

Ein Assistenzsystem zur kontextsensitiven Unterstützung des anforderungsbasierten Wissensmanagements

zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften (Dr.-Ing.)

Von der KIT-Fakultät für Maschinenbau des
Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Simon Fritz

aus Karlsruhe

Tag der mündlichen Prüfung: 03.02.2021

Erster Gutachter: Prof. Dr. Dr.-Ing. Dr. h.c. Jivka Ovtcharova
Zweiter Gutachter: Prof. Dr.-Ing Hendro Wicaksono

Kurzfassung

Das Anforderungs- und Wissensmanagement wird in kleinen und mittelständischen Unternehmen häufig in seiner Wichtigkeit unterschätzt. Vor allem das systematische Dokumentieren und Verwalten von Anforderungen, kann über Erfolg oder Misserfolg eines Projektes entscheiden. Werden Anforderungen nicht vollständig dokumentiert oder nicht an die relevanten Personen kommuniziert, kann es zu Abweichungen gegenüber dem ursprünglichen Kundenwunsch kommen, was im schlimmsten Fall zum Scheitern des Projekts führen kann.

Stand heute werden überwiegend Office-basierte Softwarelösungen für die Dokumentation und Verwaltung von Informationen, Anforderungen und Aufgaben verwendet. Diese führen jedoch aufgrund ihres filebasierten Ansatzes zu häufigen Redundanzen oder unterschiedlichen Versionsständen, was zu Fehlern führt. Spezialisierte Softwarelösungen für das Verwalten von Anforderungen hingegen sind meist für Experten entwickelt worden und erfordern daher Schulungen und/oder neue Prozesse. Das schreckt viele Unternehmer davor ab derartige Lösungen in ihr Unternehmen zu integrieren.

Die vorliegende Arbeit nimmt sich daher zum Ziel, ein Assistenzsystem zur kontextsensitiven Unterstützung eines anforderungs-basierten Wissensmanagements zu entwickeln, welches vor allem kleine und mittelständische Unternehmen befähigen soll, vollständige und standardisierte Anforderungs- und Wissensdokumentationen ohne aufwändige Schulungen oder Prozesse zu erstellen und zu verwalten. Die entwickelten Konzepte werden anhand einer prototypischen Implementierung evaluiert und die generelle Machbarkeit bewiesen.

Abstract

The importance of requirements and knowledge management in small and medium-sized companies is often underestimated. Especially the systematic documentation and administration of requirements can decide on the success or failure of a project. If requirements are not completely documented or not communicated to the relevant persons, this can lead to deviations from the original customer request, which in the worst case can lead to the failure of the project.

Today, mainly office-based software solutions are used for the documentation and administration of information, requirements and tasks. However, due to their file-based approach, these lead to frequent redundancies or different version statuses, which leads to errors. Specialized software solutions for the management of requirements, on the other hand, are usually developed for experts and therefore require training and new processes, which deters many companies from integrating such solutions into their business.

This thesis aims to develop an assistance system for context-sensitive support of a requirements-based knowledge management, which should enable small and medium-sized companies to create and manage complete and standardized requirements and knowledge documentation without extensive training or processes. The developed concepts are evaluated using a prototypical implementation and the general feasibility is proven.

Danksagung

An den Vorarbeiten zu dieser Dissertation waren viele Menschen beteiligt, bei denen ich mich an dieser Stelle herzlich bedanken möchte.

Mein besonderer Dank gilt Frau Prof. Jivka Ovtcharova für Ihr Interesse und ihre Unterstützung bei der Verwirklichung meines initialen Forschungswunsches.

Ebenso sehr möchte ich mich bei Herrn Prof. Hendro Wicaksono, für die Zweitbetreuung meiner Arbeit bedanken.

Ein herzliches Dankeschön gilt den Projektpartnern Schuster Elektronik GmbH, NETSYNO Software GmbH und der iCONDU GmbH des durch das BMBF geförderte KMU-Innovativ Projekt DAM4KMU, welche mir in hitzigen und sehr konstruktiven Diskussionen wichtigen Input für diese Arbeit lieferten und zu einer möglichen Realisierung in einem realen Produkt beitragen.

Bedanken möchte ich mich auch bei meinen Eltern, welche mich zu der Person gemacht haben die ich heute bin und mich bis heute unterstützen und für mich da sind. Ich bin sehr dankbar, dass ich euch habe und dafür, dass ihr so tolle Eltern seid! Danke!

Vor allem jedoch möchte ich mich bei meiner geliebten Lebensgefährtin bedanken, welche mir trotz eigener Promotion und der Geburt unserer gemeinsamen Tochter den nötigen Freiraum zum Schreiben und die Unterstützung gab, um diese Arbeit anzufertigen.

Karlsruhe, Februar 2021

Simon Fritz

Inhaltsverzeichnis

Kurzfassung	ii
Abstract	v
Danksagung	vii
Inhaltsverzeichnis.....	x
Abbildungsverzeichnis	xv
Tabellenverzeichnis	xix
Abkürzungsverzeichnis.....	xxi
1 Einleitung	25
1.1 Problemstellung	26
1.2 Zielsetzung	29
1.3 Rahmen der Arbeit	30
1.4 Aufbau der Arbeit	32
2 Grundlagen	35
2.1 Der Produktlebenszyklus	35
2.1.1 Der Produktentstehungsprozess	36
2.1.2 Klassische Vorgehensmodelle	37
2.1.3 Agile Vorgehensmodelle	40
2.2 Das Anforderungsmanagement	43
2.2.1 Anforderungsermittlung	45
2.2.2 Anforderungsdokumentation	46
2.2.3 Anforderungsprüfung.....	49
2.2.4 Anforderungsverwaltung	50
2.2.5 Anforderungsmanagement Werkzeuge.....	53
2.3 Maschinelle Lernverfahren	55
2.3.1 Markov-Ketten.....	55
2.3.2 Convolutional Neural Network (CNN).....	56
2.4 Computerlinguistische Grundlagen	56
2.4.1 Zuordnung von Wortarten	57

2.4.2	Dependency Parsing	57
2.4.3	Eigennamenerkennung.....	58
2.4.4	Koreferenz-Auflösung	59
2.4.5	Wortvektoren	60
2.4.6	Vortrainierte Sprachmodelle	60
2.4.7	NLP-Frameworks.....	61
2.5	Fazit	62
3	Stand der Technik.....	63
3.1	Ableitung der Problemfelder	63
3.2	Unterstützung der Dokumentation	64
3.3	Unterstützung der Wissensintegration	66
3.4	Unterstützung der Wissensverknüpfung	68
3.5	Unterstützung der Änderungsdurchführung	70
3.6	Unterstützung bei der Wiederverwendung von bereits dokumentiertem Wissen	71
3.7	Unterstützung von Recherchearbeiten	72
3.8	Ableitung von Anforderungen	73
3.9	Fazit	75
4	Konzept.....	79
4.1	Semantisches Datenmodell	81
4.1.1	Kontextelemente	83
4.1.2	Projekt.....	84
4.1.3	Text-Elemente.....	85
4.1.4	Satzschablonen.....	86
4.1.5	Systemkomponenten	87
4.1.6	Versionierung	89
4.1.7	Zusammenführen zweier Kontextelemente.....	91
4.2	Dokumentationsassistent	93
4.2.1	Semantische Rollenerkennung	94
4.2.2	Textklassifikation.....	96
4.2.3	Autovervollständigung	96
4.2.4	Kontextsensitive Hilfestellungen	97
4.2.5	Verknüpfen und speichern	101
4.3	Integrationsassistent	103

4.3.1	Dokumentenstrukturanalyse	104
4.3.2	Koreferenz-Auflösung	105
4.3.3	Satzanalyse	106
4.3.4	Informationen prüfen und freigeben	106
4.4	Rechercheassistent	109
4.4.1	Kontextsensitive Suche durchführen	110
4.4.2	Informationen extrahieren.....	110
4.4.3	Informationen prüfen und freigeben	111
4.5	Verknüpfungsassistent	111
4.5.1	Kontextsensitive Suche durchführen	112
4.5.2	Suchergebnisse analysieren	113
4.5.3	Informationen prüfen und freigeben	116
4.6	Wiederverwendungsassistent	116
4.6.1	Suchergebnisse analysieren	117
4.6.2	Informationen prüfen und freigeben	117
4.7	Änderungsassistent	119
4.7.1	Klassifikation der Änderung	120
4.7.2	Abschätzung des Einflusses	126
4.7.3	Informationen prüfen und freigeben	129
4.8	Fazit	132
5	Implementierung	133
5.1	Dokumentationsassistent	134
5.1.1	Semantische Rollenerkennung.....	134
5.1.2	Textklassifikation	137
5.1.3	Autovervollständigung.....	138
5.1.4	Kontextsensitive Hilfestellungen	139
5.2	Integrationsassistent	140
5.2.1	Dokumentenstrukturanalyse	141
5.2.2	Koreferenz-Auflösung	141
5.2.3	Satzanalyse	143
5.2.4	Informationen prüfen und freigeben	143
5.3	Rechercheassistent	145
5.3.1	Kontextsensitive Suche durchführen	146
5.3.2	Informationen extrahieren.....	146

5.3.3	Informationen prüfen und freigeben	147
5.4	Verknüpfungsassistent	147
5.4.1	Kontextsensitive Suche durchführen	148
5.4.2	Suchergebnisse analysieren	148
5.4.3	Informationen Prüfen und freigeben	150
5.5	Wiederverwendungsassistent	150
5.5.1	Suchergebnisse analysieren	150
5.5.2	Informationen prüfen und freigeben	151
5.6	Änderungsassistent	152
5.6.1	Informationen prüfen und freigeben	153
5.7	Fazit	155
6	Evaluation und Diskussion.....	157
6.1	Evaluationsaufbau	157
6.1.1	Trainings- und Testdatensätze	157
6.1.2	Bibliotheksversionen	158
6.2	Evaluation der prototypischen Implementierung	159
6.2.1	Dokumentationsassistent.....	159
6.2.2	Integrationsassistent.....	161
6.2.3	Rechercheassistent	162
6.2.4	Verknüpfungsassistent	165
6.2.5	Wiederverwendungsassistent	167
6.2.6	Änderungsassistent	167
6.3	Abgleich der Anforderungen	168
6.4	Fazit	171
7	Zusammenfassung und Ausblick.....	175
7.1	Zusammenfassung	175
7.2	Ausblick	177
7.3	Summary	178
8	Literaturverzeichnis	181
9	Anhang.....	195
9.1	Satzschablonen	195
9.2	Logische Operatoren	197

Abbildungsverzeichnis

Abbildung 1.1: DAM4KMU-Framework.....	31
Abbildung 1.2: Aufbau der vorliegenden Arbeit.....	33
Abbildung 2.1: Produktlebenszyklusphasen [Eig 13, S. 9, Reu 14, S. 7].....	35
Abbildung 2.2: Der Produktentstehungsprozess [Eig 13, S. 9].....	36
Abbildung 2.3: Beispiel eines Wasserfallmodells [Mas 12, S. 675].....	38
Abbildung 2.4: Darstellung des V-Modells [Gra 14, S. 112].....	39
Abbildung 2.5: Das integrierte Produktentstehungsmodell [Zin 13, S. 96]... ..	42
Abbildung 2.6: Funktionale Anforderungsschablone [Poh 15, S. 64].....	48
Abbildung 2.7: 10er Regel der Fehlerkosten [angelehnt an Ehr 09, S. 170] ..	49
Abbildung 2.8: Versionierungsschema [Poh 09, S. 141].....	52
Abbildung 4.1: Module des Konzepts.....	80
Abbildung 4.2: Semantisches Datenmodell.....	82
Abbildung 4.3: Satzschemadatenmodell.....	87
Abbildung 4.4: Datenmodell Eigenschaftswert.....	88
Abbildung 4.5: Logisch verkettete Eigenschaftswerte.....	89
Abbildung 4.6: Historie eines Kontextelements.....	90
Abbildung 4.7: In Verbindung stehende Kontextelemente.....	90
Abbildung 4.8: Zusammenfassung zweier Kontextelemente.....	92
Abbildung 4.9: Aufbau des Dokumentationsassistenten.....	93
Abbildung 4.10: Computerlinguistisch aufbereiteter Satz.....	94
Abbildung 4.11: Exemplarische Zuordnung semantischer Rollen.....	95
Abbildung 4.12: Beispiel der Identifikation von Wortvorschlägen.....	97
Abbildung 4.13: Kontextsensitive Hilfestellungen.....	98
Abbildung 4.14: Bereitstellung eines Referenzbeispiels.....	99
Abbildung 4.15: Bereitstellung von Hilfsfragen.....	100

Abbildung 4.16: Informationsextraktion aus Beschreibungstexten	102
Abbildung 4.17: Kontextelemente und deren Relationen.....	102
Abbildung 4.18: Funktionaler Aufbau der Relationserkennung	104
Abbildung 4.19: Beispiel einer satzübergreifenden Koreferenz.....	105
Abbildung 4.20: Mockup des Prüfdialogs	107
Abbildung 4.21: Mockup des Integrationsdialogs.....	108
Abbildung 4.22: Funktionaler Aufbau des Rechercheassistenten	109
Abbildung 4.23: Aufbau des Verknüpfungsassistenten.....	112
Abbildung 4.24: Beispiel für eine Nicht-Redundanz	113
Abbildung 4.25: Semantisch konfligierende Anforderungen	115
Abbildung 4.26: Aufbau des Wiederverwendungsassistenten	117
Abbildung 4.27: Mockup des Wiederverwendungsassistenten	118
Abbildung 4.28: Funktionaler Aufbau des Änderungsassistent	120
Abbildung 4.29: Beispiel einer Änderung	124
Abbildung 4.30: Einflussprüfungsdialog.....	130
Abbildung 4.31: Aufwandsabschätzungsdialog	131
Abbildung 4.32: Satzschablone einer Änderungsaufgabe	131
Abbildung 5.1: Aufbau des digitalen Assistenzsystems.....	133
Abbildung 5.2: Aufbau des Dokumentationsassistenten	134
Abbildung 5.3: Beispielanforderungen	136
Abbildung 5.4: Screenshot der Autovervollständigung.....	138
Abbildung 5.5: Screenshot der kontextsensitiven Hilfestellungen	139
Abbildung 5.6: Funktionaler Aufbau des Integrationsassistenten	140
Abbildung 5.7: Beispiel einer Koreferenz	141
Abbildung 5.8: Screenshot der Synonym-Integration	142
Abbildung 5.9: Mockup des Prüfdialogs.....	143
Abbildung 5.10: Screenshot des Integrationsdialogs.....	144
Abbildung 5.11: Funktionaler Aufbau des Rechercheassistenten	145

Abbildung 5.12: Funktionaler Aufbau des Verknüpfungsassistenten 148

Abbildung 5.13: Aufbau des Wiederverwendungsassistenten 150

Abbildung 5.14: Screenshot des Wiederverwendungsassistenten 151

Abbildung 5.15: Funktionaler Aufbau des Änderungsassistent 152

Abbildung 5.16: Mockup des Einflussprüfungsdialogs..... 153

Abbildung 5.17: Mockup des Aufwandsabschätzungsdialog 154

Tabellenverzeichnis

Tabelle 1: Prioritätsdefinition [Poh 15, S. 128].....	51
Tabelle 2: Kontextelementrelationen (OMG 15, OMG 18)	84
Tabelle 3: Text-Element-Relationen [OMG 15, OMG 18]	85
Tabelle 4: Systemkomponenten spezifische Relationen [OMG 18].....	87
Tabelle 5: Linguistische Regeln (angelehnt an [SOP 13])	96
Tabelle 6: Aktionsmöglichkeiten je zu prüfendes Merkmal.....	108
Tabelle 7: Definition der Kritikalität je Änderungsart	126
Tabelle 8: Definition der Erkennungsverfahren	136
Tabelle 9: Beispiel Suchanfragen	146
Tabelle 10: Gewichtung je Status.....	152
Tabelle 11: Gewichtung je Priorität	153
Tabelle 12: Annotierte Trainingsdaten.....	157
Tabelle 13: Annotierte Satzpaare für die NLI-Evaluation.....	158
Tabelle 14: Bibliotheken und deren Version	158
Tabelle 15: Evaluationsergebnisse der Semantischen Rollenerkennung.....	160
Tabelle 16: Evaluationsergebnisse der Mentions-Erkennung	162
Tabelle 17: Beispielszenarien für die Evaluation des Rechercheassistenten.....	164
Tabelle 18: Evaluationsergebnisse der Beispielszenarien	164
Tabelle 19: Evaluationsergebnisse der NLI Klassifikation	166
Tabelle 20: Beispielsätze der NLI-Klassifikationsaufgabe	167
Tabelle 21: Anforderungs-Satzschablonen und deren Objektreferenzen	197

Abkürzungsverzeichnis

Abs.	Absatz
AM	Anforderungsmanagement
BERT	Bidirektionale Encoder-Darstellung von Transformatoren
BZO	Bezugsobjekt
CNN	Convolutional Neural Network
DAM4KMU	Digitaler Assistent für das Anforderungsmanagement für KMU
e.g.	lateinisch für „exempli gratia“ (zu Deutsch „zum Beispiel“)
Dep	Dependency
ggf.	gegebenenfalls
GPM	Deutschen Gesellschaft für Projektmanagement e.V.
ID	Identifikationsnummer
IEEE	Institute of Electrical and Electronics Engineers
IPEK	Institut für Produktentwicklung
iPeM	integrierte Produktentstehungs-Modell
IREB	International Requirements Engineering Board
k.A.	keine Angabe(n)
KEA	Kontext-Extraktions-Assistent
KIT	Karlsruher Instituts für Technologie
KMU	Kleine und mittelständische Unternehmen
LSTM	Long short-term memory

NER	Named Entity Recognition
NLI	Natural Language Interference
NLP	Natural Language Processing
Nr.	Nummer
OCR	engl. optical character recognition
PEP	Produktentstehungsprozess
PLZ	Produktlebenszyklus
POS	Part-of-speech
RE	Requirements Engineering
ReqIF	Requirements Interchange Format
RIF	Rule Interchange Format
RM	Requirements Management
RNN	Rückgekoppelte neuronale Netze
SME	small and medium-sized enterprises (zu Deutsch „KMU“)
SVM	Support Vektor Maschinen
SysML	Systems Modeling Language
UML	Unified Modeling Language
VDMA	Verbands Deutscher Maschinen- und Anlagenbau
WM	Wissensmanagement
z.B.	zum Beispiel

1 Einleitung

*„Das Problem zu erkennen ist wichtiger als die Lösung zu erkennen,
denn die genaue Darstellung des Problems führt zur Lösung.“*

Albert Einstein

Wie schon Albert Einstein und sicher vielen vor ihm bereits bewusst war, ist zur Erreichung eines Ziels nicht die Lösung selbst, sondern der Weg dorthin entscheidend. Projekte scheitern oft durch unrealistische Kalkulationen, zu straffe Zeitpläne oder fehlenden Kundenbezug. Eines der größten Probleme jedoch ist meist der fehlende Informationsaustausch zwischen den betroffenen Personen und das unsystematische Dokumentieren und Managen von projekt-relevanten Informationen wie z.B. Anforderungen, Aufgaben oder Rechercheergebnissen. Vor allem Entwickler verlieren sich gerne in einer Vielzahl an Features und technischen Raffinessen, statt für das eigentliche Grundproblem des Kunden eine einfache und effiziente Lösung zu finden [The 94, The 14, The 15, Ada 13, Ada 15].

Anforderungen bilden die Basis eines jeden Entwicklungsprojektes. Sie formulieren die Erwartungen und Forderungen aller betreffenden Stakeholder an das zu entwickelnde Produkt und deren Erfüllung entscheidet über Erfolg oder Misserfolg des Entwicklungsprojekts. Stand heute werden im Mittelstand – „der Motor der deutschen Wirtschaft“¹ – und insbesondere im Maschinenbau vorrangig Office-Produkte eingesetzt, um Anforderungen und das damit in Verbindung stehende Wissen zu dokumentieren und zu verwalten. Dies birgt jedoch große Nachteile im Vergleich zu spezialisierten Werkzeugen, da

¹ <https://www.kfw.de/KfW-Konzern/KfW-Research/Mittelstand.html>, zuletzt geprüft am 20.08.2020

Office-Lösungen weder eine agile und kollaborative Zusammenarbeit unterstützen noch ein systematisches Änderungsmanagement oder die Analyse der damit verbundenen Auswirkungen unterstützen [Gro 13].

Spezialisierte Softwarelösungen auf der anderen Seite erfordern meist umfangreiche Schulungen, sind teuer in der Anschaffung und benötigen viel Zeit zur Pflege und Verwaltung der Daten. Hürden die vor allem kleine- und mittelständische Unternehmen (KMU) davon abhalten die entsprechenden Anforderungsmanagementprozesse und die dafür notwendigen Softwarelösungen zu etablieren [Gro 13, Ada 13, Ada 15].

Aber nicht nur der Mittelstand ist mit den angebotenen Lösungen für das Anforderungsmanagement unzufrieden. Auch große Unternehmen haben mit der fehlenden Prozessintegration des Anforderungsmanagements zu kämpfen [Ada 13, Ada 15]. Gerade zu Zeiten der vierten industriellen Revolution und Globalisierung, wird die deutsche Wirtschaft mit stetig wachsender Konkurrenz konfrontiert. Kürzere Produktlebenszyklen und steigende -komplexitäten machen effiziente Anforderungsmanagementprozesse unabdingbar.

Um dies leisten zu können braucht es neue Ansätze und Methoden im Anforderungsmanagement, um vor allem KMU langfristig zu befähigen dem internationalen Konkurrenzdruck Stand zu halten. Basierend auf dieser Forderung nimmt sich die vorliegende Arbeit zum Ziel, mit Hilfe des aktuellen Stands der Technik neue Methoden und Konzepte zu entwickeln, um ein Assistenzsystem zur kontextsensitiven Unterstützung des anforderungsbasierten Wissensmanagements zu entwickeln und somit das Anforderungsmanagement stärker in die Unternehmensprozesse zu integrieren.

1.1 Problemstellung

Das Anforderungsmanagement (AM) und seine Methoden werden häufig in ihrer Wichtigkeit unterschätzt, was dazu führt, dass Anforderungen zu ungenau oder gar nicht dokumentiert, betroffene Stakeholder nicht involviert oder relevante Phasen des Produktlebenszyklus außer Acht gelassen werden. Die auf Basis dieser Anforderungen entwickelten Produkte verfehlen nicht selten

den eigentlichen Kundennutzen und führen so im schlimmsten Fall zum vollständigen Scheitern des Entwicklungsprojektes. Im Rahmen der Chaos-Reporte [The 94, The 14, The 15] wurden bis 2015 ca. 50.000 Software-Projekte hinsichtlich ihres Erfolgs oder Misserfolgs untersucht. Als Hauptursachen für das Scheitern der Projekte wurden hierbei vor allem unvollständige Anforderungen, Probleme beim Management von Anforderungsänderungen und die fehlende Integration der Anwender identifiziert.

Ein weiteres Problem stellen die Werkzeuge dar, die zum Management von Anforderungen und dem damit in Verbindung stehenden Wissen eingesetzt werden. In vielen Unternehmen – vor allem bei KMU – kommen ausschließlich Office-Produkte (Excel, Word) und E-Mails zum Einsatz, welche im Gegensatz zu speziellen Softwarelösungen (z.B. IBM Rational DOORS und PTC Integrity) zwar das Sammeln und Clustern der Anforderungen und Wissen ermöglichen, aber einige erhebliche Nachteile mit sich bringen können:

- Der Erhebungs- und Managementprozess wird nicht aktiv unterstützt.
- Der Änderungsverlauf ist nicht adäquat nachvollziehbar.
- Der manuelle Austausch der Dokumente ist träge und führt zu Redundanzen.
- Die Analyse der Abhängigkeiten zwischen mehreren Anforderungen sowie die Auswirkungen einer Änderung sind durch diese Art der Dokumentation nicht möglich.
- Die Wiederverwendung bereits dokumentierter Informationen wird nicht kontextsensitiv unterstützt.

Im Rahmen einer Umfrage der Technischen Universität Bratislava [Gro 13] wurden 95 Unternehmen im Bereich Maschinen- und Anlagenbau über das Zusammenspiel von Anforderungsmanagement, Projektabwicklung und Kundenintegration bei KMU befragt. Hintergrund der Befragung waren die Tatsachen, dass der Maschinen- und Anlagenbau in den meisten Fällen durch vergleichsweise einfache Projektmanagement-Methoden geleitet wird und Anforderungen und projektrelevante Informationen überwiegend mittels Office Produkten (Excel, Word, PowerPoint) und E-Mails gemanagt werden.

Als Gründe hierfür werden im Management Summary der Umfrage unter anderem der Mangel an qualifizierten Ressourcen, der damit verbundene Mehraufwand und die hohen Kosten der entsprechenden Softwarelösungen genannt [Gro 13].

Ein weiterer wichtiger Grund für die Nichtnutzung spezialisierter Softwaretools sind die Tools selbst. Im Rahmen einer Umfrage des Fraunhofer Instituts IESE [Ada 13] wurden etwa 300 kleine und große Unternehmen unterschiedlicher Branchen zum Thema Anforderungsmanagement befragt. Hierbei stellten sich folgende Probleme mit aktuellen Softwarelösungen heraus, welche 2015 durch eine erneute Befragung von mehr als 200 Unternehmen untermauert wurden [Ada 15]:

- Keine aktive Unterstützung bei AM-Aufgaben
- Schlechte / schwierige Bedienung aus Sicht der Anwender
- Schlechte Handhabung von Anforderungsänderungen
- Geringe Akzeptanz bei Nicht-AM-Experten
- Insellösungen mit geringer bis keine Integration in bestehende Prozesse
- Mangelnde Motivation für die Einführung und Einhaltung von AM
- Fehlende Möglichkeit der Wiederverwendung und Standardisierung von Anforderungen
- Hohe Lizenzgebühren

Dies macht deutlich, dass aktuelle Softwarelösungen den Anforderungen kleiner und mittelständischer Unternehmen nicht gerecht werden. Sie benötigen Lösungen, welche die Anwender aktiv bei der Erfüllung der anfallenden Aufgaben im Anforderungsmanagement unterstützen, intuitiv und einfach zu bedienen sind und besser in bereits bestehende Prozesse integriert sind. Nur so können KMU, trotz ihres personellen und finanziellen Ressourcenmangels, Anforderungen und das damit verbundene Wissen sinnvoll und in der entsprechenden Qualität managen.

Aber auch Unternehmen, welche etablierte Anforderungsmanagement-Methoden und -Werkzeuge einsetzen, sehen weiterhin Handlungsbedarfe. Zum einen

stellt die Standardisierung und Wiederverwendung von Anforderungen und digitalisiertem Wissen, trotz des Einsatzes von Experten-Tools eine große Herausforderung dar [Ada 15]. Zum anderen bedarf es neuer Lösungsansätze, um alle Projektmitarbeiter über alle relevanten Änderungen zu informieren, ohne diese mit der ggf. aufkommenden Informationsflut zu überlasten. Dies erfordert vor allem ein Umdenken in der Art der Anwendung des Anforderungs- und Wissensmanagements, da dieses häufig von wenigen Experten getrieben wird und nur selten von allen Projektbeteiligten gelebt wird.

1.2 Zielsetzung

Den zuvor geschilderten Herausforderungen stellt sich die vorliegende Arbeit mit dem Kernziel eine kontextsensitive Unterstützung des anforderungsbasierenden Wissensmanagements und den damit verbundenen Aufgaben für Unternehmen zu entwickeln. Hierbei werden drei Kernfragestellungen adressiert welche nachfolgend motiviert werden.

Die Integration der Methoden des Anforderungsmanagements in die Unternehmensprozesse ist bis heute eine große Herausforderung, welche von vielen Unternehmen häufig nur teilweise oder gar nicht durchgeführt wird. Dies führt dazu, dass Anforderungen nicht systematisch dokumentiert, ausreichend an alle betroffenen Stakeholder kommuniziert und Änderungen nicht systematisch umgesetzt werden. Vor allem die fehlende Betrachtung der Tragweite von Änderungen führt häufig zu unerwarteten Mehrkosten und/oder zeitlichen Verzögerungen sowie in extremen Fällen zum Scheitern eines Entwicklungsvorhabens. Aus diesem Grund stellt sich der Autor der folgenden Frage:

Forschungsfrage 1: Wie lassen sich die Methoden des Anforderungsmanagements besser in Unternehmensprozesse integrieren?

Ein weiteres weit verbreitetes Risiko, stellt der Einsatz von MS Office-Lösungen zur Dokumentation von Anforderungen und projektrelevanten Informationen dar. Die so entstehenden Daten sind stark heterogen und distribuiert, was

ganzheitliche Verknüpfungen aller projektrelevanten Informationen verhindert. Um jedoch Informationen an die jeweils betroffenen Stakeholder adäquat kommunizieren zu können, sind semantische Verknüpfungen notwendig, weshalb sich die vorliegende Arbeit ebenfalls der Frage widmet:

Forschungsfrage 2: Wie lassen sich alle projektrelevanten Informationen automatisiert miteinander semantisch verknüpfen?

Die letzte Problemstellung welche mit der vorliegenden Arbeit adressiert wird, ist die systematische Nutzung bereits dokumentierter Informationen – sozusagen das digitale Erfahrungswissen der Unternehmen. In den meisten Fällen werden Dokumente nach Projektende kaum oder gar nicht wiederverwendet, wodurch kostbare bereits digitalisierte Informationen verloren gehen. Auch die parallele Nutzung dieses Wissens in anderen Projekten ist aufgrund fehlender Unterstützung nur begrenzt möglich. Die vorliegende Arbeit stellt sich daher der Frage:

Forschungsfrage 3: Wie lässt sich die Wiederverwendung bereits digital dokumentierter Informationen verbessern?

1.3 Rahmen der Arbeit

Die in der vorliegenden Arbeit beschriebenen Methoden und Konzepte wurden unter anderem im Rahmen des durch das BMBF geförderte „KMU-innovativ“ Projektes DAM4KMU entwickelt. Im Rahmen dieses Projektes werden neue Methoden und Konzepte entwickelt, welche die Beschreibung und das Managen von Anforderungen vereinfachen, den Bediener bei der Erfüllung der RE-Aufgaben methodisch und kontextsensitiv unterstützen sollen und eine bessere Vernetzung mit den Unternehmensprozessen ermöglichen sollen. Hierdurch sollen alle betroffenen Stakeholder aktiv in den Anforderungsmanagementprozess integriert werden, wodurch sich die Qualität der Produkte nachhaltig verbessern wird.

Als Anwendungsfälle dienen hierbei die Prozesse der Schuster Elektronik GmbH, welche seit über 50 Jahren kundenspezifische Messgeräte und umfangreiche Messsysteme herstellt. Durch ihre mechatronischen Produkte und ihr interdisziplinäres Entwicklungsteam stellt sie ein ideales Beispiel eines deutschen KMU aus der Hochtechnologie-Entwicklung dar. Aktuell nutzt die Schuster Elektronik GmbH handschriftliche Laufkarten sowie Office-Dokumente zum Dokumentieren und Verwalten von Anforderungen mit den damit verbundenen Informationen. Zur Sicherstellung der Übertragbarkeit der zu entwickelnden Lösung, konnten vier Evaluationspartner – Romaco Pharmatechnik GmbH, Kautenburger GmbH, Andata Entwicklungstechnologie GmbH und der J.Con GmbH – gewonnen werden, welche sich im Rahmen von Befragungen und Workshops ihre Anforderungen aus unterschiedlichen Branchen an einen durchgängig digitalisierten Anforderungsmanagement-Prozess beisteuern.

Das im Rahmen dieses Projektes geplante DAM4KMU-Framework besteht aus zwei Hauptkomponenten, einem generischen Datenmodell und dem Anforderungsmanagement-Assistenten, welche beide im Rahmen des Projektes entwickelt werden sollen (siehe Abbildung 1.1).

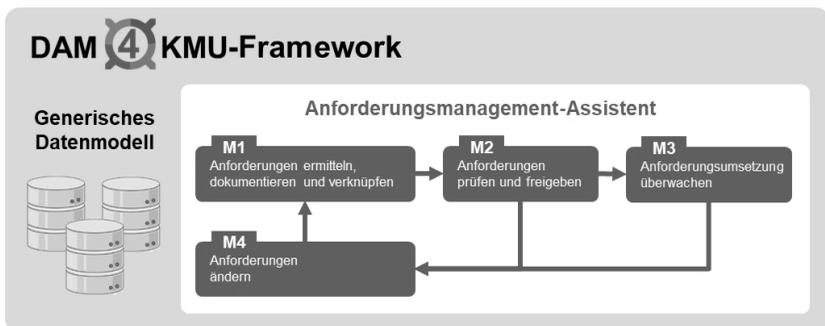


Abbildung 1.1: DAM4KMU-Framework

Derzeitige AM-Softwarelösungen weisen häufig Schwächen in der Funktionalität und Ausstattung auf. In vielen Unternehmen – vor allem bei KMU – kommen ausschließlich Office-Produkte (Excel, Word) und E-Mails zum Einsatz, welche im Gegensatz zu speziellen Softwarelösungen (z.B. IBM Rational DOORS und PTC Integrity) zwar das Sammeln und Clustern der Anforderungen ermöglichen, aber einige erhebliche Nachteile mit sich bringen können:

- Der Erhebungs- und Managementprozess wird nicht aktiv unterstützt.
- Der Änderungsverlauf von Anforderungen ist nicht nachvollziehbar.
- Der manuelle Austausch der Dokumente ist träge und führt zu Redundanzen.
- Die Analyse der Abhängigkeiten zwischen mehreren Anforderungen sowie die Auswirkungen einer Anforderungsänderung sind durch diese Art der Anforderungsdokumentation nicht möglich.

Aus diesem Grund werden im Rahmen des Projektes DAM4KMU neue Methoden und Konzepte entwickelt, welche die Beschreibung und das Managen von Anforderungen vereinfachen, den Bediener bei der Erfüllung der AM-Aufgaben methodisch und kontextsensitiv unterstützen sollen und besser mit den Unternehmensprozessen vernetzt werden können. So können alle betroffenen Stakeholder aktiv in den Anforderungsmanagementprozess integriert werden, wodurch sich die Qualität der Produkte nachhaltig verbessern wird.

1.4 Aufbau der Arbeit

Die vorliegende Arbeit gliedert sich in sieben Kapitel, welche im Folgenden kurz vorgestellt werden. Im Grundlagenkapitel wird das Anforderungsmanagement in den Produktlebenszyklus eingeordnet und die generellen Grundlagen erläutert. Da Anforderungen und das damit in Verbindung stehende Wissen überwiegend textbasiert dokumentiert werden, müssen diese für die weitere Verarbeitung computerlinguistisch aufbereitet werden. Hierzu werden häufig maschinelle Lernverfahren verwendet, welche im Rahmen des Grundlagenka-

pitels eingeleitet werden. Darauf aufbauend werden die Grundlagen der Computerlinguistik erläutert und die für diese Arbeit wesentlichen Implementierungen vorgestellt (siehe Abbildung 1.2).



Abbildung 1.2: Aufbau der vorliegenden Arbeit

Im Grundlagenkapitel wird das Anforderungsmanagement in den Produktlebenszyklus eingeordnet und die generellen Grundlagen erläutert. Da Anforderungen und das damit in Verbindung stehende Wissen überwiegend textbasiert dokumentiert werden, müssen diese für die weitere Verarbeitung computerlinguistisch aufbereitet werden. Hierzu werden häufig maschinelle Lernverfahren verwendet, welche im Rahmen des Grundlagenkapitels eingeleitet werden. Darauf aufbauend werden die Grundlagen der Computerlinguistik erläutert und die für diese Arbeit wesentlichen Implementierungen vorgestellt.

Im Stand der Technik werden wissenschaftliche Arbeiten und etablierte am Markt erhältliche Softwarelösungen in den Bereichen des Wissens- und Anforderungsmanagements analysiert und Anforderungen an das zu entwickelnde Konzept abgeleitet. Hierbei wird vor allem der Fokus auf die Unterstützung bei Dokumentation, Wissensverknüpfung, Tragweitenanalyse von

Änderungen, Wiederverwendung von Wissen sowie der Recherche neuer Informationen gelegt.

Basierend auf diesen Anforderungen wird in Kapitel vier ein Konzept eines Assistenzsystem zur kontextsensitiven Unterstützung des anforderungsbasierten Wissensmanagements entwickelt. Zur besseren Übersicht werden die einzelnen Funktionen in insgesamt sechs Assistenzmodule gegliedert, welche durch ein semantisches Datenmodell miteinander verknüpft werden.

In Kapitel fünf wird eine mögliche prototypische Implementierung des Konzepts beschrieben, welche die wesentlichen Funktionen mit aktuell verfügbaren open Source Bibliotheken realisiert.

Im Rahmen der Evaluation werden die einzelnen Funktionen des Assistenzsystems hinsichtlich ihrer Präzision, Ausbeute und dem F1-Maß evaluiert, mit dem Ziel die grundsätzliche Realisierbarkeit der Funktionen aufzuzeigen. Anschließend werden das Konzept und die prototypische Implementierung mit den im Stand der Technik abgeleiteten Anforderungen und abschließend mit den initial gestellten Forschungsfragen abgeglichen und diskutiert.

Abschließend werden die erarbeiteten Ergebnisse zusammengefasst und ein Ausblick formuliert.

2 Grundlagen

In diesem Kapitel werden alle für diese Arbeit relevanten Grundlagen erläutert. Beginnend werden der Produktlebenszyklus und seine Phasen sowie am Markt etablierte Vorgehensmodelle zur Produktentwicklung vorgestellt. Anschließend werden das Anforderungsmanagement und seine Kernaktivitäten beschrieben. Abschließend werden die für diese Arbeit relevanten Grundlagen im Bereich der computerlinguistischen Textverarbeitung (englisch: Natural Language Processing, NLP) zusammengefasst.

2.1 Der Produktlebenszyklus

In Zeiten in denen das Thema Umweltschutz und Nachhaltigkeit immer mehr an Bedeutung gewinnen, reicht es nicht mehr aus nur innovative, sondern auch nachhaltige Produkte zu entwickeln. Dieser Umstand macht es erforderlich alle Phasen des Produktlebenszyklus (siehe Abbildung 2.1) zu berücksichtigen, was die Komplexität der ohnehin meist komplexen Systeme zusätzlich erhöht [Eig 13, S. 1]. Fehler, die erst in den Spätphasen der Produktentstehung erkannt werden, sind schwer zu beheben und die Verbesserung gestaltet sich sowohl zeit- als auch kostenintensiv.

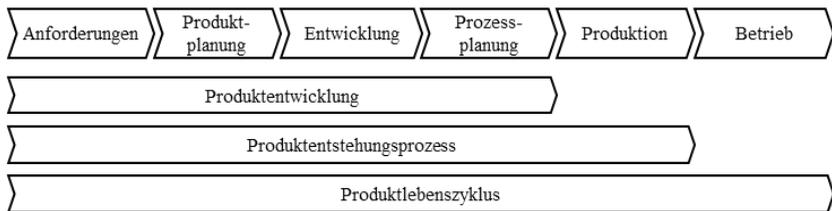


Abbildung 2.1: Produktlebenszyklusphasen [Eig 13, S. 9, Reu 14, S. 7]

Der Produktlebenszyklus (PLZ) umfasst alle Phasen welche ein Produkt in seinem „Leben“ durchschreitet, beginnend bei der Produktidee über die Phasen der Produktentwicklung bis hin zum Betrieb und dem schlussendlichen Recycling. Allgemein betrachtet ist der PLZ ein „theoretisches und abstrahierendes Konzept zur Erfassung der zeitlichen Entwicklung von Produkten“ [Sch 15, S. 21]. Je nach Sichtweise existieren unterschiedliche Darstellungsformen des PLZ, die beispielsweise nur Phasen nach dem Markteintritt des Produkts berücksichtigen [Pah 07, S. 296f.]. Im Rahmen der vorliegenden Arbeit wird jedoch der vollständige Prozess ab dem Zeitpunkt der Produktidee betrachtet.

2.1.1 Der Produktentstehungsprozess

Eine der wesentlichen und für das vorliegende Konzept entscheidenden Phasen des PLZ ist der Produktentstehungsprozess (PEP). Er umfasst im Wesentlichen die Phasen der Anforderungsermittlung, der Produktplanung, Entwicklung und der Prozessplanung (siehe Abbildung 2.2). Dieser Prozess hat zum Ziel, eine Produktidee in ein Endprodukt zu überführen, welches auf dem Markt eingeführt werden soll [Luc 14, S. 9]. Das tatsächliche Vorgehen hierbei ist in der Regel von den jeweiligen Unternehmen und den zu entwickelnden Produkten abhängig. Je nach Projekt können die einzelnen Phasen demnach unterschiedlich lange und unterschiedlich komplex sein [Loh 13, S. 13].

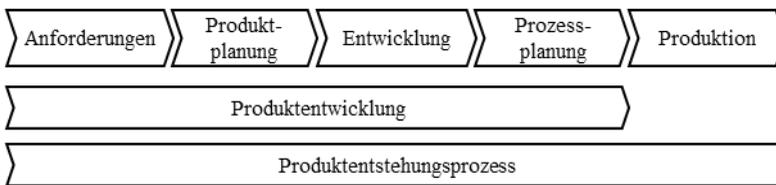


Abbildung 2.2: Der Produktentstehungsprozess [Eig 13, S. 9]

Die Produktentwicklung hat die übergeordnete Aufgabe, das „intellektuelle Produkt, d.h. die Produktbeschreibung mit allen dazugehörigen Dokumenten, Beschreibungen, Spezifikationen, digitalen Modellen und Entwurfsunterlagen

aller zugehörigen Betriebsmittel (Werkzeuge, Maschinen, Anlagen)“ [Bul 09, S. 247], zu entwickeln.

2.1.2 Klassische Vorgehensmodelle

Es existieren unterschiedliche Vorgehensmodelle, um die einzelnen Phasen der Produktentwicklung zu planen, zu strukturieren, zu dokumentieren und durchzuführen. Sie ermöglichen das Festlegen von Verantwortlichkeiten, Zielterminen und Teilschritten bzw. Aufgaben. Infolgedessen findet eine Minderung von Risiken statt, welche die Wahrscheinlichkeit einer rentablen Produktentwicklung erfolversprechend erhöht. Ein weiterer großer Vorteil vieler Vorgehensmodelle zeigt sich in der Rückverfolgbarkeit von Vorgängen, wodurch Fehler einfacher gefunden und korrigiert werden können [Fri 09, S. 1].

Jede Methode biete andere Stärken und Schwächen was es den Unternehmen ermöglichen, das eigene Vorgehensmodell den projektspezifischen Bedürfnissen anzugleichen. Das Spektrum reicht hierbei von sequentiellen Vorgehensweisen, über iterativ-inkrementelle, bis hin zu modernen agilen Strategien [Joc 11, S. 26].

Das Wasserfallmodell gilt als Basis vieler moderne Vorgehensweisen. Es wurde in den 1960er Jahren für die Softwareentwicklung konzipiert und vor allem durch seinen linearen Ablauf gekennzeichnet. Der Ausdruck „Wasserfallmodell“ leitet sich von der häufig angewandten Darstellung der Phasen als Kaskaden ab [Nie 13, S. 180]. Ein Prozess wird in verschiedene Phasen eingeteilt, die linear aufeinanderfolgend abgearbeitet werden (siehe Abbildung 2.3).

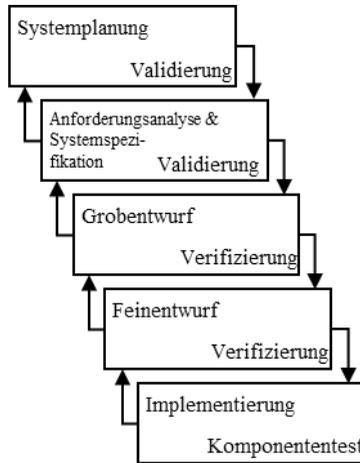


Abbildung 2.3: Beispiel eines Wasserfallmodells [Mas 12, S. 675]

Das Modell sieht im ersten Schritt eine vollständige Beschreibung des Lösungskonzeptes vor. Dieses Vorgehen ist sehr unflexibel in Hinblick auf Änderungen, welche jedoch in nahezu jedem Entwicklungsprojekt vorkommen. Viele Anforderungen und Randbedingungen werden erst im Laufe der Entwicklung klar und je nachdem können sich Kundenwünsche ändern. In der Theorie sind Rückschritte zu früheren Phasen zwar denkbar, aufgrund der festgesetzten Meilensteine sind sie in der Praxis hingegen nur sehr schwer zu realisieren [Nie 13, S. 180].

