und Extraktion der Instanzen notwendig. Nach der Eliminierung redundanter Begriffe werden die Instanzen mit den Hauptkonzepten der Ontologie verknüpft. Wenn keine geeigneten Konzepte in der Ontologie enthalten sind werden weitere Konzepte und Relationen hinzugefügt und die Ontologie weiter detailliert.

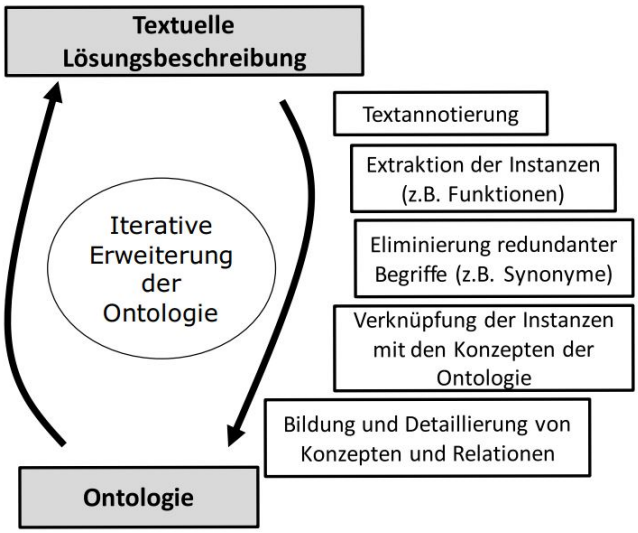

Abbildung 4.4: Erweiterung der Ontologie durch Extraktion der Instanzen aus Textdokumenten [2]

Für die Erweiterung der Ontologie wurden in dieser Bachelorarbeit Textdokumente über Autos verwendet. Die einzelnen Schritte, die auf der Abbildung 4.4 zu sehen sind werden später bei der Implementierung näher erläutert. Prinzipiell werden alle Begriffe, die aus dem Lösungsdokument extrahiert worden sind, als Instanzen festgelegt, deren Kategorie bestimmt und in die Ontologie integriert. Somit werden die vorhandenen Informationen abgebildet. Außerdem werden die Vorteile des Abstrahierens, die in den vorangehenden Kapiteln erläutert wurden, gewonnen, weil nicht nur ein spezieller Begriff sondern die Kategorien der Begriffe in die Ontologie eingebunden werden. Somit soll die Lösungssuche erleichtert werden.

5 Related Works

Ontologien werden seit den letzten Jahren immer mehr erforscht und in verschiedenen Bereichen eingesetzt[34]. Für diese Arbeit ist besonders der Einsatz im Wissensmanagement interessant. Da in dieser Bachelorarbeit die Kategorisierung von Texten analysiert wird werden im Folgenden die dafür relevanten und ähnlichen Arbeiten vorgestellt.

5.1 Text Categorization Using Automatically Acquired Domain Ontology

Eine klassische Methode der Textklassifikation ist die Betrachtung der Schlüsselwörter im Text. Das Problem hierbei ist, dass nicht jeder dieselben Schlüsselwörter verwendet. In dem Paper **Text Categorization Using Automatically Acquired Domain Ontology** von *Shih-Hung Wu, Tzong-Han Tsai* und *Wen-Lian Hsu* wird eine Methode vorgestellt, indem die Texte mithilfe automatisch erworbener Domain Ontologie klassifiziert werden[35].

Die Idee ist es, die Begriffe innerhalb eines Dokumentes im Trainingsset zusammenzustellen und diese zu benutzen, um die Dokumente im Testset zu verstehen. Der Erwerb einer Ontologie ist ein iterativer Prozess, der Schlüsselwörter sammelt und in die Struktur ordnet. Die Ontologie wird in jedem Durchlauf überarbeitet, verfeinert und gesammelt: Um das Hypernym eines Schlüsselwortes zu finden werden die Sätze, die das Schlüsselwort und die zugehörigen Hypernyme enthalten beobachtet. Dabei werden Regeln hergeleitet, um Hypernyme finden zu können. In jedem Durchlauf werden die Regeln verfeinert, um bessere Qualität der Schlüsselwort-Hyponym-Paare zu finden. Der Ablauf ist semi-automatisch. Der Mensch muss nur die Ergebnisse überprüfen.

Information Map (InfoMap), ein Framework zur Wissensdarstellung, wurde entworfen um sprachliches, allgemeines und fachliches Wissen zu integrieren. **InfoMap** bietet das Verständnis von natürlichen Sprachen und wird angewendet in Bereichen wie Question Answering (QA) und Wissensmanagement. Eine wichtige Eigenschaft von **InfoMap** ist, dass es das Geschehen in einem Satz herausfiltert, indem es die Hauptwörter festhällt (gewöhnlich sind es Subjekt-Verb-/Hypernym-Hyponym- Paare), die in der Domain Ontologie definiert sind. Wie die Domain Ontologie besteht die **InfoMap** aus Domain-Konzepten und den zugehörigen Teilkonzepten wie Kategorien, Attribute oder Aktivitäten.

Die Beziehung zwischen den Konzepten und den Teilkonzepten bilden eine Baumform (siehe Abbildung 5.1). **InfoMap** definiert Referenzen um Knoten von verschiedenen Branchen zu verknüpfen,

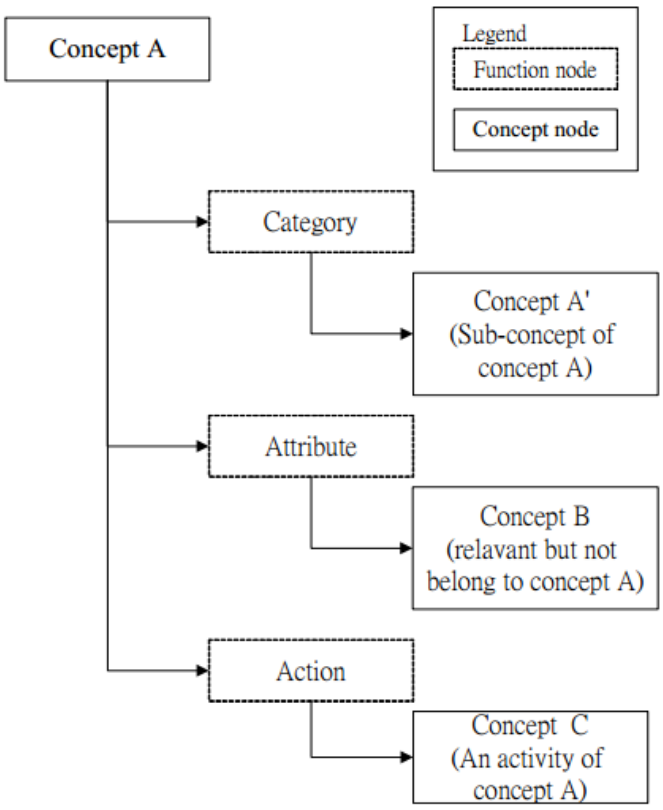

Abbildung 5.1: Ontologie-Struktur des InfoMap [35]

die im Ganzen ein Netzwerk bilden *(Concept Nodes)*. Es werden auch die Beziehungen unter den Konzepten definiert *(Function Nodes)*. Der Wurzelknoten definiert den Namen des Domains. Dieser Knoten beinhaltet die wichtigen Themen in einer hierarchischen Anordnung.

Wortpaare mit semantischer Beziehung geben mehr Informationen über einen Satz, als zwei seperate Wörter. Diese Wortpaare werden *event structure* genannt. Mithilfe der Domain Ontology können Texte kategorisiert werden, indem jeder einzelner Satz im Text kategorisiert wird. Dabei wird InfoMap als Wörterbuch benutzt und die Wörter zerlegt. Um die Mehrdeutigkeit zu beheben wird von der *domain topic* Gebrauch gemacht. Es wird die Verteilung der Wörter in den Ontologien überprüft und die Domäne der Sätze festgestellt.

Für den automatischen Erwerb einer Domain Ontology werden drei Schritte benötigt:

1. Schlüsselwörter identifizieren
2. Verwandte Begriffe finden
3. Damit verbundene Tätigkeiten verbinden

Für die Identifikation der Schlüsselwörter wird die Auftretenshäufigkeit betrachtet. Diese Schlüsselwörter sind die Kernthemen der Ontologie. Dann wird **SOAT**, ein Tool um semi-automatisch

Ontologien zu bilden, angewendet. **SOAT** bildet für ein gegebenen Korpus einen Prototyp der Domain Ontologie. **InfoMap** benutzt taxonomische und nicht-taxonomische Beziehungen zwischen den Begriffen, wobei es sich bei den taxonomischen um Kategorien und Synonyme und bei den nicht-taxonomischen um Attribute und Aktionen handelt. Um diese Beziehungen zu erfassen definiert **SOAT** Regeln, die aus den Pattern der Schlüsselwörter und Variablen bestehen. Die Regeln zur Extraktion sind morphologische Regeln, die aus part-of-speech (POS) markierten Satzstrukturen konstruiert werden.