Das in Abbildung 2.4 dargestellte V-Modell 97 ist neben dem Wasserfallmodell, eines der klassischen Vorgehensmodelle in der Produktentwicklung. Es wurde Anfang der 1990er Jahre ursprünglich für die Softwareentwicklung konzipiert, findet jedoch heute auch in vielen Unternehmen in der mechanischen und mechatronischen Produktentwicklung Anwendung [Fel 13, S. 18].

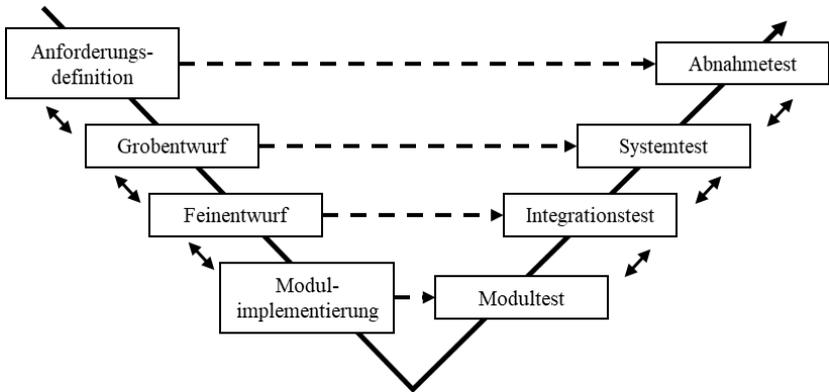


Abbildung 2.4: Darstellung des V-Modells [Gra 14, S. 112]

Der Begriff leitet sich aus dem V-förmigen Aufbau der Elemente ab [Joc 11, S. 27]. Im Gegensatz zum Wasserfallmodell sieht es Qualitätsprüfungen für alle Entwicklungsphasen vor [Stö 18, S. 25]. Aus diesem Grund ist das V-Modell vor allem für Projekte mit sehr hohen Anforderungen von Vorteil.

Das Prozessparadigma basiert, auf einer iterativ-inkrementellen Vorgehensweise, die mehrere Phasen durchläuft. So lassen sich mögliche Änderungen einfacher integrieren und Fehler bereits in der frühen Phase der Produktentwicklung reduzieren [Poh 09, S. 36f.]. Das V-Modell brilliert, wie auch das Wasserfallmodell, mit seiner Übersichtlich- und Nachvollziehbarkeit. Zudem erhöht es die Prozessqualität durch die Kontrolle der Ergebnisse am Ende jedes Abschnitts. Allerdings steht das Projektergebnis frühestens zum Abschluss fest, da die zu Beginn des Projekts gestellten Anforderungen erst durch den Abnahmetest zeitlich verzögert getestet werden können. Darüber hinaus sind frühere Auskünfte an das Projektteam oder den Kunden nicht eingeplant [Stö 18, S. 25].

Im Jahr 2005 wurde das V-Modell 97 durch das V-Modell XT (XT = „eXtreme Tailoring“) des Bundesministeriums des Inneren als Entwicklungsstandard abgelöst [Fri 09, S. 2]. Der Namenszusatz des neu aufgelegten Modells impliziert den hohen Grad an Flexibilität und Anpassungsfähigkeit, den die neue Variante im Vergleich zum starren alten Modell bietet [Poh 09, S. 37]. „Wie sein

Vorgänger [...] ist das V-Modell XT inzwischen als Vorgabe für den Einsatz im gesamten zivilen [...] Bundesbereich empfohlen beziehungsweise verbindlich vorgeschrieben“ [Höh 08, S. 3].

2.1.3 Agile Vorgehensmodelle

Ältere Modelle wie das Wasserfallmodell oder das V-Modell 97 sind oft mit der Problemstellung konfrontiert, dass sich Anforderungen während der Projektphasen ändern oder wichtige, neue Anforderungen erst spät erkannt werden. Oftmals spielen auch externe Faktoren eine Rolle, die die Projektbeteiligten dazu zwingen, die Anforderungen zu optimieren. Wie bereits erwähnt, lassen sich, durch die starre und unflexible Art der genannten Vorgehensweisen, Anforderungsänderungen in der Praxis oft nur sehr schwer vornehmen. Um diese Probleme in der Produktentwicklung zu vermeiden, wurden die agilen Vorgehensweisen entwickelt [Nie 13, S. 179].

Der Ausdruck „agil“ impliziert einen hohen Grad an Leichtgewichtigkeit bzw. Flexibilität. Agil bedeutet in diesem Zusammenhang, dass sich die Entwickler eine für das Projekt qualifizierte Methode aus einem riesigen Methodenpool auswählen können. Im Anschluss wird die ausgewählte Methode während des Projektprozesses fortlaufend hinterfragt, um gegebenenfalls eine andere, geeignetere Methodik zu selektieren [Poh 09, S. 39]. Dies erleichtert es Anforderungsänderungen zur Projektlaufzeit zu integrieren [Nie 13, S. 179f.]. Um anpassungsfähig und dynamisch zu agieren, sind sie deshalb darauf ausgelegt weniger Vorgaben zu stellen und „den bürokratischen Aufwand niedrig zu halten“ [Gra 14, S. 113].

Kernziel agiler Vorgehensmodelle ist die Reduktion dokumentationslastiger Vorgänge und den Erhalt der Individualität der Beteiligten [Stö 18, S. 26]. Dieses Ziel wurde unter anderem in dem „agilen Manifest“ fixiert, welches als Leitfaden für agile Projekte dienen soll. Dieses sieht neben der größeren Freiheit der Entwickler, eine stärkere Zusammenarbeit aller Stakeholder sowie das permanente Eingehen auf Anforderungsänderungen, als das Fundament eines erfolgreichen Projektes an.

Aufgrund der immer kürzer werdenden Produktlebenszyklen und der sich schnell wandelnden Märkte, bieten agile Vorgehen daher große Potentiale, vor

allem für KMU [Sch 14, S. 65]. Eine der bekanntesten und am weitesten verbreiteten, agilen Vorgehensweisen ist neben Extreme Programming und Feature Driven Development, die Scrum-Methode, welche im Rahmen dieser Arbeit aber nicht genauer erläutert wird [Stö 18, S. 26].

Das durch das Institut für Produktentwicklung (IPEK) des Karlsruher Instituts für Technologie (KIT) nach Albert Albers entwickelte integrierte Produktentstehungs-Modell (iPeM), ist ein leichtgewichtiges sowie agiles Metamodell zur ganzheitlichen methodischen Unterstützung der Produktentstehungsprozesse [Alb 11, S. 7]. Ziel dieses Modells ist es, einen Produktentstehungsprozess zu schaffen, welcher es interdisziplinären Teams ermöglicht, komplexe Produkte zu entwickeln und dabei ohne die nötige Flexibilität zu verlieren, auf eventuelle Änderungen reagieren zu können [Zin 13, S. 93f.]. Es bietet den Projektbeteiligten Hilfestellungen zur Orientierung, Navigation, Dokumentation, Prozess- und Wissensarbeit mit Hilfe einer transparenten und integrierten Darstellung von Informationen [Alb 12, S. 33].

Erreicht wird dies durch die Bereitstellung von Methodischen Elementen entlang des gesamten Produktlebenszykluses, welche jedoch keine zeitliche Abhängigkeiten besitzen [Alb 11, S. 11f.]. Dies unterscheidet das iPeM (siehe Abbildung 2.5) zudem von den anderen Vorgehensmodellen, welche sich rein auf die Produktentwicklung beziehen. Somit stellt das Rahmenwerk des IPEK die Basis für ein gemeinsames Verständnis des PEP dar, da es die beteiligten Akteure der unterschiedlichen Disziplinen und Ebenen integriert [Mey 14, S. 14].

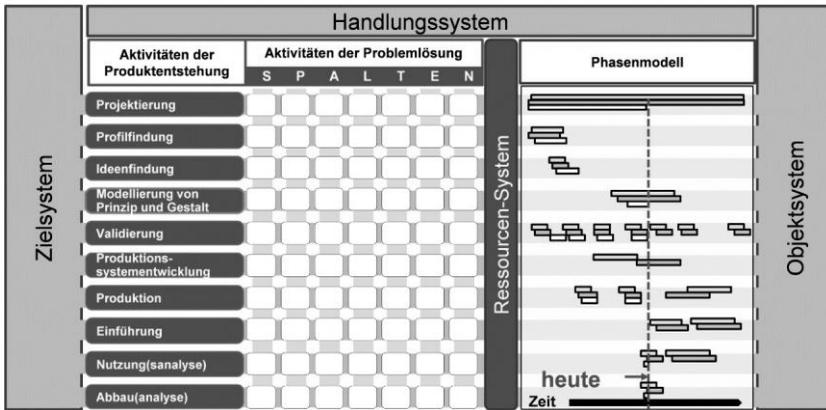


Abbildung 2.5: Das integrierte Produktentstehungsmodell [Zin 13, S. 96]

Das Fundament des iPem stellt das ZHO-Tripel das aus dem Ziel-, Handlungs-, und dem Objektsystem besteht [Mey 14, S. 15]. Das Zielsystem umfasst alle relevanten Ziele, Anforderungen, Abhängigkeiten und Randbedingungen, welche das geplante Produkt umfassen. Das Objektsystem hingegen charakterisiert die Ressourcen, die Erkenntnisobjekte und die Ergebnisse des Handlungssystems (z.B. Prototypen oder Skizzen). Den Kern des iPem stellt das Handlungssystem dar, welches das Zielsystem sowohl definiert als auch in ein Objektsystem transferiert. Es enthält sämtliche Aktivitäten und Ressourcen (z.B. Kapital, Mitarbeiter, Betriebsmittel etc.), die im Zuge des PEP erforderlich sind [Mey 14, S. 15f.]. Mit Hilfe der Aktivitätsmatrix, lassen sich die verschiedenen Aktivitäten innerhalb des Handlungssystems abbilden, welche die Überführung des Ist- in den Soll-Zustand ermöglichen. Die einzelnen Aktivitäten entlang des Produktlebenszyklus bilden die Zeilen der Aktivitätsmatrix und können entweder parallel, iterativ oder stochastisch verteilt ausgeführt werden. Die daneben stehenden SPALTEN¹ stellen eine generische Methodik dar, welche innerhalb einer jeden Aktivität angewendet werden kann und somit das strukturierte Arbeiten fördert [Mey 14, S. 18ff.]. Zusätzlich enthält das

¹ **SPALTEN:** Situationsanalyse, Problemeingrenzung, Alternative Lösungssuche, Lösungsauswahl, Tragweitenanalyse, Entscheiden und Umsetzen sowie Nacharbeiten und Lernen

Modell ein Ressourcen-System, welches alle zur Verfügung stehenden Ressourcen abbildet sowie ein Phasenmodell, welches die zeitabhängigen, dynamischen Aspekte des Produktentwicklungsprozesses repräsentiert [Alb 11, S. 16].

2.2 Das Anforderungsmanagement

In Zeiten volatiler Märkte und einer immer stärkeren Ausrichtung des Wirtschaftsgeschehens auf die Bedürfnisse der Kunden kommen effizienten Produktentwicklungsprozessen aufgrund des hohen nationalen sowie internationalen Wettbewerbsdrucks eine Schlüsselrolle zu [Dic 17, S. 1]. Der Wandel der produzierenden Industrie hin zu individuelleren Produkten hat primär komplexere Anforderungen zur Konsequenz.

Definition 1: Anforderung

Nach IEEE (1991, 1990) ist eine Anforderung:

- (1) Eine Bedingung oder eine Fähigkeit, die ein Benutzer benötigt, um ein Problem zu lösen oder ein Ziel zu erreichen.
- (2) Eine Bedingung oder Fähigkeit, die ein System oder eine Systemkomponente erfüllen oder besitzen muss, um einen Vertrag, eine Norm, eine Spezifikation oder andere formell vorgeschriebene Dokumente zu erfüllen.
- (3) Eine dokumentierte Repräsentation einer Bedingung oder Eigenschaft gemäß (1) oder (2).

Waren es „Anfang der neunziger Jahre einige wenige Steuergeräte in einem Neuwagen mit ungefähr hundert Seiten an Spezifikationen, so sind es bereits heute fünfzig und mehr Steuergeräte mit über 100.000 Seiten“ [Ebe 14, S. 2]. Aus diesen Gründen nimmt die Disziplin des Anforderungsmanagements eine immer zentralere Rolle während des gesamten Produktlebenszyklus ein, da der

richtige Umgang mit Anforderungen, gemäß verschiedener Standards und Normen, bereits zu Beginn des Produkt-entwicklungsprozesses über den Projekterfolg oder das Scheitern entscheidet.

Das Anforderungsmanagement unterstützt sämtliche Organisationseinheiten mit systematischen Vorgehensweisen zur Spezifikation und zum Management von Anforderungen, mit dem Hauptziel „eine vollständige, eindeutige und widerspruchsfreie Spezifikation aller Anforderungen“ vorzunehmen [Gli 06, S. 2]. Neben dem Prozess der Identifikation und der Handhabung von Anforderungen agiert das AM vorrangig als Vermittler zwischen den unternehmensinternen Disziplinen und den Stakeholdern [Dic 17, S. 126f.]. In dieser Funktion ist es dafür verantwortlich, die Kommunikation zwischen sämtlichen Beteiligten zu verbessern, um beispielsweise eine Übereinstimmung zwischen den Stakeholdern über alle relevanten Anforderungen und somit eine Vermeidung von Konflikten zu bewirken [Lau 16, S. 20ff.].

Definition 2: Anforderungsmanagement (Requirements Engineering)

Nach Pohl [Poh 15, S. 4] ist das Requirements Engineering (deutsch: Anforderungsmanagement) „ein systematischer und disziplinierter Ansatz zur Spezifikation und zum Management von Anforderungen mit den folgenden Zielen:

- (1) Die relevanten Anforderungen zu kennen, Konsens unter den Stakeholdern über die Anforderungen herzustellen, die Anforderungen konform zu vorgegebenen Standards zu dokumentieren und die Anforderungen systematisch zu managen.
- (2) Die Wünsche und Bedürfnisse der Stakeholder zu verstehen, zu dokumentieren sowie die Anforderungen zu spezifizieren und zu managen, um das Risiko zu minimieren, dass das System nicht den Wünschen und Bedürfnissen der Stakeholder entspricht.“

Nach dem International Requirements Engineering Board (IREB) stellt das Anforderungsmanagement (englisch: Requirements Management) grundsätzlich ein Teilgebiet des Requirements Engineering (RE) dar, das durch die Disziplin des Requirements Development ergänzt wird [Lau 16, S. 156]. Im allgemeinen Sprachgebrauch und auch in der Literatur wird das Anforderungsmanagement jedoch oftmals mit dem Requirements Engineering gleichgesetzt [Sch 09, S. 5]. Aus diesem Grund werden die beiden Begrifflichkeiten auch im Kontext dieser Arbeit gleichbedeutend verwendet.

2.2.1 Anforderungsermittlung

Vor Beginn der Planung eines Produkts – unabhängig vom Vorgehensmodell – sind in einem ersten Schritt potentielle Anforderungen an das zu fertigende Produkt zu ermitteln. Um die dafür notwendigen Anforderungsquellen zu bestimmen, muss zunächst der Systemkontext des geplanten Systems analysiert werden [Lau 16, S. 120]. Nach Pohl [Poh 15, S. 21ff.] gliedern sich die unterschiedlichen Anforderungsquellen in Stakeholder, Dokumente und bestehende Systeme.

Die Stakeholder gelten als eine der wichtigsten Quellen, da sie einen direkten bzw. indirekten Bezug auf die Anforderungen des geplanten Produkts haben. Die Nichtberücksichtigung relevanter Stakeholder sowie die mangelhafte Kommunikation untereinander, führt in der Regel zu einem schlechten Projektergebnis oder sogar zum Projektscheitern. Aus diesem Grund bedarf es von Beginn des Projekts an einer permanenten Integration der Stakeholder, um alle wichtigen Anforderungen zu realisieren [Poh 09, S. 27f.].

Der Begriff Dokumente umfasst alle für das zu entwickelnde Produkt relevanten Informationen, welche z.B. Normen/Standards, Gesetztestexten oder in Lastenheften oder Fehlerberichten von Vorgängermodellen zu entnehmen sind [Lau 16, S. 121]. Systeme, die sich bereits im Betrieb befinden, liefern weitere Anhaltspunkte für Anforderungen. Einerseits bringen ältere Vorgängersysteme Ideen für Verbesserungen und Erweiterungen von Anforderungen hervor, andererseits zeigen Konkurrenzsysteme bereits realisierte und potentielle Anforderungsumsetzungen [Poh 09, S. 27].

Nachdem alle relevanten Anforderungsquellen identifiziert wurden, beginnt der eigentliche Prozess der Ermittlung der Anforderungen. Hierzu bieten sich aus der Literatur unterschiedlichste Techniken an, welche die systematische Anforderungsermittlung unterstützen und entsprechend der gegebenen Ausgangssituation auszuwählen sind [IRE 17, S. 13]. Mit ihrer Hilfe lassen sich zum einen bewusste, aber auch unbewusste und unterbewusste Anforderungen von Stakeholdern identifizieren. Beispielsweise können detaillierte und explizite Anforderungen durch den direkten bzw. indirekten Kontakt mit dem Stakeholder über Befragungstechniken (z.B. Interview, Fragebogen etc.) ermittelt werden. In der Regel sind sich Stakeholdern am Anfang des Prozesses im Unklaren darüber, welche Spezifikationen das Endprodukt beinhalten soll oder können nicht aktiv in den Prozess integriert werden (z.B. Endkunde). Aus diesem Grund werden hierfür Beobachtungsmethoden empfohlen, um die relevanten Anforderungen zu ermitteln. Je nach Komplexität des geplanten Systems, bietet sich der Bau von Prototypen an, um mögliche nicht berücksichtigte Anforderungen noch in der frühen Phase identifizieren zu können. In anderen Fällen genügen jedoch auch Kreativmethoden, um die relevanten Anforderungen zu identifizieren [Poh 09, S. 27ff.].

2.2.2 Anforderungsdokumentation

Eine der wichtigsten Aufgaben eines Requirement Engineers, ist die Dokumentation relevanter Informationen über das zu entwickelnde System. Hierzu zählen neben z.B. Interviewprotokollen, Berichten etc. vor allem die ermittelten Anforderungen [Poh 15, S. 41]. Die dabei entstehende Dokumentation der Systemspezifikationen, dient der Bewahrung des Überblicks über alle Anforderungen. Zudem dient diese der verbesserten Kommunikation zwischen den beteiligten Stakeholdern, welche gerade hinsichtlich global agierender Unternehmen, nicht bei allen Anforderungsworkshops dabei sein können. Nur so kann sichergestellt werden, dass permanent ein gemeinsames Verständnis über die jeweiligen Anforderungen geschaffen wird [Ebe 14, S. 101]. Ein weiteres wichtiges Argument für die Anforderungsdokumentation, sind die Aspekte der rechtlichen Absicherung zwischen Auftraggeber und -nehmer. Durch die schriftliche Formulierung der geforderten Spezifikation sind die zu erfüllenden

Vorgaben zu jeder Zeit vertraglich geregelt [Poh 15, S. 42]. Die thematisierte Bedeutung der Dokumentation spiegelt sich am Anteil der Anforderungsdokumentation am Gesamtaufwand des Anforderungsmanagements wieder, der bei 60-70% liegt [Lau 16, S. 132].

Analog zur Anforderungsermittlung existieren bei der Erstellung von Anforderungsdokumenten verschiedene Techniken, deren Spektrum von natürlichsprachigen Beschreibungen, über die Nutzung grafischer Modellierungssprachen (z.B. UML oder SysML, OMG 15, OMG 18), bis hin zur Veranschaulichung der gewünschten Funktionen anhand von Prototypen reicht. Im Hinblick auf die Auswahl einer geeigneten Methode stellt die Kombination mehrerer Methoden oftmals die beste Lösung dar, da auf diese Weise verschiedene Vorteile vereint werden können [Hru 17, S. 428]. Um die Vielzahl der teils komplexen Informationen übersichtlich darzustellen, müssen die Anforderungsdokumente strukturiert aufgebaut sein. Für das Layout der Dokumente gibt es zahlreiche Entwürfe oder Standardgliederungen. Weit verbreitete und anerkannte Standardgliederungen werden unter anderem durch die Vorgehensmethoden des V-Modells bzw. des Rational Unified Process vorgegeben [Poh 09, S. 45ff.].

In der Praxis haben sich innerhalb des deutschsprachigen Raums vor allem die vorgegebenen Strukturen des V-Modells etabliert. Der Handlungsleitfaden des V-Modells sieht zwei unterschiedliche Perspektiven von Spezifikationen vor. Zunächst überträgt der Auftraggeber seine Anforderungsspezifikationen in das Lastenheft², bevor der Auftragnehmer aufbauend auf dem Lastenheft Lösungsspezifikationen zur Realisierung der Forderungen in das Pflichtenheft³ einträgt [Ebe 14, S. 103]. Mögliche Alternativen stellen das Vorgehen nach dem Rational Unified Process oder dem IEEE-Standard 890-1998 dar [Poh 09, S. 46].

² Das Lastenheft enthält die „Gesamtheit der Forderungen an die Lieferungen und Leistungen eines Auftragnehmers“ VDI 01.

³ Das Pflichtenheft beschreibt die „vom Auftragnehmer erarbeiteten Realisierungsvorgaben aufgrund der Umsetzung des vom Auftraggeber vorgegebenen Lastenhefts“ VDI 01.

Aufgrund individueller projekt- und produktspezifischer Bedürfnisse ist es nahezu unmöglich ein ideales standardisiertes Anforderungsdokument zu entwerfen. Innerhalb einer Organisation empfiehlt es sich jedoch die Dokumente so gut zu standardisieren wie möglich, um die Übertragbarkeit in Folgeprojekte, und damit die Wiederverwendung von Anforderungen, zu erleichtern. Zudem sollte je nach Projektgröße ein Glossar erstellt und gepflegt werden, welches alle innerhalb des Projektes verwendeten Fach- und Spezialbegriffe für alle Beteiligten einheitlich definiert und so ein gemeinsames Verständnis aller Beteiligten gewährleistet [Hru 17, S. 431].

In der Praxis werden Anforderungen bevorzugt natürlichsprachlich dokumentiert. Beim Verständnis natürlicher Sprache spielen jedoch zum einen der kulturelle Hintergrund als auch das Erfahrungswissen der beteiligten Personen eine entscheidende Rolle. Mit Hilfe von Satzschablonen (siehe Abbildung 2.6), kann die Qualität natürlichsprachlicher Anforderungen verbessert und somit mögliche Fehler vermieden werden. Um hierbei auch die lexikalische Eindeutigkeit der Dokumentation zu gewährleisten, empfiehlt sich der Einsatz eines Projektglossars.

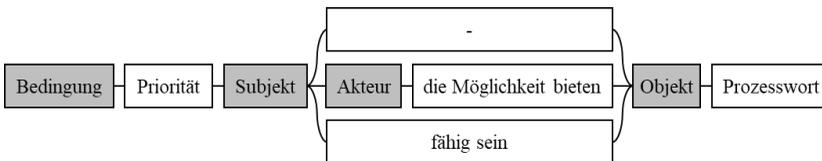


Abbildung 2.6: Funktionale Anforderungsschablone [Poh 15, S. 64]

Im Gegensatz zu frei formulierten Anforderungen bieten Anforderungsschablonen eine klare Struktur, welche es dem Autor erleichtert, Anforderungen kurz und ohne formale Redundanzen z.B. durch verschachtelte Satzstrukturen, zu formulieren und somit eine syntaktisch eindeutige Anforderung zu dokumentieren. Zudem stellen Anforderungsschablonen die Festlegung der Priorität einer Anforderung, die Art einer Anforderung (funktional oder nicht-funktional) sowie die Charakteristik funktionaler Anforderungen hinsichtlich ihrer Systemtätigkeit sicher [Poh 15, S. 58].

2.2.3 Anforderungsprüfung

Fehlerhafte oder nicht erkannte Anforderungen innerhalb der Prozesse wirken sich oftmals schwerwiegend auf den weiteren Projektverlauf aus und sorgen für hohe Kosten, Terminverzug, unzufriedene Stakeholder oder sogar für das Scheitern des Projekts (siehe Abbildung 2.7). Aus diesem Grund stellt das Prüfen und Abstimmen von Anforderungen eine der wichtigsten Kernaktivitäten des Anforderungsmanagements dar [Ebe 14, S. 3ff.].

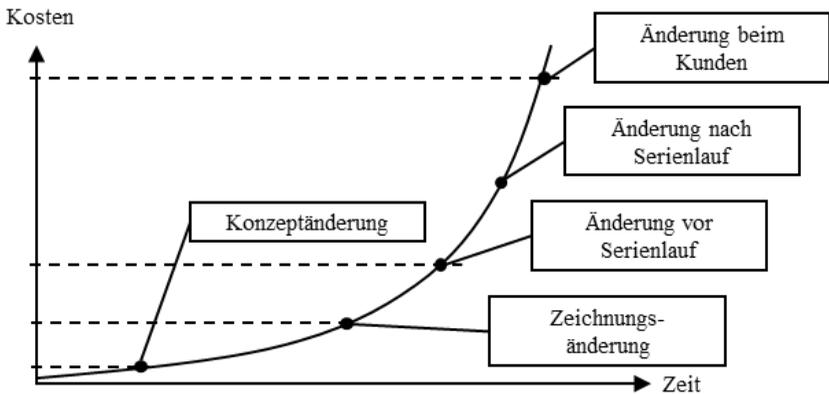


Abbildung 2.7: 10er Regel der Fehlerkosten [angelehnt an Ehr 09, S. 170]

Neben rein formellen Fehlern oder Widersprüchen innerhalb der Anforderungsdokumentation, ist das Anforderungsmanagement bestrebt, die Anforderungen hinsichtlich der Konsistenz in Bezug auf die ursprünglich vom Kunde geäußerten Wünsche aufrecht zu erhalten, um über alle Stakeholder hinweg ein gemeinsames Verständnis sicherzustellen [Lau 16, S. 140f.]. Nach ISO/IEC/IEEE 29148:2011 (29148) werden die Qualitätsmerkmale an Anforderungen definiert. Krisch [Kri 17] fasst zudem weitere in der Literatur vertretene Definitionen von Qualitätskriterien in ihrer Arbeit zusammen [Kri 17, S. 25ff.].

Zur Überprüfung der Qualität stehen laut IREB [IRE 17] mehrere Techniken zur Verfügung. Dazu zählen unter anderem die Stellungnahme (z.B. individuelle Prüfung der Anforderungen durch Kollegen), die Inspektion der Anforderungsdokumente, der Walkthrough (z.B. mit Hilfe eines Reviews), das perspektivenbasierte Lesen (z.B. aus Sicht des Endkunden), die Prüfung durch Prototypen und die Prüfung durch Checklisten [IRE 17, S. 26].

2.2.4 Anforderungsverwaltung

Nachdem die Anforderungen ermittelt, dokumentiert, geprüft und freigegeben wurden, ist es wichtig die Anforderungen systematisch und einheitlich zu verwalten. Um dies zu gewährleisten, müssen alle Anforderungen Priorisiert, mit sinnvollen Attributen versehen werden und die Nachverfolgbarkeit von Änderungen und Versionen sichergestellt werden. Die Verwaltung oder auch das Management von Anforderungen verfolgt den Zweck, allen Stakeholdern einen permanenten, schnellen Zugriff auf alle Anforderungen und Informationen über den gesamten Produktlebenszyklus zu gewährleisten [Poh 15, S. 123, 149].

Innerhalb dieses Vorgangs werden den jeweiligen Anforderungen beschreibende Eigenschaften hinzugefügt, um vor allem den späteren Nachvollziehbarkeitsprozess zu erleichtern. Typischerweise handelt es sich bei diesen Attributen z.B. um eine eindeutige ID (Kennung), einen Namen oder eine Versionsnummer.

Jeder Stakeholder hat seine Sicht auf die Anforderungen, weshalb die unterschiedlichen Perspektiven während der Priorisierung und Reihenfolgen Bestimmung für die Realisierung berücksichtigt werden müssen [Lau 16, S. 160, Poh 09, S. 129f.]. Für ein systematisches Vorgehen hierbei definiert IREB den folgenden Handlungsleitfaden: Priorisierungsziel festlegen, Priorisierungskriterien definieren, zu priorisierende Artefakte bestimmen, relevante Stakeholder definieren und Priorisierungstechnik auswählen [IRE 17, S. 30]. Das Spektrum der dabei zum Einsatz kommenden Priorisierungstechniken reicht von einfachen Workshops mit den Stakeholdern, über die Top-Ten Technik ,

bis hin zur Kano-Klassifikation [Lau 16, S. 162, Poh 15, S. 130ff.]. Für die vorliegende Arbeit leiten sich folgende Prioritätsklassen ab:

Priorität	Beschreibung
Muss	Must-have (Negation: Darf nicht) <ul style="list-style-type: none"> • Werden erwartet • Erfüllung: erzeugt keine Zufriedenheit • Nichterfüllung erzeugt Unzufriedenheit
Sollte	Should-have (Negation: Sollte nicht) <ul style="list-style-type: none"> • Werden verlangt • Erfüllt führt zu Zufriedenheit • Nichterfüllung führt zu Unzufriedenheit
Kann	Nice-to-have (Negation: Muss nicht) <ul style="list-style-type: none"> • Werden nicht erwartet • Bei Erfüllung: überproportionaler Einfluss auf die Kundenzufriedenheit • Nichterfüllung erzeugt keine Unzufriedenheit
Wird	Soon-to-be (Negation: Wird nicht) <ul style="list-style-type: none"> • Strategischer Aspekt (formuliert eine Absicht) • Wird nicht umgesetzt, aber in Zukunft

Tabelle 1: Prioritätsdefinition [Poh 15, S. 128]

Um Anforderungen sinnvoll managen zu können ist die Gewährleistung der Nachverfolgbarkeit über den gesamten Produktlebenszyklus unumgänglich [Lau 16, S. 163]. Ist dies nicht der Fall, erschwert dies die Wartung und Pflege der Anforderungen. Zudem fällt es erheblich schwerer zu ermitteln ...

- in welchem Stadium der Umsetzung sich Anforderungen befinden.
- ob sogenannte Goldrandlösungen umgesetzt werden.
- ob sich Goldrandlösungen in den Anforderungen befinden.
- welche Auswirkungen Anforderungsänderungen haben.
- ob Anforderungen Wiederverwendung finden können.
- welche Entwicklungsaufwände an die Umsetzung einer Anforderung gekoppelt sind.

Um die natürlicherweise auftretenden Änderungen von Anforderungen nachvollziehen zu können, ist die Versionierung von Anforderungen bzw. von Anforderungsdokumenten vorzunehmen. Anhand eines einfachen Nummerierungsschemas lassen sich Anforderungen durch die Zuteilung einer eindeutigen Versionsnummer klassifizieren [Lau 16, S. 166].

Abbildung 2.8 veranschaulicht die gewöhnliche Vorgehensweise der Versionierung von Anforderungen. Hierbei ist grundlegend zwischen kleinen und großen inhaltlichen Veränderungen von Anforderungen zu unterscheiden. Eine kleine Änderung hat lediglich eine Erhöhung der Inkrementzahl zur Folge, eine größere Änderung hingegen eine Erhöhung der Versionsnummer. Zusätzlich zu dieser Vorgehensweise existieren weitere Möglichkeiten die Versionierung von Anforderungen vorzunehmen [Poh 09, S. 141].

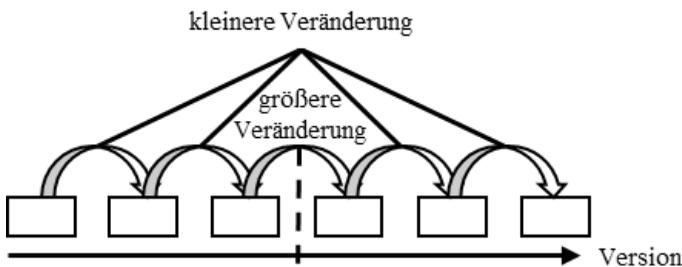


Abbildung 2.8: Versionierungsschema [Poh 09, S. 141]

Um Änderungen und deren Auswirkungen innerhalb eines Projektes stets im Blick zu behalten, ist eine der wichtigsten Aufgaben des Anforderungsmanagements das Verwalten und Managen von Anforderungsänderungen. Verantwortlich für die Entscheidung über eingehende Änderungsanträge ist das sogenannte Change-Control-Board [Lau 16, S. 168]. Neben der Entscheidung über die Annahme oder die Ablehnung eines Änderungsantrags hat dieses Gremium die Aufgaben, eingehende Änderungsanträge zu klassifizieren, Änderungsanträge hinsichtlich ihres Aufwands bzw. Nutzens zu beurteilen und zu priorisieren sowie neue Anforderungen auf Basis eingehender Änderungsan-

träge zu definieren [IRE 17, S. 32]. Damit alle Projektbeteiligten über die entsprechenden Änderungen informiert werden können, empfiehlt es sich aus allen Stakeholdergruppen entsprechende Vertreter in dieses Gremium zu integrieren. Je nach Größe der Projektgruppe kommt dem Änderungsmanager als Vorsitzender des Boards eine entscheidende kommunikative Rolle zu [Poh 09, S. 145].

Die eingehenden Änderungsanträge lassen sich gemäß IREB einer der drei Kategorien korrektive Anforderungsänderungen (z.B. Änderungen aufgrund eines Fehlerverhaltens des Systems), adaptive Anforderungsänderungen (z.B. Integration neuer Technologien) oder einer Ausnahmeänderungen (z.B. durch unerwartete Ereignisse) zuordnen [IRE 17, S. 33].

2.2.5 Anforderungsmanagement Werkzeuge

Ein umfassendes Anforderungsmanagement kann dem Benutzer den Umgang mit Änderungen eines Dokuments vereinfachen. Es gibt bereits Systeme, die standardisierte Anforderungen aus Dokumenten extrahieren können. Diese verwenden Parser, die die Position im ursprünglichen Dokument festhalten können. Mit der Kombination einer Versionierung von Dokumenten mit solchen Parsern können Abgleiche zwischen Versionen durchgeführt werden, um dem Benutzer Änderungen direkt anzuzeigen. Werkzeuge für das Anforderungsmanagement können vielseitig sein, da sie unterschiedliche Bereiche abdecken können. Microsoft Office ist ein verbreitetes Bürosoftwarepaket mit verschiedenen Komponenten zur Text- und Datenverarbeitung. In Word werden häufig Anforderungsdokumente erstellt, in Excel werden diese Anforderungen oft zwischen Kunden und Lieferanten abgestimmt. Es ist dennoch nicht geeignet, um eine Gesamtverwaltung in Verbindung mit dem Entwicklungsprozess oder vorherigen Projekte durchzuführen. Doors ist eine AM-Software der IBM, die ebenfalls für die Dokumentation, Prüfung und Abstimmung von Anforderungen geeignet ist. Sie basiert jedoch auf einer Datenbank, welche einige Vorteile mit sich bringt, wie zum Beispiel der Zugriff auf abgeschlossene Projekte und Anforderungen. Die Datenbank indexiert die Anforderungen, sodass bei der Dokumentation von Anforderungen strukturierte Dokumente entstehen (siehe Abbildung 2.9).

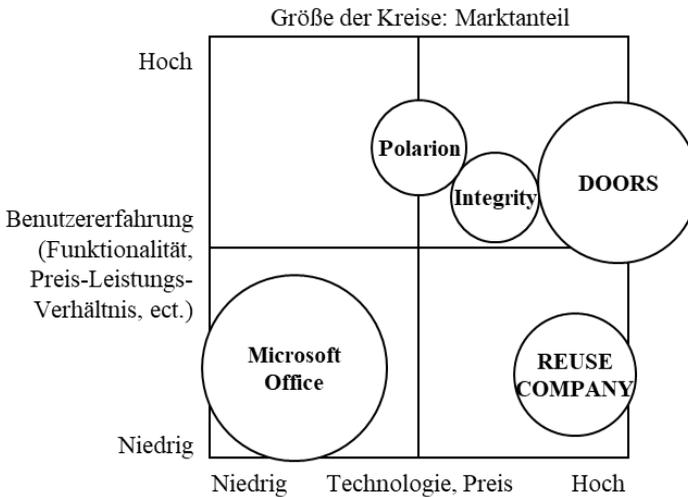


Abbildung 2.9: Vergleich von AM-Werkzeugen [Ebe 14, S. 286]

Eine weitere AM-Software, ReqMan, basiert ebenfalls auf einer Datenbank, die Dokumente und Anforderungen indexiert, versioniert und dadurch miteinander verlinkt. Zusätzlich bietet Sie die Möglichkeit mehrere Abstimmungsschleifen zwischen Kunden und Lieferanten durchzuführen, sodass alle Informationen an einer Stelle zu finden sind. Dadurch können beispielsweise Bewertungen von Anforderungen aus vergangenen Projekten direkt übernommen werden. Ähnliche Programme sind Jama und Polarion.

Diese bisher genannten Programme und Werkzeuge unterstützen bei der Verwaltung von Anforderungen. Es gibt dennoch keine Software, die den Anwender dahingehend unterstützt, dass Anforderungen aus juristischen oder technischen Dokumenten extrahiert werden. Somit muss der Benutzer sowohl die Extraktion der Dokumente als auch die der relevanten Anforderungen aus diesen Dokumenten manuell und zeitaufwendig durchführen. Für eine automatisierte technische Lösung dieses Problems können Methoden des maschinellen Lernens verwendet werden. Diese werden im weiteren Verlauf umrissen.

2.3 Maschinelle Lernverfahren

Maschinelles Lernen ist so populär wie noch nie. Dank neuester Grafikkarten und leistungsstarker Multiprozessoren, können auch komplexere Probleme mit riesigen Datenmassen in überschaubarer Zeit bewältigt werden. Nachfolgend werden die für diese Arbeit relevanten Techniken erläutert.

2.3.1 Markov-Ketten

Markov-Ketten sind ein stochastisches Verfahren, um bedingte Übergangswahrscheinlichkeiten einzelnen Zustände zueinander modellieren zu können. Hängt der Übergang von einem Zustand zu einem anderen von einer Reihe von Ereignissen ab, kann dies mit Hilfe von Markov-Ketten abgebildet werden (siehe Abbildung 2.10).

Aufgrund der Abbildbarkeit von bedingten Wahrscheinlichkeiten, können auch Sequenzen wie beispielsweise Satzstrukturen gelernt werden, weshalb Markov-Ketten auch im Bereich der Textanalyse oder -generierung genutzt werden können [Kur 20, S. 167ff.]. Da dies bereits mit geringen Datenmengen realisierbar ist, im Vergleich zu komplexeren Lernverfahren wie z.B. Long short-term memory (kurz LSTM) [Hoc 97, S. 1735ff.] stellt dieses Lernverfahren das ideale Werkzeug für die vorliegende Arbeit dar.

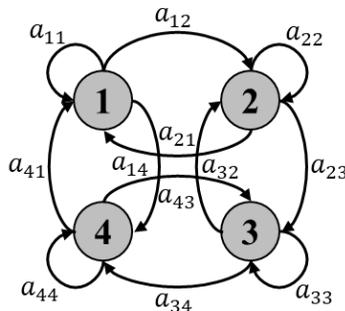


Abbildung 2.10: Beispiel einer Markov-Kette [Kur 20, S. 167ff.]

2.3.2 Convolutional Neural Network (CNN)

Ein an die Natur angelehntes Lernverfahren stellen Neuronale Netzwerke dar, welche die Neuronen des Menschlichen Gehirns nachbilden sollen, um damit selbständig Verfahren und Anomalien in den Eingabedaten zu erkennen [Rey 11]. Speziell im Bereich der Bildverarbeitung kommt eine Sonderform Neuronaler Netze zum Einsatz, die sogenannten Convolutional Neural Networks (CNN)⁴, zu Deutsch „Neuronale Faltungsnetze“. Im Vergleich zu normalen neuronalen Netzwerken, filtern diese zunächst die Eingangsdaten, um markante Merkmale (engl. Features) hervorzuheben (siehe Abbildung 2.11).

Die so extrahierten Features, werden anschließend verwendet, um das neuronale Netzwerk zu trainieren. Dadurch werden auch komplexere Strukturen erkannt. Diese Technologie kann ebenfalls zur Erkennung komplexer Satzstrukturen verwendet werden [Wan 12].

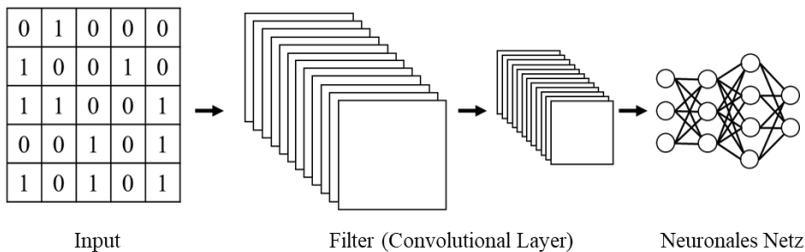


Abbildung 2.11: Darstellung eines CNN [Sup 18]

2.4 Computerlinguistische Grundlagen

Computerlinguistische Verfahren ermöglichen es dem Computer Texte in einzelne Worte zu zerlegen und diese hinsichtlich ihrer Wortart, Abhängigkeiten und möglicher Eigennamen zu klassifizieren [Bir 18]. Darauf aufbauend lassen

⁴ https://de.wikipedia.org/wiki/Convolutional_Neural_Network, zuletzt geprüft am 20.08.2020

sich höherwertige Klassifikatoren definieren, welche auch komplexe Satzstrukturen oder komplette Textpassagen klassifizieren können [Lid 01]. Stand heute werden neben der reinen Textklassifikation vermehrt große Sprachmodelle verwendet, um zusätzlich zur Identifikation der Wortarten oder Abhängigkeiten, auch semantische Informationen extrahieren zu können [Dev 18]. Es existieren zahlreiche Implementierungen in unterschiedlichen Programmiersprachen, welche speziell für die Analyse und Verarbeitung von Freitexten entwickelt wurden. Nachfolgend werden hierzu die Grundlagen erläutert und einige etablierte Implementierungen vorgestellt.

2.4.1 Zuordnung von Wortarten

Ein Wort oder ein Satzzeichen gehören immer einer Wortart an. Es gibt Fälle, bei welchen die Position im Satz eine Rolle für die Zugehörigkeit zu einer Wortart spielt. Bei der Zuordnung von Wortarten (engl. Part-of-speech-Tagging, POS) wird die Wortart eines Wortes im Kontext eines Satzes identifiziert und ausgegeben [Man 05, S. 155]. Hierzu werden die eingegebenen Sätze zunächst Tokenisiert (in die einzelnen Worte aufgeteilt) und Lemmatisiert (hierbei wird der Wortstamm eines jeden Wortes identifiziert), um anschließend die Wortart zu identifizieren (siehe Abbildung 2.12).

	Das Auto muss eine Höchstgeschwindigkeit von über 200 km/h haben.											
	[Artikel]		[Verb]		[Nomen]			[Präposition]		[Nummer]	[Hilfsverb]	
Wortarten		[Nomen]		[Artikel]					[Adverb]			[Adjektiv]

Abbildung 2.12: Beispiel POS-Tagging

2.4.2 Dependency Parsing

Neben der Erkennung von Wortarten, können auch Abhängigkeiten der einzelnen Worte im Kontext des eingegebenen Textes erkannt werden. Hiermit lässt sich identifizieren, welches Wort welche Rolle innerhalb eines Satzes einnimmt z.B. das Subjekt eines Satzes oder das Prädikat [Man 05, S. 428]. Je

nach zur Anwendung kommenden Textkorpora⁵ können hiermit unterschiedliche Abhängigkeitsbäume innerhalb eines Satzes identifiziert werden (siehe Abbildung 2.13).