SOAT Prozess:

Input: Domain Korpus mit POS Markierung
Output: Prototyp der Domain Ontologie

1. Wähle ein Schlüsselwort (oft ist es der Domainname) als den Kern aus dem Korpus um ein potentiales Wurzelset R zu setzen
2. Führe folgenden rekursiven Prozess aus:
   a) Nehme ein Schlüsselwort A als Kern aus R
   b) Finde ein neues zugehöriges Schlüsselwort B aus der Wurzel A durch die Extraktionsregel und füge es zu der Regel hinzu
   c) Lösche A aus der Wurzel, wenn es keine weiteren zugehörigen Schlüsselwörter gibt
   d) Setze B in das potentiale Wurzelset

Wiederhole Schritt zwei solange bis entweder R leer ist oder die obere Grenze an Knoten erreicht wird.

Sowohl die Grenze als auch die Features um ein Text zu erschließen werden durch den Entscheidungsbaum-Algorithmus festgelegt. Die Features des Event Structe beinhalten *event structure score*, *node score*, *fired node level* und *node type*. Event Structure Score wird verwendet um zu schauen, ob der Weg des Entscheidungsbaums relevant ist. Das Ziel einer Domain-Spekulation ist es, einen Set S in eine Domäne D bezüglich den kombinierten Scores und dem Event Structure im Satz zuzuordnen.

$$SimScore(D_j, S) = KeywordScore(D_j, S) + \alpha * EventStructureScore(D_j, S)$$

Um das KeywordScore zu berechnen wird das TF/IDF Klassifizierer benutzt. Dieser stellt eine Domäne als einen gewichteten Vektor $D_j = (w_{j1}, w_{j2}, \cdots w_{jn})$ dar, wobei n die Anzahl der Wörter in der Domäne ist und w das Gewicht des Wortes k ist. Außerdem ist

$$w_k = nf_{jk} * idf_{jk}$$

wobei nf_{jk} die Auftretenshäufigkeit des Wortes w_k in der Domäne j ist. $idf_k = log(\frac{|D|}{DF_k})$, wobei DFk die Anzahl der Domänen in der das Wort k aufkommt ist und $|D|$ die gesamte Anzahl an Domänen ist.

Die Ähnlichkeit zwischen einer Domäne j und einem Satz wird dargestellt durch einen Vektor $D_i KeywordScore(D_j, S) = Sim(D_j, D_i) = \frac{\sum_{k=1}^{n} w_{jk} * w_{ik}}{\sqrt{\sum_{k=1}^{n}(w_{jk})^2 \sum_{k=1}^{n}(w_{ik})^2}}$

Die Sätze werden bezüglich den Domänen kategorisiert und bepunktet. Somit hat jeder Satz einen individuellen *Score(D,S)* für jede Domäne. Diese werden addiert um den gesamten Score für einen Text zu berechnen. Die Domäne mit der höchsten Score ist diejenige, in die der Text kategorisiert wird.

5.2 Research on Domain Ontology Semi-automatic Construction Model towards Chinese Text

Aufgrund der fehlenden Strukturierung der Informationen im World Wide Web gewinnt das Semantik Web immer mehr an Bedeutung, da dieser versucht die Riesenmenge an Informationen in einer spezifischen Form zu repräsentieren. Es vereinfacht die Interaktion zwischen Computer und Mensch und verbessert die Wiederverwendbarkeit von Informationen. Die Technologie der Ontologie ist für das semantische Web von großer Bedeutung und kann vom Computer durch semantische Informationen leicht bearbeitet werden. Es definiert die grundlegenden Begriffe und Relationen des Domänen-Vokabulars.

Das Paper **Research on Domain Ontology Semi-automatic Construction Model towards Chinese Text** von *Liu Ning, Li Guanyu* und *Zhang Yuanfa* handelt von einem semi-automatischen Konstruktionsmodell für die Domain-Ontologie mit der Absicht den Konstruktionsprozess zu beschleunigen, indem die Beteiligung der Experten reduziert und das Automatisierungsgrad verstärkt wird. Betrachtet werden Chinesische Texte, welches den Prozess noch weiter erschwert, weil sie flexibler und komplexer als die englische Sprache ist und die meisten Forschungen sich auf das Englische beziehen[37].

Die Ontologie wird definiert als ein Fünfertupel $O := C, R, H^c, Rel, A^o$, wobei Cund R zwei disjunkte Mengen sind, indem das Erste Begriffe und das Zweite Beziehungen abbildet.H^c figuriert die Begriffshierarchie, die die taxonomische Beziehung zwischen den Begriffen bildet. Rel ist die nicht-taxonomische Beziehung zwischen den Begriffen. A^o definiert das Axiom der Ontologie.[36]

Das semi-automatische Konstruktionsmodell für die Domain Ontologie besteht aus vier Hauptmodulen: Kernkonstruktion für Ontologien, Ontologieextraktion, Ontologiegenerierung und Bestätigungsmodul für die Ontologie. Mit der Zusammenarbeit der Module und der ständigen Aktualisierung der Kernkonstruktion wird die Domain Ontologie vervollständigt. Im folgenden werden die vier Module genauer beschrieben und sind auf Abbildung 5.2 zu sehen.

Das Ziel ist es das verteilte Domänenwissen von Schiffsfelddokumenten herauszufiltern und diese in die zugehörige Ontologie zu überführen. Die Begriffe und Relationen werden vom *Chinese Thesaurus of Communication* und *Classified Chinese Thesaurus* abgeleitet. Die Ontologie der Schiffsdomäne wird nach dem Gebrauch des Schiffes unterschieden. In diesem Fall sind es drei Konzepte: das Militärschiff,

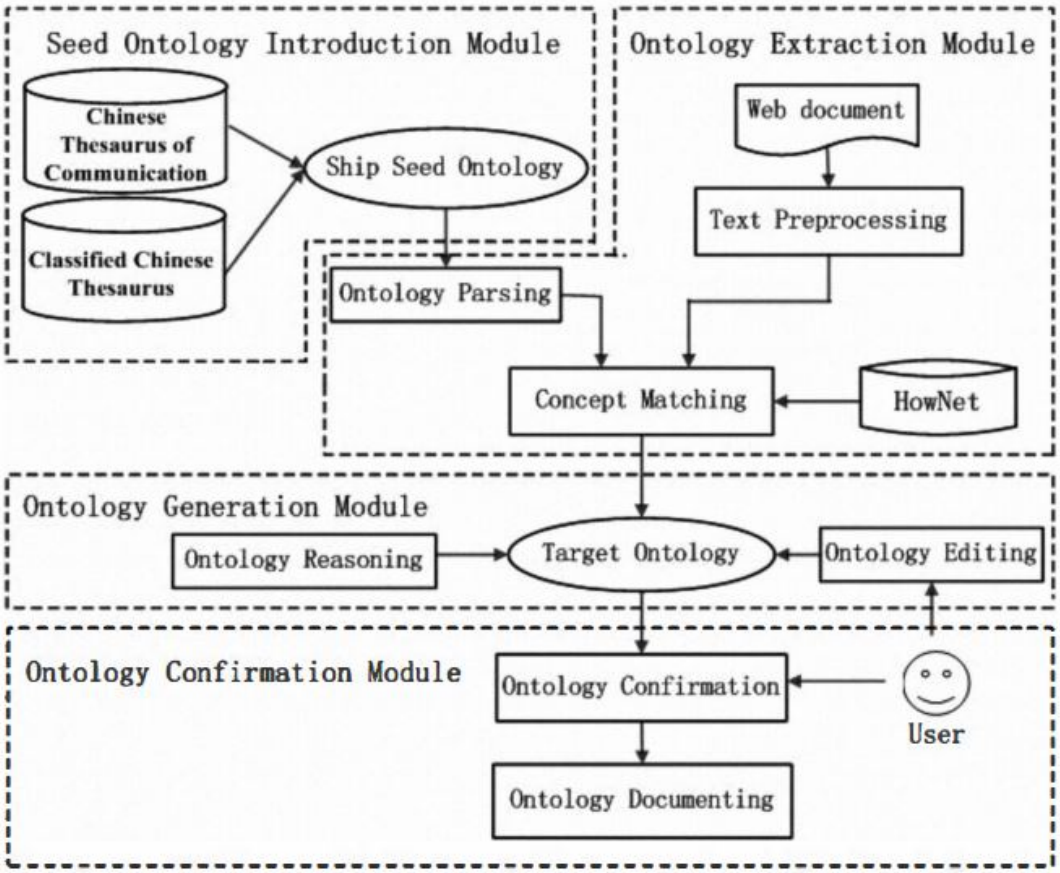

Abbildung 5.2: Framework des semi-automatischen Konstruktionsmodells für die Domain Ontologie[37]

zivile Schiffe und Wissenschaftsschiff. Die Ontologieextraktion wird in weitere drei Module unterteilt. Es setzt sich aus der Zerlegung der Ontologie, die Texvorverarbeitung und der Begriffsanpassung zusammen. Das Modul der Zerlegung der Ontologie ist für die Datenanalyse der Kernontologie von *Jena API* verantwortlich, das die folgenden Schnittstellen hat:

- **OntModel** ist eine Erweiterung des *Jena RDF* und stellt die Funktion zur Verarbeitung der Ontologiedaten bereit. Das Verarbeitungsprozess der Ontologie von *Jena* muss zunächst ein Ontologiemodell erstellen. Dann kann das Modell von der Funktion bearbeitet werden.
- **Document manager** ist eine Klasse die dabei hilft, die Dokumente zu verwalten. Desweiteren können Ontologiemodelle kontruiert werden indem Ontologiedokumente importiert werden. Dies kann auch online durchgeführt werden.
- **OntClass** definiert die relativen Funktionen für die Begriffe und Relationen.
- Von **OntResource** erben alle Klassen, die Ontologiewerte in der Ontologie API repräsentieren. Deshalb ist es eine geeignete Stelle um die geteilte Funktionalitäten solcher Klassen zu speichern. Es erfolgt eine praktische Wiedergabe der Werte für generelle Methoden.

Die Datenquelle des **Textvorverarbeitungsmoduls** sind chinesische Texte, welche viel komplexer als englische Texte in der NLP sind. Um einen Textinhalt für den Computer bestimmbar zu machen sind vier Schritte nötig.

1. Bilder, Links und Sonstiges entfernen, welches keine Zeichen sind.
2. Den Text als Sätzesammlung darstellen.
3. Die chinesischen Wörter in den Sätzen segmentieren durch Nutzung der *ICTCLAS*, welches von der *Chinese Academy of Science* entwickelt wurde.
4. Stoppwörter entfernen

Nach der Vorverarbeitung des Textes liegt der Text als Sammlung von Sätzen und diese wiederum als Sammlung von Wörtern dar.

Das **Begriffsanpassungsmodul** dient zur Berechnung der semantischen Ähnlichkeit zwischen den Begriffen, die von chinesischen Texten extrahiert und der Kernontology analysiert worden sind. Die Begriffe werden der Kernontologie zugeordnet, zu der der Wert für die semantische Ähnlichkeit am höchsten ist. Dafür wird das chinesische Wörterbuch *HowNet* in zu Hilfe genommen.

Da ein Wort aus mehreren Begriffen zusammengesetzt werden kann, wird in dem Fall seperat die Ähnlichkeit der Wörter angesehen und das Maximum der zwei Wörter als gemeinsamer Wert angenommen.

Im *HowNet* benutzen Begriffe sind Basiseinheiten von Bedeutungen, genannt Semem. Die Ähnlichkeit der Bedeutungen werden durch die Ähnlichkeiten der Sememe mit der nachfolgenden Formel berechnet:

$$Sim(C_1, C_2) = \sum_{i=1}^{4} \beta_i \prod_{j=1}^{i} Sim_j(S_1, S_2)$$

Die in der Formel addierten β sind einstellbare Parameter, wobei $\beta 1 + \beta 2 + \beta 3 + \beta 4 = 1$ und $\beta 1 \geq \beta 2 \geq \beta 3 \geq \beta 4$ gilt. Die Werte von Sim_1 bis Sim_4 sind die Ähnlichkeitswerte für die Begriffe, die die Merkmale der ersten grundlegenden Semem-Beschreibung, andere grundlegende Semem, die relationale Semem-Beschreibung und die relationale Symbolbeschreibung abbilden. Diese Merkmale sind die Merkmale der Struktur, mit dem *HowNet* Wörter ausdrückt.

Typischerweise ist die Ähnlichkeit definiert durch die Pfadlänge, sodass die Ähnlichkeit zweier Sememe invers proportional zur Pfadlänge im Baum sind. Die Ähnlichkeit der Sememe kann berechnet werden durch $Sim(S1, S2) = \frac{\alpha}{dis(S1, S2) + \alpha}$, wobei α ein einstellbarer Parameter ist und die Bedeutung der Abstand ist, wenn der Ähnlichkeitswert 0,5 beträgt.

Die Zielontologie ist nach der Durchführung der erläuterten Module erreicht. Jedoch ist für die Korrektheit und Vollständigkeit der Ontologie eine Auswertung und Bestätigung erforderlich. Das Modul der **Ontologieerzeugung** überprüft die Korrektheit der Zielontologie. Außerdem wird die Zeilontologie entwickelt und ständig verbessert als Kernontologie der nächsten Konstruktionsrunde. Um ein Wissensmodell zu konstruieren, welches ein Domänenwissen entwirft und eine virtuelle Datenansicht formt um die uneinheitlichen Datenquellen einfach zu integrieren, exportiert dieses

Modul das Ontologiedokument automatisch (durch Protege). Durch dieser Weiße wird die Effiziens der Konstruktion einer Domain Ontologie verbessert.

6 Implementierung

Die Bedeutung von Informationen bei der Produktentwicklung und auch die Problematik bei der Informationssuche wurde in den vorangehenden Kapiteln festgehalten. In dieserm Abschnitt soll nun eine Lösungsmöglichkeit vorgestellt werden, wie bei der Einordnung von Texten in Kategorien von Ontologien profitiert werden kann. Nachfolgend wird die Realisierung der Lösungsmöglichkeit präsentiert. Dabei werden zunächst die verwendeten Technologien und die Vorgehensweise bekannt gemacht. Daraufhin wird die ontologieorientierte Lösung dieser Arbeit detailliert vorgestellt.

6.1 Verwendete Technologien

Für die Umsetzung des Lösungsansatzes wurde von einigen Technologien Gebrauch gemacht. Dazu gehören sowohl NLP Werkzeuge als auch semantische Netzwerke.

6.1.1 Stanford CoreNLP

Das *Stanford CoreNLP*[1] ist ein Framework, welches verschiedene NLP-Werkzeuge enthält. Es ist in Java geschrieben und muss nicht trainiert werden. Eingegeben wird ein Text (in unserem Fall in englischer Sprache), welches verarbeitet und als ein annotierter Satz zurückgegeben wird. Währenddessen werden Wörter in ihre Grundform abgewandelt und ihre Wortart bestimmt. Außerdem werden numerische Mengen normalisiert und die Struktur des Satzes zerlegt. Die Grundstruktur des Standford CoreNLP setzt sich aus den Klassen Annotation und Annotator zusammen, die in einer Pipeline integriert sind. Annotationen sind Datenstrukturen, in denen die Resultate der Annotationen gespeichert werden. Annotators sind Funktionen, die auf Annotationen ausgeführt werden. Diese Funktionen sind Prozesse wie Tokenisierung, Parsing und weitere Sprachverarbeitungsvorgänge [38].

6.1.2 WEKA

Waikato Environment for Knowledge Analysis (WEKA)[2] ist eine Open Source Software der Universität in Waikato in Neuseeland, welches in Java geschrieben ist. Es bietet Methoden für maschinelles Lernen und Data Mining an und hat viele Klassifikatioren wie Naive-Bayes, Decision Tree oder Support

[1] http://nlp.stanford.edu/
[2] http://www.cs.waikato.ac.nz/ml/weka/

Vector Machine. Außerdem bietet Weka weitere Werkzeuge für Datenvorverarbeitung, Clustering, Regression und Visualisierung.

6.1.3 BabelNet

BabelNet[3] ist ein semantisches Netzwerk, welches an der Universität La Sapienza in Rom entwickelt wurde. Es ist eine Zusammenstellung aus *Wikipedia*[4] und *WordNet*[5]. Dabei wird durch maschinelle Übersetzung ein Wörterbuch erzeugt, in dem zu den Begriffen auch deren semantische Bedeutung und Relationen und auch Synonyme und Synsetz inbegriffen sind. BabelNet gruppiert die Wörter verschiedener Sprachen in Synonym-Sets, die Babel Synsets genannt werden. Für jeder dieser Babel Synsets werden kurze Definitionen aus *Wikipedia*, *WordNet*, *Open Multilingual Wordnet*[6], *Wikidata*[7], *Wiktionary*[8] und *OmegaWiki*[9] erzeugt.

6.2 Vorgehensweise

Bei der Produktentwicklung ist Wissen der wichtigste Baustein. Jedoch ist es nicht immer einfach Wissen anzueignen. Nachfolgend soll zunächst die Ausganglage und darauf folgend ein Lösungsansatz für die Problematik vorgestellt werden.

6.2.1 Ausgangslage

Sowohl bei Softwareprodukten als auch bei technischen Produkten ist zunächst eine Planung der Produktentwicklung notwendig. Dabei sind Informationen die grundlegendste Ressource, sowohl bei der Planung als auch ber der Herstellung. Diese Informationen sind verteilt auf textuelle Daten, die strukturiert oder unstrukturiert vorliegen. Der Mensch benötigt lange Zeit, um diese Daten zu analysieren. Neben Informationen ist die Zeit eine weitere bedeutungsvolle Ressource bei der Produktentwicklung. Vor allem für die Wettbewerbsfähigkeit ist die Schnelligkeit bei der Herstellung ausschlaggebend. Computerprogramme sind dazu fähig, Texte schnell zu verarbeiten. Dieser Vorteil bringt jedoch ein Problem mit sich. Im Vergleich zu dem Mensch können Computer den Inhalt der Texte nicht ohne Weiteres verstehen. Wir suchen eine Lösung, die die Semantik eines Textes schnell und automatisch extrahiert.