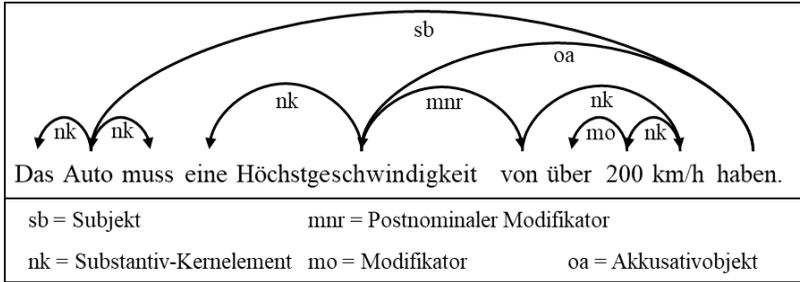


Abbildung 2.13: Beispiel eines Abhängigkeitsgraphen

2.4.3 Eigennamenerkennung

Die Eigennamenerkennung (engl. Named Entity Recognition, NER) hat zum Ziel, Eigennamen innerhalb eines Textes zu erkennen und einer vorbestimmten Entität zuzuordnen. Beispiele hierfür wären: Organisationen, Orte, Sprachen oder Personen (siehe Abbildung 2.14).

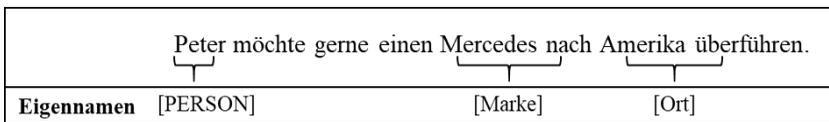


Abbildung 2.14: Beispiel einer Eigennamenerkennung

⁵ <https://de.wikipedia.org/wiki/Textkorporus>, zuletzt geprüft am 20.08.2020

2.4.4 Koreferenz-Auflösung

Ein weitere Anwendungsfall für die Computerlinguistik, ist die Auflösung von Koreferenzen innerhalb von Texten. Eine Koreferenz liegt dann vor, wenn eine Äußerung mehrfach innerhalb eines Textes mit unterschiedlichen Ausdrücken vorliegt [Soo 01]. Wie das Beispiel in Abbildung 2.15 aufzeigt, kann mit Hilfe dieses Verfahrens, automatisch erkannt werden, dass der Begriff „Er“ im zweiten Satz sich auf die Person „Peter“ aus dem ersten Satz bezieht.

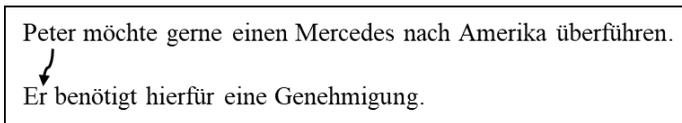


Abbildung 2.15: Beispiel einer Koreferenz

Technisch realisiert wird dies durch eine zweigeteilte Verarbeitung des Textes. In einem ersten Schritt werden Erwähnungen (engl. Mentions), welche potentielle Koreferenzen darstellen erkannt (siehe Abbildung 2.16). Anschließend werden die Mentions gepaart und die Paare hinsichtlich Ihrer Koreferenz-Wahrscheinlichkeit bewertet. Je höher dieser Wert, je höher die Wahrscheinlichkeit, dass es sich bei der bewerteten Paarung, um eine Koreferenz handelt.

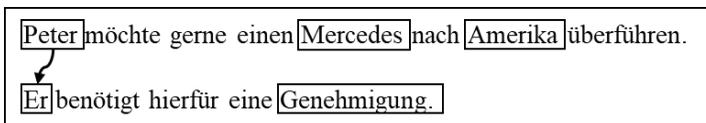


Abbildung 2.16: Beispiel einer Mention-Erkennung

Wie in diesem Beispiel dargestellt, werden insgesamt fünf Mentions erkannt, von denen lediglich die Paarung „Peter“ und „Er“ eine Koreferenz darstellen.

2.4.5 Wortvektoren

Um möglichst effizient mit Texten arbeiten zu können, wurden die sogenannten Wortvektoren entwickelt. Diese haben zum Ziel, alle Worte einer bestimmten Sprache innerhalb eines Vektorraums abzubilden [Mik 13]. Die deutsche Standardsprache umfasst ca. 75.000 Worte und der gesamte Wortschatz je nach Zählweise zwischen 300.000 bis 500.000 Worte. Würde man jedem Wort eine Dimension dieses Vektors zuordnen, wäre der resultierende Rechenaufwand enorm.

Die im Jahre 2013 veröffentlichte Methode Word2vec [Mik 13], versucht daher, mit Hilfe eines mehrere Hundert Dimensionen großen Vektorraums, jedem Wort einen korrespondierenden Vektor zuzuordnen. Typischerweise liegt die Dimension zwischen 100-1000 Dimensionen. Mit Hilfe dieses Verfahrens, ist es möglich mehr als 10 Millionen Worte zu lernen⁶.

2.4.6 Vortrainierte Sprachmodelle

2018 veröffentlichte Google die Bidirektionale Encoder-Darstellungen von Transformatoren (BERT) [Dev 18], ein vortrainiertes Sprachmodell, welches speziell dafür entwickelt wurde, mit nur wenigen Anpassungen für spezifische NLP-Aufgaben angewandt zu werden. Es wurde mit mehr als 3 Milliarden Worten trainiert und stellt damit ein umfassendes Wortverständnis zur Verfügung [Dev 18]. Seither wurden viele weitere Modelle veröffentlicht und die Performance immer weiter gesteigert wie z.B. GPT, RoBERTa, XLNet oder ELECTRA [Cla 20]. Die vorliegende Arbeit macht sich dies zu Nutze und verwendet einige auf diese Modelle aufbauende Implementierungen.

⁶ <https://de.wikipedia.org/wiki/Wortschatz>, zuletzt geprüft am 20.08.2020

2.4.7 NLP-Frameworks

spaCy ist eine Open-Source Python Bibliothek spaCy⁷ zum Einsatz. Sie bietet im Vergleich zu anderen etablierten NLP-Bibliotheken wie NLTK⁸ oder CoreNLP⁹ insbesondere für die deutsche Sprache erweiterte POS-Tags und Abhängigkeiten. Beispielsweise kann mit Hilfe von spaCy das Genitivobjekts¹⁰ – welche für die Identifikation des Elternsubjekts/-objekts benötigt wird – automatisch identifiziert werden. Zudem ist spaCy speziell für die Anwendung in industriellen Kontexten entwickelt worden, weshalb sie andere Implementierungen in den üblichen Aufgaben übertrifft [Coh 19].

SpaCy bietet für die deutsche Sprache zwei verschiedene Sprachmodelle: `de_core_news_sm` und `de_core_news_md`. Der Unterschied besteht in der Größe der Modelle, wodurch auch die Genauigkeit der Syntaxerkennung unterschiedlich ist. Das größere Modell ist `de_core_news_md` und damit auch das detailliertere, weshalb dieses in der vorliegenden Arbeit verwendet wird. Bei Updates von neueren spaCy-Versionen muss darauf geachtet werden, dass die Sprachmodelle ebenfalls verbessert werden. Dadurch können sich Syntaxerkennungen verändern, was die Implementierungsergebnisse beeinflussen könnte. Zudem bietet spaCy eine vorkonfigurierte Textklassifikations Pipeline, welche für kundenspezifische Anwendungsfälle trainiert werden kann.

Neben den vortrainierten CNN-Sprachmodellen und der integrierten Textklassifikations-Pipeline, bietet spaCy eine umfangreiche und einfach zugängliche Python-API, was die Integration in Business-Applikationen sehr vereinfacht. Die Performance, einfache Anwendung und die für die Deutsche Sprache vortrainierten Sprachmodelle sind die Gründe, weshalb die vorliegende Arbeit für die computerlinguistische Verarbeitung überwiegend spaCy nutzt.

⁷ <https://spacy.io/>, zuletzt geprüft am 20.08.2020

⁸ <https://www.nltk.org/>, zuletzt geprüft am 20.08.2020

⁹ <https://nlp.stanford.edu/>, zuletzt geprüft am 20.08.2020

¹⁰ <https://de.wikipedia.org/wiki/Genitiv>, zuletzt geprüft am 20.08.2020

2.5 Fazit

Ziel dieses Kapitels ist es, einen Überblick über die Grundlagen des Anforderungsmanagements und der Computerlinguistik zu vermitteln. Hierzu wird zunächst der Begriff des Produktlebenszyklus sowie sich davon ableitende Vorgehensmodelle zur Produktentwicklung erläutert und das Anforderungsmanagement in der Produktentstehung eingeordnet. Anschließend werden die wesentlichen Aufgaben des Anforderungsmanagements erläutert und deren wesentlichen Methoden und Werkzeuge beschrieben.

Da Anforderungen und das damit verbundene Wissen meist in textueller Form beschrieben werden, müssen diese für die weitere Verarbeitung zunächst computerlinguistisch aufbereitet werden. Stand heute werden zur Erkennung von Textpattern meist maschinelle Lernverfahren verwendet, welche daher grundlegend vorgestellt werden. Darauf aufbauend werden dem Leser die wesentlichen Grundlagen der Computerlinguistik erläutert und etablierte Textverarbeitungs-Frameworks vorgestellt.

3 Stand der Technik

Ziel dieses Kapitels ist die Analyse des aktuellen Stands der Technik und Forschung im Bereich des Anforderungs- und Wissensmanagements. Hierzu werden in einem ersten Schritt relevante Problemfelder abgeleitet, welche anschließend genauer analysiert werden. Abschließend werden alle identifizierten Probleme zusammengefasst und das weitere Vorgehen abgeleitet.

3.1 Ableitung der Problemfelder

Im Rahmen jeder Projektarbeit fallen unabhängig vom zu entwickelnden Produkt (Hardware, Software oder Dienstleistungen) zu dokumentierendes Wissen an in Form von Anforderungen, Aufgaben oder damit verbundene Informationen (z.B. Rechercheergebnisse) an. Vor allem Anforderungen sind besonders kritisch, da diese projektübergreifend verfügbar sein müssen, um eine erfolgreiche Projektdurchführung gewährleisten zu können. Häufig werden hierfür Office-Produkte zur Dokumentation der kritischen Informationen verwendet, welche im Gegensatz zu speziellen Softwarelösungen (z.B. IBM Rational DOORS und PTC Integrity) zwar das Sammeln und Clustern der Anforderungen ermöglichen, aber einige erhebliche Nachteile mit sich bringen können. Aber auch spezialisierte Softwarelösungen lösen nicht alle Problemstellungen. Nachfolgend werden die wesentlichen Herausforderungen des Anforderungsmanagements in Unternehmen unterschiedlicher Größe zusammengefasst [Ada 15, Ada 13, Gro 13]:

- Die Dokumentation von Anforderungen
- Integration von externen Anforderungsdokumenten
- Das Verknüpfen bzw. die Nachverfolgbarkeit von Anforderungen und dem damit in Verbindung stehenden Kontext
- Die Handhabung von Änderungen

- Die Wiederverwendung und Standardisierung von Anforderungen
- Die Ermittlung und Kommunikation von Anforderungen

Als Hauptherausforderung stellt der RE-Kompass [Ada 15] die Motivation zur Einführung und Einhaltung von Anforderungsmanagement-Methoden und Prozessen heraus. Hieraus schließt der Autor einen dringenden Bedarf nach kontextsensitiver Unterstützung der Anwender, um die Hemmschwelle zur Nutzung von AM-Methoden zu reduzieren. Da die Aufgaben des Anforderungsmanagements sehr umfangreich sind, wird der Stand der Technik nachfolgend hinsichtlich der Unterstützung der oben genannten Kernherausforderungen durchgeführt.

3.2 Unterstützung der Dokumentation

Die Dokumentation von projektrelevanten Informationen wie Anforderungen und dem dazugehörigen Wissen ist eine der primären und kritischsten Disziplinen des Anforderungsmanagements. Anforderungen die unvollständig, ungenau oder widersprüchlich formuliert sind, können zu Abweichungen des geplanten Produkts vom Kundenwunsch führen, was wiederum zu Mehrkosten oder gar dem Scheitern des Projekts führen kann.

Eine bereits in Kapitel 2.2.2 erwähnte Dokumentationstechnik ist die Anwendung von Satzschablonen [SOP 13, Fri 19], welche in vielen Werkzeugen zum Einsatz kommen. Eines der wohl umfänglichsten Beispiele hierfür ist die Software Systems Engineering Suite der Firma REUSE Company, dessen Teilmodul Authoring Tools¹ neben einer umfangreichen automatischen Qualitätskontrolle, auch den automatischen Abgleich des eingegebenen Textes mit vordefinierten Satzschablonen durchführt. Hierbei werden Anforderungen auf die Kriterien Vollständigkeit, Korrektheit und die grammatikalische und systemübergreifende Konsistenz hin untersucht. Um dies zu ermöglichen, werden

¹ <https://www.reusecompany.com/rat-authoring-tools>, zuletzt geprüft am 20.08.2020

die eingegebenen Texte mit Hilfe einer mehrschichtigen Ontologie und umfangreichen Wortlisten abgeglichen [Fra 15, S. 149ff.]. Zudem lassen sich mit Hilfe des Moduls Knowledge Manager², benutzerdefinierte Satzschablonen erzeugen, was die Dokumentation branchenspezifischer Anforderungen erleichtert. Ein Nachteil dieses Systems ist der hohe initiale Aufwand, um die Wortlisten mit branchenspezifischen Fachbegriffen zu füllen. Zudem erfordert es umfangreiche Schulungen und ist daher nicht als Werkzeug für alle Projektbeteiligten sinnvoll einsetzbar.

Die Software Requirements smells der Firma Qualicen³ verfolgt einen anderen Ansatz, um die Dokumentation von Anforderungen und Tests zu unterstützen. Hierbei werden Anforderungs- oder Testbeschreibungen welche zu lang sind, passiv Konstrukte beinhalten, mehrfach Negationen, ungenaue Ausdrücke oder Duplikate beinhalten entsprechend gekennzeichnet, woraufhin der Anwender entsprechende Korrekturen vornehmen kann. Dieses Werkzeug kommt zwar ohne Wortlisten aus, ermöglicht jedoch im Vergleich zur Systems Engineering Suite der Firma REUSE keine Definition standardisierter Satzschablonen und erfordert ebenfalls Expertenwissen, um die erkannten Fehler beheben zu können.

Die ReqSuite⁴ der Firma OSSENO kann Inkonsistenzen, Weakwords und andere Formale Fehler automatisch erkennen. Dies erfolgt jedoch in einem separaten Schritt nach der eigentlichen Anforderungsformulierung, was den Dokumentationsprozess verlangsamt. Zudem sind die angebotenen Analysefeatures, im Vergleich zu den zuvor beschriebenen Konkurrenten unterlegen und es fehlt ähnlich der Software Requirements smells die Möglichkeit, benutzerdefinierte Satzschablonen zu definieren.

Herr Körner [Kör 14] beschreibt in seiner Dissertation ein Konzept zur aktiven Unterstützung des Anforderungsmanagements durch automatische Qualitäts-

² <https://www.reusecompany.com/km-knowledge-manager>, zuletzt geprüft am 20.08.2020

³ <https://www.qualicen.de/de/>, zuletzt geprüft am 20.08.2020

⁴ <https://www.osseno.com/de/anforderungsmanagement-software/>, zuletzt geprüft am 20.08.2020

kontrollen von Anforderungen, der automatischen Extraktion von semantischen Modellen aus Freitexten und der Rückkopplung von Objektänderungen zurück in die textuelle Beschreibung von Anforderungen. Zur Erstellung der semantischen Modelle, werden die Informationen innerhalb des Textes mit Hilfe Thematischer Rollen klassifiziert [Gel 10]. Basierend auf den damit verbundenen Regeln, kann anschließend automatisch ein UML-Diagramm abgeleitet werden, welches die domänenspezifischen Informationen des Freitextes abbildet, wodurch der manuelle Erzeugungsaufwand reduziert wird. Für die vorliegende Arbeit stellt dieses Konzept daher eine mögliche Methode zur Integration von Domänenwissen, basierend auf Freitext dar.

In der Arbeit „RESI - A Natural Language Specification Improver“ [Kor 09, Kör 14] wird ein Konzept vorgestellt, welches Anforderungen hinsichtlich unterschiedlicher KPI's untersucht z.B. Mehrdeutigkeiten, Nominalisierungen oder das Fehlen von Argumenten. Zur Erkennung dieser Merkmale, wird eine Kombination aus Regeln und Ontologien wie z.B. WordNet [Fel 99] oder YAGO [Hof 14].

Im Rahmen der Arbeit von Frau Krisch wurden weitere Werkzeuge analysiert, welche sich auf die Analyse der Formalen-Qualität von Anforderungen fokussieren [Kri 17, S. 25ff.]. Sie selbst untersucht in ihrer Arbeit welche Merkmale Anforderungen vorweisen müssen, um testbar zu sein, was im Rahmen dieser Arbeit jedoch nicht weiter beleuchtet wird.

3.3 Unterstützung der Wissensintegration

Ein wesentlicher Teil der Produktentwicklung liegt in der Recherche und Erfassung von Anforderungen und Informationen an das geplante Zielsystem. Da häufig Office Dokumente zum Austausch von Anforderungen und Produktinformationen – z.B. in Form von Lasten- oder Pflichtenheften – verwendet werden, wurde 2004 das xml-basierte Standard Rule Interchange Format (RIF) veröffentlicht, um einen standardisierten Austausch von Anforderungen zu ermöglichen [Kif 08]. 2010 wurde daraus das Requirements Interchange Format

(ReqIF) abgeleitet [OMG 16]. Dieses kann von den meisten AM-Softwarelösungen exportiert und importiert werden, was den Datenaustausch drastisch vereinfacht. Jedoch nutzen viele Unternehmen noch immer keine AM-Softwarelösungen, weshalb weiterhin viele Anforderungen und die damit in Verbindung stehenden Systeminformationen in Form von Office-Dokumenten dokumentiert und verwaltet werden [Ada 13, Ada 15].

Um den mit der Integration dieser Dokumente verbundenen Aufwand zu reduzieren, existierend für den englischsprachigen Raum heute bereits einige Ansätze, um Anforderungen automatisch aus Freitextdokumenten zu extrahieren [Ray 16, Mu 09, Eza 18]. Als Quellsysteme werden hier unter anderem Lastenhefte oder Quellcode-Kommentar verwendet. Da es sich hierbei jedoch um reine Klassifikationsverfahren handelt, müsste die so extrahierten Anforderungen manuell ausgewertet und in die bestehenden Anforderungslisten übertragen werden.

Juristische Dokumente wie z.B. Normen, Richtlinien oder Gesetze sind neben Lastenheften die wichtigsten Quellen für Anforderungen, welche für das geplante Produkt berücksichtigt werden müssen. Um die darin enthaltenen Anforderung zu ermitteln bedarf es entweder entsprechender Experten oder aufwändiger Rechercharbeiten. Auch hierfür existieren bereits erste Ansätze, um die Extraktion von Anforderungen aus diesen Dokumenten zu automatisieren [Kiy 08, Bia 05, Dra 18, Bre 06]. Jedoch liegt auch hier der Fokus auf der grundsätzlichen Extraktion und Klassifikation und nicht auf der in Kontextstellung mit dem geplanten Zielsystem.

Um dieses Problem zu lösen, versuchen andere Ansätze, die extrahierten Anforderungen zu Clustern, welches die manuelle Zuordnung beschleunigen und somit den damit verbundenen Aufwand reduzieren soll [Aro 17]. Der manuelle Integrationsaufwand bleibt jedoch bestehen.

3.4 Unterstützung der Wissensverknüpfung

Ein großer Vorteil von spezialisierten Anforderungsmanagement-Lösungen gegenüber Office-Produkten ist die semantische Vernetzbarkeit von Anforderungen. In Werkzeugen wie z.B. IBM DOORS⁵ oder Atlassian Jira⁶ werden diese Verbindungen manuell erzeugt, was sehr zeitaufwändig ist und von den Anwendern einen sehr guten Gesamtüberblick über das Projekt erfordert, was je nach Komplexität oder Größe des Projektes nicht gegeben ist.

Andere Lösungen bieten daher automatische Relations-Erkennungs-Werkzeuge an wie z.B. das Modul Traceability Studio⁷ der Firma REUSE Company, welches nicht nur mögliche Links zwischen Anforderungen untereinander sondern auch Links zu Objekten in anderen Dokumenten z.B. UML-Diagrammen, Simulink-Modellen oder Klassen in Entwicklungsumgebungen identifiziert und vernetzen kann [REU 18]. Die dabei zum Einsatz kommenden Algorithmen basieren auf benutzerdefinierten Satzschablonen und umfangreichen Wortlisten, welche je nach Datenquelle entsprechend angepasst werden müssen, um mögliche Relationen zu erkennen. Der mit dieser Konfiguration verbundene zeitliche Aufwand und das dafür notwendige Expertenwissen, führen dazu, dass derartige Werkzeuge vorwiegend in Großunternehmen zum Einsatz kommen.

Projektmanagementlösungen wie Jira der Firma Atlassian bieten gegenüber reinen Anforderungsmanagementlösungen wie die Systems Engineering Suite der Firma REUSE Company den Vorteil, dass Anforderungen und Aufgaben zusammen verwaltet werden. Hierbei kann der Anwender, jeder Anforderung, die dazugehörigen Aufgaben manuell zuweisen, wodurch der Projektmanager einen besseren Überblick über den aktuellen Projektstatus ermöglicht wird. Dies setzt jedoch voraus, dass die betroffenen Projektmitarbeiter die entsprechenden Anforderungen kennen, was bei komplexeren Produkten jedoch nicht gewährleistet werden kann, weshalb Redundanzen oder Goldrandlösungen nur

⁵ <https://www.ibm.com/de-de/products/requirements-management>, zuletzt geprüft am 20.08.2020

⁶ <https://www.atlassian.com/de/software/jira>, zuletzt geprüft am 20.08.2020

⁷ <https://www.reusecompany.com/traceability-studio>, zuletzt geprüft am 20.08.2020

schwer vermieden werden können [Poh 15, S. 131]. Hierfür wären Automatismen ähnlich der Relationserkennung der Systems Engineering Suite der Firma REUSE Company notwendig, um alle relevanten Relationen automatisch zu erkennen.

Ein ganzheitlicher Überblick über alle projektrelevanten Kontextinformationen wird vor allem dann relevant, wenn sich Kundenanforderungen oder bestimmte Randbedingungen, zur Laufzeit eines Projektes ändern. Um die damit einhergehenden Auswirkungen abschätzen zu können, müssen die Projektverantwortlichen befähigt werden, alle von der Änderung betroffenen Elemente zu identifizieren, um die Tragweite sowie die damit verbundenen Risiken bewerten zu können. Aufgrund der Trennung des Managements projektrelevanter Kontextinformationen wie z.B. Anforderungen, Systembeschreibungen und Aufgaben sowie fehlender Assistenz in der Verknüpfung dieser Elemente untereinander, ist dies stand heute jedoch nicht sinnvoll möglich.

Ein möglicher Ansatz um Kontextinformationen ganzheitlich zu verknüpfen, stellen Ontologie basierte Ansätze dar. Der nach Wicaksono et. al. [Wic 12] beschriebene Ansatz verfolgt das Ziel, alle Produktmerkmale innerhalb einer Ontologie zu modellieren, welche es dem Anwender ermöglicht die Kunden- und Herstellersichten miteinander zu verknüpfen. Mit Hilfe des DIALOG-Frameworks sollen somit Wechselwirkungen zwischen Kundenwunsch und Produkthanforderungen systematisch identifiziert modelliert werden können und mögliche Fehler durch Änderungen der Kundenwünsche vermieden werden [EIM 12, Wic 12]. Darüber hinaus soll durch die Verknüpfung mit Feedback-Objekten, die Rückkopplung von Prozess- und Kundenfeedback zurück in die Produktentwicklung ermöglicht werden [Sch 10]. Dieses Verfahren erfordert jedoch manuellen Pflegeaufwand der Ontologie, weshalb dieser Ansatz für Nicht-Experten ungeeignet ist und entsprechender Assistenz bedarf.

3.5 Unterstützung der Änderungsdurchführung

Nur selten bleiben Anforderungen von Anfang bis Ende eines Projektes unverändert. In der Regel werden sie durch weitere ergänzt, verändert oder präzisiert. Je nach Einfluss der zu ändernden oder ergänzenden Anforderung, kann dies große Auswirkungen auf das Projekt haben, weshalb die Analyse der jeweiligen Tragweite der Änderung durchgeführt werden sollte. Neben Maßnahmen wie z.B. einem Change-Control Board, Freigabeprozesse oder durch Expertenbefragungen können diese Probleme reduziert werden, bedürfen jedoch einem großen manuellen Evaluationsaufwand [Poh 15, S. 140ff.].

Die Software Modern Requirements4DevOps bietet mit ihrem Impact Analyser die Möglichkeit sich alle mit der zu ändernden Anforderung direkt in Verbindung stehenden Elemente anzeigen zu lassen, wodurch es dem Anforderungsmanager erleichtert wird, den möglichen Einfluss der Änderung zu bewerten. Dies setzt jedoch voraus, dass alle entsprechenden Elemente miteinander vernetzt sind und dass der Prüfende alle betroffenen Elemente sowie deren Relevanz kennt. Denn nur so ist er im Stande, den mit der Änderung einhergehenden Einfluss oder die mögliche Schwere auf die jeweiligen Elemente bewerten zu können. Ein weiteres Problem hierbei ist, dass nur Elemente welche direkt mit dem zu ändernden Element verknüpft sind analysiert werden. Da heutige Produkte meist sehr komplex sind, kann davon ausgegangen werden, dass die direkt verbundenen Elemente wiederum andere Elemente beeinflussen, wodurch mit Hilfe dieses Vorgehens keine ganzheitliche Betrachtung möglich ist.

Sind die Anforderungen nicht untereinander vernetzt, bietet ein anderer Ansatz – welcher sich rein auf linguistische Änderungen von Anforderungen fokussiert – eine mögliche Lösung. Mit Hilfe von NLP-Methoden lässt sich identifizieren, welche anderen Anforderungen möglicherweise durch die Änderung der Anforderungsbeschreibung beeinflusst werden. Hierbei wird mit Hilfe von Ontologien ein möglicher Zusammenhang zwischen Anforderungen untereinander identifiziert. Anschließend wird untersucht, ob sich z.B. der Name eines Objekts geändert hat oder ob ein weiteres Objekt dazugekommen ist.

Insgesamt werden in „Change impact analysis for Natural Language requirements: An NLP approach“ [Aro 15], bis zu 14 unterschiedliche Änderungstypen unterschieden, welche je nach Zutreffenswahrscheinlichkeit entsprechend gewichtet und dem Anwender angezeigt werden. Mit Hilfe dieses Verfahrens lassen sich somit mögliche Risiken durch linguistische Veränderungen identifizieren, jedoch ähnlich dem Impact Analyser, keine ganzheitliche Betrachtung aller mit dem zu ändernden Element betroffenen Elemente.

3.6 Unterstützung bei der Wiederverwendung von bereits dokumentiertem Wissen

Ein über alle Branche verbreitetes Problem, ist die Wiederverwendung von bereits dokumentiertem Wissen aus vorangegangenen Projekten. Vor allem in Zeiten globaler Projektteams, wird dies immer relevanter. In „Automated support for reuse-based requirements engineering in global software engineering“ [Car 17] wird ein auf Semantik MediaWiki⁸ basierendes System beschrieben, welches einen zentralisierten Katalog wiederverwendbarer Anforderungen vorsieht, welcher für das gesamte Entwicklerteam zur Verfügung gestellt wird. Die Anforderungsverantwortlichen der einzelnen Projekte haben dadurch die Möglichkeit, ihren aktuellen Anforderungskatalog, mit den zentralisierten Anforderungen abzugleichen und relevante Anforderungen auszuwählen und in das eigene Projekt zu integrieren. Der zentralisierte Katalog kann zudem als Vorlage für neue Projekte verwendet werden und spart somit erheblich Zeit und Kosten, welche durch die initiale Anforderungserhebung entstehen würden [Car 17]. Da dieser Ansatz jedoch nur auf Anforderungen beschränkt ist, bietet er das Potenzial auch auf andere projektrelevante Informationen z.B. Rechercheergebnisse oder Aufgaben übertragen zu werden.

⁸ https://www.semantic-mediawiki.org/wiki/Semantic_MediaWiki, zuletzt geprüft am 20.08.2020

In „Automated Support to Capture Creative Requirements via Requirements Reuse“ [Do 19] wird ein anderer Ansatz verfolgt, um bereits dokumentierte Anforderungen wiederzuverwenden. Die Autoren dieser Arbeit nutzen öffentlich verfügbare Anforderungsdokumente, um diese als Inspiration bzw. zur Ideengenerierung für eigene Entwicklung zu nutzen. Es konnten noch weitere Ansätze zur Nutzung bereits dokumentierten Wissens identifiziert werden [siehe z.B. Pen 19, DON 15, San 15], welche jedoch für die vorliegende nicht relevant sind.

3.7 Unterstützung von Rechercharbeiten

Eine weitere relevantere Datenquelle ist das Internet, in dem sich produktrelevante Informationen nicht nur aus juristischen Quellen extrahieren lassen, sondern auch aus unzähligen anderen Datensetzen wie z.B. Wikipediaartikel [Lan 10, Reb 16], Fachforeneinträge [Aud 17, Zha 10] oder Produktreviews in Onlinestores [Yan 15, Qi 16]. Aber auch die für diese Quellen entwickelten Vorgehen haben das Problem, dass sie die extrahierten Informationen zwar extrahieren und klassifizieren können, diese jedoch nicht mit den bereits bestehenden Produktinformationen oder Anforderungslisten verknüpfen können.

In „A Study of the Importance of External Knowledge in the Named Entity Recognition Task“ [Sey 18] wird ein Konzept vorgestellt, welches mit Hilfe von Internetquellen, die Klassifizierung von Eigennamen verbessert. Hierzu werden überwiegend Wikipedia Artikel verwendet, welche mit den erkannten Objekten eines Satzes abgeglichen werden. Anschließend werden mit Hilfe von Nennungshäufigkeiten andere im Satz enthaltener Worte, die an den ehesten zutreffendsten Texten identifiziert. Anschließend können dem Anwender die entsprechenden Quellen angeboten werden. Dieses Verfahren beschränkt sich jedoch auf Wikipedia-Seiten und sieht zudem keine weitere Extraktion von möglichen relevanten Informationen aus den ermittelten Quellen vor.

3.8 Ableitung von Anforderungen

Wie zuvor beschrieben existiert bereits eine Vielzahl an Methoden und Konzepten zur Unterstützung des Anforderungs- und Wissensmanagements. Es konnten jedoch einige Probleme aufgedeckt werden, welche nachfolgend zusammengefasst und darauf basierend Anforderungen an das digitale Assistenzsystem abgeleitet werden:

Dokumentation

Problem 1: Projektrelevante Kontextinformationen wie z.B. Anforderungen, Systembeschreibungen und Aufgaben werden Stand heute in unterschiedlichen Softwarewerkzeugen gemanagt und sind daher nicht ganzheitlich miteinander vernetzt.

Anforderung 1: Das digitale Assistenzsystem muss fähig sein, alle projektrelevanten Kontextinformationen zu verwalten.

Problem 2: Spezialisierte Anforderungsmanagement-Werkzeuge sind aufgrund ihrer Komplexität und fehlenden Assistenz für Nicht-AM-Experten ungeeignet.

Anforderung 2: Das digitale Assistenzsystem sollte Nicht-AM-Experten befähigen, textuelle Beschreibungen standardisiert und vollständig formulieren können.

Informationsintegration

Problem 3: Etablierte Softwarelösungen unterstützen nur geringfügig bei der automatischen Informationsextraktion und -verknüpfung aus Freitexten.

Anforderung 3: Das digitale Assistenzsystem sollte fähig sein, Kontextinformationen automatisch aus Freitexten zu extrahieren.

Anforderungsvernetzung

Problem 4: Bisherige Ansätze zur automatischen Extraktion von Anforderungen aus Freitext können diese nicht kontextsensitiv in die bestehenden Anforderungslisten integrieren.

Anforderung 4: Das digitale Assistenzsystem sollte fähig sein, mögliche Relationen der automatisch extrahierten Kontextinformationen und der bereits existierenden Kontextelementen zu identifizieren.

Problem 5: Aktuelle Algorithmen zur Relationserkennung benötigen aufwändige domänenspezifische Anpassungen

Anforderung 5: Das digitale Assistenzsystem sollte fähig sein, mögliche Relationen der automatisch extrahierten Kontextinformationen und der bereits existierenden Kontextelementen zu identifizieren, ohne aufwändige domänenspezifische Anpassungen.

Tragweitenanalyse von Änderungen

Problem 6: Aufgrund der fehlenden Vernetzung aller projektrelevanten Informationen, ist keine ganzheitliche Analyse der Tragweite einer Änderung möglich.

Anforderung 6: Das digitale Assistenzsystem sollte dem Anwender die Möglichkeit bieten, die Tragweite einer Änderung ganzheitlich bewerten zu können.

Problem 7: Vor allem bei der Verwendung von Office-Lösungen für das Management von Wissen, werden alle Projektbeteiligten nicht über mögliche Änderungen informiert.

Anforderung 7: Das digitale Assistenzsystem muss fähig sein, alle Projektbeteiligten über mögliche Änderungen zu informieren.

Wiederverwendung von bereits dokumentiertem Wissen

Problem 8: Bisherige Ansätze zur Wiederverwendung beschränken sich auf Anforderungen und bieten keine Möglichkeit weitere damit in Verbindung stehende Informationen wiederzuverwenden.

Anforderung 8: Das digitale Assistenzsystem sollte fähig sein, bereits dokumentierte Informationen dem Anwender kontextsensitiv zur Verfügung zu stellen.

Unterstützung von Rechercharbeiten

Problem 9: Rechercharbeiten werden stand heute nicht aktiv unterstützt.

Anforderung 9: Das digitale Assistenzsystem sollte fähig sein, den Anwender aktiv bei der Recherche möglicherweise relevanter Informationen zu unterstützen.

3.9 Fazit

In diesem Kapitel werden die wesentlichen wissenschaftlichen Arbeiten und am Markt etablierten Softwarelösungen in den Bereichen des Anforderungs- und Wissensmanagements hinsichtlich möglicher Schwächen hin analysiert. Hierbei wird der Fokus auf die sechs einleitend beschriebenen Kernherausforderungen des Anforderungsmanagements in Unternehmen (siehe Kapitel 3.1)

durchgeführt. Hierbei werden wesentliche Schwächen und Forschungslücken identifiziert von denen abschließend Anforderungen an das Konzept dieser Arbeit abgeleitet werden. Diese zeigen vor allem den Bedarf nach kontextsensitiver Unterstützung der Anwender auf, um auch Nicht-Experten zu befähigen vollständige und semantisch verknüpfte Dokumentationen projektrelevanter Informationen erzeugen zu können.

Nachfolgend werden die in diesem Kapitel identifizierten Schwächen des Stands der Technik zusammengefasst. Hierzu werden am Markt etablierte Softwarelösungen und Forschungsprojekte hinsichtlich der Benutzerfreundlichkeit, der ganzheitlichen Informationsverwaltung und der Assistenz der relevanten Aufgaben während der Dokumentation, Informationsextraktion, Relationserkennung, Änderungskommunikation, Änderungsanalyse, Wiederverwendung und Recherche. Darüber hinaus werden die aktuellen Ansätze hinsichtlich der Domänenunabhängigkeit analysiert, da die meisten am Markt erhältlichen Lösungen domänenspezifische Konfigurationen benötigen um sinnvoll zu funktionieren (siehe Abbildung 3.1).

Werkzeug	Funktion										
	Benutzerfreundlichkeit (für Nicht-Experten)	Ganzheitliche Informationsverwaltung	Unterstützung bei der Dokumentation	Informationsextraktion aus Text-Dokumenten	Relationserkennung zwischen Informationen	Domänenunabhängigkeit der Relationserkennung	Assistenz bei der Analyse von Änderungstragweite	Kommunikation von Änderungen	Assistenz bei der Wiederverwendung	Assistenz bei der Recherche	
IBM Doors	⊖	⊖	⊕	⊖	⊖	n.a.	⊕	⊕	⊕	⊖	
PTC Integrity	⊕	⊖	⊕	⊖	⊖	n.a.	⊕	⊕	⊕	⊖	
REUSE Systems Engineering Suite	⊖	⊖	⊕	⊕	⊕	⊖	⊕	n.a.	⊕	⊖	
Qualicen, Requirements smells	⊕	⊖	⊕	⊖	⊕	⊕	⊖	n.a.	⊖	⊖	
OSSENO, ReqSuite	⊕	⊖	⊕	⊖	⊖	n.a.	⊖	n.a.	⊖	⊖	
Modern Requirements4DevOps	⊕	⊖	⊕	⊖	⊖	n.a.	⊕	⊕	⊕	⊖	
Semantik MediaWiki	⊖	⊕	⊕	⊖	⊖	n.a.	⊖	⊖	⊖	⊖	
Atlassian Jira and Confluence	⊕	⊕	⊕	⊖	⊖	n.a.	⊖	⊕	⊖	⊖	
Diese Arbeit	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	

Abbildung 3.1: Zusammenfassung des Stands der Technik

Analog zum Stand der Technik werden nachfolgenden verwandte Forschungsarbeiten hinsichtlich der gleichen Funktionalitäten analysiert und bewertet (siehe Abbildung 3.2):

<div style="border: 1px solid black; padding: 5px; display: inline-block;"> <p>⊕ vollständig</p> <p>⊙ teilweise</p> <p>⊖ garnicht</p> <p>n.a. nicht bekannt</p> </div>	Funktion	Benutzerfreundlichkeit (für Nicht-Experten)	Ganzheitliche Informationsverwaltung	Unterstützung bei der Dokumentation	Informationsextraktion aus Text-Dokumenten	Relationserkennung zwischen Informationen	Domänenunabhängigkeit der Relationserkennung	Assistenz bei der Analyse von Änderungstragweite	Kommunikation von Änderungen	Assistenz bei der Wiederverwendung	Assistenz bei der Recherche
ambiverse-nlu [Sey 18]	⊕	⊖	⊖	⊖	⊙	⊕	⊙	⊖	⊖	⊙	⊕
Fraga [Frau 15]	⊖	⊕	⊕	⊖	⊕	⊖	⊖	⊖	⊕	⊖	⊖
RECAA / RESI [Kor 09, Kör 14]	⊖	⊖	⊕	⊕	⊕	⊙	⊖	⊖	⊖	⊖	⊖
Krisch [Kri 17]	⊖	⊖	⊕	⊖	⊖	⊖	⊙	⊖	⊖	⊖	⊖
DIALOG [EIM 12, Wic 12]	⊖	⊕	⊙	⊖	⊖	⊖	⊙	⊕	⊕	⊖	⊖
Arora [Aro 17]	⊖	⊖	⊙	⊖	⊖	⊕	⊕	⊙	⊙	⊖	⊖
PANTALASA [Car 17]	⊖	⊖	⊙	⊖	⊙	⊖	⊖	⊖	⊙	⊕	⊖
[Do 19]	⊖	⊖	⊙	⊙	⊙	⊕	⊖	⊖	⊖	⊕	⊕
Diese Arbeit	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕

Abbildung 3.2: Zusammenfassung des Stands der Forschung

4 Konzept

In diesem Kapitel wird ein Konzept eines digitalen Assistenzsystems vorgestellt, welches die Aufgaben des Anforderungs- und Wissensmanagements projektrelevanter Informationen kontextsensitiv unterstützen soll. Das Kernziel dieses Konzepts ist die ganzheitliche Verknüpfung aller Kontextinformationen (im Folgenden Kontextelemente genannt).

Definition 3: Kontextelemente

Kontextelemente sind alle im Rahmen eines Projektes zu verwaltenden Informationen wie z.B. Anforderungen, Systemkomponenten, Aufgaben oder allgemeine Informationen.

Für die ganzheitliche Verwaltung aller Kontextelemente (Anforderung 1) sieht das vorliegende Konzept ein selbstentwickeltes semantisches Datenmodell vor (siehe Kapitel 4.1). Um die Dokumentation auch für Nicht-AM-Experten zu ermöglichen (Anforderung 2) sieht das Konzept einen Dokumentationsassistenten vor (siehe Kapitel 4.2). Stand heute werden Anforderungen überwiegend in Anforderungsdokumenten (z.B. Lasten- und/oder Pflichtenhefte) verwaltet, welche häufig von Hand extrahiert werden (Anforderung 3). Um diese in die semantische Datenstruktur zu überführen sieht das vorliegende Konzept einen Integrationsassistenten vor (siehe Kapitel 4.3).

Für die Unterstützung bei Rechercharbeiten (Anforderung 9) sieht das Konzept einen Rechercheassistenten vor, welcher kontextsensitiv Informationen aus dem Internet bereitstellt (siehe Kapitel 4.4). Um alle Kontextelemente miteinander zu verknüpfen ist normalerweise ein hoher manueller Aufwand von Nöten. Das vorliegende Konzept sieht daher einen Verknüpfungsassistenten vor (Anforderung 4), welcher den Anwender bei der Verknüpfung neuer Informationen mit bereits dokumentierten unterstützt (siehe Kapitel 4.5).

Änderungen sind je nach Projektstatus mit mehr oder weniger kritischen Auswirkungen verbunden, welche durch einen Einzelnen nicht immer überblickt werden können. Um den Anwender hierbei zu unterstützen sieht das Konzept einen Änderungsassistenten vor, welcher eine automatische Abschätzung des Änderungseinflusses vornimmt (Anforderung 6), um dem Anwender die Abschätzung der Tragweite zu erleichtern (siehe Kapitel 4.6).

Bereits dokumentierte Informationen, vor allem in Form von Office-Dokumente, finden nur selten Wiederverwendung, da der Explorer nur mäßige Suchfunktionen bietet. Dank der ganzheitlichen Abbildung aller projektrelevanter Informationen, sieht das vorliegende Konzept einen Wiederverwendungsassistenten vor, welcher kontextsensitiv bereits dokumentierte Inhalte zur Integration in neue Projekte (Anforderung 8) bereitstellt (siehe Kapitel 4.6). In Summe bilden diese Assistenzmodule das digitale Assistenzsystem zur kontextsensitiven Unterstützung des Anforderungs- und Wissensmanagements (siehe Abbildung 4.1).

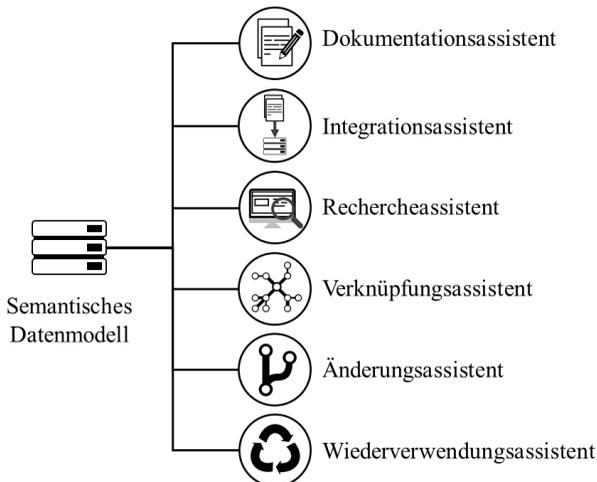


Abbildung 4.1: Module des Konzepts

Nachfolgend werden die einzelnen Module des digitalen Assistenzsystems genauer beschrieben.

4.1 Semantisches Datenmodell

Die ganzheitliche Verwaltung aller Kontextelemente birgt große Potentiale in der Automatisierung aufwändiger manueller Aufgaben (wie z.B. die automatische Verknüpfung oder unterstützte Wiederverwendung von Informationen) gegenüber den aktuell auf dem Markt erhältlichen Softwarelösungen im Anforderungs- und Wissensmanagement. Um alle projektrelevanten Informationen wie z.B. Anforderungen, Aufgaben oder damit verbundene Informationen zu dokumentieren, sieht das semantische Datenmodell eine abstrakte Klasse „Kontextelement“ als generische Oberklasse aller Kontextelemente vor (siehe Abbildung 4.2).

Relationen als auch Kontextelemente haben im Wesentlichen eine eindeutige Identifikationsnummer (kurz ID), ein Erstelldatum, einen Namen, eine Beschreibung, eine Versions- und eine Subversions-Nummer sowie die boolesche Eigenschaft aktuell oder veraltet zu sein. Zudem sollten der Autor sowie der oder die Verantwortliche(n) bekannt sein. Da dies für Relationen als auch Kontextelemente gleichermaßen zutrifft, werden diese Attribute durch die abstrakte Klasse „Informationselement“ repräsentiert (siehe Abbildung 4.2).

Jedes Kontextelement kann mehrere Relationen haben, welche aus zwei Kontextelementen und einem Relationstypen bestehen. Hiermit lassen sich jeweils zwei Kontextelemente durch unterschiedliche Relationen miteinander verbinden. Zusätzlich zu den Relationen haben Kontextelemente einen Status und eine Priorität. Je nachdem kommen Kontextelemente in ähnlicher oder gleicher Form unter unterschiedlichen Namen vor. Werden derartige Redundanzen identifiziert, werden diese zusammengefasst und die synonymen Namen unter dem Attribut Synonyme gespeichert. Zuletzt sieht das Konzept für jedes Kontextelement einen Anhang vor, welcher genutzt werden kann, um ergänzende Dokumente anhängen zu können wie z.B. Bilder, Diagramme oder technische Zeichnungen (siehe Abbildung 4.2).

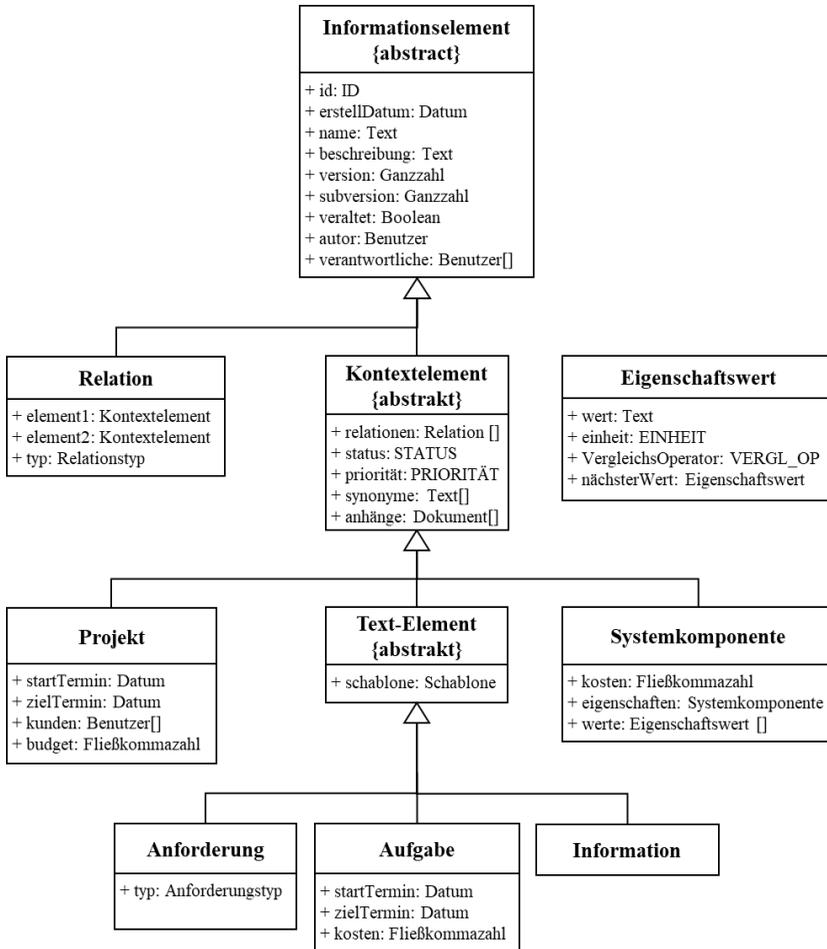


Abbildung 4.2: Semantisches Datenmodell

Nachfolgend werden alle Kontextelement- sowie deren möglichen Relationsarten genauer beschrieben.

4.1.1 Kontextelemente

Gemäß Definition 3 (auf Seite 79) umfasst die Klasse der Kontextelemente alle im Rahmen eines Projektes zu verwaltenden Informationen. Jedes dieser Kontextelemente kann durch unterschiedliche Relationsarten mit anderen Kontextelementen verknüpft werden, welche nachfolgend in Tabelle 2 definiert werden.

Relation	Grafische Darstellung
Ableitung	
<p>Mit Hilfe dieser Relation lässt sich zu einem späteren Zeitpunkt nachvollziehen, welche Kontextelemente als Kopie eines anderen Kontextelements entstanden sind. Dies ist z.B. bei der Wiederverwendung von Kontextelementen in anderen Projekten relevant, um den Ursprung des jeweiligen Kontextelements nachvollziehen zu können (siehe Kapitel 4.6).</p>	
Hierarchie	
<p>Mit Hilfe dieser Relation lassen sich hierarchische Abhängigkeiten zwischen zwei Kontextelementen abbilden z.B. „Tür ist Teil von Auto“ oder „Anforderung_XY ist Teil von Projekt_A“.</p>	
Irrelevant	
<p>Kontextelemente eines Projektes welche irrelevant für ein anderes Projekt sind, können mit Hilfe einer Irrelevant-Relation versehen werden, um diese bei der Suche von möglichen Verknüpfungen oder einer potentiellen Wiederverwendung unwichtige Elemente auszufiltern (siehe Kapitel 4.6).</p>	
Nicht-Redundanz	
<p>Kontextelemente welche fälschlicher Weise als redundant erkannt wurden, können mit Hilfe dieser Relation als „nicht-Redundant“ gekennzeichnet werden.</p>	
Überarbeitung	

<p>Um Änderungen rückwirken nachvollziehen zu können, bedarf es einer Versionierung und Persistierung aller Änderungen. Hierzu dient die Überarbeitungs-Relation, welche darstellt, welches Kontextelement von welcher Vorgängerversion abstammt (siehe Kapitel 4.1.6).</p>	
<p>Zusammenfassung</p>	
<p>Um Redundanzen oder sich ähnelnde Versionen zusammenfassen zu können, kann eine Zusammenfassung durchgeführt werden. Hierbei wird ein Element durch den Anwender ausgewählt welches die Basis für die Zusammenführung darstellt (in diesem Fall das obere). Alle Informationen des unteren Kontextelements sowie alle damit in Verbindung stehenden Kontextelemente werden auf die neue Version des oberen Elements übertragen und der Name als Synonym gespeichert (siehe Kapitel 4.5).</p>	

Tabelle 2: Kontextelementrelationen (OMG 15, OMG 18)

4.1.2 Projekt

Der Begriff Projekt kann für Unternehmen unterschiedliche Bedeutungen haben. Während es sich in einem Fall um Entwicklungsprojekte handelt, können Projekte im anderen Fall Bestellungen oder Dienstleistungen repräsentieren. In jedem Fall können einem Projekt mehrere Kontextelemente zugehörig sein welche je nachdem auch Teilprojekte darstellen können, welche wiederum mehrere Kontextelemente beinhalten können.

Für das vorliegende Konzept sind Projekte ausgeprägt durch Kunden, Start- und Zieltermin sowie ein definiertes Budget. Je nachdem können auch weitere Merkmale eines Projektes von Interesse sein, welche jedoch für das vorliegende Konzept nicht berücksichtigt werden.

4.1.3 Text-Elemente

Anforderungen, Aufgaben und damit verbundene Informationen werden maßgeblich durch ihre textuelle Beschreibung definiert, wofür jeweils entsprechende Textschablonen zur Standardisierung zum Einsatz kommen können. Folglich lassen sich diese Kontextelemente unter der abstrakten Oberklasse „Text-Element“ zusammenfassen. Da Text-Elemente ebenfalls Teil der Menge aller Kontextelemente sind, können diese wie in Tabelle 2 dargestellt mit anderen Kontextelementen verknüpft werden und Projekten zugeordnet oder für irrelevant markiert werden.

Anforderungen dienen der Spezifikation von Funktionen, Eigenschaften oder bestimmte Randbedingungen. Aufgaben können die Realisierung von Systemkomponenten oder die Verifikation von Anforderungen zum Ziel haben. Daher folgen für Text-Elemente, ergänzend zu Tabelle 2, die in Tabelle 3 definierten Relationen.

Relation	Grafische Darstellung
Realisierung	
Spezifizierung	
Verifikation	

Tabelle 3: Text-Element-Relationen [OMG 15, OMG 18]

Anforderungen haben neben den Attributen, welche bereits durch die Oberklassen Informations- und Kontextelement definiert wurden, eine Priorität, einen Anforderungstyp und einen Anforderungsstatus. Je nach Detaillierungsgrad, können Anforderungen noch weitere Attribute haben, welche jedoch für das vorliegende Konzept nicht berücksichtigt werden (siehe hierzu auch Kapitel 2.2.4).

Um Aufgaben planen zu können, bedarf es einem Start- und Endtermin und prognostizierte Kosten bzw. der prognostizierte Aufwand (siehe Abbildung 4.2). Um auch größere Arbeitspakete abbilden zu können, lassen sich Aufgaben mit Hilfe der Hierarchie-Relation (siehe Kapitel 4.1.1) anderen Aufgaben unterordnen. Somit lassen sich Arbeitspakete auf unterschiedlichen Ebenen definieren, ohne weitere Klassentypen einführen zu müssen.

In Bezug auf das Wissensmanagement können die Arten von Informationen mannigfaltig sein. Hinsichtlich des Anforderungsmanagements sind vor allem Begriffsdefinitionen oder die Dokumentation von den zugrundeliegenden Problemen ein äußerst hilfreiches Werkzeug, um die Projektübergreifende Kommunikation von Anforderungen sicherzustellen. Für das vorliegende Konzept werden daher alle Textinformationen zu einer Systemkomponente, welche keine Anforderung oder Aufgabe darstellen, als Information bezeichnet.

4.1.4 Satzschablonen

Das vorliegende Konzept sieht eine kontextsensitive Hilfestellung vor, basierend auf Satzschablonen. Da sich bestimmte Teile der Schablonen wiederholen können z.B. Bedingungsschablone (siehe Anhang 9.1), kann eine Satzschablone aus mehreren Schablonen bestehen. Die einzelnen Glieder der Schablonen wiederum bestehen aus semantischen Rollen (siehe hierzu Kapitel 4.2.1) welche einzelnen semantischen Teile des Satzes repräsentieren. Um kontextsensitive Wortvorschläge generieren zu können, hat jeder Satzteil entsprechend seiner Rolle ein Referenzbeispiel in Form eines Textes. Zudem hat es einen Pointer auf die darauffolgenden Satzteile, welche selbst wiederum Satzschablonen sein können (siehe Abbildung 4.3).

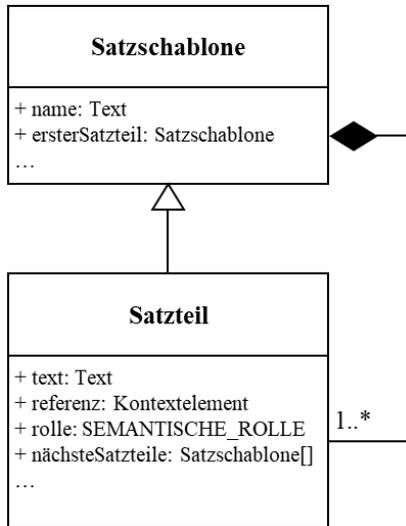


Abbildung 4.3: Satzschablonendatenmodell

4.1.5 Systemkomponenten

Systemkomponenten repräsentieren Teile des Zielsystems oder das Zielsystem selbst (z.B. der Motor des Autos oder das Auto selbst). Ihr Ziel ist es die an das Zielsystem gestellten Anforderungen zu erfüllen, wobei eine Systemkomponente nur einen Teil einer Anforderung oder mehrere Anforderungen gleichzeitig erfüllen kann.

Relation	Grafische Darstellung
Verwirklichung	

Tabelle 4: Systemkomponenten spezifische Relationen [OMG 18]

Jede Systemkomponente kann Kosten (z.B. Material- oder Beschaffungskosten) und Eigenschaften (z.B. die Farbe des Autos) haben. Jede Eigenschaft hat

einen Wert, eine Einheit und einen Operator. Der Operator, gibt die Grenzen des Wertes (z.B. „größer“, „kleiner“, „gleich“, „größer gleich“, „kleiner gleich“, „und“, „zwischen“, siehe auch Anhang 9.1). Diese lassen sich aus nichtfunktionalen Anforderungen extrahieren z.B. „Die Höchstgeschwindigkeit des Autos muss größer 100km/h betragen“. In diesem Fall wäre die Eigenschaft die „Höchstgeschwindigkeit“ der Systemkomponente „Auto“ mit dem Wert „100“ und der Einheit „km/h“ (siehe Abbildung 4.4).

Es gibt jedoch Sonderfälle, in denen Eigenschaften auch gleichzeitig Systemkomponenten sind z.B. „Die Farbe des Autos muss...“ und „die Pigmente der Farbe des Autos müssen...“. Wie dieses Beispiel zeigt, beeinflusst die Sichtweise ob es sich um eine Eigenschaft oder eine Systemkomponente handelt.

Aus diesem Grund, sieht das vorliegende Konzept vor, dass eine Systemkomponente, auch gleichzeitig eine Eigenschaft einer anderen Systemkomponente sein kann (z.B. „die Farbe des Autos“ mit Farbe gleichzeitig Systemkomponente als auch Eigenschaft).

Eigenschaftswert
+ wert: Text
+ einheit: EINHEIT
+ VergleichsOperator: VERGL_OP
+ nächsterWert: Eigenschaftswert

Abbildung 4.4: Datenmodell Eigenschaftswert

Um auch logisch verkettete Wertebereiche modellieren zu können, haben Eigenschaftswerte neben ihren Werten, Einheiten und Vergleichsoperatoren (kurz „Operator“) auch einen „nächstenWert“. Mit Hilfe dieses nächsten Eigenschaftswertes, können logisch miteinander verkettete Eigenschaftswerte abgebildet werden (siehe Abbildung 4.5).

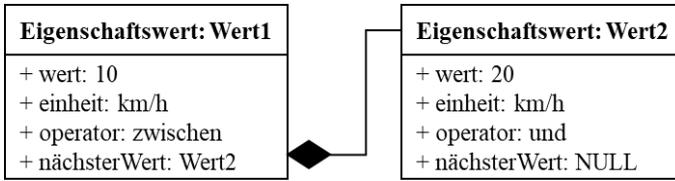


Abbildung 4.5: Logisch verkettete Eigenschaftswerte

Wie hier dargestellt, lassen sich in dieser Beispielanforderung zwei Eigenschaftswerte identifiziert „Wert1“ und „Wert2“. Damit diese zu einem späteren Zeitpunkt automatisch ausgewertet werden können, werden diese in der entsprechenden Reihenfolge miteinander verknüpft.

4.1.6 Versionierung

Ein der Hauptherausforderungen bei der Persistenz kritischer Informationen ist die Nachverfolgbarkeit von Änderungen und die Sicherstellung der Plausibilität der Informationen untereinander. Müssen beispielsweise aus bestimmten Gründen ältere Versionen eines Kontextelementes wiederhergestellt werden, kann nicht zwingend davon ausgegangen werden, dass alle, aktuell mit dem zu ändernden Kontextelement in Verbindung stehenden Kontextelemente, weiterhin in der aktuellen Relation bestehen bleiben können. Das vorliegende Konzept sieht daher mehrere Mechanismen vor, um zum einen die Nachverfolgbarkeit als auch die Plausibilität der Daten zu gewährleisten.

Neue Versionen von Kontextelementen können auf der aktuellsten aber auch auf Basis alter Versionen entstehen. Um hierbei überblicken zu können, welche Version von welcher abgeleitet wurde, werden jeder neuen Iteration eine Versions- und Subversionsnummer zugewiesen. Die Versionsnummer gibt die Tiefe des Versionsbaums an, hingegen die Subversion angibt, wie viele Subversionen es von der vorangegangenen Version gibt (siehe Abbildung 4.6).

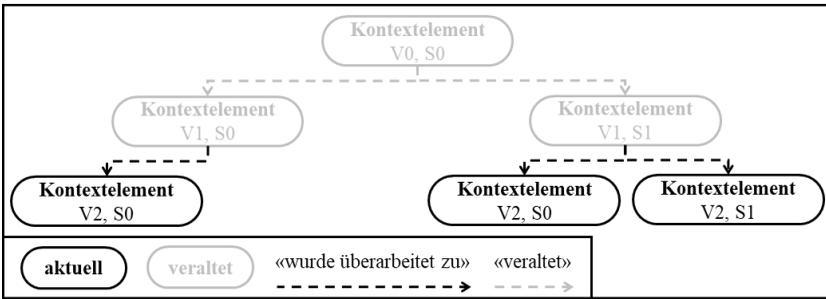


Abbildung 4.6: Historie eines Kontextelements

Zudem werden die Vorgänger Versionen nicht gelöscht, sondern lediglich der boolesche Wert „veraltet“ auf wahr gesetzt und beide Versionen mit Hilfe der Überarbeitungsrelation miteinander verbunden. Somit lässt sich zu jedem Zeitpunkt nachvollziehen, welche Version von welcher abgeleitet wurde und wie viele Subversionen einer bestimmten Version abgeleitet wurden. Um auch kontextelementübergreifend die Plausibilität der Daten zu gewährleisten, werden auch die Relationen zu anderen Kontextelementen nicht gelöscht bei der Erzeugung neuer Versionen, sondern die neue Version des Kontextelements erhält dieselben Relationen wie der Vorgänger. Dies ermöglicht es auch rückwirkend sicherzustellen, dass im Falle einer Wiederherstellung veralteter Versionen, plausible Relationen zu anderen Kontextelementen sicherzustellen (siehe Abbildung 4.7).

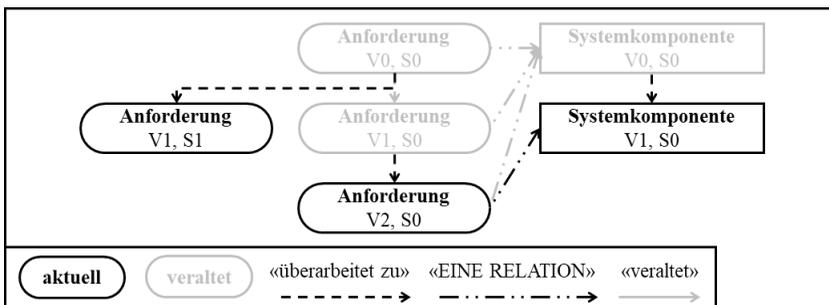


Abbildung 4.7: In Verbindung stehende Kontextelemente

Wie in dieser Abbildung zu sehen ist, existieren insgesamt vier Versionen der Anforderung und zwei Versionen der Systemkomponente. Version „V1, S1“ ist mit keiner Version der Systemkomponente verbunden hingegen die Versionen „V0, S0“ bis „V2, S0“ mit „V0, S0“ der Systemkomponente in Relation stehen. Mit „V1, S0“ der Systemkomponente steht jedoch nur die neueste Version der Anforderung in Relation. Würde nun eine der älteren Versionen der Anforderung wiederhergestellt werden müssen, wäre dies nur dann sicher möglich, wenn gleichzeitig die ältere Version der Systemkomponente genutzt werden würde, da die älteren Versionen der Anforderung zu keinem Zeitpunkt mit der neuesten Version der Systemkomponente in Relation standen und daher nicht sichergestellt werden kann, dass diese Versionen ohne Probleme miteinander funktionieren.

Dank dieses Versionierungsverfahrens können derartige Probleme aufgedeckt werden und dem Anwender dargestellt werden. Würde man beispielsweise „V0, S0“ der Anforderung wiederherstellen wollen, wäre dies lediglich mit „V0, S0“ der Systemkomponente möglich ohne eine Plausibilitätsprüfung durchführen zu müssen, da nur in dieser Version eine Verbindung freigegeben wurde (siehe Abbildung 4.7).

4.1.7 Zusammenführen zweier Kontextelemente

Neben der Versionierung von Kontextelementen, ist auch die Zusammenfassung redundanter oder ähnlicher Kontextelemente – vor allem bei größeren Entwicklungsteams – eine wichtige und kritische Funktionalität. Ähnlich dem „Merg“-Vorgang¹ (engl. für zusammenführen) in der Softwaretechnik sieht das vorliegende Konzept daher einen Zusammenfassungsprozess für derartige Fälle vor.

Hierzu wählt der Anwender im ersten Schritt eines der beiden zu verbindenden Kontextelemente aus, in welches alle Informationen des anderen Kontextelements überführt werden sollen. Anschließend werden alle Relationen beider

¹ <https://de.wikipedia.org/wiki/Merge>, zuletzt geprüft am 20.08.2020

4.2 Dokumentationsassistent

Die Kernaufgabe des Dokumentationsassistenten besteht darin, den Anwender bei der Dokumentation projektrelevanter Informationen kontextsensitiv zu unterstützen, um auch Nicht-Experten zu befähigen vollständige und standardisierte textuelle Beschreibungen zu erstellen (Anforderung 2).

Um dies zu gewährleisten, wird die Freitexteingabe des Anwenders kontinuierlich mit Hilfe der semantischen Rollenerkennungs-Funktion analysiert. Hierbei werden den einzelnen Satzteilen semantische Rollen zugewiesen, welche die jeweilige Funktion der Satzteile innerhalb des Satzes repräsentieren. Anschließend wird der aktuelle Satz mit Hilfe der Textklassifikations-Funktion klassifiziert, um zu identifizieren um welche Satzart es sich gerade handelt.

Anschließend können mit Hilfe der Autovervollständigungs-Funktion kontextsensitive Wortvorschläge generiert werden, welche neben reinen Textbausteinen, auch unmittelbare Objektreferenzen zu bereits dokumentierten Objekten in der semantischen Datenbank darstellen können (siehe Abbildung 4.9).

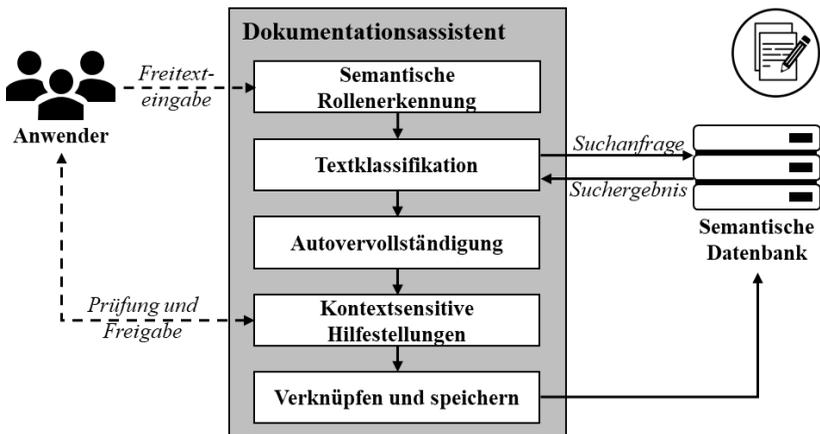


Abbildung 4.9: Aufbau des Dokumentationsassistenten

Basierend auf der aktuell geschätzten Satzart und der prognostizierten nächsten semantischen Rolle, werden kontextsensitive Hilfestellungen abgeleitet, welche dem Anwender zusätzlich zu den Wortvorschlägen, bereitgestellt werden. Nach Beenden der Texteingabe, werden die identifizierten Informationen sowie möglicherweise bereits integrierte Objektreferenzen in der semantischen Datenbank gespeichert und ggf. bereits miteinander verknüpft.

Nachfolgend werden die in Abbildung 4.9 dargestellten Funktionen einzelnen beschrieben.

4.2.1 Semantische Rollenerkennung

Eine der Kernfunktionen des Dokumentationsassistenten ist die automatische Identifikation von Informationen und deren Relationen untereinander innerhalb von Freitext (Anforderung 3). Informationen innerhalb eines Satzes, werden hierzu zunächst hinsichtlich ihrer semantischen Rollen analysiert. Anschließend wird darauf aufbauend, nach satzübergreifenden Relationen dieser Informationen gesucht, um mögliche Zusammenhänge zu identifizieren.

Dabei werden die innerhalb eines Satzes befindlichen Satzteile hinsichtlich ihrer Wortarten und Satzglieder untersucht (siehe Abbildung 4.10).

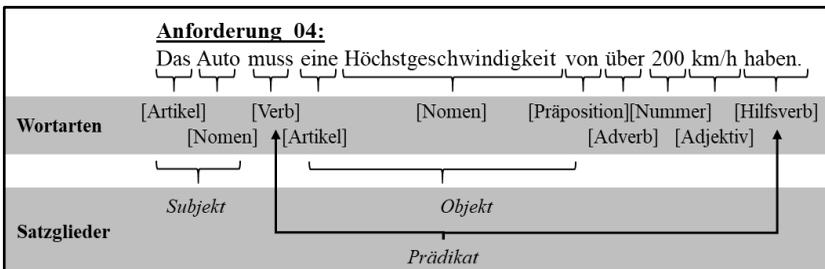


Abbildung 4.10: Computerlinguistisch aufbereiteter Satz

Je nach Art des zu analysierenden Textes, können die einzelnen Worte unterschiedliche semantische Rollen einnehmen. Da es sich in diesem Beispiel, um

eine Anforderung handelt, nimmt der vordere Teil des Prädikats die semantische Rolle des Prioritätswortes und der hintere Teil die semantische Rolle des Prozesswortes ein. Zudem nimmt das Objekt in diesem Beispiel die semantische Rolle einer Eigenschaft ein (siehe Abbildung 4.11).

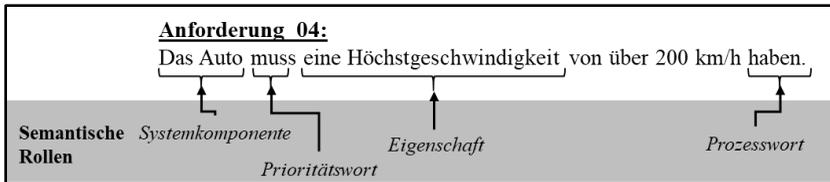


Abbildung 4.11: Exemplarische Zuordnung semantischer Rollen

Gemäß der Anforderungsschablonen nach SOPHIST [SOP 13] als auch der an [moX 18] angelehnten Aufgabenschablonen (siehe Anhang 9.1) leiten sich in Summe folgende semantische Rollen für das vorliegende Konzept ab:

Semantische Rolle	Definition
Akteur	Ausführende(s) Person oder Objekt einer Tätigkeit.
Datum	Eine Datumsangabe innerhalb der textuellen Beschreibung z.B. 01.01.2020
Eigenschaft	Ein zum Wesen eines Bezugsobjekts (kurz BZO) gehörenden Merkmals.
Eigenschaftswert	Zahlenwert mit semantisch passender Einheit wie z.B. Zeit, Gewicht, Längenmaß etc. oder eine Wortgruppe, welche einen Eigenschaftswert repräsentiert.
Einheit	Maßeinheit für physikalische Größen
Eltern-Objekt	Hierarchisch übergeordnetes Objekt des Objekts.
Eltern-Subjekt	Hierarchisch übergeordnetes Objekt des Subjekts.
Funktionswort	Standardausdrücke innerhalb von Anforderungen wie z.B. „die Möglichkeit bieten“, „in der Lage sein“ etc.
Logischer Operator	Wort oder Zeichen, welche mehrere Informationen innerhalb eines Satzes logisch miteinander verknüpfen z.B. „und“ oder „oder“.
Objekt	Das Objekt des Satzes

Person	Eine Person oder Rolle, welche Verantwortlich für eine Aufgabe oder Akteur innerhalb einer Anforderung sein kann.
Präzisierung	Ausführende Ergänzungen in Textform.
Prioritätswort	Prioritätswörter oder auch Schlüsselwörter genannt, geben die Wichtigkeit einer Anforderung an.
Prozesswort	Das Prozesswort spezifiziert die Art und Weise der Interaktion einer Bedingung oder Anforderung.
Subjekt	Subjekt des Satzes oder auch Bezugsobjekt genannt, ist jenes Objekt, auf welches sich die jeweilige Aufgabe, Anforderung oder Information des Satzes bezieht.
Systemkomponente	Siehe Kapitel 4.1.5
Vergleichsoperator	Ein Vergleichsoperator ist ein logischer Operator, welcher auf zwei Argumente angewendet wird und einen Wahrheitswert liefert.

Tabelle 5: Linguistische Regeln (angelehnt an [SOP 13])

4.2.2 Textklassifikation

Ziel der Textklassifikation ist die Ermittlung der Art des aktuell eingegebenen Text-Elements des durch den Anwender eingegebenen Text. Hierdurch lässt sich verifizieren, ob der Anwender den richtigen Text im richtigen Kontextmenü eingibt. Sollte der Anwender beispielsweise im Anforderungserzeugungs-Dialog fälschlicherweise eine Aufgabe definieren oder eine allgemeine Information definieren, kann der Anwender automatisch darauf hingewiesen werden durch kontextsensitive Hilfestellungen (siehe Kapitel 4.2.4).

4.2.3 Autovervollständigung

Ziel des Autovervollständigungs-Moduls ist es, dem Anwender während der Eingabe einer textuellen Beschreibung, Informationen bereitzustellen, welche zur aktuellen Eingabe passen. Dies soll die Eingabe beschleunigen und die Standardisierung der Eingaben verbessern (Anforderung 2).

Um dem Anwender passende Wortvorschläge bereitstellen zu können, werden mögliche Kontextelemente in der Datenbank gesucht und entsprechend der Eingabe Angeboten (siehe Abbildung 4.12).

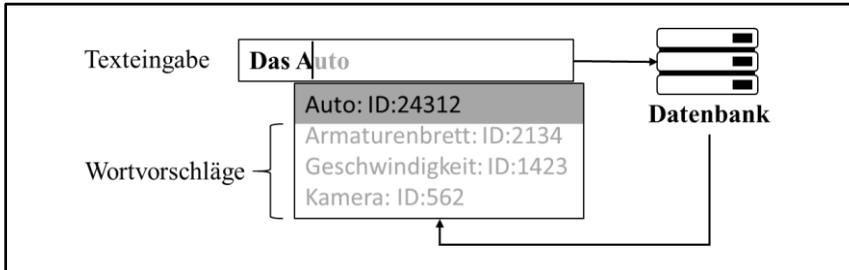


Abbildung 4.12: Beispiel der Identifikation von Wortvorschlägen

Wie in diesem Beispiel dargestellt, befinden sich zwei Systemkomponenten mit dem Anfangsbuchstaben „A“, welche am besten zur aktuellen Eingabe passen. Darüber hinaus, werden möglicherweise passende Systemkomponenten angeboten, welche in Abhängigkeit der letzten Eingaben als mögliche nächste Worte identifiziert werden.

Übernimmt der Anwender eine der Systemkomponenten, wird neben dem reinen Text auch direkt eine Objektreferenz übernommen, welche beim Speichern des Text-Elements automatisch erzeugt wird. Somit werden beispielsweise Anforderungen oder Aufgaben direkt mit den betroffenen Systemkomponenten verknüpft, wodurch die Aufwände durch manuelles Verknüpfen stark reduziert werden.

4.2.4 Kontextsensitive Hilfestellungen

Um es auch Nicht-Experten zu ermöglichen vollständige und standardisierte textuelle Beschreibungen zu formulieren (Anforderung 2), sieht der Dokumentationsassistent kontextsensitive Hilfestellungen vor. Zum einen werden jeder

vordefinierten oder kundenspezifischen Satzschablone, Referenzbeispiele zugeordnet, welche entsprechend der aktuellen Eingabe des Anwenders angepasst eingeblendet werden können. Hiermit sollen Nicht-Experten hinsichtlich der in der Literatur beschriebenen Standardformulierungen sensibilisiert werden und somit die projektweite Standardisierung von Text-Elementen wie Anforderungen oder Aufgaben zu fördern.

Zusätzlich zu den Referenzbeispielen, wird die aktuelle Eingabe hinsichtlich ihrer Vollständigkeit, mit den vordefinierten Satzschablonen abgeglichen. Werden semantische Rollen innerhalb der Eingabe vergessen oder gänzlich andere Satzstrukturen verwendet, sieht das vorliegende Konzept Leitfragen vor, welche den Anwender zur Vervollständigung seiner Formulierungen ermuntern sollen.

Da innerhalb der Unternehmen häufig unterschiedliche Begriffe bzw. Eigenheiten verwendet werden, kann dies dazu führen, dass die Erkennung der semantischen Rollen innerhalb der textuellen Beschreibung erschwert wird. Aus diesem Grund sieht das vorliegende Konzept ein mitlernendes Sprachmodell vor, welches kontextsensitiv erweitert werden kann (siehe Abbildung 4.13).

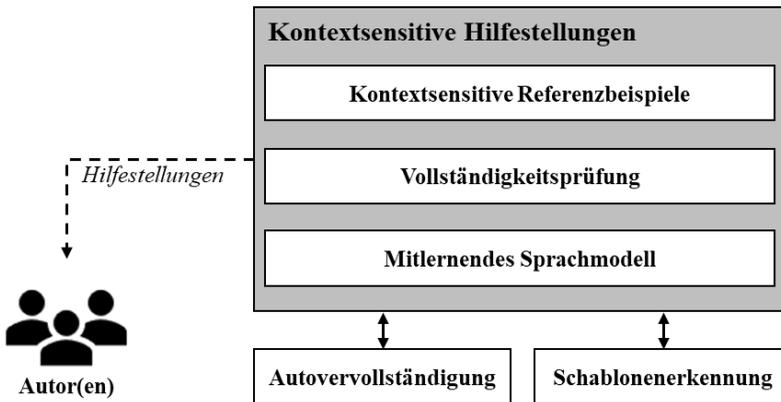


Abbildung 4.13: Kontextsensitive Hilfestellungen

Die größten Herausforderungen in der Dokumentation von Anforderungen, Aufgaben stellen die Frage nach dem richtigen Detailierungsgrad als auch die Art und Weise der Beschreibungsmethodik dar [Ada 15]. Vielen Nicht-Experten sind Satzschablonen nicht geläufig, weshalb sie diese nicht einsetzen. Um den Anwender entsprechend zu sensibilisieren, sollen daher basierend auf der aktuellen Eingabe, passende Referenzbeispiele dargestellt werden, welche es dem Anwender erleichtern sollen, die richtige Formulierungsstruktur anzuwenden.

Hierzu sieht das vorliegende Konzept die Darstellung von kontextsensitive Referenzbeispielen vor, welche entsprechend der Eingabe des Anwenders angepasst werden. Hierzu werden vordefinierte Satzschablonen verwendet, welche kontinuierlich mit der Eingabe abgeglichen werden. Die Schablone mit der höchsten Zutreffenswahrscheinlichkeit, wird dem Anwender dann dargestellt (siehe Abbildung 4.14).

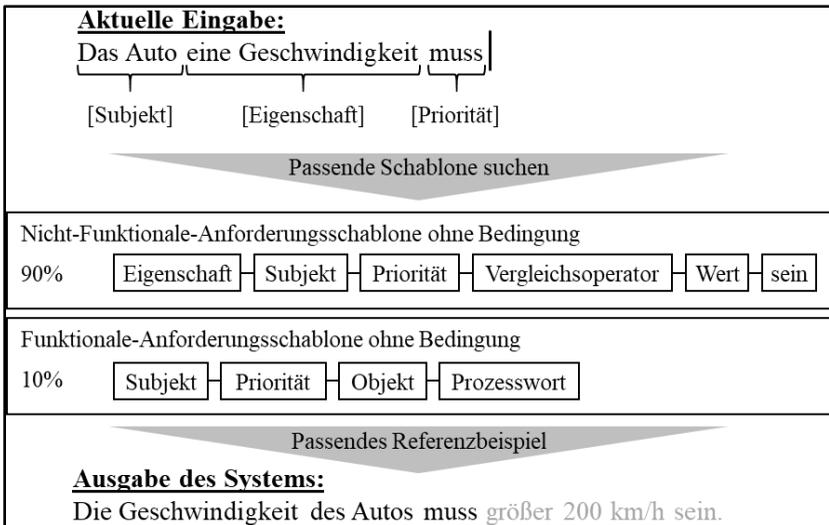


Abbildung 4.14: Bereitstellung eines Referenzbeispiels

Wie in diesem Beispiel zu sehen ist, wurde mit einer Zutreffenswahrscheinlichkeit von 90% die aktuelle Eingabe des Anwenders als „Nicht-Funktionale

Anforderung“ klassifiziert. Basierend auf den identifizierten semantischen Rollen (Subjekt, Eigenschaft und Priorität) wird dem Anwender daher das im unteren Teil der Abbildung dargestellte und an die aktuelle Eingabe angepasste Referenzbeispiel angeboten. Der Anwender hat anschließend die Möglichkeit das dargestellte Referenzbeispiel in Form eines Lückentextes zu übernehmen und die fehlenden Worte zu ergänzen, oder mit der Eingabe weiter fortzufahren.

Diese Funktion hat neben der Steigerung der Standardisierung von textuellen Eingaben einen weiteren Nutzen. Anwendern, welche keine Muttersprachler sind, kann diese Funktion helfen, grammatikalisch korrekte Sätze zu formulieren und damit die Verständlichkeit für andere Projektbeteiligte zu steigern.

Basierend auf der identifizierten Satzschablone, welche durch das Textklassifikations-Modul klassifiziert wurde (siehe Kapitel 4.2.2), kann mit Hilfe der identifizierten semantischen Rollen (siehe Kapitel 4.2.1) die aktuelle Eingabe hinsichtlich ihrer Vollständigkeit hin analysiert werden (siehe Abbildung 4.15).

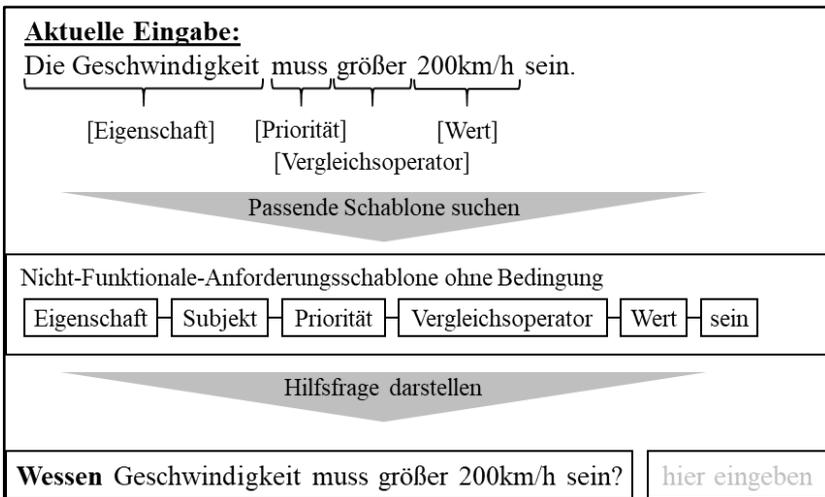


Abbildung 4.15: Bereitstellung von Hilfsfragen

Wie in diesem Beispiel zu sehen ist, kann der Dokumentationsassistent kein Subjekt innerhalb des Satzes identifizieren, weshalb dem Anwender die Leitfrage nach dem Subjekt gestellt wird. Der Anwender kann anschließend in dem dargestellten Textfeld das Subjekt eintragen (in diesem Beispiel „Auto“). Anschließend wird das Subjekt automatisch zusammen mit dem richtigen Artikel im Satz ergänzt, wodurch sich für dieses Beispiel die vollständige Anforderung „Die Geschwindigkeit des Autos muss größer 200km/h sein“ ergibt.

Mit Hilfe dieser Hilfestellung, sollen unvollständige textuelle Beschreibungen vermieden und mögliche daraus folgende Fehlinterpretationen/-entwicklungen vermieden werden. Um es der semantischen Rollenerkennung zu ermöglichen auch domänenspezifische Fachbegriffe zu erkennen, sieht das vorliegende Konzept ein mitlernendes Sprachmodell vor, welches durch kontextsensitive User-Dialoge erweitert werden kann. Hierzu werden dem Anwender die, durch die semantische Rollenerkennung erkannten, Informationen während des Speichern-Prozesses zur Prüfung bereitgestellt. Weicht eine der erkannten Kontextelementklassen ab, kann der Anwender diese direkt anpassen. Anschließend werden die falsch identifizierten Kontextinformationen dem mitlernenden Sprachmodell hinzugefügt, wodurch das System sukzessive die unternehmensspezifischen Eigenheiten mitlernt.

4.2.5 Verknüpfen und speichern

Wie bereits in Kapitel 4.2.3 motiviert wurde können Wortvorschläge, welche durch den Anwender übernommen werden direkte Objektreferenzen zu bereits dokumentierten Kontextelementen darstellen. Sobald die Eingabe einer textuellen Beschreibung eines neuen Kontextelements abgeschlossen ist, können dadurch die entsprechenden Relationen automatisch erzeugt und gespeichert werden. Dieses Verfahren vermeidet den manuellen Verknüpfungsaufwand, welcher in vielen aktuellen Anforderungsmanagement-Werkzeugen benötigt wird (siehe Kapitel 3.4). Es soll zudem das Risiko, dass wichtige Relationen zwischen abhängigen Kontextelementen nicht hergestellt werden, reduzieren und die möglicherweise damit verbundenen Fehler vermeiden (siehe Kapitel 4.5), was im folgenden Beispiel veranschaulicht wird (siehe Abbildung 4.16).

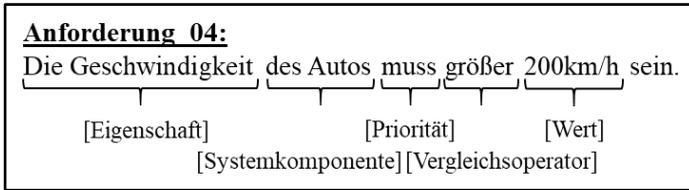


Abbildung 4.16: Informationsextraktion aus Beschreibungstexten

Wie in diesem Beispiel dargestellt, wurden durch die semantische Rollenerkennung die Systemkomponente „Auto“, sowie dessen Eigenschaft „Geschwindigkeit“ identifiziert. Zudem konnte mit Hilfe der Textklassifikation der Typ des Satzes klassifiziert werden, in diesem Fall eine „funktionale Anforderung“. Des Weiteren wurden der Wert und der Vergleichsoperator der Eigenschaft „Geschwindigkeit“ sowie die Priorität der Anforderung identifiziert. Nachfolgend werden die daraus folgenden Kontextelemente sowie deren Relationen untereinander dargestellt (Abbildung 4.17).

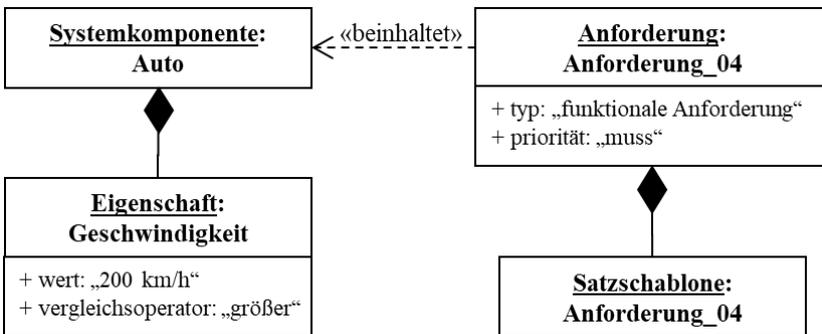


Abbildung 4.17: Kontextelemente und deren Relationen

Mit Hilfe dieser bidirektionalen Verknüpfung von Kontextelementen und deren textuellen Beschreibungen, können mögliche Fehler, welche durch eine Namensänderung verursacht werden können, vermieden werden. Zudem erspart dies dem Anwender viel manuelle Verknüpfungsbearbeitung, welche stand

heute in den meisten Anforderungsmanagement-Werkzeugen noch manuell durchgeführt werden müsse (siehe Kapitel 3.4). Dies wiederum führt dazu, dass Änderungen ganzheitlicher bewertet werden können. Denn nur wenn alle Relationen zwischen den Kontextelementen modelliert sind, kann die Tragweite einer Änderung sinnvoll analysiert und Bewertet werden (siehe Kapitel 4.6).