[3] http://babelnet.org/
[4] https://www.wikipedia.org/
[5] https://wordnet.princeton.edu/
[6] http://compling.hss.ntu.edu.sg/omw/
[7] https://www.wikidata.org/
[8] https://www.wiktionary.org/
[9] http://www.omegawiki.org/

6.2.2 Lösungsansatz

Ausgehend vom oben genannten Punkt wird in dieser Arbeit eine Software entwickelt, der Textinhalte automatisch kategorisiert. Die Implementierung basiert auf die Bachelorarbeit mit dem Thema *Erkennung und Klassifikation nichtfunktionaler Anforderungen in einem Entwicklungsprozess unter Machine Learning* von *Kheriddine Oueslati*[39]. In der Arbeit wird eine Methode zur automatischen Erkennung und Klassifikation nicht-funktionaler Anforderungen umgesetzt. Mithilfe von Natural Language Processing (NLP) Analyse-Werkzeugen werden Daten bearbeitet und infolgedessen die Wortelemente in den Sätzen identifiziert. Mit Unterstützung von Klassifikationsmodellen in der Data-Mining Workbench WEKA werden diese Daten trainiert und ausgewertet.

Diese Arbeit differenziert sich von der Arbeit von Kheriddine Oueslati dadurch, dass bei der Klassifikation die semantische Bedeutung der Wörter durch Ontologien einbezogen werden. Außerdem werden nicht Anforderungen klassifiziert, sondern die inhaltliche Bedeutung der Texte. Im Gegensatz zur Arbeit von Kheriddine Oueslati, in der die Klassen fest vorgeschrieben sind, werden die Klassen in dieser Arbeit dynamisch über den Zugriff auf *BabelNet* bestimmt. Voraussetzung ist ein Internetzugriff und ein Account bei BabelNet. Anders als bei der vorherigen Arbeit sind bei dieser Implementierung die Trainingsdaten und die zu klassifizierenden Daten dieselben. Der Grund dafür ist, dass während der Trainingsphase eine Ontologie passend zum Text konstruiert wird. Die Ontologie wird aus den jeweiligen Kategorien der im Text vorkommenden Wörter erstellt.

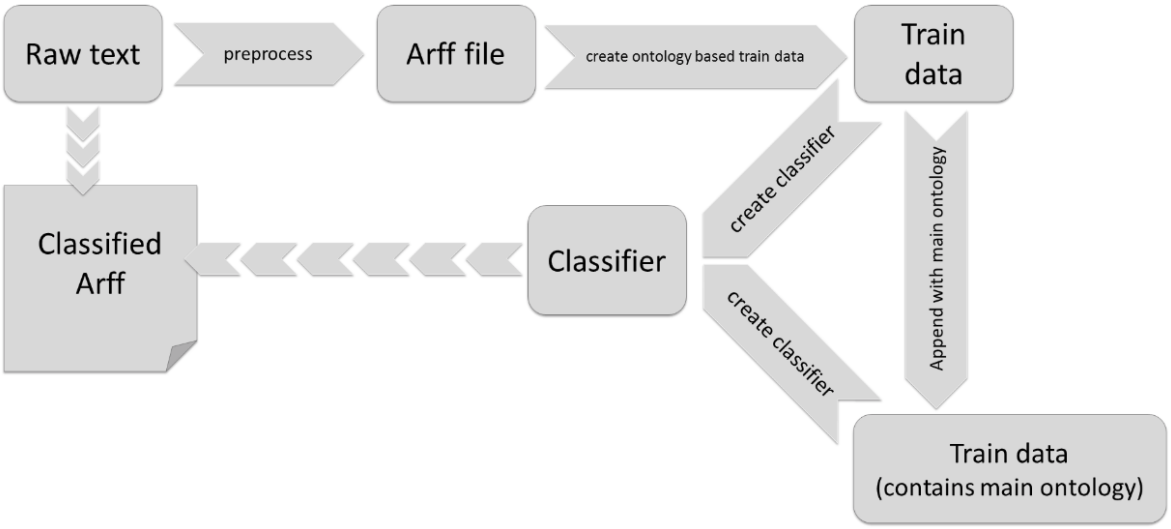

Abbildung 6.1: Programablauf

Für die Bearbeitung wurden in dieser Bachelorarbeit beispielhaft vier englische Texte ausgewählt, die sich auf die Automobil-Domäne beziehen. In das Programm wurde eine Basisontologie der Automobil Domäne integriert, die durch den Eingabetext erweitert und verfeinert wird. Der genaue Ablauf mit Ein- und Ausgabedateien ist in Abbildung 6.1 zu sehen. In das Programm wird eine Rohdatei eingegeben. Dieser Text wird im Preprocess bearbeitet und in *Attribute-Relation File Format (ARFF)* gebracht, da WEKA mit dieser Dateiform arbeitet. Aus der ARFF Datei werden Ontologie basierte Trainingsdaten

erzeugt. Die Ontologie kann der Basisontologie, die in Abschnitt 4.2 vorgestellt wurde, hinzugefügt werden und somit eine erweiterte Trainingsdatei erzeugt werden. Um ein Klassifikationsmodell zu erhalten kann eines der beiden Trainingsdateien genutzt werden. Die Klassifizierung erfolgt durch Eingabe des Rohtextes und dem Klassifizierer. Die Ausgabe ist eine klassifizierte ARFF-Datei.

Im Preprozess kann zwischen drei Methoden aus dem Natural language Processing ausgewählt und durchgeführt werden. Dazu gehört die Lemmatisierung, die Wörter auf ihre Grundform zurückführt, Eliminierung der Stoppwörter, also Wörter, die keine Bedeutung für das Verständnis des Textinhalts haben, wie zum Beispiel „and“ und „or“ und Alphabetic Tokenizer, der Wortverbindungen eliminiert.

Nach Durchführung des Preprocesses erhält man eine ARFF-Datei, die aus einem Header, der den Relationsnamen, eine Liste aus Attributen und ihren Typen enthält, und den Datensätzen besteht. In Abbildung 6.2 ist eine Beispieldatei zu sehen. Die ersten drei Zeilen gehören zum Header der ARFF-Datei. Es sind die Relationsnamen, Attribut und Attributtyp zu sehen. In den geschweiften Klammern nach *@attribute domain* steht die Wertemenge, die hier leer ist. In unserem Fall besteht sie aus den Konzepten der Ontologie, die aus BabelNet aufgerufen werden. Die nächsten Zeilen bestehen aus den Datensätzen. Zu sehen ist ein Text, auf den die drei Preprocess-Vorgänge ausgeführt worden sind. Die Fragezeichen am Ende der Zeile sind Platzhalter für die Klassenzuordnung nach der Klassifikation.

```
@relation 'PreProcess: Lemmatization & StopWords & AlphabeticTokenizer'

@attribute sentence string
@attribute domain {}

@data
'research show carbon fibre fibreglass highly effective replacement steel body work car . ',?
'innovative development , body work serve battery . ',?
'car manufacturer develop vehicle utilise technology . ',?
'car , composite material car body , replace heavy car battery . ',?
'composite material carbon fibre polymer resin replace steel panel hold charge store traditional battery . ',?
```

Abbildung 6.2: Aufbau der ARFF-Datei

Als nächsten Schritt, der Create Ontology Based Train Data wird die Semantik eingebunden. Dies geschieht in dem für jeden Begriff die Kategorie aus BabelNet zugewiesen wird (siehe Abbildung 6.3. Ein Begriff kann zu mehreren Kategorien gehören. Diese Kategorien werden auch der Wertemenge hinzugefügt. Somit werden semantische Trainingsdaten erzeugt. Diese Trainingsdaten können erweitert werden durch das Zusammenführen der Datensätze mit der Basisontologie. Die Wertemenge wird ebenfalls ergänzt. Aus diesen Datensätzen kann das Klassifikationsmodell erzeugt werden.

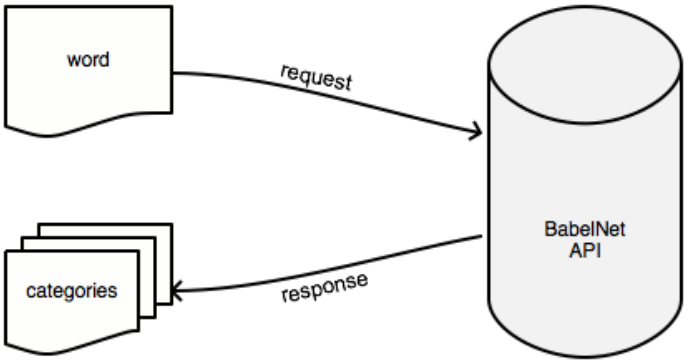

Abbildung 6.3: BabelNet Anfrage

In der Trainingsphase werden dieselben Lernalgorithmen wie in Kheriddine Oueslatis Arbeit benutzt. Diese werden hier nur kurz zusammengefasst und können in seiner Arbeit nachgelesen werden. Das Decision Tree durch Trainingsdaten aufgestellt besteht aus Knoten und Kanten, wobei Knoten die Attribute und die Kanten Attributwerte verkörpern. Die Knoten sind durch Kanten verbunden und stellen einen gerichteten Baum dar. In dem Baum existieren mehrere Pfade, wobei ein Pfad eine Klassifikationsregel angibt. Die Klassifikation erfolgt durch das Durchlaufen der Pfade. Der Naive Bayes berechnet die Wahrscheinlichket der Zugehörigkeit einer Instanz zu einer Klasse. Eine Instanz wird zur Klasse zugeordnet, deren bedingte Wahrscheinlichkeit am höchsten ist. Der Support Vector Machine fügt den Trainingsdaten eine Hyperebene für die Entscheidung der Klassenzugehörigkeit hinzu, die zu allen Datenpunkten maximale Entfernung aufweist. Die Datenpunkte, die am nahesten an der Hyperebene liegen bilden den Support Vector und werden für die Klassifikation berücksichtigt.

6.3 Technische Umsetzung

In diesem Abschnitt wird die technische Umsetzung des Lösungsansatzes erläutert. Es wurde die Implementierung von Kheriddine Oueslati, die mit Java umgestzt wurde, erweitert. Die Software stützt sich auf Standford CoreNLP, WEKA und BabelNet.

6.3.1 Oberfläche der Software

In Abbildung 6.1 ist der Ablauf von roher Texteingabe bis zur Klassifikation abgebildet. Diese Schritte sind in drei Phasen einteilbar:

- Vorverarbeitung
- Training
- Klassifizierung

Die Phasen der Vorverarbeitung und Klassifizierung wurden aus der Arbeit von Kheriddine Oueslati übernommen. In der Phase des Trainings wurde die Semantik Eingebunden.

Die Oberfläche für die Vorverarbeitung ist in Abbildung 6.4 zu sehen. Die einzelnen Bearbeitungsschritte, wie in Abschnitt 6.2.2, sind Lemmatisierung, Eliminierung der Stoppwörter und Alphabetic Tokenizer. In dieser Phase wird der Rohtext je nach Auswahl des NLP-Werkzeuges vorverarbeitet und eine ARFF-Datei erzeugt.

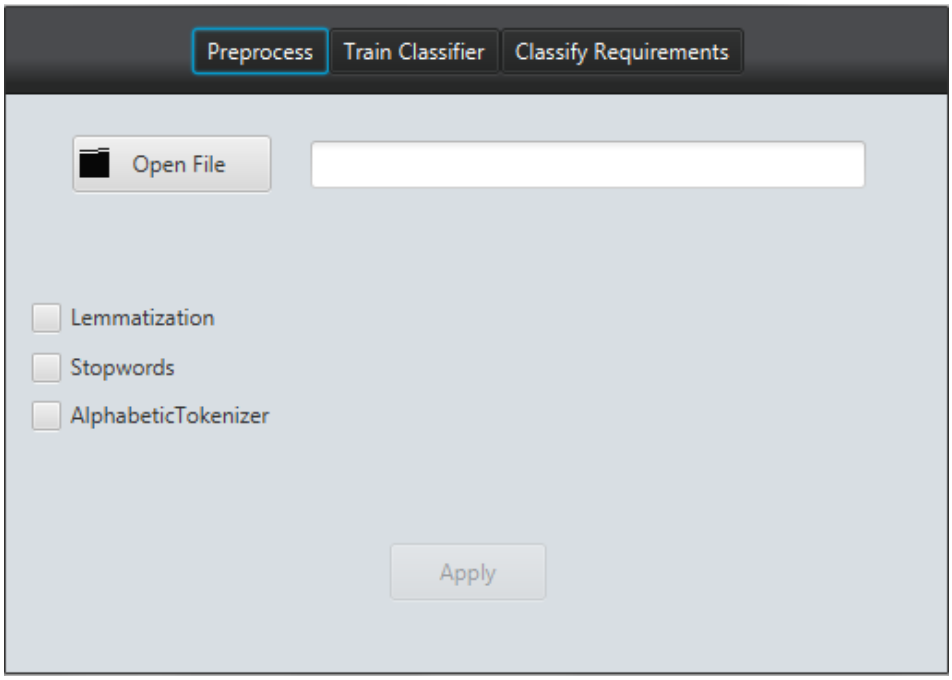

Abbildung 6.4: Tab PreProcess

In Abbildung 6.5 ist die Oberfläche vom Tab Train Clasifier abgebildet. Hier hat man die Möglichkeit einen von drei Classifyern durch Eingabe von Trainingsdaten zu erstellen. Diese drei sind Decision Tree, Naive Bayes und Support Vector Machine. Aus vorverarbeiteten Texten können mit Auswahl von *Create Ontology Based Train Data* Ontologie basierte Trainingsdaten erzeugt werden. Es werden alle Kategorien für die jeweiligen im Text vorkommenden Wörter von BabelNet abgerufen. Da pro Wort die Kategorien an der BabelNet API synchron angefragt werden, kann es bei größeren Texten mit mehreren Wörtern zu längeren Wartezeiten führen. Um die Performance in dieser Hinsicht zu verbessern, wird ein Cache verwendet. Dieser merkt sich die Ergebnisse der bereits angefragten Wörter. Bei identischen Wörtern wird ohne Anfrage an die BabelNet API die bereits gespeicherten Ergebnisse geliefert. Somit wird insgesamt nur eine Anfrage für jedes unterschiedliche Wort gemacht. Desweiteren können bestehende Trainingsdaten an die Basisontologie hinzugefügt werden. Dies ist durch Benutzung von *Append Dataset* möglich. Dadurch werden die Trainingsdaten erweitert. Außerdem müssen alle in den Trainingsdaten vorkommenden Kategorien im Header der ARFF-Datei als Wertemenge des entsprechenden Attributs aufgelistet sein.