4.3 Integrationsassistent

Mit Hilfe der semantischen Rollenerkennung (siehe Kapitel 4.2.1) lassen sich zwar Informationen und Relationen innerhalb eines Satzes erkennen, jedoch nicht satzübergreifend. Entgegen der Empfehlungen der Literatur [SOP 13], werden Anforderungsdokumente häufig unstrukturiert dokumentiert. Vielmehr werden Anforderungen überwiegend in nicht standardisierten Anforderungsdokumenten (z.B. Lasten- oder Pflichtenheft) festgehalten, was die tägliche Arbeit mit den darin befindlichen Anforderungen für die Projektbeteiligten erschwert bzw. dazu führt, dass diese nicht aktiv in die tägliche Arbeit integriert werden, wodurch es je nach Erfahrung der Mitarbeiter zu Abweichungen kommen kann. Hinzu kommen häufig Verweise auf andere Dokumente, welche ebenfalls zu Rate gezogen werden müssen, um die vorgegebenen Spezifikationen der Kunde vollständig erfüllen zu können.

Um die innerhalb dieser Dokumente befindlichen Systeminformationen und Anforderungen in die semantische Datenstruktur dieses Konzepts überführen zu können, sieht das vorliegende Konzept daher einen Integrationsassistenten vor, welcher Abhängigkeiten zwischen einzelnen Informationen sowohl satz- als auch dokumentenübergreifend identifizieren soll.

Um dies zu ermöglichen wird in einem ersten Schritt die Struktur des zu analysierenden Textes (Dokument) analysiert. Im zweiten Schritt werden die einzelnen Sätze mit Hilfe der Textklassifikation klassifiziert (siehe auch Kapitel 4.2.2) und mit Hilfe der semantischen Rollenerkennung die darin befindlichen semantischen Rollen extrahiert (siehe auch Kapitel 4.2.1).

Anschließend werden mögliche Koreferenzen zwischen den einzelnen Sätzen innerhalb des gesamten Textes (Dokuments) identifiziert und aufgelöst. Abschließend werden die Ergebnisse dem Anwender zur Prüfung und Freigabe bereitgestellt (siehe Abbildung 4.18).

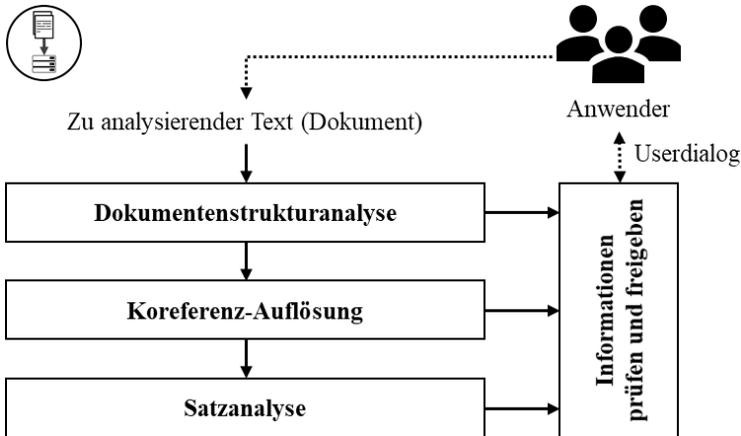


Abbildung 4.18: Funktionaler Aufbau der Relationserkennung

Nachfolgend werden die einzelnen Funktionen des Integrationsassistenten genauer erläutert.

4.3.1 Dokumentenstrukturanalyse

Da viele Anforderungsdokumente in Form von PDF übermittelt werden, welche teilweise nur Scans statt maschinenlesbarer Texte beinhalten, nutzt das vorliegende Konzept ein Texterkennungsverfahren, welches für vektorisierte Texte, als auch Bilddaten genutzt werden kann. Hierzu werden die Dokumente zunächst in Bilder konvertiert, damit diese anschließend mit Hilfe von Texterkennungsverfahren (engl. „optical character recognition“, kurz OCR) einheitlich weiterverarbeitet werden können. Damit diese Inhalte nachfolgend weiterverarbeitet werden können, werden alle erkannten Textteile, zu einem

zusammenhängenden Text zusammengefasst und der Koreferenz-Auflösung bereitgestellt.

4.3.2 Koreferenz-Auflösung

Satzübergreifende Zusammenhänge (siehe Abbildung 4.19) zwischen Objekten können mit Hilfe von Koreferenz-Auflösungs-Verfahren analysiert werden. Hierzu können maschinelle-Lernverfahren [Jos 19, Wol 17] als auch regelbasierte Ansätze zum Einsatz kommen.

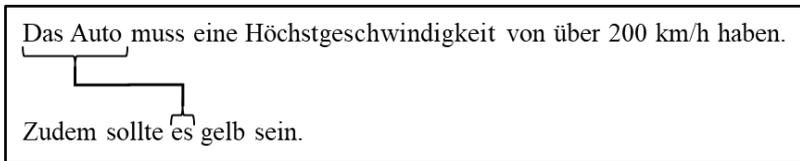


Abbildung 4.19: Beispiel einer satzübergreifenden Koreferenz

Wie in diesem Fall dargestellt, werden hier zwei Anforderungen für dasselbe Subjekt „das Auto“ spezifiziert, wobei das Subjekt in der zweiten Anforderung nicht namentlich genannt wird. Mit Hilfe der Koreferenz-Auflösung, lassen sich derartige Abhängigkeiten auflösen.

Einzelne Systemkomponenten können mehrfach innerhalb eines Dokumentes genannt werden, müssen sich jedoch nicht zwingend auf das gleiche Objekt beziehen. Beispielsweise kann das Begriff „Steuerung“ innerhalb eines Anforderungsdokuments mehrfach genannt werden, sich jedoch auf unterschiedliche Steuermodul innerhalb eines Systems beziehen (z.B. SPS und Robotersteuerung). Um diese nicht fälschlicherweise als „das gleiche Objekt“ zu bezeichnen, sieht das vorliegende Konzept vor, Systemkomponenten nur innerhalb eines Kapitels oder Absatzes automatisch mit Hilfe der Koreferenz-Auflösung zu vereinigen. Absatzübergreifende Bezüge, müssen durch den Anwender geprüft und freigegeben werden.

In den meisten Anforderungsdokumenten werden zu Beginn Glossare aufgeführt, welche spezifische Begriffe definieren und voneinander abgrenzen. Untersuchungen innerhalb des DAM4KMU Projekts haben jedoch gezeigt, dass trotz der Begriffsdefinitionen zu Beginn des Dokuments, mehrdeutige Bezeichnungen für die gleichen Elemente verwendet werden können. Aus diesem Grund sieht das vorliegende Konzept den Einsatz einer Synonymdatenbank sowie einer Ähnlichkeitsanalyse vor, um mögliche Redundanzen zu identifizieren. Werden mögliche Synonyme identifiziert (z.B. „PKW“ und „Auto“) werden der Anwender aufgefordert, diese zu prüfen und im Falle einer Übereinstimmung, diese zu vereinigen.

4.3.3 Satzanalyse

Um die innerhalb eines Satzes befindlichen Informationen extrahieren zu können, werden die Sätze mit Hilfe der semantischen Rollenerkennung analysiert und anschließend mit Hilfe der Textklassifikation klassifiziert. Hiermit lassen sich neben den Bezugsobjekten oder Querverweisen auch mögliche Eltern-Kind-Beziehungen zwischen Systemkomponenten ableiten (z.B. „Die Tür des Autos“ → Die Tür ist Kind-Element des Autos).

In Kombination mit der Koreferenz-Auflösung, ließe sich alle Informationen innerhalb des zu analysierenden Textes (Dokuments) in die semantische Datenbank überführen. Da jedoch häufig Synonyme für bestimmte Begriffe verwendet werden, müssen die erkannten Systemkomponenten mit Hilfe einer Ähnlichkeitsanalyse z.B. durch eine Synonymdatenbank, hinsichtlich ihrer Ähnlichkeit analysiert werden. Da diese jedoch nur zu einer bestimmten Wahrscheinlichkeit mögliche Abhängigkeiten identifizieren kann, müssen die automatisch identifizierten Kontextelemente durch den Anwender geprüft und freigegeben werden.

4.3.4 Informationen prüfen und freigeben

Um die innerhalb des zu analysierenden Textes befindlichen Kontextinformationen in die semantische Datenbank zu überführen, muss der Anwender die

zu integrierenden Inhalte prüfen. Hierzu sieht das vorliegende Konzept einen zweistufigen Userdialog vor.

Im ersten Schritt, muss der Anwender Kontextelemente, welche nicht eindeutig durch das System klassifiziert werden konnten, prüfen und ggf. korrigieren. Anschließend werden dem Anwender alle automatisch erkannten und manuell geprüften Kontextelemente grafisch aufbereitet dargestellt, damit dieser die Daten prüfen und anschließend freigeben oder korrigieren kann.

Dem Anwender werden hierzu der aktuell zu prüfendem Abschnitt zusammen mit den zu prüfenden Merkmalen in Form von Tickets dargestellt. Zur besseren Übersicht wird ihm zusätzlich eine Miniaturansicht des gesamten Dokuments dargestellt, mit einem Zeiger, an welcher Stelle er sich gerade befindet (siehe Abbildung 4.20).

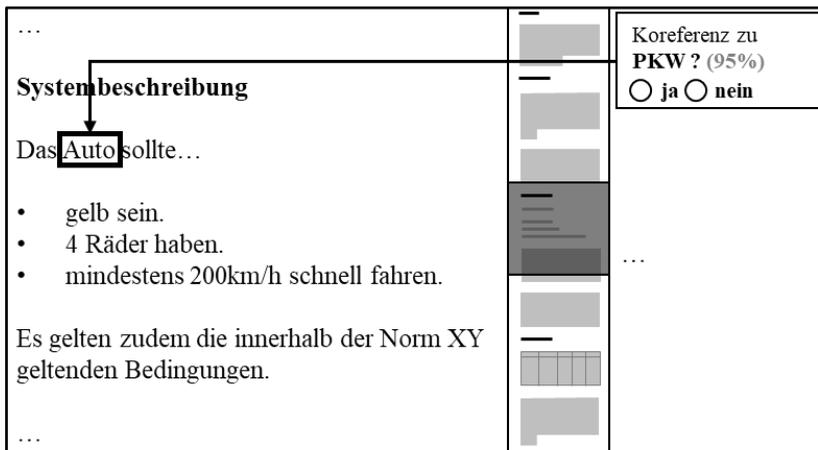


Abbildung 4.20: Mockup des Prüfdialogs

Wie hier dargestellt, ist links der aktuell zu prüfendem Abschnitt des Dokuments, rechts die zu prüfenden Merkmale in Form von Tickets und die Miniaturübersicht dazwischen. Als mögliche zu prüfenden Merkmalen können dem

Anwender folgende Ticketarten mit folgenden Aktionsmöglichkeiten dargestellt werden.

Ticketart	Aktions- bzw. Auswahlmöglichkeiten
Das System ist sich unsicher ob eine Koreferenz richtig erkannt wurde.	Sind <WORT> und <potentielle Koreferenz> gleich? <ul style="list-style-type: none"> • Ja • Nein
Das System ist sich unsicher ob es sich um eine Anforderung, Aufgabe oder Information handelt.	Bitte wählen Sie die Satzart: <ul style="list-style-type: none"> • Anforderung • Aufgabe • Weder noch (Information)
Das System ist sich unsicher ob es sich bei dem analysierten Nomen um eine Systemkomponente oder eine Eigenschaft handelt.	Bitte wählen Sie die Wortart: <ul style="list-style-type: none"> • Systemkomponente • Eigenschaft

Tabelle 6: Aktionsmöglichkeiten je zu prüfendes Merkmal

Nachdem alle uneindeutigen Kontextelemente durch den Anwender geprüft und manuell klassifiziert wurden, werden dem Anwender alle zu integrierenden Kontextelemente dargestellt, damit dieser sie final prüfen und zur Integration freigeben kann. Hierzu werden alle Kontextelemente anhand der Systemkomponenten Struktur sortiert dargestellt (siehe Abbildung 4.21).

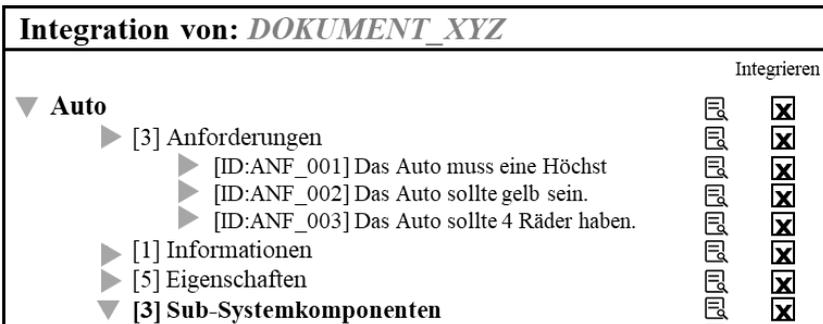


Abbildung 4.21: Mockup des Integrationsdialogs

Wie in diesem Beispiel dargestellt, wurde in dem zu integrierenden Dokument „Dokument_XYZ“ die Haupt-Systemkomponente „Auto“ identifiziert, welche mehrere Anforderungen, Informationen, Eigenschaften und Sub-Systemkomponenten hat. Der Anwender kann anschließend alle Kontextelemente sichten und auswählen welche der Kontextelemente in die bestehende Datenbank überführt werden sollen.

4.4 Rechercheassistent

Nach einem McKinsey Reports aus dem Jahre 2012 verbringen „interaction workers“ – Hochqualifizierte Experten, einschließlich Manager und Fachleute – nahezu 20% der täglichen Arbeitszeit mit der Suche nach Informationen [96]. Ziel dieser Recherche können die Identifikation von noch nicht dokumentierten Systemkomponenten, Anforderungen oder zusätzlicher Informationen sein. Um die damit verbundenen Aufwände zu reduzieren sieht das vorliegende Konzept einen Rechercheassistenten vor, welcher mögliche Kontextinformationen basierend auf den bereits dokumentierten Kontextelementen oder manuell eingegebenen Stichworten aus dem Internet extrahiert (Anforderung 9) und diese dem Anwender zur Integration bereitstellt (siehe Abbildung 4.22).

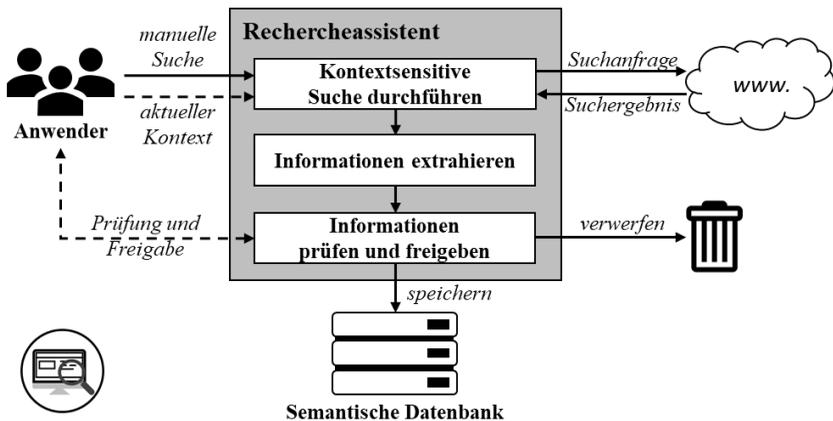


Abbildung 4.22: Funktionaler Aufbau des Rechercheassistenten

Nachfolgend werden die in Abbildung 4.22 dargestellten Funktionen des Rechercheassistenten genauer beschrieben.

4.4.1 Kontextsensitive Suche durchführen

Das vorliegende Konzept sieht zwei Arten von Recherche vor, welche den Anwender bei der Erfüllung seiner Aufgaben bestmöglich unterstützen sollen. Zum einen kann der Anwender aktiv mit Hilfe des Rechercheassistenten Suchbegriffe eingeben, welche anschließend als Grundlage für eine Internetrecherche verwendet werden (aktive Recherche). Zum anderen beobachtet der Rechercheassistent den Anwender kontinuierlich, um ihm entsprechend des aktuellen Kontextes, möglicherweise relevante Informationen automatisch aus dem Internet bereitstellen zu können (passive Recherche).

Bei der aktiven Recherche werden basierend auf durch den Anwender eingegebenen Systemkomponenten-Namen, Anforderungen, Systemkomponenten und deren Eigenschaften – welche noch nicht berücksichtigt wurden – sowie mögliche weitere Informationen in Internetquellen gesucht, um diese dem Anwender zur Integration in das aktuelle Projekt anzubieten.

Bei der passiven Recherche, werden Suchbegriffe anhand des aktuellen Kontextes automatisch abgeleitet. Hierzu werden zunächst alle durch den Anwender aktuell sichtbaren Kontextelemente identifiziert. Anschließend werden die Namen der sichtbaren Systemkomponenten sowie die innerhalb textueller Beschreibungen befindlicher Präzisierungen verwendet, um Suchbegriffe abzuleiten.

4.4.2 Informationen extrahieren

Um die Ergebnisse der Google-Suche weiterverarbeiten zu können, müssen zunächst die gefundenen Webseiten analysiert und aufbereitet werden. Hierzu

werden mit Hilfe von Scraping-Verfahren², die Inhalte der gefundenen Webseiten extrahiert. Im Vergleich zum Integrationsassistenten, wird an dieser Stelle von maschinenlesbaren Texten innerhalb der Webseiten ausgegangen. Informationen innerhalb von Bildern oder Grafiken werden für das vorliegende Konzept vernachlässigt.

Wie erste Tests eigen, werden die meisten Webseiten mit Hilfe der Header-Syntax mit Überschriften versehen, denen anschließend Text folgt, welcher sich in unterschiedlichen HTML-Syntax-Strukturen (z.B. <section>, <a> etc.) befinden kann. Folglich ergibt sich anhand der verschiedenen Header und Text-Abschnitte eine Dokumentenstruktur, welche analog zum Verfahren des Integrationsassistenten auf mögliche Koreferenzen (siehe Kapitel 4.3.2) und innerhalb der einzelnen Sätze befindlichen Informationen (siehe Kapitel 4.3.3) analysiert werden kann.

4.4.3 Informationen prüfen und freigeben

Nachdem alle Informationen aus den Webquellen extrahiert und aufbereitet wurden, werden diese dem Anwender mit Hilfe des Integrationsdialogs zur Prüfung und Freigabe dargestellt (siehe auch Kapitel 4.3.4).

4.5 Verknüpfungsassistent

Um alle miteinander in Verbindung stehenden Kontextelemente innerhalb eines Projektes als auch projektübergreifend miteinander zu verknüpfen, müssen stand heute die Anwender von Informationsmanagementsystemen die Verknüpfungen manuell herstellen (siehe auch Kapitel 3.4). Da jedoch, vor allem in größeren und komplexeren Projekten, nicht jeder Mitarbeiter alle Informationen überblicken kann, ist eine rein manuelle Verknüpfung aller miteinander in Verbindung stehender Kontextelemente nicht realistisch. Hinzu kommen

² Ein Datenextraktionsverfahren von Webquellen, https://en.wikipedia.org/wiki/Web_scraping, zuletzt geprüft am 20.08.2020

die Herausforderungen mögliche Redundanzen oder gar konfligierenden Informationen zu vermeiden. Ziel des Verknüpfungsassistenten ist es daher, die Verknüpfung zwischen neuen und bereits dokumentierten Kontextelementen kontextsensitiv zu unterstützen (Anforderung 11) und der Bildung von Redundanzen oder Konflikten vorzubeugen (Anforderung 9).

Um dies zu gewährleisten werden anhand des aktuellen Kontextes, bereits dokumentierte Kontextelemente analysiert, hinsichtlich möglicher Redundanzen oder konfligierender Informationen sowie potentieller Relationspartner. Anschließend werden dem Anwender kontextsensitive Handlungsmöglichkeiten angeboten, mit deren Hilfe der Anwender die identifizierten Informationen weiterverarbeiten kann (siehe Abbildung 4.23).

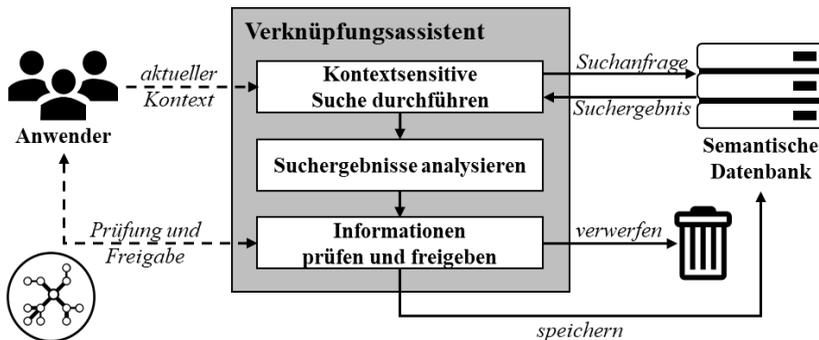


Abbildung 4.23: Aufbau des Verknüpfungsassistenten

Nachfolgend werden die funktionalen Bestandteile des Verknüpfungsassistenten detailliert beschrieben.

4.5.1 Kontextsensitive Suche durchführen

Analog zur passiven Recherche des Rechercheassistenten wird der Kontext des Anwenders kontinuierlich analysiert, um basierend auf den darin enthaltenen Kontextelementen (siehe Kapitel 4.4.1), mögliche Redundanzen oder konfligierender Informationen sowie potentielle Relationspartner finden zu können.

Im Vergleich zum Rechercheassistenten wird jedoch nicht das Internet als Datenquelle für genutzt, sondern die bereits dokumentierten Kontextelemente in der semantischen Datenbank. Um auch sinnngemäße Ähnlichkeiten identifizieren zu können, werden zu jedem Suchbegriff Synonyme abgeleitet, welche ebenfalls als Suchbegriffe verwendet werden.

4.5.2 Suchergebnisse analysieren

Bei der Analyse der Suchergebnisse, werden die bei der Suche identifizierten Kontextelemente mit den innerhalb des aktuellen Kontextes befindlichen Kontextelementen verglichen, um mögliche Redundanzen oder konfligierende Informationen innerhalb des entsprechenden Projekts sowie potentielle Relationspartner auch projektübergreifend zu identifizieren. Redundanzen liegen dann vor, wenn Kontextelemente einer bestimmten Klasse (z.B. Anforderungen oder Systemkomponenten) in gleicher oder ähnlicher Form (z.B. als Synonym) innerhalb des gleichen Projekts vorliegen. Hierbei können Namen und Beschreibungstexte identisch oder sinngemäß ähnlich sein.

Handelt es sich um Kontextelemente der gleichen Klasse (z.B. eine Anforderung oder Systemkomponente), werden in einem ersten Schritt die Kontextelementnamen mit Hilfe einer Synonymdatenbank miteinander verglichen. Sind beide Namen identisch oder entspricht der eine Name einem Synonym des anderen, wird der Anwender über die mögliche Redundanz informiert. Sollte sich hierbei jedoch herausstellen, dass die vermeintliche Redundanz keine ist, kann diese mit Hilfe der Nicht-Redundanz-Relation als solche gekennzeichnet werden (siehe Abbildung 4.24).

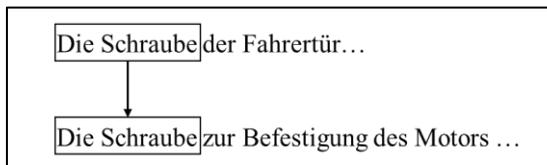


Abbildung 4.24: Beispiel für eine Nicht-Redundanz

Wie in diesem Beispiel dargestellt, können mehrere Kontextelemente gleichen Namens innerhalb eines Projektes vorkommen, ohne dass es sich hierbei um das gleiche Element handelt. Durch die Verbindung beider Elemente mit Hilfe der Nicht-Redundanz-Relation kann das erneute Finden der Nicht-Redundanz vermieden werden.

Sind beiden Kontextelementnamen nicht gleich oder ähnlich, werden die zu vergleichenden Kontextelementbeschreibungen auf ihre Ähnlichkeit hin untersucht [Ger 19, Rei 19]. Wird eine Ähnlichkeit zwischen beiden Beschreibungstexten identifiziert, werden in einem 2. Schritt mit Hilfe der semantischen Rollenerkennung, die innerhalb der Kontextelementbeschreibung befindlichen semantischen Rollen extrahiert und analog zum Abgleich der Kontextelementnamen, mit Hilfe einer Synonymdatenbank miteinander verglichen. Für jedes gleiche oder ähnliche Wortpaar wird der zuvor errechnete Ähnlichkeitswert erhöht und für jedes ungleiche Paar reduziert.

Alle identifizierten Kontextelemente mit einer sehr hohen Ähnlichkeit, werden anschließend dem Anwender als mögliche Redundanzen zur Prüfung im Kontextmenü dargestellt. Sollte der Anwender eine Redundanz bestätigen, werden die entsprechenden Elemente und deren Relationen zu einem neuen Kontextelement zusammengefasst (siehe Kapitel 4.1.7). Entsprechen die gefundenen Elemente keiner Redundanz, werden diese mit Hilfe der Nicht-Redundanz-Relation gekennzeichnet, wodurch das erneute Finden vermieden wird.

Da das zuvor genannte Verfahren lediglich Wertebereiche abgleichen und somit mögliche Konflikte identifizieren kann, jedoch keine semantisch konfliktierenden Anforderungen voneinander unterscheiden kann, sieht das vorliegende Konzept zudem den Einsatz von NLP-basierten Interferenzerkennungen vor. Hierzu werden zunächst Sätze welche sehr ähnlich sind – analog zum Vorgehen der Redundanzerkennung – identifiziert. Anschließend werden die gefundenen Sätze, mit dem zu vergleichenden Satz hinsichtlich der möglichen Abhängigkeiten „neutral“, „bedingt“ und „widersprüchlich“ analysiert. Sofern zwei Sätze sich gegenseitig widersprechen, kann mit der entsprechenden Wahrscheinlichkeit davon ausgegangen werden, dass die beiden Sätze miteinander konfliktieren (siehe Abbildung 4.25).

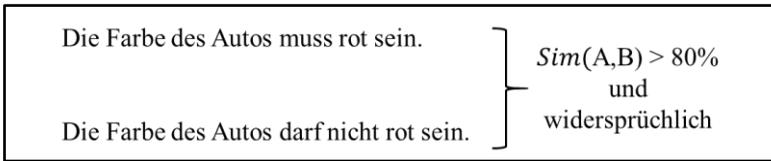


Abbildung 4.25: Semantisch konfligierende Anforderungen

Dies ermöglicht es auch rein textbasierte Anforderungen hinsichtlich möglicher Konflikte zu analysieren.

Mögliche Schwächen des Konzepts

Das vorliegende Verfahren ist nicht im Stande übergeordneten Konflikte z.B. mit Spezifikationen aus anderen Anforderungsdokumenten (z.B. Richtlinien, Normen oder Gesetze) zu identifizieren. Die Anbindung der dafür benötigten Datenbanken ist nicht Teil des vorliegenden Konzepts. Es sei zudem auf die Lösungen der Firma REUSE Company verwiesen, welche mit Hilfe von domänenspezifischen Ontologien, auch komplexere semantische Konflikte erkennen können [Fra 15]. Da die vorliegende Arbeit auf ein Konzept abzielt, welches keiner unternehmensspezifischen Anpassung bedarf, wird diese technische Lösung für die vorliegende Arbeit nicht in Betracht gezogen.

Basierend auf dem zuvor beschriebenen Verfahren der Erkennung möglicher Redundanzen, lassen sich auch mögliche Relationspartner innerhalb desselben Projekts als auch projektübergreifend identifizieren. Kontextelemente welche die gleichen oder sinngemäß gleichen Subjekte und/oder Objekte beinhalten, stellen potentielle Relationspartner dar. Beispielsweise können unterschiedliche Textelemente auf die gleiche Systemkomponente referenzieren, wodurch diese, mögliche relevante, Informationen für den Komponentenverantwortlichen beinhalten können.

Werden mögliche Ähnlichkeiten identifiziert, werden die beiden potentiellen Relationspartner hinsichtlich bereits bestehender Relationen untersucht. Sind keine gemeinsamen Relationen vorhanden, werden beide Relationspartner

dem Anwender zur Verknüpfung im Kontextmenü angeboten, welche anschließend die Paarung prüfen und ggf. freigeben kann.

4.5.3 Informationen prüfen und freigeben

Nachdem wie zuvor beschrieben die Suchergebnisse analysiert wurden und dem Anwender mit Hilfe des Kontextmenüs zur Verfügung gestellt wurden, hat dieser anschließend die Möglichkeit, die gefundenen redundanten, oder konfligierenden Informationen sowie potentielle Relationspartner zu prüfen und eventuell notwendige Maßnahmen zu ergreifen.

4.6 Wiederverwendungsassistent

Ziel des Wiederverwendungsassistenten ist es bereits dokumentierten Informationen kontextsensitiv bereitzustellen, damit die mit der erneuten Dokumentation verbundene Doppelarbeit vermieden werden kann und das damit verbundene Erfahrungswissen nachhaltig konserviert werden kann. Um dies zu gewährleisten, werden aufbauend auf den Ergebnissen des Verknüpfungsassistenten alle mit den möglichen Treffern verbundenen Kontextinformationen analysiert und kontextsensitiv zur Prüfung und Freigabe bereitgestellt (siehe Abbildung 4.26).

Nachfolgend werden die funktionalen Bestandteile des Wiederverwendungsassistenten detailliert beschrieben.

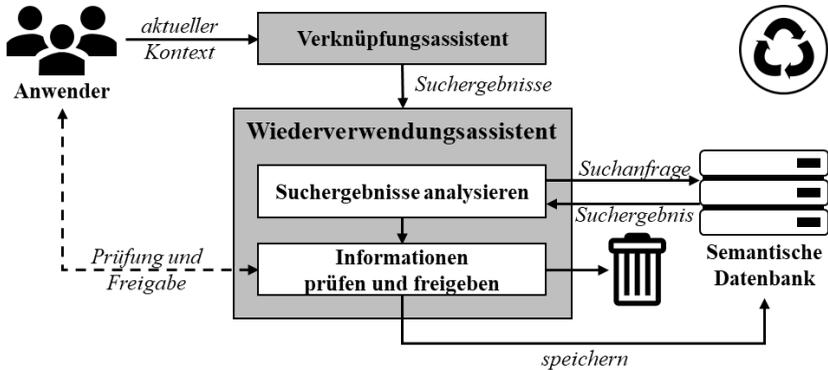


Abbildung 4.26: Aufbau des Wiederverwendungsassistenten

4.6.1 Suchergebnisse analysieren

Sobald Informationen mit dem aktuellen Kontext mit Hilfe des Verknüpfungsassistenten verknüpft werden (siehe Kapitel 4.5), welche sich außerhalb des aktuellen Projekts befinden, sind möglicherweise auch damit in Verbindung stehenden Informationen für den Anwender relevant. Werden beispielsweise gleiche oder ähnliche Systemkomponenten innerhalb eines vorangegangenen Projekts identifiziert, werden dem Anwender alle mit den gefundenen Kontextelementen in Verbindung stehenden Kontextelemente (z.B. Aufgaben, Informationen oder Anforderungen) zur Integration in das aktuelle Projekt angeboten. Da vor allem Systemkomponenten hierarchisch strukturiert werden, werden zunächst alle Kinder und Kindeskiner der betroffenen Komponente identifiziert. Anschließend werden die mit diesen in Verbindung stehenden Kontextelemente identifiziert und anschließend dem Anwender zur Prüfung und Freigabe bereitgestellt.

4.6.2 Informationen prüfen und freigeben

Um die identifizierten Informationen möglichst übersichtlich prüfen und freigeben zu können, werden die Informationen anhand der identifizierten Hierarchie der betroffenen Systemkomponenten dargestellt (siehe Abbildung 4.27).

Möchten Sie weitere Informationen von Auto aus dem Projekt ProjektXYZ übernehmen?		Referenzieren	Kopieren	Nicht relevant
▼ Auto		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
▶ [3] Anforderungen		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶▶ [ID:ANF_312] Das Auto muss eine Höchstgesc...		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶▶ [ID:ANF_12] Das Auto sollte gelb sein.		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶▶ [ID:ANF_10] Das Auto sollte 4 Räder haben.		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶ [1] Informationen		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▼ [3] Sub-Systemkomponenten		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶▶ Motor		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶▶ Getriebe		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▶▶ Chassis		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Abbildung 4.27: Mockup des Wiederverwendungsassistenten

Wie in diesem Beispiel dargestellt, wurde durch den Anwender die Systemkomponente „Auto“ aus dem Projekt „ProjektXYZ“ als relevante Komponente identifiziert, welche er nun in das bestehende Projekt übernehmen möchte. Da sich im Rahmen des neuen Projekts einige Merkmale der Komponente ändern sollen, wird diese als „Kopie“ im neuen Projekt neu instanziiert. Daraufhin wird der Anwender gebeten, die damit verbundenen Kontextelemente zu prüfen und zu markieren, welche Kontextelemente 1:1 (also als Referenz), in angepasster Form (Kopie) oder gar nicht übernommen werden sollen.

Im Anschluss darauf, werden Elemente, welche referenziert werden sollen, mit der Kopie der Komponente „Auto“ automatisch verknüpft. Kontextelemente welche ebenfalls in angepasster Form neu instanziiert werden sollen, werden durch die Ableitungs-Relation mit dem ursprünglichen Kontextelemente verknüpft, um dessen Herkunft festzuhalten. Kontextelemente welche als „nicht relevant“ markiert wurden, werden durch eine Irrelevant-Relationen mit dem aktuellen Projekt verknüpft, wodurch diese in einem späteren Suchvorgang nichtmehr dem Anwender zur Integration in das aktuelle Projekt angeboten werden. Da sich jedoch der Fokus eines Projekts über die Zeit hinweg wandeln

kann, sieht das vorliegende Konzept vor, Irrelevant-Relationen nach einer bestimmten Zeit T als veraltet zu markieren. So können diese Elemente dem Anwender zu einem späteren Zeitpunkt mit geringerer Relevanz erneut angeboten werden, wodurch im Falle einer Änderung der Randbedingungen eines Projektes möglicherweise relevante Informationen nicht verloren gehen. Sollten sie jedoch weiterhin nicht relevant sein, werden die Irrelevant-Relationen wieder aktiv und der Vorgang beginnt von neuem.

4.7 Änderungsassistent

Projekthinhalte ändern sich entlang des Projektverlaufs kontinuierlich. Es kommen neue Kontextelemente hinzu, bestehende werden geändert oder gelöscht. Je nach Art der Änderung, können diese große Auswirkungen für den Projektverlauf haben, welche sich jedoch meist nur schwer analysieren lassen. Vor allem Office-Dokumente erschweren die Analyse, da diese zum einen nicht standardisiert strukturiert und zum anderen keine bidirektionale Verknüpfung zwischen Informationen untereinander ermöglichen (siehe auch Kapitel 3.4).

Um den Anwender zu befähigen die mit einer Änderung verbundenen Auswirkungen bestmöglich bewerten zu können, sieht das vorliegende Konzept vor, dem Anwender die Abhängigkeiten der betroffenen Kontextelemente zu visualisieren und den Einfluss der betroffenen Kontextelemente automatisch zu schätzen. Hierzu wird in einem ersten Schritt die Art der Änderung kategorisiert. Im zweiten Schritt kann dann anhand der Art der Änderung und des Einflusses der betroffenen Kontextelemente, die Tragweite der Änderung abgeschätzt werden. Anschließend werden die Verantwortlichen der betroffenen Kontextelemente informiert, welche dann, unterstützt durch eine grafische Visualisierung, die Ergebnisse prüfen und freigeben können (siehe Abbildung 4.28). Dies soll es dem Anwender erleichtern, die mögliche Tragweite einer Änderung möglichst schnell und intuitiv bewerten zu können und somit das Risiko von Änderungen zu reduzieren (Anforderung 6).

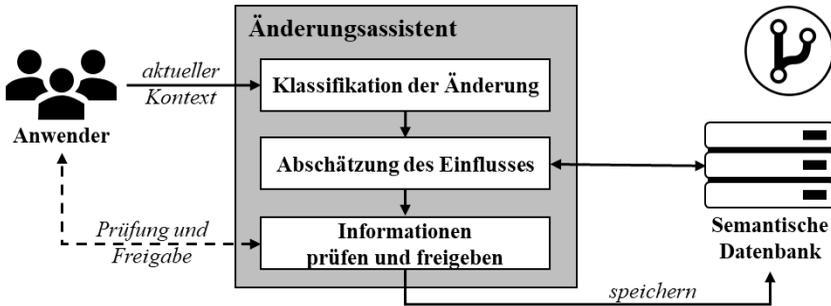


Abbildung 4.28: Funktionaler Aufbau des Änderungsassistenten

Nachfolgend werden die funktionalen Bestandteile des Änderungsassistenten detailliert beschrieben.

4.7.1 Klassifikation der Änderung

Handelt es sich um Änderungen, welche keine Änderung der Systemstruktur oder der Anforderungen zur Folge hat, kann diese als „unkritisch“ bewertet werden, da diese keine große Auswirkung auf das Gesamtprojekt haben. Änderungen welche jedoch die Komplexität oder den Gesamtaufwand beeinflussen, müssen „kritisch“ bewertet und geprüft werden, vor allem dann, wenn das Projekt bereits weit vorangeschritten ist und hohe Folgekosten entstehen können [Ehr 09, S. 170].

Für das vorliegende Konzept lassen sich hierbei folgende Arten von Änderungen unterscheiden. Zum einen können neue Kontextelemente hinzugefügt, bestehende namentlich oder hinsichtlich ihrer textuellen Beschreibung verändert sowie komplett gelöscht werden. Je nach Kontextelementart sind die unterschiedlichen Änderungsarten unterschiedlich zu bewerten, was nachfolgend je Änderungsart genauer erläutert wird.

Das Hinzufügen neuer Kontextelemente in ein bestehendes Projekt kann die Komplexität als auch den Gesamtaufwand des Projektes negativ beeinflussen.

Je fortgeschrittener das Projekt ist, je kritischer sind beispielsweise neue Anforderungen [Ehr 09, S. 170].

Kontextelemente allgemein

Neue Kontextelemente abgesehen von Informationen, können einen negativen Einfluss auf das Budget oder den Zieltermin des betreffenden Projektes haben. Aus diesem Grund ist das Hinzufügen von Kontextelementen bei weit fortgeschrittenen Projekten, kritischer zu bewerten als zu Beginn eines Projektes. Da jedoch auch zu Beginn eines Projektes, das Budget und der Zieltermin nie außer Acht gelassen werden sollten, sieht das vorliegende Konzept für jedes neu hinzugefügte Kontextelement einen Abgleich des geplanten Gesamtbudgets und dem geplanten Zieltermin vor.

Um dies zu ermöglichen, werden alle Aufwände und Kosten aller Kontextelemente des Projekts akkumuliert und mit dem geplanten Gesamtbudget und dem Zieltermin verglichen. Das Ergebnis wird dem Anwender anschließend bereitgestellt, damit dieser bereits während der Erzeugung neuer Kontextelemente, eine mögliche Überschreitung vermeiden kann.

Projekt

Wird ein neues Teilprojekt einem bestehenden Projekt hinzugefügt, wird geprüft, ob Start- und Zieltermin mit dem übergeordneten Projekt realisierbar ist. Zudem wird das geplante Budget des neuen Projekts mit den Restfinanzen des übergeordneten Projekts abgeglichen. Sollte es hierbei Überschreitungen geben, wird der Anwender darauf hingewiesen. Weitere Analysen sind erst nach dem Hinzufügen weitere Kontextelemente möglich. Die damit verbundenen Auswirkungen werden daher als „unkritisch“ bewertet.

Systemkomponente

Neue Systemkomponenten steigern die Komplexität eines Projekts und haben meist zusätzliche Anforderungen und Aufgaben zur Folge, welche wiederum die Kosten steigern. Zum Zeitpunkt der Instanziierung jedoch, können lediglich die geplanten Kosten der Komponente hinsichtlich des Gesamtbudgets des

Projekts bewertet werden. Sollte es hierbei Überschreitungen geben, wird der Anwender darauf hingewiesen. Die damit verbundenen Auswirkungen werden jedoch als „unkritisch“ bewertet, da ähnlich wie bei Projekten noch keine weiteren Auswirkungen auf das übergeordnete Projekt abgeleitet werden können.

Anforderung

Handelt es sich um Anforderungen von Systemkomponenten, welche erst kürzlich erzeugt wurden oder nur geringe Kosten oder einen geringen Einfluss auf das Gesamtsystem haben, ist das Hinzufügen einer neuen Anforderung als „unkritisch“ zu bewerten.

Werden jedoch neue Anforderungen für bereits länger existierende Systemkomponenten hinzugefügt, kann dies zu „kritischen“ Konflikten mit den eventuell bereits laufenden Arbeiten bezüglich dieser Komponente führen. Aus diesem Grund werden alle von der Anforderung betroffenen Systemkomponenten hinsichtlich ihres Status geprüft und dem Anwender grafisch aufbereitet bereitgestellt. Dieser kann anschließend die betroffenen Systemkomponenten sowie deren Anforderungen prüfen und falls notwendig Gegenmaßnahmen einleiten.

Aufgabe

Neue Aufgaben können zu jedem Zeitpunkt des Projekts entstehen und sind hinsichtlich des Gesamtsystems weniger kritisch, da diese keine Systemkomponenten oder andere Aufgaben beeinflussen im Gegensatz zu Anforderungen, welche ggf. auch Auswirkungen auf andere Aufgaben haben können. Kritisch werden neue Aufgaben dann, wenn der geschätzte Aufwand das geplante Gesamtbudget überschreitet. Aufgrund des geringen Einflusses auf andere Kontextelemente kann eine neue Aufgabe jedoch als „unkritisch“ bewertet werden.

Information

Im Vorliegenden Konzept stellen Informationen keine kritischen Kontextelemente dar, welche daher keinen direkten Einfluss auf das Gesamtprojekt haben. Namensänderungen von Kontextelementen innerhalb eines Projektes sind

in der Regel unkritisch, sofern der neue Name keinem Kontextelement der gleichen Klasse entspricht. In diesem Fall muss durch den Anwender geprüft werden ob es sich hierbei um eine eventuelle Redundanz handelt, oder eine Nicht-Redundanz, welche analog zu dem Verfahren des Verknüpfungsassistenten verarbeitet werden.

Werden keine Redundanzen identifiziert, wird der Anwender abschließend gefragt, ob der alte Name als Synonym gespeichert werden soll, was der Anwender annehmen oder ablehnen kann. Im Falle von Text-Elementen kann sich zusätzlich zu einer Namensänderung auch die textuelle Beschreibung verändern. Hierbei können sich die unterschiedlichen semantischen Rollen innerhalb der textuellen Beschreibung verändern. Besonders von Bedeutung sind hierbei jedoch vor allem die Priorität, welche eine Änderung des Einflusses auf das Gesamtsystem zur Folge hat und das Bezugsobjekt, was ebenfalls Auswirkungen auf das Gesamtsystem haben kann. Alle weiteren Änderungen können mit Hilfe des vorliegenden Konzepts nur bedingt automatisch ausgewertet werden (siehe unter anderem Kapitel 4.5).

Prioritätsänderung

Eine Abwertung der Priorität (z.B. von „muss“ zu „sollte“) reduziert den Einfluss der jeweiligen Anforderung oder Aufgabe hingegen eine Aufwertung den Einfluss einer Anforderung stärkt.

Bezugsobjektänderung

Ändert sich das Bezugsobjekt z.B. wegen einer Fehldeklaration der Zugehörigkeit einer Anforderung oder der Zuständigkeit einer Aufgabe, hat dies eine positive Komplexitätsänderung für das alte Bezugsobjekt bzw. eine Reduktion des Aufwands für den vorherigen Verantwortlichen zur Folge hingegen es für das neue eine Steigerung der Komplexität bzw. höheren Aufwand zur Folge hat. Um den sich ändernden Einfluss zu visualisieren, werden neben der textuellen Änderung auch die veränderten Relationen und betroffenen Kontextelemente sowie deren Einfluss auf das Gesamtprojekt dargestellt (siehe Kapitel 4.7.3).

Das Löschen von Kontextelementen reduziert in erster Linie die Komplexität des Gesamtsystems. Die von einem Anwender gelöschten Elemente könnten jedoch für andere Projektbeteiligte von Nutzen gewesen sein. Aus diesem Grund sieht das vorliegende Konzept vor, Inhalte zunächst zu archivieren (veraltet = wahr). So werden die Inhalte nichtmehr in der Gesamtberechnung berücksichtigt, sind im Zweifel jedoch wiederherstellbar. Sollen Inhalte endgültig gelöscht werden, muss dies durch den Projektmanager freigegeben werden. Hierdurch soll ein fahrlässiges oder unachtsames Löschen vermieden werden.

Hängen von dem zu löschenden Element hierarchisch zugeordnete Kind-Elemente ab, müssen diese ebenfalls berücksichtigt werden. Soll beispielsweise eine komplette Baugruppe gelöscht werden, macht es meist wenig Sinn lediglich das Oberelement zu löschen und die Kind-Elemente beizubehalten. Sollte dies jedoch trotzdem der Fall sein, müssen auch alle damit verbundenen Textuellen Beschreibungen angepasst werden (siehe Abbildung 4.29).

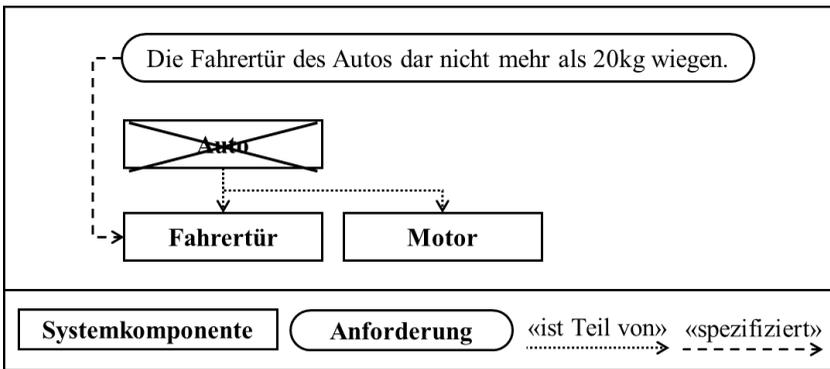


Abbildung 4.29: Beispiel einer Änderung

Wie in diesem Beispiel dargestellt, soll die Komponente „Auto“ gelöscht werden, welche die Elternkomponente der Kontextelemente „Fahrtür“ und „Motor“ darstellt. Sofern die Kind-Elemente beibehalten werden sollen, müssen die entsprechenden textuellen Beschreibungen angepasst werden, damit die Ge-

samtstruktur des Projektes konsistent bleibt. In diesem Fall würde die angepasste Anforderung wie folgt lauten: „Die Fahrertür darf nicht mehr als 20kg wiegen“.

Damit der Anwender von Fall zu Fall unterscheiden kann, sieht das vorliegende Konzept zudem vor, dem Anwender alle betroffenen Elemente darzustellen, damit er diese jeweils einzeln sichten und ggf. beibehalten kann. Sollten jedoch alle vom zu löschenden Konzeptelement abhängigen Elemente gelöscht werden, werden auch diese zunächst archiviert und auf Wunsch nach der Freigabe durch den Projektleiter endgültig gelöscht. Wie zuvor beschrieben können Änderungen positive als auch negative Einflüsse auf ein Projekt haben. Änderungen welche keinen unmittelbaren Einfluss auf andere Systemkomponenten haben, können hinsichtlich ihrer Tragweite als unkritisch bewertet werden. Hierbei müssen lediglich deren Einfluss auf das Gesamtbudget und die Projektlaufzeit berücksichtigt werden.

Änderungen welche einen unmittelbaren Einfluss auf andere Systemkomponenten haben, sollten hingegen hinsichtlich der ihrer Auswirkungen untersucht werden. Werden derartige Änderungen zu Ende eines Projektes durchgeführt, ist deren Tragweite schwerer zu bewerten als wenn die Änderungen zu Beginn eines Projektes vorgenommen werden. Beispielsweise lassen sich die Änderung von Anforderungen zu Beginn eines Projektes einfacher umsetzen, als kurz vor Projektende. Nachfolgend wird je Änderungsart die Kritikalität hinsichtlich des Einflusses auf Projektbudget und Projektlaufzeit definiert. Änderungen, welche potentiell negativen Einfluss auf die Komplexität, das Restbudget oder die Restlaufzeit des Projektes haben, werden als kritisch definiert. Änderungen, welche die Komplexität, das Restbudget oder die Restlaufzeit positiv beeinflussen, werden als unkritisch definiert.

Änderungsart	Kritikalität
Hinzufügen von Projekten, Systemkomponenten oder Aufgaben	kritisch, Budget- und Zeitprüfung notwendig. Tragweitenanalyse ohne weitere Informationen jedoch nicht möglich.
Hinzufügen von Anforderungen	kritisch, Tragweitenanalyse erforderlich
Hinzufügen von Informationen	unkritisch

Namensänderungen	Redundanzprüfung notwendig, ansonsten unkritisch
Änderung der textuellen Beschreibung allgemein	Klassifikation der Änderung anhand der geänderten semantischen Rollen mit Hilfe der semantischen Rollenerkennung
Änderung der textuellen Beschreibung Prioritätsreduktion	unkritisch, da Komplexitätsreduktion
Änderung der textuellen Beschreibung Prioritätssteigerung	kritisch, Tragweitenanalyse erforderlich da potentielle Komplexitätssteigerung
Löschen von Kontextelementen	Betroffene Kindelemente müssen geprüft werden, ansonsten positiver Einfluss auf Komplexität des Projekts daher unkritisch

Tabelle 7: Definition der Kritikalität je Änderungsart

4.7.2 Abschätzung des Einflusses

Um die Tragweite einer Änderung bestmöglich abschätzen zu können, werden in einem ersten Schritt alle betroffenen Kontextelemente identifiziert und hinsichtlich ihres Einflusses bewertet. Je nachdem ob es sich bei dem zu ändernden Kontextelement, um eine Systemkomponente oder ein Textelement handelt wird der jeweilige Einfluss unterschiedlich berechnet.

Der Einfluss einer Systemkomponente ist abhängig von dessen Position innerhalb der hierarchischen Systemstruktur. Handelt es sich um eine besonders große Baugruppe oder ein besonders wichtiges Bauteil, ist dieses in seinem Einfluss höher zu bewerten als eine Komponente, welche einen kleineren Einfluss auf das Gesamtsystem hat. Um dies automatisch ermitteln zu können, nutzt der Änderungsassistent die hierarchischen Relationen (siehe Kapitel 4.1.5), welche unter anderem automatisch mit Hilfe des Dokumentationsassistenten identifiziert und instanziiert werden (siehe Kapitel 0). Wird die rein quantitative Zahl beinhaltenen Systemkomponenten berücksichtigt je Baugruppe, ergibt sich folgender Zusammenhang für den Einfluss der zu analysierenden Systemkomponente K_S :

Formel 1:

$$E_1(K_S) = \frac{n}{m}$$

$E_1(K_S)$	Einfluss der zu ändernden Systemkomponente K_S
n	Anzahl aller Kind und Kindeskind von K_S
m	Anzahl aller Kontextelemente

Einzelne Komponenten können unterschiedliche Kosten, Prioritäten oder Status aufweisen, welche die Relevanz einzelner Komponente erheblich beeinflussen können. Aber auch diese Betrachtungsweise kann zu Fehlern führen, da z.B. zur Entwicklung oder Realisierung einer Komponente entsprechende Aufgaben verbunden sein können, welche wiederum sehr viel höhere Kosten erzeugen können als die reinen Anschaffungs- bzw. Materialkosten. Aus diesem Grund werden im vorliegenden Konzept neben den Materialkosten von Systemkomponenten auch die Summe aller damit in Verbindung stehenden Aufgabenaufwände akkumuliert und relativ zum Bezugsraum oder dem Gesamtprojekt in Relation gestellt.

Formel 2:

$$E_3(K_S) = \frac{\sum_{i=1}^n K_{Komp,i} \cdot X + \sum_{k=1}^o K_{Aufg,k} \cdot X}{\sum_{j=1}^m K_{Komp,j} \cdot X + \sum_{l=1}^p K_{Aufg,l} \cdot X}$$

$E_3(K_S)$	Einfluss der zu ändernden Systemkomponente K_S
$K_i \cdot X$	Produkt aus Priorität, Status und Kosten von K_i
$K_{Komp,i}$	i-te Systemkomponente
$K_{Aufg,i}$	i-te Aufgabe
n	Anzahl aller Kinder und Kindeskind von K_S
m	Anzahl aller Kontextelemente
o	Anzahl aller mit K_S in Verbindung stehenden Aufgaben
p	Anzahl aller Aufgaben des Projekts

Textelemente beziehen sich, wenn diese vollständig definiert sind, auf ein Bezugsobjekt (kurz BZO) bzw. eine Systemkomponente. Aus diesem Grund wird

für die Bewertung des Einflusses eines Textelements die direkte Umgebung des Bezugsobjekts verwendet. Spezifizieren beispielsweise zwei Anforderungen eine Komponente mit gleicher Priorität und gleichem Status, gleicht sich der Einfluss beider Anforderungen hinsichtlich des Bezugsobjekts. Ist die Priorität einer der beiden Anforderungen geringer oder unterscheiden sich die Status, unterscheidet sich auch der Einfluss auf das Bezugsobjekt.

Formel 3:

$$E_{Text}(K_T) = \frac{K_T.status * K_T.prio}{\sum_i^n K_{T,i}.status * K_{T,i}.prio}$$

$E_{Text}(K_T)$	Einfluss des zu ändernden Textelements K_T
$K_T.status$	Status des zu ändernden Textelements K_T
$K_T.prio$	Priorität des zu ändernden Textelements K_T
n	Anzahl aller verbundenen Text-Elemente $K_{T,i}$
$K_{T,i}.status$	Status des Textelements $K_{T,i}$
$K_{T,i}.prio$	Priorität des Textelements $K_{T,i}$

Relativ zum Gesamtprojekt spielt der Einfluss des Bezugsobjekts (der Systemkomponente) eine entscheidende Rolle. Ist die Komponente eher unwichtig, hat auch eine Änderung einer ihrer Anforderungen nur geringe Auswirkungen auf das Gesamtprojekt. Handelt es sich jedoch um eine kritische Komponente, so kann eine Änderung einer ihrer Anforderungen erhebliche Auswirkungen auf das Gesamtprojekt haben. Folglich folgt für den Gesamteinfluss eines Textelements folgender Zusammenhang:

Formel 4:

$$E_{T,ges}(K_T) = E_{Text}(K_T) * E_{Komp}(K_T.BZO)$$

$E_{T,ges}(K_T)$	Gesamteinfluss des zu ändernden Textelements K_T
$E_{Text}(K_T)$	Einfluss des zu ändernden Textelements K_T
$E_{Komp}(x)$	Einfluss der Systemkomponente x
$K_T.BZO$	Bezugsobjekt des Textelements K_T

4.7.3 Informationen prüfen und freigeben

Je nach Art der Änderung sind unterschiedliche Informationen für den Anwender von Interesse. Werden beispielsweise Kontextelemente, welche hierarchische Kindelemente haben gelöscht, sollte der Anwender überprüfen, ob diese ebenfalls gelöscht werden sollen oder nicht. Werden bestehende Kontextelemente geändert, ist für den Anwender zudem von Interesse, was geändert wurde, welche Kontextelemente betroffen sind, welche möglichen Aufgaben daraus resultieren und wie sich dies auf das Gesamtprojekt auswirkt. Selbiges gilt, wenn neue Kontextelemente hinzukommen. Werden beispielsweise neue Unterprojekte, neue Systemkomponenten oder Aufgaben angelegt, muss überprüft werden ob das Gesamtbudget und die -laufzeit des Projekts durch das neue Unterprojekt, Aufgaben oder Systemkomponenten überschritten werden. Ist dies der Fall, muss das entsprechende Kontextelement entsprechend umgeplant werden oder das Budget und/oder die -laufzeit des Gesamtprojektes angepasst werden.

Je weiter fortgeschritten ein Projekt ist, je kritischer werden neue oder sich ändernde Anforderungen, da die damit verbundenen Auswirkungen mit reinen Budget- und Zeitprüfungen nicht abschätzbar sind. Aus diesem Grund sieht das vorliegende Konzept ein Userdialogsystem vor, welches den Anwender bei der Beantwortung folgender Fragen unterstützen soll:

1. Was wurde geändert/hinzugefügt?
2. Welche Kontextelemente sind durch diese Änderung betroffen?
3. Welchen Einfluss haben diese Elemente auf das Gesamtprojekt?
4. Welche möglichen Aufwände resultieren aus dieser Änderung?
5. Welche Auswirkungen hat dies auf das Gesamtprojekt?

Um die erste Frage zu beantworten, wird dem Anwender entweder das neu hinzugefügte Element oder ein Vorher-Nachher-Vergleich des geänderten Kontextelements dargestellt.

Für Frage zwei werden dem Anwender alle durch die Änderung betroffenen Kontextelement und deren Kindelemente dargestellt. Da nicht zwingend alle Kindelemente von der Änderung betroffen sein müssen, kann der Anwender

zunächst nichtrelevante Kontextelemente entsprechend kennzeichnen, um diese aus der weiteren Analyse auszuschließen.

Frage vier wird mit Hilfe der Einflussabschätzung (siehe Kapitel 4.7.2) beantwortet. Hierzu wird der Einfluss jedes relevanten Kontextelements abgeleitet (siehe Kapitel 4.7.2) und dem Anwender zusätzlich dargestellt (siehe Abbildung 4.30).

Änderung_2020_01_01_003		Relevant	Nicht relevant
Vorher: Das Auto muss grün sein.			
Nachher: Das Auto muss gelb sein.			
▼ Auto (E=100%)		<input checked="" type="checkbox"/>	<input type="checkbox"/>
▶ [3] Anforderungen		<input type="checkbox"/>	<input type="checkbox"/>
▶ [ID:ANF_312, E=50%] Das Auto muss eine Höchst		<input type="checkbox"/>	<input checked="" type="checkbox"/>
▶ [ID:ANF_12, E=25%] Das Auto sollte gelb sein.		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
▶ [ID:ANF_10, E=25%] Das Auto sollte 4 Räder haben.		<input type="checkbox"/>	<input checked="" type="checkbox"/>
▶ [1] Informationen		<input type="checkbox"/>	<input checked="" type="checkbox"/>
▼ [3] Sub-Systemkomponenten		<input type="checkbox"/>	<input type="checkbox"/>
▶ Motor (E=40%)		<input type="checkbox"/>	<input checked="" type="checkbox"/>
▶ Getriebe (E=15%)		<input type="checkbox"/>	<input checked="" type="checkbox"/>
▶ Chassis (E=4.5%)		<input checked="" type="checkbox"/>	<input type="checkbox"/>

Abbildung 4.30: Einflussprüfungsdialog

Nachdem alle nicht relevanten Kontextelemente ausgeblendet wurden, kann der Anwender mögliche resultierende Aufwände für jedes betroffene Kontextelement abschätzen und die dazugehörigen Aufgabenbeschreibungen eintragen, geführt durch einen weiteren Userdialog. Die dabei eingetragenen Werte werden parallel mit dem geplanten Budget und der Laufzeit des Projektes abgeglichen und grafisch dargestellt, um den Anwender dabei zu unterstützen, abzuschätzen, welche Auswirkungen die Änderung auf das Gesamtprojekt haben wird (siehe Abbildung 4.31).

Änderung_2020_01_01_003	
Vorher: Das Auto muss grün sein.	
Nachher: Das Auto muss gelb sein.	
Mögliche Aufwände für folgende Systemkomponenten	
<ul style="list-style-type: none"> • Chassis 3000 €, von 01.05.2020 bis 30.12.2020, Beschreibung • ... 	
Aktueller Projektstatus	<input checked="" type="checkbox"/> Ist-Stand <input type="checkbox"/> Änderungsauswirkung
Laufzeit	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Budget 45.013 € von 96.000 €

Abbildung 4.31: Aufwandsabschätzungsdialog

Sobald der Anwender den Prozess abgeschlossen hat, wird für die Änderung ein Änderungsprojekt angelegt mit je einer Aufgabe pro betroffenem Kontextelement inkl. der abgeschätzten Aufwände, welches mögliche Änderungen bedarf. Die entsprechenden Aufgaben werden automatisch den betroffenen Kontextelementverantwortlichen zugeordnet, wodurch gleichzeitig eine Kommunikation der Änderung gewährleistet wird (siehe Abbildung 4.32).

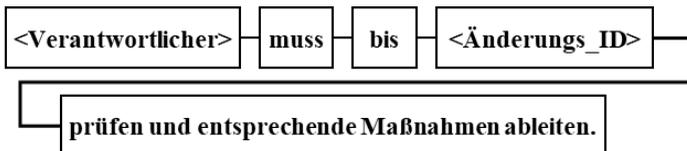


Abbildung 4.32: Satzschablone einer Änderungsaufgabe

Die Aufgabenverantwortlichen können anschließend die jeweiligen Aufgaben hinsichtlich des Aufwands und der Laufzeit prüfen, ggf. anpassen und anschließend zur Bearbeitung freigeben. Sollte sich hierbei der Aufwand oder die Laufzeit einer der Aufgaben nachteilig nach oben verändern, wird der Änderungsverantwortliche entsprechend informiert, damit dieser erneut das Änderungsprojekt prüfen und ggf. anpassen kann.

4.8 Fazit

Wie in Kapitel 3 herausgearbeitet wurde, sind aktuell am Markt erhältliche Softwarelösungen zum Verwalten von Anforderungen und dem damit in Verbindung stehenden Wissen für Nicht-Experten ungeeignet. Um projektrelevante Informationen jedoch allen Projektbeteiligten zur Verfügung zu stellen, müssen auch Nicht-Experten befähigt werden Informationen zu erstellen und verwalten zu können. Folglich bedarf es digitaler Assistenzsysteme, welche den Anwender kontextsensitiv unterstützen.

Im Rahmen dieses Kapitels wird daher ein Konzept vorgestellt, welches die wesentlichen Aufgaben des anforderungsbasierten Wissensmanagements kontextsensitiv unterstützt. Hiermit soll es auch Nicht-Experten ermöglicht werden, vollständige und semantisch verknüpfte Projektdokumentationen zu dokumentieren und semantisch miteinander zu verknüpfen. Basierend auf der so entstehenden zentralisierten semantischen Datenbasis, werden Konzepte zur unterstützten Wiederverwendung bereits dokumentierter Informationen vorgestellt sowie ein Verfahren zur teilautomatisierten Tragweitenanalyse möglicher Änderungen.

5 Implementierung

Für die prototypische Realisierung des vorliegenden Konzepts wurde das Python-Framework Django¹ verwendet. Es bietet neben umfangreichen vorkonfigurierten Vorlagen für die Benutzeroberfläche auch ein integriertes Datenbankmanagement, welche für die Umsetzung des semantischen Datenbankmodells verwendet wird. Die Umsetzung der einzelnen Assistenzmodule, lässt sich mit Hilfe etablierter Python-Bibliotheken in das Django-Framework einbetten (siehe Abbildung 5.1).

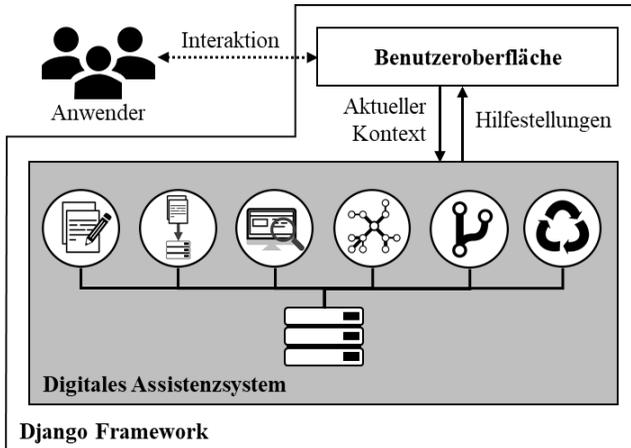


Abbildung 5.1: Aufbau des digitalen Assistenzsystems

Nachfolgend wird die Implementierung der zuvor in Kapitel 4 beschriebenen Konzepte der Assistenzmodule beschrieben.

¹ <https://www.djangoproject.com/>, zuletzt geprüft am 20.08.2020

5.1 Dokumentationsassistent

Wie in Kapitel 4.2 beschrieben wurde, hat der Dokumentationsassistent zum Ziel die Dokumentation von Text-Elementen kontextsensitiv zu unterstützen (siehe Abbildung 5.2).

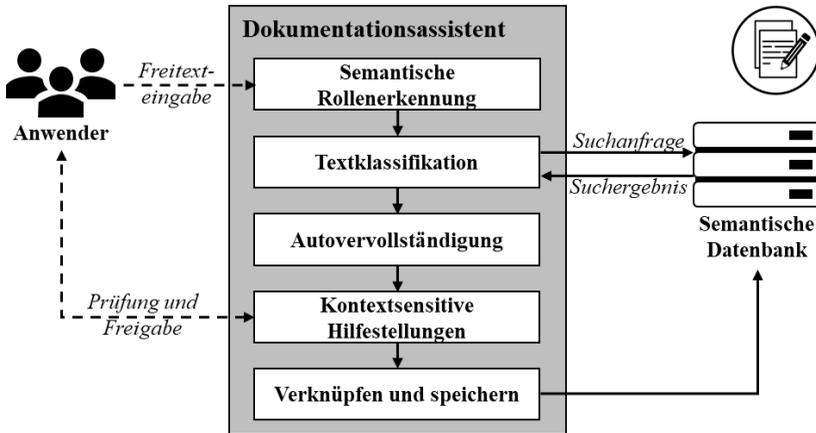


Abbildung 5.2: Aufbau des Dokumentationsassistenten

5.1.1 Semantische Rollenerkennung

Zur Extraktion von Informationen und ihren möglichen Relationen aus textuellen Beschreibungen, kommt die Python Bibliothek spaCy² zum Einsatz. Sie bietet im Vergleich zu anderen etablierten NLP-Bibliotheken wie NLTK³ oder CoreNLP⁴ insbesondere für die deutsche Sprache erweiterte POS-Tags und Abhängigkeiten. Beispielsweise kann mit Hilfe von spaCy das Genitivobjekts⁵

² <https://spacy.io/>, zuletzt geprüft am 20.08.2020

³ <https://www.nltk.org/>, zuletzt geprüft am 20.08.2020

⁴ <https://nlp.stanford.edu/>, zuletzt geprüft am 20.08.2020

⁵ <https://de.wikipedia.org/wiki/Genitiv>, zuletzt geprüft am 20.08.2020

– welche für die Identifikation des Elternsubjekts/-objekts benötigt wird – automatisch identifiziert werden (siehe auch Kapitel 2.4.6). Zudem ist spaCy speziell für die Anwendung in industriellen Kontexten entwickelt worden, weshalb sie andere Implementierungen in den üblichen Aufgaben übertrifft [Coh 19].

Zur Erkennung der semantischen Rollen wird hierbei eine Kombination aus linguistischen Regeln, Wortlisten und eine Eigennamenerkennung (engl. Named Entity Recognition, kurz NER) verwendet. Nachfolgend werden die für die Erkennung der semantischen Rollen verwendeten Verfahren aufgezählt. Die zur Beschreibung der linguistischen Regeln verwendeten Abkürzungen zur Beschreibung der Part-of-speech (kurz POS) und Dependency-Tags (kurz Dep) werden zur Wahrung der Übersichtlichkeit nicht erläutert⁶:

ID	Semantische Rolle	Erkennungsverfahren
1	Akteur	POS: NOUN, PROPN Dep: ag, pnc oder nk des Objekts
2	Datum	NER
3	Eigenschaft	NER
4	Eigenschaftswert	NER
5	Einheit	NER
6	Eltern-Objekt	POS: NOUN, PROPN Dep: ag, pnc oder nk des Objekts
7	Eltern-Subjekt	POS: NOUN, PROPN Dep: ag, pnc oder nk des Subjekts
8	Events	<u>NER</u>
9	Funktionswort	NER
10	Logischer Operator	NER
11	Objekt	POS: NOUN, PROPN Dep: oa
12	Person	NER
13	Präzisierung	POS: alle

⁶ <https://spacy.io/api/annotation>, zuletzt geprüft am 20.08.2020

		Dep: nk, cm oder pnc und POS: ADP oder ADV DEP: mo, mnr, op
14	Prioritätswort	POS: VERB, VMFIN Dep: ROOT Zusätzlich wird hier eine Wortliste genutzt (siehe weiter unten)
15	Prozesswort	POS: VERB, AUX Dep: oc, cj, mo
16	Subjekt	POS: NOUN, PROPN Dep: sb oder ROOT
17	Systemkomponente	NER
18	Vergleichsoperator	NER

Tabelle 8: Definition der Erkennungsverfahren

Um Eigenschaften innerhalb einer Anforderung zu identifizieren, reichen linguistische Regeln alleine nicht aus (siehe Abbildung 5.3).

Anforderung 07:
 Die Farbe des Autos muss gelb sein.
 └───┬───
 Subjekt

Anforderung 08:
 Die Tür des Autos muss gelb sein.
 └───┬───
 Subjekt

Abbildung 5.3: Beispielanforderungen

Wie in diesem Beispiel dargestellt, wird die „Farbe“ des Autos sowie die „Tür“ des Autos als Subjekt identifiziert. Nur anhand des semantischen Wissens über die Unterschiede zwischen beiden Subjekten, ist der Mensch im Stand die Eigenschaft „Farbe“ von der Subkomponente „Tür“ des Autos zu unterscheiden. Ähnliches gilt auch für andere semantische Rollen, welche sich rein mit linguistischen Regeln nicht mit ausreichender Genauigkeit erkennen lassen. Aus

diesem Grund werden an dieser Stelle Maschinelle Lernansätze verwendet, wie nachfolgend beschrieben.

Einige semantische Rollen lassen sich allein durch linguistische Regel oder regulären Ausdrücken ⁷ nicht mit ausreichender Sicherheit identifizieren. Daher wird die in spaCy integrierte Named Entity Recognition (kurz NER) verwendet, welche auf einem CNN (siehe Kapitel 2.3.2) basiert und speziell auf die genannten semantischen Rollen trainiert wird. Zur Erzeugung der dafür notwendigen Trainingsdaten wird das Werkzeug doccano⁸ verwendet, welches die Klassifikation von Sätzen als auch einzelnen Satzbausteinen ermöglicht, jedoch keine geschachtelten Entitäten oder Relationen annotieren kann, welche für die vorliegende Arbeit an dieser Stelle jedoch nicht notwendig sind.

Da auch dieser Ansatz je nach Domäne nicht immer korrekt ist, sieht das vorliegende Konzept einen Userdialog vor, welcher den Anwender in die Prüfung der automatisch identifizierten semantischen Rollen einbezieht. Werden Abweichungen durch den Anwender identifiziert, werden die durch den Anwender korrigierten Annotationen zwischengespeichert und für das nächste Training des NER-Modells verwendet. Um die Qualität hierbei zu sichern, müssen diese durch den Administrator geprüft und freigegeben werden. So kann eine hohe initiale Genauigkeit gewährleistet werden und sichergestellt werden, dass das Sprachmodell die unternehmensspezifischen Eigennamen mitlernt.

5.1.2 Textklassifikation

Um zu erkennen, welche Art von Text-Element ein Satz entspricht, werden die Sätze mit Hilfe der Text-Klassifikationspipeline in spaCy klassifiziert. Diese wird hierzu auf den speziellen Anwendungsfall der Unterscheidung zwischen Anforderungen, Aufgaben und allgemeinen Informationen trainiert. Hierzu werden entsprechende Beispielsätze mit Hilfe der spaCy integrierten Funktion

⁷ https://de.wikipedia.org/wiki/Regulärer_Ausdruck, zuletzt geprüft am 20.08.2020

⁸ <https://github.com/doccano/doccano>, zuletzt geprüft am 20.08.2020

tok2vec in Vektoren überführt mit denen ein Convolutional Neural Network trainiert wird, welches ebenfalls innerhalb von spaCy vorkonfiguriert vorliegt⁹.

5.1.3 Autovervollständigung

Für die Realisierung der Autovervollständigungsfunktion wird die Bibliothek JQuery-UI Autocomplete¹⁰ verwendet, welche die Wortvorschläge dem Anwender darstellt (siehe Abbildung 5.4).

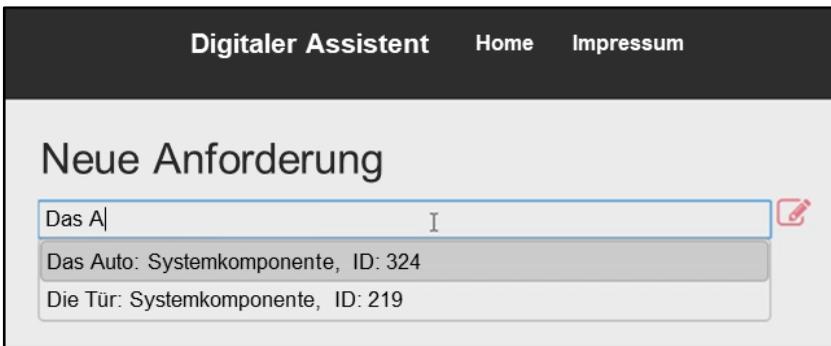


Abbildung 5.4: Screenshot der Autovervollständigung

Wie hier zu sehen, wird anhand der aktuellen Eingabe die nächsten Wortvorschläge entsprechend gefiltert. Anstelle reiner Textbausteine – wie bei herkömmlichen Autovervollständigungen üblich – werden dem Anwender Objektpreferenzen angeboten. Um dies dem Anwender kenntlich zu machen, werden sowohl die Kontextelement-Art als auch die ID innerhalb der Datenbank angezeigt.

⁹ https://github.com/explosion/spaCy/blob/master/examples/training/train_textcat.py, zuletzt geprüft am 20.08.2020

¹⁰ <https://jqueryui.com/autocomplete/>, zuletzt geprüft am 20.08.2020

5.1.4 Kontextsensitive Hilfestellungen

Wie bereits in Kapitel 4.2.4 beschrieben wurde, bietet das digitale Assistenzsystem dem Anwender kontextsensitive Hilfestellungen, um auch Nicht-Experten zu befähigen vollständige und standardisierte textuelle Beschreibungen zu erstellen. Im ersten Schritt sieht das Konzept kontextsensitive Referenzbeispiele vor, welche entsprechend der Eingabe dem Anwender kontextsensitiv dargestellt werden (siehe auch Kapitel 5.1.3). Zudem werden die eingegebenen Texte hinsichtlich ihrer Vollständigkeit mit den vordefinierten Satzschablonen abgeglichen.

Parallel zu den Ergebnissen der Prädiktion des nächsten Wortes und der Textklassifikation, werden dem Anwender Referenzbeispiele angezeigt. Sollte die Eingabe unvollständig sein und eine semantische Rolle wie z.B. die Priorität nicht innerhalb der Eingabe vorkommen, werden zudem Hilfsfragen eingeblendet inkl. eines Texteingabefeldes oder eines Dropdowns (sofern es vordefinierte Worte für die fehlende Rolle gibt), wie im folgenden Beispiel dargestellt (siehe Abbildung 5.5).

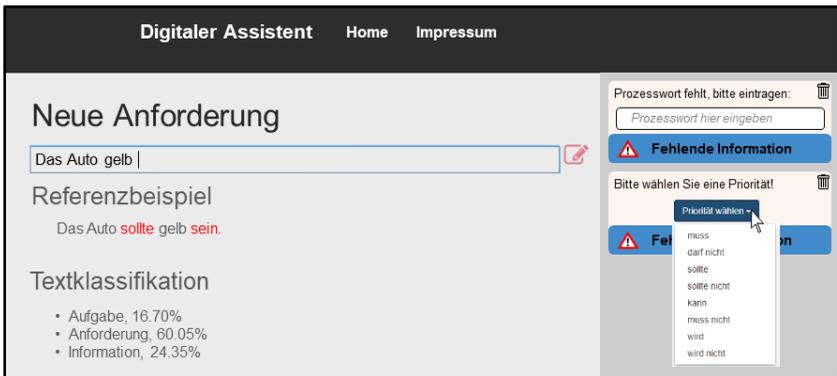


Abbildung 5.5: Screenshot der kontextsensitiven Hilfestellungen

Wie hier zu sehen, wird dem Anwender anhand der aktuellen Eingabe das angepasste Referenzbeispiel angeboten (siehe links unter der Texteingabe). Fehlen semantische Rollen innerhalb der Eingabe, werden rot gekennzeichnet im Referenzbeispiel. Auf der rechten Seite wird der Anwender in Form von Tickets bei der Korrektur unterstützt. In diesem Beispiel fehlen die beiden semantischen Rollen „Priorität“ und das „Prozesswort“. Für das Prozesswort, wird dem Anwender ein Eingabefeld Angebote, welches er nutzen kann. Er kann jedoch auch weiterhin das Eingabefeld links oben nutzen. Für die Priorität kann der Anwender aus vordefinierten Begriffen wählen, welche ebenfalls automatisch in das Textfeld überführt werden.

5.2 Integrationsassistent

Wie in Kapitel 4.3 beschrieben, hat der Integrationsassistent zum Ziel, die Integration von Anforderungen und Systembeschreibungen aus Anforderungsdokumenten (z.B. Lasten- und Pflichtenheft) in die semantische Datenbank, kontextsensitiv zu unterstützen (siehe Abbildung 5.6).

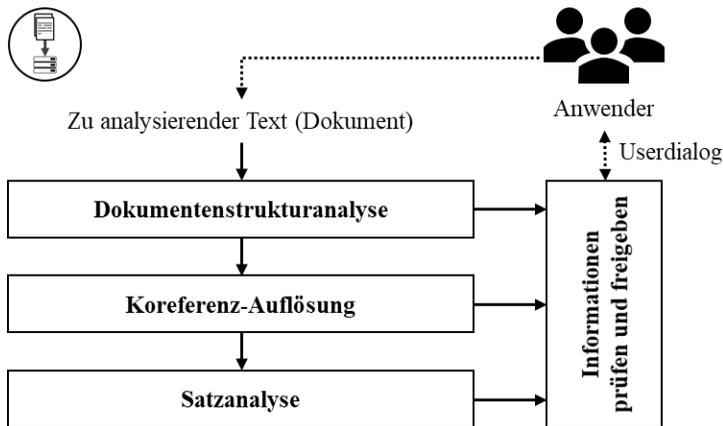


Abbildung 5.6: Funktionaler Aufbau des Integrationsassistenten

Nachfolgend wird die prototypische Implementierung der einzelnen Funktionen beschrieben.

5.2.1 Dokumentenstrukturanalyse

Für die Analyse der Dokumente wird auf der Django basierten Bibliothek `django_ocr_server`¹¹ aufgebaut, welche bereits über Funktionen wie Dokumentupload und OCR-Analysen verfügt. Hiermit lassen sich PDF-Dokumente am Frontend hochladen, welche anschließend durch die auf `tesseract-ocr`¹² basierte Implementierung `pdf2text` in Text umgewandelt werden.

5.2.2 Koreferenz-Auflösung

Um Satzübergreifende Abhängigkeiten auflösen zu können, werden in der Computerlinguistik Koreferenz-Auflösungsverfahren verwendet. Speziell für die deutsche Sprache konnten stand heute nur die regelbasierte Implementierung `CorZu` identifiziert werden [Tug 16]. Aufgrund ihres rein regelbasierten Ansatzes, ist diese Implementierung jedoch sehr fehleranfällig und identifiziert z.B. nicht mal die Zusammenhänge zwischen Nomen und Pronomen (siehe Abbildung 5.7).

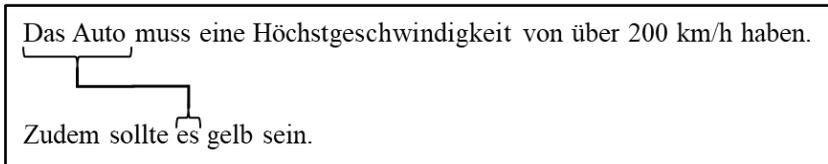


Abbildung 5.7: Beispiel einer Koreferenz

¹¹ <https://django-ocr-server.readthedocs.io/>, zuletzt geprüft am 20.08.2020

¹² <https://github.com/tesseract-ocr/>, zuletzt geprüft am 20.08.2020

Bibliotheken wie spaCy oder CoreNLP bieten nativ keine Unterstützung für eine deutsche Koreferenz-Auflösung. Aus diesem Grund musste für die vorliegende Arbeit mit Hilfe eines öffentlich zugänglichen Datensatzes eine Koreferenz-Auflösung trainiert werden. Eine der standard he Art Implementierungen in diesem Bereich stellt die Bibliothek NeuralCoref 4.0¹³ dar [Wol 17], welche speziell als Erweiterung für spaCy entwickelt wurde und sich daher ideal in die bisherige Systemstruktur eingliedert.

Systemkomponenten-Namen können innerhalb von Anforderungsdokumenten mehrfach vorkommen, obwohl diese nicht die gleiche Komponente bezeichnen. Um dieses Problem, bei der automatischen Erkennung von Koreferenzen zu lösen, werden gleiche Systemkomponenten-Namen zwischengespeichert und als möglich Koreferenzen dem Anwender zur Prüfung bereitgestellt. Für die Erkennung möglicher Synonyme wird die Synonymdatenbank Thesaurus¹⁴ verwendet und die Ergebnisse dem Anwender zur Integration angeboten (siehe Abbildung 5.8).

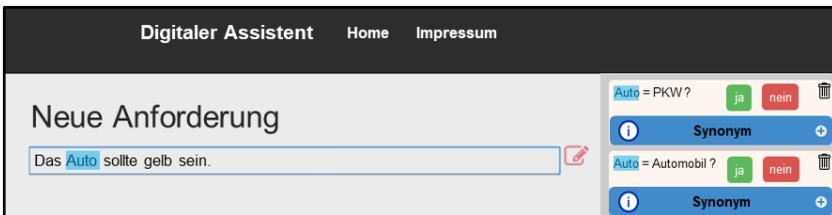


Abbildung 5.8: Screenshot der Synonym-Integration

Wie hier dargestellt, werden dem Anwender kontextsensitiv mögliche Synonyme angeboten, welche er entweder als solche betätigen kann oder verwerfen. Bestätigt der Anwender ein Synonym, wird dieses in der Datenbank an die Systemkomponente (in diesem Fall das „Auto“) angefügt.

¹³ <https://github.com/huggingface/neuralcoref>, zuletzt geprüft am 20.08.2020

¹⁴ <https://www.thesaurus.com/>, zuletzt geprüft am 20.08.2020

5.2.3 Satzanalyse

Die Analyse der einzelnen Sätze des zuvor aufbereiteten Textes, werden mit Hilfe der semantischen Rollenerkennung analysiert, um die darin enthaltenen Informationen und Relationen zu extrahieren (siehe hierzu Kapitel 5.1.1).

5.2.4 Informationen prüfen und freigeben

Um die Ergebnisse des Integrationsassistenten prüfen und freigeben zu können, sieht das vorliegende Konzept wie in Kapitel 4.3.4 beschrieben, zwei Us-erdialoge vor: Prüfdialog und Integrationsdialog. Für den Prüfdialog wird das PDF-Dokuments mit Hilfe des PDF.js¹⁵ Viewer dargestellt. Für die Miniaturdarstellung des gesamten Dokuments wird die Implementierung pagemap¹⁶ verwendet. Die Tickets sind HTML basiert und selbst entwickelt (siehe Abbildung 5.9).

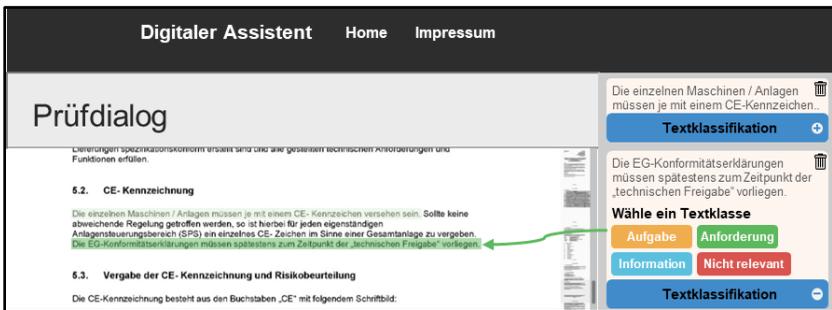


Abbildung 5.9: Mockup des Prüfdialogs

Wie hier zu sehen ist, wird dem Anwender links das Originaldokument dargestellt, augmentiert durch die erkannten Informationen. Auf der rechten Seite werden die zu prüfenden Informationen in Form von Tickets dargestellt. Für

¹⁵ <https://mozilla.github.io/pdf.js/>, zuletzt geprüft am 20.08.2020

¹⁶ <https://larsjung.de/pagemap/>, zuletzt geprüft am 20.08.2020

den Integrationsdialog wurde `jquerytreetable`¹⁷ verwendet und angepasst, um alle erkannten Kontextelemente darzustellen (siehe Abbildung 5.10).

Found the following Requirements			
Requirement	Matching Score	integrate	ignor
Das Auswählen und Kaufen von Autoteilen mit unserer Hilfe wird eine angenehme Erfahrung für Sie sein!	0.7684242725372314	<input type="checkbox"/>	<input type="checkbox"/>
Die besonderen Anforderungen an ein für Fahranfänger geeignetes Auto machen die Suche nicht einfach.	0.7600272297859192	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Wer problemlos Auto fahren möchte, sollte zu einem Gebrauchten mit neu durchgeführter Hauptuntersuchung greifen.	0.7598123550415039	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Autos mit Seitenairbags zwischen den Sitzplätzen dürften hier künftig deutlich besser abschneiden.	0.7486659288406372	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Wenn Sie einen vollständig abgerundeten Schutz für Ihr Auto möchten, empfehlen wir Ihnen die Vollkasko.	0.741127610206604	<input type="checkbox"/>	<input type="checkbox"/>
Found the following Information			
Information	Matching Score		
An einem Auto sind viele Bauteile auf Verschleiß ausgelegt.	0.7735753059387207	<input type="checkbox"/>	<input type="checkbox"/>
Deren anziehende und abstoßende Kräfte setzen das Auto in Bewegung.	0.7565983533859253	<input type="checkbox"/>	<input type="checkbox"/>
Der Rotor dreht sich und bringt das Auto in Bewegung.	0.7529045939445496	<input type="checkbox"/>	<input type="checkbox"/>

Abbildung 5.10: Screenshot des Integrationsdialogs

Bei dem hier dargestellten Beispiel handelt es sich um ein Auto, dessen Anforderungen, Informationen sowie die damit verbundenen Aufgaben und Sub-Komponenten. Wie in Abbildung 5.10 dargestellt, werden dem Anwender die erkannten Systemkomponenten, deren Anforderungen, Informationen und die erkannten Aufgaben zur Prüfung bereitgestellt. Mit Hilfe der Checkbox auf der

¹⁷ <http://ludo.cubicphuse.nl/jquery-treetable/>, zuletzt geprüft am 20.08.2020

rechten Seite, kann der Anwender anschließend wählen, welche Kontextelemente er integrieren und welche er verwerfen möchte.

5.3 Rechercheassistent

Wie in Kapitel 4.4 beschrieben, hat der Rechercheassistent zum Ziel, die Recherchearbeiten kontextsensitiv zu unterstützen. Hierzu werden anhand des aktuellen Kontextes des Anwenders, möglicherweise relevante Informationen aus dem Internet gecrawlt und dem Anwender zur Integration in das aktuelle Projekt bereitgestellt (siehe Abbildung 5.11).

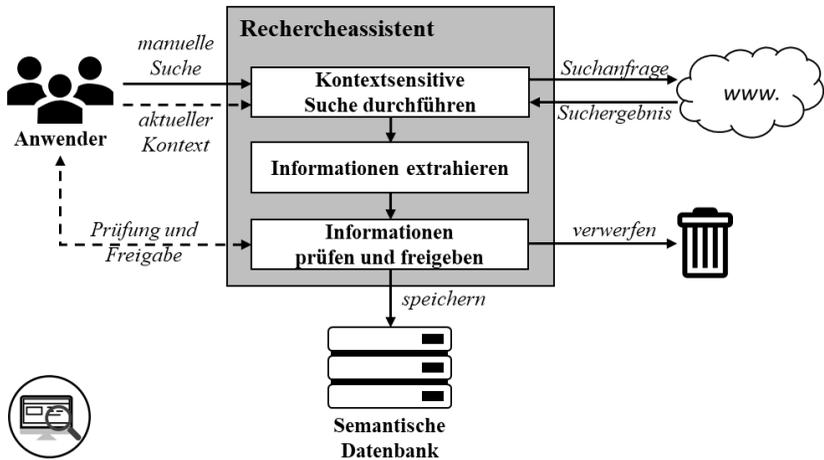


Abbildung 5.11: Funktionaler Aufbau des Rechercheassistenten

Nachfolgend wird die prototypische Implementierung der einzelnen Funktionen beschrieben.

5.3.1 Kontextsensitive Suche durchführen

Um entsprechend des aktuellen Kontextes des Anwenders adäquate Informationen bereitstellen zu können, werden mit Hilfe eines JQuery-Scripts alle aktuell durch den Anwender einsehbaren Kontextelemente identifiziert. Anschließend werden Suchanfragen erzeugt, welche mit Hilfe der Python Implementierung `google-api-python-client`¹⁸ der Google-API an Google übermittelt werden. Hierzu werden folgende Suchanfragen basierend auf den aktuell einsehbaren Kontextelementen generiert, welche durch den Anwender ergänzt werden können:

ID	Suchanfrage
1	„Bauteile einer/eines <Systemkomponente.name>“
2	„technische Eigenschaften einer/eines<Systemkomponente.name>“
3	„was muss ein <Systemkomponente.name> können?“

Tabelle 9: Beispiel Suchanfragen

Die Textteile „<Systemkomponente.name>“ werden entsprechend durch die durch den Anwender sichtbaren Systemkomponenten-Namen ersetzt. Die Suchergebnisse werden anschließend für die Informationsextraktion bereitgestellt.

5.3.2 Informationen extrahieren

Um die Texte und Überschriften aus den HTML-Seiten zu extrahieren, wird die Scraping-Bibliothek `BeautifulSoup`¹⁹ eingesetzt. Mit ihrer Hilfe lassen sich die Überschriften und Textpassagen einfach aus den HTML-Rohdaten extrahieren. Iterativ wurde identifiziert, dass die meisten Webseiten Ihre Textinhalte in Sektionen (in HTML <section>) einteilen. Häufig werden diese Sektionen durch Überschriften (engl. Header) angeführt, welche sich mit Hilfe des

¹⁸ <https://github.com/googleapis/google-api-python-client>, zuletzt geprüft am 20.08.2020

¹⁹ <https://www.crummy.com/software/BeautifulSoup/>, zuletzt geprüft am 20.08.2020

HTML-Tags ‚<h>‘ gefolgt durch eine Ganzzahl (z.B. ‚<h1>‘) – welche die absteigende Hierarchie der Überschriften darstellt – erkennen lassen.

Hin und wieder werden innerhalb der Sektionen Bilder oder Codesnippets eingefügt, welche für die reine Textverarbeitung zunächst nicht von Bedeutung sind, weshalb diese vor der weiteren Verarbeitung der Textinhalte gelöscht werden müssen. Um die Informationen jedoch nicht zu verlieren, werden den jeweiligen Textpassagen ein Link angefügt, damit der Anwender bei Interesse, die jeweilige Seite besuchen kann, um mehr zu erfahren. Der anschließende Resttext der Sektionen, kann anschließend analog zur Textverarbeitung des Integrationsassistenten analysiert werden, um mögliche Anforderungen, Aufgaben oder Informationen für den aktuellen Kontext des Anwenders zu identifizieren. Anstelle der Detailsansicht im PDF, kann der Anwender über den zwischengespeicherten Link auf die Webseite gelangen, um sich ggf. noch weiter über die gefundene Information zu informieren.

5.3.3 Informationen prüfen und freigeben

Zur Integration der durch den Rechercheassistenten identifizierten Kontextelemente, wird der Integrationsdialog genutzt (siehe Kapitel 5.2.4).

5.4 Verknüpfungsassistent

Wie in Kapitel 4.5 beschrieben, hat der Verknüpfungsassistent zum Ziel, mögliche noch nicht miteinander verknüpfte Kontextelemente automatisch miteinander zu verknüpfen. Hierdurch soll zum einen der damit verbundene manuelle Aufwand reduziert und sichergestellt werden, dass alle zu verknüpfenden Informationen miteinander verknüpft sind (siehe Abbildung 5.12).

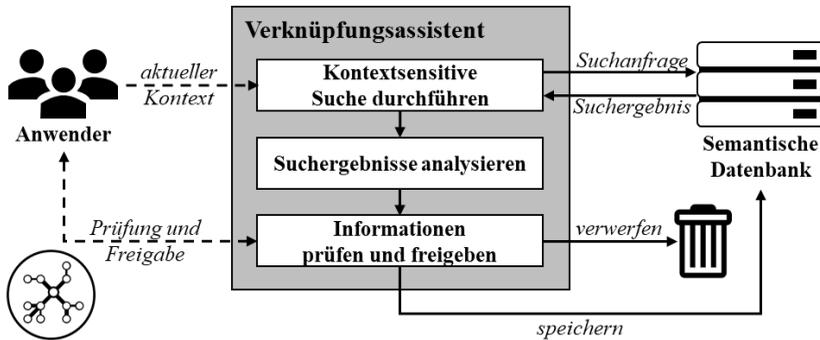


Abbildung 5.12: Funktionaler Aufbau des Verknüpfungsassistenten

Nachfolgend wird die prototypische Implementierung der einzelnen Funktionen beschrieben.

5.4.1 Kontextsensitive Suche durchführen

Analog zur Kontextanalyse des Rechercheassistenten, werden anhand der aktuell durch den Anwender einsehbaren Kontextelementen, Suchanfragen generiert. Im Falle des Verknüpfungsassistenten, wird jedoch nicht das Internet, sondern direkt die semantische Datenbank durchsucht. Hierzu werden die Namen und textuellen Beschreibungen der einsehbaren Kontextelemente verwendet, um mögliche Redundanzen, konfligierende Informationen oder potentielle Relationspartner zu identifizieren.

5.4.2 Suchergebnisse analysieren

Mögliche Redundanzen liegen dann vor, wenn sich Text-Elemente semantisch sehr stark ähneln. Um dies automatisiert zu erkennen, können Sätze mit Hilfe vortrainierter Sprachmodelle, in Satzvektoren überführt werden. Anschließend lässt sich die Kosinus-Ähnlichkeit²⁰ beider Vektoren bestimmen, welche die

²⁰ <https://de.wikipedia.org/wiki/Kosinus-%C3%84hnlichkeit>, zuletzt geprüft am 20.08.2020

Ähnlichkeit beider Vektoren ausdrückt. Das Ergebnis kann zwischen -1 (genau entgegengerichtet) und 1 (genau gleichgerichtet) liegen, wobei eine 0 eine Unabhängigkeit beider Vektoren signalisiert.

Dieses Verfahren macht sich die Bibliothek `sentence-transformers`²¹ zu Nutze. Für die vorliegende Arbeit wird hierzu das vortrainierte Sprachmodell „`distiluse-base-multilingual-cased`“²² verwendet. Die `sentence-transformers` Bibliothek erzeugt aus den zu vergleichenden Sätzen, Satzvektoren, welche anschließend mit Hilfe der Bibliothek `SciPy`²³ und der Funktion `scipy.spatial.distance` zu Berechnung der Kosinus-Ähnlichkeit verwendet werden. Sobald die Ähnlichkeit einen positiven Wert von 90% überschreitet, werden die beiden möglichen Redundanzen dem Anwender zur Prüfung bereitgestellt.

Um mögliche Konflikte oder potentielle Relationspartner zu erkennen, kann die sogenannte Natural Language Interferenz (NLI) genutzt werden. Mit Hilfe dieses Verfahrens, lassen sich Sätze hinsichtlich der dreier Abhängigkeitsarten analysieren: „neutral“, „widersprüchlich“ und „beinhaltend“ [Bow].

Konflikte liegen dann vor, wenn sich zwei Sätze widersprechen. Sätze deren Relation zueinander als „neutral“ bewertet wird, sind weder Redundant, konfliktierend oder potentielle Relationspartner. Satzrelationen welche als „beinhaltend“ klassifiziert werden, stellen mögliche Relationspartner dar. Für die Erkennung der drei NLI-Klassen, wird die Bibliothek `SemBERT`²⁴ verwendet.

²¹ <https://github.com/UKPLab/sentence-transformers>, zuletzt geprüft am 20.08.2020

²² <https://github.com/UKPLab/sentence-transformers/blob/master/docs/pretrained-models/multilingual-models.md>, zuletzt geprüft am 20.08.2020

²³ <https://www.scipy.org/>, zuletzt geprüft am 20.08.2020

²⁴ <https://github.com/cooelf/SemBERT>, zuletzt geprüft am 20.08.2020

5.4.3 Informationen Prüfen und freigeben

Gefundene Verknüpfungen werden analog zu dem Ticketsystem des Integrations- und Rechercheassistenten als Tickets bereitgestellt. Der Anwender kann diese daraufhin prüfen und ggf. als neue Relationen erzeugen.

5.5 Wiederverwendungsassistent

Wie in Kapitel 4.6 beschrieben, hat der Wiederverwendungsassistent zum Ziel, bereits dokumentierte Informationen welche ggf. für das aktuelle Projekt relevant sind, dem Anwender kontextsensitiv zur Integration in das aktuelle Projekt bereitzustellen (siehe Abbildung 5.13). Nachfolgend wird die prototypische Implementierung der einzelnen Funktionen beschrieben.

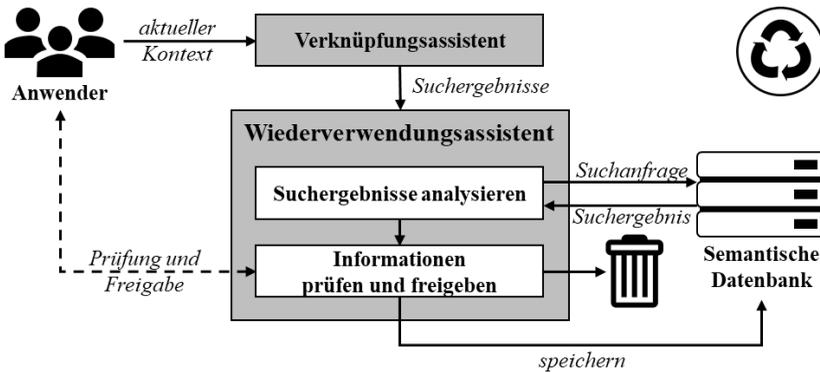


Abbildung 5.13: Aufbau des Wiederverwendungsassistenten

5.5.1 Suchergebnisse analysieren

Erfolg analog zur Ergebnisanalyse des Verknüpfungsassistenten (siehe Kapitel 5.4.2). Die so identifizierten Kontextelemente anderer Projekte, werden dem Anwender anschließend inkl. aller Kinder und Kindeskindern bereitgestellt.

5.5.2 Informationen prüfen und freigeben

Werden mögliche Wiederverwendungspartner gefunden, erhält der Anwender ein Ticket, mit der Information, dass mögliche Kontextelemente zur Integration in das aktuelle Projekt verfügbar sind. Anschließend, kann der Anwender mit Hilfe des Integrationsdialogs, die entsprechenden Kontextelemente prüfen und auswählen, ob die Kontextelemente Referenziert, Kopiert werden sollen oder nicht relevant sind (siehe Abbildung 5.14).

Found the following Requirements				
Requirement	Matching Score	ref.	copy	ignor
Das Auswählen und Kaufen von Autoteilen mit unserer Hilfe wird eine angenehme Erfahrung für Sie sein!	0.7684242725372314	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Die besonderen Anforderungen an ein für Fahranfänger geeignetes Auto machen die Suche nicht einfach.	0.7600272297859192	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Wer problemlos Auto fahren möchte, sollte zu einem Gebrauchten mit neu durchgeführter Hauptuntersuchung greifen.	0.7598123550415039	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Autos mit Seitenairbags zwischen den Sitzplätzen dürften hier künftig deutlich besser abschneiden.	0.7486659288406372	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Wenn Sie einen vollständig abgerundeten Schutz für Ihr Auto möchten, empfehlen wir Ihnen die Vollkasko.	0.741127610206604	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Found the following Information				
Information	Matching Score			
An einem Auto sind viele Bauteile auf Verschleiß ausgelegt.	0.7735753059387207	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Deren anziehende und abstoßende Kräfte setzen das Auto in Bewegung.	0.7565983533859253	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Der Rotor dreht sich und bringt das Auto in Bewegung.	0.7529045939445496	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Abbildung 5.14: Screenshot des Wiederverwendungsassistenten

5.6 Änderungsassistent

Wie in Kapitel 4.7 beschrieben wurde, hat der Änderungsassistent zum Ziel, den Anwender bei der Analyse der Tragweite einer Änderung kontextsensitiv zu unterstützen. Hierzu wird in einem ersten Schritt die Art der Änderung kategorisiert. Im zweiten Schritt werden die von der Änderung betroffenen Elemente identifiziert und hinsichtlich ihres Einflusses auf das Gesamtsystem bewertet. Anschließend werden die betroffenen Verantwortlichen informiert, welche dann die Ergebnisse grafisch visualisiert bereitgestellt bekommen inkl. der möglichen Handlungsempfehlungen (siehe Abbildung 5.15).

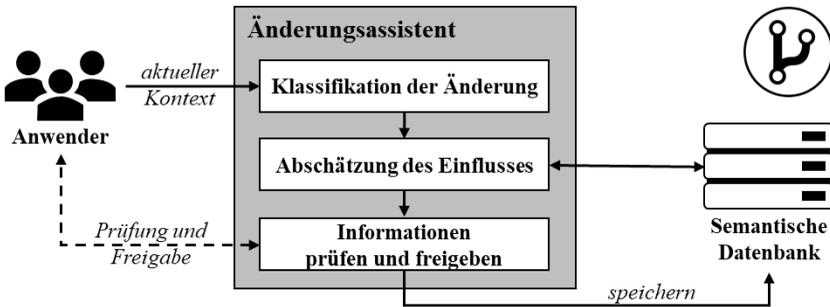


Abbildung 5.15: Funktionaler Aufbau des Änderungsassistent

Die beiden ersten Schritte erfolgen analog zu den in Kapitel 4.7.1 und Kapitel 4.7.2 beschriebenen Vorgehen. Um einheitliche Werte für Status und Priorität zu erhalten, werden für das vorliegende Konzept folgende Werte festgelegt.

Status	Gewichtung
Offen (unbearbeitet)	25%
In Bearbeitung	50%
In Prüfung	75%
Fertiggestellt	100%

Tabelle 10: Gewichtung je Status

Priorität	Gewichtung
Wird (wird nicht)	25%
Kann (muss nicht)	50%
Sollte (sollte nicht)	75%
Muss (darf nicht)	100%

Tabelle 11: Gewichtung je Priorität

Analog zu den in Kapitel 4.7 beschriebenen Formeln ergeben sich somit je Kontextelementart die entsprechenden Einflusswerte.

5.6.1 Informationen prüfen und freigeben

Analog zum Wiederverwendungsassistenten wird im Falle einer kritischen Änderung (siehe hierzu Kapitel 4.7.1) ein zusätzliches HTML-basiertes Kontextmenü dargestellt, welches den Anwender bei der Einflussprüfung der Änderung unterstützt (siehe Abbildung 5.16).

Änderung_2020_01_01_003

Vorher: Das Auto muss grün sein.

Nachher: Das Auto muss gelb sein.

	Relevant	Nicht relevant
▼ Auto (E=100%)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
▶ [3] Anforderungen	<input type="checkbox"/>	<input type="checkbox"/>
▶ [ID:ANF_312, E=50%] Das Auto muss eine Höchst	<input type="checkbox"/>	<input checked="" type="checkbox"/>
▶ [ID:ANF_12, E=25%] Das Auto sollte gelb sein.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
▶ [ID:ANF_10, E=25%] Das Auto sollte 4 Räder haben.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
▶ [1] Informationen	<input type="checkbox"/>	<input checked="" type="checkbox"/>
▼ [3] Sub-Systemkomponenten	<input type="checkbox"/>	<input type="checkbox"/>
▶ Motor (E=40%)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
▶ Getriebe (E=15%)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
▶ Chassis (E=45%)	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Abbildung 5.16: Mockup des Einflussprüfungsdialogs

Zur Unterstützung des Anwenders, werden die markanten Punkte „Subjekt“ und die geänderte semantische Rolle „Eigenschaftswert“ farblich gehighlightet. Nach der Prüfung des möglichen Einflusses, wird dem Anwender der Aufwandsabschätzungsdialog dargestellt (siehe Abbildung 5.17).

Digitaler Assistent Home Impressum

Aufwandsabschätzungsdialog

Vorher: Das Auto muss grün sein.
 Nachher: Das Auto muss gelb sein.

Mögliche Aufwände für folgende Systemkomponenten:

- Die Tür des Autos **Priorität** - bis **VON** **Verantwortlicher** -
 - Peter
 - Hans
 - Max Mustermann

Beschreibung hier eintippen

Ca. € und/oder h

- Die Farbe des Autos **Priorität** - bis **VON** **Verantwortlicher** -

Aktueller Projektstatus

Laufzeit

Geplantes Budget

Abbildung 5.17: Mockup des Aufwandsabschätzungsdialog

Die hiermit definierten Aufgaben, werden als Teil eines neuen Teilprojektes instanziiert und die mit der Änderung verbundenen Informationen werden automatisch mit dem Änderungsprojekt verknüpft. Die jeweiligen Bauteilbeauftragten werden wie in Kapitel 4.7.3 beschrieben automatisch als Verantwortliche in der Änderungsaufgabe-Satzschablone eingetragen und verknüpft, wodurch automatisch eine Kommunikation der neuen Aufgaben an die Verantwortlichen erfolgt.

5.7 Fazit

Ziel dieses Kapitels ist die Beschreibung der prototypischen Implementierung des in Kapitel 4 beschriebenen Konzepts. Hierbei werden die einzelnen Module des Konzepts sowie das semantische Datenmodell mit Hilfe des Web-Frameworks Django realisiert. Für die computerlinguistische Aufbereitung der Roh-Texte und der Extraktion der semantischen Rollen, wird das NLP-Framework spacy verwendet. Für die Klassifikation von Satzpaaren zur automatischen Erkennung von möglichen Relationspartnern, Redundanzen oder semantischen Konflikten, werden die Implementierungen sentence-transformers und SemBERT verwendet. Für die Erkennung von Koreferenzen innerhalb der Dokumente kommt NeuralCoref zum Einsatz. Zur automatischen Verarbeitung von PDF-Dokumenten wird die Bibliothek django_ocr_server eingesetzt, welche innerhalb von Bilddaten befindliche Texte automatisch extrahiert und maschinenlesbar macht.

6 Evaluation und Diskussion

Im folgenden Kapitel werden die einzelnen Module des digitalen Assistenzsystems aus technischer Sicht hinsichtlich ihrer Funktion und Praxistauglichkeit hin analysiert. Hierzu werden zunächst der Evaluationsaufbau und die verwendeten Trainings- und Testdatensätze beschrieben. Anschließend wird das vorliegende Konzept hinsichtlich der in Kapitel 3.8 abgeleiteten Anforderungen hin evaluiert. Abschließend werden die Ergebnisse zusammengefasst und mit den Zielen dieser Arbeit (Forschungsfragen) abgeglichen und diskutiert.

6.1 Evaluationsaufbau

6.1.1 Trainings- und Testdatensätze

Für das Training des NER-Modells (siehe Kapitel 5.1.1) wurden 441 Sätze für die Textklassifikation 588 Sätze manuell annotiert. Zur Evaluation werden 224 Sätze verwendet (siehe Tabelle 12):

Text-Element-Art	Trainingsdaten		Evaluationsdaten
	NER	Textklassifikation	
Anforderung	252	337	112
Aufgabe	62	83	25
Information	126	168	56
Summe	441	588	224

Tabelle 12: Annotierte Trainingsdaten

Für die Evaluation der Natural Language Inference (NLI) Aufgabe, welche für die automatische Erkennung von möglichen Redundanzen, Konflikten oder Relationspartnern genutzt wird, wurden insgesamt 147 Satzpaare aus Realdaten hinsichtlich der Klassen „neutral“, „konfligierend“ und „beinhaltend“ von

Hand klassifiziert in folgender Aufteilung (siehe Tabelle 13). Für die Evaluation der Koreferenz-Auflösung werden randomisiert 15% des TüBa-D/Z [SFS 18] Datensatzes verwendet.

Satzpaarungsart	Anzahl
Neutral	47
Konfligierend	53
Beinhaltend	47
Summe	147

Tabelle 13: Annotierte Satzpaare für die NLI-Evaluation

6.1.2 Bibliotheksversionen

Für die Evaluation der zuvor beschriebenen Funktionen werden folgende Versionen der jeweiligen Bibliotheken verwendet:

Name der Bibliothek	Version
spaCy ¹	2.2.4
sentence-transformers ²	0.2.6
django_ocr_server ³	1.9
NeuralCoref ⁴	4.0
SciPy ⁵	1.4.1
SemBERT ⁶	Commit 16.07.2020

Tabelle 14: Bibliotheken und deren Version

¹ <https://spacy.io/>, zuletzt geprüft am 20.08.2020

² <https://github.com/UKPLab/sentence-transformers>, zuletzt geprüft am 20.08.2020

³ <https://django-ocr-server.readthedocs.io/>, zuletzt geprüft am 20.08.2020

⁴ <https://github.com/huggingface/neuralcoref>, zuletzt geprüft am 20.08.2020

⁵ <https://www.scipy.org/>, zuletzt geprüft am 20.08.2020

⁶ <https://github.com/cooelf/SemBERT>, zuletzt geprüft am 20.08.2020

6.2 Evaluation der prototypischen Implementierung

Für die Evaluation der einzelnen in Kapitel 5 implementierten Funktionen wird die Metrik Präzision⁷, Ausbeute⁸ und das F1-Maß⁹ verwendet. Nachfolgend werden die einzelnen Module des digitalen Assistenzsystems evaluiert.

6.2.1 Dokumentationsassistent

Die Kernaufgabe des Dokumentationsassistenten besteht darin, den Anwender bei der Dokumentation projektrelevanter Informationen kontextsensitiv zu unterstützen, um auch Nicht-Experten zu befähigen vollständige und standardisierte textuelle Beschreibungen zu erstellen (Anforderung 2).

Nachfolgend werden die in Kapitel 5.1 beschriebene Funktionen Semantische Rollenerkennung, Textklassifikation, Autovervollständigung und die Kontextsensitiven Hilfestellungen evaluiert. Für die Evaluation der semantischen Rollenerkennung werden die annotierten Daten (siehe Kapitel 6.1.1) verwendet. Nachfolgend werden die Ergebnisse aufgelistet (siehe Tabelle 15).

ID	Semantische Rolle	Präzision	Ausbeute	F1-Maß
1	Akteur	0,928	0,796	0,857
2	Datum	0,923	0,857	0,888
3	Eigenschaft	0,750	0,803	0,775
4	Eigenschaftswert	1,000	0,803	0,898
5	Einheit	0,926	0,862	0,893
6	Eltern-Objekt	0,958	0,772	0,855
7	Eltern-Subjekt	0,906	0,892	0,899
8	Events	0,974	0,864	0,916

⁷ <https://de.wikipedia.org/wiki/Präzision>, zuletzt geprüft am 20.08.2020

⁸ [https://www.chemie-schule.de/KnowHow/Ausbeute_\(Chemie\)](https://www.chemie-schule.de/KnowHow/Ausbeute_(Chemie)), zuletzt geprüft am 20.08.2020

⁹ https://en.wikipedia.org/wiki/F1_score, zuletzt geprüft am 20.08.2020

9	Funktionswort	1,000	0,807	0,893
10	Logischer Operator	1,000	0,981	0,991
11	Objekt	0,957	0,770	0,854
12	Person	0,933	0,757	0,836
13	Präzisierung	0,953	0,944	0,948
14	Prioritätswort	1,000	0,807	0,893
15	Prozesswort	0,981	0,828	0,899
16	Subjekt	0,906	0,887	0,896
17	Systemkomponente	0,969	0,852	0,907
18	Vergleichsoperator	0,921	0,904	0,913
	Durchschnitt	0,944	0,844	0,889

Tabelle 15: Evaluationsergebnisse der Semantischen Rollenerkennung

Wie hier zu sehen ist, schneidet die semantische Rollenerkennung mit einem durchschnittlichen F1-Maß von 88,9 % sehr gut ab, was die generelle Machbarkeit dieses Ansatzes beweist. Am schwächsten schneidet die Erkennung von Eigenschaften ab, was auf die geringe Trainingsdatenmenge zurückzuführen ist. Im Rahmen nachfolgender Arbeiten sollten daher zusätzliche Trainingsdaten erzeugt werden, um zum einen die Genauigkeit zu steigern und zum anderen die Übertragbarkeit auf weitere Domänen sicherzustellen.

Die Regeln für die Erkennung von Elternelementen kann zudem um das Präfix „von“ erweitert werden (z.B. „Die Tür von dem Auto“), welche zum Zeitpunkt der Evaluation nicht erkannt werden konnten. Durch die Ergänzung dieser Regel, würde das F1-Maß auf über 90% ansteigen und somit die Zuverlässigkeit der semantischen Rollenerkennung verbessern. Allgemein ist das trainierte NER-Modell noch sehr Fehleranfällig, was durch einen vergrößerten Trainingsdatensatz sicherlich weiter verbessert werden kann.

Darüber hinaus führen Sonderzeichen zu Fehlern, weshalb diese in einer verbesserten Version der semantischen Rollenerkennung vor der Verarbeitung gelöscht werden müssen. Zusätzlich zum F1-Maß wurde auch die Geschwindigkeit evaluiert. Hierzu wurden alle Sätze des Trainingsdatensatzes mit Hilfe des SRL untersucht und die dafür notwendige Zeit erfasst. Hierbei ergab sich eine

Die durchschnittliche Bearbeitungsdauer von ungefähr 10ms auf einem Lenovo P71 Laptop mit einem Intel Xeon E3-1505M Prozessor, 32 RAM und einer NVIDIA Quadro P4000 Grafikkarte. Dies zeigt die Performanz der auf spaCy basierenden Implementierung, welche neben der Wiederholgenauigkeit maßgeblich für die Praxistauglichkeit verantwortlich ist.

Für das Training und die Evaluation wurden die in Kapitel 6.1.1 beschriebenen Sätze verwendet und mit 20 Epochen trainiert. Hierbei konnte eine Präzision von 94,9 %, eine Ausbeute von 94,3 % und ein F1-Maß von 94,6 % erreicht werden. Um auch dieses Verfahren auf seine Praxistauglichkeit zu evaluieren, wurden auch hierfür alle zur Verfügung stehenden Sätze klassifiziert und die Bearbeitungsdauer ermittelt. Hierbei ergab sich eine durchschnittliche Bearbeitungsdauer von 1,9ms pro Satz, weshalb auch die Textklassifikation als praxistauglich eingestuft werden kann.

Sowohl die Referenzbeispiele als auch die Vollständigkeitsprüfung hängen von der semantischen Rollenerkennung als auch der Textklassifikation ab. Wie bereits gezeigt werden konnte, sind beide Verfahren mit einem F1-Maß von 88,9 % ausreichend zuverlässig. Dementsprechend zuverlässig können dem Anwender auch die Hilfestellungen angeboten werden, was entsprechende Mehrwerte für den Anwender generiert. Dank der hohen Geschwindigkeit der Verfahren, können die Hilfestellungen zudem mit sehr geringer Latenz (entsprechend der Verbindung zum Server) angeboten werden, was auch die Praxistauglichkeit des Verfahrens sicherstellt.

6.2.2 Integrationsassistent

Die Kernaufgabe des Dokumentationsassistenten besteht darin, den Anwender bei der Dokumentation projektrelevanter Informationen kontextsensitiv zu unterstützen, um auch Nicht-Experten zu befähigen vollständige und standardisierte textuelle Beschreibungen zu erstellen (Anforderung 2). Nachfolgend wird das in Kapitel 4.2 beschriebene Konzept und die in Kapitel 5.1 beschriebene Implementierung evaluiert.

Für die Evaluation der Koreferenz-Auflösung, basierend auf der Bibliothek NeuralCoref 4.0, wurde ein Datensatz der Tübinger Universität genutzt TüBa-D/Z [SFS 18], welcher mit über 39.484 manuell erzeugten Koreferenz-Ketten, aus über 3000 unterschiedlichen Dokumenten, annotiert wurde. Da dieser jedoch in seiner kostenfreien Version nur rein für Forschungszwecke genutzt werden darf, wurde zusätzlich ein öffentlich und auch kommerziell nutzbarer Datensatz „ParCorFull“¹⁰ [Lap 18] verwendet mit ca. 2500 annotierten Koreferenzen. Nachfolgend werden die Testergebnisse der beiden evaluierten Koreferenz-Modelle aufgelistet:

ID	Datensatz	Präzision	Ausbeute	F1-Maß
1	TüBa D/Z	77,57	62,30	69,10
2	ParCorFull	56,91	46,77	51,34

Tabelle 16: Evaluationsergebnisse der Mentions-Erkennung

Wie diese Ergebnisse aufzeigen, kann mit Hilfe des TüBa D/Z Datensatzes ein Modell erzeugt werden, welches dem englischen Referenzmodell der NeuralCoref 4.0 Implementierung mit einem F1-Maß von ca. 70% ebenbürtig ist. Der ParCorFull Datensatz schneidet jedoch erheblich schlechter ab. Jedoch reicht ein F1-Maß von 70% nicht aus, um mit der nötigen Wiederholgenauigkeit Koreferenzen zuverlässig genug zu identifizieren, weshalb zum jetzigen Zeitpunkt keine umfängliche Automatisierung der Kontextinformationsintegration möglich ist. Jedoch kann dieser Ansatz genutzt werden, um zumindest einen Teil zu automatisieren, welcher jedoch weiterhin durch Manuelle Kontrollen ergänzt werden muss.

6.2.3 Rechercheassistent

Die Aufgabe des Rechercheassistenten ist es, die Recherchearbeiten kontextsensitiv zu unterstützen. Hierzu werden anhand des aktuellen Kontextes des

¹⁰ <https://github.com/chardmeier/parcor-full>, zuletzt geprüft am 20.08.2020

Anwenders, möglicherweise relevante Informationen aus dem Internet gecrawlt und dem Anwender zur Integration in das aktuelle Projekt bereitgestellt.

Ziel der nachfolgenden Evaluation ist die generelle Funktionsprüfung des Konzepts hinsichtlich folgender Fragestellungen:

- Wie viele der identifizierten Inhalte enthalten relevante Informationen?
- Wie groß ist die Anzahl neuer Informationen?

Zur Ermittlung der Relevanz wird an dieser Stelle die TF-IDF Metrik¹¹ verwendet, welche die Nennungshäufigkeit der gefundenen Begriffe und die inverse Dokumentenhäufigkeit in denen die jeweiligen Begriffe gefunden wurden ins Verhältnis setzt:

Formel 5:

$$\text{TF}(X) = \frac{\text{Häufigkeit des Wortes } X \text{ in Dokument } A}{\text{Gesamtzahl aller Worte in Dokument } A}$$

$$\text{IDF}(X) = \frac{\text{Dokumentenanzahl in denen das Wort } X \text{ vorkommt}}{\text{Gesamtzahl aller Worte in Dokument } A}$$

Für die Ermittlung dieser Werte wurden drei Beispielsszenarien zu unterschiedlichen Produkten definiert und anschließend mit Hilfe der in Kapitel 5.3 beschriebenen Implementierung ausgeführt. Nachfolgend werden die Produktmerkmale der Beispielszenarien aufgeführt:

Bezeichnung	Auto	3D-Drucker	E-Commerce App
Eigenschaften	<ul style="list-style-type: none"> • Reichweite mind. 300km 	<ul style="list-style-type: none"> • Multicolor • Schichtdicke unter 0,05mm 	<ul style="list-style-type: none"> • 10000 Produkte • Mehrsprachig • Ratenzahlung
Komponenten	<ul style="list-style-type: none"> • Motor Antrieb elektrisch 	<ul style="list-style-type: none"> • Extruder 	

¹¹ https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html#sklearn.feature_extraction.text.TfidfTransformer, zuletzt geprüft am 20.08.2020

Informationen		• DIN SPEC 17071	• GDPR
----------------------	--	------------------	--------

Tabelle 17: Beispielszenarien für die Evaluation des Rechercheassistenten

Basierend auf diesen Beispielszenarien wurde ein Suchdurchlauf mit Hilfe des Rechercheassistenten durchgeführt. Die zehn mit Hilfe der TF IDF Methode an den höchsten bewerteten Begriffen, wurden nachfolgend hinsichtlich ihrer Relevanz und ihres Neuheitsgrades bewertet. Als relevant wurden Begriffe gewertet, welche grundsätzlich mit dem Zielprodukt zu tun haben. Der Neuheitsgrad errechnete sich anhand der Prozentzahl neuer Begriffe innerhalb der relevanten Begriffe (siehe Tabelle 18):

Ergebnis 2 Subseiten	Rex, LKW, E-PKW, Elektro-PKW, Auftanken, Economy, Akkutechnik, GO, Plug-In-Hybride, Serien-PKW	Anwendungsgebiet, EU.InhaltsverzeichnisFilament, Warenwert, Wunschort, Online-Kauf, Fialment, CARBON, PETG-Basis, Polycarbonate-ABS, Tabellenübersicht	Hinweis, Software, International, Netto, Bestandsprüfung, Tool, Euro, Postleitzahlengebiet, Erstbestellung, Werbematerialien
Relevanz	0,80	0,40	1,00
davon neu	1,00	1,00	1,00
Ergebnis 4 Subseiten	Sechszylinder-Reihenmotor, Zweizylinder-Boxermotor, OHV, 5er, Flugmotor, Erdgas-Variante, g-tron, ECE-R, Leasingzeit, Leasing-Deal	CARBON, InhaltsverzeichnisFilament, Warenwert, Wunschort, EU.Inhaltsverzeichnis-Filament, Online-Kauf, Filament, Metal, Tabellenübersicht, Polycarbonate-ABS	StaatenUnited, EMEAGlobal, A5Office, Asien-PazifikGlobal, 21VianetProjekt, Kundenaccount, UnionEuropean, K1Office, EssentialsOffice, BusinessOffice
Relevanz	1,00	0,40	1,00
Neuheitsgrad	1,00	1,00	1,00

Tabelle 18: Evaluationsergebnisse der Beispielszenarien

Wie diese Stichprobe aufzeigt, werden für jedes der Produkte relevante Informationen identifiziert, von denen in diesem Fall alle neu, also zuvor noch nicht in der Datenbank dokumentiert, sind. Letzteres ist jedoch vor allem auf die geringe Informationsdichte der Suchanfrage zurückzuführen. Je komplexer die bereits dokumentierten Projektinformationen sind, je höher ist die Wahrscheinlichkeit, dass der Rechercheassistent Informationen identifiziert, welche bereits in der semantischen Datenbank dokumentiert wurden.

Darüber hinaus bedarf es noch einer ausführlicheren Evaluation des Verfahrens, um den tatsächlichen Nutzen der so erhaltenen Ergebnisse bewerten zu können. Die Stichprobe zeigt jedoch das Potential einer automatischen Recherchefunktion und die grundsätzliche Machbarkeit des Ansatzes.

6.2.4 Verknüpfungsassistent

Der Verknüpfungsassistent hat zum Ziel, mögliche noch nicht miteinander verknüpfte Kontextelemente automatisch miteinander zu verknüpfen. Hierdurch soll zum einen der damit verbundene manuelle Aufwand reduziert und sichergestellt werden, dass alle zu verknüpfenden Informationen miteinander verknüpft sind. Darüber hinaus, sollen hierdurch mögliche Redundanzen oder Konflikte bereits während der Dokumentation neuer Text-Elemente vermieden werden.

Wie in Kapitel 5.4.2 beschrieben wurde, besteht der Verknüpfungsassistent im Wesentlichen aus zwei Methoden, einer regelbasierten Erkennung möglicher Konflikte und einem Sprachmodell, welches auf die sogenannte Natural Language Inference (NLI) Aufgabe trainiert wurde.

Für die Evaluation der semantischen Konflikterkennung, basierend auf dem NLI-Verfahren, wurden mehrere öffentlich zugängliche Datensätze hinsichtlich ihrer Anwendbarkeit im Bereich des Anforderungsmanagements evaluiert. Hierzu wurde ein englischer Evaluationsdatensatz erzeugt, welcher mögliche

Konflikte, Relationspartner oder neutrale Anforderungen und Aufgaben beschreibt. Für das Training wurde die Implementierung SemBERT¹² verwendet. Nachfolgend werden die Testergebnisse folgender Trainingsdatensätze aufgeführt:

Datensatz / Label	Präzision	Ausbeute	F1-Maß
SNLI ¹³			
Konfligierend	0,37	0,81	0,5
Neutral	n.a.	n.a.	n.a.
Beinhaltend	0,33	0,26	0,29
MNLI ¹⁴			
Konfligierend	0,42	0,21	0,28
Neutral	0,80	0,09	0,15
Beinhaltend	0,37	0,83	0,51
DNLI ¹⁵ [Wel 18]			
Konfligierend	0,17	0,19	0,18
Neutral	0,55	0,74	0,63
Beinhaltend	0,10	0,06	0,07

Tabelle 19: Evaluationsergebnisse der NLI Klassifikation

Wie die Ergebnisse aufzeigen, sind die öffentlich zugänglichen Datensätze für die Klassifikation von Anforderungssatzpaaren nur bedingt geeignet. Jedoch liefert jeder Datensatz für eine der drei Klassen im Schnitt ein F1-Maß von über 50% (hier grau hinterlegt), was die generelle Machbarkeit belegt. Bei näherer Betrachtung der Trainingsdatensätze fällt auf, dass die Trainingsbeispiele stark personenbezogene sind (siehe Tabelle 20).

¹² <https://github.com/cooelf/SemBERT>, zuletzt geprüft am 20.08.2020

¹³ <https://nlp.stanford.edu/projects/snli/>, zuletzt geprüft am 20.08.2020

¹⁴ <https://cims.nyu.edu/~sbowman/multinli/>, zuletzt geprüft am 20.08.2020

¹⁵ https://wellecks.github.io/dialogue_nli/, zuletzt geprüft am 20.08.2020

Satz 1	Satz 2	Klassifikation
Ein Mann inspiziert die Uniform einer Figur in einem ostasiatischen Land.	Der Mann schläft	Konfligierend
Ein älterer und ein jüngerer Mann lächeln.	Zwei Männer lächeln und lachen über die auf dem Boden spielenden Katzen.	Neutral
Ein Fußballspiel mit mehreren männlichen Spielern.	Einige Männer treiben einen Sport.	Beinhaltend

Tabelle 20: Beispielsätze der NLI-Klassifikationsaufgabe¹⁶

Wie diese Beispiele aufzeigen, ist das Subjekt jedes Satzes eine Person oder Personengruppe, hingegen das Subjekt von Anforderungssätzen meist eine Systemkomponente oder Funktion darstellen. Dies lässt den Schluss zu, dass eine Anpassung der Trainingsdaten zu besseren Ergebnissen führen könnte. Zum jetzigen Zeitpunkt jedoch kann mit Hilfe keiner der öffentlich verfügbaren Datensätze eine ausreichend genaue Klassifikation vorgenommen werden, um dies sinnvoll einsetzen zu können.

6.2.5 Wiederverwendungsassistent

Wie auch der Verknüpfungsassistent, nutzt der Wiederverwendungsassistent die NLI-Klassifikation zur Identifikation möglicher Relationspartner innerhalb bereits abgeschlossener oder parallellaufender Projekte. Siehe daher Kapitel 6.2.4.

6.2.6 Änderungsassistent

Zum Zeitpunkt der Evaluation wurde der Änderungsassistent noch nicht prototypisch realisiert, weshalb dieser im Rahmen dieser Arbeit nicht evaluiert werden konnte.

¹⁶ <https://nlp.stanford.edu/projects/snli/>, zuletzt geprüft am 20.08.2020

6.3 Abgleich der Anforderungen

Nachfolgend wird das Konzept hinsichtlich der Erfüllung der in Kapitel 3.8 abgeleiteten Anforderungen evaluiert:

Anforderung 1: Das digitale Assistenzsystem muss fähig sein, alle projekt-relevanten Kontextinformationen zu verwalten.

Das in Kapitel 4.1 beschriebene Datenmodell ermöglicht es dem Anwender alle projektrelevanten Informationen wie Anforderungen, Aufgaben, Systemkomponenten oder allgemeine Informationen innerhalb eines Werkzeugs zu managen und miteinander zu verknüpfen. Dank der Implementierung in Django ist auch die Verwaltung aller Kontextelemente durch den Admin-Modus¹⁷ sehr einfach und intuitiv realisiert. Folglich erfüllt das vorliegende Konzept diese Anforderung.

Anforderung 2: Das digitale Assistenzsystem sollte Nicht-AM-Experten befähigen, textuelle Beschreibungen standardisiert und vollständig formulieren können.

Dank des kontextsensitiven Ansatzes aller Module des Assistenzsystems, wird der Anwender in allen anfallenden Aufgaben von der Integration neuer Anforderungsdokumente (Kapitel 4.3), der Dokumentation (Kapitel 4.2) und Verknüpfung (Kapitel 4.5) neuer Kontextelemente, über die unterstützte Tragweitenabschätzung bei möglichen Änderungen (Kapitel 4.7) bis hin zur Wiederverwendung bereits dokumentierter Kontextelemente (Kapitel 4.6) in neuen oder parallel laufenden Projekten unterstützt.

Anforderung 3: Das digitale Assistenzsystem sollte fähig sein, Kontextinformationen automatisch aus Freitexten zu extrahieren.

¹⁷ https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Admin_site, zuletzt geprüft am 20.08.2020

Mit Hilfe des Integrationsassistenten (Kapitel 4.3), lassen sich Anforderungsdokumente unterschiedlicher Formate automatisch in maschinenlesbaren Text konvertieren (Kapitel 5.2.1), welcher anschließend automatisch strukturiert (Kapitel 5.2.2) und klassifiziert (Kapitel 5.2.3) wird. So wird es möglich Kontextinformationen automatisch aus Freitext zu extrahieren und semantische miteinander zu verknüpfen.

Anforderung 4: Das digitale Assistenzsystem sollte fähig sein, mögliche Relationen der automatisch extrahierten Kontextinformationen und der bereits existierenden Kontextelementen zu identifizieren.

Mit Hilfe der Koreferenz-Auflösung (Kapitel 4.3.2) lassen sich auch satzübergreifende Relationen von Informationen identifizieren und miteinander verknüpfen. In Kombination mit dem Verknüpfungsassistenten (Kapitel 4.5) bzw. dem Wiederverwendungsassistenten (Kapitel 4.6), kann die manuelle Verknüpfung von Kontextelementen bestmöglich unterstützt werden. Darüber hinaus, sind beide Assistenten dank der NLI-Funktion in der Lage neben möglichen Relationspartnern auch mögliche Konflikte oder Redundanzen automatisch zu identifizieren und reduzieren somit das Risiko vor inkonsistenten Anforderungsdokumentationen.

Anforderung 6: Das digitale Assistenzsystem sollte dem Anwender die Möglichkeit bieten, die Tragweite einer Änderung ganzheitlich bewerten zu können.

Im Rahmen dieser Arbeit wurde mit dem Änderungsassistenten ein Konzept entwickelt, welches den Anwender bei der Analyse der möglichen Tragweite einer Änderung unterstützt (Kapitel 4.7). Mit Hilfe des beschriebenen Änderungsprozesses, soll der Anwender hinsichtlich der möglichen Schwere der geplanten Änderung sensibilisiert werden, wodurch sich der Autor eine Reduktion des Änderungsrisikos erhofft. Nachfolgende Arbeiten werden zeigen, ob dieses Konzept in der Praxis funktioniert oder nicht.

Anforderung 7: Das digitale Assistenzsystem muss fähig sein, alle Projektbeteiligten über mögliche Änderungen zu informieren.

Durch die ganzheitliche Verknüpfung aller Kontextelemente (Kapitel 4.1), kann das System recht einfach so gestaltet werden, dass die betroffenen Personen (z.B. Bauteil- oder Baugruppenverantwortliche) über neue Kontextelementen informiert werden können. Vergleichbare Systeme, in denen Informationen manuell verknüpft werden müssen (z.B. IBM Doors, Jira etc.) können Änderungen nur dann sinnvoll kommunizieren, sofern der Autor eines neuen Kontextelements, die Verantwortlichen korrekt verknüpft. Da dies vor allem bei größeren Projekten jedoch nicht immer gewährleistet werden kann, bietet das vorliegende Konzept erhebliche Vorteile gegenüber den etablierten Softwarelösungen am Markt.

Anforderung 8: Das digitale Assistenzsystem sollte fähig sein, bereits dokumentierte Informationen dem Anwender kontextsensitiv zur Verfügung zu stellen.

Dank des Ticketbasierten Ansatzes und der beiden Assistenten zur Verknüpfung- (Kapitel 4.5) und Wiederverwendung (Kapitel 4.6) von Kontextelementen, können dem Anwender entsprechend seines aktuellen Kontexts möglicherweise relevante Informationen bereitgestellt werden, welche dieser prüfen und ggf. integrierten bzw. bearbeiten kann. Somit ist das vorliegende Konzept fähig, bereits dokumentierte Informationen kontextsensitiv zur Verfügung zu stellen.

Anforderung 9: Das digitale Assistenzsystem sollte fähig sein, den Anwender aktiv bei der Recherche möglicherweise relevanter Informationen zu unterstützen.

Mit Hilfe des Rechercheassistenten (Kapitel 4.4), bietet das vorliegende Konzept einen Lösungsansatz zur kontextsensitiven und automatisierten Recherche im Internet nach möglicherweise relevanten Informationen.

Abschließend lässt sich somit zusammenfassen, dass das vorliegende Konzept, alle in Kapitel 3.8 abgeleiteten Anforderungen erfüllt.

6.4 Fazit

Ziel der vorliegenden Arbeit ist die Entwicklung eines Assistenzsystems zur kontextsensitiven Unterstützung des anforderungsbasierten Wissensmanagements. Wie in Kapitel 1.1 beschrieben widmet sich die vorliegende Arbeit drei übergeordneten Forschungsfragen, deren Zielerfüllung nachfolgend diskutiert wird:

Forschungsfrage 1: Wie lassen sich die Methoden des Anforderungsmanagements besser in Unternehmensprozesse integrieren?

Das vorliegende Konzept adressiert diese Forschungsfrage durch eine ganzheitliche Dokumentation und Verknüpfung aller projektrelevanten Informationen (Kontextelemente) miteinander sowie die kontextsensitive Unterstützung des Recherche-, Wiederverwendungs- und Änderungsprozesses. Mit Hilfe dieses Ansatzes soll es auch Nicht-Experten ermöglicht werden alle für sie relevanten Anforderungen und Informationen erzeugen und verwalten zu können ohne hierfür komplexe Werkzeuge nutzen zu müssen, welche zusätzlich Schulungen erforderlich machen würden. Durch diesen Ansatz wird eine überwiegend unsichtbare Integration der Anforderungsmanagement-Methoden erzwungen, welche jedoch mehr Arbeit abnehmen als erzeugen soll. Somit liefert das vorliegende Konzept eine mögliche Antwort auf Forschungsfrage 1.

Forschungsfrage 2: Wie lassen sich alle projektrelevanten Informationen automatisiert miteinander semantisch verknüpfen?

Um im Falle einer Änderung alle möglichen Interdependenzen automatisiert erkennen und analysieren zu können, bedarf es einer ganzheitlichen Verknüpfung

fung aller projektrelevanten Informationen (Kontextelemente). Das vorliegende Konzept nutzt hierzu die Verknüpfungs- und Wiederverwendungsassistenten, um den Anwender bei der Verknüpfung aller Kontextelemente untereinander zu unterstützen. Dank des Dokumentationsassistenten, können mögliche Abhängigkeiten bereits während der Dokumentation identifiziert und verknüpft werden. Durch dieses Verfahren werden die mit der Verknüpfung verbundenen manuellen Aufwände auf ein Minimum reduziert und eine ganzheitliche Verknüpfung aller Kontextelemente sichergestellt. Darüber hinaus, ist das Konzept dank der Natural Language Interference (NLI) Methode in der Lage mögliche Redundanzen, potenzielle Konflikte oder semantische Abhängigkeiten zu identifizieren ohne umfangreiche und domänenspezifische Ontologien. Somit liefert die vorliegende Arbeit auch eine mögliche Antwort auf Forschungsfrage 2.

Forschungsfrage 3: Wie lässt sich die Wiederverwendung bereits digital dokumentierter Informationen verbessern?

Eine der größten Herausforderungen vieler Unternehmen ist die Wiederverwendung bereits dokumentierter Informationen – sozusagen dem digitalen Erfahrungswissen des Unternehmens. Mit Hilfe des Verknüpfungs- und Wiederverwendungsassistenten liefert vorliegende Konzept einen Lösungsansatz für Forschungsfrage 3, um bereits dokumentierte Kontextelemente dem Benutzer kontextsensitiv bereitzustellen. So können zum einen Redundanzen vermieden werden und zum anderen mögliche Zusatzinformationen aus bereits vergangenen oder parallellaufenden Projekten kontextsensitiv bereitgestellt werden. Dies ermöglicht es dem Anwender, neben der Konflikterkennung, mögliche für das aktuelle Projekt relevante Informationen wiederzuverwenden und somit wertvolles Wissen nachhaltig zu nutzen und die mit den ansonsten notwendigen Aufwänden zur Neuentwicklung zu reduzieren.

Wie die zuvor vorgestellten Ergebnisse aufzeigen, kann die generelle Machbarkeit des vorgestellten Konzepts gezeigt werden, auch wenn zum jetzigen Zeitpunkt nicht alle Funktionen ihr vollständiges Potential ausschöpfen. Folglich können auch die initial definierten Ziele (Forschungsfragen) dieser Arbeit,

als grundsätzlich beantwortet betrachtet werden. Nachfolgende Arbeiten – darunter die fortlaufenden Arbeiten des Projektes DAM4KMU – werden jedoch noch zeigen müssen, wie praxistauglich die in dieser Arbeit entwickelten Ansätze und Konzepte sind.

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung

Die vorliegende Arbeit hat zum Ziel, ein digitales Assistenzsystem zur kontextsensitiven Unterstützung des anforderungsbasierten Wissensmanagements vorrangig für kleine und mittelständische Unternehmen (KMU) zu entwickeln und mit Hilfe des Stands der Technik, prototypisch zu implementieren und evaluieren.

Die Arbeit stellt ein Konzept vor, bestehend aus sechs Modulen, welche den Anwender bei der Dokumentation, Integration, Verknüpfung, Änderung, Wiederverwendung als auch der Recherche textueller Beschreibungen kontextsensitiv unterstützt.

Der sogenannte **Dokumentationsassistent** ist in der Lage, Informationen aus textuellen Beschreibungen automatisch zu analysieren und semantisch miteinander zu verknüpfen. Dies reduziert zum einen den manuellen Aufwand, welcher mit dem Verknüpfen von Informationen verbunden ist und ermöglicht zum anderen, automatische Vollständigkeitsprüfungen durchzuführen, welche wiederum die Standardisierung textueller Beschreibungen fördern.

Mit Hilfe des **Integrationsassistenten** wird der Anwender bei der Integration von Informationen aus Dokumenten (z.B. Lasten oder Pflichtenhefte), in die bereits bestehende Datenbank unterstützt. Mit Hilfe eines webbasierten Dialogsystems, wird der Anwender bei der Prüfung und Freigabe der automatisch identifizierten Informationen unterstützt und spart somit wertvolle Zeit.

Um auch möglicherweise bisher nicht berücksichtigte Aspekte zu erkennen, unterstützt der **Rechercheassistent** durch eine automatische und kontextsensitive Suchfunktion basierend auf der Google-Such-API. Ziel der Suche, ist die Identifikation möglicher noch nicht berücksichtigter Systemkomponenten, Ei-

enschaften oder Normen. Ähnlich der Werbeeinblendungen im Internet, werden dem Anwender daher möglicherweise relevante Suchergebnisse kontextsensitiv bereitgestellt, was mögliche Innovationen oder möglicherweise noch nicht berücksichtigte Aspekte in das laufende Projekt einbringen soll.

Um bei der Erstellung oder Integration neuer Informationen, mögliche Redundanzen, Relationspartner oder potentielle Konflikte möglichst zeitnah identifizieren zu können, wurde das Konzept des **Verknüpfungsassistent** entwickelt. Mit Hilfe vortrainierter Sprachmodelle, können Ähnlichkeiten oder mögliche Konflikte auch ohne den Einsatz von Ontologien identifiziert werden. Dieses Verfahren reduziert den manuellen Aufwand, welcher bei der Erstellung einer domänenspezifischen Ontologie andernfalls notwendig wäre und reduziert somit die Kosten für die anwendenden Unternehmen.

Um bereits dokumentiertes Wissen aus vorangegangenen oder parallellaufenden Projekten auch für neue Projekte verfügbar zu machen, sieht das Konzept einen **Wiederverwendungsassistenten** vor. Basierend auf der Logik des Verknüpfungsassistenten, wird kontextsensitiv nach möglichen Redundanzen oder Relationspartnern innerhalb anderer Projekte gesucht. Die Ergebnisse werden dem Anwender anschließend inkl. aller Kinder und Kindeskinde zur Prüfung und Integration angeboten. Anschließend kann der Anwender wählen, ob er eine Kopie erstellen oder das gleiche Kontextelement referenzieren möchte. Dies steigert zum einen den projektübergreifenden Wissensaustausch als auch die nachhaltige Nutzung von Wissen, was potentiell Kosten und Zeit erspart.

Das letzte Modul, widmet sich der Unterstützung bei der Abschätzung möglicher Risiken bei der Durchführung von Änderungen. Der sogenannte **Änderungsassistent** versucht anhand der Relationspartner des betroffenen Kontextelements, anhand deren Relationspartner, des Status und der ggf. damit verbundener Kosten, eine mögliche Tragweite der Änderung abzuschätzen. Die Ergebnisse werden dem Anwender bereitgestellt, um die finale Bewertung des Risikos der geplanten Änderung bewerten zu können und ggf. Maßnahmen zu ergreifen. So sollen mögliche Fehler durch unbedachte Änderungen vermieden werden.

Im Rahmen der Evaluation der prototypischen Implementierung des Konzepts, wird aufgezeigt, dass die Thesen des vorliegenden Konzepts technisch realisiert und damit entsprechende Mehrwerte für die Anwender generieren werden können.

7.2 Ausblick

Im Rahmen dieser Arbeit wurde aufgezeigt, dass die wesentlichen Funktionen des in Kapitel 4 beschriebenen Konzepts, mit Hilfe aktueller computerlinguistischer Verfahren realisiert werden können. Wie die Evaluation aufzeigt, besteht jedoch noch Handlungsbedarf, um die entwickelten Konzepte praxistauglich zu machen. Das Projekt DAM4KMU wird zeigen, welche Teile des Konzepts ihren Weg in die Praxis finden werden. Nachfolgend werden mögliche Potenziale welche im Rahme der Evaluation aufgedeckt wurden zusammengefasst:

Semantische Rollenerkennung: Die semantische Rollenerkennung ist die wichtigste Funktion dieses Konzepts. Mit einem F1-Maß von durchschnittlich 88,9%, trotz vergleichsweise kleiner Trainingsdatenmengen für die NER-Erkennung, stellt dies ein sehr gutes Ergebnis dar. Auch die rein regelbasiert erkannten semantischen Rollen werden mit einem F1-Maß von über 85% erkannt. Jedoch sind 15% Fehlerquote noch ein zu hoher Faktor, um ein derartiges System vollständig automatisiert zur Integration und Verknüpfung von Anforderungsdokumenten zu verwenden. Der Autor empfiehlt daher die Weiterentwicklung der Trainingsdatensätze zum Training des NER-Modells, sowie eine kontinuierliche Verbesserung der regelbasierten Rollenerkennung.

Autovervollständigung: Der in dieser Arbeit zum Einsatz kommende Ansatz der Autovervollständigung nutzt noch nicht alle Potentiale des aktuellen Stands der Technik. Wie Demonstrationen des Sprachmodells GPT2¹ eindrucksvoll veranschaulichen, sind vortrainierte Sprachmodelle in der Lage

¹ <https://anotherdatum.com/gpt-2.html>, zuletzt geprüft am 20.08.2020

auch komplexe Sätze automatisch zu erzeugen und somit auch domänenübergreifend adäquate Wort- und Satzvorschläge zu generieren.

Natural-Language-Interference: Wie die in Kapitel 5.4 evaluierten Trainingsdatensätze aufzeigen, sind öffentlich zugängliche Datensätze noch nicht zur automatischen Erkennung semantischer Konflikte von Anforderungs-Satzpaaren geeignet. Wie die Analyse der Datensätze aufzeigt, sind die meisten Datensätze vor allem mit Personen oder Personengruppen als Subjekt formuliert worden, jedoch nur selten mit Gegenständen, Funktionen oder Geräten als Subjekt. Da die für die sinnvolle Anwendbarkeit notwendigen Datensätze jedoch sehr groß sind, konnte im Rahmen dieser Arbeit kein an den Anwendungsfall des Anforderungsmanagements übertragener Datensatz erzeugt werden. Nachfolgende Arbeiten könnten sich daher mit der Anpassung dieser Datensätze an die Domäne des Anforderungsmanagements befassen und die aufgezeigten Potenziale voraussichtlich drastisch verbessern.

Rechercheassistent: Das im Rahmen dieser Arbeit vorgestellte Konzept zur automatischen Rechercheunterstützung, stützt sich auf die automatische Informationssuche basierend auf einzelnen Stichworten. Durch dieses Verfahren werden zwar viele neue Informationen gefunden, welche jedoch nicht alle relevant oder innovationshebend sind. Nachfolgende Arbeiten könnten versuchen, den kompletten Kontext der projektrelevanten Dokumentation zu verwenden, um auch kontextbezogene Informationen besser extrahieren und dem Anwender zur Verfügung zu stellen.

7.3 Summary

The aim of this thesis is to develop a digital assistance system for the context-sensitive support of a requirements-based knowledge management primarily for small and medium-sized enterprises (SME) and to implement and evaluate it prototypically with the help of state-of-the-art technology.

The thesis presents a concept consisting of six modules, which supports the user in documentation, integration, linking, modification, reuse and research of textual descriptions in a context sensitive way.

The so-called **documentation assistant** is able to automatically analyze information from textual descriptions and to link them semantically. On the one hand, this reduces the manual effort involved in linking information and, on the other hand, enables automatic completeness checks to be carried out, which in turn promotes the standardization of textual descriptions.

With the help of the **integration assistant**, the user is supported in the integration of information from documents (e.g. requirements or specifications) into the existing database. With the help of a web-based dialogue system, the user is supported in checking and releasing the automatically identified information, thus saving valuable time.

In order to also recognize aspects that may have been ignored up to now, the **research assistant** supports the user with an automatic and context-sensitive search function based on the Google search API. The aim of the search is to identify possible system components, properties or standards that have not yet been considered. Similar to advertising on the Internet, the user is therefore provided with context-sensitive search results, which are intended to introduce possible innovations or aspects not yet considered into the current project.

In order to be able to identify possible redundancies, relation partners or potential conflicts as quickly as possible when creating or integrating new information, the concept of the **link assistant** was developed. With the help of pre-trained language models, similarities or potential conflicts can be identified even without the use of ontologies. This procedure reduces the manual effort that would otherwise be necessary when creating a domain-specific ontology and thus reduces the costs for the applying companies.

In order to make already documented knowledge from previous or parallel projects available for new projects, the concept includes a **reuse assistant**. Based on the logic of the link assistant, a context-sensitive search for possible redundancies or relation partners within other projects is carried out. The results are then offered to the user including all children and grandchildren for checking and integration. The user can then choose whether to create a copy or reference the same context element. This increases both the cross-project

exchange of knowledge and the sustainable use of knowledge, which potentially saves time and money.

The last module is dedicated to supporting the assessment of possible risks during the implementation of changes. The so-called **change assistant** attempts to estimate the possible consequences of the change by means of the relation partners of the affected context element, their relation partners, the status and the costs that may be involved. The results are made available to the user so that he can evaluate the final assessment of the risk of the planned change and take measures if necessary. In this way, possible errors caused by careless changes are to be avoided.

In the course of the evaluation of the prototypical implementation of the concept, it is shown that the theses of the present concept can be technically realized and thus generate corresponding added value for the users.

8 Literaturverzeichnis

- [Ada 13] Adam, S.; Wüsch, C.; Koch Matthias: Ergebnisbericht RE-Kompass 2013. https://www.iese.fraunhofer.de/content/dam/iese/de/dokumente/oeffentliche_studien/Ergebnisbericht_RE-Kompass_2013.pdf, aufgerufen am 20.08.2020.
- [Ada 15] Adam, S.; Wüsch, S.; Seyff, N.: Ergebnisbericht 2014/15 RE-Kompass. https://www.hood-group.com/fileadmin/user_upload/requirements/Downloads/Paper/DE/HOOD_RE-Kompass_2014_15_v1.6.pdf, aufgerufen am 20.08.2020.
- [Alb 11] Albers, A.; Braun, A.: A generalised framework to compass and to support complex product engineering processes. In *International Journal of Product Development*, 2011, 15; S. 6–25.
- [Alb 12] Albers, A.; Braun, A.; Pinner, T.: Integrated Modelling of Information to Support Product Engineering Processes. In *DS 72: Modelling and Management of Engineering Processes-Concepts, Tools and Case Studies*, Cambridge, United Kingdom, 2012.
- [Aro 15] Arora, C. et al.: Change impact analysis for Natural Language requirements: An NLP approach. In (Zowghi, D. Hrsg.): *2015 IEEE 23rd International Requirements Engineering Conference (RE)*. 24 - 28 Aug. 2015, Ottawa, ON, Canada. IEEE, Piscataway, NJ, 2015; S. 6–15.
- [Aro 17] Arora, C. et al.: Automated Extraction and Clustering of Requirements Glossary Terms. In *IEEE Transactions on Software Engineering*, 2017, 43; S. 918–945.
- [Aud 17] Audeh, B. et al.: Vigi4Med Scraper: A Framework for Web Forum Structured Data Extraction and Semantic Representation. In *PloS one*, 2017, 12; e0169658.

- [Bia 05] Biagioli, C. et al.: Automatic semantics extraction in law documents. In (Sartor, G. Hrsg.): Proceedings of the 10th international conference on Artificial intelligence and law. ACM, New York, NY, 2005; S. 133.
- [Bir 18] Bird, S.; Klein, E.; Loper, E.: Natural Language Processing with Python. 9. Building Feature Based Grammars. <https://www.nltk.org/book/ch09.html>, aufgerufen am 20.08.2020.
- [Bow] Bowman, S. R. et al.: A large annotated corpus for learning natural language inference. In (Márquez, L.; Callison-Burch, C.; Su, J. Hrsg.): Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Stroudsburg, PA, USA; S. 632–642.
- [Bre 06] Breaux, T. D.; Vail, M. W.; Anton, A. I.: Towards Regulatory Compliance: Extracting Rights and Obligations to Align Requirements with Regulations. In (Glinz, M. Hrsg.): 14th IEEE International Requirements Engineering Conference, 2006, RE 2006. 11 - 15 Sept. 2006, Minneapolis/St. Paul, Minnesota, USA. IEEE Computer Society, Los Alamitos, Calif., 2006; S. 49–58.
- [Bul 09] Bullinger, H.-J. et al.: Handbuch Unternehmensorganisation. Strategien, Planung, Umsetzung. Springer-Verlag, Berlin, Heidelberg, 2009.
- [Car 17] Carrillo de Gea, J. M. et al.: Automated support for reuse-based requirements engineering in global software engineering. In Journal of Software: Evolution and Process, 2017, 29; e1873.
- [Cla 20] Clarc, K.; Luong, T.: More Efficient NLP Model Pre-training with ELECTRA. <https://ai.googleblog.com/2020/03/more-efficient-nlp-model-pre-training.html>, aufgerufen am 20.08.2020.

- [Coh 19] Cohen, O.: A Comparison Between Spacy NER & Stanford NER Using All US City Names. <https://towardsdatascience.com/a-comparison-between-spacy-ner-stanford-ner-using-all-us-city-names-c4b6a547290>, aufgerufen am 20.08.2020.
- [Dev 18] Devlin, J. et al.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. <https://www.aclweb.org/anthology/N19-1423.pdf>, aufgerufen am 20.08.2020.
- [Dic 17] Dick, J.; Hull, E.; Jackson, K.: Requirements Engineering. Springer International Publishing, Cham, 2017.
- [Do 19] Do, Q. A.; Chekuri, S. R.; Bhowmik, T.: Automated Support to Capture Creative Requirements via Requirements Reuse. In (Peng Hrsg.): Reuse in the Big Data Era. Springer International Publishing, Cham, 2019; S. 47–63.
- [DON 15] DONG, M. et al.: The Reuse Method of Design Knowledge Based on Knowledge Component. In (Qi, E.; Shen, J.; Dou, R. Hrsg.): Proceedings of the 21st International Conference on Industrial Engineering and Engineering Management 2014. Atlantis Press, Paris, s.l., 2015; S. 57–60.
- [Dra 18] Dragoni, M. et al.: Combining Natural Language Processing Approaches for Rule Extraction from Legal Documents. In (Pagallo, U. et al. Hrsg.): AI approaches to the complexity of legal systems. AICOL international workshops 2015-2017: AICOL-VI@JURIX 2015, AICOL-VII@EKAW 2016, AICOL-VIII@JURIX 2016, AICOL-IX@ICAIL 2017 and AICOL-X@JURIX 2017 revised selected papers. Springer, Cham, 2018; S. 287–300.
- [Ebe 14] Ebert, C.: Systematisches Requirements Engineering. Anforderungen ermitteln, spezifizieren, analysieren und verwalten. dpunkt.verl., Heidelberg, 2014.
- [Ehr 09] Ehrlenspiel, K.: Integrierte Produktentwicklung. Denkabläufe, Methodeneinsatz, Zusammenarbeit. Hanser, München, 2009.

- [Eig 13] Eigner, M.; Stelzer, R.: Product Lifecycle Management. Ein Leitfaden für Product Development und Life Cycle Management. Springer, Dordrecht, 2013.
- [EIM 12] ElMaraghy, H. A.: Enabling Manufacturing Competitiveness and Economic Sustainability. Proceedings of the 4th International Conference on Changeable, Agile, Reconfigurable and Virtual production (CARV2011), Montreal, Canada, 2-5 October 2011. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [Eza 18] Ezami, S.: Extracting Non-Functional Requirements from Unstructured Text. Master thesis, University of Waterloo, Computer Science, Waterloo, Ontario, Canada, 2018.
- [Fel 13] Feldhusen, J.; Grote, K.-H. Hrsg.: Pahl/Beitz Konstruktionslehre. Methoden und Anwendung erfolgreicher Produktentwicklung. Springer Vieweg, Berlin, Heidelberg, 2013.
- [Fel 99] Fellbaum, C.: WordNet. An electronic lexical database. MIT Press, Cambridge, Mass, 1999.
- [Fra 15] Fraga, A. et al.: Ontology-Assisted Systems Engineering Process with Focus in the Requirements Engineering Process. In (Boulanger, F. et al. Hrsg.): Complex Systems Design & Management. Proceedings of the Fifth International Conference on Complex Systems Design & Management CSD&M 2014. Springer International Publishing, Cham, s.l., 2015; S. 149–161.
- [Fri 09] Friedrich, J.: Das V-Modell XT. Für Projektleiter und QS-Verantwortliche kompakt und übersichtlich. Springer, Berlin, 2009.
- [Fri 19] Fritz, S.; Weber Felix; Ovtcharova, J.: A Guideline for the Requirements Engineering Process of SMEs Regarding to the Development of CPS. <https://ieeexplore.ieee.org/document/8710732>, aufgerufen am 20.08.2020.

- [Gel 10] Gelhausen, T.: Modellextraktion aus natürlichen Sprachen. Eine Methode zur systematischen Erstellung von Domänenmodellen. Zugl.: Karlsruhe, KIT, Diss., 2010. KIT Scientific Publishing, Karlsruhe, 2010.
- [Ger 19] Gergely D., N.: BLEU-BERT-y: Comparing sentence scores. <https://towardsdatascience.com/bleu-bert-y-comparing-sentence-scores-307e0975994d>, aufgerufen am 20.08.2020.
- [Gli 06] Glinz, M.: Requirements Engineering I. In Nicht funktionale Anforderungen. Universität Zürich, Institut für Informatik, Zürich, 2006.
- [Gra 14] Grande, M.: 100 Minuten für Anforderungsmanagement. Kompaktes Wissen nicht nur für Projektleiter und Entwickler. Springer Vieweg, Wiesbaden, 2014.
- [Gro 13] Großmann, L.: Kooperatives Anforderungsmanagement in KMU. https://www.gpm-ipma.de/fileadmin/user_upload/GPM/Know-How/Ergebnisse_Anforderungsmanagement.pdf, aufgerufen am 20.08.2020.
- [Hoc 97] Hochreiter, S.; Schmidhuber, J.: Long short-term memory. In: Neural computation, 1997, 9; S. 1735–1780.
- [Hof 14] Hoffmann, C.: Kurzstudie CPS. Industrie 4.0 und CPS – Bedarfe und Lösungen aus Sicht des Mittelstands, 2014, https://www.baymevbm.de/Redaktion/Frei-zugaengliche-Medien/Abteilungen-GS/Regionen-und-Marketing/2016/Downloads/Kurzstudie_CPS_20141007.pdf, aufgerufen am 15.03.2021,
- [Höh 08] Höhn, R.; Höppner, S.; Rausch, A.: Das V-Modell XT. Anwendungen, Werkzeuge, Standards. Springer, Berlin, Heidelberg, 2008.

- [Hru 17] Hruschka, P.: Requirements Engineering. In (Tiemeyer, E. Hrsg.): Handbuch IT-Management. Konzepte, Methoden, Lösungen und Arbeitshilfen für die Praxis. Hanser, München, 2017; S. 421–452.
- [IRE 17] IREB: Foundation Level, 2017, https://www.ireb.org/content/downloads/2-syllabus-foundation-level/ireb_cppe_syllabus_fl_de_v221.pdf, aufgerufen am 20.08.2020.
- [Joc 11] Jochem, R.; Landgraf, K. Hrsg.: Anforderungsmanagement in der Produktentwicklung. Komplexität reduzieren, Prozesse optimieren, Qualität sichern. Symposium Publishing GmbH, Düsseldorf, 2011.
- [Jos 19] Joshi, M. et al.: BERT for Coreference Resolution: Baselines and Analysis, 2019, <https://www.aclweb.org/anthology/D19-1588/>, aufgerufen am 15.03.2021
- [Kif 08] Kifer, M.: Rule Interchange Format: The Framework. In (Calvanese, D.; Lausen, G. Hrsg.): Web reasoning and rule systems. Second international conference, RR 2008, Karlsruhe, Germany, October 31 - November 1, 2008 ; proceedings. Springer, Berlin, 2008; S. 1–11.
- [Kiy 08] Kiyavitskaya, N. et al.: Automating the Extraction of Rights and Obligations for Regulatory Compliance. In (Li, Q. et al. Hrsg.): Conceptual modeling - ER 2008. 27th International Conference on Conceptual Modeling, Barcelona, Spain, October 20 - 24, 2008 ; proceedings. Springer, Berlin, Heidelberg, 2008; S. 154–168.
- [Kor 09] Korner, S. J.; Brumm, T.: RESI - A Natural Language Specification Improver: IEEE International Conference on Semantic Computing, 2009. ICSC '09 ; Berkeley, California, USA, 14 - 16 Sept. 2009 ; including workshop papers. IEEE, Piscataway, NJ, 2009; S. 1–8.

- [Kör 14] Körner, S. J.: RECAA - Werkzeugunterstützung in der Anforderungserhebung. Zugl.: Karlsruhe, KIT, Diss., 2014. Technische Informationsbibliothek u. Universitätsbibliothek; KIT Scientific Publ, Hannover, Karlsruhe, 2014.
- [Kri 17] Krisch, J.: Sprachliche Kontrolle von Anforderungsdokumenten, 2017, https://hildok.bsz-bw.de/files/737/Krisch_Sprachliche+Kontrolle.pdf, aufgerufen am 20.08.2020.
- [Kur 20] Kurt, N.: Stochastik für Informatiker. Eine Einführung in einheitlich strukturierten Lerneinheiten. Springer Vieweg, 2020.
- [Lan 10] Lange, D.; Böhm, C.; Naumann, F.: Extracting structured information from Wikipedia articles to populate infoboxes. In (Huang, J. et al. Hrsg.): Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM '10. ACM Press, New York, New York, USA, 2010; S. 1661.
- [Lap 18] Lapshinova-Koltunski, E.; Hardmeier, C.; Krielke, P.: a Parallel Corpus Annotated with Full Coreference. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Japan, 2018; S. 423–428.
- [Lau 16] Lauenroth, K.; Schreiber, Fabian; Schreiber, Felix: Maschinen- und Anlagenbau im digitalen Zeitalter. Requirements Engineering als systematische Gestaltungskompetenz für die Fertigungsindustrie Industrie 4.0. Beuth Verlag GmbH, 2016.
- [Lid 01] Liddy E.D.: Natural Language Processing. In Encyclopedia of Library and Information Science, 2001, <https://surface.syr.edu/istpub/63/>, aufgerufen am 20.08.2020.
- [Loh 13] Lohmeyer, Q.: Menschzentrierte Modellierung von Produktentstehungssystemen unter besonderer Berücksichtigung der Synthese und Analyse dynamischer Zielsysteme, 2013 <https://publikationen.bibliothek.kit.edu/1000035102/2616039>, aufgerufen am 15.03.2021.

- [Luc 14] Lucke, D. et al.: Strukturstudie „Industrie 4.0 für Baden-Württemberg“. In Baden-Württemberg auf dem Weg zu Industrie, 2014, https://wm.baden-wuerttemberg.de/fileadmin/redaktion/m-wm/intern/Dateien_Downloads/Innovation/IPA_Strukturstudie_Industrie_4.0_BW.pdf, aufgerufen am 15.03.2021.
- [Man 05] Manning, C. D.; Schütze, H.: Foundations of statistical natural language processing. MIT Press, Cambridge, Mass., 2005.
- [Mas 12] Maske, P.: Mobile Applikationen 2. Interdisziplinäre Entwicklung am Beispiel des Mobile Learning. Zugl.: Hannover, Univ., Diss., 2011. Gabler Verlag, Wiesbaden, 2012.
- [Mey 14] Meyer-Schwickerath, B.: Vorausschau im Produktentstehungsprozess-Das integrierte Produktentstehungs-Modell (iPeM) als Bezugsrahmen für Vorausschau am Beispiel von Szenariotechnik und strategischer Frühaufklärung, 2014, <https://publikationen.bibliothek.kit.edu/1000044947/3358800>, aufgerufen am 15.03.2021.
- [Mik 13] Mikolov, T. et al.: Efficient Estimation of Word Representations in Vector Space, 2013, <https://arxiv.org/abs/1301.3781>, aufgerufen am 15.03.2021.
- [moX 18] moXimo OHG: Actron - Satzschablone für Aufgaben, 2018, <https://www.moximo.de/V6/?controller=pages&action=methoden&q=Actron%20-%20Satzschablone%20f%C3%BCr%20Aufgaben>, aufgerufen am 20.08.2020.
- [Mu 09] Mu, Y.; Wang, Y.; Guo, J.: Extracting Software Functional Requirements from Free Text Documents: International Conference on Information and Multimedia Technology, 2009. I-CIMT '09 ; 16-18 Dec. 2009, Jeju Island, South Korea ; proceedings ; 2009 International Conference on Mobile, Wireless and Optical Communication (MWOC 2009) and 2009 International Conference on Semi-Conductor Technology (ICSCT 2009). IEEE, Piscataway, NJ, 2009; S. 194–198.

- [Nie 13] Niebisch, T.: Anforderungsmanagement in sieben Tagen. Der Weg vom Wunsch zur Konzeption. Springer Gabler, Berlin, 2013.
- [OMG 15] OMG UML 2.5:01.03.2015, Unified Modeling Language, <https://www.omg.org/spec/UML/About-UML/>, aufgerufen am 15.03.2021.
- [OMG 16] OMG ReqIF 1.2:01.06.2016, Requirements Interchange Format, <https://www.omg.org/reqif/>, aufgerufen am 15.03.2021.
- [OMG 18] OMG SysML 1.6:01.10.2018, Systems Modeling Language, <https://www.omg.org/spec/SysML/>, aufgerufen am 15.03.2021.
- [Pah 07] Pahl, G. et al.: Konstruktionslehre. Grundlagen erfolgreicher Produktentwicklung; Methoden und Anwendung. Springer, Berlin, Heidelberg, 2007.
- [Pen 19] Peng Hrsg.: Reuse in the Big Data Era. Springer International Publishing, Cham, 2019.
- [Poh 09] Pohl, K.; Rupp, C.: Basiswissen Requirements Engineering. Aus- und Weiterbildung zum "Certified Professional for Requirements Engineering"; Foundation Level nach IREB-Standard. dpunkt-Verl., Heidelberg, 2009.
- [Poh 15] Pohl, K.; Rupp, C.: Basiswissen Requirements Engineering. Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering: foundation level nach IREB-Standard. dpunkt, Heidelberg, 2015.
- [Qi 16] Qi, J. et al.: Mining customer requirements from online reviews: A product improvement perspective. In: Information & Management, 2016, 53; S. 951–963.
- [Ray 16] Rayadurgam, S.; Tkachuk, O. Hrsg.: NASA Formal Methods. 8th International Symposium, NFM 2016, Minneapolis, MN, USA, June 7-9, 2016, Proceedings. Springer International Publishing, Cham, s.l., 2016.

- [Reb 16] Rebele, T. et al.: YAGO: A Multilingual Knowledge Base from Wikipedia, Wordnet, and Geonames. In (Groth, P. Hrsg.): The semantic web - ISWC 2016. 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016 proceedings. Springer, Cham, 2016; S. 177–185.
- [Rei 19] Reimers, N.; Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, 2019, <https://arxiv.org/pdf/1908.10084>, aufgerufen am 15.03.2021.
- [Reu 14] Reuter, M.: Technischer und wirtschaftlicher Vergleich von Herstellungsverfahren bei der Entwicklung von Kunststoff-hohlkörpern in Automobilanwendungen, 2014, https://duepublico2.uni-due.de/receive/duepublico_mods_00032835, aufgerufen am 15.03.2021.
- [REU 18] REUSE Company: Managing the quality ecosystem: DOORS, Rhapsody, Simulink and Modelica. <https://www.reusecompany.com/webinars/431-managing-the-quality-ecosystem-doors-rhapsody-simulink-and-modelica>, aufgerufen am 20.08.2020.
- [Rey 11] Rey, G. D.; Wender, K. F.: Neuronale Netze. Eine Einführung in die Grundlagen, Anwendungen und Datenauswertung. Huber, Bern, 2011.
- [San 15] Sandkuhl, K.: Knowledge Reuse: Survey of Existing Techniques and Classification Approach. In (Zimányi, E.; Kutsche, R.-D. Hrsg.): Business Intelligence. 4th European Summer School, eBISS 2014, Berlin, Germany, July 6-11, 2014 tutorial lectures. Springer, Cham, 2015; S. 126–148.
- [Sch 09] Schedl, S.: Integration von Anforderungsmanagement in den mechatronischen Entwicklungsprozess. Herbert Utz Verlag, München, 2009.
- [Sch 10] Schubert, V.; Wicaksono, H.; Rogalski, S.: Knowledge-based Product Configuration through Product Life Cycle Oriented

- Feedback-Driven Requirements Engineering. https://www.researchgate.net/publication/285583070_Knowledge-based_Product_Configuration_through_Product_Life_Cycle_Oriented_Feedback-Driven_Requirements_Engineering, aufgerufen am 04.10.2020.
- [Sch 14] Schoeneberg, K.-P. Hrsg.: Komplexitätsmanagement in Unternehmen. Herausforderungen im Umgang mit Dynamik, Unsicherheit und Komplexität meistern. Springer Gabler, Wiesbaden, 2014.
- [Sch 15] Schubert, P.: Entscheidungsunterstützte Methodik zur Produktkonzeptauswahl. Grundlagen, Systematik und exemplarische Anwendung. Dissertation, Karlsruhe: KIT Scientific Publishing, 2015.
- [Sey 18] Seyler, D. et al.: A Study of the Importance of External Knowledge in the Named Entity Recognition Task. In (Gurevych, I.; Miyao, Y. Hrsg.): Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Association for Computational Linguistics, Stroudsburg, PA, USA, 2018; S. 241–246.
- [SFS 18] SFS Tübingen: TüBa-D/Z Release 11.0. <https://uni-tuebingen.de/fakultaeten/philosophische-fakultaet/fachbereiche/neu-philologie/seminar-fuer-sprachwissenschaft/arbeitsbereiche/allg-sprachwissenschaft-computerlinguistik/ressourcen/corpora/tueba-dz/>, aufgerufen am 20.08.2020.
- [Soo 01] Soon, W. M.; Ng, H. T.; Lim, D. C. Y.: A Machine Learning Approach to Coreference Resolution of Noun Phrases. In Computational Linguistics, 2001, 27; S. 521–544.
- [SOP 13] SOPHIST GmbH: Schablonen für alle Fälle. Die SOPHISTen. https://www.sophist.de/fileadmin/user_upload/Bilder_zu_Seiten/Publikationen/Wissen_for_free/MASTeR_Broschuere_3-Auflage_interaktiv.pdf, aufgerufen am 20.08.2020.

- [Stö 18] Stöhler, C.; Förster, C.; Brehm, L.: Projektmanagement lehren. Studentische Projekte erfolgreich konzipieren und durchführen. Springer Fachmedien Wiesbaden, Wiesbaden, 2018.
- [Sup 18] SuperDataScience: Convolutional Neural Networks (CNN): Summary. <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-summary>, aufgerufen am 20.08.2020.
- [The 14] The Standish Group: Chaos Report. 21ST Anniversary Edition. https://www.standishgroup.com/sample_research_files/CHAOSReport2014.pdf, aufgerufen am 20.08.2020.
- [The 15] The Standish Group: Chaos Report 2015. https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf, aufgerufen am 20.08.2020.
- [The 94] The Standish Group: The Chaos Report (1994). https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf, aufgerufen am 20.08.2020.
- [Tug 16] Tuggener, D.: Incremental Coreference Resolution for German. Doktorarbeit, Zürich, Universität Zürich, 2016.
- [VDI 01] VDI Richtlinie 2519: Vorgehensweise bei der Erstellung von Lasten/Pflichtenheften. In VDI-Verlag, Düsseldorf, 2001.
- [Wan 12] Wang, T. et al.: End-to-End Text Recognition with Convolutional Neural Networks. <https://ai.stanford.edu/~ang/papers/ICPR12-TextRecognitionConvNeuralNets.pdf>, aufgerufen am 20.08.2020.
- [Wel 18] Welleck, S. et al.: Dialogue Natural Language Inference, 2018, <https://arxiv.org/pdf/1811.00671>, aufgerufen am 15.03.2021.
- [Wic 12] Wicaksono, H.; Rogalski, S.; Ovtcharova, J.: Knowledge Management Approach to improve Energy Efficiency in Small Medium Enterprises. 10th International Conference on Manufacturing Research (ICMR 2012), 760-765, Birmingham, UK, 2012.

- [Wol 17] Wolf, T.: State-of-the-art neural coreference resolution for chatbots. <https://medium.com/huggingface/state-of-the-art-neural-coreference-resolution-for-chatbots-3302365dcf30>, aufgerufen am 20.08.2020.
- [Yan 15] Yan, Z. et al.: EXPRS: An extended pagerank method for product feature extraction from online consumer reviews. In: *Information & Management*, 2015, 52; S. 850–858.
- [Zha 10] Zhang, C.; Zhang, J.: InForCE: Forum data crawling with information extraction: 4th International Universal Communication Symposium (IUCS), 2010. 18 - 19 Oct. 2010, Beijing, China ; proceedings. IEEE, Piscataway, NJ, 2010; S. 367–373.
- [Zin 13] Zingel, J. C.: Basis definition of a common language of product engineering in the context of modeling of technical systems and a modeling technique for the systems of objectives and objects of technical systems on the basis of the ZHO-principle. In: IPEK-Institut für Produktentwicklung am Karlsruher Institut für Technologie, Forschungsbericht, 2013.

9 Anhang

9.1 Satzschablonen

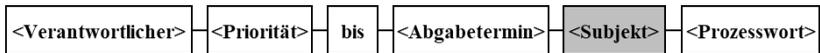
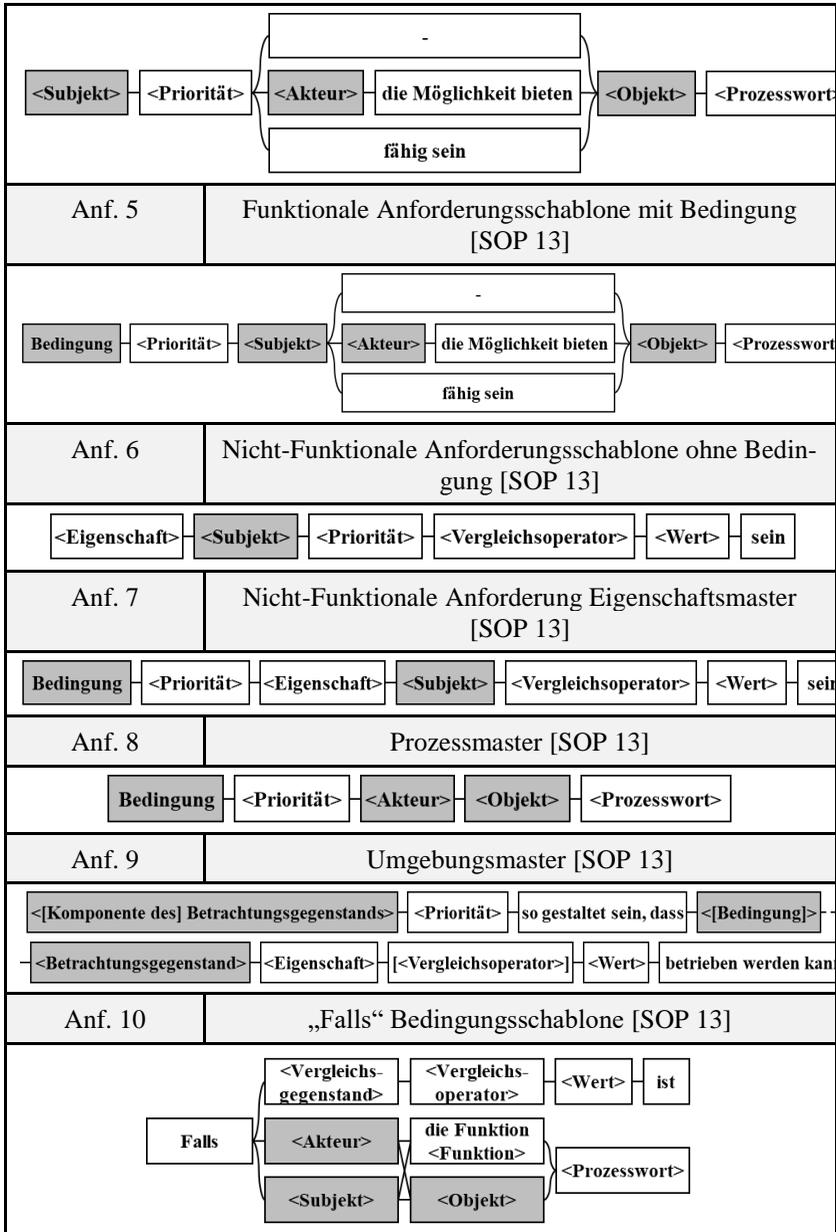


Abbildung 50: Aufgaben-Satzschablonen und deren Objektreferenzen

ID	Name
Anf. 1	Subjekt-Schablone (angelehnt an [SOP 13])
Anf. 2	Objekt-Schablone
Anf. 3	Akteur-Schablone
Anf. 4	Funktionale Anforderungsschablone ohne Bedingung [SOP 13]



Anf. 11	„Sobald“ Bedingungsschablone [SOP 13]
Anf. 12	„Solange“ Bedingungsschablone [SOP 13]

Tabelle 21: Anforderungs-Satzschablonen und deren Objektreferenzen

9.2 Logische Operatoren

Nachfolgend sind alle logischen Operatoren¹ aufgelistet, welche automatisch durch die semantische Rollenerkennung erkannt werden.

„größer“, „>“, „kleiner“, „<“, „gleich“, „=“, „==“, „ungleich“, „!=“, „größer gleich“, „>=“, „kleiner gleich“, „<=“, „und“, „&“, „&&“, „oder“, „|“, „||“, „zwischen“

¹ https://de.wikipedia.org/wiki/Logischer_Operator