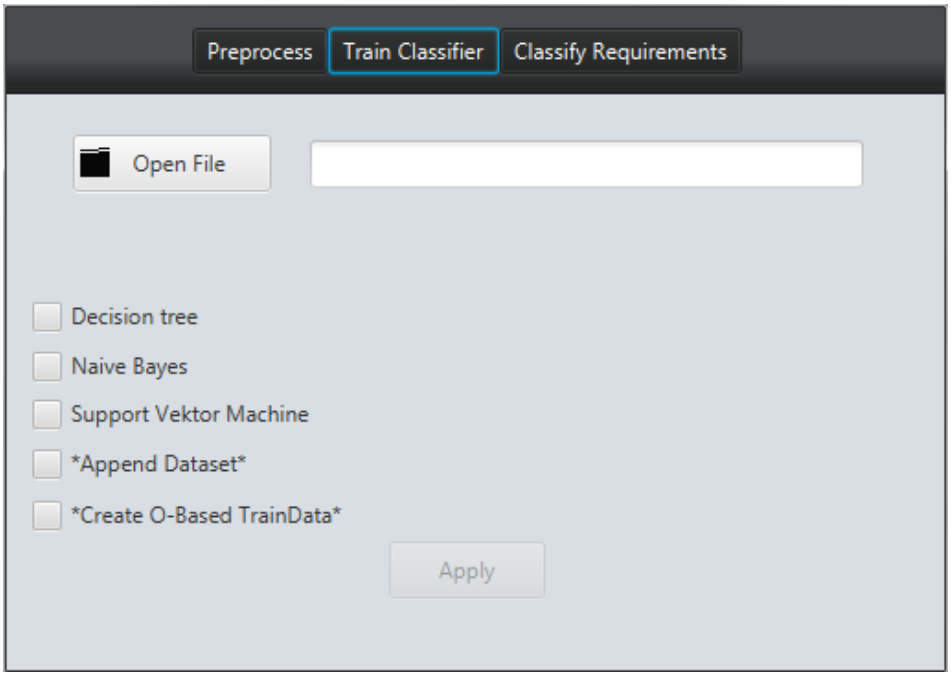

Abbildung 6.5: Tab Train Classifier

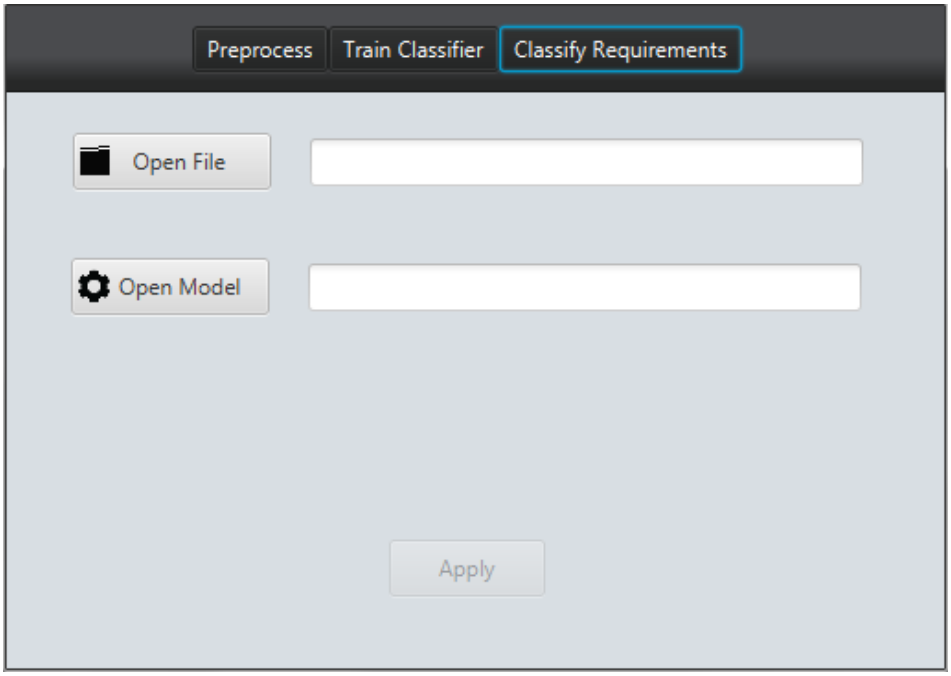

Abbildung 6.6: Tab Classify Requirements

Zur letzten Phase *Klassifizierung* werden als Eingabe zwei Dateien benötigt. Zum Einen der Rohtext und zum Anderen der Classfier, der in unserem Fall die Endung *.model* hat. Die Oberfläche ist in Abbildung 6.6 zu sehen.

6.3.2 Datenstruktur

In Abbildung 6.7 ist die Datenstruktur der Software in einem Klassendiagramm abgebildet. Die Methoden, die im Tab Preprocess verwendet werden, werden grundsätzlich in der Klasse Preprocess zur Verfügung gestellt. Diese greift bei einigen Fällen auf die Klasse Utils. Die Utils Klasse wird von allen drei Phasen verwendet. Zum Beispiel befinden sich die Methoden *createArffFile()* und *saveArffFile()* in der Utils Klasse, die in allen Phasen verwendet werden.

Die Methoden für die Erstellung Ontologie basierter Trainingsdaten und das Erweitern der Basisontologie befinden sich ebenfalls in der Utils Klasse, die von der Train Klasse aus verwendet werden. Die Klasse, die als Classifier gespeichert wird, ist FilteredClassifierExt. Die FilteredClassifierExt Klasse leitet von der eigentlichen FilteredClassifier aus WEKA Klasse ab. Die einzige Erweiterung ist, dass diese Klasse eine Liste von Domains, die beim Erstellen des Classifiers erzeugt werden, hält. Diese Liste wird bei der Klassifizierung benötigt. Die Klasse ModelEvaluation stellt Methoden für die Evaluierung des klassifizierten Textes zur Verfügung. Die Klasse Ontology baut eine Verbindung zu BabelNet und liefert nach Anfragen die Kategorien durch Aufruf von *gerSuperSynset()*. Weitere Relationen der Klassen sind in der Abbildung 6.7 ersichtlich.

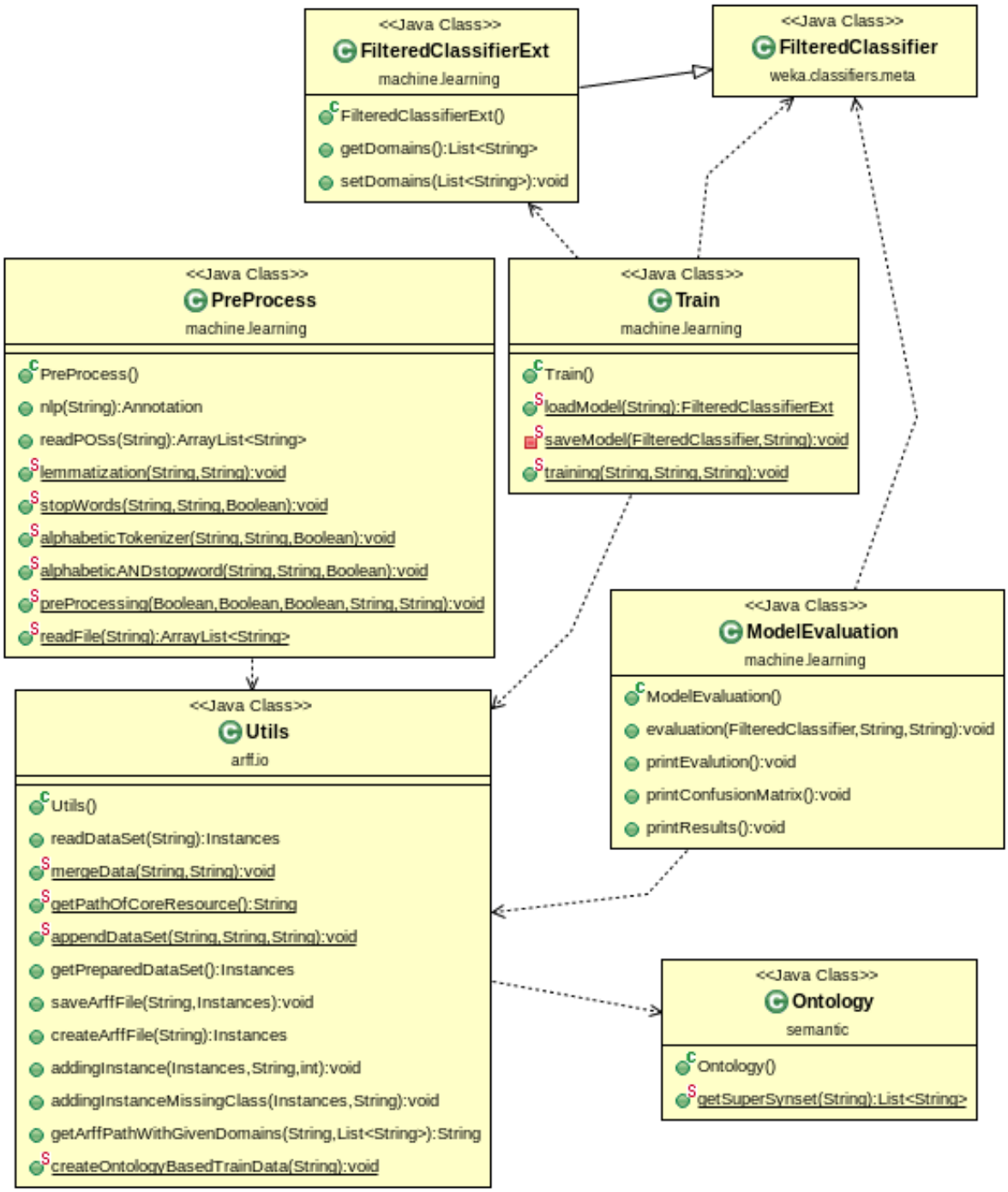

Abbildung 6.7: Klassendiagramm

6.4 Evaluation

In diesem Abschnitt werden die Lösungen der Klassifizierung veranschaulicht. Zuerst werden die Beispieltexte und die Vergleichsmöglichkeiten dargelegt. Daraufhin werden die Ergebnisse verglichen.

6.4.1 Vorgehen

Als Testmenge wurden vier Texte ausgewählt, die zur Automobil-Domäne gehören. Die Texte handeln jeweils von der Karosserie, den Scheinwerfern, dem Autositz und den Materialien eines Fahrzeuges. Die Längen der Texte sind zwischen vier bis sieben Sätze. Alle Texte liegen in englischer Sprache vor.

Es wurden jeweils Preprocess Vorgänge in drei Kombinationen auf die Texte angewendet. Daraus sind jeweils drei Arff-Datein entstanden, die durch Anwendung folgender Preprocess-Vorgängen entstanden sind:

- AlphabeticTokenizer & Lemmatisierung
- AlphabeticTokenizer & Stoppworteliminierung
- AlphabeticTokenizer & Stoppworteliminierung & Lemmatisierung

Aus diesen drei Dateien wurden Trainingsdaten erzeugt, die jeweils nur mit Zugriff auf BabelNet und auch mit Zusammenführung der Basisontologie entstanden sind. Folglich haben sich für ein Text sechs Traningsdaten ergeben. Mit diesen Trainingsdaten wurden jeweils drei Classifier mit Decision Tree, Naive Bayes und Support Vector Machine erzeugt. Pro Text existieren somit insgesamt 18 Classifier mit denen der Rohtext klassifiziert wird.

6.4.2 Ergebnisse

Im folgenden werden die Ergebnisse der Klassifizierung präsentiert. In den Tabellen sind die Anteile des Textes in Prozent angegeben, die korrekt klassifiziert wurden. Dieser Anteil setzt sich aus der Summe der korrekt klassifizierten Sätze im Text. Die falsch klassifizierten Sätze wurden manuell identifiziert und als falsch klassifiziert gekennzeichnet.

Aus der Tabelle 6.1 ist ersichtlich, dass die Texte, bei denen die Trainingsdaten aus der Basisontologie gemeinsam BabelNet erzeugt worden sind, generell besser klassifiziert wurden, als die Texte, bei denen die Trainingsdaten nur aus der Ontologie durch BabelNet erstellt worden sind. Außerdem ist nicht zu übersehen, dass die Vorverarbeitungsprozesse einen großen Einfluss haben. In allen Texten erkennt man eine bessere Klassifizierung, wenn die Eliminierung der Stoppwörter als Preprocess durchgefürt wurde. Die Texte, die nur mit Lemmatisierung vorverarbeitet worden sind, wurden deutlich schlechter klassifiziert. Das ist vor allem bei der Klassifizierung mit Support Vector Machine zu sehen. Über die Klassifikatoren lässt sich sagen, dass Decision Tree und Support Vector Machine zu gleich guten und besseren Ergebnissen geführt haben als der Naive Bayes. In Text eins und zwei liefert Decision Tree die besten Ergebnisse und in Text drei und vier Support Vector Machine.

	Decision Tree			Naive Bayes			Support Vector Machine		
	TL-A	**TS-A**	**TSL-A**	**TL-A**	**TS-A**	**TSL-A**	**TL-A**	**TS-A**	**TSL-A**
	TL-B	**TS-B**	**TSL-B**	**TL-B**	**TS-B**	**TSL-B**	**TL-B**	**TS-B**	**TSL-B**
Text 1	40%	80%	80%	20%	40%	60%	0%	60%	60%
	20%	100%	80%	40%	60%	60%	0%	60%	60%
Text 2	50%	100%	100%	25%	50%	25%	0%	50%	25%
	25%	100%	75%	25%	50%	75%	0%	50%	25%
Text 3	33%	33%	33%	100%	100%	33%	33%	33%	33%
	33%	66%	33%	33%	66%	66%	33%	33%	66%
Text 4	0%	57%	71%	57%	57%	57%	0%	71%	71%
	0%	71%	57%	29%	57%	57%	0%	71%	71%

Legende:

- **TL:** AlphabeticTokenizer & Lemmatisierung
- **TS:** AlphabeticTokenizer & Stoppworteliminierung
- **TSL:** AlphabeticTokenizer & Stoppworteliminierung & Lemmatisierung
- Der Suffix **A** bezeichnet die Semantik aus BabelNet.
- Der Suffix **B** bezeichnet die zusammengeführte Ontologie aus BabelNet und der Basisontologie.

Tabelle 6.1: Testergebnisse der Klassifikation

6.4.3 Erkenntnisse

In dem vorherigen Abschnitt wurden die Ergebnisse in einer Tabelle dargelegt. Bei einer genauen Analyse lassen sich einige Feststellungen treffen. Aus den Ergebnissen lassen sich folgende Rückschlüsse hinsichtlich der Semantik ziehen:

- Die Ergebnisse aus der Kategorisierung durch die Ontologie aus BabelNet und der Basisontologie sind besser als die Ergebnisse die nur aus BabelNet erzeugt werden. Daraus kann gefolgert werden, dass eine domänenspezifische Ontologie die Klassifizierung verbessert. Dies ist verständlich, da größtenteils domänenspezifische Begriffe in die Wertemenge der Kategorien aufgenommen wird. So ist die Wahrscheinlichkeit eine irrelevante Kategorie zu erzeugen niedrig.
- Der Preprocess Stoppworteliminierung ist wichtig, da für jedes Begriff eine BabelNet Abfrage aufgerufen wird. Wenn bedeutungslose Wörter nicht eliminiert werden, werden zur Wertemenge irrelevante Wörter hinzugefügt, welches die Wahrscheinlichkeit einer falschen Klassifizierung erhöht. Daraus lässt sich folgende Aussage ableiten: Je mehr irrelevante Wörter eliminiert werden, desto höher ist die Wahrscheinlichkeit einer korrekten Klassifizierung.

7 Zusammenfassung und Ausblick

Zum Abschluss wird in diesem Kapitel diese Ausarbeitung resümiert und bewertet. Dazu werden die wichtigsten Punkte zusammengefasst und ein Fazit geschlossen. Anschließend werden auf die Erweiterungsmöglichkeiten dieser Arbeit eingegangen.

7.1 Zusammenfassung

Die Bedeutsamkeit von Informationen wurde in dieser Arbeit des Öfteren appelliert. Insbesondere in dieser Generation ist sie im Vergleich zur früheren Zeit, in der in Büchern gestöbert werden musste um Informationen zu finden, durch das Internet griffbereit. Das Internet fasst eine gigantische Menge an Daten, was auch Nachteile mit sich zieht. In Anbetracht zur kompetitiven Produktentwicklung ist die Zeit ein wichtiger Faktor. Eine vollständig neue Planung und Entwicklung würde mehr Zeit in anspruch nehmen als eine vorhandene Lösung einzusetzen. Deshalb werden bereits existierende oder ähnliche Lösungen gesucht. Es ist langwierig in dieser Vielheit die relevanten Lösungen herauszufiltern. Deshalb wird eine Lösung gesucht, die die Semantik eines Textes schnell extrahiert. Die für diese Bachelorarbeit entwickelte Software sollte dieser Problematik entgegenkommen. Mit Unterstützung von NLP-Werkzeugen und semantischen Netzwerken soll ein Text klassifiziert werden. Somit kann bei der Lösungssuche Zeit gespart werden.

Die Qualität der Klassifizierung ist abhängig von den Vorverarbeitungsprozessen, die an den Rohtext angewendet werden. Vor allem die Klassifizierung nach der Stoppworteliminierung führt zu besseren Ergebnissen als Klassifizierungen ohne Stoppworteliminierung. Dies ist darauf zurückzuführen, dass für jeden im Text vorkommenden Begriff eine BabelNet Anfrage gemacht wird. Demzufolge können bei Anfragen bezüglich bedeutungslosen Begriffen irrelevante Kategorien in die Wertemenge gelangen. Es ist auch zu berücksichtigen, dass es bei sehr langen Dokumenten zu längeren Wartezeiten kommen kann, da die API-Zugriffe Zeit kosten. Um das zu verhindern wurde ein Cache integriert, sodass zu einem Begriff jeweils nur eine Abfrage durchgeführt wird. Trotzdem wird mit steigender Anzahl an verschiedenen Wörtern die Wartezeit verlängert. Außerdem ist zu erwähnen, dass in dieser Arbeit eine Basisontologie bezüglich der Automobi-Domäne integriert wurde. Bei einer Lösungssuche im Bereich einer anderen Domäne ist es vorteilhaft eine entsprechende Ontologie einzubauen.

7.2 Ausblick

Solange eine Software nicht hundertprozentig fehlerfreie Ergebnisse liefert ist sie verbesserungswürdig. Demnach ist auch diese Software ausbaufähig. Nachstehend werden einige Verbesserungsvorschläge bekannt gemacht.

Zumal diese Arbeit eine ontologiebasierte Lösung im Fokus hat wird an diesem Punkt angesetzt. Ein Leitgedanke ist die Ontologie funktionsbasiert darzustellen. Dabei wird näher auf die Wortart eingegangen. Eine Funktion wird geschildert durch die Verknüpfung von einem Objekt und einem Verb. Sie verfasst die Eigenschaftsänderung des Produktes, die durch einen Funktionsträger veranlasst wird. Der Funktionsträger ist das Konstruktionsobjekt. Eine Funktion kann ausßerdem als Beschreibung eines technischen Systems über deren Wirkung gesehen werden. Infolgedessen werden technische Systeme auch als Funktionsträger betrachtet[40].

Durch funktionsorientierte Betrachtung der Sätze kann ein Text ausführlich analysiert werden. Unter Sichtweise kann beispielsweise eine Auswahl zwischen den Begriffen gemacht werden, für die eine BabelNet Anfrage gemacht werden soll. Es können präzisere Untersuchungen durchgeführt werden, indem je nach gesuchtem Verb oder Objekt eine dementsprechende Abfrage gemacht wird. Somit verringert sich auch die Anzahl und damit die Zeit für die Anfragen.

Literaturverzeichnis

[1] H. Stuckenschmidt, H. Stuckenschmidt, and H. Stuckenschmidt, *Ontologien: Konzepte, Technologien und Anwendungen.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. (Zitiert auf Seite 9)

[2] A. F. Gaag, *Entwicklung einer Ontologie zur funktionsorientierten Lösungssuche in der Produktentwicklung.* Dissertation, Technische Universität München, München, 2010. (Zitiert auf den Seiten 7, 9, 11, 12, 17, 19, 21, 22, 24, 26, 27, 28, 32 und 35)

[3] U. Lindemann, *Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden.* (Zitiert auf den Seiten 7, 11, 12 und 14)

[4] R. Haberfellner, *Systems engineering: Methodik und Praxis.* Zürich: Verl. Industrielle Organisation, 11. durchges. aufl. ed., 2002.

[5] J. Feldhusen, ed., *Pahl/Beitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung.* Berlin; Heidelberg: Springer Vieweg, 8., vollst. überarb. aufl. ed., 2013. (Zitiert auf Seite 11)

[6] K. Ehrlenspiel and H. Meerkamm, *Integrierte Produktentwicklung: Denkabläufe, Methodeneinsatz, Zusammenarbeit.* München: Hanser Verlag, 5., überarbeitete und erweiterte auflage ed., 2013. (Zitiert auf den Seiten 7, 12 und 13)

[7] G. Ropohl, "Allgemeine technologie: eine systemtheorie der technik," hochschulschrift, München, 1999. (Zitiert auf Seite 14)

[8] G. Ropohl, *Allgemeine Systemtheorie: Einführung in transdisziplinäres Denken.* Berlin: Ed. Sigma, 2012. (Zitiert auf Seite 14)

[9] J. Ponn and U. Lindemann, *Konzeptentwicklung und Gestaltung technischer Produkte: Systematisch von Anforderungen zu Konzepten und Gestaltlösungen.* VDI-Buch, Springer Berlin Heidelberg, 2011. (Zitiert auf den Seiten 7, 15, 17 und 29)

[10] J. Gramann, *Problemmodelle und Bionik als Methode.* Dissertation, Technische Universität München, München, 2004. (Zitiert auf Seite 17)

[11] W. Boysen, *Interorganisationale Geschäftsprozesse in virtuellen Marktplätzen: Chancen und Grenzen für das B-to-B-Geschäft.* Deutscher Universitätsverlag, 2013. (Zitiert auf Seite 17)

[12] A. Gaag and J. Ponn, "„suchen und finden" im maschinen- und anlagenbau - eine studie in zusammenarbeit mit dem vdma," *CiDaD Working Paper Series*, no. 4, 2008. (Zitiert auf Seite 18)

[13] G. Probst, S. Raub, and K. Romhardt, *Wissen managen: Wie Unternehmen ihre wertvollste Ressource optimal nutzen.* Gabler Verlag, 2013. (Zitiert auf Seite 21)

[14] W. Sturz, *Wissensmanagement- Mehr als Informationsmanagement.* Dissertation, Institut für Management und Kommunikation Dr. Sturz. (Zitiert auf Seite 22)

[15] M. Haun, *Handbuch Wissensmanagement- Grundlagen und Umsetzung, Systeme und Praxisbeispiele.* Springer-Verlag Berlin Heidelberg, 2002. (Zitiert auf Seite 22)

[16] A. Henrich, *Information Retrieval 1- Grundlagen, Modelle und Anwendungen.* Otto-Friedrich-Universität Bamberg, 2008. (Zitiert auf den Seiten 23 und 25)

[17] D. Lewandowski, *Web Information Retrieval - Technologien zur Informationssuche im Internet.* Düsseldorf: Deutsche Gesellschaft für Informationswissenschaft und Informationspraxis e.V, 2005. (Zitiert auf Seite 25)

[18] T. Pellegrini and A. B. (Hrsg.), *Semantic Web - Wege zur vernetzten Wissensgesellschaft.* Springer-Verlag Berlin Heidelberg, 2006. (Zitiert auf den Seiten 25 und 26)

[19] H. Stachowiak, *Allgemeine Modelltheorie.* Springer Verlag, 1973. (Zitiert auf Seite 26)

[20] P. Jaeger and U. Linck, *Wissensmanagement heute- zwischen semantischen Suchtechnologien und Schwarmintelligenz.* Projektarbeit, Hochschule Ravensburg- Weingarten, 2009. (Zitiert auf den Seiten 7, 26, 27 und 31)

[21] T. R. Gruber, *Toward Principles for the Design of Ontologies Used for Knowledge Sharing.* PhD thesis, Stanford Knowledge Systems Laboratory, 1993. (Zitiert auf Seite 27)

[22] A. Gilchrist, "Thesauri, taxonomies and ontologies – an etymological note," *Journal of Documentation*, vol. 59, no. 1, pp. 7–18, 2003. (Zitiert auf Seite 27)

[23] E. Moench, M. Ullrich, H.-P. Schnurr, and J. Angele, "Semanticminer - ontology-based knowledge retrieval," 2003. (Zitiert auf Seite 27)

[24] S. Seifert, S. Zillner, M. Huber, M. Sintek, D. Sonntag, and A. Cavallaro, "Medico," in *Internet der Dienste* (L. Heuser and W. Wahlster, eds.), vol. 0 of *acatech DISKUTIERT*, pp. 97–111, Springer Berlin Heidelberg, 2011. (Zitiert auf Seite 27)

[25] R. Stevens, C. A. Goble, and S. Bechhofer, "Ontology-based knowledge representation for bioinformatics," *Briefings in bioinformatics*, vol. 1, no. 4, pp. 398–414, 2000. (Zitiert auf Seite 27)

[26] T. BERNERS-LEE, J. HENDLER, and O. LASSILA, "The semantic web," *Scientific American: Feature Article*, 2001. (Zitiert auf Seite 27)

[27] R. Volz, *Web Ontology Reasoning with Logic Databases.* Dissertation, Universität Fridericiana zu Karlsruhe, Karlsruhe, 2004. (Zitiert auf den Seiten 7 und 28)

[28] S. Hami-Nobari and L. Blessing, "Effect-oriented description of variant rich products," *International Conference on Engineering Design*, 2005. (Zitiert auf Seite 28)

[29] N. Lohse, G. Valtchanov, S. Ratchev, M. Onori, and J. Barata, "Towards a unified assembly system design ontology using protégé," 2005. (Zitiert auf Seite 28)

[30] S. Ahmed, S. Kim, and K. Wallace, "A methodology for creating ontologies for engineering design," *Journal of Computing and Information Science in Engineering*, vol. 7, no. 2, pp. 132–140, 2007. (Zitiert auf den Seiten 7, 28, 32 und 33)

[31] M. F. Lopez, A. Gomez-Perez, J. P. Sierra, and A. P. Sierra, "Building a chemical ontology using methontology and the ontology design environment," *IEEE Intelligent Systems*, vol. 14, no. 1, pp. 37–46, 1999. (Zitiert auf Seite 31)

[32] Y. Sure, S. Staab, and R. Studer, "On-to-knowledge methodology (otkm)," in *Handbook on Ontologies* (S. Staab and R. Studer, eds.), International Handbooks on Information Systems, pp. 117–132, Springer Berlin Heidelberg, 2004. (Zitiert auf den Seiten 7, 31 und 32)

[33] C. Parent and S. Spaccapietra, "An overview of modularity," in *Modular Ontologies* (H. Stuckenschmidt, C. Parent, and S. Spaccapietra, eds.), vol. 5445 of *Lecture Notes in Computer Science*, pp. 5–23, Springer Berlin Heidelberg, 2009. (Zitiert auf Seite 32)

[34] S. Zelewski, "Wissensmanagement mit ontologien," (Essen), 2002. (Zitiert auf Seite 37)

[35] S.-H. Wu, T.-H. Tsai, and W.-L. Hsu, "Text categorization using automatically acquired domain ontology," in *Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages - Volume 11*, AsianIR '03, (Stroudsburg, PA, USA), pp. 138–145, Association for Computational Linguistics, 2003. (Zitiert auf den Seiten 7, 37 und 38)

[36] A. Maedche and S. Staab, "Ontology learning for the semantic web," *Intelligent Systems, IEEE*, vol. 16, pp. 72–79, Mar 2001. (Zitiert auf Seite 40)

[37] L. Ning, L. GuanYu, and Z. Yuanfa, "Research on domain ontology semi-automatic construction model towards chinese text," in *Educational and Network Technology (ICENT), 2010 International Conference on*, pp. 334–337, June 2010. (Zitiert auf den Seiten 7, 40 und 41)

[38] F. Zwirn, *Analyse und Auswertung von gewichteten Anforderungen in technischen Spezifikationen.* Dissertation, Universität Stuttgart, Stuttgart, 2013. (Zitiert auf Seite 45)

[39] Kheriddine'Oueslati', *Erkennung und Klassifikation nichtfunktionaler Anforderungen in einem Entwicklungsprozess unter Machine Learning.* Dissertation, Universität Stuttgart, Stuttgart, 2015. (Zitiert auf Seite 47)

[40] E. Zwicker, *Unterstützung der unternehmensübergreifenden Produktentwicklung durch den Einsatz moderner Informationstechnologien.* Fortschritt-Berichte VDI / 20, VDI-Verlag, 1999. (Zitiert auf Seite 58)

Alle URLs wurden zuletzt am 09. 03. 2016 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift