

# **Design for Customer**

Methodik für nachhaltige Kundenlösungen unter Zuhilfenahme eines  
bedürfnisorientierten Leistungskonfigurators

Zur Erlangung des akademischen Grades eines

**DOKTORS DER INGENIEURWISSENSCHAFTEN**

Fakultät für Maschinenbau des  
Karlsruher Instituts für Technologie  
genehmigte

**DISSERTATION**

von

Dipl. Wi.-Ing. Alexander Burger  
geboren am 16. Oktober 1980

Tag der mündlichen Prüfung:	30. August 2013
Vorsitzender des Prüfungsausschusses:	Prof. Dr.-Ing. Martin Heilmaier
Hauptreferent:	Prof. Dr. Dr.-Ing. Dr. h.c. Jivka Ovtcharova
Korreferent:	Prof. Dr.-Ing. Rainer Stark



## **Vorwort**

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Informationsmanagement im Ingenieurwesen (IMI) des Karlsruher Instituts für Technologie (KIT).

Frau Prof. Dr. Dr.-Ing. Dr. h.c. Jivka Ovtcharova, Leiterin des Instituts, danke ich herzlich für das mir entgegengebrachte Vertrauen in meine Arbeit sowie die wissenschaftliche Betreuung bei der Erstellung der Dissertation.

Mein Dank gilt weiterhin Herrn Prof. Dr.-Ing. Rainer Stark, Leiter des Fachgebietes Industrielle Informationstechnik der Technischen Universität Berlin, für die Übernahme des Zweitgutachtens und die wertvollen Beiträge. Herrn Prof. Dr.-Ing. Martin Heilmaier danke ich für die Übernahme des Prüfungsvorsitzes.

Bei meinen ehemaligen Kollegen am Institut für Informationsmanagement im Ingenieurwesen bedanke ich mich für die vielen herausfordernden Diskussionen, die einen wertvollen Beitrag zum Gelingen dieser Arbeit beigetragen haben. Insbesondere bin ich meinen Kollegen Vitalis Bittel, Ramez Awad und Stilian Stanev für die vielen inhaltlichen Anregungen während und auch außerhalb der Arbeitszeit verbunden.

Ein besonderer Dank gilt meiner Familie, insbesondere meinen Eltern Doris und Dietrich Burger, die durch ihre Unterstützung meiner akademischen und beruflichen Entwicklung die Durchführung dieser Arbeit erst möglich gemacht haben.

Der größte Dank gilt meiner Frau Sonja, die mir durch ihre verständnisvolle Unterstützung und den Verzicht vieler gemeinsamer Stunden den nötigen Rückhalt gab und somit einen wertvollen Anteil zum Entstehen dieser Arbeit geleistet hat.

Karlsruhe, im August 2013

Alexander Burger



## **Kurzfassung**

Zunehmender Wettbewerbsdruck, insbesondere durch Niedriglohnländer sowie die Internationalisierung der Märkte stellen Unternehmen vor neue Herausforderungen. Die Differenzierung über technische Produktmerkmale aufgrund der wachsenden Vergleichbarkeit von Produktfunktionalitäten, Qualität und Preis verliert zunehmend an Gewicht. Eine Möglichkeit sich diesen Herausforderungen zu stellen ist die Vermarktung von Kundenlösungen als Kombination aus Produkten und Dienstleistungen, welche am Markt als integrierte Leistungsbündel zur Bewältigung kundenspezifischer Probleme angeboten werden. Häufig wird jedoch vernachlässigt, dass der Wandel vom produzierenden Unternehmen zum Lösungsanbieter ein tiefgreifendes Umdenken erfordert. Insbesondere gilt es, der hervorgehobenen Stellung des Kunden Rechnung zu tragen. Erst dadurch werden die notwendigen Voraussetzungen geschaffen, um die Potentiale eines Lösungsanbieters strategisch nutzen zu können.

Die im Rahmen des Forschungsvorhabens entwickelte Methodik ‚Design for Customer‘ baut auf diesem Handlungsbedarf auf. Sie stellt einen Ansatz zur Integration des Kunden in die Prozesse von Lösungsanbietern bereit, auf Basis dessen die Anforderungen und Wünsche des Kunden kontinuierlich erfasst und in die Entwicklungsprozesse integriert werden können. Die Kundenintegration erfolgt auf Grundlage eines bedürfnisorientierten Lösungskonfigurators, der eine Konfiguration basierend auf Kundennutzen und Kundenbedürfnissen anstelle von Produktmerkmalen erlaubt. Dabei wird natürliche Unschärfe in menschlicher Sprache berücksichtigt. Eine Analyse des erfassten Kundenwissens ermöglicht die nachgelagerte Rückführung in das Anforderungsmanagement. Diese erfolgt automatisiert und berücksichtigt bestehende und etablierte Entwicklungsmodelle.

Die Methodik stellt einen Baustein dar, die Nutzenpotentiale des Wandels zum Lösungsanbieter in ihrer Gänze auszuschöpfen. Neben der Stärkung der Kundenbeziehungen sowie der Differenzierung im Wettbewerb fällt darunter auch letztlich die Steigerung von Umsatz und Gewinnmarge.

## **Abstract**

An increased competitive pressure particularly from low-wage countries as well as the internationalization of markets is forcing companies to face new challenges. As one of the major challenges the aggravated differentiation of competitors solely based on product functionality, quality and price is to be mentioned. One way of dealing with these challenges is to offer

integrated product-service bundles, often referred to as product-service systems or customer solutions. In many cases, however, the fact is neglected that the change from manufacturing company to solution provider requires a profound change of business practices. Only an adequate realignment of business creates the necessary conditions being able to exploit the potentials of a solution provider holistically. In particular, the customer holds an emphasized position and needs to be integrated more holistically into the business activities of the solution provider.

The methodology postulated in the present dissertation project called ‘Design for Customer’ builds on this identified need for action and provides an approach for integrating the customer into the processes of solution providers. Based upon this methodology a solution provider is enabled to determine the requirements and needs of customers continuously and integrate them into the development processes of the company. This customer integration is facilitated using a necessity-based configuration system which allows a configuration on the basis of customer requirements and customer needs rather than product features. Thereby, natural uncertainty within human language is considered explicitly for ensuring an ideal configuration result. Subsequently, the customer information retrieved within the configuration process triggers a feedback stream into requirements management using appropriate methods of data mining. This process is carried out automatically and takes into account existing and well-established procedure models for product-service development.

Thus, the methodology ‘Design for Customer’ represents a building block for exploiting the potential benefits of solution providers entirely. Next to strengthening customer relationships and establishing differentiation potentials in comparison to competitors, the methodology ultimately targets the increase of revenue and profit margin.

# Inhaltsverzeichnis

<b>Vorwort</b> .....	<b>I</b>
<b>Kurzfassung</b> .....	<b>III</b>
<b>Inhaltsverzeichnis</b> .....	<b>V</b>
<b>Abkürzungsverzeichnis</b> .....	<b>1</b>
<b>1 Einleitung</b> .....	<b>3</b>
1.1 Motivation .....	3
1.2 Zielsetzung und Nutzen .....	4
1.3 Methodisches Vorgehen .....	5
<b>2 Grundlagen und Handlungsbedarf</b> .....	<b>9</b>
2.1 Kundenlösungen .....	9
2.1.1 Sachleistungen .....	10
2.1.2 Dienstleistungen .....	11
2.1.3 Kundenlösungen .....	13
2.1.4 Nutzenpotentiale des Wandels zum Lösungsanbieter .....	16
2.1.5 Vorgehensmodelle zur Entwicklung von Kundenlösungen .....	17
2.1.6 Zusammenfassung .....	23
2.2 Management von Kundenanforderungen .....	23
2.2.1 Grundlagen .....	23
2.2.2 Methoden der Anforderungsermittlung .....	26
2.2.3 Methoden der Anforderungsanalyse .....	28
2.2.4 Methoden der Anforderungsspezifikation .....	30
2.2.5 Integrierte Ansätze .....	32
2.2.6 Zusammenfassung .....	32
2.3 Industrieller Handlungsbedarf .....	33
2.3.1 Grundlagen .....	33
2.3.2 Analyse des Bedarfs an Kundenlösungen .....	34
2.3.3 Analyse der Integration von Kundenbedürfnissen .....	35
2.3.4 Zusammenfassung .....	37
2.4 Ableitung des Handlungsbedarfs .....	37
<b>3 Analyse ausgewählter Ansätze in Wissenschaft und Technik</b> .....	<b>41</b>
3.1 Ausgewählte Methoden der Künstlichen Intelligenz .....	41
3.1.1 Bayes'sche Netze .....	42
3.1.2 Künstliche neuronale Netze .....	44

3.1.3	Fuzzy-Systeme .....	47
3.1.4	Zusammenfassung .....	49
3.2	<i>Semantische Technologien</i> .....	50
3.2.1	Grundlagen .....	50
3.2.2	Sprachen zur Wissensrepräsentation .....	53
3.2.3	Fuzzy-Ontologien .....	54
3.2.4	Ontology Engineering .....	56
3.2.5	Zusammenfassung .....	60
3.3	<i>Knowledge Discovery in Databases</i> .....	60
3.3.1	Grundlagen .....	61
3.3.2	Herkömmliche Verfahren der Assoziationsanalyse .....	62
3.3.3	Verfahren zur Assoziationsanalyse bei unscharfen Datenbeständen .....	64
3.3.4	Zusammenfassung .....	65
3.4	<i>Zusammenfassung</i> .....	65
<b>4</b>	<b>Design for Customer – Methodik der nachhaltigen Kundenintegration .....</b>	<b>67</b>
4.1	<i>Zielsetzung der Methodik</i> .....	67
4.2	<i>Methodische Schritte</i> .....	68
4.3	<i>DfC Fuzzy-Controller</i> .....	70
4.3.1	Grundlegende Definitionen .....	71
4.3.2	Architektur des DfC Fuzzy-Controllers .....	75
4.3.3	Zusammenfassung .....	82
4.4	<i>DfC Fuzzy-Ontologie</i> .....	83
4.4.1	Annotationen .....	83
4.4.2	Syntaktische Anforderungen .....	84
4.4.3	Fuzzy-Erweiterungen von OWL 2 .....	85
4.4.4	Repräsentation von Fuzzy-Ontologien in OWL 2.....	87
4.4.5	Reasoning mit Fuzzy-Ontologien.....	94
4.4.6	Zusammenfassung .....	97
4.5	<i>DfC Unified Process</i> .....	98
4.5.1	Überblick: Der DfC Unified Process.....	98
4.5.2	Anforderungsworkflow .....	100
4.5.3	Analyseworkflow .....	101
4.5.4	Designworkflow .....	103
4.5.5	Implementierungsworkflow .....	104
4.5.6	Testworkflow.....	105
4.5.7	Stab.....	107
4.5.8	Zusammenfassung .....	108
4.6	<i>DfC Usability Engineering</i> .....	108

4.6.1	Überblick .....	109
4.6.2	Guidelines und Baukasten des Usability-Projektmanagements .....	110
4.6.3	Guideline und Baukasten der Usability-Anforderungserfassung .....	114
4.6.4	Guidelines und Baukasten der Usability-Evaluation .....	115
4.6.5	Zusammenfassung .....	118
4.7	<i>DfC Kundenanforderungsgenerierung</i> .....	119
4.7.1	Überblick .....	119
4.7.2	Grundlegende Definitionen .....	121
4.7.3	Der Fuzzy-Apriori-Algorithmus .....	124
4.7.4	Zusammenfassung .....	133
4.8	<i>DfC Anforderungsintegration</i> .....	134
4.8.1	Vorgehensweise .....	135
4.8.2	Zusammenfassung .....	136
4.9	<i>Nutzen der Methodik</i> .....	137
4.10	<i>Zusammenfassung</i> .....	140
<b>5</b>	<b>Validierung der Methodik</b> .....	<b>143</b>
5.1	<i>Validierungskonzept</i> .....	143
5.2	<i>Konsumgüter-Anwendungsszenario</i> .....	144
5.2.1	Definition des Anwendungsszenarios .....	144
5.2.2	Prototypische Realisierung .....	145
5.2.3	Anwendung der Methodik am Fallbeispiel .....	146
5.3	<i>Automotive-Anwendungsszenario</i> .....	166
5.3.1	Definition des Anwendungsszenarios .....	166
5.3.2	Prototypische Realisierung .....	167
5.3.3	Anwendung der Methodik am Fallbeispiel .....	168
5.4	<i>Zusammenfassung</i> .....	172
<b>6</b>	<b>Zusammenfassung und Ausblick</b> .....	<b>173</b>
6.1	<i>Zusammenfassung</i> .....	173
6.2	<i>Ausblick</i> .....	175
	<b>Glossar</b> .....	<b>179</b>
	<b>Anhang A. Methodiken des Ontology Engineering</b> .....	<b>183</b>
	<b>Abbildungsverzeichnis</b> .....	<b>195</b>
	<b>Tabellenverzeichnis</b> .....	<b>197</b>
	<b>Literaturverzeichnis</b> .....	<b>199</b>



## Abkürzungsverzeichnis

AHP	Analytic Hierarchy Process
BITV	Barrierefreie Informationstechnik-Verordnung
BPMN	Business Process Model and Notation
COA	Center of Area
CQ	Competence Question
DIN	Deutsches Institut für Normung
DL	Description Logic
EBIT	Earnings Before Interest and Taxes
ECR	Engineering Change Request
EN	Europäische Norm
FIPA	Foundation for Intelligent Physical Agents
HLB	Hybrides Leistungsbündel
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Standardization Organization
KDD	Knowledge Discovery in Databases
KI	Künstliche Intelligenz
KNN	Künstliches Neuronales Netz
KPI	Key Performance Indicator
MGMT	Management
MLP	Multilayer Perceptron
MVC	Model View Controller
OEM	Original Equipment Manufacturer
OWL	Ontology Web Language
PDD	Property-Driven Development
PSS	Product-Service System
PSSE	Product-Service Systems Engineering
QFD	Quality Function Deployment
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SWOT	Strength Weaknesses Opportunities Threats
UML	Unified Modeling Language
UP	Unified Process
UPON	Unified Process for Ontologies
VDI	Verein Deutscher Ingenieure
W3C	World Wide Web Consortium
XML	Extensible Markup Language



# 1 Einleitung

*“A customer is the most important visitor on our premises. He is not dependent on us. We are dependent on him. He is not an interruption in our work. He is the purpose of it. He is not an outsider on our business. He is part of it. We are not doing him a favor by serving him. He is doing us a favor by giving us an opportunity to do so.”*

(Mahatma Gandhi)

## 1.1 Motivation

Zunehmender Wettbewerbsdruck, insbesondere durch Niedriglohnländer sowie die Internationalisierung der Märkte stellen Unternehmen vor neue Herausforderungen. Die Differenzierung über technische Produkteigenschaften aufgrund der wachsenden Vergleichbarkeit von Produktfunktionalitäten, Qualität und Preis verliert zunehmend an Gewicht. Parallel dazu steigt die Erwartungshaltung der Kunden nach individuellen Lösungen, welche die ihnen immanenten, einzigartigen Anforderungen und Bedürfnisse befriedigen. Ein isoliertes Angebot an Produkten kann diese steigenden Kundenanforderungen zumeist immer weniger erfüllen. Unternehmen treten infolge dessen immer häufiger in einen intensiven Preiswettbewerb, was deren Gewinnmargen und Marktstellung negativ beeinträchtigt.

Viele Unternehmen stellen sich diesen Herausforderungen, indem sie ihr Kerngeschäft um Dienstleistungen erweitern und somit Lösungen und Geschäftsfunktionalitäten anstelle reiner Produkte offerieren (Meier & Uhlmann, 2012, S. 2). Im Zentrum dieses Wandels stehen Kundenlösungen als gezielte, aufeinander abgestimmte Kombinationen von Produkten und Dienstleistungen zu kundenindividuellen und maßgeschneiderten Komplettlösungen, deren kundenseitig wahrgenommener Wert den Wert der Leistungsbestandteile übersteigt.

Bei der Bereitstellung von Kundenlösungen im Rahmen der hybriden Wertschöpfung wandelt sich die Selbstauffassung des Unternehmens vom Produzenten zum Lösungsanbieter. Die Nutzenpotentiale einer erfolgreichen Transformation sind vielfältig sowohl für den Lösungsanbieter als auch für den Kunden. Für den Lösungsanbieter ergibt sich eine Differenzierung vom Wettbewerb, indem er den Nutzen des Kunden als den Mittelpunkt aller Bemühungen begreift, was sich wiederum in einer Steigerung der Marktmacht sowie einer Intensivierung der Kundenbindung niederschlägt. Zusätzlich erlaubt das Angebot von Kundenlösungen eine Steigerung der operativen Gewinnmarge des Unternehmens, da die Gewinnspanne von Dienstleistungen in der Regel weit über derer reiner Sachleistungen liegt. Der Kunde auf der anderen Seite profitiert von einem Mehrwert durch die Bereitstellung von auf ihn

zugeschnittenen Problemlösungen und kann sich somit auf sein eigentliches Kerngeschäft fokussieren.

Bei aller Euphorie um die hybride Wertschöpfung ist jedoch zu beachten, dass der Wandel vom produktorientierten Unternehmen hin zum Lösungsanbieter den schrittweisen Auf- und Ausbau zusätzlicher Kompetenzen erforderlich macht. Diesen gilt es in einem vielschichtigen Prozess strukturiert zu planen, durchzuführen und einer kontinuierlichen Fortschrittskontrolle zu unterziehen. Der Kunde als prozessauslösendes Element im Kontext der hybriden Wertschöpfung nimmt bei diesem Wandel einen zentralen Stellenwert ein, da die Strukturen und Prozesse des Unternehmens konsequent auf ihn auszurichten sind. Insbesondere ist für einen Lösungsanbieter die exakte Kenntnis der Anforderungen und Wünsche des Kunden essentiell, um eine zielgerichtete Bedürfnisbefriedigung auf Basis eines individuell anpassbaren Lösungsportfolios zu gewährleisten.

Die Ergebnisse einer im Rahmen dieser Arbeit durchgeführten Industriestudie verdeutlichen jedoch, dass genau an diesem Punkt ein Handlungsbedarf besteht. Zwar sehen knapp vier von fünf der befragten Studienteilnehmer den Kunden als Initiator von Leistungsverbesserungen, jedoch wird der Kunde nur in den wenigsten Fällen tatsächlich in die Unternehmensprozesse eingebunden. Strukturierte Vorgehensweisen zur Kundenintegration und Anforderungserfassung wie das Quality Function Deployment oder die Nutzwertanalyse werden von weniger als 15 Prozent der Unternehmen eingesetzt. Die Global CEO Studie von IBM (IBM Deutschland GmbH, 2010) stützt diese These: 88 Prozent der befragten CEOs sehen in einer Intensivierung der Kundenbeziehungen die wichtigste strategische Stoßrichtung der nächsten fünf Jahre.

### 1.2 Zielsetzung und Nutzen

Das vorliegende Dissertationsvorhaben baut auf dem identifizierten Handlungsbedarf auf und hat zum Ziel, eine Methodik zur Integration des Kunden in die Prozesse von Lösungsanbietern bereitzustellen, auf Basis derer die Anforderungen und Wünsche des Kunden kontinuierlich erfasst und in die Entwicklungsprozesse integriert werden können.

Zur Erreichung dieses Gesamtziels baut die Methodik insbesondere auf dem bestehenden und in Wissenschaft und Industrie etablierten Konzept der Trennung von Eigenschaften und Merkmalen (Weber, Steinbach, & Botta, 2004) (Steinbach, Botta, & Weber, 2005) (Botta, 2007) (Conrad, 2010) (Wanke, 2010) auf. Während Merkmale die Struktur, Gestalt und Beschaffenheit einer Kundenlösung beschreiben und vom Entwickler direkt beeinflusst werden können, stellen die nicht direkt beeinflussbaren Eigenschaften (Ovtcharova, 1997, S. 236) das Verhalten einer Kundenlösung dar. Die vom Kunden artikulierten Anforderungen beziehen sich zumeist auf die Eigenschaften einer Kundenlösung.

Auf der Grundlage dieser Trennung wird das gesetzte Gesamtziel dieser Arbeit in zwei Hauptteilen erreicht. In einem ersten Teil wird ein Ansatz zur Kundenintegration vorgestellt, der sich weniger am Produkt selbst als vielmehr am Kundennutzen ausrichtet. Dies erfolgt durch die Konzeption eines Lösungskonfigurators, der eine Konfiguration basierend auf Kundennutzen und Kundenbedürfnissen anstelle von Produktmerkmalen erlaubt. Dabei ist die Methodik so konzipiert, dass das Gesamtsystem bestehend aus Mensch, Werkzeug und Aufgabe den Bedürfnissen des Kunden entsprechend gestaltet ist und ihn durch die Nutzung natürlicher Interaktionspunkte unterstützt. Ziel des zweiten Teiles ist es, eine kontinuierliche Rückführung von Kundenanforderungen auf Basis des zuvor konzipierten Lösungskonfigurators als kontinuierliche Bezugsquelle sicherzustellen. Im Gegensatz zu herkömmlichen Ansätzen des Kundenanforderungsmanagements ermöglicht der vorliegende Ansatz eine automatisierte Rückführung der Anforderungen und minimiert somit den manuellen Aufwand für das beteiligte Unternehmen. Die Integration der identifizierten Anforderungen in die Entwicklungsprozesse wird über bestehende und etablierte Vorgehensmodelle der Entwicklung von Kundenlösungen sichergestellt.

Somit unterstützt die im Rahmen des vorliegenden Forschungsvorhabens entwickelte Methodik Lösungsanbieter darin, die Nutzenpotentiale der hybriden Wertschöpfung in ihrer Gänze auszuschöpfen. Durch die stringente Ausrichtung am Kundennutzen erlaubt die Methodik die Schaffung einer dauerhaften Kundenbindung sowie den Aufbau von Differenzierungspotential im Vergleich zum Wettbewerb. Durch die Kenntnis der Bedürfnisse der Kunden und die kontinuierliche Rückführung von Kundenanforderungen wird schlussendlich eine Steigerung der Gewinnmarge ermöglicht, indem die Entwicklung eines an Kundenbedürfnisse angepassten Leistungsportfolios sichergestellt wird.

### 1.3 Methodisches Vorgehen

Der Aufbau der Arbeit orientiert sich am forschungsmethodischen Vorgehen nach Ulrich (Ulrich, 1981), welches dem angewandten Charakter der ingenieurwissenschaftlichen Forschung Rechnung trägt.

Im *ersten Kapitel* werden der zugrundeliegende Handlungsbedarf sowie die Zielsetzung des Forschungsvorhabens kurz und übersichtlich herausgearbeitet sowie das forschungsmethodische Vorgehen erläutert. Eine detaillierte Identifikation des Handlungsbedarfs aus praktischer wie auch aus wissenschaftlicher Sicht findet in *Kapitel 2* statt. Die dort vorgenommene Analyse des Anwendungszusammenhangs aus wissenschaftlicher Sicht liefert grundlegendes Verständnis des relevanten Anwendungszusammenhangs sowie den Handlungsbedarf aus einer theoretischen Perspektive. Die Ergebnisse einer im Rahmen des Forschungsvorhabens durchgeführten



empirischen Industriestudie validieren den Handlungsbedarf in der Folge unter industriellen Gesichtspunkten. Abschließendes Ergebnis des zweiten Kapitels ist eine detaillierte Darstellung des abgeleiteten Handlungsbedarfs. Im *dritten Kapitel* werden bestehende problemrelevante Ansätze aus Wissenschaft und Technik mit dem Ziel analysiert, eine Basis zur Befriedigung des Handlungsbedarfs zu liefern. Diese dienen als Grundlage für den Aufbau der Methodik ‚Design for Customer‘ als elementarem Bestandteil des vorliegenden Forschungsvorhabens. Die Methodik wird in *Kapitel 4* vorgestellt und gliedert sich in fünf methodische Schritte, die sequentiell vorgestellt und abschließend einer Nutzenanalyse unterworfen werden. *Kapitel 5* validiert die Methodik hinsichtlich ihres Zielerreichungsgrades und ihrer praktischen Anwendbarkeit. Hierzu wurden zwei unabhängige Anwendungsszenarien aufgestellt, welche die

Anwendung der Methodik in einem praktischen Kontext aufzeigen und durch prototypische Softwarewerkzeuge unterstützt werden. Das sechste Kapitel fasst die Ergebnisse der Arbeit zusammen und gibt einen Ausblick auf weitere Handlungsfelder.

Abbildung 1 veranschaulicht die soeben vorgestellte Vorgehensweise grafisch. Die Forschungsmethodischen Schritte nach Ulrich sind auf der linken Seite ersichtlich und nehmen Bezug zur Struktur der vorliegenden Arbeit.



## **2 Grundlagen und Handlungsbedarf**

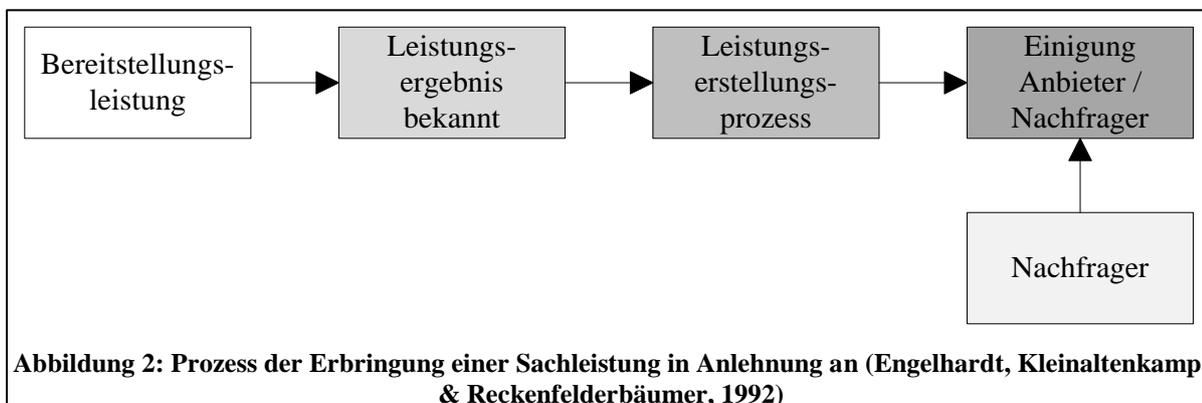
Die vorliegende Arbeit beschäftigt sich mit der Integration von Kunden und Anbietern im Kontext der hybriden Wertschöpfung. Zentrale Betrachtungsgegenstände der Arbeit sind daher Kundenlösungen als an Kundenwünsche angepasste Kombinationen aus Sach- und Dienstleistungen einerseits sowie die Anbieter-Kunden-Interaktion im Rahmen des Anforderungsmanagements andererseits. Ziel dieses Kapitels ist es, Grundlagenwissen zu diesen beiden Betrachtungsgegenständen zu vermitteln, um eine Basis für die eigentliche Arbeit zu schaffen. Darüber hinaus werden die Handlungsbedarfe der vorliegenden Arbeit hinsichtlich wissenschaftlicher und industrieller Gesichtspunkte abgeleitet. Hierbei wird zunächst auf Grundlagen der hybriden Wertschöpfung, deren Nutzenpotentiale sowie auf Vorgehensmodelle zur Entwicklung von Kundenlösungen eingegangen. Die hervorgehobene Stellung des Kunden im Kontext der hybriden Wertschöpfung sowie die Notwendigkeit zur Intensivierung der Kundenbeziehungen werden deutlich. Aufgrund dieser Erkenntnisse werden in der Folge bestehende Ansätze des Kundenanforderungsmanagements vorgestellt und kritisch beleuchtet. Aus der Analyse ergibt sich ein Handlungsbedarf nach neuartigen Ansätzen des Anforderungsmanagements, die den herausgearbeiteten Charakteristika von Kundenlösungen Rechnung tragen. Der identifizierte wissenschaftliche Handlungsbedarf wird abschließend hinsichtlich industrieller Gesichtspunkte validiert. Als Basis der Validierung dienen die Ergebnisse einer im Rahmen dieser Arbeit durchgeführten empirischen Untersuchung. Durch die integrierte Darstellung des validierten Handlungsbedarfs wird das Kapitel abgeschlossen.

### **2.1 Kundenlösungen**

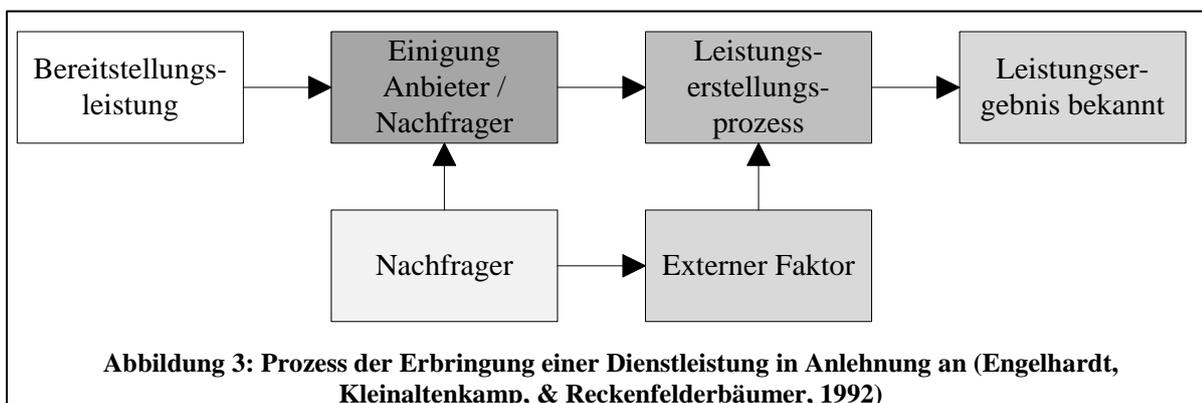
*Die hybride Wertschöpfung befasst sich mit der Entwicklung und Bereitstellung von Kundenlösungen als an Kundenwünsche angepasste Kombinationen aus Sach- und Dienstleistungen. Im Rahmen dieses Unterkapitels werden ausgehend vom Produktgeschäft Dienstleistungen als eigenständiges Betrachtungsobjekt vorgestellt und diese in den Kontext der hybriden Wertschöpfung gesetzt. Eine Übersicht der für die vorliegende Arbeit relevanten Vorgehensmodelle zur Entwicklung von Kundenlösungen sowie daraus abgeleitete Handlungsbedarfe bilden den Abschluss des Unterkapitels. Dabei wird insbesondere die Notwendigkeit zur Intensivierung der Kundenbeziehungen verdeutlicht.*

### 2.1.1 Sachleistungen

Sachleistungen oder auch Produkte werden nach Wöhe in die Elementarfaktoren Rohstoffe, Produktions-/Betriebsmittel und Verbrauchsgüter eingeteilt. Zu Sachleistungen zählen Material, gefertigte Vorprodukte und gefertigte Endprodukte (Wöhe, 2010, S. 540). Nach der betriebswirtschaftlichen Produktionstheorie von Gutenberg stehen als Elementarfaktoren Betriebsmittel, menschliche Arbeitsleistung und Werkstoffeinsatz sowie der dispositive Faktor zur Verfügung (Gutenberg, 1983, S. 2, 11ff.). Eine Sachleistung wird dem Kunden in der Folge in Form eines materiell existierenden Guts angeboten. Der Leistungsempfänger kann das materielle Produkt vor dem Kauf daher tangibel erfassen. Der Leistungsempfänger ist jedoch in der Regel vom Produktionsprozess ausgeschlossen. Den Prozess der Erstellung einer Sachleistung veranschaulicht Abbildung 2.



Während im folgenden Unterkapitel zur fundierten Unterscheidung zwischen Sach- und Dienstleistung eine Analyse des Begriffes Dienstleistung erfolgt, sei bereits vorab auf einen ersten Unterschied hingewiesen, welcher sich im Leistungserstellungsprozess niederschlägt. Demnach ist der Leistungsempfänger im Gegensatz zur Sachleistung als externer Faktor in den Dienstleistungsprozess unmittelbar eingebunden. Dieser Sachverhalt macht den Prozess der Erbringung einer Dienstleistung und somit die Dienstleistung an sich zu einem komplexen Betrachtungsgegenstand (Scheer, Griebler, & Klein, 2006, S. 22). Eine Übersicht über den Prozess der Erbringung einer Dienstleistung liefert Abbildung 3.

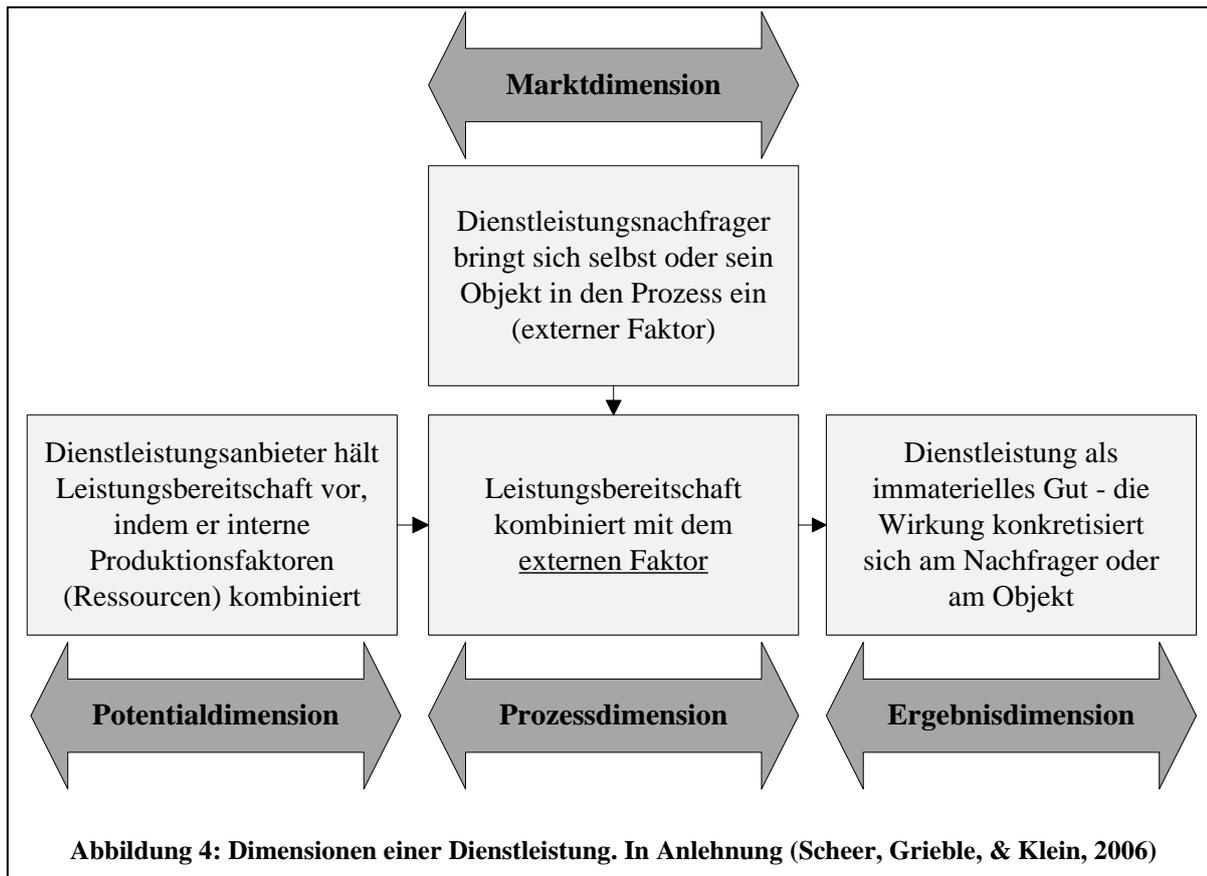


### 2.1.2 Dienstleistungen

Dienstleistungen werden von der Betriebswirtschaftslehre und insbesondere vom Marketing seit den 1980er Jahren intensiv betrachtet. Im deutschsprachigen Raum wird die Entwicklung von Dienstleistungen seit Mitte der 1990er Jahre unter dem Begriff Service Engineering diskutiert (Fähnrich & Opitz, 2006, S. 86). Im Gegensatz zur eher marketing-geprägten Sichtweise des US-amerikanischen Forschungsgebietes „New Service Development“ bedient sich der Ansatz des Service Engineering ingenieurwissenschaftlich geprägter Methoden und Werkzeuge (Meiren & Barth, 2002, S. 10ff.).

Traditionelle Definitionsansätze des Dienstleistungsbegriffs befassen sich insbesondere mit abgrenzenden Definitionsansätzen zwischen materiellen Sachleistungen und immateriellen Dienstleistungen. Darunter fallen enumerative Definitionen, negative Definitionen und institutionelle Abgrenzungsansätze (Corsten & Gössinger, 2007, S. 19ff.) (Meffert & Bruhn, 2009, S. 29) (Langeard, 1991, S. 233-235) (Nüttgens, Heckmann, & Luzius, 1998, S. 15). Im Rahmen konstitutiver Definitionsansätze werden Dienstleistungen hingegen charakteristische Merkmale zugesprochen, welche sie in positivem Sinne von Sachleistungen abgrenzen (Engelhardt, Kleinaltenkamp, & Reckenfelderbäumer, 1993, S. 398f.) (Corsten & Gössinger, 2007, S. 19ff.) (Meffert & Bruhn, 2009, S. 29). Ein konstitutives Merkmal ist eine prägende Eigenschaft, die grundlegend den Wesenskern einer Dienstleistung beschreibt (Nüttgens, Heckmann, & Luzius, 1998, S. 15). Besondere Bedeutung innerhalb dieser Merkmale haben die Immaterialität, sowie die Integration des Kunden als externen Faktor in den Leistungserstellungsprozess. Die überwiegende Mehrzahl konstitutiver Definitionsansätze gliedern Dienstleistungen in Dimensionen, welche bestimmten Phasen im Lebenszyklus zugeordnet werden können. Die ursprüngliche Idee der Differenzierung einzelner Ebenen und deren Integration in ein Gesamtbild geht zurück auf Donabedian (Donabedian, 1966). Während darauf beruhende traditionelle Ansätze lediglich die drei Dimensionen Potential, Prozess und Ergebnis beschreiben, wird zunehmend die Marktdimension als vierte Dimension mit in die Betrachtung aufgenommen. Somit soll die Bedeutung des Kunden als externer Faktor hervorgehoben werden (Meyer & Blümelhuber, 1994, S. 16ff.) (Chini, 1976, S. 3). Die vier Dienstleistungsdimensionen konstitutiver Definitionsansätze werden im Folgenden näher erläutert und sind in Abbildung 4 grafisch dargestellt.

- Die *potentialorientierte Dienstleistungsdimension* fokussiert die Fähigkeit des Anbieters, die notwendigen Potentialfaktoren vorzuhalten, um mittels deren Kombination die Dienstleistung in der Folge tatsächlich erbringen zu können (Engelhardt, Kleinaltenkamp, & Reckenfelderbäumer, 1992, S. 399).



- Die *Marktdimension* spiegelt den Markt sowie den Kunden als externen Faktor wider, welcher unmittelbar in den Leistungserstellungsprozess zu integrieren ist, um einen kontinuierlichen Abgleich der Kundenanforderungen mit den Leistungen des Anbieters zu gewährleisten.
- Basierend auf den Anforderungen der Marktdimension werden in der *Prozessdimension* die Potentialfaktoren über vordefinierte Prozesse auf den externen Faktor übertragen (Meffert & Bruhn, 2009, S. 12ff., 29). Charakteristisches Merkmal ist die Simultanität von Leistungserstellung und Leistungsabgabe, das heißt Erbringung und Verbrauch der Dienstleistung erfolgen gleichzeitig (Uno-actu-Prinzip) (Meffert & Bruhn, 2009, S. 29).
- Die *ergebnisorientierte Dimension* beschreibt den Zustand, der nach vollzogener Faktorkombination, also nach Abschluss des Dienstleistungsprozesses, vorliegt. Charakteristisch für die Ergebnisdimension ist deren Immaterialität (Griebler & Scheer, 2000, S. 27).

Zentraler Punkt des Forschungsgebietes Service Engineering ist es, Dienstleistungen in Analogie zur Produktentwicklung als Entwicklungsobjekt aufzufassen, welches eine kontinuierliche und lebenszyklusbegleitende Handhabung und Verwaltung auf Basis informationstechnischer Referenzmodelle ermöglicht (Burger, Bittel, Awad, Stanev, & Ovtcharova, 2010, S. 5). Die

Tatsache, dass der wirtschaftliche Erfolg eines Dienstleistungsangebots maßgeblich von dessen Konstruktion und kundenindividueller Gestaltung abhängt, wird heutzutage in der Praxis jedoch häufig vernachlässigt (Thomas & Nüttgens, 2009, S. V). Die zentrale Herausforderung bei der Bereitstellung eines Dienstleistungsportfolios liegt demzufolge in der systematischen Entwicklung sowie der kontinuierlichen Verbesserung von Dienstleistungen an die Bedürfnisse des Marktes. Substantielle Vorgehensweisen in Analogie zur Produktentwicklung sind jedoch kaum verbreitet. Methoden und Werkzeuge der Informationstechnologie unterstützen die systematische Gestaltung nur unzureichend (Thomas & Scheer, 2003, S. 1).

### 2.1.3 Kundenlösungen

Aktuell erhält die Dienstleistungsforschung einen Innovationsschub, der durch Bestrebungen zur modellbasierten Gestaltung hybrider Produkte hervorgerufen wird (Thomas & Nüttgens, 2009, S. V-VI). Dabei werden die Begriffe hybrides Leistungsbündel (HLB) (Meier, Uhlmann, & Kortmann, Hybride Leistungsbündel, Nutzenorientiertes Produktverständnis durch interferierende Sach- und Dienstleistungen, 2005), hybrides Produkt (Scheer, Griebel, & Klein, 2006), Product-Service System (PSS) (McAloone & Andreasen, 2004) oder Kundenlösung (Davies, Brady, & Hobday, 2006) oftmals synonym verwendet. Im Rahmen der vorliegenden Arbeit wird vorrangig der Begriff Kundenlösung verwendet.

Die zentrale Annahme, die der hybriden Wertschöpfung zugrunde liegt, ist die Erkenntnis, dass Kunden kein Interesse nach Produkten oder Dienstleistungen *per se* haben, sondern vielmehr die Lösung einer Problemstellung oder eines Bedürfnisses erwarten (Leimeister & Glauner, 2008, S. 248). Die Frage, ob die seitens der Kunden geforderten Produkteigenschaften durch eine Sach- oder eine Dienstleistung erfüllt werden, rückt zunehmend in den Hintergrund (Thomas & Nüttgens, 2009, S. VI). Aus dieser Problemstellung heraus ergeben sich - im Vergleich zur Bereitstellung von Einzelleistungen (Sach- oder Dienstleistung) oder nicht-integrierter Leistungspakete - deutlich gesteigerte Anforderungen an ein Unternehmen, um die angestiegene Erwartungshaltung der Kunden befriedigen zu können (Becker, et al., 2008, S. 112). Unternehmen müssen demzufolge jederzeit die grundlegenden Probleme und Bedürfnisse ihrer Kunden kennen, um ein darauf abgestimmtes Leistungsportfolio entwickeln und bereitstellen zu können. In Zeiten sich schnell wandelnder Märkte ergibt sich die Dringlichkeit eines hohen Maßes der Kundenintegration, um schnell auf geänderte Bedürfnisse reagieren und ein angepasstes Leistungsportfolio anbieten zu können.

Innerhalb der einschlägigen Literatur werden Kundenlösungen von einfachen Leistungspaketen durch drei zentrale Charakteristika definitorisch voneinander abgegrenzt. Eine Kundenlösung ist demnach (1) eine Kombination aus Sach- und Dienstleistungen, die (2) speziell an die Kundenwünsche angepasst werden können und (3) welche eng miteinander verbunden

sind (Tuli, Kohli, & Bharadwaj, 2007) (Sawhney, 2006) (Davies & Brady, 2000). Anhand der vorliegenden Definition lassen sich die beiden zentralen Aspekte des Begriffes Kundenlösung verdeutlichen: Integration sowie Individualisierung.

### 2.1.3.1 Integrationsaspekt von Kundenlösungen

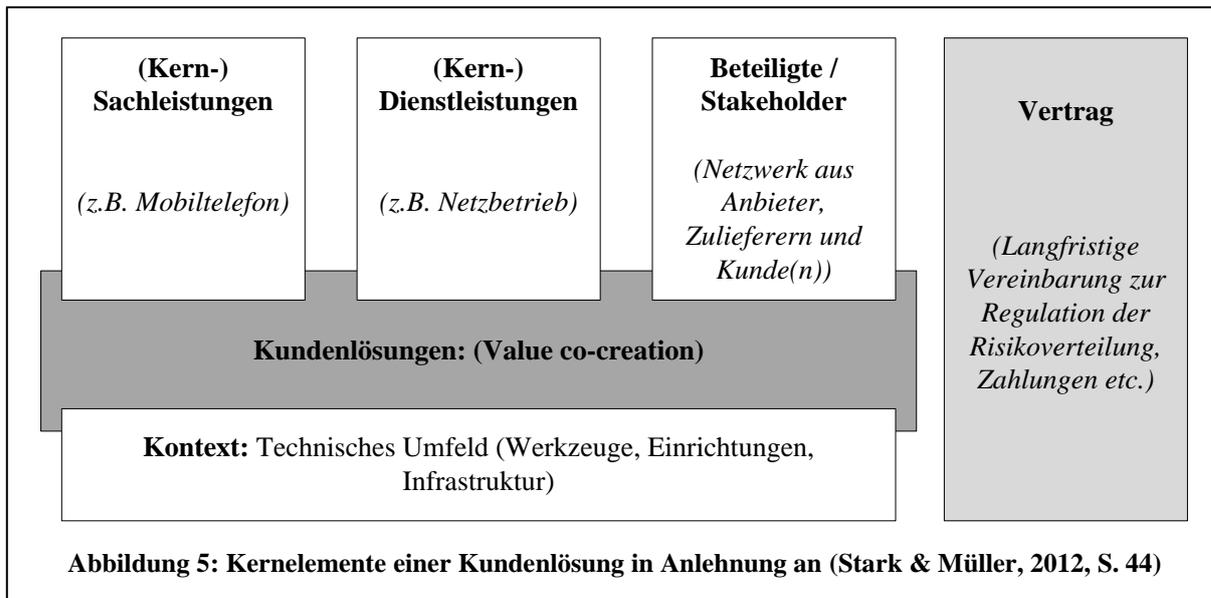
Die Integration als erstes charakteristisches Merkmal von Kundenlösungen soll bewirken, dass der wahrgenommene Wert eines hybriden Produktes aus Kundensicht den Wert bei Inanspruchnahme der Einzelleistungen übersteigt (Stauss & Bruhn, 2007, S. 17) (Johansson, Krishnamurthy, & Schlissberg, 2003, S. 117) (Sawhney, 2006, S. 372). Der mit der Integration verbundene Nutzenanstieg muss daher höher sein als die mit der Integration verbundenen Kosten. In Bezug auf die hybride Wertschöpfung unterscheidet die einschlägige Literatur die Integration in eine technische sowie eine marketingbezogene Ebene. Seltener werden darüber hinaus organisatorische und wertschöpfende Ebenen genannt.

- Die *technische Integration* beschreibt, in welchem Ausmaß Sach- und Dienstleistungsbestandteile innerhalb eines Lösungsangebotes aufeinander abgestimmt sind. Die einzelnen Komponenten der Sach- und Dienstleistungsdimension sind so zu entwickeln und bereitzustellen, dass durch die Kombination der einzelnen Komponenten ein zusätzlicher Nutzen für den Kunden entsteht (Stauss & Bruhn, 2007, S. 249).
- Die *marketingbezogene Integration* erlaubt es dem Kunden, die für sein Problem spezifische Lösung direkt aus einer Hand beziehen zu können. Anstelle mehrerer Anbieter tritt der Lösungsanbieter als Generalunternehmer auf, wodurch für den Abnehmer ein Wertzuwachs über den gesamten Entscheidungs- und Kaufzyklus entsteht (Sawhney, 2006, S. 372). Unter diesem Zusammenhang fallen partnerschaftliche Arrangements von Lösungsanbietern mit Lieferanten und Wettbewerbern (Windahl, Andersson, Berggren, & Nehler, 2004, S. 227).

In Kombination sorgen beide in diesem Abschnitt erläuterten Integrationsformen dafür, dass sich ein Lösungsanbieter durch integrierte und weniger substituierbare Lösungsangebote vom Markt differenzieren kann. Dabei gilt, dass das Differenzierungspotential im Vergleich zum Wettbewerb umso ausgeprägter erkannt wird, je höher der Integrationsgrad auf technischer und marketingbezogener Ebene vorliegt (Burger, Bittel, Awad, Stanev, & Ovtcharova, 2010, S. 4f.).

### 2.1.3.2 Individualisierungsaspekt hybrider Leistungsbündel

Der Individualisierungsaspekt des Lösungsbegriffs spricht die Fähigkeit eines Unternehmens an, kundenindividuelle Bedürfnisse auf Basis eines vordefinierten Leistungsangebotes möglichst vollständig befriedigen zu können. Die Integration einzelner Kunden beschränkt sich dabei nicht darauf, den Kunden im Rahmen der Leistungsspezifikation zu integrieren. Der Kunde ist stattdessen als gleichberechtigter Wertschöpfungspartner frühzeitig und kontinuierlich in die

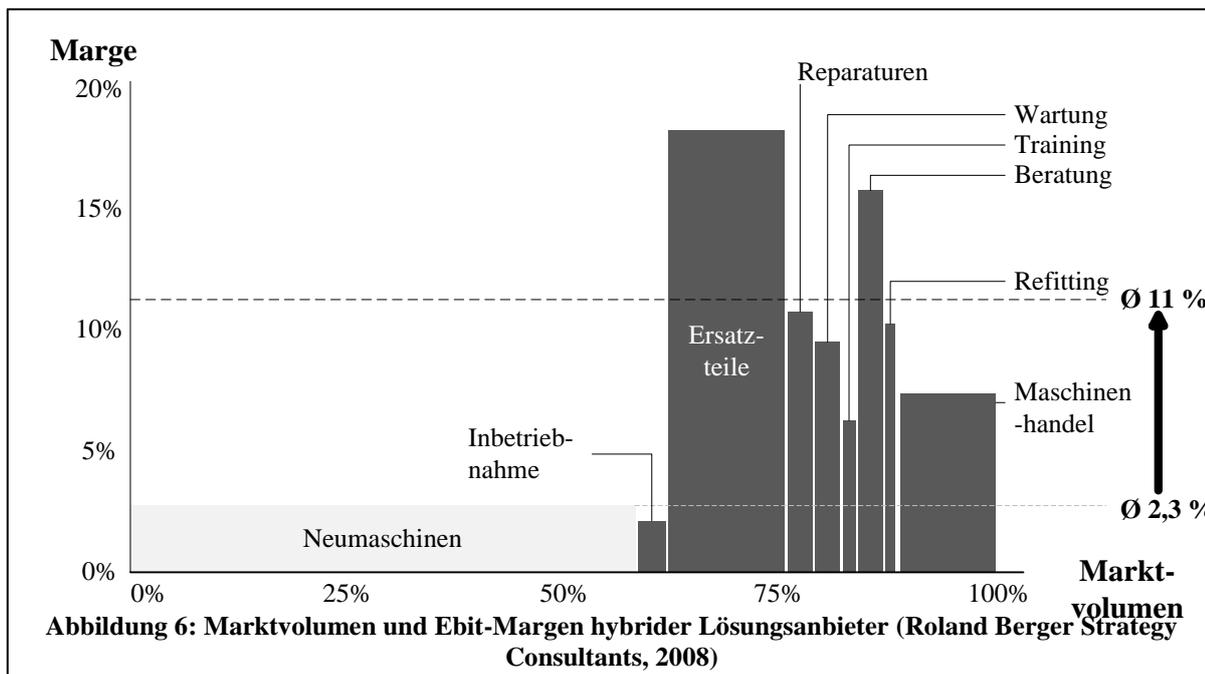


Prozesse des Anbieters einzubinden (Ericson, Müller, Larsson, & Stark, 2009, S. 62f.). Der Prozess der individuellen Kundenintegration lässt sich dabei in die Prozessschritte Analyse, Lösungsspezifikation, Lösungserbringung sowie Lösungsergebnis untergliedern. Die Analysephase dient dem Anbieter zum Verständnis der spezifischen Probleme und Bedürfnisse des Kunden. In der Phase der Lösungsspezifikation wird unter Mitwirkung des Kunden eine problemadäquate Ausgestaltung der Lösung entwickelt, die innerhalb der Phase der Lösungserbringung und des Lösungsergebnis auf den Kunden und wiederum unter dessen Mitwirkung übertragen wird.

Damit trägt die Individualisierung neben dem Integrationsaspekt als zweites charakteristisches Merkmal zu einem hohen wahrgenommenen Wert aus Kundensicht (Schmitz, 2008, S. 665) sowie zur Differenzierung des Anbieters gegenüber dem Wettbewerb bei. Voraussetzung zur Entwicklung wertsteigernder, individueller Lösungen ist die Identifikation sowie die daran anschließende differenzierte Analyse der jeweiligen Kundenprobleme, sowie deren Inkorporation in das angebotene Leistungsspektrum (Davies, 2004, S. 732ff.).

### 2.1.3.3 Kernelemente von Kundenlösungen

Eine Kundenlösung integriert neben Sach- und Dienstleistungen auch die prozessbeteiligten Personen wie Anbieter, Kunde und Zulieferer. Dabei ist die Prozessintegration von Wertschöpfungsprozessen der Beteiligten essentiell für eine gemeinsame Wertschöpfung (Stark & Müller, 2012, S. 43). Abbildung 5 stellt die Kernelemente einer Kundenlösung gemäß der klassischen Trennung von Sach- und Dienstleistungen dar. Eine detaillierte Beschreibung der Elemente ist in der öffentlichen Spezifikation DIN PAS 1094 (Deutsches Institut für Normung e.V., 2009) ersichtlich.



#### 2.1.4 Nutzenpotentiale des Wandels zum Lösungsanbieter

Eine strukturierte Einführung des Lösungsgedankens in einem Unternehmen unter Beachtung des Integrations- als auch des Individualisierungsaspektes ist mit enormen Anstrengungen verbunden. Der aus dem Wandel zum Lösungsanbieter resultierende Nutzen muss demnach die Kosten des Wandels sowie die Folgekosten übersteigen. Der Nutzen lässt sich in die drei Kategorien Aufbau von Differenzierungspotential, Steigerung der Gewinnmarge und Dauerhafte Kundenbindung untergliedern (Ovtcharova & Burger, 2011, S. 14).

*Aufbau von Differenzierungspotential:* Industrieunternehmen vieler Branchen sehen sich seit einigen Jahren einem fundamentalen Wandel der globalen wirtschaftlichen Rahmenbedingungen konfrontiert. Dieser Wandel wird bedingt durch einen verschärften internationalen Wettbewerb, insbesondere aus Niedriglohnländern, und resultiert in einer Homogenisierung der angebotenen Produkte in technischer sowie qualitativer Sichtweise (Homburg, Faßnacht, & Günther, 2002, S. 487) (Günther, 2001, S. 1f.). Eine Differenzierung über Innovation, Technologie oder Qualität und damit die Generierung nachhaltiger Wettbewerbsvorteile wird für produzierende Unternehmen demnach zunehmend schwieriger. Als Folge der fehlenden Möglichkeit, sich über die zur Verfügung stehende Produktpalette zu differenzieren, treten die Marktteilnehmer in einen intensiven Preiswettbewerb. Empirische Studien belegen, dass Industrieunternehmen den wachsenden Preisdruck als Folge einer mangelnden Möglichkeit der Differenzierung vom Wettbewerb als wichtiges Problemfeld im wirtschaftlichen Umfeld erkennen, welchem es zu begegnen gilt (McKinsey Corporation, 2005, S. 61). Der Ausbau des Dienstleistungsgeschäftes

sowie die Integration in das bestehende Produktportfolio stellen einen Weg dar, verlorenes Differenzierungspotential zurückzugewinnen.

*Steigerung der Gewinnmarge:* Neben der Möglichkeit, Differenzierungspotentiale aufzubauen und somit seine Marktmacht zu steigern, erlaubt der Eintritt in das Dienstleistungsgeschäft eine Steigerung der Gewinnmarge des Unternehmens. Die Margen produzierender Unternehmen des Anlagenbaus liegen nach einer Studie des Beratungshauses Roland Berger bei rund 2,3%, während die mithilfe von Kundenlösungen erzielbare Marge mit bis zu 11,0% weit über der des Neumaschinengeschäfts liegt (Roland Berger Strategy Consultants, 2008, S. 29) (Gebauer, Hildenbrand, & Fleisch, 2006, S. 47). Die Entwicklung und Bereitstellung lebenszyklusphasenspezifischer Dienstleistungen erlaubt im Gegensatz zum Neumaschinengeschäft eine langfristig stabile Einnahmequelle. Daneben ist für den Vertrieb von Dienstleistungen im Vergleich zum Produktgeschäft weniger gebundenes Kapital notwendig (Bruhn, 2006, S. 241).

*Dauerhafte Kundenbindung:* Über die Bereitstellung von Kundenlösungen wird das Ziel verfolgt, Kunden dauerhaft an das Unternehmen zu binden (Woisetschläger, Backhaus, & Michaelis, 2009). Eine tiefgreifende Integration des Lösungsanbieters in die Prozesse der Kunden schafft ein dauerhaftes Vertrauensverhältnis in der Kunde-Anbieterbeziehung und baut somit Wechselbarrieren ab. Unternehmen wird es ermöglicht, sich einer direkten Vergleichbarkeit zum Wettbewerb entziehen zu können. Beispielhaft sei hier als Erfolgsbeispiel die Transformation von IBM zum „Solution Seller“ genannt (Gerstner, 2002).

Zur Umsetzung der vorgestellten Nutzenpotentiale in der Praxis ist ein schrittweiser Ausbau der Kompetenzen eines Unternehmens erforderlich. Eine detaillierte Darstellung der für den Wandel zum Lösungsanbieter relevanten Kompetenzen ist in Burger et al. (Burger, Ovtcharova, & Brenner, 2012) ersichtlich. Eine wesentliche Kompetenz stellen die Entwicklungsprozesse des Unternehmens dar, welche die die Grundlage für ein an Kundenwünsche angepasstes Lösungsportfolio bilden. Im Folgenden werden Vorgehensmodelle zur Unterstützung dieser Entwicklungsprozesse vorgestellt.

### 2.1.5 Vorgehensmodelle zur Entwicklung von Kundenlösungen

Entwicklungsprozesse sind durch ein hohes Maß an Kreativität gekennzeichnet und weisen eine hohe Komplexität auf. Zur Minimierung dieser Komplexität haben sich in der Konstruktionslehre verschiedene Vorgehensmodelle etabliert, die eine strukturierte Entwicklung technischer Systeme basierend auf konkreten Handlungsanweisungen ermöglichen (Becker, Beverungen, & Knackstedt, 2008, S. 106). Beispielhaft sei auf die VDI-Richtlinie 2221 (Verein Deutscher Ingenieure, 1993) verwiesen, die sich über die letzten Jahre im deutschsprachigen

Raum für den Bereich der Sachleistungsentwicklung als De Facto-Standard etabliert hat. Die hybride Wertschöpfung baut auf solchen etablierten Grundsätzen aus den klassischen Ingenieurwissenschaften auf und entwickelt angelehnte Vorgehensweisen. Bei der Entwicklung dieser Vorgehensmodelle sind die grundlegenden Charakteristika von Kundenlösungen zu berücksichtigen. In Anlehnung an Gräßle et al. (Gräßle, Thomas, Fellmann, & Krumeich, 2010, S. 2033) ergeben sich die wichtigsten Gestaltungsrichtlinien der hybriden Wertschöpfung zu:

- Sach- und Dienstleistungen sind als gleichberechtigte Bestandteile von Kundenlösungen in die Vorgehensmodelle zu integrieren. Vorgehensmodelle, die einen Fokus auf die Entwicklung von Sach- oder Dienstleistungen legen, verkennen das Wesen der hybriden Wertschöpfung, wonach Kunden kein Interesse nach Produkten oder Dienstleistungen per se haben (Thomas et al. 2008, S. 209f.).
- Zur Sicherstellung der Bedürfnisbefriedigung des Kunden sind geeignete Methoden des Anforderungsmanagements zu berücksichtigen. Eine fehlende Integration des Kunden in die Entwicklungsprozesse kann eine nicht auf die Anforderungen des Kunden abgestimmte Realisierung der Kundenlösungen zur Folge haben. (Sturm & Bading, 2008, S. 184) (Rexfelt, Ornäs, & V.H., 2008, S. 15ff.). Diese Integration ist über den gesamten Lebenszyklus zu gewährleisten und methodisch rückzukoppeln (Müller & Blessing, Entwicklungsprozesse hybrider Leistungsbündel - Erweiterter Handlungsraum für den HLB-Entwickler, 2007, S. 519)
- Die Entwicklung von Kundenlösungen sollte nicht auf ein einzelnes Unternehmen beschränkt sein, da diese oftmals nicht über die erforderlichen Ressourcen und Kompetenzen verfügen. (Beverungen, Knackstedt, & Müller, 2008, S. 220f.).

Eine detaillierte Übersicht und eine kritische Analyse sämtlicher existierenden Vorgehensmodelle im Bereich der hybriden Wertschöpfung ist in (Gräßle, Thomas, Fellmann, & Krumeich, 2010) ersichtlich. Im Rahmen der vorliegenden Arbeit soll lediglich eine Auswahl relevanter Vorgehensmodelle vorgestellt werden. Zunächst wird auf die Ansätze von Weber et al. (Weber, Steinbach, & Botta, 2004) und Thomas et al. (Thomas, Walter, & Loos, 2010) eingegangen, welche auf der Trennung von Eigenschaften und Merkmalen (siehe Definition in Abschnitt 2.1.5.1) basieren und somit die theoretische Grundlage dieser Arbeit teilen. Anschließend wird die umfassendere HLB-Entwicklungsmethodik (Stark & Müller, 2012) vorgestellt, die eine Auswahl praktikabler Methoden zum operativen Einsatz bereitstellt, unter die auch die Ergebnisse dieser Arbeit fallen können.

### 2.1.5.1 Entwicklungsmethodik des PSSE

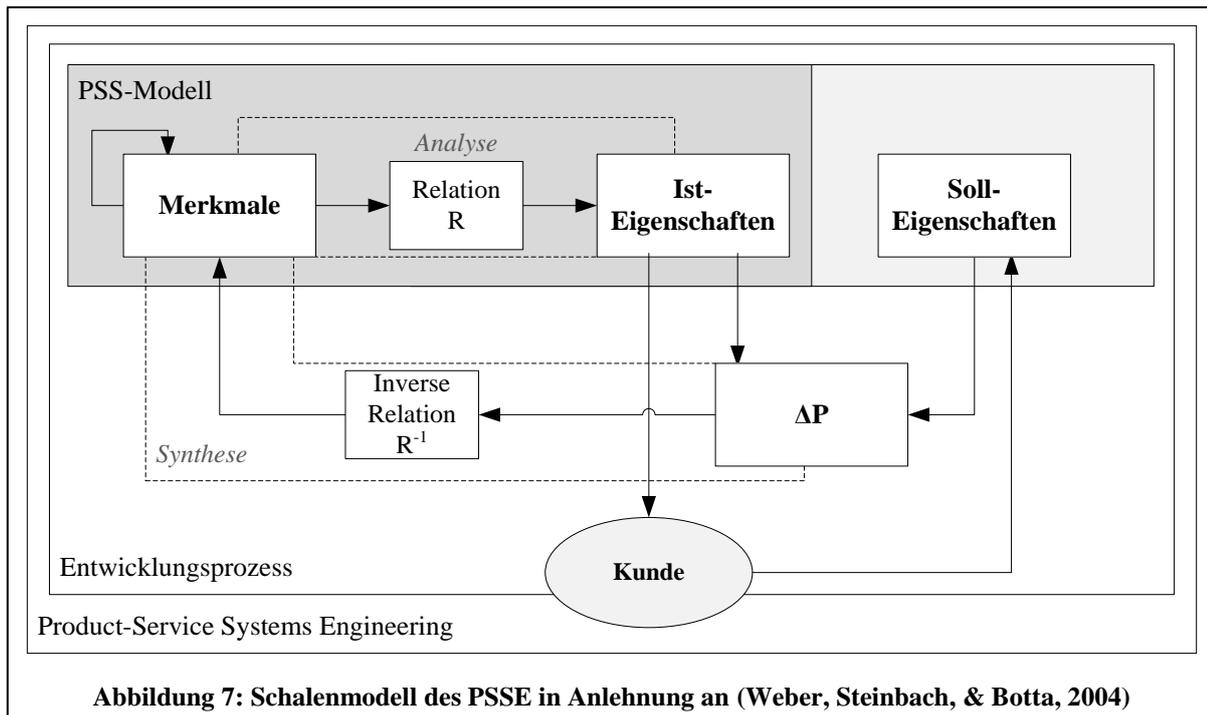
Das Phasenmodell der PSSE (Product-Service Systems Engineering) basiert auf dem Konzept des Property-Driven Development (PDD), dem die Trennung von Eigenschaften und Merkmalen

als zentralen Gestaltungspunkt innewohnt. Eine umfassende Darstellung des Konzeptes findet sich in (Weber, Steinbach, & Botta, 2004) (Steinbach, Botta, & Weber, 2005) (Botta, 2007) (Conrad, 2010) (Wanke, 2010). Zentrale Objekte beider Ansätze stellen Eigenschaften und Merkmale dar, die angelehnt an Conrad (Conrad, 2010, S. 40f.) wie folgt definiert werden:

- *Merkmale* beschreiben die Struktur, Gestalt und Beschaffenheit einer Kundenlösung und können vom Entwickler direkt beeinflusst werden. Beispiele für Merkmale sind Geometrie, Materialien, Dimensionen oder Oberflächen.
- *Eigenschaften* beschreiben das Verhalten einer Kundenlösung und können vom Entwickler nicht direkt beeinflusst werden. Die vom Kunden artikulierten Anforderungen beziehen sich zumeist auf die Eigenschaften eines PSS (Steinbach, Botta, & Weber, 2005, S. 547). Beispiele für Eigenschaften sind Sicherheit, Ästhetik, Umwelteigenschaften oder Kosten.

Ziel der Entwicklungsmethodik des PSSE (Weber, Steinbach, & Botta, 2004) ist es, basierend auf einem Eigenschaftenkatalog die Entwicklung der gewünschten Kundenlösung in Form von Merkmalen zu ermöglichen. Innerhalb der PSSE Methodik erfolgt die Bestimmung von Eigenschaften und Merkmalen durch Iterationen von Synthese- und Analyseschritten. Der schematische Aufbau ist in Abbildung 7 dargestellt. Ausgangspunkt des Ansatzes sind die Anforderungen des Kunden, die als Soll-Eigenschaften in das Schalenmodell aufgenommen werden.  $\Delta P$  stellt zu jedem Zeitpunkt des Entwicklungsprozesses die aktuelle Differenz zwischen den vom Kunden geforderten Soll-Eigenschaften und den tatsächlich in der Kundenlösung vorhandenen Ist-Eigenschaften dar. Über Syntheseschritte werden einzelne Soll-Eigenschaften dadurch realisiert, dass Merkmale innerhalb der Kundenlösung angepasst oder gänzlich neu entwickelt werden. Die Synthese stellt den eigentlichen kreativen Prozess innerhalb der Entwicklung dar und kann durch Integration bestehender Synthesemethoden wie der VDI Richtlinie 2221 unterstützt werden. Durch die Definition neuer oder angepasster Merkmale resultieren Auswirkungen auf die in der Kundenlösung vorhandenen Ist-Eigenschaften, die innerhalb eines Analyseschrittes an die veränderten Merkmale angepasst werden.

Somit kann aus einem Abgleich von aktuellen Ist- und Soll-Eigenschaften in einem nächsten Schritt  $\Delta P$  neu ermittelt werden. Das resultierende Ergebnis ist Ausgangspunkt weiterer Syntheseschritte und gleichzeitig Steuergröße für den Gesamtprozess. So wird sukzessive die Merkmalsebene gefüllt, was eine weitere Detaillierung und ein Fortschreiten im Entwicklungsprozess bedeutet. Die Identifikation von Kundeneigenschaften ist dabei kontinuierlich durchzuführen, um Änderungen der Soll-Eigenschaften durch adäquate Schritte auffangen zu können. Aus Gründen der Verständlichkeit und Nachvollziehbarkeit wird die entwickelte Methode am Beispiel der Kundenlösung „Guter Schlaf“ illustriert, in dem materielle und immaterielle Leistungen so konfiguriert werden, dass sie dem Kunden eine erholsame Nachtruhe sichern (Thomas, Walter, & Loos, 2008, S. 209). Die Identifikation der

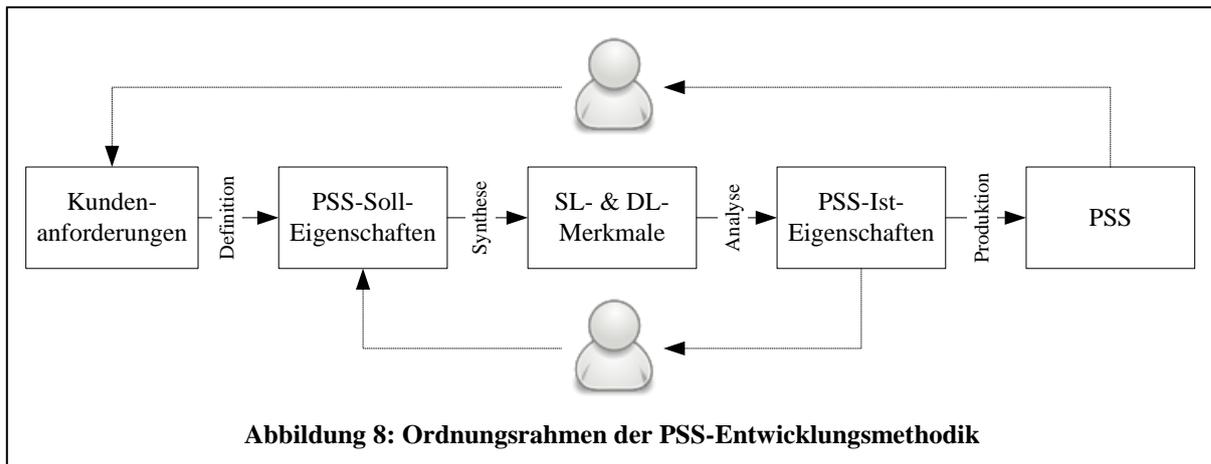


Kundeneigenschaften wird innerhalb des Anwendungsszenarios manuell durchgeführt und sukzessive verfeinert. Eine Möglichkeit der automatischen und kontinuierlichen Überwachung der vom Kunden geforderten Soll-Eigenschaften wird nicht vorgestellt.

### 2.1.5.2 Ordnungsrahmen der PSSE Entwicklung

Der Ansatz nach Thomas, Walter und Loos (Thomas, Walter, & Loos, 2010) baut auf der Methodik des PSSE auf und verwendet insbesondere die aus dem Property-Driven Development (PDD) stammende Unterscheidung zwischen strukturbeschreibenden Merkmalen und verhaltensbeschreibenden Eigenschaften. Eine Kundenlösung wird innerhalb des Ansatzes in einem iterativen und inkrementellen Prozess entwickelt, der sich aus zwei Zyklen zusammensetzt. Innerhalb des übergeordneten Zyklus steht der Kunde mit seinen Anforderungen sowie als Abnehmer der resultierenden Kundenlösung im Mittelpunkt. Der darunterliegende innere Zyklus dient dem Entwickler zur iterativen Anpassung der zu entwickelnden Kundenlösung an die vom Kunden geäußerten Anforderungen.

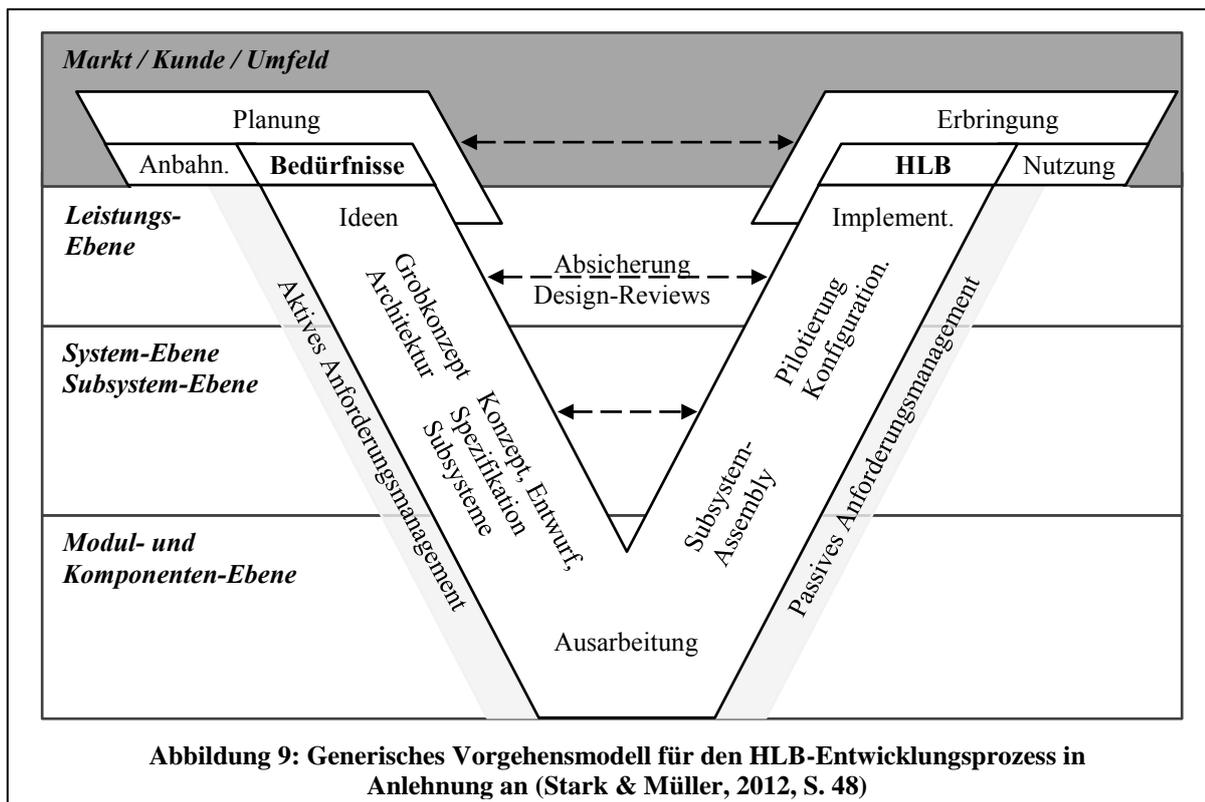
Beide Zyklen bilden gemeinsam einen fünfphasigen Ordnungsrahmen, der mit der Phase zur Bestimmung der Kundenanforderungen beginnt. Die Autoren betonen die Wichtigkeit der Kundenanforderungen für den wirtschaftlichen Erfolg eines Unternehmens und stellen dies als einen Differenzierungspunkt gegenüber dem Ansatz des PSSE heraus. Die Bestimmung der Kundenanforderungen erfolgt aus Aufwandsgründen einmalig zu Beginn eines jeden Entwicklungszyklus. Basierend auf den (Müller & Stark, A Generic PSS Development Process Model Based on Theory and an Empirical Study, 2010) Kundenanforderungen werden in einer



zweiten Phase die Soll-Eigenschaften der Kundenlösung definiert. Diese übersetzen die Kundenanforderungen aus der Sprache des Kunden in die Sprache des Entwicklers. Als Beispiel geben die Autoren die Kundenanforderung „ruhiger Innenraum“ bei einem Automobil an, welche in die beiden Eigenschaften „gute Dämpfung von Außengeräuschen“ und „guter Ausgleich von Bodenunebenheiten“ transformiert werden kann. In der Synthesephase werden auf Basis der Soll-Eigenschaften der Kundenlösung deren Sach- und Dienstleistungsmerkmale bestimmt. Basierend auf der so entstandenen Merkmalsstruktur werden in der Analysephase die Ist-Eigenschaften auf Basis der Merkmalsstruktur der Synthesephase abgeleitet. Die Analysephase erlaubt dem Entwickler, den Grad der Zielerreichung einer von ihm konstruierten Kundenlösung zu überprüfen und gegebenenfalls einen weiteren Entwicklungszyklus zu durchlaufen. In der abschließenden Phase der Produktion der Kundenlösung wird „die Kombination und Transformation von Produktionsfaktoren (...) zu Produkten“ (Schweitzer, 1993, S. 3328) verstanden. Der Ordnungsrahmen wird beispielhaft anhand eines Anwendungsszenarios im Bereich der Bereitstellung mobiler Anwendungssysteme für Kundendienstmitarbeiter im Sanitärbereich erläutert und validiert.

### 2.1.5.3 HLB-Entwicklungsmethodik

Innerhalb der HLB-Entwicklungsmethodik (Stark & Müller, 2012) werden ein spezifisches Systemverständnis hybrider Leistungsbündel, ein generisches Vorgehensmodell für den Entwicklungsprozess sowie Methoden für die Planung und -Entwicklung (insbesondere zur Ideenfindung und Anforderungsgenerierung) definiert und bereitgestellt. Eine Implementierungsrichtlinie, welche die Transformation vom Ist- zum Soll-Entwicklungsprozess sowie das benötigte Prozessmanagement beinhaltet, befindet sich in der Erforschung (Nguyen, Müller, & Stark, 2013). Somit liefert die HLB-Entwicklungsmethodik einen Baukasten, der sowohl die generische Betrachtung über ein Vorgehensmodell als auch die operative Ausgestaltung über einen Methodenkatalog erlaubt. Die Grundlage liefert ein Systemverständnis nach den in Abschnitt 2.1.3.3 beschriebenen Kernelementen von Kundenlösungen.



Das generische Vorgehensmodell für den HLB-Entwicklungsprozess ähnelt dem in VDI Richtlinie 2206 (Verein Deutscher Ingenieure, 2004) beschriebenen V-Modell und verbindet Elemente aus der klassischen Produktentwicklung mit Einflüssen aus dem Service Engineering und dem Systems, Software und Requirements Engineering. Das Vorgehensmodell ist in vereinfachter Form in Abbildung 9 ersichtlich. Durch die oberen Ebenen Kund/Markt/Umfeld und Leistungsebene wird die Ausrichtung am Kundennutzen verankert, die für diese Arbeit wichtige Kundenorientierung findet sich im Element Konfiguration wieder. Das Anforderungsmanagement ist wie häufig üblich als Begleitprozess modelliert. Eine detaillierte Beschreibung des Vorgehensmodells findet sich in (Müller & Stark, 2010).

Als Besonderheit finden sich in der HLB-Entwicklungsmethodik Methoden zur operativen Umsetzung einzelner Bestandteile des generischen Vorgehensmodells. So wird mit der HLB-Layer-Methode Entwicklung neuer Ideen, die Klärung der Entwicklungsaufgabe und in Verbindung mit der HLB-Anforderungsscheckliste die Anforderungsgenerierung unterstützt. Die HLB-Anforderungsscheckliste gruppiert über 100 Kriterien zur Anforderungsgenerierung und ermöglicht die Überführung der Anforderungen in eine objekt- und prozessorientierte Sichtweise.

### 2.1.6 Zusammenfassung

Kapitel 2.1 gibt einen Überblick über das Forschungsumfeld der Kundenlösungen als an die Bedürfnisse des Kunden angepasste Leistungsbündel aus Sach- und Dienstleistungsbestandteilen. Neben der historischen Entwicklung und der definitorischen Abgrenzung wurden Nutzenpotentiale für den Wandel zum Lösungsanbieter aufgeführt. Darunter zählen die Möglichkeit der Differenzierung vom Wettbewerb, die Steigerung der Gewinnmarge wie auch die Erhöhung der Kundenbindung. Bei der Beschreibung des Themenumfeldes wird insbesondere die hervorgehobene Stellung des Kunden deutlich, der als prozessauslösendes Element kontinuierlich in die Prozesse des Lösungsanbieters zu integrieren ist. Insbesondere bei den vorgestellten Vorgehensmodellen zur Entwicklung von Kundenlösungen wird die Notwendigkeit einer kontinuierlichen Anforderungserfassung auch über die Entwicklungsphasen hinaus deutlich. Alle Modelle bleiben jedoch detaillierte Beschreibungen zur fortwährenden Erfassung von Kundenanforderungen schuldig. Stattdessen wird, wie auch in der übrigen Literatur, auf bestehende Methoden des klassischen Anforderungsmanagements verwiesen (Hanna & Newman, 2006). Aufgrund dieser Verweise wird im Folgenden näher auf die zitierten Methoden des Anforderungsmanagements für Produkte eingegangen.

## 2.2 Management von Kundenanforderungen

*Die Darstellung des Themengebietes der Kundenlösungen legte die hohe Relevanz und Notwendigkeit des kontinuierlichen Managements von Kundenanforderungen für Lösungsanbieter offen. Im Folgenden soll das Forschungsfeld des Kundenanforderungsmanagements näher beleuchtet und auf Schwachstellen hinsichtlich der Anwendbarkeit im Umfeld von Kundenlösungen hingewiesen werden. Hierzu werden zunächst Grundlagen des allgemeinen Anforderungsmanagements vorgestellt und insbesondere auf das Prozessmodell als Rahmenwerk des Anforderungsmanagements eingegangen. Basierend auf dieser grundlegenden Analyse werden die existierenden Methoden des Anforderungsmanagements kurz vorgestellt und einer kritischen Analyse hinsichtlich ihrer Einsatzfähigkeit im Bereich von Kundenlösungen unterzogen.*

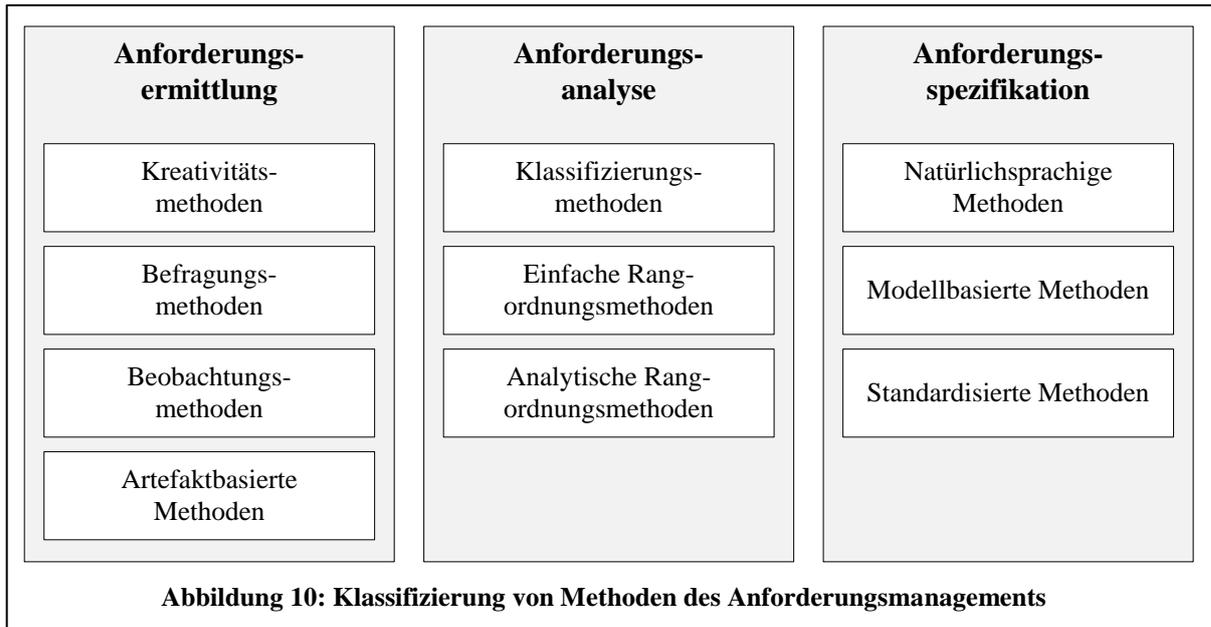
### 2.2.1 Grundlagen

Beim Anforderungsmanagement handelt es sich um einen Prozess, der alle Aktivitäten zur Entwicklung und Dokumentation einer Vielzahl an Anforderungen bei der Entwicklung eines Objektes zusammenfasst (Kotonya & Sommerville, 1998, S. 1f.). Betrachtungsgegenstand des Anforderungsmanagements sind Anforderungen. Für den Begriff der Anforderungen existiert

eine Vielzahl an Definitionen von denen jedoch keine als allgemeingültig betrachtet werden kann. Generell bezieht sich der Begriff der Anforderungen eher darauf, *was* ein System erfüllen soll, und nicht *wie* dies umzusetzen ist (Sommerville & Sawyer, 1997). Im deutschsprachigen Raum ist die Definition nach Rupp weit verbreitet, die eine Anforderung als „Aussage über eine zu erfüllende Eigenschaft oder zu erbringende Leistung eines Produktes, Prozesses oder der am Prozess beteiligten Personen“ definiert (Rupp & SOPHISTen, 2009, S. 158f.). Die Generierung und Dokumentation von Anforderungen ist dabei wie bereits angedeutet in ein Prozessmodell eingebettet, welches als Anforderungsmanagement bezeichnet wird und nach Pohl (Pohl K. , 2007, S. 12) wie folgt definiert:

*"Das Anforderungsmanagement ist ein kooperativer, iterativer, inkrementeller Prozess, dessen Ziel es ist zu gewährleisten, dass alle relevanten Anforderungen bekannt und in dem erforderlichen Detaillierungsgrad verstanden sind, die involvierten Stakeholder eine ausreichende Übereinstimmung über die bekannten Anforderungen erzielen, alle Anforderungen konform zu den Dokumentationsvorschriften dokumentiert bzw. konform zu den Spezifikationsvorschriften spezifiziert sind."*

Innerhalb der Definition können drei einzelne Prozessschritte identifiziert werden, die von Pohl als die drei Hauptaktivitäten des Anforderungsmanagements bezeichnet werden (Pohl K. , 1994, S. 13). Nach Jiao und Chen werden diese drei Prozessschritte Anforderungsermittlung, Anforderungsanalyse sowie Anforderungsspezifikation genannt (Jiao & Chen, 2006, S. 173f.). Zentraler Bezugspunkt des Anforderungsmanagements sind Stakeholder als auslösendes Element. Als Stakeholder können sowohl Personen oder Organisatoren verstanden werden, die einen gewissen Einfluss, direkter oder indirekter Natur, auf die Anforderungen ausüben. Werden die Stakeholder auf die Gruppe der Kunden reduziert, so spricht man vom Kundenanforderungsmanagement. Die Stakeholder bzw. im Kundenanforderungsmanagement der Kunde wirkt insbesondere auf die Anforderungsermittlung, indem er seine Ziele, Bedingungen und Wünsche an das zu entwickelnde Objekt stellt. Dabei verwenden die Kunden unterschiedliche sprachliche Ausdrücke, meist in nicht-fachlicher Form, auf Basis derer die Bedürfnisse, Bedingungen und Anforderungen konkret zu ermitteln und zu verstehen sind (Pohl, Böckle, & van der Linden, 2005, S. 197). Im folgenden Schritt erfolgt die Anforderungsanalyse auf Basis der Ergebnisse der vorgelagerten Ermittlung. Diese beinhaltet eine Verarbeitung und Interpretation der gesammelten Informationen zur Findung eines Konsenses über die ermittelten Anforderungen. Dabei gilt es, Priorisierungen und eine Selektion von differenzierten Anforderungen festzulegen, um eine gemeinsame Übereinstimmung zu erzielen. Die strukturierte Dokumentation aller ermittelten Anforderungen ist Aufgabe der Anforderungsspezifikation (Pohl, Böckle, & van der Linden, 2005, S. 89ff.), welche die Grundlage für die spätere Entwicklung am Produkt bildet (Alexander & Stevens, 2002) (Kovitz, 1998).

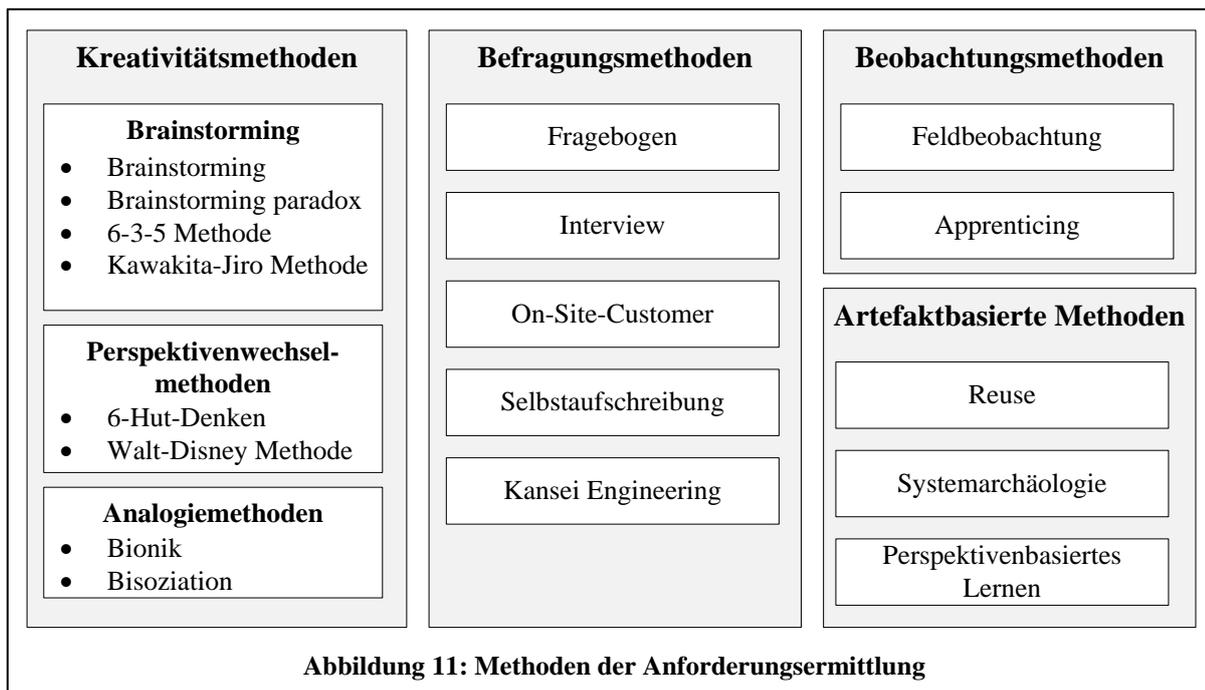


Das Anforderungsmanagement stützt sich auf eine Vielzahl an Methoden, welche die Durchführung des dargestellten dreistufigen Anforderungsprozesses unterstützen. Unter einer Methode wird dabei ein planmäßiges, auf einem Regelsystem beruhendes Vorgehen verstanden, das der Erreichung von vordefinierten Zielen und der Unterstützung der Denk- und Arbeitsweise der jeweiligen Benutzer dient (Ebert, 2009, S. 36f). Methoden liefern Handlungsvorschriften, die ein Ziel auf Basis von gegebenen Bedingungen durch eine festgelegte Schrittfolge erreichen und die anwendungsneutral sein sollen (Chroust, 1992, S. 50).

Die im Anforderungsmanagement bereitgestellten Methoden unterstützen in der Regel nicht den gesamten Prozess des Anforderungsmanagements, sondern fokussieren einzelne der oben vorgestellten Prozessschritte. Die Nutzung einzelner Methoden ist dabei abhängig von Ziel und Anwendungskontext, für welche die Anforderungen erfasst werden sollen. Im Folgenden sollen die Methoden des Anforderungsmanagements zunächst identifiziert und im Anschluss hinsichtlich ihrer Einsetzbarkeit im Kontext der hybriden Wertschöpfung bewertet werden. Die Bewertung richtet sich dabei nach folgenden Kriterien:

- Kundenaufwand bei der Erfassung und Analyse der Anforderungen
- Automatisierbarkeit der Methoden
- Häufigkeit der Anforderungserfassung
- Möglichkeit zur Berücksichtigung der Entwicklung nachgelagerter Phasen

Aufgrund der Vielzahl an existierenden Methoden werden diese in Anlehnung an Zehnter, Burger und Ovtcharova (Zehnter, Burger, & Ovtcharova, 2012, S. 15) in Klassen mit ähnlichen Merkmalsausprägungen eingeteilt und somit die Anzahl an zu untersuchenden Methoden verringert. Eine Übersicht über die Einteilung der Methoden gibt Abbildung 10.



## 2.2.2 Methoden der Anforderungsermittlung

Innerhalb der Anforderungsermittlung artikulieren Kunden deren Ziele, Bedingungen und Wünsche an das zu entwickelnde Objekt. Dabei verwenden die Kunden sprachliche Ausdrücke, meist in nicht-fachlicher Form, auf Basis derer die Bedürfnisse, Bedingungen und Anforderungen konkret zu ermitteln und zu verstehen sind. Die diesen Prozessschritt unterstützende Methoden der Anforderungserfassung werden im Rahmen dieser Arbeit in Kreativitätsmethoden, Befragungsmethoden, Beobachtungsmethoden und artefaktbasierte Methoden untergliedert.

### 2.2.2.1 Analyse von Methoden der Anforderungsermittlung

Der Einsatz von *Kreativitätsmethoden* dient insbesondere der Generierung neuer Ideen und der Entwicklung von Visionen an ein zu entwickelndes Objekt. Es gilt, die Gedanken in verschiedene Richtungen zu lenken, um möglichst eine Vielzahl an differenzierten Ideen zu einer Thematik zu erhalten und das implizite Wissen der Beteiligten zu ermitteln. Aufgrund der Vielzahl an Methoden innerhalb dieser Klasse werden Kreativitätsmethoden weiter untergliedert in Methoden des Brainstormings, Methoden des Perspektivenwechsels, sowie Analogiemethoden.

Ziel des Brainstormings (Schawel & Billing, 2009) sowie dessen Varianten ist die Generierung möglichst vieler Ideen in kurzer vorgegebener Zeit. Unter Methoden des Brainstormings fallen das Brainstorming selbst, sowie dessen Varianten Brainstorming paradox

(Kellner, 2002, S. 85), die 6-3-5 Methode (Rupp & SOPHISTen, 2009, S. 88), sowie die Kawakita-Jiro Methode (Kawakita, 1975).

Methoden des Perspektivenwechsels, wie das 6-Hut-Denken (de Bono, 2000) oder die Walt-Disney-Methode (Dilts, 2003), sind durch die Einnahme verschiedener Extrempositionen gekennzeichnet. Durch das Hineinversetzen in eine vorgegebene Perspektive sollen Teilnehmer einen anderen Blickwinkel erhalten, um neue Ideen entwickeln zu können.

Eine weitere Klasse zur Entwicklung kreativer Ideen stellen Analogiemethoden dar, innerhalb derer aktuelle Problemstellungen und Sachverhalte auf einen anderen Bereich abstrahiert werden. Von besonderer Bedeutung sind die beiden Analogiemethoden Bionik und Bisoziation.

*Befragungsmethoden* dienen der Ermittlung von Anforderungen durch die explizite Erhebung unter einer Gruppe von Kunden oder Stakeholdern. Ziel ist es, das bewusste Wissen der Beteiligten nach außen zu tragen. Unter Befragungsmethoden fallen Fragebögen (Rupp & SOPHISTen, 2009, S. 96), Interviews (Pohl K. , 2007, S. 325), die Methode On-Site-Customer (Beck, 1999) und das Kansei Engineering (Nagamachi, Kansei Engineering, 1989) (Nagamachi, 2011). Befragungsmethoden werden mit einer größeren Gruppe von Nutzer durchgeführt und ermöglichen somit die Akquise einer repräsentativen Masse an Kundenanforderungen.

*Beobachtungsmethoden* kommen zum Einsatz, wenn die Kunden keine Zeit finden, ihr Wissen nach außen zu artikulieren oder nicht die nötigen Fähigkeiten besitzen, um ihre Kenntnisse zu formulieren (Pohl & Rupp, 2009, S. 37). In diesen Fällen wird ein Beobachter hinzugezogen, der durch den Abstand zur Thematik Fehler in Prozessen erkennen, Risiken identifizieren, relevante Abläufe dokumentieren und schließlich daraus Basisanforderungen ableiten kann. Bekannte Methoden stellen die Feldbeobachtung (Bray, 2002) sowie das Apprenticing (Beyer & Holtzblatt, 1995) dar.

Auf der Analyse bereits vorhandener Dokumente, wie beispielweise bestehenden Systemen oder Altsystemen, basieren *Artefaktbasierte Methoden*. Die gesammelten Erfahrungen und Lösungen der zuvor genutzten Systeme werden bei dieser Methode wiederverwendet. Hierbei kann unterbewusst vorhandenes Wissen wiedergewonnen werden – insbesondere die derzeitige Funktionalität des Altsystems (Rupp & SOPHISTen, 2009, S. 99). Zu den artefaktbasierten Methoden gehören die Systemarchäologie (John & Dörr, 2003), der Reuse (Rupp & SOPHISTen, 2009, S. 101) und das Perspektivenbasierte Lesen (Regnell, Runeson, & Thelin, 2000).

### 2.2.2.2 Bewertung von Methoden der Anforderungsermittlung

Ziel der Methoden der Anforderungserfassung ist es, individuelle Meinungsbilder des Kunden in möglichst hoher Detailtiefe zu erfassen und ohne manuellen Aufwand in die Anforderungsanalyse zu überführen. Die Analyse bestehender Methoden der

Anforderungserfassung ergibt jedoch, dass diese für den Kunden entweder mit erhöhtem Mehraufwand verbunden sind oder die Anforderungen einen niedrigen Formalisierungs- und Detaillierungsgrad aufweisen. Insbesondere aufgrund des Mehraufwandes für den Kunden finden die Methoden nur zu fest definierten Meilensteinen, nicht jedoch kontinuierliche Anwendung.

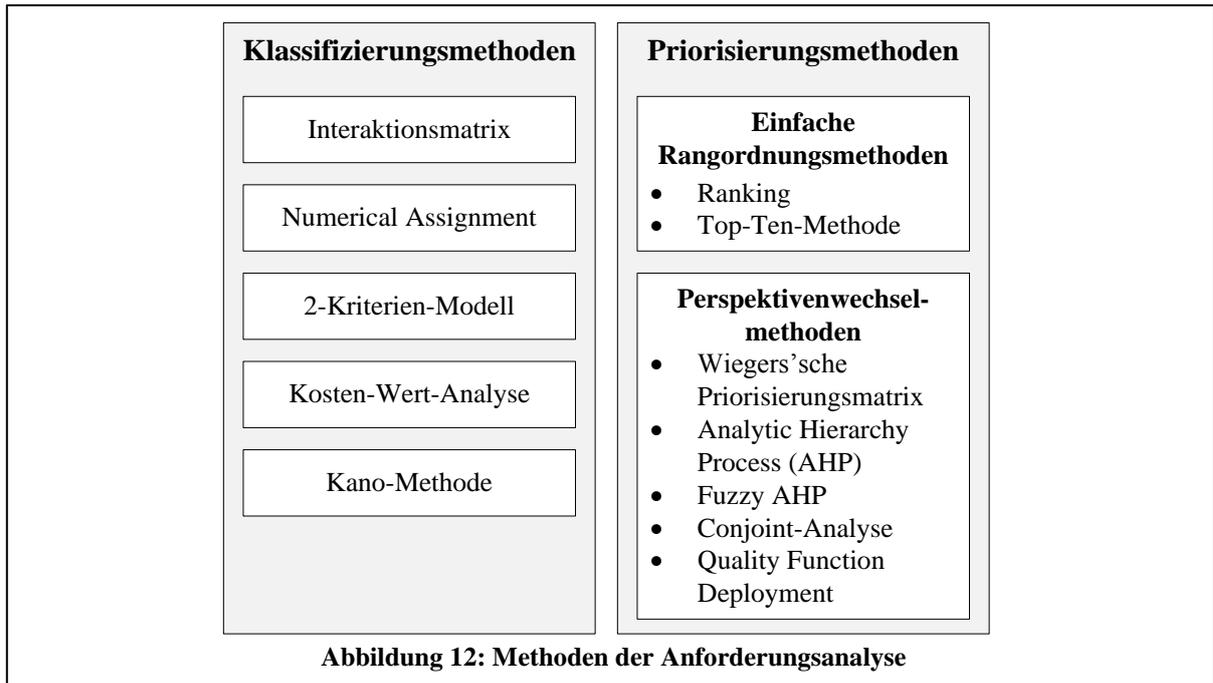
Kreativitätsmethoden erfassen zwar individuelle Meinungsbilder einer repräsentativen Gruppe an Kunden, ergeben jedoch aufgrund des kreativen Ansatzes lediglich unspezifische Ideen, welche in unstrukturierter Form vorliegen und manuell weiterzuverarbeiten sind. Dahingegen erlauben sowohl Beobachtungs- als auch artefaktbasierte Methoden die Identifikation detaillierter Anforderungen. Sie sind jedoch nicht in der Lage, individuelle Meinungsbilder einer repräsentativen Gruppe an Kunden widerzugeben. Bei Beobachtungsmethoden ist dies dem großen Aufwand zur Durchführung der Methode geschuldet, bei artefaktbasierten Methoden dem aggregierten Charakter der zugrundeliegenden Informationen. Befragungsmethoden weisen einen hohen Formalisierungs- und Detaillierungsgrad auf und können daher je nach Aufbau ohne manuellen Aufwand in die Anforderungsspezifikation fließen. Die Befragung ist für die Kunden jedoch mit einem erhöhten Mehraufwand verbunden, so dass die Anwendung in der Regel lediglich auf bestimmte Punkte innerhalb der Entwicklung begrenzt bleibt.

### 2.2.3 Methoden der Anforderungsanalyse

Die Aktivität der Anforderungsanalyse soll zu einer Übereinstimmung führen, indem eine konsolidierte Sicht aller geforderten Anforderungen gebildet wird. Dabei gilt es, Priorisierungen und eine Selektion differenzierter Anforderungen festzulegen, um eine gemeinsame Übereinstimmung zu erzielen. Methoden der Anforderungsanalyse werden in der vorliegenden Arbeit in Klassifizierungsmethoden und Priorisierungsverfahren gegliedert. Innerhalb der Priorisierungsverfahren erfolgt aufgrund der Vielzahl an vorhandenen Methoden eine weitere Unterteilung in einfache und analytische Rangordnungsmethoden.

#### 2.2.3.1 Analyse von Methoden der Anforderungsanalyse

*Klassifizierungsmethoden* dienen, wie der Name bereits andeutet, der Klassifizierung von Anforderungen. Ziel ist es, eine Menge von Handlungsalternativen abzuleiten, die eine einheitliche und systematische Vorgehensweise für alle Anforderungen einer Klasse impliziert. Interaktionsmatrizen (Kotonya & Sommerville, 1998, S. 78ff), das Numerical Assignment (Aurum & Wohlin, 2005), die 2-Kriterien-Klassifikation (Pohl K., 2007, S. 535), die Kosten-Wert-Analyse (Karlsson, Olsson, & Ryan, 1997, S. 53) und das Kano-Modell (Kano, Tsuji, Seraku, & Takahashi, 1984) werden in Anlehnung an Zehnter, Burger und Ovtcharova (Zehnter, Burger, & Ovtcharova, 2012, S. 25) zu den Klassifizierungsmethoden gezählt.



Zu den Priorisierungsverfahren zählende *einfache Rangordnungsmethoden* repräsentieren Verfahren zur Ermittlung einer Ordinalskala sämtlicher zur ermittelten Anforderungen, welche die relative Gewichtung zwischen den einzelnen Elementen nicht berücksichtigt. Zu diesen Methoden werden das Ranking (Pohl K. , 2007, S. 531) sowie die Top-Ten Methode (Pohl & Rupp, 2009, S. 131) gezählt.

*Analytische Rangordnungsmethoden* gehen über die ordinale Zuordnung einfacher Rangordnungsmethoden hinaus und ermöglichen die Generierung eines Kardinalsystems an Anforderungen, welche zusätzlich zur Reihenfolge auch die relative Gewichtung sowie Beziehungen zwischen den Anforderungen erfasst. Aufgrund ihres ausgeprägten analytischen Vorgehens erfordern analytische Rangordnungsverfahren einen großen Erstellungsaufwand. Zu den Methoden werden die Wiegers'sche Priorisierungsmatrix (Wiegers, 2005, S. 236-239), der Analytic Hierarchy Process (AHP) (Saaty, 1980), der Fuzzy Analytic Hierarchy Process (Laarhoven & Pedrycz, 1983), die Conjoint-Analyse (Green & DeSarbo, 1978) sowie das Quality Function Deployment (QFD) (Akao, 1990) gezählt.

### 2.2.3.2 Bewertung von Methoden der Anforderungsanalyse

Ziel der Methoden der Anforderungsanalyse ist es, eine konsistente Sicht der geforderten Anforderungen zu entwerfen und diese nach Bedeutung zu klassifizieren und zu priorisieren. Idealerweise wird im Rahmen des Kundenanforderungsmanagements die Priorisierung vom Kunden selbst vorgenommen, um mögliche Falschbewertungen zu vermeiden. Methoden wie der Analytic Hierarchy Process oder die Conjoint Analyse sind typische Vertreter eines solchen

Vorgehens. Die Erfassung der Informationen ist jedoch für den Kunden mit einem Mehraufwand verbunden, weshalb alternative Ansätze eine Beurteilung basierend auf Expertenwissen ermöglichen - darunter fallen das Quality Function Deployment wie auch die Wiegers'sche Priorisierungsmatrix. Die Ansätze weisen jedoch den Nachteil auf, dass die tatsächlichen Absichten des Kunden missverstanden werden können, was im negativen Fall zu Fehlentwicklungen führt. Unabhängig von der Wahl einer Alternative ist die Durchführung der Anforderungsanalyse mit einem hohen Anteil an manuellem Aufwand verbunden, weshalb die Anforderungsanalyse auf bestimmte Punkte innerhalb der Produktentwicklung begrenzt ist und nicht kontinuierlich vorgenommen wird.

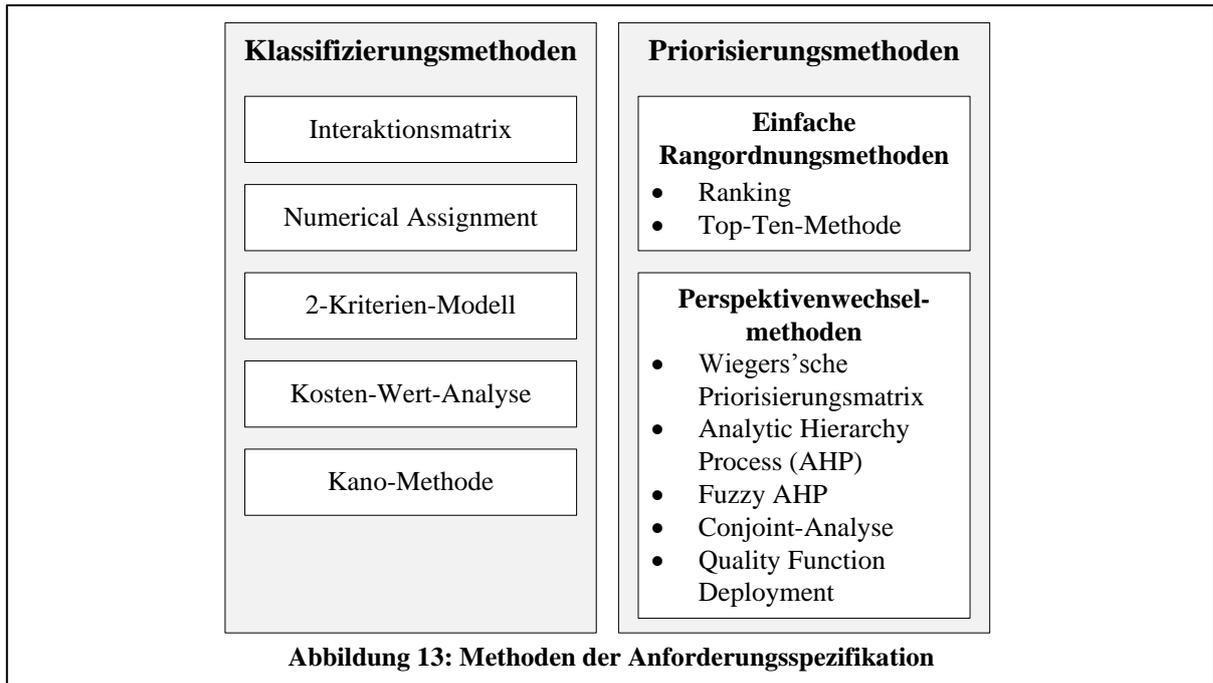
### 2.2.4 Methoden der Anforderungsspezifikation

Innerhalb des Prozessschrittes der Anforderungsspezifikation werden die zuvor ermittelten und analysierten Anforderungen in strukturierter Form dokumentiert, um als Grundlage der späteren Entwicklung dienen zu können. Methoden der Anforderungsspezifikation können in modellbasierte Methoden, natürlichsprachige Methoden sowie standardisierte Methoden untergliedert werden.

#### 2.2.4.1 Analyse von Methoden der Anforderungsspezifikation

*Modellbasierte Methoden* beruhen auf einer Dokumentation der Anforderungen durch die Erstellung eines Modells, das ein abstrahiertes Abbild der vorhandenen Realität oder der zu entwickelnden Realität darstellt (Pohl & Rupp, 2009, S. 68). Das Vorhandensein einer vorgegeben Syntax und Semantik in Kombination mit natürlicher Sprache ist kennzeichnend für die modellbasierten Methoden. Um die ermittelten, priorisierten und konsistenten Anforderungen in eine wiederverwendbare und strukturierte Dokumentation zu transformieren sind zunächst Daten, Struktur, Funktionen und Verhalten der zu entwickelnden Lösung zu analysieren. Modellbasierte Methoden können aufgrund der existierenden Methodenvielfalt weiter untergliedert werden in Daten- und Strukturmodelle sowie Verhaltensmodelle bzw. Prozessmodelle.

- Daten- und Strukturmodelle spezifizieren zuvor geäußerte Anforderungen, indem sie die zu entwickelnde Realität durch Objekte und zwischen den Objekten wirkende Beziehungen modellieren. Die Modelle lassen dabei nur Rückschlüsse auf die statische Struktur der Realität zu. Zu den gängigsten Modellen zählen das Entity-Relationship-Modeling (Chen, 1976) sowie UML Klassendiagramme.
- Prozessmodelle oder auch Verhaltensmodelle dienen der Modellierung des dynamischen Verhaltens des zu entwickelnden Systems. Zu den Prozessmodellen zählen UML



Aktivitätsdiagramme, Petrinetze, Ereignisgesteuerte Prozessketten (Nüttgens & Rupp, 2002, S. 64), Statecharts (Pohl & Rupp, 2009, S. 92), Datenflussmodelle (DeMarco, 1979, S. 51) sowie Entscheidungstabellen und Entscheidungsbäume (Davis, 1990, S. 223ff).

*Natürlichsprachige Methoden* der Anforderungsspezifikation werden durch den Einsatz der natürlichen Sprache ohne den Gebrauch eines vordefinierten Satzes an Notationen gekennzeichnet. Dadurch weisen sie zwar einen geringen Erstellungsaufwand, jedoch auch einen schwachen Formalitätsgrad auf. Zu den natürlichsprachigen Methoden zählen Prosa (Rupp & SOPHISTen, 2009, S. 115 ff.), die Satzschablone (Pohl & Rupp, 2009, S. 61ff) sowie der Glossar (Pohl K., 2007, S. 244).

Typisches Kennzeichen *standardisierter Methoden* ist deren vordefinierte oder standardisierte Gliederung der Anforderungsspezifikation, welche die Grobstruktur der zu dokumentierenden Anforderungen darstellt (Pohl & Rupp, 2009, S. 47). Zu den standardisierten Methoden zählen die Software Requirements Specification (SRS) nach dem IEEE Standard 830-1998 (IEEE, 1998) sowie das Volere Requirements Specification Template (Robertson & Robertson, 2010).

#### 2.2.4.2 Bewertung von Methoden der Anforderungsspezifikation

Methoden der Anforderungsspezifikation haben zum Ziel, die konsistenten Anforderungen aus der Anforderungsanalyse strukturiert zu dokumentieren, um als Basis der nachgelagerten Entwicklung dienen zu können. Wie auch bei Methoden der Anforderungsanalyse überwiegen manuelle Abläufe. Der manuelle Aufwand ist dabei umso höher, je detaillierter und formalisierter die Spezifikation vorgenommen wird. Insbesondere modellbasierte und

standardisierte Methoden, die häufig in Verbindung miteinander verwendet werden, weisen einen hohen Aufwand auf. In der Regel werden daher auch Anforderungsspezifikationen nicht kontinuierlich weitergepflegt und sind auf die Entwicklungsphase beschränkt.

### 2.2.5 Integrierte Ansätze

Integrierte Ansätze des Anforderungsmanagements mit Fokussierung der Nutzungsphase von Produkten und Dienstleistungen finden sich selten. Rzehorz (Rzehorz, 1998) liefert eine erste grundlegende Beschreibung der Thematik, indem er die Erfassung von Erfahrungswissen aus der Nutzungsphase und deren Integration in den Konstruktionsprozess fordert. Produktzentrierte Ansätze werden von Jungk (Jungk, 2007) und Neubach (Neubach, 2010) erläutert. Sie beschreiben jeweils einen Prozess zur Erfassung von Produktnutzungsinformationen, die ihrerseits Erkenntnisse zur Weiterentwicklung der Produkte liefern. Einen kundenzentrierten Ansatz beschreibt Schulte (Schulte, 2007), der das Erfahrungswissen des Kunden als Bezugsquelle für das Anforderungsmanagement sieht. Hierfür erweitert er den Ansatz des Quality Function Deployment. Einen gesamtheitlichen Ansatz zur Erfassung von Kundenanforderungen über den Lebenslauf einer individuellen Kundenbeziehung liefern die Ergebnisse des BMBF-Verbundprojekts DIALOG (Franke, Grein, & Türck, 2011). Über ein zentrales Feedback-Bezugsobjekt (FBO) wird die Wirkung, Nutzung und der Zustand des Produktes beim Kunden kontinuierlich erfasst. Insbesondere wird ein Produktkonfigurator bereitgestellt, der auf dem Ansatz des fallbasierten Schließens beruht und zur Vorschlagsgenerierung genutzt wird (Schubert, Rogalski, & Ovtcharova, 2011, S. 43). Wie auch die vorliegende Arbeit ist es das erklärte Ziel, Kunden- und Anbietersicht miteinander zu verknüpfen (Schubert, Wicaksono, & Rogalski, 2011, S. 357f.). Die vorliegende Arbeit erweitert den Fokus bestehender Arbeiten und geht im Besonderen auf Spezifika im Bereich der Kundenlösungen ein.

### 2.2.6 Zusammenfassung

Die Durchführung von bestehenden Methoden des Anforderungsmanagements ist sowohl für den Kunden als auch für das betroffene Unternehmen mit hohem zusätzlichem Aufwand verbunden. In Bezug auf den Kunden sind die Methoden nicht darauf ausgelegt, bestehende Interaktionspunkte zwischen Kunde und Unternehmen zu nutzen. Stattdessen werden neue, für den Kunden mit Mehraufwand verbundene Interaktionspunkte definiert, was zu Verzerrungen der Objektivität der Aussagen führen kann. Für die betroffenen Unternehmen sind insbesondere die Methoden der Anforderungsanalyse als auch der Anforderungsspezifikation mit einem hohen manuellen Aufwand verbunden. Unüberwachte Methoden, welche eine automatisierte

Rückführung der Anforderungen insbesondere in der Nutzungsphase der Produkte erlauben, existieren in der klassischen Literatur nicht. Als Folge des Mehraufwands bleibt die Durchführung der Methoden auf bestimmte Punkte innerhalb der Entwicklungsphase begrenzt. Die Erfassung der Kundenanforderungen erfolgt somit zu fest definierten Meilensteinen. Eine fortlaufende Kontrolle der Anforderungen, insbesondere in den der Entwicklung nachgelagerten Phasen, findet bedingt durch den hohen Akquise- und Verarbeitungsaufwand nicht statt.

Bislang getätigte Forschung im Themenumfeld von Kundenlösungen begnügte sich damit, auf herkömmliche Methoden des Anforderungsmanagements für die Produktentwicklung zu verweisen (Hanna & Newman, 2006). Sämtliche der zuvor untersuchten Methoden entstammen daher der herkömmlichen Produktentwicklung. Da die hybride Wertschöpfung wie gezeigt spezifische Charakteristika aufweist, die sich vom reinen Sachleistungsbereich unterscheiden, sind die zugrundeliegenden Methoden des Anforderungsmanagements entsprechend anzupassen.

Der somit identifizierte wissenschaftliche Handlungsbedarf wird in der Folge auf dessen praktischen Nutzen hin validiert. Ziel ist es, die industrielle Relevanz der Thematik explizit herauszustellen.

## 2.3 Industrieller Handlungsbedarf

*Im Rahmen der vorliegenden Arbeit wurde eine empirische Marktuntersuchung durchgeführt, innerhalb derer die Relevanz des identifizierten wissenschaftlichen Handlungsbedarfs unter industriellen Gesichtspunkten validiert und ein industrieller Handlungsbedarf abgeleitet wurde. Im Speziellen wird die aktuelle Entwicklung des Marktes für Kundenlösungen analysiert, um dessen Relevanz im Gesamtmarkt einschätzen zu können. Zum anderen werden die Identifikation und Einarbeitung von Kundenbedürfnissen in die Prozesse von Anbietern kritisch beleuchtet und Schwachstellen aufgezeigt.*

### 2.3.1 Grundlagen

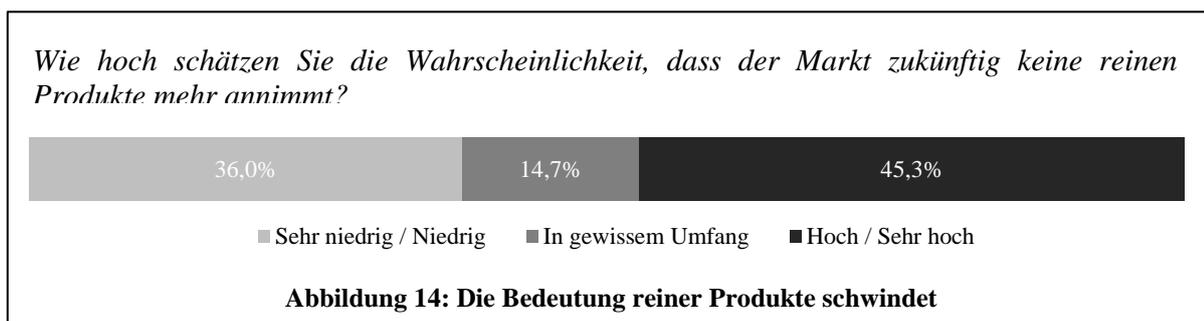
Die Studie von Burger et al. (Burger, Ovtcharova, & Brenner, 2012) wurde als branchenübergreifende Onlineumfrage durchgeführt, an der 120 führende Industrieunternehmen aus dem deutschsprachigen Raum teilnahmen. Zur Abbildung des industriellen Status Quo wurden sechs Gestaltungsfelder identifiziert, welche den Wandel zum Lösungsanbieter in seiner Gänze erfassen und somit greifbar und quantifizierbar machen. Auf strategischer Ebene geben die Gestaltungsfelder Markt, Management und Kunde Informationen über die übergeordnete Ausrichtung eines Unternehmens (Ovtcharova, Burger, & Kilger, 2012, S. 2) (Burger & Ovtcharova, 2011). Die strategische Ausrichtung ist in der Folge auf operativer Ebene zu

verankern. Diese wird in der Studie durch die beiden Gestaltungsfelder Organisation und Prozesse charakterisiert. Auf unterster Ebene findet sich die IT und andere Enabling Technologies. Diese unterstützen die operativen Prozesse mithilfe geeigneter Systeme und stellen das letzte der sechs Gestaltungsfelder dar.

Von den befragten Studienteilnehmern entstammen 85 Prozent dem Führungskreis ihres Unternehmens. Die Studie vermittelt somit ein repräsentatives Bild heutiger und zukünftiger Entscheider von Industrieunternehmen. Insbesondere fällt auf, dass die Teilnehmer zum überwiegenden Teil Großunternehmen mit mehr als 1.000 Mitarbeitern angehören. Die Thematik der Kundenlösungen spricht demnach eher Konzerne an, welche die Lieferkette anführen oder eine vordere Position innerhalb der Lieferkette bekleiden: Knapp 80 Prozent der befragten Teilnehmer arbeiten für einen OEM oder einen 1st Tier Supplier. Die beteiligten Unternehmen repräsentieren ein weit gefächertes Branchenspektrum wie den Maschinen- und Anlagenbau, die Automobilindustrie, den High Tech und Elektronik Bereich, die Nahrungsmittelindustrie, die Energie- und Umwelttechnik oder auch die chemischen Industrie. Diese Ergebnisse belegen, dass das Interesse an der Thematik der Kundenlösungen ein breites Kundenspektrum anspricht und nicht auf spezifische Industriezweige beschränkt ist.

### 2.3.2 Analyse des Bedarfs an Kundenlösungen

Innerhalb der Studie wurden die Teilnehmer befragt, wie sich ihre Unternehmen zum aktuellen Zeitpunkt maßgeblich vom Wettbewerb differenzieren. Dabei zeigte sich, dass innovative Produkteigenschaften die zentralen Differenzierungspunkte im Vergleich zum Wettbewerb darstellen. Während mehr als 63 Prozent der befragten Unternehmen ihre Produkte vollständig oder zumindest überwiegend als unternehmerisches Alleinstellungsmerkmal identifizieren, sind es bei Kundenlösungen lediglich rund 32 Prozent. Zunehmender Wettbewerbsdruck, insbesondere durch Niedriglohnländer und die Homogenisierung der Märkte erschweren eine Differenzierung rein über Produktfunktionalitäten jedoch immer mehr. Die Studienteilnehmer sind sich dieser Situation bewusst: Knapp die Hälfte der Studienteilnehmer stuft die Wahrscheinlichkeit, dass der Markt in Zukunft keine reinen Sachleistungen mehr annimmt, als hoch bis sehr hoch ein. Abbildung 14 stellt den Sachverhalt grafisch dar.



Das Ziel eines nachhaltig wirtschaftenden Unternehmens muss es daher sein, alternative Wege zur Differenzierung vom Wettbewerb zu finden. Bedingt durch eine steigende Marktmacht der Kundenseite ist dabei insbesondere auf deren Bedürfnisse sowie auf die Integration dieser Bedürfnisse in das Leistungsportfolio des Anbieters zu achten. Dabei spielt weniger die Art der Umsetzung eine Rolle als vielmehr die Tatsache, dass die individuellen Bedürfnisse einzelner Kunden befriedigt werden. Die Antworten der Studienteilnehmer untermauern diese These: 72,5% aller Studienteilnehmer gehen davon aus, dass die Kunden zukünftig in verstärktem Maße individuelle Kundenlösungen fordern werden (siehe Abbildung 15). Die Unternehmen reagieren auf diese Forderung, indem sie den signifikanten Ausbau des Dienstleistungsanteils am Leistungsportfolio planen. Mehr als die Hälfte der befragten Unternehmen haben diesbezüglich

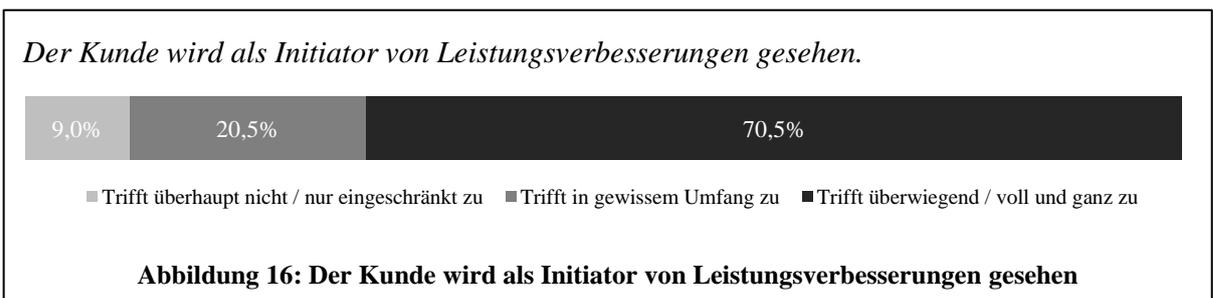


bereits Anstrengungen unternommen. Mit dem Ausbau des Dienstleistungsportfolios und dessen enger Verzahnung mit dem bestehenden Produktspektrum sind die Weichen für das Angebot von Kundenlösungen gestellt.

Die Wachstumsraten des Marktes für Kundenlösungen schätzen dabei knapp 99 Prozent der Studienteilnehmer als positiv ein. Über ein Viertel der befragten Unternehmen prognostizieren die jährlichen Wachstumsraten sogar im zweistelligen Bereich. Das Potential von Kundenlösungen sowie deren Möglichkeit der Differenzierung wurde also erkannt.

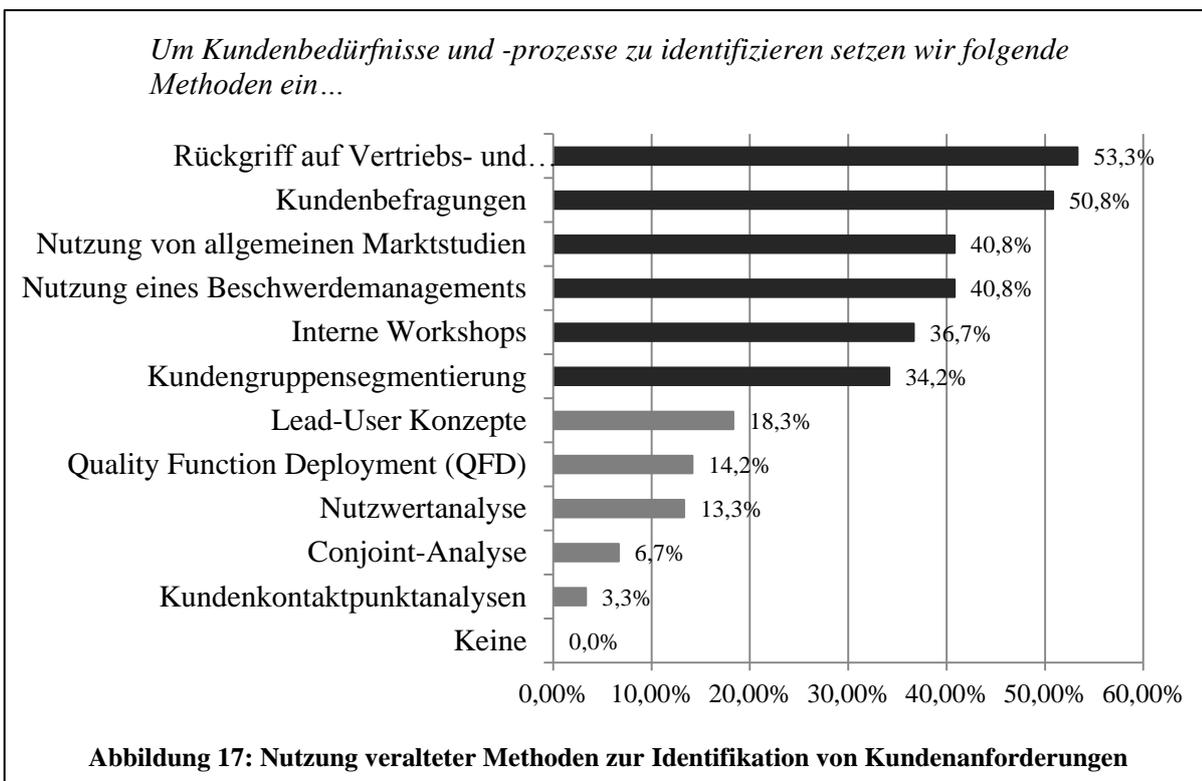
### 2.3.3 Analyse der Integration von Kundenbedürfnissen

Kundenlösungen orientieren sich per Definition an der Lösung individueller Kundenprobleme und setzen den Kunden in den Mittelpunkt sämtlicher Aktivitäten. Die Existenz eines Lösungsanbieters beruht daher auf der Fähigkeit, die Bedürfnisse seiner Zielgruppe zu kennen



und diese in Form von Leistungen befriedigen zu können. Ausgangspunkt jeglicher unternehmerischer Aktivitäten ist die Kenntnis der Bedürfnisse der Kunden. Diese Auffassung wird von über 70 Prozent der befragten Unternehmen geteilt, wie Abbildung 16 verdeutlicht. Im Rahmen der Global CEO Study von IBM (IBM Deutschland GmbH, 2010) wird die enge Ausrichtung des Unternehmens am Kunden ebenfalls deutlich: 88 Prozent der befragten CEOs sehen in einer Intensivierung der Kundenbeziehungen die wichtigste strategische Stoßrichtung der nächsten fünf Jahre.

Innerhalb der Studie wurde zudem darauf eingegangen, welche Methoden zur Erfassung und Einarbeitung von Kundenbedürfnissen in das Leistungsspektrum des Anbieters angewendet werden. Bei der Integration von Anforderungen und Bedürfnissen des Kunden zeigte sich deutlich das Vorherrschen altbewährter und wenig strukturierter Methoden wie Kundenbefragungen oder die Nutzung allgemeiner Marktstudien. Abbildung 17 veranschaulicht den Sachverhalt grafisch. Um eine systematische Analyse einzelner Kundenanforderungen zu ermöglichen, sind klassische Methoden jedoch zunehmend durch fortschrittlichere Methoden zu ersetzen. Somit wird eine vereinfachte Überführung von individuellen Anforderungen in individuelle Leistungen sichergestellt. Solche Methoden der Anforderungsintegration werden jedoch nur unzureichend verwendet.



Im Zusammenhang mit der Identifikation von Kundenbedürfnissen können Systeme zur Konfiguration individueller Leistungsbündel zum Tragen kommen, da diese an der Schnittstelle

zwischen Kunde und Anbieter agieren. Der Einsatz solcher Systeme wird von den Studienteilnehmern jedoch zu knapp 50 Prozent als sehr niedrig bis niedrig eingestuft.

### 2.3.4 Zusammenfassung

Die Studie untermauert die auf wissenschaftlicher Seite identifizierten Handlungsbedarfe unter praktischen Gesichtspunkten. Insbesondere wird deutlich, dass der Markt für Kundenlösungen als attraktiver Wachstumsmarkt angesehen wird. Hinsichtlich der Integration von Kundenanforderungen zeigte sich das Vorherrschen altbewährter und wenig strukturierter Methoden, die von den Unternehmen als unzureichend bezeichnet werden.

## 2.4 Ableitung des Handlungsbedarfs

Basierend auf den im Rahmen dieses Kapitels beschriebenen Grundlagen konnte ein wissenschaftlicher Handlungsbedarf abgeleitet werden, der in der Folge unter industriellen Gesichtspunkten bestätigt wurde. Der somit validierte Handlungsbedarf dient als Motivation und Grundlage dieser Arbeit und soll im Folgenden im Detail diskutiert werden.

Bei der Darstellung des Themengebietes Kundenlösungen wurde insbesondere die hervorgehobene Stellung der Kunden deutlich, welche kein Interesse nach Produkten oder Dienstleistungen per se haben, sondern vielmehr die Lösung einer Problemstellung oder eines Bedürfnisses erwarten. Auch die dargestellten Vorgehensmodelle zur Entwicklung von Kundenlösungen betonen die Wichtigkeit der Kundenintegration, insbesondere der kontinuierlichen Anforderungserfassung, auch über die Entwicklungsprozesse hinaus. Der hervorgehobenen und im Vergleich zur klassischen Produktentwicklung geänderten Stellung des Kunden werden bestehende Ansätze des Kundenanforderungsmanagements jedoch nicht gerecht. Bei der genauen Analyse ergab sich, dass existierende Methoden des Kundenanforderungsmanagements spezifisch auf die Entwicklung von Produkten ausgelegt sind. Als die schwerwiegendsten Nachteile wurden die Begrenzung der Methoden auf bestimmte Punkte innerhalb der Produktentwicklung und damit die Vernachlässigung der kontinuierlichen Erfassung von Kundenanforderungen identifiziert. Insbesondere werden die der Entwicklung nachgelagerten Phasen nicht berücksichtigt und manuelle Prozesse fokussiert. Der aus wissenschaftlicher Sicht identifizierte Handlungsbedarf wurde in der Folge auf Basis der Ergebnisse einer Marktstudie auf dessen industrielle Notwendigkeit hin überprüft. Grundlage der Überprüfung unter industriellen Gesichtspunkten war eine empirische Studie, an der 120 führende Industrieunternehmen aus dem deutschsprachigen Raum teilnahmen. Die Studie untermauerte den zuvor identifizierten wissenschaftlichen Handlungsbedarf.

Der dieser Arbeit zugrundeliegende Handlungsbedarf ergibt sich damit einerseits zur Notwendigkeit an Ansätzen zur Kundenintegration, die sich weniger am Produkt als vielmehr am Kundennutzen ausrichten. Im Rahmen der vorliegenden Arbeit soll dies durch die Konzeption eines transparenten Lösungskonfigurators erfolgen, der eine Konfiguration basierend auf Kundennutzen und Kundenbedürfnissen anstelle von Produktmerkmalen erlaubt. Dieser Handlungsbedarf wird als erstes Hauptziel der Arbeit *H1* bezeichnet. Als Unterziele wurden folgende Punkte abgeleitet:

*H1 Konzeption eines transparenten Lösungskonfigurators, der eine Konfiguration basierend auf Kundennutzen und Kundenbedürfnissen anstelle von Produktmerkmalen erlaubt*

*U1-1 Berücksichtigung natürlicher menschlicher Unschärfe bei der Verknüpfung von Kunden- und Anbietersicht*

Menschliche Kognition beruht nicht nur auf zwei Wahrheitswerten, sondern weißt in der Regel ein hohes Maß an Unschärfe und Vagheit auf (siehe auch (Sakao, Panschef, & Dörsam, 2009, S. 140)). Um ein optimales Verständnis zwischen Kunde und Anbieter sicherzustellen, ist natürliche Unschärfe in umgangs-sprachlichen Beschreibungen bei der Integration des Kunden zu berücksichtigen.

*U1-2 Erkennen impliziten Wissens innerhalb der Kundenaussagen*

Menschliche Ausdrucksweise beinhaltet neben expliziten auch implizite Informationen. Diese sind in einem Sachverhalt eingeschlossen und können durch logisches Schlussfolgern explizit gemacht werden. Die zu entwickelnde Methodik soll die Bedeutung der Kundenaussagen ermitteln und somit auf implizites Wissen zugreifen können.

*U1-3 Bereitstellung eines Vorgehensrahmens zur Unterstützung des Lösungsanbieters bei der Einführung der Methodik im Unternehmen*

Um die Methodik strukturiert formalisieren und im Unternehmen einführen zu können, bedarf es eines Vorgehensrahmens, der die beteiligten Akteure darin unterstützt, den Wissensaustausch zwischen Menschen und Informationssystemen zu ermöglichen.

*U1-4 Sicherstellung der Nutzbarkeit unter Gesichtspunkten der Gebrauchstauglichkeit (Usability)*

Die zu entwickelnde Methodik richtet sich am Nutzen des Kunden (siehe *U2-1*) aus. Nicht nur die methodische Basis, sondern das Gesamtsystem bestehend aus Mensch, Werkzeug und Aufgabe hat jedoch den Bedürfnissen des Benutzers entsprechend

gestaltet zu sein, um das Wohlbefinden des Menschen und die Leistung des Gesamtsystems zu optimieren.

Als zweiter übergeordneter Handlungsbedarf *H2* wurde die Notwendigkeit einer kontinuierlichen, automatisierten Erfassung von Kundenanforderungen für die Entwicklung von Kundenlösungen identifiziert, ohne dabei einen Mehraufwand für den Kunden zu generieren. Im Rahmen der vorliegenden Arbeit ist dies durch eine kontinuierliche Rückführung von Kundenanforderungen aus der Entwicklung nachgelagerten Phasen auf Basis des zuvor geforderten Konfigurators als kontinuierliche Bezugsquelle sicherzustellen. Als Nebenbedingungen wurden folgende Unterziele definiert:

*H2*      *Bedarf einer kontinuierlichen, automatisierten Erfassung von Kundenanforderungen für die Entwicklung von Kundenlösungen, ohne einen Mehraufwand für den Kunden zu generieren*

*U2-1*    *Ausrichtung auf die Charakteristika von Kundenlösungen*

Kunden haben kein Interesse nach Produkten oder Dienstleistungen per se, sondern erwarten vielmehr die Lösung einer Problemstellung oder eines Bedürfnisses. Die zu entwickelnde Methodik hat diesem Umstand Rechnung zu tragen und ist an den Anforderungen und Bedürfnissen des Kunden auszurichten.

*U2-2*    *Nutzung bestehender Interaktionspunkte mit dem Kunden zur Vermeidung von Mehraufwand*

Existierende Ansätze des Managements von Kundenanforderungen nutzen keine bestehenden Interaktionspunkte zwischen Kunde und Unternehmen. Stattdessen werden neue, für den Kunden mit Mehraufwand verbundene Interaktionspunkte definiert, was zu Verzerrungen der Objektivität der Aussagen führt.

*U2-3*    *Unüberwachte Analyse des Lösungsportfolios hinsichtlich der Abdeckung von Markttrends und Qualitätsbelangen*

Die zu entwickelnde Methodik muss einen Ansatz zum Management von Kundenanforderungen bereitstellen, der eine automatisierte Rückführung der Anforderungen insbesondere in der Nutzungsphase der Leistungen ermöglicht. Diese existieren in der klassischen Literatur nicht. Durch Minimierung des manuellen Aufwands wird eine fortlaufende Kontrolle von Kundenanforderungen ermöglicht.

Zur Zielerreichung werden bestehende Ansätze aus verschiedenen Forschungsbereichen, welche im Zusammenhang mit einem der aufgestellten Haupt- oder Unterziele stehen,

identifiziert und im folgenden Kapitel vorgestellt und analysiert. Die Analyse soll Potentiale aber auch Schwächen bestehender Ansätze aufzeigen und als Grundlage zur Spezifikation des Konzeptteils dieser Arbeit dienen.

### **3 Analyse ausgewählter Ansätze in Wissenschaft und Technik**

Das vorliegende Kapitel liefert eine Analyse existierender Methoden und Konzepte in Bezug auf die in Abschnitt 2.4 ‚Ableitung des Handlungsbedarfs‘ definierten Anforderungen. Es wird analysiert, auf welchen Ansätzen aus Wissenschaft und Technik die zu entwickelnde Methodik ‚Design for Customer‘ aufbauen kann, um die Ziele der Arbeit bestmöglich zu erreichen. Im zunächst folgenden Abschnitt 3.1 wird auf ausgewählte Methoden der Künstlichen Intelligenz eingegangen, welche in der Lage sind, menschliche Unschärfe bei der Verknüpfung von Kunden- und Anbietersicht zu berücksichtigen. Darauf aufbauend werden in Abschnitt 3.2 semantische Technologien erläutert, welche implizites Wissen in Kundenaussagen erkennen und daraus Schlussfolgerungen ableiten können. Bestehende Vorgehensmodelle zur Bereitstellung eines Vorgehensrahmens werden ebenfalls in diesem Abschnitt erläutert. Abschnitt 3.3 liefert eine Übersicht über bestehende Verfahren der Assoziationsanalyse aus dem Bereich des Knowledge Discovery in Databases, um aus den generierten Kundendaten Anforderungen ableiten zu können. Eine abschließende Bewertung der analysierten Ansätze wird in Abschnitt 3.4 vorgenommen. Darin werden die analysierten Methoden hinsichtlich ihres Zielerreichungsgrades bezüglich des identifizierten Handlungsbedarfs übersichtlich zusammengefasst und somit auf den Konzeptteil übergeleitet.

#### **3.1 Ausgewählte Methoden der Künstlichen Intelligenz**

*Im vorliegenden Unterkapitel sollen ausgewählte Methoden der Künstlichen Intelligenz dahingehend untersucht werden, inwiefern sie eine Verknüpfung von Kunden- und Anbietersicht erlauben und dabei natürlichsprachige menschliche Unschärfe berücksichtigen. Hierfür werden zunächst auf dem probabilistischen Schließen beruhende Bayes'sche Netze vorgestellt und analysiert. In der Folge werden sowohl Künstliche Neuronale Netze als auch Fuzzy-Systeme evaluiert, die der Computational Intelligence als Untermenge der Künstlichen Intelligenz entspringen.*

Die Künstliche Intelligenz (KI) als Informatikdisziplin beschäftigt sich mit der Modellierung intelligenter Maschinen und der Nachahmung menschlichen Verhaltens am Computer (Görz, Rollinger, & Schneeberger, 2003, S. 1). Die Definition des Begriffes ist jedoch insofern schwierig, da es keine allgemein anerkannte Definition von Intelligenz gibt (Zöller-Greer, 2010, S. 5). Auf eine detaillierte Behandlung der Definition wird im Rahmen der vorliegenden Arbeit jedoch verzichtet und stattdessen auf die einschlägige Literatur verwiesen.

#### 3.1.1 Bayes'sche Netze

*Im Rahmen dieses Abschnittes werden Bayes'sche Netze als graphische Modelle zur Repräsentation probabilistischer Zusammenhänge vorgestellt. Hierbei wird zunächst auf die zugrundeliegende Bayes-Statistik eingegangen, bevor Bayes'sche Netze sowie Schlussfolgerungsprozesse in Bayes'schen Netzen erläutert werden. Abschließend wird eine kritische Analyse des Ansatzes in Bezug auf das vorliegende Forschungsvorhaben durchgeführt.*

Während sich die traditionelle Statistik auf Wahrscheinlichkeiten zufälliger Ereignisse beschränkt, erweitert die Bayes-Statistik den Begriff der Wahrscheinlichkeit, indem sie diesen auf generelle Aussagen ausdehnt. Grundlage der gesamten Bayes-Statistik und damit auch Bayes'scher Netze bildet das Bayes Theorem. Mit dessen Hilfe lassen sich unbekannte Parameter schätzen, Konfidenzregionen für die unbekannt Parameter angeben sowie Hypothesen für die Parameter prüfen (Koch, 1999, S. 253). Formal dargestellt lautet das Bayes Theorem:

$$P(A|BC) = \frac{P(A|C)P(B|AC)}{P(B|C)} \quad [3.1]$$

Dabei wird unter  $A$  eine Aussage über ein unbekanntes Phänomen verstanden.  $B$  ist eine Aussage, die Informationen über das unbekannt Phänomen enthält und  $C$  eine Aussage über zusätzliches Wissen (Koch, 2000, S. 14).  $P(A|C)$  wird als Priori-Wahrscheinlichkeit bezeichnet,  $P(A|BC)$  als Posteriori-Wahrscheinlichkeit und  $P(B|AC)$  als Likelihood (Koch, 1999, S. 254). Um zur Posteriori-Wahrscheinlichkeit über das unbekannt Phänomen zu gelangen wird die Priori-Wahrscheinlichkeit über die Likelihood-Wahrscheinlichkeit modifiziert.

Bayes'sche Netze bauen auf dem beschriebenen Bayes Theorem auf und stellen einen Ansatz zur Repräsentation probabilistischer Zusammenhänge mithilfe graphischer Modelle dar. Zur Darstellung von Beziehungen und Annahmen bedienen sich die Bayes'sche Netze gerichteter Graphen aus der Graphentheorie sowie weiterer Methoden der klassischen Wahrscheinlichkeitstheorie, um Unsicherheiten in die Netze zu integrieren. Auf dieser Basis können komplexe Modelle dargestellt werden, die sowohl Unvollständigkeiten als Unsicherheiten enthalten. Zur Mathematisierung des beschriebenen Sachverhaltes werden Bayes'sche Netze in Anlehnung an (Wittig, 2002, S. 15) wie folgt definiert:

**Definition 3.1 (Bayes'sches Netz):** Ein Bayes'sches Netz  $B = (G, \theta)$  für eine Menge  $X = \{X_1, \dots, X_n\}$  von Zufallsvariablen besteht aus zwei Teilen:

1. Einem gerichteten azyklischen Graphen  $G = (X, E)$ , dessen Knoten den Zufallsvariablen entsprechen und mit dessen Kanten die bedingten Wahrscheinlichkeiten zwischen den Variablen kodiert werden. Man spricht von  $G$  als der Struktur von  $B$ .

2. Einer Menge  $\theta = \{\theta_1, \dots, \theta_n\}$  von mit den Knoten assoziierten Tabellen bedingter Wahrscheinlichkeiten  $\theta_i = P(X_i | Pa(X_i))$  mit  $i=1, \dots, n$ . Sie beinhalten als Einträge die bedingten Wahrscheinlichkeiten  $\theta_{ijk} = P(x_{ij} | pa_k(X_i))$  der  $n_i$  Zustände  $x_{ij}$ ,  $j=1, \dots, n_i$ , der Variablen  $X_i$  konditioniert auf die möglichen Zustandskombinationen  $pa(X_i)$  der Eltern  $Pa(X_i)$ . Mit  $pa_k(X_i)$  wird die  $k$ -te der Zustandskombination  $pa(X_i)$  der Eltern bezeichnet. Falls  $pa(X_i) = \emptyset$ , so ist diese Tabelle auf  $P(X_i)$  reduziert.

Dabei gilt, dass eine Zufallsvariable  $X_i$  bedingt unabhängig von ihren Nicht-Nachfolgern innerhalb der Struktur des Bayes'schen Netzes ist, falls alle Zustände der Eltern  $Pa(X_i)$  bekannt sind. Dies bedeutet, dass die Wahrscheinlichkeit von  $X_i$  deren Nicht-Nachfolger nicht beeinflusst.

Ein Bayes'sches Netz entspricht dabei nicht einem statischen Gebilde. Über Inferenzalgorithmen können sich Informationen durch ein Netz fortpflanzen und somit Wahrscheinlichkeitswerte erlernen. Dieser Vorgang wird als Propagation bezeichnet. Ziel der Propagation ist es, die bislang unbekannte Verteilung einer Variablen basierend auf so genannten Evidenzen zu erlernen. Unter einer Evidenz versteht man eine Aussage, die beobachtbar ist und damit über eine gewisse Sicherheit verfügt (Son, 2002, S. 18). Dies bedeutet konkret, dass Evidenzen als Variablen des Bayes'schen Netzes einen bestimmten Zustand annehmen. Auf dieser Grundlage können die Wahrscheinlichkeiten anderer, nicht beobachteter Variablen, geschätzt werden. Das Wissen pflanzt sich somit ausgehend von der beobachteten Evidenz über das Netz fort. Man spricht in diesem Zusammenhang von der Interpretation der Evidenzen. Unter die bekanntesten Vertreter solcher Inferenzalgorithmen werden die Algorithmen nach Pearl (Pearl, 1996), Lauritzen und Spiegelhalter (Lauritzen & Spiegelhalter, 1988), Jensen et al. (Jensen, Olesen, & Andersen, 1990) sowie nach Shenoy-Shafer (Shenoy & Shafer, 1986) aufgeführt. Eine detaillierte Analyse der Inferenzalgorithmen findet sich in (Lepar & Shenoy, 1999).

Als entscheidende Stärke Bayes'scher Netze lässt sich anführen, dass der Repräsentationsformalismus der Bayes'schen Wahrscheinlichkeitstheorie folgt und damit wissenschaftlich mit eindeutigem Axiomensystem fundiert ist. Darüber hinaus lernt das Netz bei Bekanntwerden des Wahrheitswertes einzelner Aussagen automatisch. Ein Bayes'sches Netz ist damit ein dynamisches Gebilde, das sich an wandelnde Umweltzustände anpassen kann. Als nachteilig lässt sich herausstellen, dass durch die Notwendigkeit zur Bestimmung einer umfangreichen Menge von bedingten Wahrscheinlichkeiten ein hoher Grad an informativischer Anforderungen verwaltet werden muss. Entscheidender Nachteil im Rahmen des vorliegenden Forschungsvorhabens ist, dass sich Unwissenheit im Netz nicht repräsentieren lässt, da die Summe von These und Antithese immer eine Wahrscheinlichkeit von 1 erreichen muss ( $P(h) = 1 - P(\neg h)$ ). Zwischen nicht verbundenen Variablen innerhalb eines Bayes'schen Netzes werden

somit implizit bedingte Unabhängigkeiten unterstellt, womit menschnenne Unschärfe nicht mehr verarbeitbar ist. Bayes'sche Netze eignen sich aus diesem Grund eher für maschinennahe Einsatzgebiete, bei denen sich Wahrscheinlichkeiten schätzen lassen.

Stärken	Schwächen
<ul style="list-style-type: none"> <li>• Ist wissenschaftlich eindeutig fundiert</li> <li>• Bei Bekanntwerden des Wahrheitswertes einzelner Aussagen lernt das Netz automatisch</li> </ul>	<ul style="list-style-type: none"> <li>• Hoher Grad an informatorischen Anforderungen</li> <li>• Es werden implizit bedingte Unabhängigkeiten unterstellt</li> <li>• Menschennahe Unschärfe ist nicht verarbeitbar</li> </ul>

Tabelle 1: Stärken-Schwächen Analyse Bayes'scher Netze

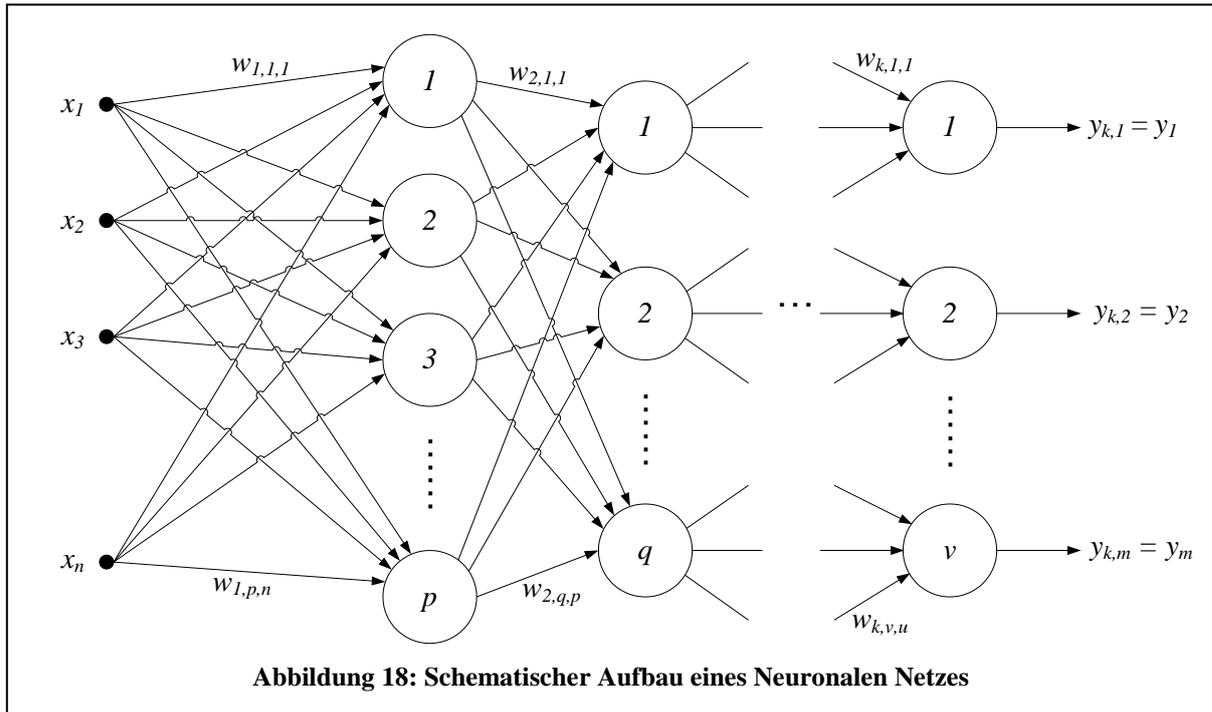
### 3.1.2 Künstliche neuronale Netze

*Künstliche neuronale Netze (KNN) stellen Modelle biologischer neuronaler Netze dar, welche aus einer Ansammlung einzelner Neuronen bestehen, die miteinander vernetzt Reizsignale verstärken oder dämpfen. Das vorliegende Unterkapitel geht auf die Grundlagen künstlicher neuronaler Netze ein, stellt Multilayer Perceptrons (MLP) als deren typische Vertreter vor und geht auf die Lernfähigkeit ein, welches eine der wichtigsten Eigenschaften neuronaler Netze darstellt. Abschließend wird eine kritische Analyse neuronaler Netze vorgenommen.*

Künstliche neuronale Netze sind informationsverarbeitende Systeme mit einer am biologischen Gehirn orientierten Struktur und Funktionsweise. Sie bestehen aus einer Menge parallel wirkender Einheiten („Neuronen“), welche Informationen in Form von Aktivierungssignalen versenden. Künstliche neuronale Netze arbeiten meist mit einfachen Schwellenwertelementen, die auch „McCulloch-Pitts-Neuronen“ genannt werden.

Aus systemtheoretischer Sicht betrachtet stellt ein Neuron ein nichtlineares System mit  $n$  Eingangsgrößen  $x_i$  und einer Ausgangsgröße  $y$  dar. Jeder Eingangsgröße  $x_i$  ist dabei ein Gewicht  $w_i$  zugeordnet. Nur wenn die Summe der Eingangsgrößen multipliziert mit deren jeweiligen Gewichtungsfaktoren einen vordefinierten Schwellenwert  $\theta$  überschreitet, sendet das Neuron einen Impulsfrequenz genannten Ausgangswert  $y$  aus. Der Ausgangswert ist dabei abhängig von der Reizintensität der Eingangswerte sowie von einer Ausgangs- beziehungsweise Aktivierungsfunktion  $S$ . Bei der Modellierung der Aktivierungsfunktion  $S$  sind folgende Punkte zu beachten:

- Zwischen Reizintensität und Impulsfrequenz sollte ein annähernd linearer Zusammenhang bestehen.
- Ab einer bestimmten Reizintensität sollte eine Begrenzung der Impulsfrequenz eintreten.



Zur Gewährleistung einer einfachen mathematischen Handhabbarkeit kann in bestimmten Fällen auch nur eines der beiden Merkmale erfüllt sein (Adamy, 2007, S. 134). Neuronen stellen die Elementarbausteine eines neuronalen Netzes dar. Die Netzstruktur eines neuronalen Netzes entsteht durch die Verknüpfung der Ein- und Ausgänge verschiedener Neuronen. Ein neuronales Netz ist dabei anwendungsabhängig aus einer fest definierten Anzahl an Neuronenschichten aufgebaut. Die Eingabeschicht enthält hierbei keine Neuronen, sondern besteht aus einer Menge an Eingangsgrößen  $x_i$  für die erste Neuronenschicht. Die Ausgangsgrößen ergeben sich aus den Ausgangswerten  $y_i$  der letzten Neuronenschicht.

Gebräuchliche neuronale Netze stellen die so genannten Multilayer Perceptrons (MLP) dar. Multilayer Perceptrons sind mehrschichtige Netze mit sigmoidaler Aktivierungsfunktion  $S$ . Sigmoidale Funktionen werden dabei als Funktionen definiert, die sowohl nach oben als auch nach unten beschränkt, stetig differenzierbar und streng monoton steigend sind. Diese Form der Aktivierungsfunktion findet aufgrund ihrer Leistungsfähigkeit und guten mathematischen Handhabbarkeit häufigen Einsatz (Adamy, 2007, S. 139). Insbesondere sind Multilayer Perceptrons dadurch charakterisiert, dass jedes Neuron einer Schicht mit jedem Neuron der Nachfolgeschicht verbunden ist. Am weitesten verbreitet sind MLPs mit zwei oder drei Schichten, wobei Schichten zwischen Ein- und Ausgabeschicht auch als „hidden units“ bezeichnet werden (Tzikas & Likas, 2010, S. 87). Die Gewichtungsfaktoren  $w_{r,j,l}$  bei MLPs sind gemäß ihrer Position indiziert:  $r$  identifiziert das Gewicht als Eingangsgewicht der  $r$ -ten Schicht,  $j$  gibt den Index des Neurons in Schicht  $r$  an und  $l$  den Index des Neurons in Schicht  $r - 1$ . Abbildung 18 veranschaulicht den Aufbau eines Multilayer Perceptrons.

Steht man in der Anwendung eines künstlichen neuronalen Netzes vor der Aufgabe, spezifische Gewichte und Schwellenwerte für ein gewünschtes Verhalten zu ermitteln, so sind die notwendigen Aktivierungsfunktionen festzulegen sowie die Gewichtungsfaktoren und die Schwellenwerte  $\theta$  zu bestimmen. Bereits bei Problemstellungen mit mehr als drei Eingängen je Neuron übersteigt die Komplexität des Lösungsraumes jedoch die Vorstellungskraft des Anwenders, womit eine manuelle Bestimmung der Netzparameter nicht mehr handhabbar ist. Stattdessen existiert die Möglichkeit, die Gewichtungsfaktoren und die Schwellenwerte einzelner Neuronen automatisch anzutrainieren. Man spricht dabei vom Lernen oder Trainieren eines Netzes. Hierzu sind zunächst  $n$  Lerndatenpaare

$$(\underline{\tilde{x}}^v, \underline{\tilde{y}}^v) \quad v = 1, \dots, n \quad [3.2]$$

zu sammeln, die das zu lernende Verhalten möglichst repräsentativ abbilden. Vor dem Lernen besitzt das Netz beliebige Gewichtungsfaktoren  $w_{r,j,l}$ , die durch den Lernprozess so umgeformt werden, dass der Ausgangswert  $y$  den durch das Lerndatum vorgegebenen Wert  $\underline{\tilde{y}}^v$  möglichst genau abbildet. Das Ziel des Lernalgorithmus ist es, den quadratischen Fehler

$$\epsilon = \frac{1}{n} \sum_{v=1}^n \sum_{j=1}^m (y_j^v - \tilde{y}_j^v)^2 = \frac{1}{n} \sum_{v=1}^n \sum_{j=1}^m (w_{r,j,l} \tilde{x}^v - \tilde{y}_j^v)^2 \quad [3.3]$$

über alle Lerndatenpaare zu minimieren. Das Training kann dabei in zwei unterschiedlichen Arten angewendet werden. Die eine Möglichkeit, Online-Training genannt, führt eine Anpassung der Gewichte und des Schwellenwertes nach jeder Berechnung einer Schaltmöglichkeit durch. Die andere Möglichkeit, das Batch-Training, summiert den Fehler während der Berechnung aller Schaltmöglichkeiten und passt die Gewichte und den Schwellenwert erst nach einem vollständigen Durchlauf aller Möglichkeiten an. Dabei ist zu betonen, dass die vorgestellten Trainingsarten nur auf einzelne Neuronen angewendet werden können. Die Anpassung ganzer Netze stellt eine ungemein größere Herausforderung dar und wird derzeit durch keine bekannte Methode unterstützt.

Als große Stärke künstlicher neuronaler Netze lässt sich deren Lernfähigkeit herausstellen, die an die Vernetzung von Neurozyten in Gehirn und Rückenmark angelehnt ist. Somit sind sie von Grund auf selbst anpassungsfähig. Da das Verhalten eines Netzes von vielen Einflussfaktoren wie Gewichtungsfaktoren, Schwellenwerten und Aktivierungsfunktionen abhängig ist, ist das gespeicherte Wissen jedoch unstrukturiert und nicht intuitiv verständlich (Scheer C. , 2006, S. 168-176). Die zugrundeliegenden Netzparameter ermöglichen darüber hinaus zwar eine genaue Abbildbarkeit des gewünschten Verhaltens, daneben sind sie jedoch auch dafür verantwortlich, dass eine große Rechenkapazität zum Lösen der Probleme benötigt wird. Zudem weist die Lernfähigkeit Schwachstellen auf, da in vielen Fällen keine oder eine sehr langsame Konvergenz

der Netzparameter auftritt. Durch irreguläre Eingangswerte kann das erlernte Wissen darüber hinaus wieder vergessen werden (Tu, 1996, S. 1229). Eine Anwendung im Rahmen des vorliegenden Forschungsvorhabens empfiehlt sich insbesondere durch mangelnde Intuitivität für den Menschen nicht.

Stärken	Schwächen
<ul style="list-style-type: none"> <li>• Lernfähigkeit</li> <li>• Vielzahl an Netzparametern ermöglichen sehr genaue Abbildbarkeit eines erwünschten Verhalten</li> </ul>	<ul style="list-style-type: none"> <li>• Unstrukturierte und nicht intuitive Speicherung des Wissens</li> <li>• Rechenintensivität</li> <li>• Komplexität</li> <li>• Schwachstellen der Lernfähigkeit wie langsame Konvergenz oder Instabilitäten</li> </ul>

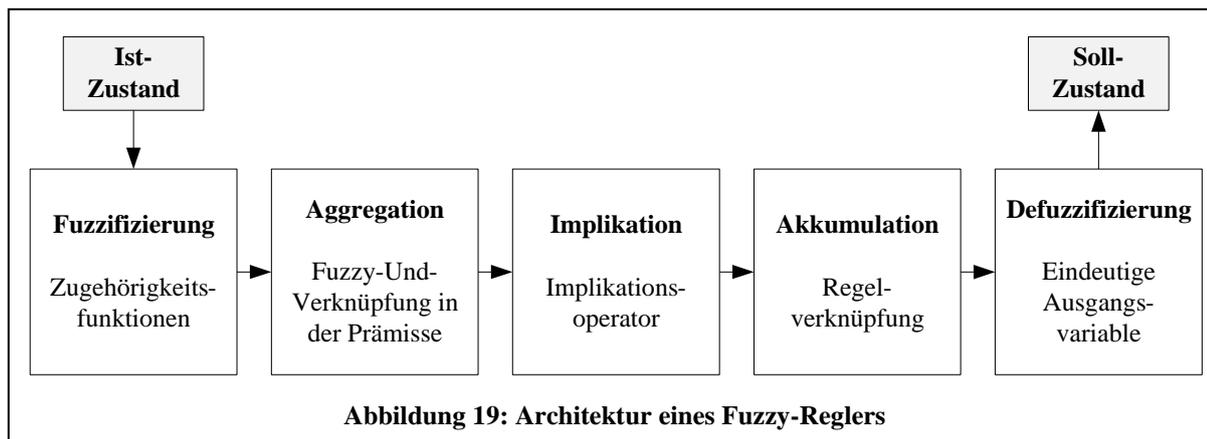
Tabelle 2: : Stärken-Schwächen Analyse Künstlicher Neuronaler Netze

### 3.1.3 Fuzzy-Systeme

*Das vorliegende Unterkapitel stellt den Ansatz der Fuzzy-Logik vor, der die Verarbeitung unscharfer umgangssprachlicher Ausdrücke ermöglicht. Dabei wird zunächst auf Grundlagen der Fuzzy-Logik eingegangen. Darauf aufbauend werden Fuzzy-Regler als auf Fuzzy-Logik basierende Gesamtsysteme zur Ableitung logischer Schlussfolgerungen vorgestellt und im Anschluss eine kritische Analyse des Fuzzy-Ansatzes vorgenommen.*

Die Fuzzy-Logik ist eine Theorie, die zur Modellierung von Unschärfe in umgangssprachlichen Beschreibungen entwickelt wurde (Adamy, 2007, S. 5). Sie basiert auf der Erkenntnis, dass die menschliche Kognition nicht nur auf zwei Wahrheitswerten beruht, sondern stattdessen in der Regel ein hohes Maß an Unschärfe und Vagheit aufweist. Ein Beispiel hierfür ist die Aussage „Das Fahrzeug fährt sehr schnell“. Fuzzy-Systeme sind ein Ansatz, solche unpräzisen menschlichen Ausdrücke und Regeln auf einem mathematisch basierten System zu modellieren. Der Vorläufer der Fuzzy-Logik wurde 1920 von Łukasiewicz vorgestellt, der das System der booleschen Logik auf alle reellen Zahlen im Intervall  $[0,1]$  erweiterte. Die erste vollständige Theorie der unscharfen Logik entwickelte Lofti Zadeh 1965 (Zadeh, 1965) basierend auf Łukasiewicz's Ansatz. Mit der somit entstandenen Theorie konnten auch Schlussfolgerungsprozesse exakt beschrieben werden.

Innerhalb der Aussagenlogik werden logischen Aussagen einer der beiden Wahrheitswerte wahr oder falsch zugeordnet. Die Fuzzy-Logik hingegen modelliert unscharfe Aussagen, indem sie diesen einen Wahrheitsgrad im Wertebereich  $[0,1]$  zuweist. Dabei steht der Wert 1 für volle Zugehörigkeit und 0 als Wert für vollständige Nichtzugehörigkeit. Ziel ist es, sprachliche Konzepte durch mathematische Funktionen auszudrücken, die jedem Element einer Grundmenge



$G$  eine Zugehörigkeit zuordnen (Kramer, 2009, S. 79). Das modellierte Konzept wird dabei als *linguistische Variable* bezeichnet, die Ausprägungen des modellierten Konzeptes als *linguistische Terme*. Zur Mathematisierung dieser sprachlichen Konzepte werden *Fuzzy-Mengen* eingeführt, die über eine *Zugehörigkeitsfunktion*  $\mu$  definiert werden.

**Definition 3.2 (Fuzzy-Menge):** Sei  $G$  eine Grundmenge und  $\mu$  eine Funktion:  $G \rightarrow [0,1]$ , die jedem  $x \in G$  eine Zugehörigkeit  $\mu(x) \in [0,1]$  zuordnet. Dann ist die Fuzzy-Menge  $M$  definiert als die Menge aller Paare  $M = \{(x, \mu(x)) \mid x \in G\}$ .

Fuzzy-Mengen dienen als Konzept zur Formulierung unscharfer menschlicher Aussagen. Der menschlichen Kognition sind auf Basis dieser unscharf formulierter Aussagen Schlussfolgerungsprozesse möglich (Kramer, 2009, S. 85), die innerhalb der Fuzzy-Logik als approximatives Schließen bezeichnet werden. Die Fuzzy-Logik bedient sich hierfür einer verallgemeinerten Variante des Modus Ponens (auch Ableitungsregel genannt) aus der booleschen Logik. Weiterführende Informationen finden sich in (Kramer, 2009). Auf Basis des Fuzzy Modus Ponens werden Fuzzy-Regeln der Form

$$R_k: \text{Wenn } x \text{ ist } A, \text{ dann } y \text{ ist } B \quad [3.4]$$

ermöglicht. Die Implikation wird als Relation  $\Omega \subset A \times B$  aufgefasst und kann auf unterschiedliche Weise implementiert werden. Durch die unterschiedlichen Realisierungsformen wird die Konstruktion von Inferenzsystemen mit unterschiedlichen Eigenschaften und Verhaltensweisen ermöglicht. Industrielle Anwendungsgebiete für auf Fuzzy-Logik basierende Gesamtsysteme zur Ableitung logischer Schlussfolgerungen sind Fuzzy-Bildverarbeitung, Fuzzy-Datenanalyse, Fuzzy-Datenbanken und Fuzzy-Regelungen (Kruse, Gebhardt, & Klawonn, 1995, S. 68). Die meisten industriellen Anwendungen wurden im Bereich der Fuzzy-Regelung entwickelt, auf die im Folgenden genauer eingegangen wird.

Beim Regeln besteht die Aufgabe darin, einen Parameter auf einen Sollwert einzustellen (Kramer, 2009, S. 89). Die Architektur eines solchen Fuzzy-Reglers ist in Abbildung 19 ersichtlich. Dem Regler wird dabei ein scharfer Ist-Wert übergeben, der zunächst zu mathematisieren (fuzzifizieren) ist. Der folgende Inferenzprozess wird auf Basis einer Fuzzy-Regelbasis ausgeführt und erfolgt in mehreren Schritten. Zunächst wird innerhalb des Aggregationsschrittes die Prämisse einer Regel  $R_k$  ausgewertet, indem für jede Regel das Maß an Übereinstimmung mit dem Eingabeparameter bestimmt wird. Im Implikationsschritt erfolgt die Verknüpfung von Prämisse und Konklusion einer Regel. Die Vereinigung der Ausgaben aller Regeln zu einer Gesamtausgabe erfolgt innerhalb der Akkumulation. Um aus diesem Ergebnis einen scharfen Zahlenwert zu gewinnen, ist das erhaltene Ergebnis zu defuzzifizieren. Eine detaillierte Darstellung des Prozesses ist in (Adamy, 2007) ersichtlich.

Fuzzy-Logik ist dazu ausgelegt, menschliches Kausalwissen sowie sprachliche Unschärfe zu mathematisieren (Adamy, 2007, S. 7). Durch diese Möglichkeit ist der Grundstein für die rechnergestützte Abbildung menschlichen Verhaltens gelegt. Das Wissen ist strukturiert in Regeln gespeichert, welche auf einfache Weise menschliche Verhaltensmuster nachempfinden (Burger, Bittel, Awad, & Ovtcharova, 2011). Die Funktionsweise ist somit intuitiv nachvollziehbar, sowie einfach zu interpretieren und implementieren (Nauck, Borgelt, Klawonn, & Kruse, 2003, S. 185). Als Nachteil von Fuzzy-Systemen lässt sich anführen, dass weder Regeln noch Fuzzy-Mengen basierend auf Beispieldaten generiert werden können. Fuzzy-Systeme sind demnach anders als Bayes'sche Netze oder Neuronale Netze nicht direkt lernfähig (Nauck, Borgelt, Klawonn, & Kruse, 2003, S. 185).

Stärken	Schwächen
<ul style="list-style-type: none"> <li>• Wissen ist strukturiert in Regeln gespeichert</li> <li>• Menschliches Kausalwissen und menschliche Unschärfe können mathematisiert werden</li> <li>• Funktionsweise kann intuitiv nachvollzogen werden</li> </ul>	<ul style="list-style-type: none"> <li>• Fuzzy-Systeme sind direkt nicht lernfähig</li> </ul>

Tabelle 3: : Stärken-Schwächen Analyse von Fuzzy-Systemen

### 3.1.4 Zusammenfassung

Im vorliegenden Abschnitt wurden ausgewählte Methoden der Künstlichen Intelligenz auf ihre Eignung hin untersucht, Kunden- und Anbietersicht auf intuitive Weise und unter Berücksichtigung sprachlicher Unschärfe miteinander zu verknüpfen. Die Untersuchung liefert

somit die Grundlage zur Befriedigung des Unterziels *UI-1* dieser Arbeit, in dem die Berücksichtigung natürlicher menschlicher Unschärfe bei der Verknüpfung von Kunden- und Anbietersicht gefordert wird. Hierbei wurden Fuzzy-Systeme als aussichtsreiche Kandidaten identifiziert, jedoch auch deren fehlende Lernfähigkeit als entscheidende Schwachstelle herausgestellt.

Der aufgestellte Handlungsbedarf *UI-2* fordert das Erkennen impliziten Wissens innerhalb der Kundenaussagen. Semantische Technologien stellen eine Möglichkeit dar, die Bedeutung ihrer zugrundeliegenden Daten zu ermitteln und somit auf implizites Wissen zuzugreifen. Diese sollen im Folgenden vorgestellt und einer kritischen Analyse unterzogen werden.

## 3.2 Semantische Technologien

*Semantische Technologien erlauben es, die Bedeutung ihrer zugrundeliegenden Daten zu ermitteln und somit auf implizites Wissen zuzugreifen. Das Rückgrat semantischer Technologien bilden Ontologien, die mithilfe verschiedener Sprachen zur Wissensrepräsentation abgebildet werden können. Das vorliegende Unterkapitel diskutiert in Abschnitt 3.2.1 die Grundlagen von Ontologien und geht in Abschnitt 3.2.2 auf Sprachen zur Wissensrepräsentation ein. Insbesondere werden in Abschnitt 3.2.3 Fuzzy-Ontologien vorgestellt, welche die Charakteristika von Ontologien mit denen der Fuzzy-Logik vereinen. Abschnitt 3.2.4 stellt Methoden des Ontology Engineering vor, welche die strukturierte Erstellung von Ontologien, aber auch herkömmlicher Informationsmodelle ermöglichen. Abschließend werden Anwendungspotentiale semantischer Technologien im Rahmen der vorliegenden Arbeit dargestellt und somit die Analyse in Abschnitt 3.2.5 geschlossen.*

### 3.2.1 Grundlagen

Unter einer Ontologie wird ein Konstrukt verstanden, das Wissen über ein bestimmtes Gebiet repräsentiert und den Wissensaustausch innerhalb dieses Gebiets ermöglicht (Jarrar, 2006, S. 497). Der Begriff Ontologie stammt ursprünglich aus der Philosophie und beschäftigt sich in diesem Zusammenhang mit der systematischen Beschreibung des Wesens des Seins (Gomez-Perez, Fernandez-Lopez, & Corcho-Garcia, 2003, S. 3). Eine Ontologie versucht zu beantworten, welche Objekte existieren und in welcher Form diese geordnet und innerhalb einer Hierarchie in Beziehung gesetzt werden können (Wallace, 2007, S. 175). So beschäftigte sich beispielsweise Aristoteles in seinem Werk „Metaphysik“ mit der Frage, was das Wesen eines Samens ausmacht, wenn dieser zu einem Baum wird. Im Zuge fortschreitender Technologisierung in der letzten Dekade des zwanzigsten Jahrhunderts hielten Ontologien Einzug in das Gebiet der

Informationstechnik, innerhalb dessen sie in den Bereichen des Wissensmanagements sowie der künstlichen Intelligenz Einsatz finden. Im Englischsprachigen wird zwischen dem großgeschriebenen Begriff „Ontology“ und dem kleingeschriebenen Begriff „ontology“ unterschieden, wobei ersteres auf den informationstechnischen Einsatz und letzteres auf den philosophischen Sinn verweist (Guarino & Garetta, 1995).

Zur Bestimmung des Begriffs Ontologie im Bereich der Informationstechnik wird innerhalb der Wissenschaft in der Regel die Definition nach Gruber verwendet, die eine Ontologie als eine „*explizite Spezifikation einer Konzeptualisierung*“<sup>1</sup> festlegt (Gruber T. , 1993, S. 199). Die Definition nach Studer baut auf Grubers Definition auf und erweitert diese um eine Erläuterung der einzelnen definatorischen Bestandteile. Zum besseren Verständnis für den Leser wird im Zuge der vorliegenden Arbeit die Definition nach Studer verwendet (Studer, Benjamins, & Fensel, 1998, S. 185).

Eine Ontologie ist demnach eine „*formale, explizite Spezifikation einer gemeinsamen Konzeptualisierung*“<sup>2</sup>. Studer erläutert weiterhin:

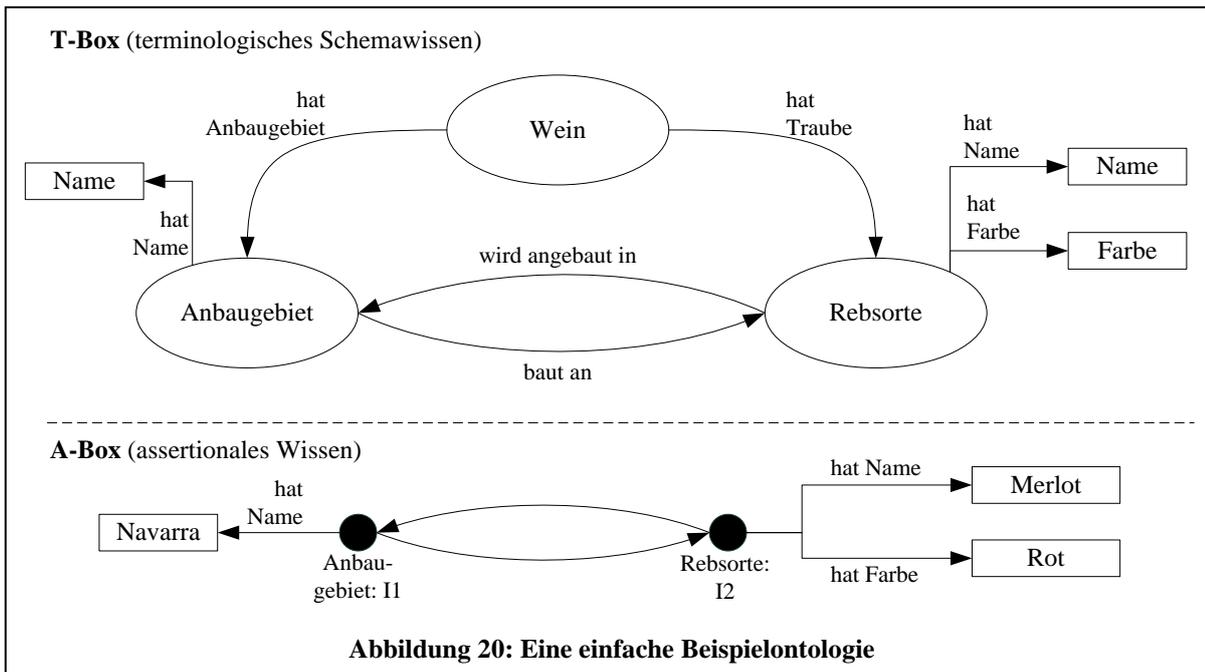
- *„Konzeptualisierung bezieht sich auf ein abstraktes Modell eines Phänomens, zu dem alle relevanten Konzepte identifiziert wurden.*
- *Explizit bedeutet, dass die Art der verwendeten Konzepte und die Bedingungen ihrer Nutzung eindeutig definiert wurden.*
- *Formal bezieht sich auf die Tatsache, dass die Ontologie maschinenlesbar sein sollte.*
- *Gemeinsam spiegelt die Absicht wider, dass eine Ontologie Konsenswissen erfasst, das nicht von einem Individuum, sondern einer Gruppe akzeptiert wird.“*

Während Ontologien in den 1990er Jahren hauptsächlich unter Verwendung der Prädikatenlogik erster Ordnung (Barwise & Etchemendy, 2005) modelliert wurden, setzten sich seit Aufkommen des Semantic Web Beschreibungslogiken als Grundlage der Ontologiemodellierung durch. Unter Beschreibungslogiken wird eine Familie von Sprachen zur Wissensrepräsentation verstanden, die eine Untermenge der Prädikatenlogik erster Ordnung darstellen. Im Gegensatz zu dieser sind Beschreibungslogiken in der Regel entscheidbar, das heißt es existiert ein Algorithmus, der für jedes Element einer Menge beantworten kann, ob es eine vorgegebene Eigenschaft aufweist oder nicht (Essler, Martínez Cruzado, & Labude, 2001, S. 325f.). Dies ermöglicht Schlussfolgerungsprozesse, die neues Wissen aus bestehendem Wissen

---

<sup>1</sup> *“An ontology is an explicit specification of a conceptualization.”*

<sup>2</sup> *“An ontology is a formal, explicit specification of a shared conceptualization. Conceptualization refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. Explicit means that the type of concepts used, and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine-readable. Shared reflects the notion that an ontology captures consensual knowledge, that is, it is not private of some individual, but accepted by a group.”*



generieren. Formal setzt sich eine Beschreibungslogik aus einer T-Box sowie aus einer A-Box zusammen. Die T-Box beinhaltet terminologisches Schemawissen, welches Aussagen über generelle Eigenschaften von Konzepten macht. Die A-Box beinhaltet assertionales Wissen über die Individuen des Anwendungsgebietes (Baader, Calvanese, McGuinness, Nardi, & Patel-Schneider, 2003, S. 17). Abbildung 20 stellt eine einfache Beispielontologie für die Domäne Wein dar. Als Basiselemente zur Modellierung von Ontologien werden Konzepte, Rollen und Individuen verwendet.

- Konzepte verkörpern Klassen von Objekten
- Rollen beschreiben binäre Verbindungen zwischen Konzepten
- Individuen stellen Instanzen von Klassen dar

Ihre aktuelle Popularität verdanken Ontologien neuen Entwicklungen im Bereich des Semantic Web, einer semantischen Version des World Wide Web, die es Webressourcen erlaubt, die Bedeutung der zugrundeliegenden Daten zu ermitteln. Grundlegende Idee des Semantic Web ist es, Informationen mitsamt ihrer Bedeutung derart zur Verfügung zu stellen, so dass eine Verarbeitung durch Maschinen ermöglicht wird (Berners-Lee, Hendler, & Lassila, 2001, S. 34ff.). Hierfür bedient sich das Semantic Web Grundlagentechnologien, die Semantische Technologien genannt werden. Zu diesen gehören neben Wissensrepräsentationssprachen für Ontologien auch Methoden und Werkzeuge zu deren Erstellung, Wartung und Anwendung (Hitzler, Krötzsch, Rudolph, & Sure, 2008, S. 12). Auf Sprachen zur Wissensrepräsentation soll im Folgenden eingegangen werden.

### 3.2.2 Sprachen zur Wissensrepräsentation

Offene Standards für Sprachen zur Wissensrepräsentation im Bereich des Semantic Web definiert das World Wide Web Consortium (W3C). Zu diesen zählen XML, RDF(S) und OWL, wobei XML weniger dem Semantic Web als vielmehr dem herkömmlichen Web zuzuordnen ist. Mit RDF(S) und OWL hingegen wurden speziell für die Verwendung im Semantic Web nutzbare Ontologiesprachen entwickelt. Das Resource Description Framework (RDF) und dessen Erweiterung RDF-Schema (RDF(S)) stellen die grundlegenden Ontologiesprachen des Semantic Web dar. Eine erste Empfehlung des W3C wurde 2003 veröffentlicht (Allemang & Hendler, 2008, S. 28). Zentrale Idee ist es, binäre Relationen als Tripel der Form (Subjekt, Prädikat, Objekt) zwischen eindeutig bezeichneten Ressourcen zu beschreiben (Stuckenschmidt, 2011, S. 102f.). Das Prädikat innerhalb des Tripels entspricht der Relation zwischen den durch Subjekt und Objekt bezeichneten Ressourcen. RDF als auch RDF(S) weisen jedoch grundsätzliche Einschränkungen auf, welche die Modellierungsfähigkeit beeinträchtigen. Zu den gravierendsten Einschränkungen gehört die Unmöglichkeit negierte Aussagen auszudrücken (Hitzler, Krötzsch, Rudolph, & Sure, 2008, S. 119).

Zur Darstellung komplexeren Wissens werden ausdrucksstarke Repräsentationssprachen verwendet, die wie auch schon RDF(S) auf formaler Logik basieren. Durch logisches Schlussfolgern wird somit auch der Zugriff auf implizites Wissen ermöglicht (Hitzler, Krötzsch, Rudolph, & Sure, 2008, S. 125). Hierfür wurde die Sprache Web Ontology Language (OWL) entwickelt, die seit Februar 2004 als Empfehlung des W3C vorliegt (World Wide Web Consortium, 2004). OWL basiert auf der RDF-Syntax und der hier nicht näher erwähnten DAML+OIL Markup-Language (Joint US/EU Agent Markup Language Committee, 2001), geht über deren Ausdrucksmächtigkeit jedoch weit hinaus. Seit Oktober 2009 existiert eine Empfehlung des W3C mit der zweiten Version von OWL – OWL 2 genannt –, um Entwicklern eine einfachere Implementierung zu ermöglichen (World Wide Web Consortium, 2009).

Bei der Entwicklung von OWL 2 war wie auch bereits bei der Vorgängerversion die Skalierbarkeit der Sprache abhängig vom Anwendungskontext ein wichtiges Ziel. Da ausdrucksmächtige Sprachen zu hohen Komplexitäten bei der Berechnung führen, wurden drei Teilsprachen von OWL 2 definiert, zwischen denen der Anwender wählen kann. Die am wenigsten ausdrucksmächtige Teilsprache OWL 2 Lite weist ein polynomielles Laufzeitverhalten auf und ist damit entscheidbar. Sie ist vor allem zur Erstellung einfacher Taxonomien gedacht. OWL 2 DL als nächstmächtigere Teilsprache entspricht der Beschreibungslogik *SR<sub>Q</sub>IQ* und gehört damit zu einer entscheidbaren Untermenge der Prädikatenlogik erster Ordnung (Sachs, Helbig, Herrmann, Welge, & Bollow, 2010). Sie erweitert OWL 2 Lite um weitere Sprachkonstrukte und lockert gewisse dort vorhandene Einschränkungen. OWL 2 Full als Erweiterung von OWL 2 DL baut auf denselben

Sprachkonstrukten wie OWL 2 DL auf, verzichtet allerdings ganz auf Einschränkungen (Degle, 2005). Damit besitzt OWL 2 Full zwar die größte Ausdrucksmächtigkeit aller Teilsprachen, ist allerdings auch unentscheidbar.

#### 3.2.3 Fuzzy-Ontologien

Klassische Beschreibungslogiken und Sprachen zur Wissensrepräsentation weisen Schwächen in denjenigen Anwendungsgebieten auf, in denen unscharfe Informationen zu verarbeiten sind. Unscharfe Informationen beginnen jedoch, eine bedeutende Rolle innerhalb Forschungsaktivitäten zu spielen, die sich mit dem Web oder dem Semantic Web auseinandersetzen. Beispielsweise wird die Reihenfolge der Antworten, die Google im Anschluss an eine Anfrage ausgibt, auf Basis probabilistischer Algorithmen errechnet. Auch im Forschungsumfeld gehen Bemühungen zur Nutzung von Fuzzy-Logik im Bereich der Beschreibungslogiken auf logikbasierte Algorithmen zur Informationsgewinnung zurück. Innerhalb des W3C entstand daher eine so genannte Incubator Group, die sich mit unscharfen Schlussfolgerungsprozessen innerhalb des Web auseinandersetzt (Lukasiewicz & Straccia, 2008, S. 292). Ein Standard für die Repräsentation unscharfen Wissens wird in absehbarer Zukunft jedoch nicht erwartet (Bobbilo & Straccia, 2011, S. 2).

Eine detaillierte Analyse bestehender Ansätze zur Nutzung der Fuzzy-Logik im Bereich der Beschreibungslogiken und Ontologiesprachen geben Lukasiewicz und Straccia (Lukasiewicz & Straccia, 2008). Im Folgenden soll in Anlehnung an Lukasiewicz und Straccia kurz auf diese Ansätze eingegangen werden und eine Analyse erfolgen. Die erste Arbeit geht auf Yen (Yen, 1991) zurück, der eine Fuzzyerweiterung von  $\mathcal{FL}$ -, einer sehr begrenzten Untersprache von  $\mathcal{ALC}$ , vorschlägt. Dabei werden lediglich terminologische Konstrukte unterstützt. Auch der Ansatz von Tresp und Molitor (Tresp & Molitor, 1998) unterstützt keine Individuen, sondern lediglich terminologische Konstrukte. Bei ihrer etwas gemeingültigeren Erweiterung der Beschreibungslogik  $\mathcal{ALC}$  erlauben sie zudem Fuzzy-Modifikatoren zur Umformung von Konzepten. Eine weitere Fuzzy-Erweiterung von  $\mathcal{ALC}$  schlägt Straccia (Straccia, 1998) vor, der zum ersten Mal neben terminologischen Konstrukten auch Individuen erlaubt. Wie in den ersten beiden Ansätzen auch legt er die Logik nach Zadeh als Fuzzy-Logik zugrunde. Eine Unterstützung von Fuzzy-Modifikatoren wird jedoch nicht vorgesehen. Eine Erweiterung der Beschreibungslogik  $\mathcal{ALCQ}$  wird von Sanchez und Tettamanzi (Sanchez & Tettamanzi, 2004) vorgeschlagen. Auch bei ihnen bildet Zadehlogik die Grundlage, Individuen werden wiederum nicht unterstützt. Im Ansatz nach Hájek (Hájek, 2005) wird die Beschreibungslogik  $\mathcal{ALC}$  zugrunde gelegt. Zum ersten Mal wird dabei keine Fuzzy-Logik fest vorgegeben. Stattdessen

Autor	Datum	DL	Konzepte	Individuen	Fuzzy-Modifizier	Axiome	Fuzzy-Logik	Reasoning
Yen	1991	$\mathcal{FL}$	●				Zadeh	Unvollständig (Auf Crisp sets beschränkt)
Tresp & Molitor	1998	$\mathcal{ALC}$	●		●		Zadeh	Vollständiger Algorithmus
Straccia	2001	$\mathcal{ALC}$	●	●			Zadeh	Vollständiger Algorithmus
Sanchez & Tettamanzi	2004	$\mathcal{ALCQ}$	●				Zadeh	Unvollständig (Berechnung des Erfüllbarkeitsintervalls)
Hájek	2005	$\mathcal{ALC}$	●				Mehrere	Unvollständig (Alg. zur Tautologieerkennung)
Bobillo & Straccia	2011	$\mathcal{SROIQ}$	●	●	●	●	Mehrere	Vollständiger Algorithmus

**Tabelle 4: Analyse bestehender Ansätze zur Nutzung der Fuzzy-Logik im Bereich der Beschreibungslogiken und Ontologiesprachen**

erlaubt Hájek eine Menge frei wählbarer t-Normen<sup>3</sup>, womit mehrere Fuzzy-Logiken verwendbar werden. Der aktuellste Ansatz stellt ein Vorschlag nach Bobillo und Straccia (Bobillo & Straccia, 2011) dar, die eine Fuzzy-Erweiterung der Beschreibungslogik  $\mathcal{SROIQ}$  vorschlagen. Im Detail stellt die Methode eine Fuzzy-Erweiterung der Wissensrepräsentationssprache OWL 2 DL dar, deren Grundlage die Beschreibungslogik  $\mathcal{SROIQ}$  bildet. Der Ansatz erlaubt neben terminologischen Konstrukten auch die Definition von Individuen sowie Fuzzy-Modifikatoren. Die Verwendung einer bestimmten Fuzzy-Logik wird nicht vordefiniert, stattdessen sind ähnlich zum Ansatz von Hájek mehrere Fuzzy-Logiken verwendbar. Zwar existieren mit bereits mehrere Ansätze zur Erweiterung von OWL bzw. OWL 2 um Fuzzy-Logiken, der vorliegende Ansatz unterscheidet sich von diesen jedoch dadurch, dass der bestehende Standard nicht erweitert werden muss. Stattdessen wird das Grundgerüst der Erweiterung von OWL 2 gestellt, das bereits als W3C Standard eine breite Anwendung findet. Die Erweiterung um Fuzzy-Bestandteile erfolgt über dort übliche Annotationen. Tabelle 4 stellt die Ergebnisse der Analyse bestehender Ansätze zur Nutzung der Fuzzy-Logik im Bereich der Beschreibungslogiken und Ontologiesprachen übersichtlich dar.

<sup>3</sup> Siehe Definition in Kapitel 4.3.2 Architektur des DfC Fuzzy-Controllern

Name	Datum	Veröffentlichung
CommonKADS	1992	(Schreiber, Wielinga, de Hoog, Akkermans, & Van De Velde, 1994)
DILIGENT	2005	(Tempich C. , Pinto, Sure, & Staab, 2005)
DynamOnt	2006	(Gruber, Westenthaler, & Gahleitner, 2006)
Grüninger und Fox	1995	(Grüninger & Fox, 1995)
METHONTOLOGY	1997	(Fernandez & Gomez-Perez, 1997)
On-To-Knowledge	2002	(Sure, Staab, & Studer, 2002)
UPON	2009	(De Nicola, Missikoff, & Navigli, 2009)

**Tabelle 5: Methoden des Ontology Engineering**

#### 3.2.4 Ontology Engineering

*Aufgabe des Ontology Engineering ist es, die strukturierte Entwicklung, Wartung und Pflege von Ontologien zu ermöglichen. Somit kann das Ontology Engineering bei der Bereitstellung eines Vorgehensrahmens zur Unterstützung des Lösungsanbieters bei der Einführung der Methodik im Unternehmen (U1-3) behilflich sein. Im Rahmen des vorliegenden Unterkapitels werden bestehende Vorgehensmodelle des Ontology Engineering vorgestellt und im Anschluss daran einer auf IEEE Standard basierenden Vergleichsanalyse unterzogen. Um den Rahmen der vorliegenden Arbeit nicht zu sprengen, findet sich die detaillierte Analyse der Vorgehensmodelle in Anhang A dieser Arbeit.*

Das Ontology Engineering beschäftigt sich mit der strukturierten Entwicklung, Wartung und Pflege von Ontologien. Trotz der engen Verwandtschaft zur reinen Datenmodellierung geht es wesentlich über diese hinaus. Die Informatik definiert das Ontology Engineering als den Zweig des Knowledge Engineering, welcher das Ziel hat, Domänenwissen innerhalb einer Gruppe von Personen strukturiert zu formalisieren, um den Wissensaustausch zwischen Menschen und Informationssystemen zu ermöglichen (Jarrar & Meersman, 2002, S. 1238) (Gomez-Perez, Fernandez-Lopez, & Corcho-Garcia, 2003, S. V). Das Knowledge Engineering selbst ist ein Bestandteil des Knowledge Management (Wissensmanagement), das sich dediziert mit der Kontrolle und Verwaltung des im Unternehmen vorhandenen Wissens auseinandersetzt, um eine vereinfachte Erreichung der Unternehmensziele zu ermöglichen (van der Spek & Spijkervet, 2005, S. 25-26).

In den letzten beiden Jahrzehnten entstand eine Reihe von Methoden des Ontology Engineering mit verschiedenen Schwerpunkten der inhaltlichen Ausrichtung. So fokussieren gewisse Methoden insbesondere Aspekte des Projektmanagements, andere Methoden legen Schwerpunkt auf die inhaltliche Ausrichtung und vernachlässigen die detaillierte Darstellung der

Projektplanung. Ein allgemein anerkannter Standard für die Entwicklung von Ontologien existiert daher nicht. Im Folgenden sollen die für das vorliegende Forschungsvorhaben relevanten Methoden kurz vorgestellt werden. Eine Übersicht der gewählten Methoden ist in Tabelle 5 ersichtlich. Der interessierte Leser findet eine detaillierte Analyse in Anhang A dieser Arbeit.

*CommonKADS* (Schreiber, Wielinga, de Hoog, Akkermans, & Van De Velde, 1994) ist die erste Methode, welche die Abbildung von verteiltem Domänenwissen als Modellierungsaktivität begreift. Die Methodik beschreibt keinen bestimmten Prozess, sondern stellt Metamodelle für spezifische Modellierungstätigkeiten bereit (Schreiber, Wielinga, de Hoog, Akkermans, & Van De Velde, 1994, S. 33). Die in *CommonKADS* vorhandenen Metamodelle werden in vielen weiteren Methoden des Ontologie Engineering wieder aufgegriffen (Gruber, Westenthaler, & Gahleitner, 2006, S. 256). Der Beschreibung mangelt es jedoch einer formalen Tiefe sowie an einer ungenügenden Unterstützung des Modellierungsprozesses.

Die Methode von *Grüninger und Fox* (Grüninger & Fox, 1995) liefert ein Grundgerüst zur Erstellung und Weiterentwicklung von Ontologien zur Abbildung eines Unternehmens sowie dessen Prozessen. Das Modell nach Grüninger und Fox baut als erstes Modell auf so genannten Kompetenzfragen (Competency Questions) auf, welche die Anforderungen an die Ontologie darstellen und die finale Ontologie beantworten können muss. Die Methodik bedient sich der Robustheit klassischer Logik, indem informelle Beschreibungen in prädikatenlogische Ausdrücke umgewandelt werden. Die Methode hat eine klare Struktur, versäumt es jedoch, eine Beschreibung in ausreichender Detailtiefe zu liefern (Dahlem & Hahn, 2009, S. 4).

Die von Fernandez und Gomez-Perez entwickelte Methode *METHONTOLOGY* (Fernandez & Gomez-Perez, 1997) proklamiert einen Entwicklungszyklus für Ontologien auf Basis sich kontinuierlich weiterentwickelnder Prototypen (Evolving prototypes) (Gomez-Perez, Fernandez-Lopez, & Corcho-Garcia, 2003, S. 126). Die Methode basiert auf dem IEEE Standard für Softwareentwicklungsprozesse (IEEE, 1996) und wird von der FIPA für die Entwicklung von Ontologien empfohlen (Gomez-Perez, Fernandez-Lopez, & Corcho-Garcia, 2003, S. 125). Die eigentliche Entwicklung wird genauso betrachtet wie Aktivitäten des Projektmanagements. Insbesondere die klare Struktur sowie die detaillierte Beschreibung der Konzeptualisierung sind hervorzuheben.

Die Methode *On-To-Knowledge* (OTK) (Sure, Staab, & Studer, 2002) wurde im Rahmen eines gleichnamigen EU-Forschungsprojektes entwickelt und hat eine starke Betonung auf Projektmanagementaktivitäten im Bereich der Ontology Engineering. Details zur Wissenserfassung und Konzeptualisierung werden hingegen nur oberflächlich behandelt. Der Knowledge Meta Process als erster Hauptprozess hat die Einführung und Pflege neuer Anwendungen des Wissensmanagements im Fokus, während sich der zweite Hauptprozess

(Knowledge Process) mit der Handhabung bereits bestehender Anwendungen befasst. Zur Unterstützung der Nutzer werden formale Aspekte des Ontology Engineering durch grafische Repräsentationsformen wie MindMaps versteckt.

Im Gegensatz zu anderen Methoden fokussiert *DILIGENT* (Tempich C. , Pinto, Sure, & Staab, 2005) die nutzerzentrierte kontinuierliche (Weiter-)Entwicklung von Ontologien. Während unterstützende Aktivitäten des Projektmanagements nicht betrachtet werden, wird besonderer Wert auf die Extraktion von Konsenswissen und kollaborative Prozesse gelegt. *DILIGENT* beschreibt die Ontologieentwicklung als Prozess, der einer verteilten Gruppe von Nutzern die Möglichkeit gibt, Konsenswissen über eine Argumentation formal zu extrahieren. Die formale Abbildung von Diskussionen zwischen verteilten beteiligten Nutzergruppen wird über eine auf der IBIS Methodik (Buckingham Shum, Motta, & Domingue, 2002) basierenden Vorgehensweise erlaubt.

*DynamOnt* (Gruber, Westenthaler, & Gahleitner, 2006) ist eine Methode zur schnellen Erstellung prototypischer Ontologien. Mit dem 3D Modell wird ein Werkzeug zur schnellen Kategorisierung und Segmentierung eines Wissensraums zur Verfügung gestellt, das einen schnellen ersten Überblick über eine gegebene Problemstellung erlaubt. Neben einem Prozessmodell fehlt es der Methode an Detailtiefe und formaler Unterstützung (Dahlem & Hahn, 2009, S. 4).

*UPON* (De Nicola, Missikoff, & Navigli, 2009) bedient sich als einzige Methode des Ontology Engineering bereits bestehender und akzeptierter Prozesse und Technologien. Die Basis des Prozessmodells liefern der Unified Process (UP) (Jacobson, M., Jonson, & Övergaard, 1992) aus dem Software Engineering, sowie die Unified Modeling Language (UML) zum Ontologieentwurf. Somit sollen Eintrittsbarrieren verhindert sowie die Lernkurve der beteiligten Personen verkürzt werden. Insbesondere hier liegt der Vorteil zu *METHONTOLOGY*, das in etwa dieselben Entwicklungsprozesse proklamiert (De Nicola, Missikoff, & Navigli, 2009, S. 273). Ähnlich zu *METHONTOLOGY* wird der Prozess in einer hohen Detailtiefe beschrieben, beinhaltet jedoch keine Aussagen zu unterstützenden Aktivitäten des Projektmanagements.

Die im Rahmen dieses Kapitels vorgestellten Methodiken werden im Weiteren einer Vergleichsanalyse unterzogen. Grundlage der Bewertung bildet ein auf Basis des IEEE 1074-1995 Standards für Lebenszyklen der Softwareentwicklung (IEEE, 1996) entwickeltes Rahmenwerk. Bereits Fernández und Gómez-Pérez (Fernandez & Gomez-Perez, 1997, S. 109 ff.) als auch De Nicola et al. (De Nicola, Missikoff, & Navigli, 2009, S. 272 ff.) beschreiben die Tauglichkeit des Rahmenwerks für die Bewertung der Ansätze des Ontology Engineering. Tabelle 6 zeigt die Ergebnisse der auf dem IEEE 1074-1995 Standard basierenden Vergleichsanalyse auf. Analog zu De Nicola et al. (De Nicola, Missikoff, & Navigli, 2009, S. 272 ff.) erfolgt eine Bewertung in durch die Methodik *unterstützte Kriterien* (●), *nicht*

IEEE 1074-1995 Standardprozesse			CommonKADS	Grüniger und Fox	METHONTOLOGY	On-To-Knowledge	DILIGENT	DynamOnt	UPON
Projektmanagement-Prozesse	Projektinitiierung			●		●			
	Monitoring und Kontrolle		●	●	●	●			
	Qualitätsmanagement		●		●			●	●
Ontologieentwicklungsorientierte Prozesse	Vor-Entwicklung	Umweltstudie						●	●
		Machbarkeitsstudie				●		●	
	Entwicklung	Anforderungen	●	●	●	●	●	●	●
		Design	●	●	●	●	●		●
		Implementierung	●	●	●	●			●
	Nach-Entwicklung	Installation							
		Betrieb							
Support									
Wartung				●	●	●		●	
Stilllegung									
Integrale Prozesse	Wissenserfassung			●			●		●
	Evaluation	●		●	●	●	●	●	●
	Konfigurationsmgmt.			●					
	Dokumentation			●		●			●
	Training								●

Tabelle 6: Vergleichsanalyse von Methoden des Ontology Engineering auf Basis IEEE 1074-1995

unterstützte Kriterien ( ) und teilweise unterstützte Kriterien (●). Eine teilweise Unterstützung liegt dann vor, wenn das Kriterium in der Methodik zwar erwähnt, aber nur ungenügend erläutert wird oder die Praxistauglichkeit nur bedingt möglich ist.

Aus der Übersicht wird deutlich, dass keine bestehende Methode die geforderten Kriterien zur Gänze erfüllt. Stattdessen setzen die Methoden unterschiedliche Schwerpunkte. Während On-To-Knowledge seinen Fokus auf die begleitenden Prozesse des Projektmanagements legt, unterstützt DILIGENT mit seiner Argumentationsmethodik primär die Wissensakquise und Anforderungserhebung in verteilten Gruppen. Das Klassifikationsmodell von DynamOnt ermöglicht insbesondere detaillierte Untersuchungen im Rahmen der Vor-Entwicklung und lässt andere Gesichtspunkte außer Acht. Bei den allgemeinen Methodiken zeichnen sich insbesondere METHONTOLOGY und UPON durch eine breite Abdeckung der geforderten Kriterien aus. UPON besitzt hierbei den Vorteil, auf bestehenden und weitverbreiteten Methoden und Technologien aufzubauen, um Eintrittsbarrieren zu verhindern.

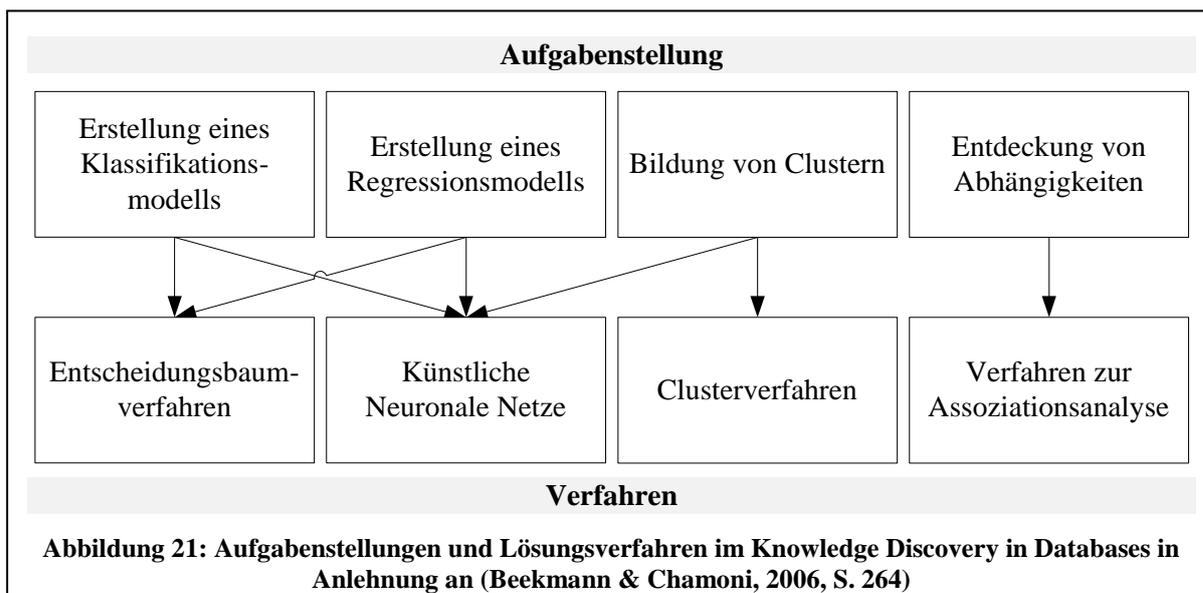
#### 3.2.5 Zusammenfassung

Im Rahmen des vorliegenden Kapitels wurde auf semantische Technologien eingegangen, die in der Lage sind, Daten mitsamt deren Bedeutung zu hinterlegen und somit nicht nur explizites Wissen erfassen, sondern auch implizites Wissen ableiten zu können. Somit sind sie grundsätzlich in der Lage, die Zielsetzung *UI-2* des in Kapitel 2.4 aufgestellten Handlungsbedarfs abzudecken. Fuzzy-Ontologien verbinden die Vorteile der Fuzzy-Logik und semantischer Technologien miteinander. Insbesondere wurde der Ansatz von Bobillo & Straccia (Bobillo & Straccia, 2011) hervorgehoben, der bestehende Standards von Sprachen zur Wissensrepräsentation nutzt, um die unscharfe Informationen darzustellen. Die Ableitung von Schlussfolgerungen ist bislang jedoch nur bedingt möglich. Zur strukturierten Entwicklung von Ontologien wurden Methoden des Ontology Engineering vorgestellt und innerhalb eines IEEE-basierten Rahmenwerks kritisch beleuchtet. Dabei wurden insbesondere die Methoden UPON und On-To-Knowledge hervorgehoben, jedoch auch auf deren Schwachstellen eingegangen. Die Methoden bieten einen Vorgehensrahmen zur Unterstützung bei der Einführung der Methodik im Unternehmen und befriedigen somit Handlungsbedarf *UI-3*.

Die identifizierten Konzepte für Fuzzy-Ontologien als auch Methoden des Ontology Engineering sind im Konzeptteil geeignet anzupassen, um die Zielstellung der vorliegenden Arbeit sicherzustellen. Während der Ansatz nach Bobillo & Straccia (Bobillo & Straccia, 2011) bislang keine Schlussfolgerungsprozesse ermöglicht, bilden die analysierten Methoden des Ontology Engineering nicht alle geforderten Bedingungen des IEEE-Frameworks ab.

### 3.3 Knowledge Discovery in Databases

*Knowledge Discovery in Databases (KDD) ermöglicht die Analyse eines Datenbestands mit dem Ziel, implizit vorhandenes Wissen zu entdecken und explizit zu machen (Düsing, 2006, S. 242). Im Kontext der vorliegenden Arbeit können somit Zusammenhänge zwischen Kunden- und Anbietersicht identifiziert und als neue Anforderungen in die Entwicklungsprozesse von Lösungsanbietern rückgeführt werden, um Hauptziel H2 der Arbeit abzudecken. Im folgenden Abschnitt 3.3.1 werden zunächst Grundlagen des KDD erläutert und darauf basierend in Abschnitt 3.3.2 auf Verfahren der Assoziationsanalyse eingegangen. Diese Verfahren ermöglichen die Extraktion signifikanter Abhängigkeiten zwischen den Objekten des Datenbestandes, sind jedoch in der Regel auf scharfe Daten beschränkt. Assoziationsanalysen mit dem Fokus auf unscharf artikuliertes Kundenwissen werden daher zusätzlich in Abschnitt 3.3.3 diskutiert und im Anschluss einem kritischen Fazit unterzogen.*



### 3.3.1 Grundlagen

Knowledge Discovery in Databases (KDD) ist ein Ansatz der Datenanalyse und zielt darauf ab, in umfangreichen Datenbeständen implizit vorhandenes Wissen zu entdecken und explizit zu machen (Düsing, 2006, S. 242). Insbesondere werden Beziehungsmuster ermittelt, die für einen möglichst großen Anteil der Datenbasis Geltung haben und leicht verständliche Zusammenhänge in den Daten zum Ausdruck bringen. Während Düsing (Düsing, 2006, S. 245) noch zwischen KDD als Gesamtprozess und Data Mining als einer Phase des KDD-Prozesses unterscheidet, verwenden heute die meisten Autoren beide Begriffe synonym. Als grundlegende Eigenschaften des KDD gelten einerseits die Generierung von Hypothesen im Verlauf der Analyse und andererseits deren anschließende Verifizierung im Rahmen eines allgemeinen Beweisverfahrens. Das im Rahmen des KDD entdeckte und explizit gemachte Wissen kann in der Folge zum Erreichen der Wissensziele eines Unternehmens, wie beispielsweise die Gewinnung eines besseren Kundenverständnisses, beitragen.

Die Wissensziele eines Unternehmens können sich je nach Aufgabenstellung stark voneinander unterscheiden, weshalb auch die möglichen Verfahren zur Lösung der Wissensziele stark differieren. In der Literatur existiert keine allgemein anerkannte Einteilung aller zugrundeliegenden Aufgabenstellungen und Lösungsverfahren, ein vielzitiertes Klassifikationsmodell wird jedoch von Beekmann und Chamoni (Beekmann & Chamoni, 2006, S. 264) vorgestellt. Das Modell klassifiziert in vier mögliche Aufgabenstellungen, welche (1) die Erstellung eines Klassifikationsmodells, (2) die Erstellung eines Regressionsmodells, (3) die Bildung von Clustern und (4) die Entdeckung von Abhängigkeiten beinhalten. Abbildung 21

stellt das Klassifikationsmodell grafisch dar. Die vier Aufgabenstellungen sowie deren Ziele sollen im Folgenden kurz vorgestellt werden.

Ziel bei der Erstellung eines Klassifikationsmodells ist es, eine Zuordnung von Elementen in vordefinierte Klassen zu ermöglichen. Als typisches Beispiel wird häufig eine Kreditwürdigkeitsprüfung angegeben (Hippner & Wilde, 2008, S. 213). Die zweite Aufgabenstellung beinhaltet die Erstellung eines Regressionsmodells. Ziel eines solchen Modells ist es, Beziehungen zwischen einer abhängigen und einer oder mehreren unabhängigen Variablen festzustellen. Im Unterschied zur Klassifikation zielt die Regression auf stetige Variablen ab, es kann also nicht mehr von einzelnen Klassen ausgegangen werden. Typische Beispiele sind Prognosen zukünftiger Verkaufswerte im Rahmen einer Zeitreihenanalyse. Bei der Bildung von Clustern als drittes Aufgabengebiet werden die Elemente derart in Cluster zusammengefasst, dass die Elemente eines Clusters möglichst ähnlich, Elemente aus unterschiedlichen Clustern jedoch möglichst abweichend sind. Die Gruppen sind dabei im Gegensatz zu Klassen nicht a priori bekannt, sondern stellen das Ergebnis des Clusterverfahrens dar. Ein typisches Beispiel für Anwendungen der Clusterentdeckung befindet sich im Marketing wie beispielsweise die Individualisierung der Kundenansprache. Das vierte Aufgabengebiet befasst sich mit der Entdeckung von Abhängigkeiten zwischen einzelnen Elementen eines Datenbestandes. Zumeist wird die Warenkorbanalyse als typisches Beispiel für Assoziationsanalysen herangezogen (Hagemann, 2008, S. 11). Eine Möglichkeit der Abhängigkeitsentdeckung besteht im Auffinden von Assoziationsregeln zwischen einzelnen Attributwerten der Datensätze. Solche Verfahren zur Assoziationsanalyse sollen im Folgenden im Detail erläutert werden, um das entdeckte Wissen als neue Anforderungen in die Entwicklungsprozesse von Lösungsanbietern rückführen zu können.

#### 3.3.2 Herkömmliche Verfahren der Assoziationsanalyse

Als Verfahren der Assoziationsanalyse kommen verschiedene Assoziationsalgorithmen zum Einsatz, die häufig gemeinsam auftretende Objekte aus einem Datenbestand extrahieren. Das Ergebnis der Anwendung solcher Algorithmen sind Assoziationsregeln, die signifikante Abhängigkeiten zwischen den untersuchten Objekten des Datenbestandes beschreiben. Im Zusammenhang mit Assoziationsanalysen werden die in der Datenbasis auftretenden Objekte häufig als Items bezeichnet. Eine beispielhafte Assoziationsregel kann wie folgt aussehen:

*Wenn Wein und Trauben gekauft wurden, wurde in 90% der Fälle auch Käse gekauft.*

Tabelle 7 gibt eine Übersicht über sämtliche gängigen Verfahren der Assoziationsanalyse in Anlehnung an (Petersohn, 2005, S. 103). Alle Verfahren weisen in ihrer Struktur einen ähnlichen

Algorithmus	Autor	Jahr
AIS	Agrawal, Imielinski, Swami	1993
SetM	Houtsma, Swami	1993
Apriori-Klasse	Agrawal, Srikant	1994
Partition	Savasere, Omiecinski, Navathe	1995
Sampling	Toivonon	1996
DHP	Park, Chen, Yu	1997
DIC	Brin, Motwani, Ullman, Tsur	1997
FP-Growth	Han, Pei, Yin	2000

**Tabelle 7: Verfahren der Assoziationsanalyse. In Anlehnung an (Petersohn, 2005, S. 103)**

Ablauf auf, der in der Folge in Kürze beschrieben wird. Vor der Beschreibung des grundlegenden Ablaufs werden zum besseren Verständnis zunächst einige grundlegende Deklarationen durchgeführt.

Es sei  $I = \{i_1, i_1, \dots, i_m\}$  die Menge aller vorkommenden Objekte, die Items genannt werden.  $\mathcal{D}$  ist eine Datenbank mit Transaktionen  $\mathcal{T}$ . Eine Transaktion  $\mathcal{T}$  besteht aus einer Menge an Items, so dass  $\mathcal{T} \subseteq I$  gilt. Die Anzahl der Items in einer Menge, auch Itemset oder Itemmenge genannt, gibt dessen Größe an. Ein Itemset der Größe  $k$  wird auch als  $k$  – *Itemset* bezeichnet.

Wichtige Leitgrößen zur Durchführung von Verfahren zur Assoziationsanalyse stellen der Support sowie die Konfidenz einer Regel dar.

Der *Support* betrifft die Anzahl an Transaktionen  $\mathcal{T}$  in der Datenbasis  $\mathcal{D}$ , die ein Itemset  $\mathcal{X}$  beinhalten. Für ein Itemset  $\mathcal{X} \subseteq \mathcal{I}$  ist der Unterstützungsgrad (Support) wie folgt definiert:

$$\text{support}_{\mathcal{D}}(\mathcal{X}) = \frac{|\{\mathcal{T} \in \mathcal{D} | \mathcal{T}, \mathcal{X} \subseteq \mathcal{T}\}|}{|\mathcal{D}|} \quad [3.5]$$

Der Support  $\text{support}_{\mathcal{D}}(\mathcal{X})$  misst also den Anteil der Transaktionen, in denen  $\mathcal{X}$  auftritt in der Menge aller Transaktionen in  $\mathcal{D}$ .

Die *Konfidenz* misst die Stärke und damit den Vertrauensgrad einer Regel und ist definiert als:

$$\text{confidence}_{\mathcal{D}}(\mathcal{X} \rightarrow \mathcal{Y}) = \frac{|\{\mathcal{T} \in \mathcal{D} | \mathcal{T}, (\mathcal{X} \cup \mathcal{Y}) \subseteq \mathcal{T}\}|}{|\{\mathcal{T} \in \mathcal{D} | \mathcal{T}, \mathcal{X} \subseteq \mathcal{T}\}|} = \frac{\text{support}_{\mathcal{D}}(\mathcal{X} \rightarrow \mathcal{Y})}{\text{support}_{\mathcal{D}}(\mathcal{X})} \quad [3.6]$$

Die Konfidenz  $\text{confidence}_{\mathcal{D}}(\mathcal{X} \rightarrow \mathcal{Y})$  misst also den Anteil der Transaktionen, in denen  $(\mathcal{X} \cup \mathcal{Y})$  auftritt, in der Menge der Transaktionen in  $\mathcal{D}$ , deren Itemsets  $\mathcal{X}$  enthalten.

Basierend auf den Deklarationen kann der grundlegende Ablauf einer Assoziationsanalyse umrissen werden. Dieser vollzieht sich in zwei Phasen:

1. Es werden alle Itemmengen gesucht, für die der Support größer als ein vom Nutzer angegebener Mindestsupport ist. Itemmengen, die den Mindestsupport erreichen, werden große Itemmengen genannt, alle anderen kleine Itemmengen.
2. Im zweiten Schritt werden die großen Itemmengen benutzt, um die gewünschten Regeln zu erzeugen. Jede große Itemmenge wird dabei in jeweils zwei disjunkte Teilmengen X und Y aufgeteilt, um Assoziationsregeln der Form  $X \rightarrow Y$  zu generieren. Gilt die Beziehung  $\text{confidence}_D(X \rightarrow Y) \geq c_{\min}$ , so wird die Regel  $X \rightarrow Y$  ausgegeben.

Auf eine detailliertere Erläuterung soll im Rahmen dieser Arbeit verzichtet werden, da die Literatur diese bereits liefert, wie beispielsweise (Beekmann & Chamoni, 2006, S. 277 ff.) (Han & Kamber, 2012, S. 559 ff.) (Petersohn, 2005, S. 104 ff.). Sämtliche vorgestellte Verfahren der Assoziationsanalyse betrachten jedoch lediglich Datenbasen mit sicheren Datenbeständen. Von Menschen geäußertes unscharfes Wissen kann mit diesen Algorithmen nicht ausgewertet werden. Daher ist in einem weiteren Schritt eine Analyse solcher Verfahren durchzuführen, die mit unscharfen Datenbeständen arbeiten und somit menschliche Aussagen in Regeln überführen können.

#### 3.3.3 Verfahren zur Assoziationsanalyse bei unscharfen Datenbeständen

Seit Kurzem werden Verfahren zur Assoziationsanalyse mit Ansätzen der Fuzzy-Logik verbunden, um Rückschlüsse auf unscharfe Aussagen treffen zu können. Insbesondere fokussieren sich die Forschungsbemühungen auf die Generierung unscharfer Assoziationsregeln, in der Literatur häufig Fuzzy-Assoziationsregeln genannt. Die meisten Ansätze beschränken sich jedoch weiterhin darauf, eine sichere Datenbasis als Datengrundlage zu nehmen und lediglich die abgeleiteten Regeln unscharf zu definieren. Exemplarisch seien die Ansätze nach Hong et al. (Hong, Kuo, & Wang, 2004) und nach Hu et al. (Hu, Chen, & Tzeng, 2003) genannt.

Forschungsbemühungen, welche eine unscharfe Datenbasis bereits als Grundlage eines Assoziationsalgorithmus verwenden, wurden bislang nur in unzureichendem Maße unternommen. Unter die wenigen vorhandenen Ansätze fällt derjenige nach Djouadi et al. (Djouadi, Redaoui, & Amroun, 2007). Darin wird die mathematische Grundlage eines Assoziationsverfahrens vorgeschlagen, das Assoziationsregeln auf Grundlage einer unscharfen Datenbasis mit Fuzzy-Zugehörigkeitsfunktionen ableiten kann. Ein vollständig implementierbares Konzept bleiben die Autoren jedoch schuldig. Weng und Chen (Weng & Chen, 2010) erweitern den Ansatz von Djouadi et al. und definieren drei potentielle Muster, in denen unscharfes Wissen vorliegen kann. Weng und Chen stellen ausführlich einen implementierbaren Algorithmus vor, der jedoch keinen Bezug auf Zugehörigkeitsfunktionen

nimmt. Beide Ansätze basieren weiterhin auf dem allgemein anerkannten zweiphasigen Ablauf von Verfahren der Assoziationsanalyse, der im vorangegangenen Abschnitt erläutert wurde.

### 3.3.4 Zusammenfassung

Verfahren der Assoziationsanalyse eignen sich grundsätzlich dazu, Kundenwissen aus großen Datenbeständen zu extrahieren und somit wie in Handlungsbedarf *H2* gefordert Kundenanforderungen zu identifizieren. In der Regel beschränken sich die Verfahren jedoch auf scharfe Datenbestände, die natürliche Unschärfe in menschlichen Ausdrücken nicht berücksichtigen. Die beiden bestehenden Verfahren zur Assoziationsanalyse bei unscharfen Datenbeständen weisen weiteren Forschungsbedarf auf, da sie entweder kein vollständig implementierbares Konzept vorlegen oder aber keine Assoziationsanalyse unter Zuhilfenahme herkömmlicher Zugehörigkeitsfunktionen zulassen.

## 3.4 Zusammenfassung

Im dritten Kapitel wurden ausgewählte Ansätze in Wissenschaft und Technik hinsichtlich ihres Zielerreichungsgrades in Bezug auf das vorliegende Forschungsvorhaben bewertet. Dabei wurden Methoden und Konzepte mit dem Ziel evaluiert, diese auf die anwendungsspezifischen Anforderungen der vorliegenden Forschungsarbeit anpassen zu können.

Im Rahmen dessen wurden zunächst Methoden der Künstlichen Intelligenz dahingehend bewertet, inwieweit sie eine Verknüpfung zwischen Kunden- und Anbietersicht unter Berücksichtigung menschlicher Unschärfe erlauben. Hierbei zeichneten sich insbesondere Fuzzy-Systeme durch die Möglichkeit aus, unscharfe umgangssprachliche Ausdrücke verarbeiten und intuitiv darstellen zu können. Menschliches Kausalwissen kann auf der Basis von Fuzzy-Systemen strukturiert in Regeln gespeichert werden. Als nachteilig wurde die fehlende Lernfähigkeit von Fuzzy-Systemen festgehalten. Da dieser Nachteil in der Folge durch Methoden des Knowledge Discovery in Databases beseitigt werden kann, liefern Fuzzy-Systeme eine gute Ausgangsbasis zur Definition einer anwendungsspezifischen Verknüpfungsrelation zwischen Eigenschaften und Merkmalen.

Wissensbasierte Systeme eignen sich dazu, implizites Wissen in Kundenaussagen erkennen zu können. Diese Eigenschaft ermöglicht die Ableitung von Schlussfolgerungen zu Kundenaussagen, die nicht explizit getroffen wurden. Basierend auf dieser Erkenntnis wurde auf Fuzzy-Ontologien eingegangen, welche die Vorteile wissensbasierter Systeme einerseits und Fuzzy-Systeme andererseits verknüpfen. Somit können reine Fuzzy-Systeme als auch Fuzzy-Ontologien zur Verknüpfung von Kunden- und Anbietersicht herangezogen werden. Die Wahl

der geeigneten Verknüpfungsrelation ist unternehmensspezifisch zu beantworten. Vorgehensrahmen, welche Unternehmen bei der Erstellung der Verknüpfungsrelation behilflich sind, liefern Methoden des Ontology Engineering. Die auf Basis eines IEEE Standards evaluierten Methoden UPON und On-To-Knowledge liefern hier eine geeignete Grundlage zur Ableitung eines anwendungsspezifischen Vorgehensmodells im Rahmen der vorliegenden Arbeit. Das Vorgehensmodell kann sowohl bei reinen Fuzzy-Systemen ohne Bezug zu wissensbasierten Systemen als auch bei Fuzzy-Ontologien zum Einsatz kommen.

Abschließend wurde auf Verfahren der Assoziationsanalyse aus dem Bereich Knowledge Discovery in Databases eingegangen. Diese erlauben es, Assoziationsregeln zwischen Objekten eines Datenbestandes zu generieren. Im vorliegenden Anwendungsfall kann diese Funktion dazu genutzt werden, die Fuzzy-Regelbasis eines Fuzzy-Systems oder einer Fuzzy-Ontologie kontinuierlich an sich wandelnde Marktbedingungen anzupassen. Die mangelnde Lernfähigkeit von Fuzzy-Systemen wird damit eliminiert. Auf der anderen Seite können durch Verfahren der Assoziationsanalyse Zusammenhänge zwischen Eigenschaften und Merkmalen aufgedeckt werden, die im Lösungsportfolio noch nicht berücksichtigt wurden. Nicht im Lösungsportfolio abgedeckte Markttrends können somit frühzeitig identifiziert und als Änderungsanfrage in die Entwicklungsabteilungen rückgeführt werden. In diesem Zusammenhang wurde insbesondere auf Assoziationsalgorithmen eingegangen, welche menschliche Unschärfe berücksichtigen. Hierbei wurde erheblicher Nachholbedarf festgestellt. Ziel ist daher die Definition eines anwendungsspezifischen Algorithmus, welcher beide oben genannte Ziele erreicht.

In Kapitel 2 wurde der grundlegende Handlungsbedarf, welche die vorliegende Forschungsarbeit motiviert, hergeleitet und unter industriellen als auch wissenschaftlichen Gesichtspunkten validiert. Darauf aufbauend analysierte das vorliegende Kapitel den bestehenden Stand in Forschung und Technik mit dem Ziel, bestehende Ansätze anwendungsspezifisch anpassen zu können aber auch Unzulänglichkeiten zu identifizieren. Somit liefert Kapitel 3 grundlegende Lösungsparadigmen zur Befriedigung der zuvor gestellten Anforderungen. Folgend wird die Methodik „Design for Customer“ vorgestellt, welche den Kern der Forschungsarbeit bildet. Darin werden teilweise die identifizierten Lösungsparadigmen anwendungsspezifisch angepasst und teilweise neue, vom Stand der Technik abgekoppelte, Lösungsansätze entwickelt.

## **4 Design for Customer – Methodik der nachhaltigen Kundenintegration**

### **4.1 Zielsetzung der Methodik**

Die Zielsetzung der im Rahmen der vorliegenden Arbeit entwickelten Methodik ‚Design for Customer‘ (DfC) ergibt sich aus dem im Grundlagenteil herausgearbeiteten Handlungsbedarf, der die grundlegende Motivation der Arbeit darstellt.

Da Kunden im Rahmen der hybriden Wertschöpfung kein Interesse nach Produkten oder Dienstleistungen per se haben, sondern vielmehr die Lösung einer Problemstellung oder eines Bedürfnisses erwarten, sind diese weitaus umfassender als im herkömmlichen Produktgeschäft in die Wertschöpfungsprozesse des Anbieters zu integrieren. Erstes Hauptziel der vorliegenden Arbeit ist daher die Entwicklung eines Ansatzes zur Kundenintegration, der sich weniger am Produkt als vielmehr am Nutzen des Kunden ausrichtet. Im Rahmen der Methodik ‚Design for Customer‘ wird dieses Ziel durch die Entwicklung eines Lösungskonfigurators verfolgt, der eine Konfiguration basierend auf Kundennutzen und Kundenbedürfnissen anstelle von Produktmerkmalen erlaubt. Grundlage hierfür bildet die Trennung in Kundeneigenschaften und Anbietermerkmale, die auch den im Grundlagenteil beschriebenen Entwicklungsmodellen von Weber et al. (Weber, Steinbach, & Botta, 2004) und Thomas et al. (Thomas, Walter, & Loos, 2008) zu eigen ist. Während Kundeneigenschaften die Bedürfnisse des Kunden in nicht-fachlicher Form beschreiben, bilden Anbietermerkmale die strukturbeschreibenden Charakteristika von Kundenlösungen ab. Dem Kunden wird es somit ermöglicht, basierend auf der Artikulation seiner Anforderungen und Bedürfnisse direkt zu einer Konfigurationsentscheidung zu kommen. Eine Kenntnis der im Lösungsportfolio enthaltenen Leistungsmerkmale wird nicht vorausgesetzt, was dem Kunden die detaillierte Kenntnis der Wissensdomäne des Anbieters erübrigt.

Als zweites Hauptziel wurde das Fehlen von Ansätzen des Anforderungsmanagements speziell für den Bereich der hybriden Wertschöpfung herausgestellt. Diese sind idealerweise nicht auf die Entwicklungsphase einer Kundenlösung beschränkt, erfolgen automatisiert und insbesondere ohne Generierung eines Mehraufwandes für den Kunden, indem bestehende Interaktionspunkte zwischen Kunde und Anbieter berücksichtigt werden. Im Rahmen der vorliegenden Arbeit wird dieses Ziel durch die kontinuierliche Rückführung von Kundenanforderungen auf Basis des zuvor geforderten Konfigurators als kontinuierliche Bezugsquelle sichergestellt. Die Kenntnis der vom Kunden artikulierten Bedürfnisse und der daraus abgeleiteten Kaufentscheidungen ermöglicht es, ein repräsentatives Bild des Kunden zu

gewinnen und bei Bedarf in die Entwicklungsprozesse des Anbieters rückzuführen. Dabei können sowohl wandelnde Markttrends aufgespürt, als auch nicht mit dem bestehenden Lösungsportfolio abgedeckte Kundenbedürfnisse identifiziert werden.

Zur Zielerreichung wurden bestehende Ansätze aus Wissenschaft und Technik herangezogen und im vorangegangenen Kapitel einer kritischen Analyse unterzogen. Auf Basis der Analyse konnte eine Menge an Ansätzen identifiziert werden, welche eine Grundlage zur Befriedigung der Handlungsbedarfe bieten. Die Ansätze finden sich im folgenden Konzeptteil der Arbeit wieder, innerhalb dessen sie erweitert und anwendungsfallspezifisch angepasst werden.

### 4.2 Methodische Schritte

Kern der vorliegenden Arbeit bildet die Methodik ‚Design for Customer‘, die aus fünf methodischen Schritten besteht und in ihrer Gesamtheit die zuvor erläuterten Handlungsbedarfe abdeckt. Der erste methodische Schritt liefert ein Vorgehensmodell, das die strukturierte Entwicklung eines bedürfnisorientierten Lösungskonfigurators erlaubt. In Schritt zwei werden zwei alternative Ansätze zur Verknüpfung von Kundeneigenschaften und Leistungsmerkmalen vorgestellt, welche das Rückgrat des Lösungskonfigurators bilden. Im dritten Schritt wird durch Methoden- und Werkzeugbaukästen des Usability Engineerings sichergestellt, dass nicht nur die methodische Basis des Ansatzes, sondern das Gesamtsystem den Bedürfnissen des Benutzers entsprechend gestaltet ist. Für die Analyse und nachgelagerte Rückführung von Kundenbedürfnissen in die Entwicklungsprozesse von Lösungsanbietern und somit ein kontinuierliches Anforderungsmanagement sorgen die Schritte vier und fünf der Methodik. Die Abfolge der methodischen Schritte der Methodik ‚Design for Customer‘ ist in Abbildung 22 ersichtlich. Die methodischen Schritte sowie die zugrundeliegende Lösungsparadigmen werden im Folgenden nochmals umrissen, bevor sie im Konzeptteil der Arbeit im Detail vorgestellt werden.

Kernstück der Methodik bildet der DfC Lösungskonfigurator, dessen grundlegender Ansatz im *zweiten Schritt* der Methodik vorgestellt wird. Dieser basiert auf der Trennung von Eigenschaften und Merkmalen und erlaubt eine Konfiguration basierend auf Kundennutzen und Kundenbedürfnissen anstelle auf strukturbeschreibenden Leistungsmerkmalen. Um Unschärfe in natürlicher Sprache verarbeiten zu können und um ein optimales Konfigurationsergebnis sicherzustellen, kommt dabei ein Fuzzy-Regler zum Einsatz. Im Rahmen der vorliegenden Arbeit werden zwei alternative Lösungsansätze zur Realisierung des Fuzzy-Reglers vorgestellt. Kapitel 4.3 ‚DfC Fuzzy-‘ beschreibt den ersten Lösungsansatz, innerhalb dessen ein herkömmlicher Fuzzy-Regler basierend auf der in Kapitel 3.1.3 ‚Fuzzy-Systeme‘ vorgestellten Architektur anwendungsfallspezifisch angepasst wurde. Dieser kann ausschließlich Unschärfe in natürlicher

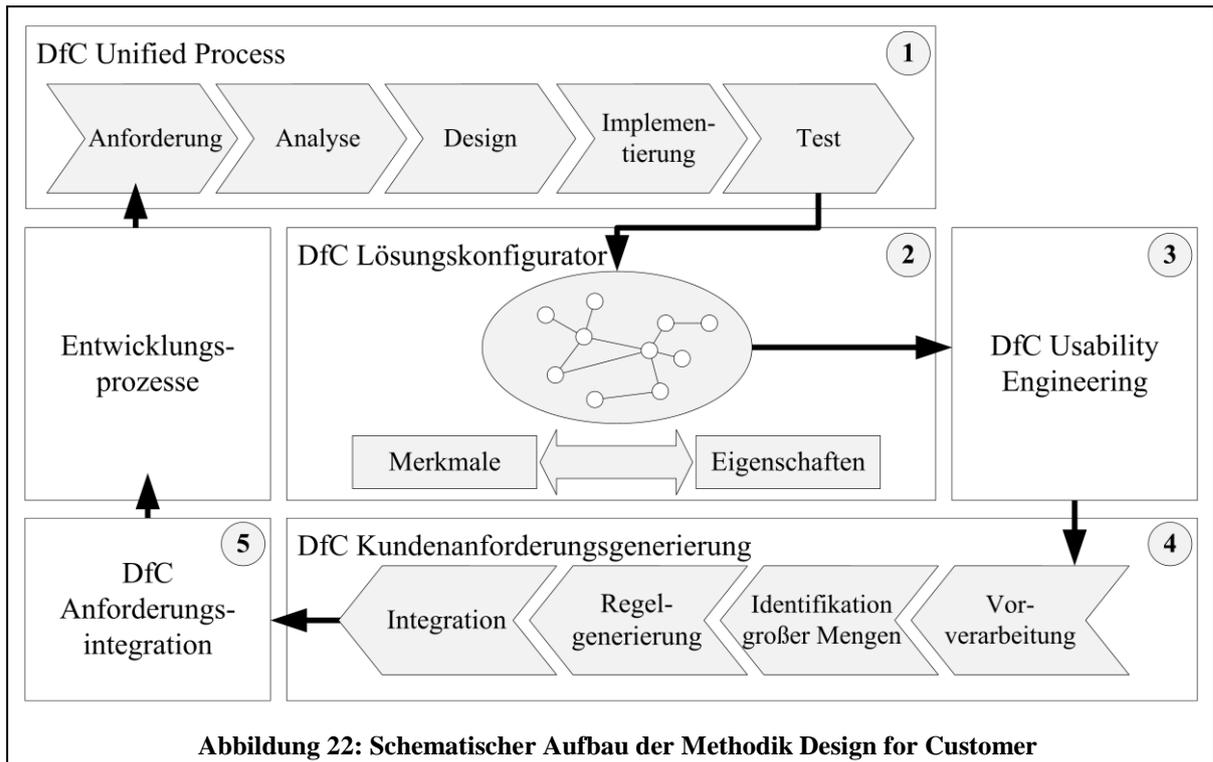


Abbildung 22: Schematischer Aufbau der Methodik Design for Customer

Sprache verarbeiten. Sollen darüber hinaus implizierte Kundenaussagen erkannt werden, liefert Kapitel 4.4 ‚DfC Fuzzy-Ontologie‘ den zugehörigen zweiten Lösungsansatz. Dadurch können auch nicht explizit geäußerte Aussagen ausgewertet und darauf aufbauend logische Schlussfolgerungen abgeleitet werden.

Um die strukturierte Entwicklung, Wartung und Pflege des Lösungskonfigurators zu ermöglichen, sind dessen Inhalte zunächst methodisch herzuleiten. Darunter fällt die Identifikation der relevanten Eigenschaften und Merkmale wie auch die Wahl der geeigneten Verknüpfungsrelation. Dieser Prozess wird durch ein strukturiertes Vorgehen sichergestellt, das den *ersten Schritt* der Methodik ‚Design for Customer‘ bildet. Hierfür wurde basierend auf der Analyse bestehender Methoden des Ontology Engineerings eine neue Methode abgeleitet, die sich der Best-Practices bestehender Methoden bedient und in Kapitel 4.5 ‚DfC Unified Process‘ vorgestellt wird.

Im *dritten Schritt* der Methodik wird sichergestellt, dass nicht nur die methodische Basis des Ansatzes, sondern das Gesamtsystem bestehend aus Mensch, Werkzeug und Aufgabe den Bedürfnissen des Benutzers entsprechend gestaltet ist. Hierfür wurden in Kapitel 4.6 ‚DfC Usability Engineering‘ Richtlinien und Baukästen entwickelt, welche den betroffenen Entscheidungsträgern Werkzeuge und Methoden zur Sicherstellung des Ziels zur Verfügung stellen. Die Grundlage hierfür liefert das Usability Engineering als wissenschaftliche und praktische Basis.

Das während der Nutzung des Konfigurators gesammelte Wissen wird in der Folge analysiert und als Kundenanforderungen in die Entwicklungsprozesse des Anbieters rückgeführt. Die entsprechende Funktionalität gewährleistet *Schritt vier* der Methodik ‚Design for Customer‘. Die Analyse des erfassten Kundenwissens beruht dabei auf Methoden und Algorithmen des Knowledge Discovery in Databases (KDD). Aus Ermangelung bestehender Algorithmen zur Ableitung von Assoziationsregeln auf Grundlage unscharfer Datenbestände wurde ein eigener Algorithmus entwickelt, der die Struktur des weitverbreiteten Apriori-Algorithmus adaptiert. Markttrends können somit identifiziert und hinsichtlich ihrer Abdeckung im Lösungsportfolio überprüft werden. Schritt vier ist in Kapitel 4.7 ‚DfC Kundenanforderungsgenerierung‘ ersichtlich.

*Schritt fünf* beinhaltet die Integration der Methodik ‚Design for Customer‘ in die Entwicklungsprozesse des Anbieters und wird in Kapitel 4.8 ‚DfC Anforderungsintegration‘ vorgestellt. Im vierten Schritt identifizierte, aber nicht im Lösungsportfolio abgedeckte Markttrends fließen dabei als Kundenanforderungen in die Entwicklungsprozesse des Anbieters. Hierfür wurde das Prozessmodell nach Weber (Steinbach, Botta, & Weber, 2005) entsprechend erweitert. Die Methodik ‚Design for Customer‘ fügt sich somit nahtlos in wissenschaftlich und praxisorientiert etablierte Ansätze ein.

Die nachfolgenden Unterkapitel gehen im Detail auf die hier nur kurz umrissenen methodischen Schritte ein und verschaffen somit ein umfassendes Bild der Methodik ‚Design for Customer‘. Dabei wird – wie in der vorangegangenen Beschreibung – zunächst Schritt 2 als zentraler Bestandteil der Methodik vorgestellt, da dieser die Grundlage für das weitere Verständnis bildet.

### 4.3 DfC Fuzzy-Controller

*Ziel des vorliegenden Unterkapitels ist die Beschreibung des DfC Fuzzy-Controllers, der den Kern des DfC Lösungskonfigurators bildet. Der Algorithmus erlaubt es, die auf Eigenschaften basierende Sprache des Kunden mit der merkmalsbasierten Sprache des Anbieters zu verknüpfen. Dabei artikuliert der Kunde innerhalb eines Konfigurationsschrittes den von ihm gewünschten Kundennutzen sowie seine Kundenbedürfnisse und erhält als Ergebnis der Konfiguration die zu den Eigenschaften passenden strukturbeschreibenden Merkmale. Ziel ist es, Unschärfe in natürlicher Sprache erkennen und verarbeiten zu können, um ein optimales Konfigurationsergebnis sicherzustellen. Zur Sicherstellung dieses Ziels kommt basierend auf der in Kapitel 3.1 vorgenommenen Analyse ausgewählter Methoden der künstlichen Intelligenz ein Fuzzy-Regler zum Einsatz.*

### 4.3.1 Grundlegende Definitionen

Das folgende Unterkapitel liefert eine mathematische Beschreibung des Fuzzy-Controllers. Vor der Beschreibung des DfC Fuzzy-Controllers sind zunächst einige grundlegende Variablendeklarationen durchzuführen. Die Variablendeklarationen basieren auf den Fachtermini herkömmlicher Fuzzy-Logik wie in der Literatur ersichtlich, siehe beispielsweise (Kramer, 2009, S. 79-88). Abgeschlossen werden die Deklarationen durch die Definition der Fuzzy-Regelbasis, welche Eigenschaften und Merkmale miteinander verbindet. Im Anschluss an die Variablendeklarationen wird ein Fuzzy-Gesamtsystem aufgestellt, das auf der in Kapitel 3.1.3 beschriebenen Architektur aufbaut und dessen Prozessmodell befolgt. Dieses Gesamtsystem ermöglicht die Konfiguration basierend auf der Trennung von Eigenschaften und Merkmalen und kann als Kern der Methodik ‚Design for Customer‘ aufgefasst werden.

#### 4.3.1.1 Modellierung der Kundeneigenschaften

Die Kundeneigenschaften repräsentieren die *linguistischen Variablen* des Fuzzy-Systems aus Kundensicht und werden als n-dimensionaler Fuzzy-Vektor

$$\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n) \quad [4.1]$$

im Vektorraum  $\mathcal{V} = \mathcal{V}_1 \times \mathcal{V}_2 \times \dots \times \mathcal{V}_n$  deklariert, wobei  $\times$  den Kartesischen Produktoperator repräsentiert. Der Gegenstandsbereich von  $\mathcal{V}$  wird durch natürliche umgangssprachliche Ausdrücke vorgegeben. Der Vektorbereich  $\mathcal{P}$  schränkt den Vektorbereich  $\mathcal{V}$  auf Aussagen zur Artikulation der firmenspezifischen Eigenschaften hybrider Leistungsbündel ein.

Eine kundenspezifische Instanz des allgemeinen Eigenschaftensvektors  $\mathcal{P}$  wird deklariert als

$$p = (p_1, p_2, \dots, p_n) \quad [4.2]$$

wobei der Vektor  $p$  dabei diejenigen Kundeneigenschaften repräsentiert, die ein Kunde als Kriterium zur Bewertung der von ihm gewünschten Leistung ansieht.



#### Verständnisbeispiel

Als Kundeneigenschaften für den Kauf von Navigationsgeräten mit zugehörigen Dienstleistungen werden mehrere linguistische Variablen  $\mathcal{P}$  angeboten. Eine linguistische Variable  $\mathcal{P}_i$  ist mit dem Wert „Exklusivität“ ausgeprägt. Wählt ein Kunde die Modellvariable aus, wird die Modellvariable  $\mathcal{P}_i$  mit dem Wert „Exklusivität“ für ihn als  $p_i$  instanziiert.

Jede Kundeneigenschaft stellt ein modelliertes Konzept des Fuzzy-Systems dar und besitzt eine vordefinierte Menge an Ausprägungen, die als *linguistische Terme* bezeichnet werden. Linguistische Terme dienen dazu, die kundenspezifische Einschätzung der jeweiligen Eigenschaft abzubilden. Für jede Kundeneigenschaft werden die linguistischen Terme  $\mathcal{L}\mathcal{T}_{i,j}$  wie folgt definiert:

**Definition 4.1 (Linguistischer Term):** Für jede Kundeneigenschaft  $\mathcal{P}_i$  gibt es  $q$  linguistische Terme  $\mathcal{L}\mathcal{T}_{i,j}$  mit  $j = 1, 2, \dots, q$ .

Im Zusammenhang mit einer instanziierten Kundeneigenschaft  $p_i$  bilden die linguistischen Terme  $\mathcal{L}\mathcal{T}_{i,j}$  Kundenaussagen der Form  $p_i \sim \mathcal{L}\mathcal{T}_{i,j}$ , welche die Haltung des Kunden in Bezug auf die Kundeneigenschaft  $p_i$  ausdrücken.

**Definition 4.2 (Kundenaussage):** Eine Kundenaussage der Form  $p_i \sim \mathcal{L}\mathcal{T}_{i,j}$  gibt die Erwartungshaltung des Kunden in Bezug auf die Kundeneigenschaft  $p_i$  an.



#### Verständnisbeispiel

Die Kundeneigenschaft  $p_i$  mit dem Wert „Exklusivität“ hat einen hohen Stellenwert für den Kunden. Er bewertet die Kundeneigenschaft daher mit einer linguistischen Variable  $\mathcal{L}\mathcal{T}_{i,j}$ , mit dem Wert „sehr wichtig“, welche seine Erwartungshaltung ausdrückt.

Die Kundenaussage hat damit die Form  $p_i(\text{Exklusivität}) \sim \mathcal{L}\mathcal{T}_{i,j}(\text{Sehr wichtig})$ .

#### 4.3.1.2 Modellierung der Anbietermerkmale

Aus Anbietersicht werden die Merkmale einer Kundenlösung direkt durch den zuständigen Entwickler festgelegt. Im Folgenden werden Anbietermerkmale analog zu den Kundeneigenschaften definiert.

Anbietermerkmale stellen die *linguistischen Variablen* des Fuzzy-Systems aus Anbietersicht dar und bilden einen  $m$ -dimensionalen Vektor

$$\mathcal{C} = (\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_m) \quad [4.3]$$

im Vektorraum  $\mathcal{W} = \mathcal{W}_1 \times \mathcal{W}_2 \times \dots \times \mathcal{W}_m$ , wobei  $\times$  den Kartesischen Produktoperator repräsentiert. Der Gegenstandsbereich von  $\mathcal{W}$  wird durch die Menge aller möglichen Produkt- und Dienstleistungsmerkmale vorgegeben.

Zu jeder Modellvariablen  $C_i$  existiert eine instanziierte Merkmalsvariable  $c_i$ , welche ein mit den Kundeneigenschaften  $\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n)$  verknüpft Merkmal darstellt. Der basierend auf den Kundeneigenschaften generierte Merkmalsvektor ergibt sich zu:

$$c = (c_1, c_2, \dots, c_m) \quad [4.4]$$

Analog zu den Kundeneigenschaften werden zu jedem Anbietermerkmal  $c_i$  linguistische Terme  $\mathcal{LV}_{i,j}$  mit  $j = 1, 2, \dots, r$  definiert.

**Definition 4.3 (Linguistischer Term):** Für jedes Anbietermerkmal  $c_i$  gibt es  $r$  linguistische Terme  $\mathcal{LV}_{i,j}$  mit  $j = 1, 2, \dots, r$ .

Die Verknüpfung der Anbietermerkmale  $c_i$  mit den zugehörigen linguistischen Termen formen Aussagen der Form  $c_i \sim \mathcal{LV}_{i,j}$ . Die anbieterspezifischen Aussagen geben Aufschluss über den anbieterspezifischen Sicherheitsgrad, dass Merkmal  $c_i$  aufgrund der eingegebenen Kundeneigenschaften gewählt wird.

**Definition 4.4 (Anbieteraussage):** Eine Anbieteraussage der Form  $c_i \sim \mathcal{LV}_{i,j}$  gibt in qualitativer Weise den Sicherheitsgrad an, dass Merkmal  $c_i$  aufgrund der eingegebenen Kundeneigenschaften im Rahmen der Konfiguration in die Kundenlösung einfließt.



#### Verständnisbeispiel

Anbietermerkmale umfassen diejenigen Sach- oder Dienstleistungskomponenten, welche vom Kunden als Modul innerhalb der Kundenlösung konfiguriert werden können. Dies kann beispielsweise das Produktmerkmal „Echtglas-Oberfläche“ beinhalten. Die Anbieteraussage  $c_i \sim \mathcal{LV}_{i,j}$  gibt den Sicherheitsgrad, mit der das Merkmal vom Kunden konfiguriert wird, als linguistischen Term an.

Ist die Konfiguration des Produktmerkmals „Echtglas-Oberfläche“ mit einem hohen Sicherheitsgrad verbunden, so hat die Anbieteraussage die Form:  $c_i$  (Echtglas-Oberfläche)  $\sim \mathcal{LV}_{i,j}$  (Sehr hoch)

#### 4.3.1.3 Modellierung der Fuzzy-Mengen und Zugehörigkeitsfunktionen

Zur Mathematisierung der sprachlichen Konzepte werden *Fuzzy-Mengen* eingeführt. Grundidee ist es, die für die Definition einer Menge verwendete charakteristische Funktion in ihrem Wertebereich zu ändern und auf das Einheitsintervall  $[0,1]$  zu projizieren.

**Definition 4.5 (Fuzzy-Menge):** Eine Fuzzy-Menge  $\mu$  von  $\Omega$  ist eine Funktion von der Referenzmenge  $\Omega$  in das Einheitsintervall:  $\mu: \Omega \rightarrow [0,1]$ .

Die charakteristische Funktion  $\mu$  wird dabei als *Zugehörigkeitsfunktion* bezeichnet.

**Definition 4.6 (Zugehörigkeitsfunktion):** Eine Zugehörigkeitsfunktion  $\mu_{\mathcal{L}\mathcal{T}_{i,j}}$  existiert zu jeder Kundenaussage der Form  $p_i \sim \mathcal{L}\mathcal{T}_{i,j}$  sowie zu jeder Anbieteraussage der Form  $c_i \sim \mathcal{L}\mathcal{V}_{i,j}$ . Eine Zugehörigkeitsfunktion weist jedem zu einer Kundeneigenschaft  $p_i$  gehörigen linguistischen Term  $\mathcal{L}\mathcal{T}_{i,j}$  Zugehörigkeitswerte  $\mu_{\mathcal{L}\mathcal{T}_{i,j}}(x)$  zu.

Für Zugehörigkeitsfunktionen  $\mu_{\mathcal{L}\mathcal{T}_{i,j}}$  gibt es keine Einschränkungen hinsichtlich Grundmenge oder Funktionsverlauf (Witte, 2002, S. 53f). Die gängigsten Formen der Zugehörigkeitsfunktionen beinhalten die Dreiecksfunktion, die Trapezfunktion sowie die nicht-linearen S-, Z- und P-Funktionen. Im Rahmen dieser Arbeit werden lediglich Dreiecksfunktionen, Linksschulter-Funktionen und Rechtsschulter-Funktionen betrachtet, die im Folgenden definiert werden:

**Dreiecksfunktion**

$$\mu_{\mathcal{L}\mathcal{W}}(x, m, \alpha, \beta) = \begin{cases} 0 & x \leq (m - \alpha), x \geq (m + \beta) \\ \frac{x - (m - \alpha)}{\alpha} & (m - \alpha) < x \leq m \\ \frac{(m + \beta) - x}{\beta} & m < x < m + \beta \end{cases} \quad [4.5]$$

**Linksschulter-Funktion**

$$\mu_{\mathcal{L}\mathcal{W}}(x, m, \beta) = \begin{cases} 1 & x \leq m \\ \frac{(m + \beta) - x}{\beta} & m < x \leq (m + \beta) \\ 0 & m + \beta < x \end{cases} \quad [4.6]$$

**Rechtsschulter-Funktion**

$$\mu_{\mathcal{L}\mathcal{W}}(x, m, \alpha) = \begin{cases} 0 & x < (m - \alpha) \\ \frac{x - (m - \alpha)}{\alpha} & (m - \alpha) < x \leq m \\ 1 & x \geq m \end{cases} \quad [4.7]$$

Zugehörigkeitsfunktionen und Fuzzy-Mengen zu einer Anbieteraussage  $c_i \sim \mathcal{L}\mathcal{V}_{i,j}$  sind analog zu den hier vorgenommenen Deklarationen einer Kundenaussage  $p_i \sim \mathcal{L}\mathcal{T}_{i,j}$  definiert.

**4.3.1.4 Modellierung der Fuzzy-Regelbasis**

Basierend auf den Deklarationen der Kundeneigenschaften und Anbietermerkmale wird die Definition einer Fuzzy-Regelbasis hergeleitet. Die Fuzzy-Regelbasis dient dazu, unter Berücksichtigung natürlicher Unschärfe Kundeneigenschaften  $p_i$  mit Anbietermerkmalen  $c_i$  zu

verknüpfen. Die Regelbasis besteht aus  $s$  Regeln  $\mathcal{R}_k$  ( $k = 1, 2, \dots, s$ ). Eine allgemeine Fuzzy-Regel  $\mathcal{R}_k$  bezieht sich auf ein spezifisches Anbietermerkmal  $c_i$  mit linguistischen Termen  $\mathcal{LV}_{i,j}$  ( $j = 1, 2, \dots, r$ ). Über die Regel  $\mathcal{R}_k$  wird das Anbietermerkmal  $c_i$  mit mehreren Kundeneigenschaften  $p_i$  und deren Bewertung durch linguistische Terme  $\mathcal{LT}_{i,j}$  ( $j = 1, 2, \dots, q$ ) verknüpft.

**Definition 4.7 (Fuzzy-Regel):** Über die Regel  $\mathcal{R}_k$  wird das Anbietermerkmal  $c_i$  mit mehreren Kundeneigenschaften  $p_i$  und deren Bewertung durch linguistische Terme  $\mathcal{LT}_{i,j}$  ( $j = 1, 2, \dots, q$ ) verknüpft. Eine Regel  $\mathcal{R}_k$  kann ausgedrückt werden als

$$\mathcal{R}_k: \text{Wenn } p_1 \text{ ist } \mathcal{LT}_{1,1} \text{ und } p_2 \text{ ist } \mathcal{LT}_{2,2} \text{ und } \dots p_n \text{ ist } \mathcal{LT}_{n,n}, \text{ dann } c_i \text{ ist } \mathcal{LV}_{i,j} \quad [4.8]$$

Die Prämisse der Regel  $\mathcal{R}_k$  beinhaltet  $n$  Kundeneigenschaften  $p_i$  mitsamt deren Ausprägungen durch linguistische Terme  $\mathcal{LT}_{i,j}$  ( $j = 1, 2, \dots, q$ ). Die Konklusion der Regel ergibt sich aus genau einem Anbietermerkmal  $c_i$  sowie des zugehörigen linguistischen Terms  $\mathcal{LV}_{i,j}$  ( $j = 1, 2, \dots, r$ ).



#### Verständnisbeispiel

Eine beispielhafte Regel  $\mathcal{R}_k$  bei der Konfiguration eines Navigationsgerätes kann wie folgt aussehen:

*Wenn „Exklusivität“ ist „wichtig“ und „Geländetauglichkeit“ ist „sehr wichtig“, dann „Echtglas-Oberfläche“ ist „hoch“.*

Über die Regelbasis wird in qualitativer Weise dem Sicherheitsgrad Ausdruck verliehen, dass ein Merkmal vom Kunden konfiguriert wird, wenn gewisse Kundeneigenschaften gewählt wurden. Die Regelbasis ist initial vom Unternehmen zu befüllen.

### 4.3.2 Architektur des DfC Fuzzy-Controllers

Die im bisherigen Verlauf dieses Kapitels erfolgten Definitionen liefern die Grundlage für den Aufbau eines Fuzzy-Reglers, welcher der Architektur aus Kapitel 3.1.3 entspricht. Die vom Kunden gelieferten Eingangswerte werden im Prozessschritt der Fuzzifizierung in Fuzzy-Mengen überführt. Der Inferenzschritt besteht aus den Einzelschritten Aggregation, Implikation und Akkumulation, innerhalb derer die Eingangswerte auf Grundlage der Fuzzy-Regelbasis in fuzzifizierte Ausgangswerte transformiert werden. Um einen scharfen Ausgangswert zu erhalten, ist das Ergebnis der Inferenz im fünften Prozessschritt der Defuzzifizierung umzuwandeln. Im Folgenden werden die fünf Prozessschritte im Detail vorgestellt.

#### 4.3.2.1 DfC Fuzzy Controller Schritt 1: Fuzzifizierung

Die Quantifizierung der Kundenaussagen erfolgt innerhalb des Prozessschrittes der Fuzzifizierung. Der Kunde bewertet eine Kundeneigenschaft  $p_i$  mit einem numerischen Wert  $x$  aus einem Intervall  $[y, z]$  mit  $y, z \in \mathbb{N}$  und  $y < z$ . Im vorliegenden Fall wird das Intervall  $[0, 100]$  proklamiert, wobei eine Kundeneingabe  $x = 0$  absolutes Desinteresse und eine Kundeneingabe  $x = 100$  absolute Aufgeschlossenheit gegenüber Kundeneigenschaft  $p_i$  ausdrückt.

Die Fuzzifizierung der Kundeneingabe  $x$  zu Kundeneigenschaft  $p_i$  erfolgt, indem die Kundeneingabe  $x$  mit den zu Kundeneigenschaft  $p_i$  gehörigen Zugehörigkeitsfunktionen  $\mu_{\mathcal{L}_{i,j}}$  geschnitten und damit auf das Einheitsintervall  $[0, 1]$  projiziert wird. Der resultierende Vektor  $\mathcal{A}_i$  setzt sich zusammen aus den Zugehörigkeitswerten  $\mu_{\mathcal{L}_{i,j}}(p_i)$  sämtlicher linguistischer Terme  $\mathcal{L}_{i,j}$  zu  $p_i$ .  $\mathcal{A}_i$  ist definiert als

$$\mathcal{A}_i = \left( \mu_{\mathcal{L}_{i,1}}(p_i), \mu_{\mathcal{L}_{i,2}}(p_i), \dots, \mu_{\mathcal{L}_{i,q}}(p_i) \right) \quad (i = 1, 2, \dots, N) \quad [4.9]$$

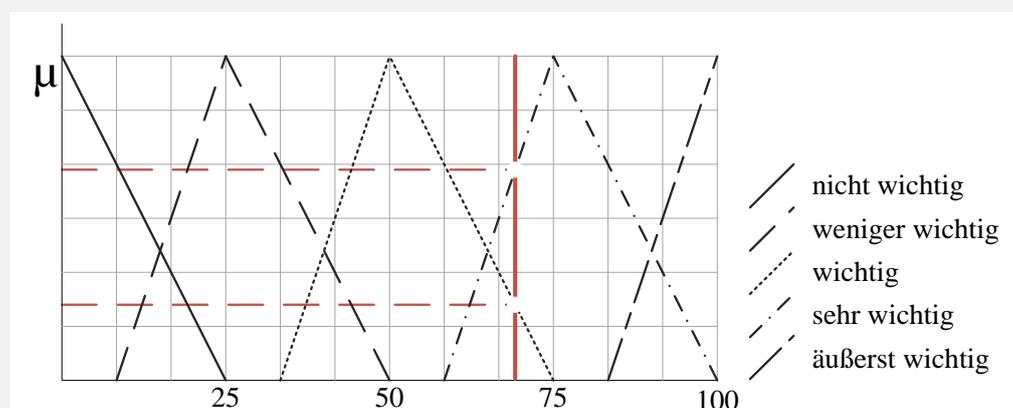
wobei  $\mu_{\mathcal{L}_{i,j}}(p_i) \ (j = 1, 2, \dots, q) \in \mathbb{R}$  aus dem Intervall  $[0, 1]$  ist. Die Variablen  $\mu_{\mathcal{L}_{i,j}}(p_i)$  repräsentieren den Sicherheitsgrad der Kundeneigenschaft  $p_i$  in quantitativer Weise.



#### Verständnisbeispiel

Ein Kunde möchte die Wichtigkeit der Eigenschaft „Exklusivität“ bewerten, hat allerdings lediglich eine unscharfe Vorstellung seiner Meinung. Auf einer Skala von 0 (unwichtig) bis 100 (sehr wichtig) entscheidet er sich für eine Wichtigkeit von 70.

Zur Kundeneigenschaft „Exklusivität“ sind fünf Zugehörigkeitsfunktionen  $\mu$  hinterlegt, die als Dreiecksfunktionen deklariert sind. Die Kundeneingabe, die in untenstehender Grafik auf der x-Achse aufgetragen ist, schneidet die beiden Zugehörigkeitsfunktionen „wichtig“ und „sehr wichtig“. Die Existenz zweier Schnittpunkte generiert zwei unscharfe Kundenaussagen der Form „Exklusivität ist wichtig“ sowie „Exklusivität ist sehr wichtig“.



Die Projektion auf das Einheitsintervall  $[0,1]$  erfolgt über die y-Koordinate der Schnittpunkte, die Aufschluss über den Sicherheitsgrad der Kundenaussagen zu Kundeneigenschaft „Exklusivität“ gibt. Die Kundenaussagen ergeben mit den zugehörigen Sicherheitsgraden in textueller Form den Vektor  $\mathcal{A}_i$  zu:

- Kundeneigenschaft  $\mathcal{P}_i$  „Exklusivität“ hat linguistische Variable  $\mathcal{LW}_{i,1}$  „wichtig“. Der Sicherheitsgrad ergibt sich zu  $\mu_{\text{Exklusivität} \sim \text{wichtig}} = 0,65$ .
- Kundeneigenschaft  $\mathcal{P}_i$  „Exklusivität“ hat linguistische Variable  $\mathcal{LW}_{i,2}$  „sehr wichtig“. Der Sicherheitsgrad ergibt sich zu  $\mu_{\text{Exklusivität} \sim \text{sehr wichtig}} = 0,23$ .

#### 4.3.2.2 DfC Fuzzy-Controller Schritt 2: Aggregation

Die Aggregation stellt die Fuzzy-Und-Verknüpfung innerhalb der Prämisse einer Fuzzy-Regel  $\mathcal{R}_k$  dar. Der Wert  $\mu_{agg,k}(\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n)$  ist das Ergebnis der Prämisse einer Regel  $\mathcal{R}_k$ , also des Wenn-Teils einer Fuzzy-Implikation. Die Verknüpfung der verschiedenen Fuzzy-Vektoren  $\mathcal{A}_i$  zu den linguistischen Variablen  $\mathcal{P}_i$  erfolgt über die Wahl eines geeigneten Aggregationsoperators, der als t-Norm bezeichnet wird.

**Definition 4.8 (t-Norm):** Eine Funktion  $T: [0,1]^2 \rightarrow [0,1]$  heißt genau dann t-Norm, wenn für alle  $a, b, c \in [0,1]$  gilt (Witte, 2002, S. 51):

- |       |  |                     |
|-------|--|---------------------|
| (i)   | $T(a, 1) = a$                            | (neutrales Element) |
| (ii)  | $T(a, b) = T(b, a)$                      | (Kommutativität)    |
| (iii) | $a \leq b \implies T(a, c) \leq T(b, c)$ | (Monotonie)         |
| (iv)  | $T(a, T(b, c)) = T(T(a, b), c)$          | (Assoziativität)    |

Die Wahl des Aggregationsoperators ist dabei abhängig von der Wahl der allgemeinen Fuzzy-Logik. Hierbei werden die drei Basis-Fuzzy-Logiken Gödel, Łukasiewicz und Produkt unterschieden. Deren Wichtigkeit lässt sich aus der Tatsache ableiten, dass jede stetige t-Norm aus einer Kombination der t-Normen der drei Logiken hergeleitet werden kann. Zudem ist es üblich, die von Zadeh vorgeschlagene Logik als Kombination der drei Basislogiken zu erwähnen, da diese die häufigste Nutzung erfährt.

Im vorliegenden Anwendungsfall wird die Fuzzy-Logik nach Zadeh angewandt, die Anpassung des Reglers an andere Fuzzy-Logiken ist jedoch ohne weiteres möglich, indem der zugrundeliegende Aggregationsoperator gewählt wird. Durch die Wahl der Zadeh-Fuzzy-Logik ist der Minimum-Operator als t-Norm für die innerhalb der Aggregation ausgeführte Konjunktion zu verwenden. Damit ergibt sich  $\mu_{agg,k}(\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n)$  zu:

$$\mu_{agg,k}(\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n) = \min \{ \mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n \} = \min \{ \mu_{\mathcal{LW}_{1,1}}(\mathcal{P}_1), \dots, \mu_{\mathcal{LW}_{n,q}}(\mathcal{P}_n) \} \quad [4.10]$$



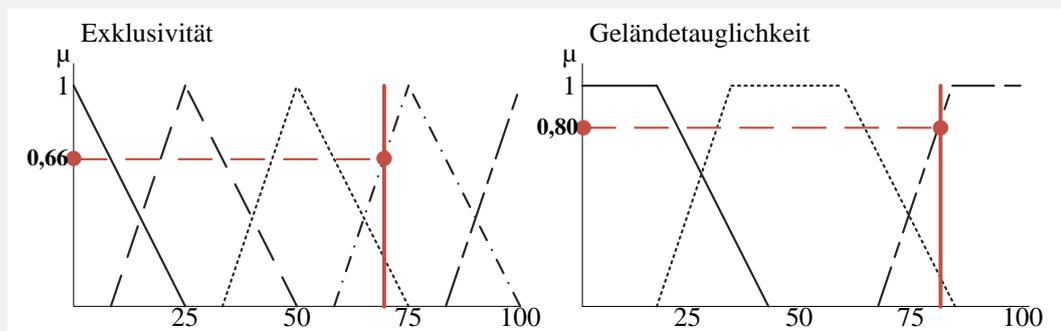
### Verständnisbeispiel

Ein Kunde bewertet im Rahmen der Konfiguration mehrere Kundeneigenschaften. Unter anderem werden über seine Eingaben die Eigenschaften „Exklusivität“ mit „wichtig“ und „Geländetauglichkeit“ mit „sehr wichtig“ bewertet.

In der Fuzzy-Regelbasis wird aufgrund der Eingaben folgende Regel angesprochen:

- Wenn „Exklusivität“ ist „wichtig“ und „Geländetauglichkeit“ ist „sehr wichtig“, dann „Echtglas-Oberfläche“ ist „hoch“.

Innerhalb des Schrittes der Fuzzifizierung wurden einzelne Kundeneigenschaften  $p_i$  mit einem Sicherheitsgrad  $\mu_{LW_i}$  bewertet, die im Vektor  $\mathcal{A}_i$  hinterlegt wurden. Im vorliegenden Schritt der Aggregation werden diese Sicherheitsgrade miteinander verknüpft. Somit erhalten wir einen quantifizierten Wert für die Prämisse einer Regel.



Im Beispiel sind durch oben angegebene Regel folgende Werte betroffen:

- $\mu_{\text{Exklusivität}} \sim \text{wichtig} = 0.66$
- $\mu_{\text{Geländetauglichkeit}} \sim \text{sehr wichtig} = 0.80$
- Der Sicherheitsgrad der Prämisse der gegebenen Regel  $\mu_{\text{agg},k}(\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n)$  ergibt sich über den Minimum-Operator zu
- $\mu_{\text{agg}}(\mathcal{P}_{\text{Exklusivität}}, \mathcal{P}_{\text{Geländetauglichkeit}}) = \min(0.80; 0.66) = 0.66$

#### 4.3.2.3 DfC FuzzyController Schritt 3: Implikation

Innerhalb des Implikationsschrittes wird das in der Aggregation ermittelte Ergebnis der Prämisse mit der Konklusion einer Regel  $\mathcal{R}_k$  verbunden. Die Verknüpfung der Prämisse der Regel mit der Konklusion erfolgt im Prämissesteil über den während der Aggregation errechneten Wert  $\mu_{\text{agg},k}(\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n)$  und im Konklusionsteil über die durch die Zugehörigkeitsfunktionen  $\mu_{LV_{i,l}}(c_i)$  ( $i = 1, 2, \dots, M$ ) beschriebene Menge  $\mathcal{M}_i$ .

$\mu_{\mathcal{LV}_{i,\ell}}(c_i) \in \mathbb{R}$  ist aus dem Intervall  $[0,1]$  und repräsentiert in quantitativer Weise den Sicherheitsgrad, dass ein Kunde Anbietermerkmal  $c_i$  konfiguriert. Mathematisch betrachtet repräsentiert  $\mu_{\mathcal{LV}_{i,\ell}}(c_i)$  den Sicherheitsgrad, zu dem Anbietermerkmal  $c_i$  zur unscharfen Menge  $\mathcal{M}_i$  gehört. Menge  $\mathcal{M}_i$  errechnet sich über eine einzelne Zugehörigkeitsfunktion  $\mu_{\mathcal{LV}_{i,\ell}}$ , welche den linguistischen Termen  $\mathcal{LV}_{i,j}$  Zugehörigkeitswerte  $\mu_{\mathcal{LV}_{i,\ell}}(c_i)$  zuweist. Mathematisch ausgedrückt erfolgt dies unter Zuhilfenahme der Integralrechnung:

$$\mathcal{M}_i = \int \mu_{\mathcal{LV}_{i,\ell}}(c_i) d\mathcal{LV}_{i,\ell} \quad [4.11]$$

Basierend auf den somit vorgenommenen Variablendeklarationen kann die Beschreibung des Verknüpfungsschrittes vorgenommen werden. Dieser erfolgt über eine horizontale Repräsentation, bei der aus der vorhandenen Fuzzy-Menge  $\mathcal{M}_i$  eine neue Menge – ein so genannter  $\alpha$ -Schnitt – gebildet wird. Jeder  $\alpha$ -Schnitt enthält dabei diejenige Teilmenge der Menge  $\mathcal{M}_i$ , die mindestens mit dem Zugehörigkeitsgrad  $\alpha$  in der Fuzzy-Menge  $\mathcal{M}_i$  enthalten ist. Der Zugehörigkeitsgrad  $\alpha$  ergibt sich aus dem innerhalb der Aggregation errechneten Wert  $\mu_{agg,k}(p_1, p_2, \dots, p_n)$ . Die entstehende Teilmenge wird als  $\mathcal{M}_i^{<\alpha}$  deklariert. Ein  $\alpha$ -Schnitt eines Fuzzy-Sets beschreibt die Teilmenge  $\mathcal{M}_i^{<\alpha} \subset \mathcal{M}_i$  für die, mit einem  $\alpha \in [0,1]$  gilt:

$$\mathcal{M}_i^{<\alpha} = \left\{ (x, \mu_{\mathcal{M}_i^{<\alpha}}(x)) \mid x \in \mathcal{M}_i \wedge \mu_{\mathcal{M}_i^{<\alpha}} = \min(\alpha, \mu_{\mathcal{M}_i}) \right\} \quad [4.12]$$

Die neue Fuzzy-Menge  $\mathcal{M}_i^{<\alpha}$  lässt sich durch die neue Zugehörigkeitsfunktion  $\mu_{\mathcal{M}_i^{<\alpha}}(x)$  beschreiben.



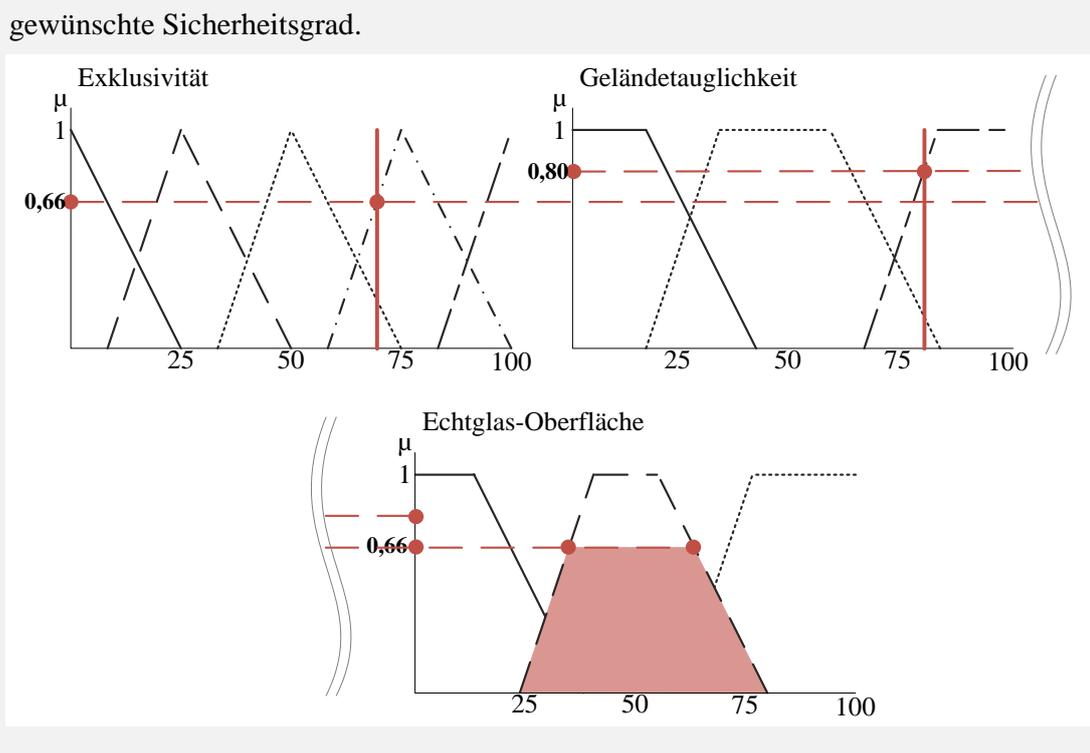
### Verständnisbeispiel

Für die Regel

- Wenn „Exklusivität“ ist „wichtig“ und „Geländetauglichkeit“ ist „sehr wichtig“, dann „Echtglas-Oberfläche“ ist „hoch“.

sollen Regelprämisse (Wenn-Teil) und Regelkonklusion (Dann-Teil) miteinander verknüpft werden. Im Aggregationsschritt wurde der Sicherheitsgrad  $\mu_{agg} = 0,66$  ermittelt. Dieser Wert wird nun als Eingangsparameter für die Implikation benötigt, um den  $\alpha$ -Schnitt auf der Zugehörigkeitsfunktion von Anbietermerkmal „Echtglas-Oberfläche“ mit linguistischer Variable „hoch“ durchführen zu können.

Dabei wird aus der Menge, welche die Zugehörigkeitsfunktion  $\mu_{\mathcal{LV}_{i,p}}$  mit der x-Achse bildet, um diejenige Teilmenge reduziert, bei der der Sicherheitsgrad größer als der Sicherheitsgrad des Inputparameters  $\mu_{agg} = 0,66$  ist. Somit wird sichergestellt, dass die kundenspezifische Wahrscheinlichkeit, dass Anbietermerkmal „Echtglas-Oberfläche“ zur Konfiguration angeboten wird, nicht größer ist als der vom Kunden



#### 4.3.2.4 DfC Fuzzy-Controller Schritt 4: Akkumulation

Die Verbindung aller  $k$  Regeln beziehungsweise derer Implikationsergebnisse  $\mu_{k,c_i}(\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n, c_i)$  erfolgt über die Wahl eines geeigneten Disjunktionsoperators. Der Disjunktionsoperator – auch t-Conorm genannt – ist wie der Aggregationsoperator abhängig von der Wahl der zugrundeliegenden Fuzzy-Logik.

**Definition 4.9 (t-Conorm):** Eine Funktion  $\perp : [0,1]^2 \rightarrow [0,1]$  heißt genau dann t-Conorm, wenn für alle  $a, b, c \in [0,1]$  gilt (Witte, 2002, S. 51):

- (i)  $\perp(a, 0) = a$  (neutrales Element)
- (ii)  $\perp(a, b) = \perp(b, a)$  (Kommutativität)
- (iii)  $a \leq b \Rightarrow \perp(a, c) \leq \perp(b, c)$  (Monotonie)
- (iv)  $\perp(a, \perp(b, c)) = \perp(\perp(a, b), c)$  (Assoziativität)

Der im Rahmen des vorliegenden Regelsystems verwendeten Zadeh-Fuzzy-Logik liegt der Max-Operator als t-Conorm zugrunde. Als Ergebnis erhalten wir:

$$\mu_{akk,c_i}(\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n, c_i) = \max\{\mu_1(\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n, c_i), \dots, \mu_k(\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n, c_i)\} \quad [4.13]$$

Die Zugehörigkeitsfunktion  $\mu_{akk,c_i}$  stellt das Ergebnis der Auswertung aller vom Anbietermerkmal  $c_i$  abhängiger Regeln dar. In der Regel liefert die Zugehörigkeitsfunktion ein Maß für die Output-Variable zu Anbietermerkmal  $c_i$ , die nicht direkt weiterverwendbar ist.

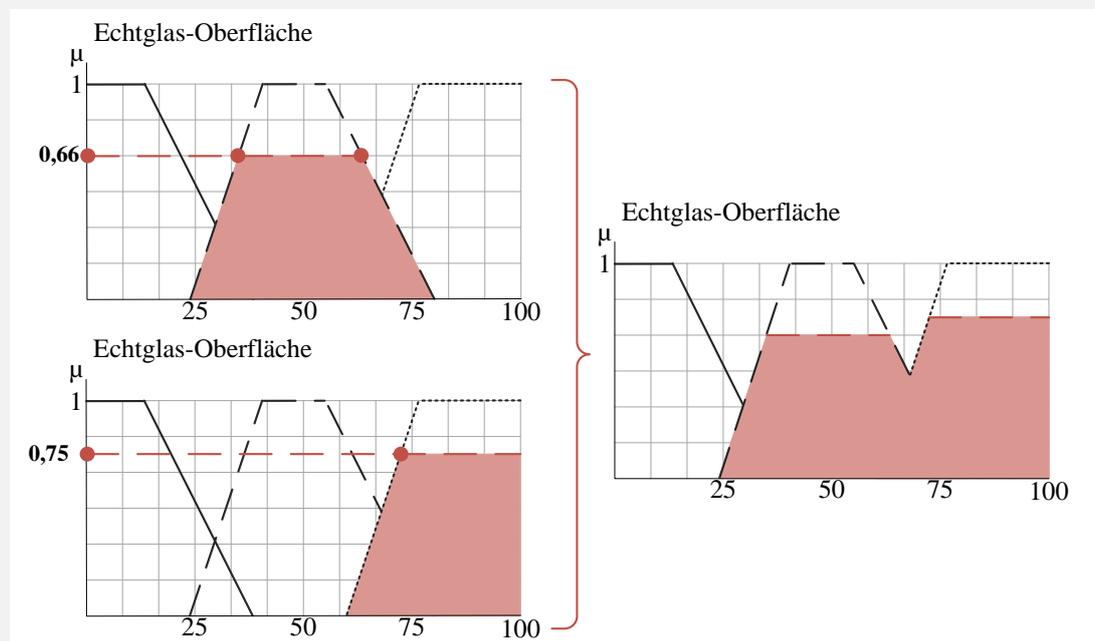


### Verständnisbeispiel

Die vom Kunden bewerteten Kundeneigenschaften lösen die beiden Regeln

- Wenn „Exklusivität“ ist „wichtig“ und „Geländetauglichkeit“ ist „sehr wichtig“, dann „Echtglas-Oberfläche“ ist „hoch“.
- Wenn „Exklusivität“ ist „wichtig“ und „Multimediatauglichkeit“ ist „sehr wichtig“, dann „Echtglas-Oberfläche“ ist „sehr hoch“.

aus. Die im Prozessschritt der Implikation erhaltenen Mengen werden im vorliegenden Schritt der Akkumulation miteinander vereinigt. Als Ergebnis erhält man  $\mu_{akk,c_i}(\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n, c_i)$ .



#### 4.3.2.5 Schritt 5 - Defuzzifizierung

Um eine eindeutige Zuordnung der Inputvariablen  $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n$  zur Output-Variablen  $c_i$  zu gewährleisten, ist das innerhalb der Akkumulation in Form einer Fläche erhaltene Ergebnis  $\mu_{akk,c_i}$  zu defuzzifizieren. Neben der Maximum-Kriterium-Methode und der Mittelwert-Maximum-Methode findet die Schwerpunkt-Methode die häufigste Verwendung. Mit letzterer wird ein glatteres Regelverhalten als mit den beiden anderen Methoden, die oftmals ein Springen des Ausgabewertes bei leichter Variation der Eingabewerte bewirken, erreicht (Kramer, 2009, S. 93). Durch Wahl der Schwerpunktmethode errechnet sich der Wert  $c_{i,res}$  wie folgt.

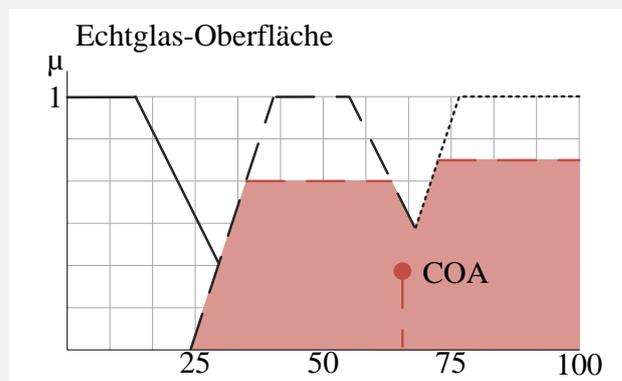
$$c_{i,res} = \frac{\int c * c_i(c) dc}{\int c_i(c) dc} = \frac{\int c * \mu_{akk,c_i}(\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n, c_i) dc}{\int \mu_{akk,c_i}(\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n, c_i) dc} \quad [4.14]$$

wobei  $c_{i,res} \in \mathbb{R}$  aus dem Intervall  $[0,1]$  ist. Die Variable  $c_{i,res}$  beschreibt die Zugehörigkeit des Anbietermerkmals  $c_i$  zu den vom Kunden eingegebenen Kundeneigenschaften  $\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n)$  in eindeutiger und quantifizierter Form. Je höher der Wert  $c_{i,res}$ , desto stärker sind die Kundeneigenschaften  $\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n)$  mit Anbietermerkmal  $c_i$  korreliert.



#### Verständnisbeispiel

Die im Akkumulationsschritt erhaltene unscharfe Menge  $\mu_{akk,c_i}(\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n, c_i)$  ist auf einen scharfen Ausgangswert zu defuzzifizieren. Dabei wird mithilfe der Integralrechnung der Schwerpunkt der Menge  $\mu_{akk,c_i}(\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n, c_i)$  ermittelt. Der Mengenschwerpunkt wird Center of Area (COA) genannt. Die auf das Einheitsintervall normierte x-Koordinate des Mengenschwerpunktes gibt Aufschluss über den Sicherheitsgrad der Konfiguration des Anbietermerkmals durch den Kunden.



#### 4.3.3 Zusammenfassung

Im vorliegenden Unterkapitel wurde ein Fuzzy-Regler vorgestellt, welcher die Verknüpfung der eigenschaftsbasierten Sprache des Kunden mit der merkmalsbasierten Sprache des Anbieters ermöglicht. Der Algorithmus unterstützt die Identifikation und Verarbeitung von Unschärfe in natürlicher Sprache. Somit ermöglicht er die Erstellung eines Lösungskonfigurators, der die Konfiguration basierend auf Kundennutzen und Kundenbedürfnissen anstelle von Produktmerkmalen erlaubt.

Die Ableitung logischer Schlussfolgerungen aus nicht explizit geäußerten Kundenwünschen ist basierend auf dem DfC Fuzzy-Controller dagegen nicht möglich. Zur Befriedigung dieser Problemstellung können semantische Technologien eingesetzt werden, die in Kapitel 3.2 analysiert wurden. Die Methodik ‚Design for Customer‘ stellt daher einen zum vorliegenden DfC Fuzzy-Controller alternativen Ansatz zur Konfiguration bereit, welcher zusätzlich die Verarbeitung impliziten Wissens zulässt.

## 4.4 DfC Fuzzy-Ontologie

*Das vorliegende Unterkapitel stellt einen zum DfC Fuzzy-Controller alternativen Ansatz zur Konfiguration bereit, der zusätzlich zur Verarbeitung von Unschärfe in natürlicher Sprache auch implizites Wissen erkennen und daraus auf die expliziten Wünsche und Bedürfnisse des Kunden schlussfolgern kann. Wie auch der zuvor vorgestellte DfC Fuzzy-Controller erlaubt die DfC Fuzzy-Ontologie, die auf Eigenschaften basierende Sprache des Kunden mit der merkmalsbasierten Sprache des Anbieters zu verknüpfen. Die DfC Fuzzy-Ontologie liefert dabei keine fest vorgegebene Ontologie, sondern eine Erweiterung der Syntax der Wissensrepräsentationssprache OWL 2. Mithilfe dieser Erweiterung können Fuzzy-Informationen in OWL 2 abgelegt und Schlussfolgerungsprozesse durchgeführt werden. Somit wird die Modellierung von projekt- und anwendungsfallabhängigen Fuzzy-Ontologien erlaubt.*

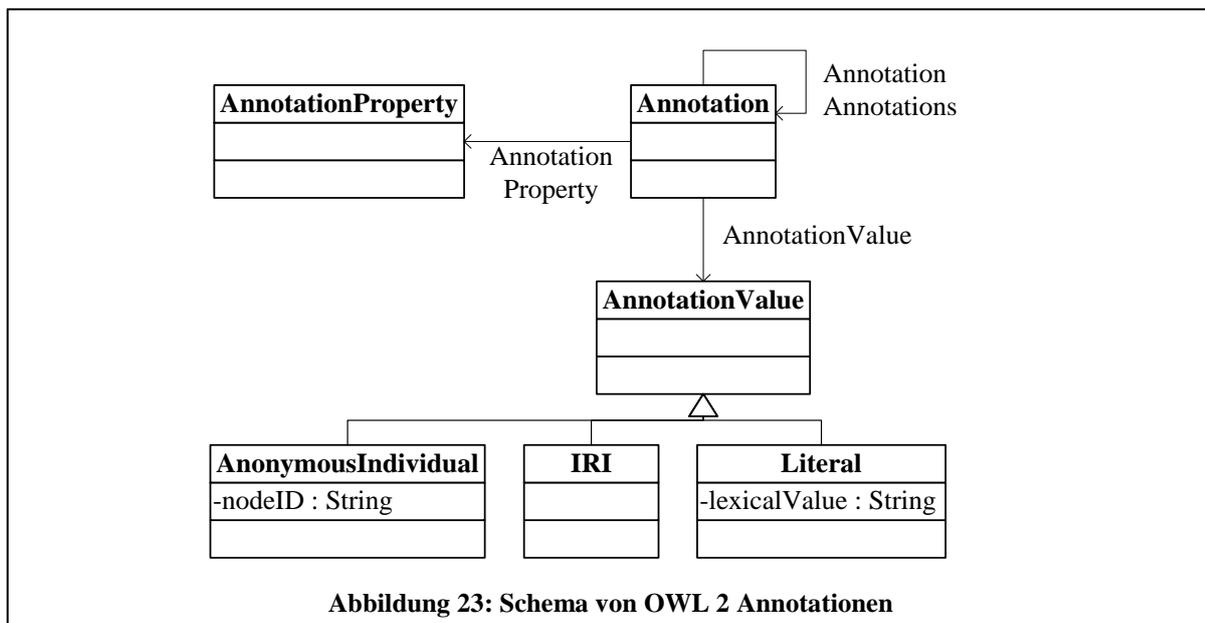
Basierend auf dem zuvor vorgestellten DfC Fuzzy-Controller werden zunächst Anforderungen identifiziert, die an die Wissensrepräsentationssprache OWL 2 zu stellen sind, um Fuzzy-Informationen darstellen und daraus Schlussfolgerungsprozesse ableiten zu können. Diese Anforderungen werden in der Folge in syntaktische Erweiterungen von OWL 2 umgewandelt. Um weiterhin bestehende Standards von OWL 2 beizubehalten werden die syntaktischen Erweiterungen als Annotation hinterlegt. Dabei wird auf Ergebnisse der Arbeit von Bobillo und Straccia (Bobillo & Straccia, 2011) zurückgegriffen.

### 4.4.1 Annotationen

Herkömmliche Sprachen des Semantic Web sind nicht in der Lage, mit vagen Informationen zu arbeiten und können somit keine Fuzzy-Informationen abbilden. Ein Standard zur Abbildung von Fuzzy-Informationen in Sprachen des Semantic Web ist ebenfalls nicht absehbar. Ziel ist es daher, bestehende Sprachen des Semantic Web zur Abbildung von Fuzzy-Logik zu verwenden und diese in der Art und Weise zu erweitern, so dass sie

- einerseits mit vagen Informationen umgehen können und
- andererseits die bestehenden Standards einhalten.

Als Vorteil einer solchen Vorgehensweise kann aufgeführt werden, dass weiterhin bestehende Werkzeuge zur Erstellung von Ontologien verwendet werden können. Desweiteren ist der Einsatz bestehender Reasoner möglich, die den Fuzzy-Teil einer Ontologie auslassen. Eine Möglichkeit zur Durchführung dieser Vorgehensweise ist die Verwendung von OWL 2 Annotationen, die dazu gedacht sind, Zusatzinformationen in Ontologien, deren Entitäten und Axiomen abzubilden. Dies können einerseits Kommentare zur besseren Verständnis der ontologischen Konstrukte durch den Menschen sein, andererseits können in Annotationen



zusätzliche Informationen abgelegt werden. Der Aufbau von Annotationen in OWL 2 folgt dem in Abbildung 23 ersichtlichen Schema. Eine Annotation besteht aus einer Annotationseigenschaft (Annotation property) und einem Annotationswert (Annotation value).

In Anlehnung an die Vorgehensweise nach Bobillo und Straccia (Bobillo & Straccia, 2011) wird im Rahmen des vorliegenden Anwendungsfalls die Annotationseigenschaft *fuzzyLabel* generiert. Die Annotation der Fuzzy-Informationen erfolgt innerhalb eines Literals, das durch das Starttag `<fuzzyOWL2>` und das Endtag `</fuzzyOWL2>` begrenzt wird. Innerhalb dieser Begrenzung wird ein Attribut *fuzzyType* spezifiziert, das angibt, welche Art der Fuzzy-Information annotiert wird. Eine detaillierte Darstellung der Annotationen erfolgt im Laufe der folgenden Unterkapitel.

#### 4.4.2 Syntaktische Anforderungen

Im vorangehenden Abschnitt wurden Annotationen von OWL 2 beschrieben, die im Rahmen der vorliegenden Arbeit zur Abbildung von Fuzzy-Informationen genutzt werden. Daher soll nun analysiert werden, welche syntaktischen Anforderungen an OWL 2 sowie dessen Annotationen zu stellen sind, um Fuzzy-Informationen abbilden und darauf Schlussfolgerungsprozesse ausführen zu können. Grundlage der Analyse sind diejenigen Objekte, die bereits im DfC Fuzzy-Controller grundlegend definiert wurden und die Basis eines Fuzzy-Reglers darstellen. Dabei wird untersucht, welche Objekte innerhalb OWL 2 als Pendant zu diesen Variablen genutzt werden können. Die Objekte sind in der Folge mit geeigneten syntaktischen Zusätzen innerhalb der Annotationen zu versehen, um die benötigten Informationen abbilden zu können.

Im Speziellen wird innerhalb der Analyse das Vorhandensein linguistischer Variablen, linguistischer Terme, einer Fuzzy-Regelbasis sowie die Abbildbarkeit einer Fuzzy-Logik überprüft.

- *Linguistische Variablen* entsprechen den modellierten Konzepten innerhalb des Fuzzy-Systems. Sie können in Ontologien durch herkömmliche Konzepte abgebildet werden. Sollen gewichtete Konzepte basierend auf einer Gewichtung mehrerer atomarer Konzepte erlaubt werden, so besitzen diese Konzepte keine Entsprechung im Nicht-Fuzzy-Fall.
- *Linguistische Terme* inklusive derer Zugehörigkeitsfunktionen haben keine Entsprechung innerhalb herkömmlicher Ontologien. Sie können jedoch durch syntaktische Erweiterungen von Datentypen modelliert werden. Hierbei sind Zugehörigkeitsfunktionen und Fuzzy-Modifikatoren zu definieren.
- Eine *Regelbasis* wird in Ontologien nicht verwendet, kann jedoch mithilfe von Axiomen abgebildet werden. Die Syntax der Axiome ist dahingehend anzupassen.
- Eine Definition der verwendeten *Fuzzy-Logik* wird in Ontologien bislang nicht benötigt. Zu diesem Zweck kann die Deklaration der Ontologie mit einer Annotation erweitert werden.

### 4.4.3 Fuzzy-Erweiterungen von OWL 2

Im Folgenden werden die im vorangegangenen Abschnitt identifizierten OWL 2 Entitäten im Detail vorgestellt. Darüber hinaus wird definiert, welche Erweiterungen diese Entitäten abbilden können müssen, um die benötigten Fuzzy-Informationen verwalten und speichern zu können.

#### 4.4.3.1 Konzepte

Ein *Konzept* beschreibt eine Menge von Objekten, die gemeinsame Eigenschaften aufweisen. Konzepte werden auch als Klassen bezeichnet, die in einer Klassenstruktur mit Ober- und Unterklasse angeordnet werden können. Konzepte sind hinreichend zur Modellierung linguistischer Terme, wie in Kapitel 4.3 ‚DfC Fuzzy-Controller‘ definiert.

Wird gewünscht, dass linguistische Variablen mit unterschiedlichen Gewichten versehen werden können, so sind bestehende Konzepte zu erweitern. Eine Gewichtung von Konzepten ist dann hilfreich, wenn verschiedenen Kundeneigenschaften eine unterschiedliche Wertigkeit zugeordnet werden soll. In diesem Fall ergibt sich ein neuer Objekttyp von Konzepten für gewichtete Summen  $C_{ws}$ . Der Konstruktor des Konzeptes ergibt sich aus

$$C_{ws} \rightarrow (\alpha_1 \cdot C_1) + \dots + (\alpha_k \cdot C_k) \quad [4.15]$$

wobei  $C_i$  herkömmliche Konzepte darstellen und es gilt  $\sum_{i=1}^k \alpha_i = 1$ .

#### 4.4.3.2 Datentypen

Datentypen bezeichnen die Zusammenfassung konkreter Wertebereiche und darauf definierter Operationen zu einer Einheit. Innerhalb OWL 2 werden herkömmliche Datentypen wie beispielsweise *owl:real* oder *owl:rational* unterstützt. Ferner existiert die Möglichkeit, Wertebereiche konkreter Rollen zu beschränken. Eine Abbildung von Zugehörigkeitsfunktionen ist jedoch nicht möglich.

Wie innerhalb der Beschreibung des DfC Fuzzy-Controllers gefordert, sollen für den vorliegenden Anwendungsfall Dreiecksfunktionen, Linksschulter-Funktionen und Rechtsschulter-Funktionen als Zugehörigkeitsfunktionen unterstützt werden. Die Funktionen sind über ein Intervall  $[k_1, k_2] \subseteq \mathbb{R}$  definiert und werden über die Punkte  $a, b, c$  bei einer Dreiecksfunktion beziehungsweise  $a, b$  in den übrigen beiden Fällen charakterisiert.

Zusätzlich sollen Fuzzy-modifizierte Datentypen unterstützt werden, die über die Anwendung eines Fuzzy-Modifikators auf eine Zugehörigkeitsfunktion entstehen. Fuzzy-Modifikatoren dienen dazu, Fuzzy-Mengen zu verändern, um beispielsweise Begriffe wie *sehr*, *ziemlich* und *etwas* zu modellieren. Ein Beispiel der Verschärfung ist die Transformation der Zugehörigkeitsfunktion <hoch> auf <sehr hoch> über den Fuzzy-Modifikator <sehr>. Vorteile von Fuzzy-Modifikatoren sind deren Wiederverwendbarkeit für mehrere Fuzzy-Mengen. Ein Fuzzy-Modifikator wird dabei als Funktion  $f_{mod}: [0,1] \rightarrow [0,1]$  definiert, welche die Zugehörigkeitsfunktion einer Fuzzy-Menge transformiert. Im vorliegenden Fall soll ein linearer Fuzzy-Modifikator zur Verfügung stehen, der eine bestehende Zugehörigkeitsfunktion verschärft (Konzentrationsoperator) oder entschärft (Dilationsoperator). Der hierfür benötigte lineare Fuzzy-Modifikator *mod* wird hierfür definiert als linearer Operator:

$$mod \rightarrow linear(x, \alpha, \beta) = \begin{cases} 0 & x = 0 \\ \frac{\beta}{\alpha}x & 0 \leq x \leq \alpha \\ \frac{\beta}{\alpha} + \frac{1-\beta}{1-\alpha}x & x > \alpha \end{cases} \quad [4.16]$$

mit  $\alpha, \beta \in \mathbb{R}$  aus dem Intervall  $[0,1]$ .

Somit sind insgesamt die folgenden Zugehörigkeitsfunktionen als Datentypen zu unterstützen:

$$\begin{aligned} \mu \rightarrow left(k_1, k_2, a, b) & \quad (Linksschulter-Funktion) \\ right(k_1, k_2, a, b) & \quad (Rechtsschulter-Funktion) \\ triangular(k_1, k_2, a, b, c) & \quad (Dreiecksfunktion) \\ mod(\mu) & \quad (Fuzzy-Modifikator) \end{aligned} \quad [4.17]$$

#### 4.4.3.3 Axiome

Axiome werden in der Regel verwendet, um logische Aussagen darzustellen, die als wahr erachtet werden. Die wichtigsten in OWL 2 vorhandenen Typen von Axiomen sind Klassenaxiome (Class Axioms), Objekteigenschaftenaxiome (Object Property Axioms), Dateneigenschaftenaxiome (Data Property Axioms) und Fakten (Facts). Eine einfache und intuitive Darstellung von Implikationsregeln über die in OWL 2 vorhandenen Axiome, wie für Fuzzy-Systeme gewünscht, ist nicht möglich.

Im vorliegenden Anwendungsfall sind Regeln der Form

$\mathcal{R}_k$ : **Wenn**  $p_1$  ist  $\mathcal{L}\mathcal{T}_{1,1}$  und  $p_2$  ist  $\mathcal{L}\mathcal{T}_{2,2}$  und ...  $p_n$  ist  $\mathcal{L}\mathcal{T}_{n,n}$ , **dann**  $c_i$  ist  $\mathcal{L}\mathcal{V}_{i,j}$

abzubilden. In der Konklusion jeder Regel wird dabei auf ein Anbietermerkmal  $c_i$  verwiesen, welches als Konzept oder gewichtetes Konzept in der Ontologie abgebildet wird. Um die Zusammengehörigkeit dieses Anbietermerkmals zur Regel auch visuell zu veranschaulichen, sollen die Regeln als Unterklasse des sie implizierenden Anbietermerkmals angelegt werden. Die Verknüpfung zwischen Anbietermerkmal  $c_i$  und Regelbasis erfolgt somit über das Klassenaxiom *subClassOf*. Die Annotationen des Klassenaxioms sind geeignet anzupassen, um den Unterschied zu herkömmlichen Klassenaxiomen *subClassOf* zu verdeutlichen.

#### 4.4.3.4 Ontologien

Herkömmliche Ontologien weisen keine Informationen zur innerhalb der Ontologie verwendeten Fuzzy-Logik auf. Um die Ontologie mit der Information zu versorgen, welche Form der Fuzzy-Logik zugrunde liegt, ist die Ontologie dementsprechend zu annotieren.

### 4.4.4 Repräsentation von Fuzzy-Ontologien in OWL 2

*Der vorangegangene Abschnitt beschäftigte sich damit, diejenigen OWL 2 Objekte vorzustellen, die für die Fuzzy-Anwendung des vorliegenden Anwendungsfalles anzupassen sind. Dabei wurde beschreibend darauf eingegangen, welche Funktionen und Operationen die angepassten Objekte aufweisen müssen. Im Rahmen dieses Abschnitts wird eine Methodik vorgestellt, Fuzzy-Ontologien auf Basis von OWL 2 Syntax zu repräsentieren.*

Die im weiteren Verlauf vorgestellte Repräsentation basiert auf dem Konzept nach Bobillo und Straccia (Bobillo & Straccia, 2011) und erweitert dieses um die Möglichkeit, logische Schlussfolgerungsprozesse abzubilden. Wie auch bei Bobillo und Straccia werden herkömmliche OWL 2 Elemente verwendet und diese durch Annotationen erweitert. Zur anschaulichen Darstellung der Erweiterungen wird in der vorliegenden Arbeit die OWL/XML Syntax verwendet, welche die Ontologie als eine an die XML Notation angelehnte Baumstruktur modelliert und damit intuitiv dargestellt wird.

### 4.4.4.1 Konzepte

#### Atomare Konzepte

Kundeneigenschaften und Anbietermerkmale können wie im vorherigen Abschnitt gezeigt in den meisten Fällen als herkömmliche OWL 2 Konzepte deklariert werden. Die Deklaration von Konzepten in OWL 2 XML Syntax geschieht folgendermaßen.

*Syntax einer Klassendeklaration.*

```
<Declaration>
  <Class IRI="#<CLASSNAME>" />
</Declaration>
```

Die Variable <CLASSNAME> steht dabei für den Namen der Klasse.



#### Verständnisbeispiel

Der Begriff „Multimediatauglichkeit“ soll als Kundeneigenschaft verfügbar sein. Die notwendige Deklaration erfolgt dann über die folgende Deklaration:

```
<Declaration>
  <Class IRI="# Multimediatauglichkeit " />
</Declaration>
```

#### Gewichtete Konzepte

Gewichtete Konzepte setzen sich aus einzelnen atomaren Konzepten sowie eines Gewichtungsfaktors zusammen. Die Gewichtungsfaktoren geben an, welche Wertigkeit die einzelnen Konzepte zueinander haben. Hierfür wird ein zusätzliches Konzept erstellt, das mit einer geeigneten Annotation zu versehen ist, um die Gewichtung abzubilden. Bei der Abbildung eines gewichteten Konzepts sind mindestens zwei atomare Konzepte zu verwenden.

*Zu annotierendes Objekt:* Ein OWL 2 Konzept

*Syntax der Annotation:*

```
<fuzzyOwl2 fuzzyType="concept">
  <Concept type="weightedSum">
    (<Concept type="weighted" value="<GEWICHTUNGSFAKTOR>" base="<BASISKONZEPT>" />)+
  </Concept>
</fuzzyOwl2>
```

Die Variable <GEWICHTUNGSFAKTOR> steht für die Gewichtungsfaktoren der einzelnen Konzepte, die in Summe 1 ergeben müssen. Dabei sind mindestens zwei Konzepte als <BASISKONZEPT> zu wählen.



### Verständnisbeispiel

Die Kundeneigenschaft „Multimediatauglichkeit“ soll sich aus den vom Kunden bewerteten Kundeneigenschaften „Internetzugang“ und „Vorinstallierte Spiele“ zusammensetzen. Die Eigenschaft „Internetzugang“ soll dabei mit dem Gewichtungsfaktor 0.7 bewertet werden. Damit ergibt sich der Gewichtungsfaktor für die Eigenschaft „Vorinstallierte Spiele“ zu 0.3. Die Syntax der gesamten Annotation ergibt sich zu:

```
<AnnotationAssertion>
  <AnnotationProperty IRI="#fuzzyLabel"/>
  <IRI>#Multimediatauglichkeit</IRI>
  <Literal datatypeIRI="&rdf;PlainLiteral">
    <fuzzyOwl2 fuzzyType="concept">
      <Concept type="weightedSum">
        <Concept type="weighted" value="0.7" base="Internetzugang" />
        <Concept type="weighted" value="0.3" base="Vorinstallierte Spiele" />
      </Concept>
    </fuzzyOwl2>
  </Literal>
</AnnotationAssertion>
```

#### 4.4.4.2 Datentypen

##### Fuzzy-Datentypen

Zur Erstellung von Fuzzy-Mengen werden zugrundeliegende Zugehörigkeitsfunktionen benötigt, die in OWL 2 als das Objekt Datentyp repräsentiert werden. Die Datentypen weisen unter anderem die Parameter *min* und *max* auf, welche die untere sowie die obere Grenze des Wertebereiches abbilden.

Die weiteren Parameter der Zugehörigkeitsfunktionen *a*, *b*, *c* sind abhängig vom Typ der Zugehörigkeitsfunktion. Während Linksschulter- und Rechtsschulter-Funktionen lediglich die Parameter *a* und *b* aufweisen, benötigt die Dreiecksfunktion den zusätzlichen Parameter *c*.

*Zu annotierendes Objekt:*

Ein OWL 2 Datentyp im Wertigkeitsbereich (Restriction Base) von Ganzzahlen (Integer) oder Gleitkommazahlen (Double). Die Parameter *min* und *max* sind nicht in der Annotation, sondern direkt in der Deklaration des OWL 2 Datentyps zu hinterlegen. Beispielhaft wird die Deklaration des Parameters <MIN> dargestellt. Die Deklaration des <MAX> Parameters erfolgt analog.

```
<FacetRestriction facet="&xsd;minInclusive">
  <Literal datatypeIRI="&xsd;double"><MIN></Literal>
</FacetRestriction>
```

*Syntax der Annotation:*

Die Deklaration der Parameter *a*, *b*, *c* erfolgt innerhalb der Annotation des Datentyps.

```
<fuzzyOwl2 fuzzyType="datatype">
  <Datatype type="leftshoulder" a="PARAMETER_A" b="PARAMETER_B" /> |
  <Datatype type="rightshoulder" a=" PARAMETER_A" b=" PARAMETER_B"/> |
  <Datatype type="triangular" a=" PARAMETER_A" b=" PARAMETER_B" c=" PARAMETER_C" />
</fuzzyOwl2>
```



**Verständnisbeispiel**

Zur Modellierung des linguistischen Terms „hoch“ wird eine Zugehörigkeitsfunktion erstellt, deren Gesamtwertebereich sich im Intervall [0,100] befindet. Die Zugehörigkeitsfunktion selbst soll eine Rechtsschulter-Funktion mit den Parametern *a*=60 und *b*=80 ergeben.

Zunächst wird der Datentyp mit dem Gesamtwertebereich [0,100] erstellt:

```
<DatatypeDefinition>
  <Datatype IRI="#Hoch"/>
  <DataIntersectionOf>
    <DatatypeRestriction>
      <Datatype abbreviatedIRI="xsd:double"/>
      <FacetRestriction facet="&xsd;minInclusive">
        <Literal datatypeIRI="&xsd;double">0.0</Literal>
      </FacetRestriction>
    </DatatypeRestriction>
    <DatatypeRestriction>
      <Datatype abbreviatedIRI="xsd:double"/>
      <FacetRestriction facet="&xsd;maxInclusive">
        <Literal datatypeIRI="&xsd;double">100.0</Literal>
      </FacetRestriction>
    </DatatypeRestriction>
  </DataIntersectionOf>
</DatatypeDefinition>
```

Darauf aufbauend wird eine Annotation erstellt, welche die Rechtsschulter-Funktion mitsamt deren Parametern modelliert:

```
<AnnotationAssertion>
  <AnnotationProperty IRI="#fuzzyLabel"/>
  <IRI>#Hoch</IRI>
  <Literal datatypeIRI="&rdf;PlainLiteral">
    <fuzzyOwl2 fuzzyType="datatype">
      <Datatype type="rightshoulder" a="60.0" b="80.0" />
    </fuzzyOwl2>
  </Literal>
</AnnotationAssertion>
```

#### 4.4.4.3 Fuzzy-Modifikatoren

Fuzzy-Modifikatoren ermöglichen die Transformation von Fuzzy-Mengen, um beispielsweise Begriffe wie *sehr*, *ziemlich* und *etwas* zu modellieren. Der für den vorliegenden Fall benötigte lineare Fuzzy-Modifikator soll als OWL 2 Datentyp deklariert werden können und weist die Parameter  $a$ ,  $b$  auf. Die Grenzen des Wertigkeitsbereiches sind mit 0 und 1 gegeben und sind analog zu den Fuzzy-Datentypen zu deklarieren.

*Zu annotierendes Objekt:* Ein OWL 2 Datentyp im Wertigkeitsbereich Double

*Syntax der Annotation:*

```
<fuzzyOwl2 fuzzyType="modifier">
  <Modifier type="linear" a="PARAMETER_A" b="PARAMETER_B" />
</fuzzyOwl2>
```

Dabei stammen  $a$ ,  $b$  aus dem Intervall  $[0,1]$ .



#### Verständnisbeispiel

Der Fuzzy-Modifikator „sehr“ soll so definiert werden, dass eine Konzentration einer Zugehörigkeitsfunktion ermöglicht wird. Die Parameter werden daher mit  $a=0.7$  und  $b=0.3$  gewählt. Die entsprechende Syntax in OWL 2 XML lautet wie folgt:

```
<AnnotationAssertion>
  <AnnotationProperty IRI="#fuzzyLabel"/>
  <IRI>#sehr</IRI>
  <Literal datatypeIRI="&rdf;PlainLiteral">
    <fuzzyOwl2 fuzzyType="modifier">
      <Modifier type="linear" a="0.7" b="0.3" />
    </fuzzyOwl2>
  </Literal>
</AnnotationAssertion>
```

#### Modifizierte Fuzzy-Datentypen

Modifizierte Fuzzy-Datentypen basieren auf bestehenden Fuzzy-Datentypen einerseits und auf Fuzzy-Modifikatoren andererseits. Je nach Ausprägung des Fuzzy-Modifikators stellen sie verschärfte oder verschwommene Ausprägungen desjenigen Fuzzy-Datentypen dar, der ihnen zugrunde liegt.

*Zu annotierendes Objekt:* Ein OWL 2 Datentyp

*Syntax der Annotation:*

```
<fuzzyOwl2 fuzzyType="datatype">
  <Datatype type="modified " modifier="<FUZZY_MODIFIKATOR>" base="<FUZZY_BASIS_DATENTYP>" />
</fuzzyOwl2>
```

Der Parameter  $\langle \text{FUZZY\_MODIFIKATOR} \rangle$  gibt dabei den Fuzzy-Modifikator an, der den Fuzzy-Datentyp  $\langle \text{FUZZY\_BASIS\_DATENTYP} \rangle$  transformiert.



### Verständnisbeispiel

Zusätzlich zum zuvor definierten Fuzzy-Datentyp „hoch“ wird ein weiterer Datentyp „sehr hoch“ benötigt. Dies erfolgt durch Definition eines neuen Fuzzy-Datentypen vom Typ modifizierter Fuzzy-Datentyp. Der Datentyp hat als Basis den Fuzzy-Datentypen „hoch“ sowie den Fuzzy-Modifikator „sehr“ inne. Die zugrundeliegende Syntax setzt sich wie folgt zusammen:

```
<AnnotationAssertion>
  <AnnotationProperty IRI="#fuzzyLabel"/>
  <IRI>#sehrHoch</IRI>
  <Literal datatypeIRI="&rdf;PlainLiteral">
    <fuzzyOwl2 fuzzyType="datatype">
      <Datatype type="modified " modifier="sehr" base="hoch" />
    </fuzzyOwl2>
  </Literal>
</AnnotationAssertion>
```

#### 4.4.4.4 Axiome

Axiome ermöglichen die Darstellung logischer Aussagen, die als wahr erachtet werden, weshalb sie sich zur Abbildung der Fuzzy-Regelbasis eignen. Zur Abbildung einer Regel ist ein neues Konzept zu bilden, welches als Unterklasse desjenigen Konzeptes angelegt ist, das im Konklusionsteil der Regel auftaucht. Die Regel ist in der Folge als Annotation des zugehörigen subClassOf-Axioms zu hinterlegen.

*Zu annotierendes Objekt:* Ein OWL 2 Axiom vom Typ Class Assertion

*Syntax der Annotation:*

```
<fuzzyOwl2 fuzzyType="axiom">
  <Axiom type="implication" premis="<REGEL_PRAEMISSE>" conclusion="<REGEL_KONKLUSION>" />
</fuzzyOwl2>
```

Die Syntax besteht aus den Parametern <REGEL\_PRAEMISSE> zur Abbildung des Wenn-Teils der Regel (Prämisse) sowie <REGEL\_KONKLUSION> zur Abbildung des Dann-Teils der Regel (Konklusion). Die Prämisse der Regel formt dabei einzelne Aussagen der Form (<KONZEPT\_KUNDENEIGENSCHAFT> some <FUZZY\_DATENTYP>). Mehrere Aussagen werden mit dem Operator *And* voneinander abgegrenzt. Die Konklusion besteht aus einer einzelnen Aussage (<KONZEPT\_ANBIETERMERKMAL> some <FUZZY\_DATENTYP>).



### Verständnisbeispiel

Es soll eine neue Regel erstellt werden, die genau dann greift, wenn Kundeneigenschaft „Internetzugang“ mit „sehr hoch“ bewertet wurde und Kundeneigenschaft „vorinstallierte Spiele“ mit „hoch“. Als Ergebnis der Regel soll Anbietermerkmal „Media Player“ mit „hoch“ bewertet werden. Vollständig ausgeschrieben lautet die Regel: Wenn Internetzugang ist „Sehr hoch“ und vorinstallierte Spiele ist „Hoch“, dann Media Player ist „Hoch“.

Zunächst ist ein neues Konzept „Regel1“ anzulegen, das eine Unterklasse des Anbietermerkmals „MediaPlayer“ darstellt. In der Folge ist das *subClassOf* Axiom zu annotieren. Die zugehörige exemplarische Syntax lautet wie folgt:

```
<SubClassOf>
  <Annotation>
    <AnnotationProperty IRI="#fuzzyLabel"/>
    <Literal datatypeIRI="&rdf;PlainLiteral">
      <fuzzyOwl2 fuzzyType="axiom">
        <Axiom type="implication" premis="(Internetzugang some sehrHoch) And ...
          (VorinstallierteSpiele some hoch)" conclusion="(MediaPlayer some hoch)" />
      </fuzzyOwl2>
    </Literal>
  </Annotation>
  <Class IRI="#Regel1"/>
  <Class IRI="#MediaPlayer"/>
</SubClassOf>
```

#### 4.4.4.5 Ontologie

Zur Spezifikation der zu verwendenden Fuzzy-Logik ist die Ontologie geeignet zu annotieren. Die Annotation erfolgt im Header der Ontologie. Die Auswertung der Regelbasis erfolgt dabei nicht innerhalb der Ontologie, sondern basierend auf geeigneten Werkzeugen, so genannten Reasonern. Auf diese Reasoner wird in Abschnitt 4.4.5 eingegangen.

*Zu annotierendes Objekt:* Eine OWL 2 Ontologie

*Syntax der Annotation:*

```
<fuzzyOwl2 fuzzyType="ontology">
  <FuzzyLogic logic ="<FUZZY_LOGIK>" />
</fuzzyOwl2>
```

Der Parameter ist mit Werten „lukasiewicz“, „zadeh“, „goedel“ oder „product“ zu befüllen. Dabei ist darauf zu achten, dass der verwendete Reasoner die Logik versteht und weiterverarbeiten kann.



### Verständnisbeispiel

Die Fuzzy-Ontologie soll mit dem angeschlossenen Reasoner FUZZYDL ausgewertet werden. Dieser unterstützt unter anderem die Fuzzy-Logik nach Zadeh. Im Kopf der Ontologie ist folgende Annotation einzutragen.

```
<fuzzyOwl2 fuzzyType="ontology">  
  <FuzzyLogic logic = "<zadeh>" />  
</fuzzyOwl2>
```

#### 4.4.4.6 Zusammenfassung

Im Rahmen des vorliegenden Unterkapitels wurde auf Basis der Ontologiesprache OWL 2 die Möglichkeit geschaffen, eine Fuzzy-Wissensbasis innerhalb einer Ontologie ablegen zu können. Hierfür wurde OWL 2 um Konzepte der Unschärfe erweitert und eine Methode vorgestellt, Implikationsregeln in OWL 2 abzubilden. Der Vorteil des verwendeten Verfahrens besteht darin, dass ein existierender Standard zur Definition von Fuzzy-Ontologien zum Einsatz kommt. Der Entwicklung einer proprietären Lösung, die separate Standards benötigt, wird somit vorgebeugt. Als Träger der Fuzzy-Informationen innerhalb des OWL 2 Standards kommen Annotationen zum Einsatz. Vom W3C vorgesehener Einsatzzweck dieser Annotationen ist es, Zusatzinformationen in Ontologien, deren Entitäten und Axiomen abzubilden. Die Ontologie bleibt somit auch für herkömmliche Softwarewerkzeuge verarbeitbar, da Annotationen von diesen einfach übergangen werden können.

Die bislang vorgestellten Fuzzy-Erweiterungen von OWL 2 ermöglichen lediglich die Abbildung der Fuzzy-Wissensbasis. Die Aufgabe eines Fuzzy-Reglers, der die Kundeneigenschaften in passende Anbietermerkmale mitsamt der Angabe von Sicherheitsgraden umwandelt, ist in der Folge von geeigneten Reasonern zu gewährleisten.

#### 4.4.5 Reasoning mit Fuzzy-Ontologien

*Der vorliegende Abschnitt baut auf der Methode zur Definition einer Fuzzy-Wissensbasis auf, die im vorangegangenen Abschnitt erläutert wurde. Auf Grundlage dieser Wissensbasis sollen Kundeneigenschaften aufgrund einer Kundeneingabe in passende Anbietermerkmale umgewandelt werden. Dieser Prozess ist Aufgabe eines Fuzzy-Reglers und wird von geeigneten Werkzeugen – Reasoner genannt – unterstützt.*

Inferenzmaschinen zur Schlussfolgerung impliziten Wissens werden als Reasoner definiert (Müller J. , 2011, S. 15). Im vorliegenden Kontext beinhalten Reasoner die Anwendungslogik zur Bewältigung der Aufgaben eines Fuzzy-Reglers, wie er in Kapitel 4.3.2 vorgestellt wurde.

#### 4.4.5.1 Anforderungen an Fuzzy-Reasoner

Aufgabe eines Reasoners im vorliegenden Anwendungsfall ist es, basierend auf der Fuzzy-Ontologie die zu den eingegebenen Kundeneigenschaften passenden Anbietermerkmale zu identifizieren. Diese Anbietermerkmale werden mit Sicherheitsgraden bewertet, die eine Aussage darüber erlauben, wie wahrscheinlich die Selektion eines Anbietermerkmals basierend auf der Kundeneingabe ist. Der Reasoner muss hierfür in der Lage sein, folgende Informationen aus der OWL 2 Ontologie zu extrahieren und entsprechende Rückmeldungen zu geben:

- Extraktion der linguistischen Variablen des Fuzzy-Systems, die innerhalb der OWL 2 Konzepte gespeichert sind
- Extraktion der linguistischen Terme sowie derer Zugehörigkeitsfunktionen aus den OWL 2 Datentypen
- Extraktion der Fuzzy-Regelbasis aus den Axiomen der Fuzzy-Ontologie
- Extraktion der zu verwendenden Fuzzy-Logik aus der Deklaration der Fuzzy-Ontologie
- Rückmeldung der Anbietermerkmale (linguistische Terme) sowie deren Sicherheitsgrade basierend auf der Kundeneingabe

Entsprechende Reasoningalgorithmen sowie IT-basierte Implementierungen können selbst entwickelt werden, alternativ kann auf bestehende Ansätze zurückgegriffen werden.

#### 4.4.5.2 Existierende Implementierungen von Fuzzy-Reasonern

Im Rahmen der vorliegenden Forschungsarbeit wird darauf verzichtet, einen eigenen Reasoningalgorithmus zur Durchführung des Inferenzschrittes zu entwickeln. Stattdessen wird auf bestehende Implementationen von Reasonern verwiesen, welche die Möglichkeit besitzen, als Fuzzy-Regler zu agieren. Tabelle 8 stellt eine Übersicht bestehender Fuzzy-Reasoner vor. Detaillierte Informationen können den referenzierten Publikationen entnommen werden.

Die Wahl des passenden Reasoners ist abhängig vom Anwendungsfall, da den Reasonern unterschiedliche Beschreibungslogiken und Algorithmen zugrunde liegen. So unterstützt der Reasoner FuzzyDL die Beschreibungslogik *SHIF*, während der Reasoner GERDS lediglich die weniger ausdrucksstarke Beschreibungslogik *ALC* verarbeiten kann. Die Fuzzy-Ontologie darf folglich lediglich diejenige Ausdrucksstärke aufweisen, die der Reasoner im Inferenzschritt auch unterstützt. Im vorliegenden Forschungsvorhaben wurde der Reasoner *FuzzyDL* gewählt.

#### 4.4.5.3 Parsing der Fuzzy-Ontologie

Da jeder Reasoner ein proprietäres Datenformat verwendet, ist die auf OWL 2 basierende Fuzzy-Ontologie zunächst in das Datenformat des jeweiligen Reasoners zu übersetzen. Die Übersetzung erfolgt automatisch unter Zuhilfenahme spezieller Softwareprogramme, Parser genannt. Ein

Name	Publikation	Datum	Fuzzy-Logik
DeLorean	(Bobbilo, Delgado, & Gomez-Romero, 2007)	2007	Zadeh
Fire	(Stoilos, Simou, Stamou, & Kollias, 2006)	2006	Zadeh
FuzzyDL	(Bobbilo & Straccia, 2008)	2008	Zadeh, Łukasiewicz
GERDS	(Habiballa, 2007)	2007	Łukasiewicz
GURDL	(Haarslev, Pai, & Shiri, 2007)	2007	Allgemein
YADLR	(Konstantopoulos & Apostolikas, 2007)	2007	Łukasiewicz

**Tabelle 8: Existierende Implementierungen von Fuzzy-Reasonern**

Parser wird als Programm definiert, das nach bestimmten, vordefinierten Regeln für die Zerlegung und Umwandlung einer ebenso in einem bestimmten, vordefinierten Format vorliegenden Eingabe in ein für die Weiterverarbeitung brauchbares Format zuständig ist (Zerbst, 2002, S. 105). Geeignete Parser sind anwendungsfallabhängig zu entwickeln.

Im Folgenden wird der Zusammenhang zwischen den Objekttypen der Fuzzy-Ontologie und dem Datenformat des Reasoners beispielhaft am FuzzyDL Reasoner gezeigt.

### Linguistische Variablen

Die linguistischen Variablen werden in der Fuzzy-Ontologie als herkömmliche Konzepte definiert. Sie repräsentieren die Kundeneigenschaften und Anbietermerkmale. Dabei wird direkt der zulässige Wertebereich definiert.

*Syntax des FuzzyDL Reasoners:*

*(define-concrete-feature <Concept> \*<datatype>\* <a> <b>)*

wobei die Variablen <Concept> für die Bezeichnung des Konzeptes, <datatype> für die Datentypen Integer oder Real und <a>, <b> mit  $a < b$  für die Eingrenzung des Wertebereiches stehen.

### Linguistische Terme

Linguistische Terme entsprechen den Zugehörigkeitsfunktionen, die in der Fuzzy-Ontologie als Datentypen abgebildet werden.

*Syntax des FuzzyDL Reasoners:*

*(define-fuzzy-concept <Datatype> triangular(<a>,<b>, <c>, <d>, <e>)) |*

*(define-fuzzy-concept <Datatype> left-shoulder(<a>,<b>, <c>, <d>)) |*

*(define-fuzzy-concept <Datatype> right-shoulder(<a>,<b>, <c>, <d>))*

wobei <Datatype> für den Namen des Datentypen steht und die Variablen <a>, <b>, <c>, <d>, <e> die Zugehörigkeitsfunktion eindeutig beschreiben.

### Fuzzy-Regelbasis

Die Fuzzy-Regelbasis wird innerhalb der Fuzzy-Ontologie durch Axiome abgebildet und enthält das Wissen, über die Verknüpfung von Kundeneigenschaften und Anbietermerkmalen. Im Beispiel wird lediglich eine Regel vorgestellt, die in der Prämisse zwei Kundeneigenschaften und in der Konklusion ein Anbietermerkmal beinhaltet.

*Syntax des FuzzyDL Reasoners:*

```
(define-concept <Rule> (l-implies (g-and (some <Concept1> <Datatype1>) (some <Concept2> <Datatype2>))...
(some <Concept3> <Datatype3>)))
```

wobei *<Concept1>* und *<Concept2>* zwei Kundeneigenschaften, sowie *<Datatype1>* und *<Datatype2>* für die adäquaten Zugehörigkeitsfunktionen stehen. *<Concept3>* und *<Datatype3>* sind analog für die Anbietermerkmale definiert. Der Operator *g-and* steht für den Und-Operator in der Prämisse, während *l-implies* den Implikationsoperator darstellt.

### Fuzzy-Logik

Die innerhalb des Reasoningschrittes zu verwendende Fuzzy-Logik wird aus der Deklaration der Fuzzy-Ontologie übernommen.

*Syntax des FuzzyDL Reasoners:*

```
(define-fuzzy-logic [<fuzzy-logic>])
```

wobei *<fuzzy-logic>* entweder den Wert ‚lukasiewicz‘ oder ‚zadeh‘ für die entsprechenden Fuzzy-Logiken annehmen kann.

#### 4.4.6 Zusammenfassung

Das vorliegende Unterkapitel 4.4 ‚DfC Fuzzy-Ontologie‘ stellt einen zum DfC Fuzzy-Controller alternativen Ansatz zur Konfiguration bereit, der zusätzlich zur Verarbeitung von Unschärfe in natürlicher Sprache auch implizites Wissen erkennen und daraus auf die expliziten Wünsche und Bedürfnisse des Kunden schlussfolgern kann. Wie auch der zuvor vorgestellte DfC Fuzzy-Controller erlaubt die DfC Fuzzy-Ontologie die auf Eigenschaften basierende Sprache des Kunden mit der merkmalsbasierten Sprache des Anbieters zu verknüpfen. Hierfür wurde die Wissensrepräsentationssprache OWL 2 erweitert, um Fuzzy-Informationen abbilden zu können. Die Konfiguration findet unter Zuhilfenahme bestehender Software-Werkzeuge, Reasoner genannt, statt.

Zur strukturierten Entwicklung eines Lösungskonfigurators basierend auf den beiden vorgestellten Ansätzen werden geeignete Vorgehensmodelle benötigt. Eine solche strukturierte Vorgehensweise stellt das folgende Unterkapitel 4.5 ‚DfC Unified Process‘ vor, das den ersten methodischen Schritt der Methodik ‚Design for Customer‘ bildet.

### 4.5 DfC Unified Process

*Im vorliegenden Unterkapitel wird ein Vorgehensmodell bereitgestellt, das die strukturierte Entwicklung, Wartung und Pflege des DfC Konfigurators ermöglicht. Das Vorgehensmodell wurde als Best-Practice-Methode aus bestehenden Ansätzen des Ontology Engineering abgeleitet. Die grundlegende Struktur basiert auf dem Unified Process aus der Softwareentwicklung, wurde jedoch um anwendungsspezifische Charakteristika sowie Aspekte des Projektmanagements erweitert. Die Einsetzbarkeit des Vorgehensmodells ist für beide in den Kapiteln 4.3 und 4.4 vorgestellten alternativen Ansätze gewährleistet. Wie auch der Unified Process ist der vorliegende Ansatz anwendungsgetrieben mit sowohl iterativen als auch inkrementellen Bestandteilen. Fünf Workflows mit festgelegten Ein- und Ausgabeparametern geben feste Handlungsanweisungen sowie benötigte Arbeitsergebnisse vor.*

Das im vorliegenden Kapitel vorgestellte Vorgehensmodell DfC Unified Process basiert auf den Ergebnissen der Vergleichsanalyse bestehender Methoden des Ontology Engineering auf Basis des IEEE 1074-1995 Standards für Lebenszyklen der Softwareentwicklung. Techniken bestehender Methoden, insbesondere der Vorgehensmodelle UPON und On-To-Knowledge, fließen in das entstandene Vorgehensmodell ein, das die Vorteile bestehender Ansätze in sich vereint. Im Folgenden wird zunächst die generelle Architektur des Vorgehensmodells vorgestellt und im Anschluss detailliert auf die einzelnen Workflows, welche die Kernbestandteile des Vorgehensmodells bilden, eingegangen. Diese sind unter Zuhilfenahme der grafischen Notationssprache BPMN (Business Process Model and Notation) in Version 2.0 modelliert<sup>4</sup>.

#### 4.5.1 Überblick: Der DfC Unified Process

Basierend auf der Evaluation des Standes der Technik zum Themengebiet Ontology Engineering wurde ein Vorgehensmodell abgeleitet, das auf dem aus der Softwareentwicklung stammenden Unified Process sowie dessen an Ontologien angepasste Adaption, dem Vorgehensmodell UPON, beruht. Das hier beschriebene Vorgehensmodell wurde dabei stark an die speziellen Anforderungen des DfC Konfigurators angepasst. Zusätzlich wird das Fehlen von begleitenden Projektmanagementaktivitäten durch Konzepte ausgeglichen, die aus der Projektmanagementlastigen Methodik On-To-Knowledge übernommen wurden.

Wie auch der Unified Process aus dem Software Engineering ist der vorliegende Ansatz anwendungsgetrieben mit sowohl iterativen als auch inkrementellen Bestandteilen. Die Anwendungsgetriebenheit bezieht sich darauf, dass keine generischen Domänen-Ontologien entwickelt werden, sondern auf ein klares Anwendungsgebiet ausgerichtete Ontologien.

---

<sup>4</sup> Die offizielle Spezifikation der BPMN Notation ist unter der Webadresse <http://www.omg.org/spec/BPMN/> ersichtlich.

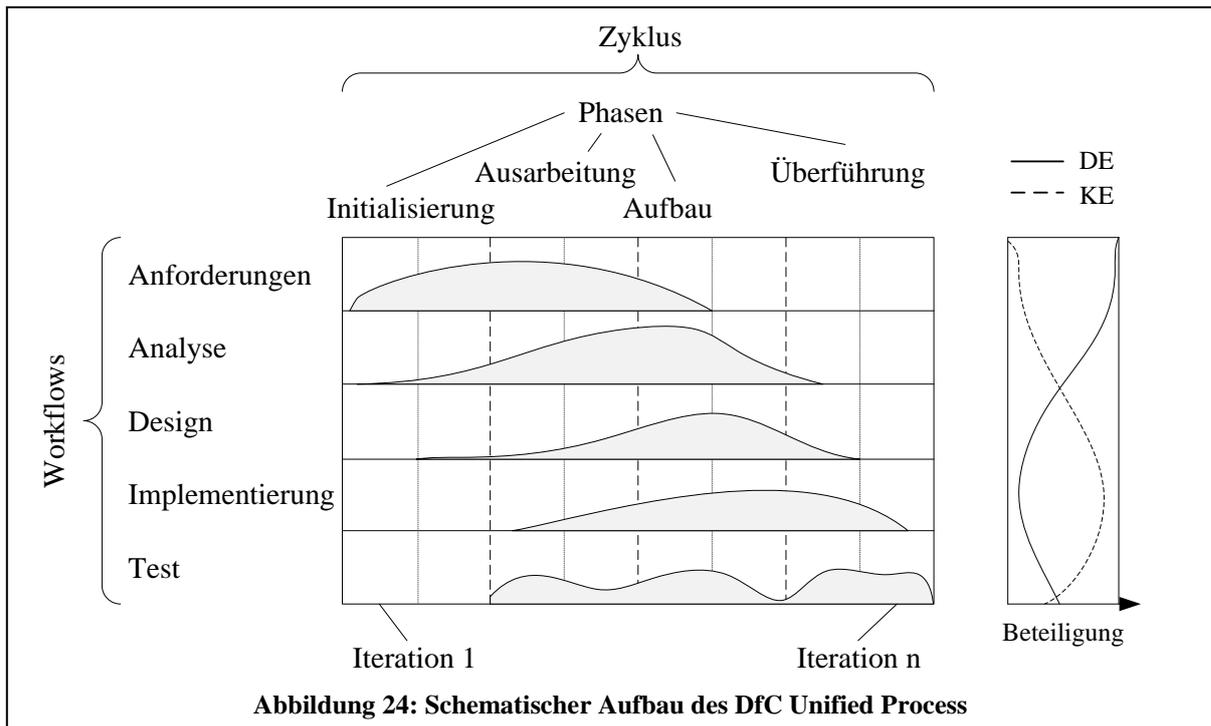


Abbildung 24: Schematischer Aufbau des DfC Unified Process

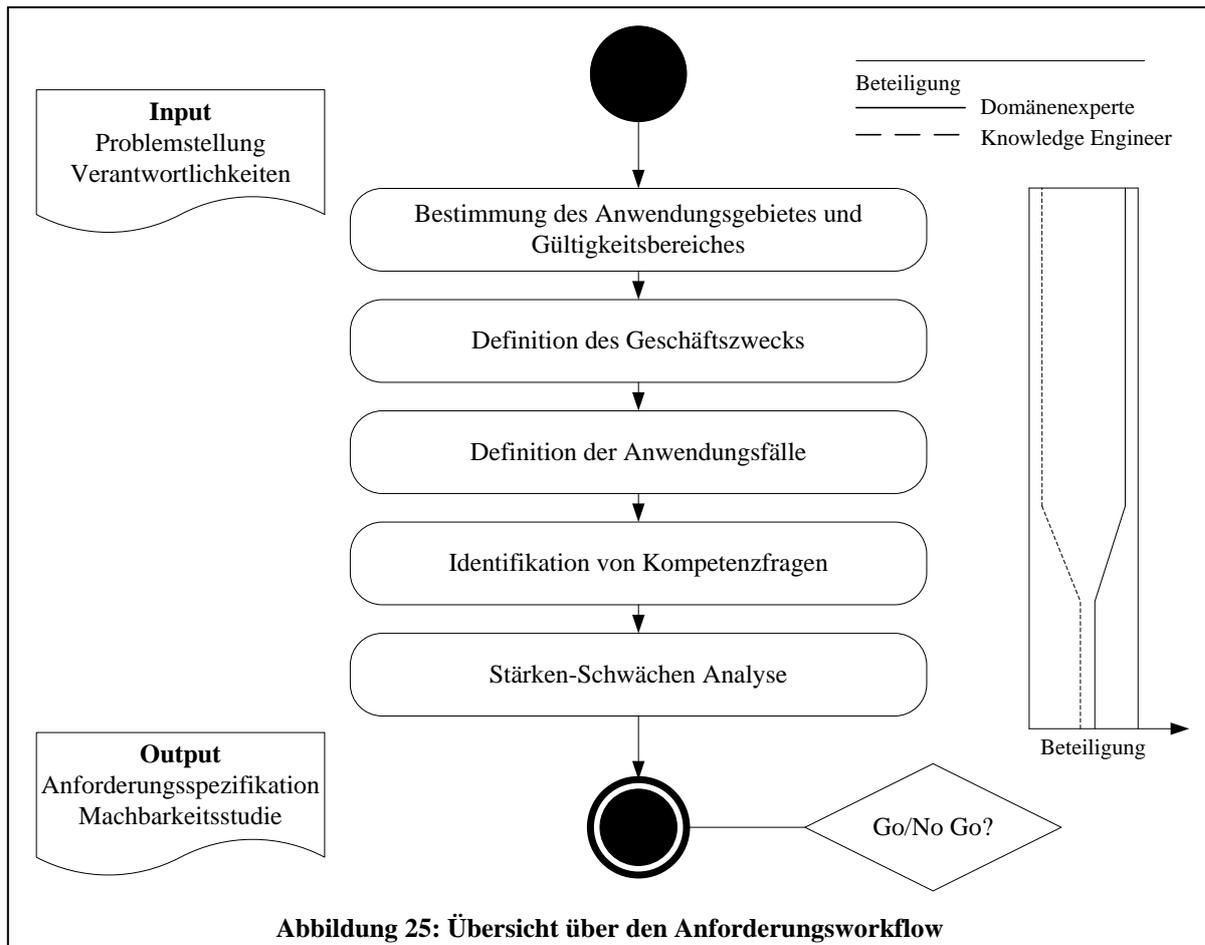
Das Vorgehensmodell folgt dem Aufbau des Unified Process und teilt sich wie auch dieser in Zyklen, Phasen, Iterationen und Workflows ein. Jeder Zyklus besteht aus den vier Phasen Initialisierung, Ausarbeitung, Aufbau und Überführung. Am Ende jedes Zyklus entsteht eine neue Version der Ontologie. Die Phasen eines Zyklus werden iterativ durchlaufen, wobei jede Iteration wiederum aus fünf Workflows besteht, die sukzessive abgearbeitet werden. Diese sind der Anforderungsworkflow, der Analyseworkflow, der Designworkflow, der Implementierungsworkflow und der Testworkflow. Workflows und Phasen können als orthogonal zueinander stehend begriffen werden, da der Beitrag eines Workflows zur Iteration einer Phase mehr oder weniger relevant sein kann. Frühe Phasen (Initialisierungsphase) befassen sich vorrangig mit der Anforderungsgenerierung und fokussieren demnach insbesondere den Anforderungsworkflow. Innerhalb der späten Phasen wird hingegen die Umsetzung der Anforderungen forciert, weshalb der Implementierungs- und Testworkflow im Mittelpunkt des Interesses stehen. Abbildung 24 zeigt den schematischen Aufbau des DfC Unified Process.

Im Fokus des vorliegenden Vorgehensmodells stehen die fünf oben genannten Workflows, die operative Arbeitsanweisungen für die an der Erstellung des DfC Konfigurators beteiligten Personen beinhalten. Voraussetzung, um einen Workflow starten zu können, ist ein spezifischer, vordefinierter Input. Am Ende eines jeden Workflows wiederum wird ein vordefinierter Output erwartet, welcher innerhalb eines Meilensteines bewertet wird. Werden die Anforderungen des Meilensteines vollständig erfüllt, kann mit dem nächsten Workflow gestartet werden. Die Arbeitsschritte des Workflows sind andernfalls so lange zu wiederholen, bis der Meilenstein passiert werden kann oder das Projekt abgebrochen wird.

### 4.5.2 Anforderungsworkflow

Der Anforderungsworkflow legt mit der Definition der Szenarien und Geschäftsziele den Grundstein des Ansatzes. Innerhalb des Workflows werden die Ziele der Fuzzy-Ontologie aus dem Blickwinkel der Nutzer aufgestellt. Dafür ist es notwendig, (1) das Anwendungsgebiet und den Gültigkeitsbereich zu bestimmen, (2) den Geschäftszweck zu definieren, (3) die Anwendungsszenarien aufzuzeigen, (4) Kompetenzfragen zu definieren, welche die Ontologie beantworten können muss und schlussendlich (5) in eine Stärken-Schwächen Analyse münden. Eingangsgrößen des Workflows stellen die Problemstellung sowie die Verantwortlichkeiten dar. Der Output umfasst die Anforderungsspezifikation sowie eine Machbarkeitsstudie in Form einer Stärken-Schwächen Analyse.

1. *Bestimmung des Anwendungsgebietes und Gültigkeitsbereiches:* Die Umschreibung des Anwendungsgebietes ist ein grundlegender Schritt, um den Gültigkeitsbereich einer Ontologie festzulegen (Uschold & King, 1995). Das Anwendungsgebiet beschreibt dabei den Rahmen, innerhalb dessen der DfC Konfigurator eingesetzt werden soll, wobei die Beschreibung bereits die wichtigsten Konzepte der zukünftigen Ontologie beinhalten kann. Innerhalb der Darstellung des Anwendungsgebietes kann auch der explizite Ausschluss gewisser Anwendungsszenarien vorgenommen werden.
2. *Definition des Geschäftszwecks:* Der anvisierte Nutzen des DfC Konfigurators für die Organisation ist bereits früh herauszustellen. Nach De Nicola et al. (De Nicola, Missikoff, & Navigli, 2009, S. 261) lassen sich drei prinzipielle Geschäftsnutzen identifizieren: Die Informationsbereitstellung, die Mediation zwischen Prozessen und schließlich der Informationsabgleich zwischen Nachrichten, die zwischen einem Sender und Empfänger ausgetauscht werden. Im vorliegenden DfC Ansatz steht der dritte identifizierte Geschäftsnutzen im Vordergrund, der im Rahmen des vorliegenden Prozessschrittes weiter zu spezifizieren ist.
3. *Identifikation der Anwendungsfälle:* Die Anwendungsfälle verfeinern die Definition des Anwendungsgebietes und beschreiben in groben Umfang die notwendigen Prozessschritte, welche die Ontologie abzudecken hat. Zur Identifikation der Anwendungsfälle können Kreativitätsmethodiken wie in (Pohl & Rupp, 2009) beschrieben zum Einsatz kommen. Die Anwendungsfälle werden im Analyseworkflow zu semi-formalen Anwendungsszenarien erweitert.
4. *Identifikation von Kompetenzfragen:* Kompetenzfragen werden von Grüninger und Fox (Grüninger & Fox, 1995) erstmals eingeführt. Sie stellen konzeptuelle Fragen dar, welche die Ontologie nach ihrer Fertigstellung beantworten können muss. Die Fragen werden

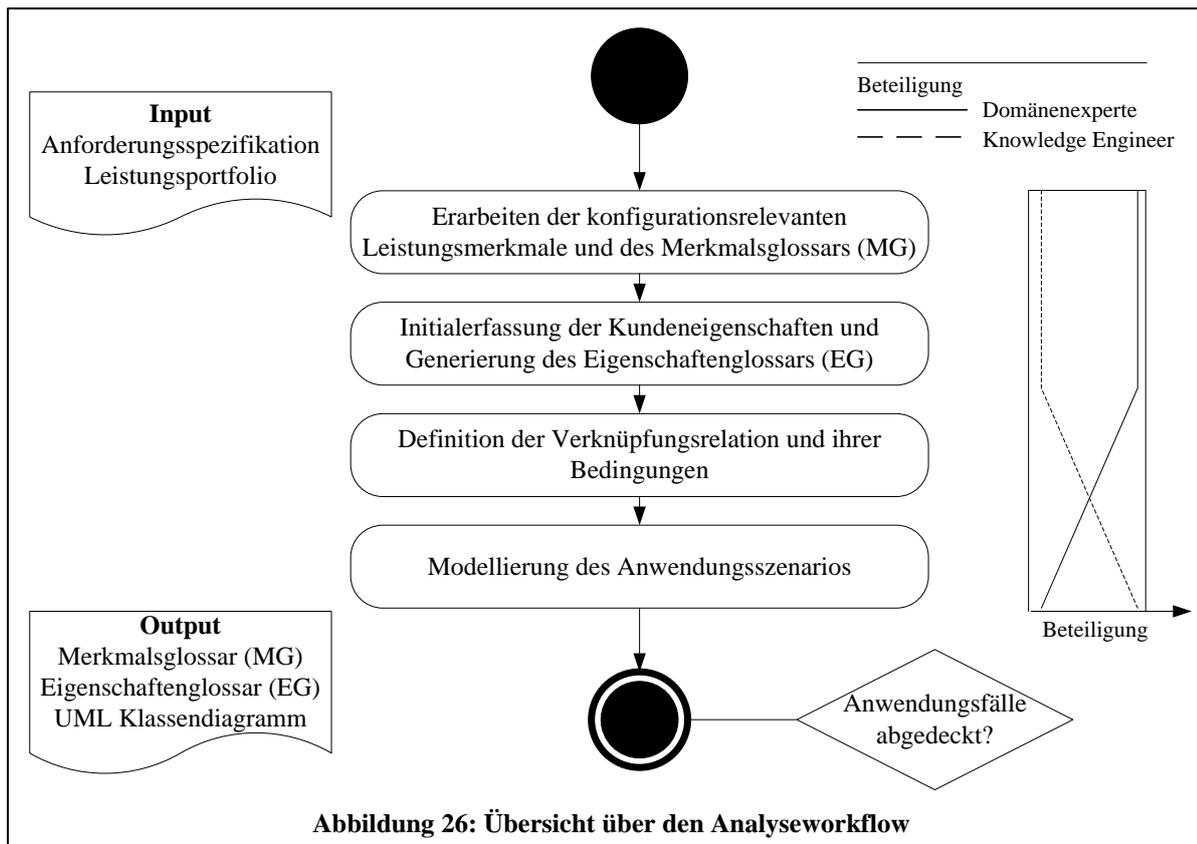


während Brainstorming Sessions mit den zukünftigen Nutzern identifiziert und dienen während des Testworkflows der Evaluation der Ontologie.

5. *Stärken-Schwächen Analyse:* Die Stärken-Schwächen Analyse (SWOT Analyse) dient als letzter Prozessschritt innerhalb des Anforderungsworkflows einer Bewertung der bislang identifizierten Punkte. Basierend auf der SWOT Analyse wird der Nutzen des Einsatzes den verbundenen Kosten und Aufwänden gegenübergestellt. Die SWOT Analyse dient daher als Input des ersten Meilensteins, der eine Go oder No-Go Entscheidung beinhaltet. Das Projekt wird je nach Entscheidung entweder fortgesetzt oder gestoppt.

### 4.5.3 Analyseworkflow

Der Analyseworkflow schließt sich dem Anforderungsworkflow nach erfolgreich passierter Go-Entscheidung an. Zentrale Punkte des Analyseworkflows sind die Analyse der Produktmerkmale sowie die Identifikation der relevanten Kundeneigenschaften. Der Workflow teilt sich folglich (1) in die Generierung des Merkmalsglossars (MG), gefolgt von der (2) Initialerfassung der Kundeneigenschaften und Generierung des Eigenschaftenglossars (EG), (3) die Definition der



geeigneten Verknüpfungsrelation zwischen Eigenschaften und Merkmalen und schließlich (4) die Modellierung des Anwendungsszenarios. Eingangsvariablen bilden die im Anforderungsworkflow erstellte Anforderungsspezifikation sowie eine Übersicht des im Unternehmen bestehenden Leistungsportfolios. Ein Merkmalsglossar, ein Eigenschaftenglossar sowie UML Klassendiagramme der Anwendungsszenarios bilden den Output des Workflows, der im zweiten Meilenstein dahingehend überprüft wird, ob sämtliche Anwendungsfälle abgedeckt sind.

1. *Erarbeiten der konfigurationsrelevanten Leistungsmerkmale und Generierung des Merkmalsglossars (MG):* In einem ersten Schritt wird die relevante Terminologie des Leistungsspektrums erfasst. Dies kann manuell oder teilweise werkzeugunterstützt beispielsweise durch Text Mining Tools wie Text2Onto<sup>5</sup> geschehen. Die so identifizierten Konzepte werden in einem Merkmalslexikon (ML) hinterlegt und in einem zweiten Schritt zu einem Merkmalsglossar erweitert. Während das Merkmalslexikon lediglich eine Auflistung der relevanten Konzepte darstellt, werden die Konzepte im Merkmalsglossar zusätzlich durch informelle Definitionen ergänzt.
2. *Initialerfassung der Kundeneigenschaften und Generierung des Eigenschaftenglossars (EG):* Analog zu den Leistungsmerkmalen sind die Eigenschaften aus Kundensicht zu

<sup>5</sup> <http://semanticweb.org/wiki/Text2Onto>. Text2Onto ist ein Softwarewerkzeug zur automatisierten Ontologierstellung. Es extrahiert Konzepte, Instanzen und Relationen aus Text-, XML-, HTML- und PDF-Dateien.

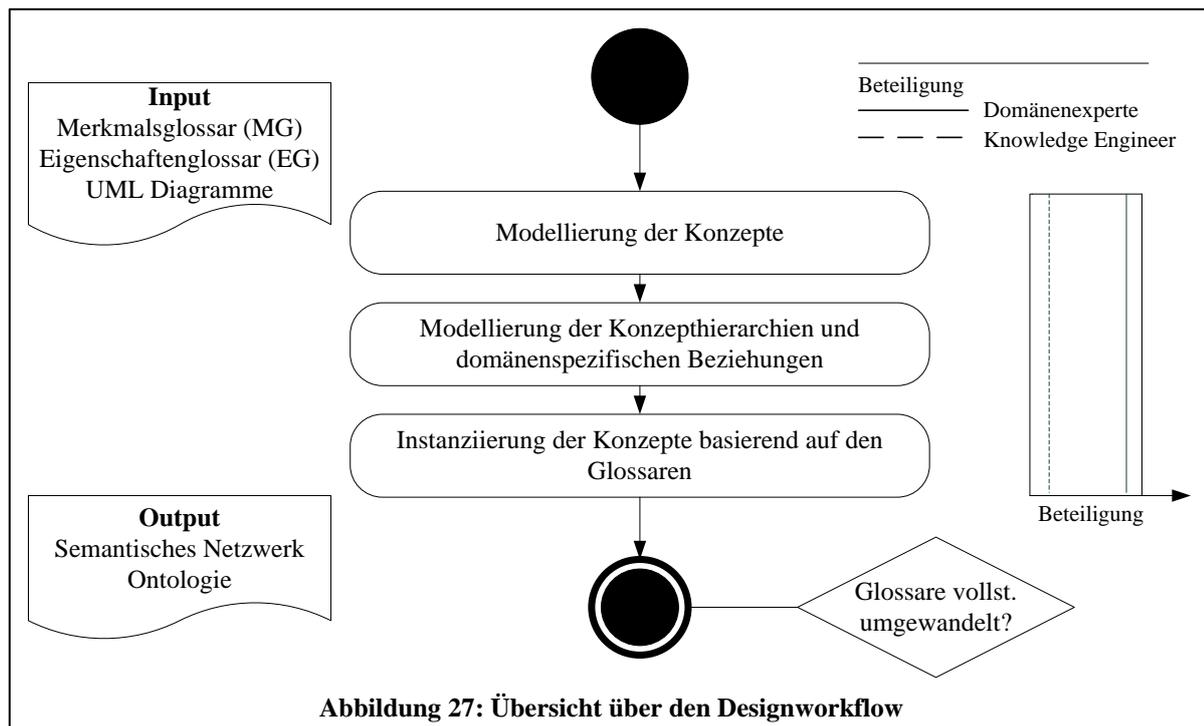
evaluieren. Dies hat unter Einbezug einer repräsentativen Kundengruppe zu erfolgen. Wie auch im vorangegangenen Prozessschritt wird zunächst ein Lexikon erstellt, das die relevanten Kundeneigenschaften auflistet. Dieses Eigenschaftenslexikon (EL) wird wiederum um informelle Definitionen zu einem Eigenschaftenglossar (EG) ergänzt, das dazu dient, ein einheitliches Begriffsverständnis herzustellen.

3. *Definition der Verknüpfungsrelation und ihrer Bedingungen:* Im Rahmen der vorliegenden Arbeit wird als Verknüpfungsrelation zwischen Eigenschaften und Merkmalen eine Fuzzy-Relation vorgeschlagen. Wie in Kapitel 3.1 ‚Ausgewählte Methoden der Künstlichen Intelligenz‘ gezeigt, kann die Relation nach Unternehmenseinschätzung unterschiedlich definiert werden.
4. *Modellierung des Anwendungsszenarios:* Ein Anwendungsszenario besteht in der Regel aus mehreren Use Cases, welche das Anwendungsszenario weiter spezifizieren. Als Modellierungskonzepte der Anwendungsszenarien kommen UML Use Case Diagramme und UML Use Case Spezifikationen zum Einsatz (Rupp, 2004, S. 161). Sind sämtliche Anwendungsszenarien in den Glossaren und den UML Diagrammen abgedeckt, so kann mit Meilenstein 2 der Analyseworkflow abgeschlossen werden.

#### 4.5.4 Designworkflow

Innerhalb des Designworkflows werden die in den Glossaren festgehaltenen Eigenschaften und Merkmale in eine ontologische Struktur überführt. Die Basis der Modellierung bilden die UML Diagramme aus dem Analyseworkflow, zusammen mit den zu modellierenden Konzepten aus den beiden Glossaren. Der Prozess unterteilt sich dabei in die Schritte (1) Modellierung der Konzepte, Datentypen und Konzeptionshierarchien (2) Modellierung der Regelbasis und (3) Instanziierung der Konzepte basierend auf den Glossaren. Der Output des Workflows stellt folglich ein semantisches Netzwerk in Form einer Ontologie dar. Treiber des Designworkflows sind Knowledge Engineers, Domänenexperten besitzen lediglich noch eine unterstützende Funktion.

1. *Modellierung der Konzepte, Datentypen und Konzeptionshierarchien:* Basierend auf dem Eigenschaftens- (EG) und Merkmalsglossar (MG) werden die dort beschriebenen Informationen als Konzepte und Datentypen modelliert. Die Modellierung sollte dabei auf den in Kapitel 4.4.3 ‚Fuzzy-Erweiterungen von OWL 2‘ vorgestellten Objekten beruhen, um diese im nächsten Schritt in eine geeignete formale Sprache überführen zu können. Zur Modellierung der Konzeptionshierarchien existieren in der Literatur drei Ansätze (Ushold & Grüninger, 1996): Der Top-Down Ansatz, der Konzepte vom Allgemeinen zum Speziellen modelliert, der Bottom-Up Ansatz, der umgekehrt dazu verfährt und schließlich der

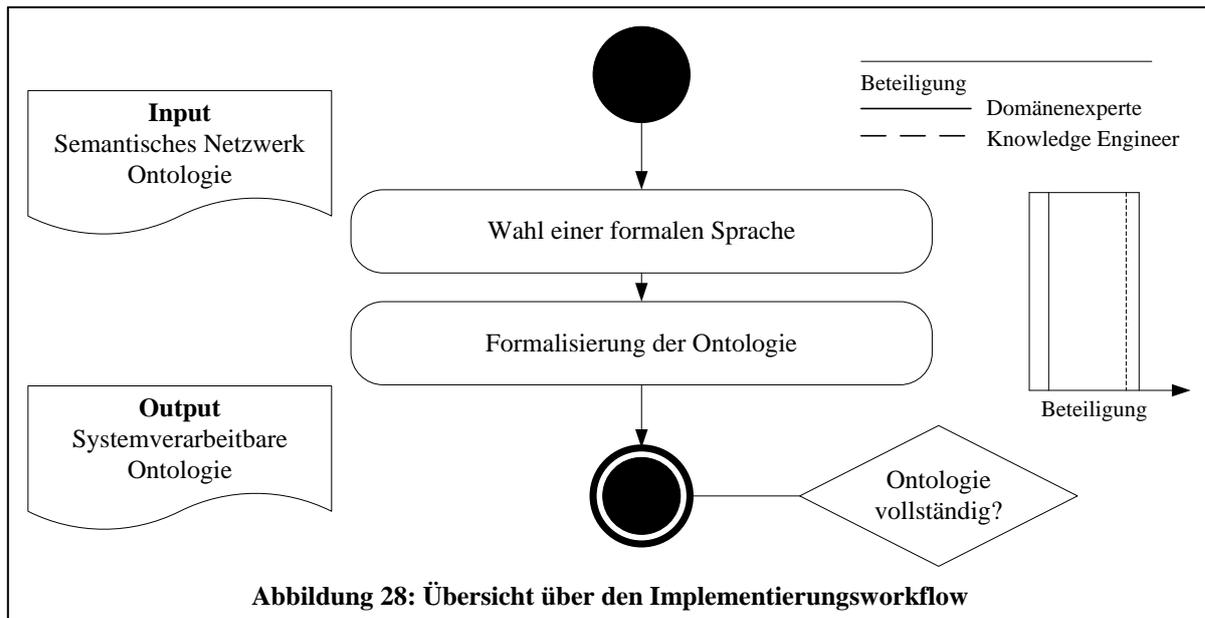


Middle-Out Ansatz. Dabei werden die auffälligsten Konzepte zuerst modelliert, die in der Regel den mittleren Konzeptionshierarchien zugehörig sind. Dieser Ansatz wird in der Literatur als erfolgversprechendster Ansatz hervorgehoben (Gomez-Perez, Fernandez-Lopez, & Corcho-Garcia, 2003, S. 118) und auch hier vorgeschlagen.

2. *Modellierung der Regelbasis:* Die Regelbasis ist entsprechend der im vorangegangenen Workflow definierten Verknüpfungsrelation zu modellieren. Das in Kapitel 4.4.4 Konzept zur Erweiterung von OWL 2 unterstützt Fuzzy-Logik als Verknüpfungsbasis. Weitere Verknüpfungsrelationen können analog zu der dort vordefinierten Syntax spezifiziert werden.
3. *Instanziierung der Konzepte basierend auf den Glossaren:* Die auf Basis des Merkmalsglossars definierten Konzepte geben Aufschluss über generelle Merkmale des Leistungsportfolios wie beispielsweise Abmessungen. Die Instanzen der Konzepte stehen hingegen für die tatsächlichen Leistungen, welche im Leistungsportfolio enthalten sind. Diese sind bei den zugehörigen Konzepten als Instanzen zu modellieren.

#### 4.5.5 Implementierungsworkflow

Ziel des Implementierungsworkflows ist es, die Ontologie in eine formale Sprache zu überführen und im System umzusetzen. Hierfür ist zunächst (1) eine geeignete Sprache zu wählen. Im Anschluss ist das im Designworkflow entstandene Semantische Netz als (2) systemverarbeitbare Ontologie umzusetzen und zu formalisieren. Als Eingangsdokument dient das im

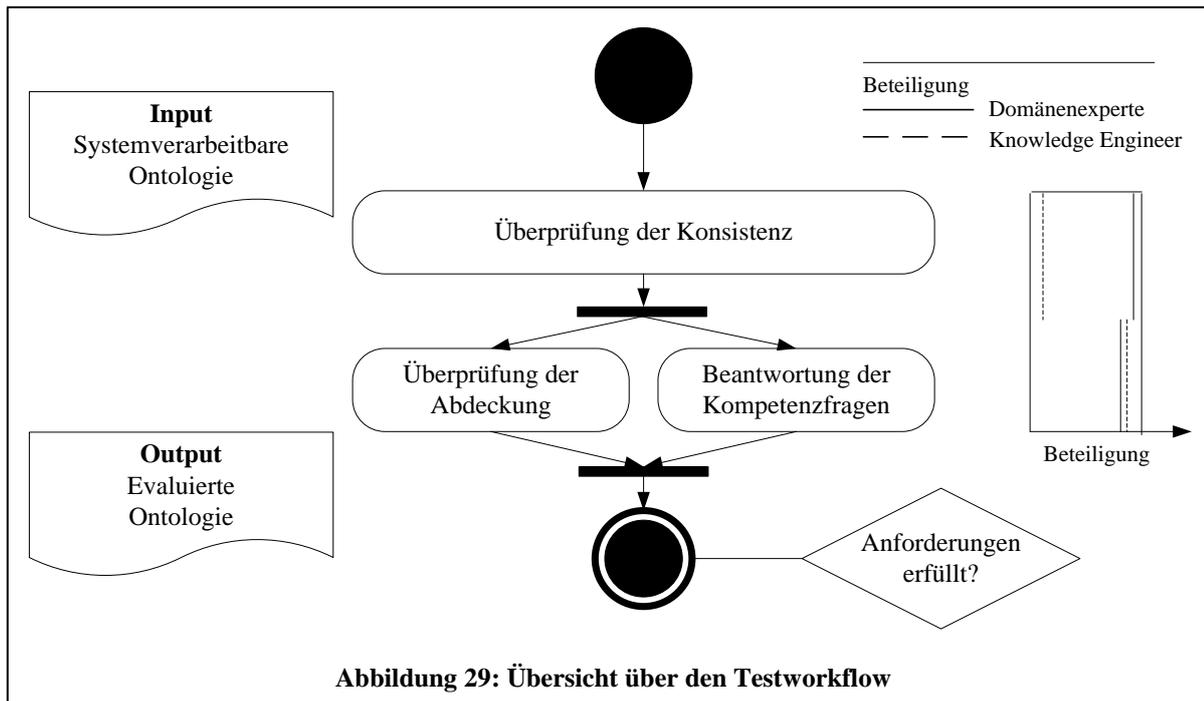


vorangegangenen Workflow entstandene Semantische Netz, Output des aktuellen Workflows bildet eine systemverarbeitbare Ontologie. Wie auch der Designworkflow wird der vorliegende Prozess stark von Knowledge Engineers getrieben.

1. *Wahl einer formalen Sprache:* Im Rahmen dieser Arbeit wurden zwei alternative Ansätze zur Abbildung des DfC Konfigurators vorgestellt. Die mathematische Beschreibung in Kapitel 4.3 ‚DfC Fuzzy-Controller‘ lässt die Wahl einer formalen Sprache offen. In der zweiten Alternative, die in Kapitel 4.4 ‚DfC Fuzzy-Ontologie‘ vorgestellt wurde, wurde OWL 2 als erfolgversprechende Sprache identifiziert und darauf basierend eine Syntax zur Abbildung des Anwendungsszenarios vorgestellt. In diesem Fall ist zur Modellierung die in Kapitel 4.4.4 ‚Repräsentation von Fuzzy-Ontologien in OWL 2‘ vorgestellte Syntax zu verwenden.
2. *Formalisierung der Ontologie:* Im Rahmen des Designworkflows wurden die Glossare mitsamt der Regelbasis den entsprechenden ontologischen Objekten zugewiesen und in einem semantischen Netz hinterlegt. Dieses Semantische Netz ist nun in die formale Sprache zu überführen, die im vorangegangenen Prozessschritt ausgewählt wurde. Wurden sämtliche Objekte des Semantischen Netzes in die formale Ontologie überführt, so kann der Meilenstein passiert und die Ontologie getestet werden.

#### 4.5.6 Testworkflow

Der Testworkflow dient der Qualitätssicherung und ermöglicht es, das Lösungsdesign auf die im Anforderungsworkflow getroffenen Anforderungen hin zu überprüfen. Nach Burton-Jones et al.



(Burton-Jones, Storey, Sugumaran, & Ahluwalia, 2005, S. 98) existieren vier Kategorien, nach denen Ontologien beurteilt werden können:

- Syntaktische Qualität, die die Ontologie nach der Qualität ihrer formalen Struktur bewertet
- Semantische Qualität, bei der nach widersprüchlichen Konzepten gesucht wird
- Pragmatische Qualität, welche die Qualität einer Ontologie nach der Unterstützung ihrer Benutzer bewertet
- Soziale Qualität, die erst nach der Veröffentlichung bewertet werden kann und allgemeinere Kriterien der Ontologienutzung fokussiert.

Die syntaktische und semantische Qualität wird im Prozessschritt (1) Überprüfung der Konsistenz bewertet, während die pragmatische Qualität durch die Schritte (2) Überprüfung der Abdeckung und (3) Beantwortung der Kompetenzfragen validiert wird. Die systemverarbeitbare Ontologie wird im vorliegenden Workflow bewertet. Der Output ist eine evaluierte Ontologieversion, die zum Einsatz freigegeben ist.

1. *Überprüfung der Konsistenz:* Mit der syntaktischen und semantischen Konsistenz werden Kriterien zur formalen Struktur der Ontologie überprüft. Neben einer manuellen Prüfung durch Knowledge Engineers können mit Reasonern auch automatisierte Werkzeuge zur Qualitätssicherung zum Einsatz kommen. Eine Übersicht geeigneter Reasoner wird in Kapitel 4.4.5.2 „Existierende Implementierungen von Fuzzy-Reasonern“ geliefert.
2. *Überprüfung der Relevanz:* Im zweiten Prozessschritt wird überprüft, ob die im Anforderungsworkflow definierten Forderungen inhaltlich korrekt beantwortet werden.

Insbesondere helfen die dort aufgestellten Kompetenzfragen, die Richtigkeit der Ontologie aus Sicht der Domänenexperten zu verifizieren. Der aktuelle Prozessschritt besitzt einen engen Zusammenhang zum folgenden Prozessschritt und wird parallel zu diesem durchgeführt.

3. Überprüfung der Vollständigkeit: Hierbei wird nicht die Korrektheit der Aussagen überprüft, sondern festgestellt, ob sämtliche Anforderungen innerhalb der Ontologie aufgegriffen werden. Der Prozessschritt gibt folglich Auskunft über den Abdeckungsgrad der Anforderungen innerhalb der Ontologie. Werden sämtliche Tests zur Zufriedenheit abgeschlossen, so terminiert der Workflow und eine neue Version der Ontologie entsteht. In der Folge können die Workflows entsprechend der Architektur des Gesamt-Prozessmodells neu gestartet werden.

#### 4.5.7 Stab

Die Verantwortung für den Projektverlauf der Ontologieerstellung übernimmt ein hierfür zu installierender Stab. Die Aufgabe des Stabs ist die Überwachung des Projektverlaufes und die zentrale Sammlung der Teilergebnisse des Projektes. Insbesondere ist der Stab zuständig für:

- *Erstellung eines Projektplanes und Überwachung des Projektverlaufes:* Die operativ am Projekt beteiligten Personen berichten an den Stab, der den Projektplan zentral überwacht. Insbesondere ist der Stab bei nachträglichen Änderungen der Anforderungen für die Neuinitiierung des Projektes verantwortlich. Der Stab tritt daher auch nach erfolgreicher Erstellung der Ontologie zusammen, um geänderte Anforderungen in die Ontologie zu inkorporieren. Hierfür sind unter Umständen Neustarts des Projektes notwendig.
- *Zuweisung von Verantwortlichkeiten:* Der Stab weist den Projektmitgliedern Verantwortlichkeiten und Rollen innerhalb der Workflows zu. Die Projektmitglieder berichten an den Stab.
- *Zentrale Speicherung der Projektergebnisse:* Innerhalb der einzelnen Workflows erstellte Projektergebnisse werden zentral beim Stab gespeichert, um einen gemeinsamen Zugriff auf die jeweiligen aktuellen Versionen der Ergebnisse zu gewährleisten. Somit wird die Nachhaltigkeit der durchgeführten Tätigkeiten sichergestellt.

Im Anschluss an jedes abgeschlossene Projekt übernimmt der Stab die Verantwortung über die Kommunikation an die relevanten Benutzergruppen sowie den produktiven Einsatz der Ontologie. Darunter fällt die Inkorporation der Ontologie in die relevanten Systeme sowie die Erstellung und Durchführung der Benutzerschulungen.

### 4.5.8 Zusammenfassung

Im Rahmen des vorliegenden Kapitels wurde das Vorgehensmodell DfC Unified Process entwickelt, das eine strukturierte Entwicklung und Bereitstellung der Wissensbasis des DfC Konfigurators ermöglicht. Die Einsetzbarkeit des Vorgehensmodells ist für beide in den Kapiteln 4.3 und 4.4 vorgestellten alternativen Ansätze gewährleistet. Das Vorgehensmodell basiert auf bestehenden Modellen des Ontology Engineering, die anwendungsspezifisch angepasst und erweitert wurden. Das Vorgehensmodell kann damit als Best-Practice-Ansatz verstanden werden. Die inhaltliche Grundlage bildet das auf dem Unified Process beruhende Vorgehensmodell UPON, das um Methoden des Projektmanagements erweitert wurde. Die eingearbeiteten Methoden beruhen teilweise auf dem projektmanagement-lastigen Vorgehensmodell On-To-Knowledge. Beide Vorgehensmodelle finden praktischen Einsatz in der Industrie und verwenden bestehende und allgemein anerkannte Werkzeuge und Methoden (UML, Microsoft Visio, ...), um Eintrittsbarrieren der Modellierung zu überwinden.

Mit den bislang vorgestellten Methoden kann ein bedürfnisorientierter Lösungskonfigurator strukturiert entwickelt und implementiert werden. Für die Implementierung stehen mit dem DfC Fuzzy-Controller und der DfC Fuzzy-Ontologie zwei alternative Ansätze zur Verfügung. Im Rahmen der vorgestellten Methodik soll jedoch nicht nur der Konfigurationsansatz selbst, sondern das Gesamtsystem an der Gebrauchstauglichkeit sowie der Benutzerfreundlichkeit ausgerichtet sein. Mit der Sicherstellung dieser Nutzerergonomie befasst sich das folgende Unterkapitel 4.6.

### 4.6 DfC Usability Engineering

*Die vorangegangenen methodischen Schritte der Methodik ‚Design for Customer‘ liefern die methodische und informationstechnische Basis, um einen Lösungskonfigurator basierend auf der Trennung von Eigenschaften und Merkmalen bereitzustellen. Nicht nur die methodische Basis, sondern das Gesamtsystem bestehend aus Mensch, Werkzeug und Aufgabe hat jedoch den Bedürfnissen des Benutzers entsprechend gestaltet zu sein. Die Wahl der richtigen und relevanten Methoden, Normen und Gesetze hinsichtlich der Nutzerergonomie gestaltet sich aufgrund deren Vielfalt und Intransparenz äußerst schwierig. Die vorliegende Arbeit trägt diesem Umstand Rechnung und stellt Richtlinien (Guidelines) und Methodenbaukästen bereit, welche Entscheidungsträger basierend auf einem Fragekatalog eindeutige Handlungsoptionen aufzeigen.*

Das Unterkapitel gibt zunächst einen Überblick über den Themenkomplex des Usability Engineering, welches sich mit der Nutzerergonomie von Systemen befasst. Dabei werden das

Usability-Projektmanagement, die Usability-Anforderungserfassung und die Usability-Evaluation als die wichtigsten Prozessschritte des Usability Engineerings vorgestellt. In der Folge werden die als Ergebnis der vorliegenden Arbeit entstandenen Guidelines und Methodenbaukästen sukzessive für jeden Prozessschritt herausgearbeitet. Die Guidelines führen den Nutzer in einem ersten Schritt transparent durch einen projektspezifisch zu beantwortenden Fragekatalog, der eine schnelle und bedarfsgerechte Auswahl relevanter Methoden, Normen und Gesetze liefert. Die Methodenbaukästen eignen sich darauf aufbauend, um weiterführende Informationen zur Gesetz- und Methodenauswahl zu erhalten und diese weiter zu verfeinern.

### 4.6.1 Überblick

Die vorangegangenen Abschnitte liefern die methodische und informationstechnische Basis, um einen Lösungskonfigurator basierend auf der Trennung von Eigenschaften und Merkmalen bereitstellen zu können. Nicht nur die methodische Basis, sondern das Gesamtsystem bestehend aus Mensch, Werkzeug und Aufgabe hat jedoch den Bedürfnissen des Benutzers entsprechend gestaltet zu sein. Mit dieser Aufgabe beschäftigen sich die wissenschaftliche Disziplin der Ergonomie sowie die Usability als qualitative Messgröße der Ergonomie (Burger & Ovtcharova, 2012, S. 28). Nach der internationalen Normung ist Ergonomie definiert als „wissenschaftliche Disziplin, die sich mit dem Verständnis der Wechselwirkungen zwischen menschlichen und anderen Elementen eines Systems befasst (...) mit dem Ziel, das Wohlbefinden des Menschen und die Leistung des Gesamtsystems zu optimieren.“ (DIN EN ISO 6385, 2004). Anders als die Ergonomie ist Usability keine eigenständige wissenschaftliche Disziplin, stattdessen stellt sie das Ziel der Gestaltung eines Gesamtsystems nach den Erkenntnissen der Ergonomie dar. Usability wird folgerichtig als das Ausmaß definiert, in dem ein Benutzer ein System in einem vordefinierten Nutzungskontext verwenden kann, um seine Ziele effektiv, effizient und zufriedenstellend zu erreichen (DIN EN ISO 9241, 2006). Die Usability ist damit das Maß der vom Nutzer wahrgenommenen Qualität der Zielerfüllung.

Als methodischer Weg zur Erzeugung der Eigenschaft Usability kommt das Usability Engineering zum Einsatz. Dieses ist ein Teilprozess der Entwicklung und Gestaltung technischer Systeme und ergänzt das klassische Software Engineering um ergonomische Aspekte (Sarodnick & Brau, 2006, S. 19). Das Usability Engineering stellt verschiedene Prozessmodelle bereit, deren Phasen durch verschiedene Methoden unterstützt werden können. Existierende Prozessmodelle sind das Vorgehensmodell der Daimler Forschung (Wandke, Oed, Metzker, Ballegooy, & Nitschke, 2001), der Usability Engineering Lifecycle nach Mayhew (Mayhew, 1999) sowie das Prozessmodell Usability Engineering nach Sarodnick und Brau (Sarodnick & Brau, 2006). Die Modelle ähneln in ihrem Aufbau aufgrund ihrer thematischen Nähe sehr stark den Modellen des Software Engineerings. Alle Modelle des Usability Engineering weisen daher einen ähnlichen

Aufbau auf. Startpunkt jedes Modells stellt das Projektmanagement dar, innerhalb dessen rechtliche Grundlagen und Normen zu evaluieren sind. Nach erfolgtem Projektstart findet die Analyse als zweite relevante Projektphase statt, innerhalb derer die Anforderungen des Gesamtsystems aus ergonomischen Aspekten erfasst werden. Diese Phase wird durch Methoden der Usability-Anforderungserfassung unterstützt. Auf die Analysephase folgend sind die Prozessschritte Konzeption, Implementierung sowie Einführung durchzuführen. Innerhalb jeder Phase ist der Projektfortschritt auf Übereinstimmung mit den Anforderungen hin zu evaluieren, was unter Zuhilfenahme von Methoden der Usability-Evaluation erfolgt.

Die Wahl der richtigen und relevanten Normen und Gesetze, Methoden der Usability-Anforderungserfassung, als auch Methoden der Usability-Evaluation gestaltet sich aufgrund deren Vielfalt und Intransparenz jedoch äußerst schwierig. Klare Übersichten für Usability-Verantwortliche sind nicht verfügbar. Die vorliegende Arbeit trägt diesem Umstand Rechnung und stellt für jeden der drei Prozessschritte sowohl Guidelines als auch Methodenbaukästen bereit, welche die betroffenen Personen in ihrer Auswahl unterstützen. Die Guidelines führen den Nutzer in einem ersten Schritt transparent durch einen projektspezifisch zu beantwortenden Fragekatalog, der eine schnelle und bedarfsgerechte Auswahl relevanter Gesetze oder Methoden liefert. Die Methodenbaukästen eignen sich darauf aufbauend, um weiterführende Informationen zur Gesetz- und Methodenauswahl zu erhalten und diese weiter zu verfeinern. Sowohl Guidelines als auch Methodenbaukästen werden in der Folge für die Prozessschritte Usability-Projektmanagement, Usability-Anforderungserfassung und Usability-Evaluation vorgestellt.

#### **4.6.2 Guidelines und Baukästen des Usability-Projektmanagements**

Das Usability-Projektmanagement dient dem Planen, Steuern und Kontrollieren von Usabilityaspekten in Softwareprojekten. Unter den Planungsaspekt fällt unter anderem die Auswahl der für das jeweilige Projekt relevanten Normen und Gesetze, die sich mit dem Thema Usability beschäftigen. Die Auswahl gestaltet sich aufgrund deren Intransparenz jedoch häufig schwierig und ist mit größerem Einarbeitungsaufwand verbunden. In einem ersten Schritt wird daher eine Guideline bereitgestellt, welche es Projektmanagern erlaubt, im Rahmen eines geführten Fragenkatalogs die für ihr Projekt relevanten Normen und Gesetze zu identifizieren. In der Folge sollen die Inhalte der Guideline kurz erläutert werden.

Bindend für alle Softwareprojekte im deutschen Rechtsgebiet ist die Einhaltung der Bildschirmarbeitsverordnung (BildschArbV), die Gestaltungsrichtlinien für Softwareprodukte festlegt. In ihr wird zusätzlich die Einhaltung der DIN EN ISO 9241 Norm zur Ergonomie der Mensch-System-Interaktion gefordert. Die Einhaltung dieser beiden Gesetze und Normen ist für alle Unternehmen verbindlich. Teil 210 der ISO 9241 beschreibt dabei einen Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme, der nur dann obligatorisch ist, wenn

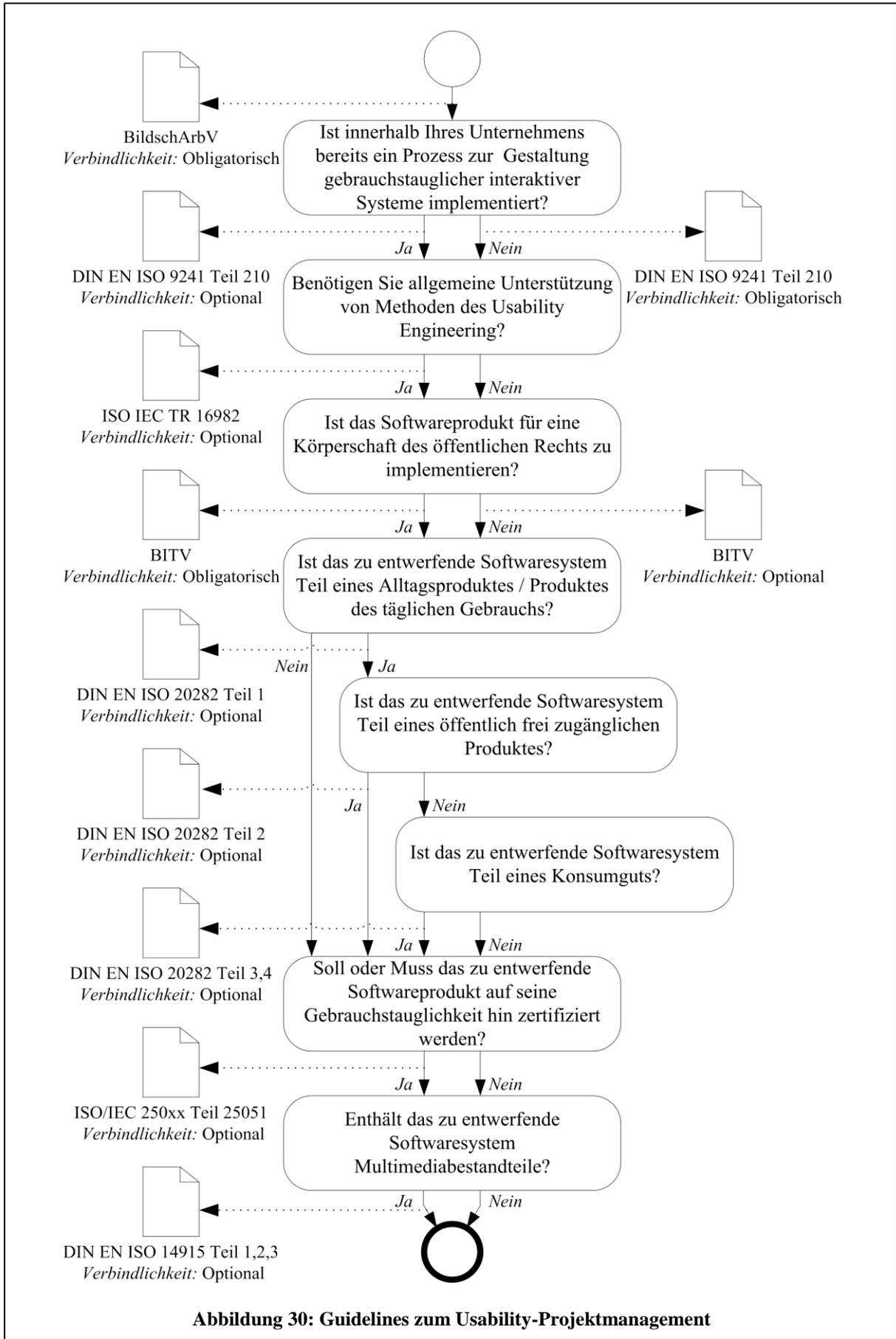


Abbildung 30: Guidelines zum Usability-Projektmanagement

bislang kein solcher Prozess im Unternehmen implementiert ist. Diese Situation bildet die Eingangsfrage der Guideline. Die zweite Fragestellung klärt, ob eine allgemeine Unterstützung von Usability-Methoden gewünscht wird. Hier sei zusätzlich auf die detaillierte Methodenübersicht im Rahmen dieser Arbeit verweisen. In der dritten Fragestellung wird auf die Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz (BITV) eingegangen, die für Körperschaften des öffentlichen Rechts verbindlich vorgeschrieben ist. Die BITV bildet jedoch auch für alle weiteren Unternehmen eine zu bedenkende Grundlage bei der Erstellung barrierefreier Softwareprodukte. Die Fragen 4 bis 6 gehen auf Spezifika von Produkten des täglichen Gebrauchs ein. Je nach Beantwortung der Fragen sind Teile der Norm DIN EN ISO 20282 einzuhalten, die eine Spezifizierung der ISO 9241 für Alltagsprodukte darstellt. Die siebte und damit vorletzte Frage beschäftigt sich mit der Notwendigkeit der Zertifizierung einer Software hinsichtlich derer Usability. Dies ist im Rahmen der ISO/IEC 250xx Normenreihe möglich, genau bildet der Teil 25051 hierfür die Grundlage. Enthält das zu entwerfende Softwareprodukt viele Multimediabestandteile kann DIN EN ISO 14915 zu Rate gezogen werden, welche sich mit der Software-Ergonomie für Multimedia Benutzungsschnittstellen befasst.

Ergebnis des Durchlaufens der Guideline ist eine Sammlung verbindlicher und optionaler Normen und Gesetze, welche die Entwicklung ergonomischer Softwareprodukte unterstützen. Einen detaillierten Aufschluss, inwieweit die Normen und Gesetze das Projekt unterstützen können, gibt in einem zweiten Schritt der Baukasten des Usability-Projektmanagements. Darin werden die wichtigsten Normen und Gesetze aufgeführt, womit die in der Guideline gesammelten Dokumente verglichen und den zuständigen Personen und Projektschritten zugeordnet werden können. Im Detail sind im Baukasten folgende Informationen hinterlegt: Zunächst werden die Normen und Gesetze mitsamt ihren Teildokumenten aufgeschlüsselt. Die Dokumentenart gibt in der Folge an, welchen Status ein Dokument erreicht hat - darunter fallen Verordnungen (Gesetze), Normen, Vornormen, Spezifikationen und Technische Berichte. Unter betroffene Produktarten wird ersichtlich, für welche Arten von Softwareprodukten die Dokumente eine Relevanz aufweisen, während unter Punkt die Verbindlichkeit der Dokumente für Produkt- und Organisationsarten aufgeschlüsselt wird. Die letzten beiden Spalten erlauben die Zuordnung der Dokumente an die verantwortlichen Personen sowie eine Zuordnung zu spezifischen Projektphasen.

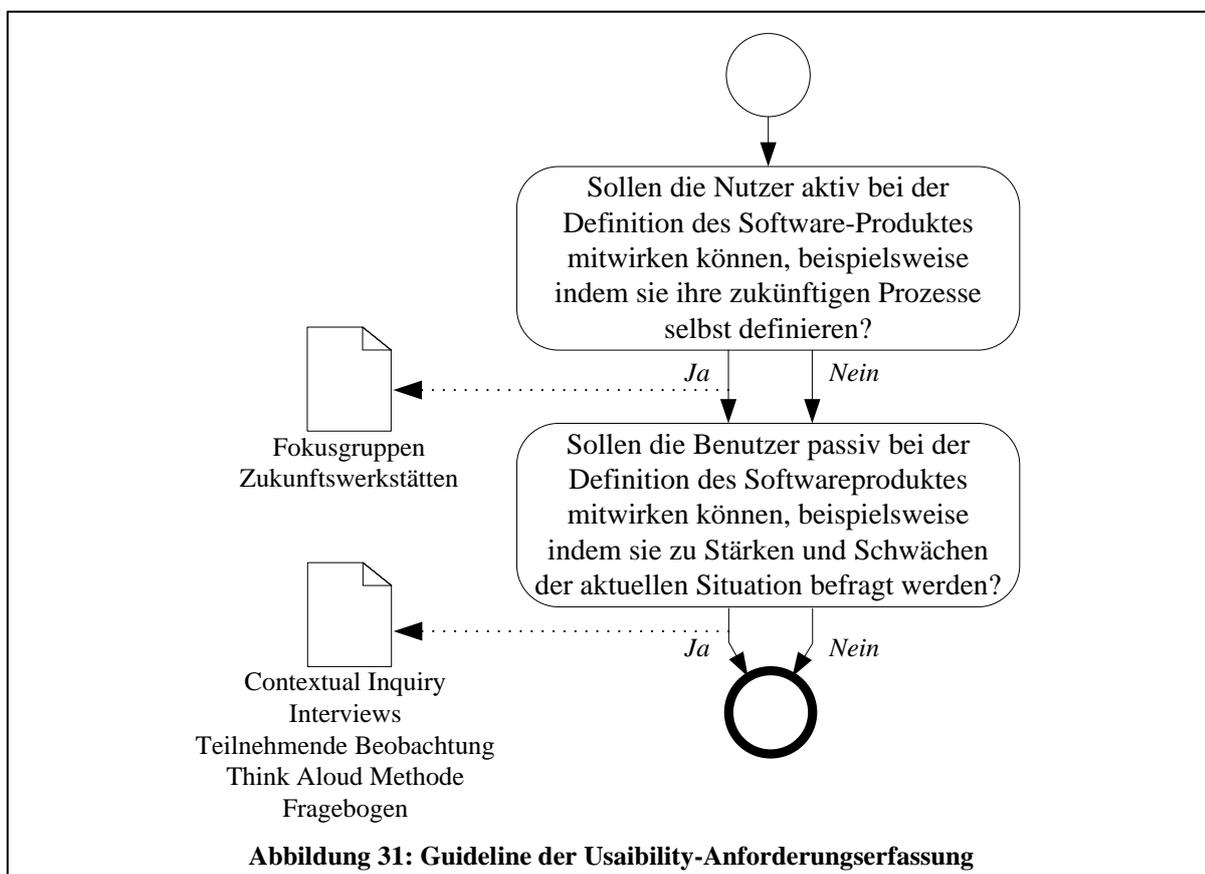
Der Projektmanager hat somit durch die Anwendung der Guideline und des Baukastens die Möglichkeit, die benötigten Normen und Gesetze zu identifizieren, sie den Projektphasen zuzuordnen und den verantwortlichen Personen zur Einhaltung zu übergeben.

Norm/Gesetz	Normbeschreibung	Teilnorm	Teilnormbeschreibung	Dokumentenart	Betroffene Produktarten	Verbindlichkeit	Zielgruppe	Projektrelevanz	Nutzermitwirkung
<b>Norm/Gesetz</b>	<b>Normbeschreibung</b>	<b>Teilnorm</b>	<b>Teilnormbeschreibung</b>						
<b>BildschArbV</b>	Bildschirmarbeitsverordnung			Gesetz	alle SW-Produkte	verbindlich	Projektmanager	Evaluation	Keine Passiv Aktiv
<b>DIN EN ISO 9241</b>	Ergonomie der Mensch-System-Interaktion	Teil 110	Grundsätze der Dialoggestaltung	Norm	alle SW-Produkte	verbindlich durch BildschArbV			
		Teil 11	Anforderungen an die Gebrauchstauglichkeit	Norm	alle SW-Produkte	verbindlich durch BildschArbV			
		Teil 210	Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme	Norm	alle SW-Produkte	verbindlich durch BildschArbV			
		Teil 1	Einfachheit der Handhabung von Produkten des täglichen Gebrauchs	Norm	Alltagsprodukte	keine Verbindlichkeit			
<b>DIN EN ISO 20282</b>	Ergonomie der Mensch-System-Interaktion	Teil 2	Prüfverfahren für öffentlich zugängliche Produkte	Vornorm	öffentlich zugängliche Produkte	keine Verbindlichkeit			
		Teil 3	Testmethode für Konsumgüter	Spezifikation	Konsumgüter	keine Verbindlichkeit			
		Teil 4	Testmethode für die Installation von Konsumgütern	Spezifikation	Konsumgüter	keine Verbindlichkeit			
			Unterstützung für ISO 9241 durch Methoden des Usability Engineering	Technischer Bericht	alle SW-Produkte	keine Verbindlichkeit			
<b>ISO IEC TR 16982</b>	Ergonomie der Mensch-System-Interaktion		Barrierefreie Softwareprodukte	Gesetz	alle SW-Produkte	für Körperschaften des öffentlichen Rechts			
<b>BITV</b>	Verordnung zur Schaffung barrierefreier IT nach dem Behindertengleichstellungsgesetz								
<b>ISO/IEC 250xx</b>	Software-Engineering – Qualitätskriterien und Bewertung von Softwareprodukten (SQuaRE)	Teil 25051	Software-Engineering. Softwareproduktbewertung. Qualitätsanforderungen an kommerzielle serienmäßig produzierte Softwareprodukte	Norm	alle SW-Produkte	keine Verbindlichkeit			
		Teil 1	Gestaltungsgrundsätze und Rahmenbedingungen	Norm	Multimediaprodukte	keine Verbindlichkeit			
		Teil 2	Multimedia-Navigation und Steuerung	Norm	Multimediaprodukte	keine Verbindlichkeit			
<b>DIN EN ISO 14915</b>	Software-Ergonomie für Multimedia Benutzungsschnittstellen	Teil 3	Auswahl und Kombination von Medien	Norm	Multimediaprodukte	keine Verbindlichkeit			

### 4.6.3 Guideline und Baukasten der Usability-Anforderungserfassung

Im Rahmen der Usability-Anforderungserfassung werden die mit der Thematik der Usability verbundenen Projektanforderungen erfasst. Von hoher Relevanz ist dabei eine Kenntnis der Endanwender sowie des zugrundeliegenden Nutzungskontexts der zukünftigen Lösung. Bei der Erfassung der Anforderungen können die auch im herkömmlichen Requirements Engineering verwendeten Methoden Einsatz finden. Im Rahmen der vorliegenden Arbeit wurden einige dieser Methoden, die im Zusammenhang mit der Anforderungserfassung bei Usability-Themen häufig Erwähnung finden, bewertet.

Der größte Unterschied der betrachteten Methoden findet sich dabei in der Art der Nutzerintegration während der Anforderungserfassung. Die Integration der Nutzer kann sich einerseits darauf beschränken, den zugrundeliegenden Nutzungskontext, Benutzerprofile oder andere Informationen zur aktuellen Situation zu erfassen. Die Konzeption der zukünftigen Lösung findet bei einer solchen passiven Nutzermitwirkung in der Folge ohne Nutzerbeteiligung statt. Andererseits können die Nutzer neben Informanten zur aktuellen Situation auch Ideengeber für die zukünftige Lösung sein und aktiv an deren Gestaltung mitwirken. In diesem Zusammenhang spricht man von aktiver Nutzerbeteiligung. Die Guideline beschränkt sich im Rahmen der Usability-Anforderungserfassung daher lediglich auf die Abfrage, inwieweit eine Nutzerbeteiligung stattfinden soll. Je nach Antwort werden unterschiedliche Methoden



zurückgeliefert, die als Alternativen betrachtet werden können. Die Wahl einer oder mehrerer Alternativen erfolgt daran anknüpfend durch Vergleich der vorgeschlagenen Alternativen innerhalb des Methodenbaukastens zur Usability-Anforderungserfassung. Darin sind die Art der Nutzerbeteiligung sowie die unterstützten Prozessschritte ersichtlich. Daneben können Informationen zum Aufwand sowie zum Nutzen der jeweiligen Methoden eingeholt werden. Die Bewertung ist dabei stark von Evaluator und Erhebungsmaterial abhängig, sollte also nur als Richtschnur verstanden werden und weniger als absolute Größe. Die Pfeilsymbole stehen für hohen (↑), mittleren (→) und niedrigen (↓) Aufwand beziehungsweise Nutzen.

Methode	Nutzer-mitwirkung			Unterstützte Prozessschritte		Aufwand					Nutzen		
	Aktiv	Passiv	Keine	Kontinuierlich	Finale Evaluation	Zeitaufwand	Benötigte Nutzer	Benötigte Evaluatoren	Benötigte Expertise	Materieller Aufwand	Vorhersagekraft	Detaillierungsgrad	Objektivität
Contextual Inquiry		●		●		2-3 Tage	3-6	1+	↑	↓	↑	↑	↑
Interviews		●		●		2-3 Tage	10+	1+	↑	↓	→	→	↑
Teilnehmende Beobachtung		●		●		1-2 Tage	8-12	1+	↑	↓	→	↓	→
Think Aloud Methode		●		●		1-2 Tage	3+	1+	→	→	→	↓	→
Fragebogen		●		●	●	1-2 Tage	30+	1	↓	↓	↑	↓	→
Fokusgruppen	●			●	●	1-2 Tage	8-12	1+	↑	↓	↑	→	↑
Zukunftswerkstätten	●			●		2-3 Tage	8-12	1+	→	↓	↑	→	↑

Tabelle 9: Methodenbaukasten der Usability-Anforderungserfassung

Als Ergebnis erhält der Projektmanager eine Auswahl an Methoden, die innerhalb der Anforderungserfassung zu ergonomischen Gesichtspunkten zum Einsatz kommen können.

#### 4.6.4 Guidelines und Baukasten der Usability-Evaluation

Die Usability-Evaluation dient der Überprüfung des Projektfortschritts auf Übereinstimmung mit den aufgestellten Anforderungen. Auch die Methodenauswahl der Usability-Evaluation sollte vom Projektmanager bereits in der Planungsphase des Projektes durchgeführt werden. Die Methoden der Usability-Evaluation lassen sich dabei in drei Kategorien einteilen: Methoden der Software-Ergonomie ermöglichen die Evaluation genereller Anforderungen ohne Betrachtung des Nutzungskontextes, was jedoch einen zentralen Gegenstandsbereich des Usability-Engineerings darstellt. Methoden der kontinuierlichen Evaluation werden bereits während der Entwicklungsphase zur fortlaufenden Kontrolle des Entwicklungsfortschritts hinsichtlich der Nutzer-Ergonomie angewandt. Als letzte Kategorie dienen Methoden der finalen Evaluation zur Abnahmekontrolle bereits fertiggestellter Softwareprodukte hinsichtlich deren Ergonomie.

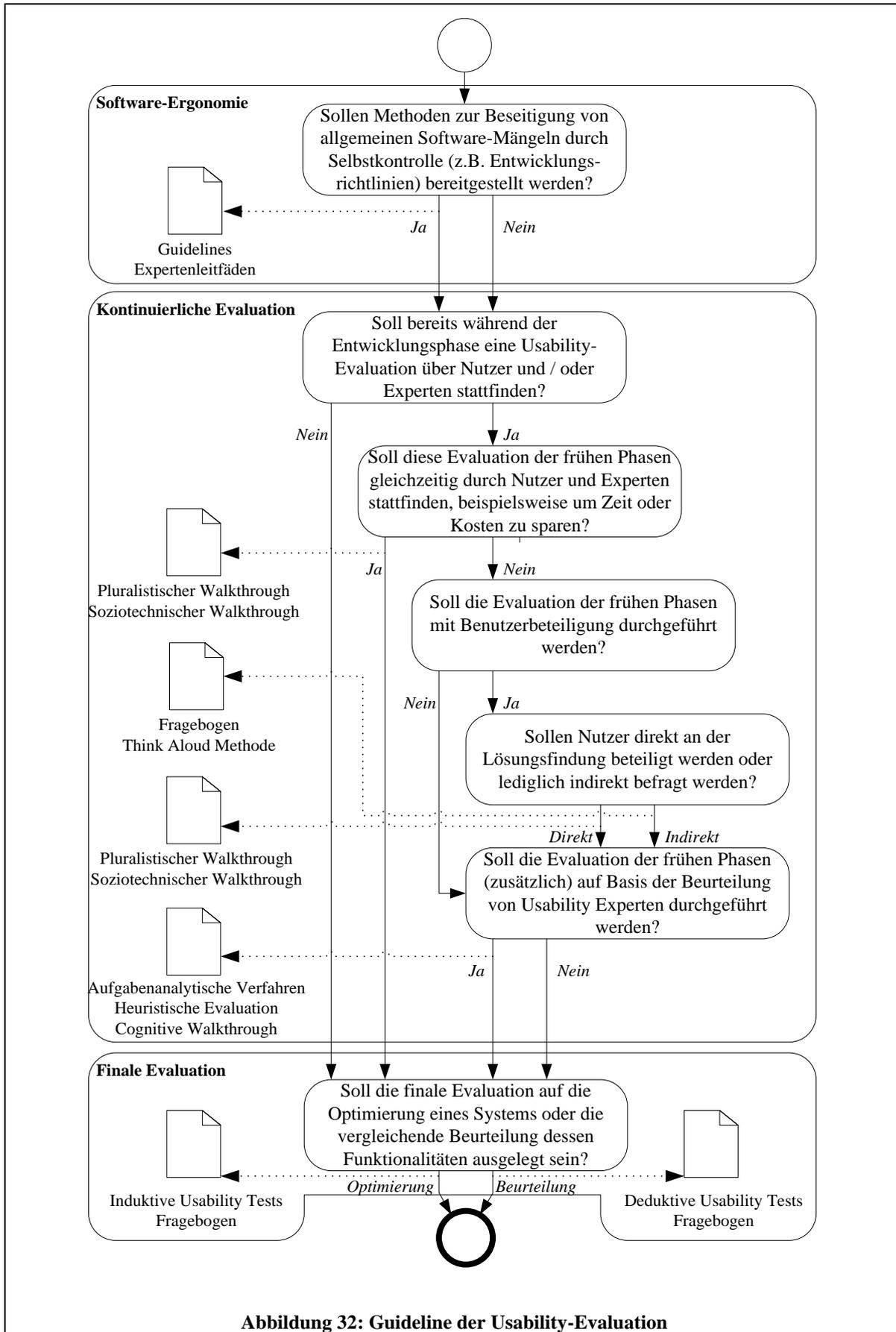


Abbildung 32: Guideline der Usability-Evaluation

Die Guideline gliedert ihre Fragen in die drei vorgestellten Kategorien und fragt zunächst, ob Methoden der Software-Ergonomie Verwendung finden sollen. Hier können Expertenleitfäden oder Usability-Guidelines zum Einsatz kommen. Fragen bezüglich Methoden der kontinuierlichen Evaluation bilden den zentralen Mittelteil der Guideline – hier wird insbesondere zwischen verschiedenen Arten der Nutzerbeteiligung unterschieden. Für jede Art der Nutzerbeteiligung stehen mehrere Methoden zur alternativen Auswahl. Abschließend findet eine Abfrage statt, ob die finale Evaluation lediglich zur Beurteilung oder darüber hinaus zur Optimierung des Softwareprodukts dienen soll. Auch hier können je Antwort mehrere Alternativen zur Auswahl kommen.

Methode	Nutzermitwirkung			Unterstützte Prozessschritte		Evaluationsart		Aufwand					Nutzen		
	Aktiv	Passiv	Keine	Kontinuierlich	Finale Evaluation	Formativ	Summativ	Zeitaufwand	Benötigte Nutzer	Benötigte Evaluatoren	Benötigte Expertise	Materieller Aufwand	Vorhersagekraft	Detaillierungsgrad	Objektivität
Guidelines			●	●		➔	⬆	5-10 Tage	0	3+	⬇	⬇	➔	⬆	⬆
Expertenleitfäden			●	●		➔	⬆	3-5 Tage	0	3+	⬇	⬇	➔	➔	⬆
Aufgabenanalytische Verfahren			●	●		⬆	⬇	5-10 Tage	0	3+	⬆	⬇	➔	⬆	⬆
Heuristische Evaluation			●	●		⬆	➔	2-4 Tage	0	3+	➔	⬇	➔	⬆	➔
Cognitive Walkthrough			●	●		⬆	⬇	2-4 Tage	0	3+	➔	⬇	⬆	➔	➔
Pluralistischer Walkthrough	●			●		⬆	⬆	2-4 Tage	1-3	1+	➔	➔	⬆	⬆	⬆
Soziotechnischer Walkthrough	●			●		⬆	⬆	2-4 Tage	1-3	1+	➔	➔	➔	⬆	⬆
Induktive Usability Tests		●			●	⬆	⬇	5-10 Tage	8-12	1+	⬇	⬆	⬆	⬆	➔
Deduktive Usability Tests		●			●	➔	⬆	5-10 Tage	8-12	1+	⬇	⬆	⬆	⬆	➔
Fragebogen		●		●	●	⬆	⬆	1-2 Tage	30+	1+	⬇	⬇	⬆	⬆	➔
Think Aloud Methode		●		●		⬇	⬆	1-2 Tage	3+	1+	➔	➔	➔	⬆	➔

Tabelle 10: Methodenbaukasten der Usability-Evaluation

Das Resultat einer Durchführung der Guideline ist wie im vorangegangenen Abschnitt eine Auswahl an Methoden, die mithilfe des Methodenbaukastens weiter verfeinert werden kann. Neben der Eignung für verschiedene Arten der Nutzermitwirkung und gewisse Phasen im Prozess werden zusätzliche Kriterien wie Evaluationsart, zugrundeliegender Aufwand und resultierender Nutzen miteinander verglichen. Die Evaluationsart differenziert zwischen formativer Evaluation, die direktes Feedback zur Optimierung des betrachteten Programms gibt, und summativer Evaluation, die auf die zusammenfassende Beurteilung der direkten und nachträglichen Effekte und Vorteile des Software-Produktes ausgelegt ist. Das Kriterium Aufwand untergliedert sich in mehrere Unterkriterien wie Zeitaufwand, benötigte Nutzer sowie

benötigte Expertise wie auch der Nutzen sich in mehrere Unterkriterien gliedert. Die Bewertung von Aufwand und Nutzen ist dabei stark projektspezifisch und von der Expertise der Nutzer und Evaluatoren abhängig. Die vorgenommene Bewertung der Kriterien dient daher lediglich als grobe Richtschnur zum initialen Vergleich mehrerer verschiedener Methoden und sollte nicht als absolute Größe aufgefasst werden.

### 4.6.5 Zusammenfassung

Schritt drei der Methodik ‚Design for Customer‘ ermöglicht es, nicht nur die methodische Basis, sondern das Gesamtsystem bestehend aus Mensch, Werkzeug und Aufgabe den Bedürfnissen des Benutzers entsprechend zu gestalten. Das Usability-Projektmanagement, die Usability-Anforderungserfassung und die Usability-Evaluation wurden als die wesentlichen Prozessschritte des Usability Engineerings herausgearbeitet. Für jeden Prozessschritt wurden in der Folge Guidelines und Methodenbaukästen bereitgestellt, welche eindeutige Handlungsoptionen zur Verbesserung der Endanwender-Usability aufzeigen. Die Guidelines stellen Fragekataloge dar, welche den Entscheidungsträger transparent zu einer schnellen und bedarfsgerechten Auswahl relevanter Methoden, Normen und Gesetze führt. Die Methodenbaukästen eignen sich darauf aufbauend, um weiterführende Informationen zur identifizierten Gesetz- und Methodenauswahl zu erhalten und diese weiter zu verfeinern. Dieses Unterkapitel richtet sich insbesondere an Entscheidungsträger beziehungsweise Projektmanager, welche die Gebrauchstauglichkeit des Softwareproduktes zu verantworten haben.

Mit den bislang vorgestellten Schritten eins bis drei der Methodik ‚Design for Customer‘ wird das erste Hauptziel der vorliegenden Arbeit erreicht. Darin wird ein Ansatz zur Kundenintegration gefordert, der sich weniger am Produkt als vielmehr am Kundennutzen ausrichtet und die Entwicklung eines bedürfnisorientierten Lösungskonfigurators verfolgt. Dieser erlaubt eine Konfiguration basierend auf Kundennutzen und Kundenbedürfnissen anstelle von Produktmerkmalen und ist durchgängig an der Gebrauchstauglichkeit und Ergonomie des Nutzers ausgerichtet.

Als weiterer Handlungsbedarf wurde das Fehlen von Ansätzen des Anforderungsmanagements für das Themenumfeld der hybriden Wertschöpfung herausgestellt. Diese sind idealerweise nicht auf die Entwicklungsphase einer Kundenlösung beschränkt, erfolgen automatisiert und insbesondere ohne Generierung eines Mehraufwandes für den Kunden. Im Folgenden wird daher Schritt 4 der Methodik ‚Design for Customer‘ beschrieben, welche die kontinuierliche Rückführung von Kundenanforderungen auf Basis des in den Schritten 1-3 beschriebenen Lösungskonfigurators als kontinuierliche Bezugsquelle ermöglicht.

## 4.7 DfC Kundenanforderungsgenerierung

*Die Kenntnis der vom Kunden artikulierten Bedürfnisse und der daraus abgeleiteten Kaufentscheidungen ermöglicht es, ein repräsentatives Bild des Kunden zu gewinnen und in die Entwicklungsprozesse des Anbieters rückzuführen. Schritt vier der Methodik ‚Design for Customer‘ hat daher zum Ziel, das über den Lösungskonfigurator gesammelte Kundenwissen zu analysieren und in Kundenanforderungen zu überführen. Die Analyse der Wissensdatenbank übernimmt ein Fuzzy-Apriori-Algorithmus, der im Rahmen der vorliegenden Arbeit entwickelt wurde. Dieser kann sowohl wandelnde Markttrends aufspüren, als auch nicht mit dem bestehenden Lösungsportfolio abgedeckte Kundenbedürfnisse identifizieren.*

### 4.7.1 Überblick

Über den in den vorangegangenen Kapiteln beschriebenen Lösungskonfigurator kann das Kaufverhalten des Kunden erfasst und dem Lösungsanbieter in formalisierter Form für die weitere Verarbeitung zur Verfügung gestellt werden. Basierend auf seiner Eingabe wählt der Kunde für ihn relevante Leistungsmerkmale aus und tätigt somit eine Kauftransaktion. Dabei muss der Kunde nicht notwendigerweise die basierend auf der Fuzzy-Regelbasis vorgeschlagenen Leistungsmerkmale auswählen, sondern kann davon abweichende Kaufentscheidungen treffen. Im Extremfall findet der Kunde trotz Systemvorschlag keine für ihn geeigneten Leistungsmerkmale und bricht die Konfiguration ab. Das Kundenverhalten wird dem Unternehmen durch den Lösungskonfigurator in seiner Gänze zur Verfügung gestellt und bildet die Basis für eine weitere Verarbeitung des Kundenwissens. Die Verarbeitung verfolgt im Rahmen der vorliegenden Arbeit zwei Ziele: Die Rückführung von Kundenanforderungen in die Entwicklungsprozesse sowie die Lernfähigkeit des Fuzzy-Systems. Die beiden Anwendungsfelder sollen im Folgenden kurz umrissen werden.

- *Lernfähigkeit des Fuzzy-Systems:* Fuzzy-Logik ist dazu ausgelegt, menschliches Kausalwissen sowie sprachliche Unschärfe zu mathematisieren. Als Nachteil wurde in der Analyse bestehender Ansätze der Künstlichen Intelligenz (siehe Kapitel 3.1) die fehlende Lernfähigkeit von Fuzzy-Systemen herausgestellt. Über die Kenntnis des aktuellen Konsumverhaltens kann das Fuzzy-System kontinuierlich angepasst und somit lernfähig werden.
- *Rückführung von Kundenanforderungen in die Entwicklungsprozesse:* Als einer der wesentlichen Handlungsbedarfe wurde im Rahmen der vorliegenden Arbeit das Fehlen an Ansätzen des Anforderungsmanagements für das Themenumfeld der hybriden Wertschöpfung herausgestellt. Diese sind an den Bedürfnissen und am Nutzen des Kunden

auszurichten. Das über den Lösungskonfigurator erfasste Konsumverhalten des Kunden bietet hierfür eine holistische Basis zum Aufbau einer Wissensdatenbank. Die Kenntnis der vom Kunden artikulierten Bedürfnisse und der daraus abgeleiteten Kaufentscheidungen ermöglicht es, ein repräsentatives Bild des Kunden zu gewinnen und bei Bedarf in die Entwicklungsprozesse des Anbieters rückzuführen. Dabei können sowohl wandelnde Markttrends aufgespürt, als auch nicht mit dem bestehenden Lösungsportfolio abgedeckte Kundenbedürfnisse identifiziert werden.

Abhängig vom Kundenverhalten kann in den beiden zuvor erläuterten Anwendungsfeldern zwischen verschiedenen Anwendungsfällen unterschieden werden. Die Anwendungsfälle stellen unterschiedliche Szenarien des Kundenverhaltens dar, aus denen unterschiedliche Handlungsoptionen abgeleitet werden können. Es lassen sich vier Anwendungsfelder unterscheiden, welche im Folgenden näher erläutert werden.

- *Anwendungsfall 1:* Ein Kunde bewertet eine Menge an Kundeneigenschaften und bekommt vom System eine Menge an zugehörigen Leistungsmerkmalen zurückgeliefert. In der Folge erwirbt der Kunde eine Menge der vom Lösungskonfigurator empfohlenen Leistungsmerkmale. Kundenwünsche und Fuzzy-Regelbasis stimmen überein, es ist kein Handlungsbedarf abzuleiten.
- *Anwendungsfall 2:* Ein Kunde bewertet eine Menge an Kundeneigenschaften und bekommt vom System eine Menge an zugehörigen Leistungsmerkmalen zurückgeliefert, aus denen er wählen kann. In der Folge erwirbt der Kunde andere als vom Lösungskonfigurator empfohlene Leistungsmerkmale. Mögliche Gründe sind ein fehlerhaftes Mapping innerhalb der Fuzzy-Regelbasis oder ein wandelnder Markttrend. Bedingt durch einen Markttrend kann sukzessive ein Leistungsmerkmal durch ein anderes ersetzt werden. Beispielsweise wird ein Kunde, der der Exklusivität hohe Beachtung schenkt, im Laufe der Jahre ein Navigationsgerät mit einer immer höheren Auflösung verlangen. Als Handlungsbedarf ist die genaue Ursache zu identifizieren und zu beheben.
- *Anwendungsfall 3:* Ein Kunde bewertet eine Menge an Kundeneigenschaften, das System liefert auf Basis der Kundenbewertung jedoch keine Leistungsmerkmale zurück. Der Kunde tätigt dennoch einen Kaufvorgang. Möglicher Grund ist ein fehlendes Mapping in der Fuzzy-Regelbasis, das in der Folge zu beheben ist.
- *Anwendungsfall 4:* Ein Kunde bewertet eine Menge an Kundeneigenschaften, das System liefert keine Leistungsmerkmale oder eine Menge an Leistungsmerkmalen zurück. Der Kunde bricht den Konfigurationsvorgang jedoch ab. Grund ist das Fehlen eines Markttrends im Lösungsportfolio. Die artikulierten Kundeneigenschaften sind in der Folge als Anforderungen in die Entwicklungsprozesse rückzuführen.

Tabelle 11 gibt eine Übersicht über das im DfC Lösungskonfigurator erfasste Kaufverhalten von Kunden und den abgeleiteten anwendungsfallspezifischen Handlungsbedarf.

	<b>Lernfähigkeit des Fuzzy-Systems</b>	<b>Rückführung von Kundenanforderungen</b>
Anwendungsfall 1	-	-
Anwendungsfall 2	Fehlerhaftes Mapping der Fuzzy-Regelbasis	Wandelnder Markttrend
Anwendungsfall 3	Fehlendes Mapping der Fuzzy-Regelbasis	-
Anwendungsfall 4	-	Fehlender Markttrend im Lösungsportfolio

**Tabelle 11: Im Lösungskonfigurator erfasstes Kaufverhalten und abgeleiteter Handlungsbedarf**

Die identifizierten Anwendungsfälle sowie die Handlungsoptionen sind im Kundenverhalten des Lösungskonfigurators nicht explizit ersichtlich. Daher ist das implizit vorhandene Wissen zunächst zu identifizieren und daraus die oben vorgestellten Handlungsoptionen abzuleiten.

Das in Kapitel 3.3 vorgestellte Themenumfeld des Knowledge Discovery in Databases (KDD) zielt dediziert darauf ab, in umfangreichen Datenbeständen implizit vorhandenes Wissen zu entdecken und explizit zu machen. Insbesondere werden Beziehungsmuster ermittelt, die für einen möglichst großen Anteil der Datenbasis Geltung haben und leicht verständliche Zusammenhänge in den Daten zum Ausdruck bringen. Im Speziellen sind Verfahren zur Assoziationsanalyse als Teilbereich des KDD dazu geeignet, die oben erläuterten Anwendungsfälle explizit zu machen. Unter deren Aufgabe fällt es, die Zusammenhänge zwischen vom Kunden gewünschten Eigenschaften und tatsächlich gewählten Merkmalen darzustellen und die vier genannten Anwendungsfelder somit zu identifizieren. Das Ergebnis der Anwendung eines solchen Algorithmus sind Assoziationsregeln, die signifikante Abhängigkeiten zwischen den untersuchten Objekten des Datenbestandes beschreiben. Ein solcher Algorithmus, der unscharfe Datenbestände untersucht, ist wie in Kapitel 3.3.3 ‚Verfahren zur Assoziationsanalyse bei unscharfen Datenbeständen‘ nur unzureichend verfügbar.

In den folgenden Abschnitten wurde ein solcher Algorithmus entwickelt, welcher den Anforderungen der Methodik ‚Design for Customer‘ genügt. Hierzu werden zunächst einige grundlegende Definitionen vorgenommen und der Leser in die Begrifflichkeiten von Verfahren zur Assoziationsanalyse eingeführt. In der Folge wird der im Rahmen der vorliegenden Arbeit entstandene Fuzzy-Apriori-Algorithmus im Detail vorgestellt.

## 4.7.2 Grundlegende Definitionen

Im vorliegenden Abschnitt werden grundlegende Definitionen vorgenommen, die für die Beschreibung sowie zum vereinfachten Verständnis des Fuzzy-Apriori-Algorithmus benötigt

werden. Da der Fuzzy-Apriori-Algorithmus auf den Daten des DfC Fuzzy-Controller bzw. der DfC Fuzzy-Ontologie zurückgreift, finden sich einige der dort verwendeten Variablen auch im vorliegenden Algorithmus wieder. Diese Variablen wurden bereits in Kapitel 4.3 ‚DfC Fuzzy-Controller‘ definiert, weshalb eine redundante Definition im vorliegenden Kontext unterlassen und stattdessen auf die ursprüngliche Deklaration verwiesen wird.

#### Items

Es sei  $\mathcal{I} = \mathcal{P}, \mathcal{C} = p_1, p_2, \dots, p_m, c_1, c_2, \dots, c_n$  eine Menge von Objekten, die Items genannt werden. Innerhalb der Objektmenge wird zwischen Eigenschafts-Items  $\mathcal{P} = p_1, p_2, \dots, p_m$ , welche die von Kunden wählbare Eigenschaften beschreiben, und Merkmals-Items  $\mathcal{C} = c_1, c_2, \dots, c_n$ , welche die von Kunden erwerbenden Leistungsmerkmale beinhalten, unterschieden.



#### Verständnisbeispiel

Innerhalb des DfC Fuzzy-Controllers wurden Kundeneigenschaften (z.B. Exklusivität) und Anbietermerkmale (z. B. Echtglas-Oberfläche) als linguistische Variablen definiert. Die linguistischen Variablen des Fuzzy-Systems werden bei der Beschreibung eines Apriori-Algorithmus als Items benannt.

#### Transaktionen der Datenbank

$\mathcal{D}$  ist eine Datenbank mit Transaktionen  $\mathcal{T}$ , innerhalb derer ein Kunde eine Menge an Eigenschafts-Items bewertet und eine Menge an Merkmals-Items gekauft hat. Folglich beinhaltet jede Transaktion eine Menge an Items  $\mathcal{I}$ , in Abhängigkeit von der Auswahl des Kunden. Jede Transaktion  $\mathcal{T}$  wird durch einen Vektor repräsentiert, mit  $\mathcal{T}[k] = x_{p_k}$  für ein Eigenschafts-Item  $p_k$ , bei dem der Kunde  $p_k$  mit  $x_{p_k} \in [0,1]$  bewertet hat. Es ergibt sich  $\mathcal{T}[k] = 1$ , falls der Kunde vollständig mit der Eigenschaft übereinstimmt und  $\mathcal{T}[k] = 0$ , falls er sich vollständig von ihr distanziert. Für ein Merkmals-Item  $c_k$  ergibt sich  $\mathcal{T}[k] = 1$ , falls der Kunde das Leistungsmerkmal kauft und  $\mathcal{T}[k] = 0$  falls nicht.



#### Verständnisbeispiel

Eine beispielhafte Transaktion  $\mathcal{T}$  in einer Datenbank  $\mathcal{D}$  kann wie folgt aussehen:

$p_1 = \text{Exklusivität}, \mathcal{T}[1] = 0,9$

$c_3 = \text{Echtglas-Oberfläche}, \mathcal{T}[2] = 1$

Ein Kunde hat demzufolge eine Kundeneigenschaft (Eigenschafts-Item) „Exklusivität“ mit dem Wert 0,9 bewertet und darauf basierend eine Leistung mit einer „Echtglas-Oberfläche“ gekauft.

**f-Item**

Ein f-Item ist ein Paar aus Eigenschafts-Item  $p_i$  und zum Eigenschafts-Item gehöri-  
gem linguistischen Term  $\mathcal{L}\mathcal{T}_{i,j}$ . Die Notation eines f-Items  $fi$  lautet wie folgt:  $fi = (p_i, \mathcal{L}\mathcal{T}_{i,j})$

**Verständnisbeispiel**

Zur Vereinfachung der Beschreibung des Algorithmus wird der Begriff f-Item definiert. Dieser besteht aus einer linguistischen Variable und einem linguistischen Term (siehe Kapitel 4.3.1.1 und 4.3.1.2) und kann wie folgt aussehen:

$$fi = (p_i, \mathcal{L}\mathcal{T}_{i,j}) = (\text{Exklusivität, hoch})$$

Ziel des Algorithmus ist es, Regeln bestehend aus f-Items zu generieren und somit Assoziationen zwischen verschiedenen f-Items herauszufinden.

**Fuzzy-Kandidaten**

Ein Fuzzy-Kandidat  $fc_k$  ist eine Kandidatenmenge des Fuzzy-Apriori-Algorithmus. Im Gegensatz zu Kandidaten des herkömmlichen Apriori-Algorithmus besteht ein Fuzzy-Kandidat aus f-Items  $fi$ , welche die unscharfe Information mit in den Algorithmus einbeziehen. Die Anzahl an Elementen  $k$  in einem Fuzzy-Kandidaten  $fc_k$  ist abhängig von der Iteration des Algorithmus. In Iteration  $k$  beinhaltet jeder Fuzzy-Kandidat  $k$  f-Items als Elemente. Die Notation eines Fuzzy-Kandidaten für Iteration  $k$  lautet wie folgt:  $fc_k = \{(p_1, \mu_{\mathcal{L}\mathcal{T}'_{1,1}}), (p_2, \mu_{\mathcal{L}\mathcal{T}'_{2,2}}), \dots, (p_k, \mu_{\mathcal{L}\mathcal{T}'_{k,k}})\}$  mit  $p_1, p_2, \dots, p_k \in \mathcal{P}$ . Die Kandidatenmenge  $\mathcal{C}_k$  fasst alle Fuzzy-Kandidaten  $fc_k$  aus Iteration  $k$  zusammen.

**Verständnisbeispiel**

Ein Fuzzy-Kandidat  $fc_k$  ist die Repräsentation eines Kandidaten im herkömmlichen Apriori-Algorithmus angereichert mit unscharfen Informationen. Die unscharfen Informationen manifestieren sich durch die Verwendung von f-Items anstelle herkömmlicher Items für die Definition des Kandidaten. Beispielhaft kann ein Fuzzy-Kandidat  $fc_2$  in Iteration 2 des Algorithmus wie folgt aussehen:

$$fc_2 = \{(p_1, \mu_{\mathcal{L}\mathcal{T}'_{1,1}}), (p_2, \mu_{\mathcal{L}\mathcal{T}'_{2,2}})\} = \{(\text{Exklusivität, hoch}), (\text{Geländetauglichkeit, niedrig})\}$$

Damit wurden die erforderlichen grundlegenden Variablendeklarationen durchgeführt und eine Einführung in die terminologischen Grundbegriffe gegeben. In der Folge wird der Fuzzy-Apriori-Algorithmus vorgestellt.

### 4.7.3 Der Fuzzy-Apriori-Algorithmus

Innerhalb des vorliegenden Abschnittes wird der Fuzzy-Apriori-Algorithmus zur Erstellung unscharfer Fuzzy-Assoziationsregeln für die Methodik ‚Design for Customer‘ vorgestellt (siehe auch (Burger & Ovtcharova, 2013)). Basis des Algorithmus bildet der weitverbreitete Apriori-Algorithmus zur Generierung von Assoziationsregeln unter Sicherheit (Agrawal & Srikant, 1994). Obwohl die grundlegende Struktur dem Aufbau des hier vorgestellten Algorithmus dem Apriori-Algorithmus ähnelt, unterscheiden sich beide in den folgenden Punkten:

- *Unscharfer Datenbestand:* Während der Apriori-Algorithmus lediglich unter Sicherheit arbeitet, wird dessen Funktionalität im vorliegenden Algorithmus auf einen zugrundeliegenden unscharfen Datenbestand ausgeweitet.
- *Einbezug von Zugehörigkeitsfunktionen:* Um auf einem unscharfen Datenbestand aufsetzen zu können, sind neben den herkömmlichen Itemmengen auch Fuzzy-Zugehörigkeitsfunktionen zu berücksichtigen, welche einem Itemset einen gewissen Sicherheitsgrad zuweisen.
- *Bestimmung des Supports:* Im herkömmlichen Apriori-Algorithmus ist ein Itemset entweder vollständig innerhalb einer Transaktion vorhanden oder aber überhaupt nicht. Im vorliegenden Anwendungsfall kann ein Item auch lediglich teilweise zu einem Itemset gehören, weshalb die Berechnung des Supports an den zugrundeliegenden Sicherheitsgrad angepasst wird.
- *Standardabweichung als Qualitätsmaß:* Die Itemsets im herkömmlichen Apriori-Algorithmus weisen vollständige Sicherheit auf, dahingegen können die Sicherheitsgrade unscharfer Itemsets stark voneinander abweichen. Um dennoch eine Aussage über die Ähnlichkeit mehrerer Itemsets treffen zu können, wird die Standardabweichung als Qualitätskriterium hinzugezogen.
- *Konfidenzintervalle statt Mindestkonfidenz:* Um die Zugehörigkeit eines linguistischen Terms zu einer Assoziationsregel herstellen zu können, wird nicht wie beim herkömmlichen Algorithmus eine Mindestkonfidenz, sondern stattdessen je linguistischem Term ein Konfidenzintervall benötigt.

Wie bereits erwähnt verwenden die Algorithmen trotz der erläuterten Unterschiede eine ähnliche Struktur. Beide Algorithmen untergliedern ihren Lösungsansatz in zwei Teilansätze: In einem ersten Schritt werden große Item- bzw. Objektmengen gesucht, welche innerhalb der Grenzen gewisser Schwellenwerte liegen. In der Folge werden aus diesen großen Objektmengen Regeln generiert, deren Konfidenz den Bestimmungen eines weiteren Schwellenwertes genügt. Abbildung 33 zeigt den schematischen Aufbau des Fuzzy-Apriori-Algorithmus auf. Dabei ist ersichtlich, dass ein zusätzlicher Schritt vor der Bestimmung der großen Objektmengen eingefügt

**Input:** Eine Datenbank  $\mathcal{D}$  mit Transaktionen  $\mathcal{T}$ , zu den Eigenschafts-Items  $p_i$  und Merkmals-Items  $c_i$  gehörende Zugehörigkeitsfunktionen  $\mu_{\mathcal{L}\mathcal{T}_{i,j}}$ , ein vordefinierter Fuzzy-Mindestsupport  $\sigma_{\text{fuzzy-supp}}$ , eine vordefinierte Fuzzy-Maximalstandardabweichung  $\sigma_{\text{fuzzy-dev}}$ , vordefinierte Fuzzy-Konfidenzintervalle  $\lambda_{\text{ges}}$ .

**Output:** Eine Menge an Fuzzy-Assoziationsregeln  $FAR$ , die Eigenschafts-Items und Merkmals-Items durch die Zugehörigkeitsfunktionen  $\mu_{\mathcal{L}\mathcal{T}_{i,j}}$  verbinden.

- (1) Für jedes Merkmals-Item  $c_{\text{recent}}$  {
  - //**Schritt 1:** Bereite Datenbank auf, indem Zugehörigkeitsgrade zu den f-Items berechnet werden.
- (2)  $\mathcal{D}_{\text{preprocess}} = \text{preprocess\_database}$ 
  - //**Schritt 2:** Erstelle große Fuzzy-Itemmengen mit zunächst einem Element durch Überprüfung auf Fuzzy-Support und Fuzzy-Standardabweichung und erhöhe die Elementmenge iterativ.
- (3)  $\mathcal{L}_1 = \text{generate\_fuzzy\_1\_itemsets}$
- (4)  $\mathcal{L}_{\text{all}} = \text{generate\_fuzzy\_k\_itemsets}$ 
  - //**Schritt 3:** Erstelle Fuzzy-Assoziationsregeln und verifiziere diese durch ihre Konfidenz.
- (5)  $FAR = \text{generate\_FAR}$
- (6) }

**Abbildung 33: Schematischer Grundaufbau des Fuzzy-Apriori-Algorithmus**

wurde. Dieser zum herkömmlichen Apriori Algorithmus zusätzliche Schritt dient der Aufbereitung der Eingangsinformationen in eine weiter verarbeitbare Form. Die drei resultierenden Teilansätze des Fuzzy-Apriori-Algorithmus lassen sich kurz wie folgt beschreiben:

1. Aufbereitung der Datenbank mit Kundentransaktionen, indem über die Zugehörigkeitsfunktionen die Sicherheitsgrade zu jedem f-Item errechnet werden. Der Schritt dient der Leistungsverbesserung des Algorithmus, da in der Folge auf diese Daten mehrfach zugegriffen wird.
2. Generierung sämtlicher Kombinationen an f-Items, welche innerhalb gewisser Grenzen liegen. Die Grenzen werden im Falle des Fuzzy-Apriori-Algorithmus durch den Fuzzy-Mindestsupport sowie durch die Fuzzy-Standardabweichung festgelegt. Für jede Kombination an f-Items wird zunächst überprüft, ob der Support oberhalb des Fuzzy-Mindestsupports liegt. In der Folge darf die Standardabweichung der Sicherheitsgrade einzelner f-Items nicht über den durch die Fuzzy-Maximal-Standardabweichung definierten Grenzen liegen. Genügt eine Kombination an f-Items den geforderten

Kriterien, so wird die Kombination große Objektmenge genannt, ansonsten kleine Objektmenge.

3. Für jede große Objektmenge sind Regeln der Form  $\mathcal{X} \rightarrow \mathcal{Y}$  zu bilden, wobei  $\mathcal{X}$  sämtliche f-Items der großen Objektmenge und  $\mathcal{Y}$  genau ein Merkmals-Item darstellt, für welches der Algorithmus durchgeführt wurde.

Der Algorithmus wird dabei für jedes in der Gesamtdatenbasis  $\mathcal{D}_{ges}$  vorhandene Merkmals-Item separat durchgeführt. Somit erlaubt der Algorithmus die Generierung von Assoziationsregeln für jedes Anbietermerkmal. Die Reduktion der Gesamtdatenbasis  $\mathcal{D}_{ges}$  auf die Teilmenge, in der Anbietermerkmal  $c_{recent}$  käuflich erworben wurde führt zu Teildatenbasis  $\mathcal{D}$  als Ausgangsbasis des Algorithmus. Ergebnis eines Durchlaufes für ein Anbietermerkmal  $c_{recent}$  ist die Menge aller Assoziationsregeln zu  $c_{recent}$ . Im folgenden Abschnitt wird der Fuzzy-Apriori-Algorithmus im Detail vorgestellt und insbesondere auf dessen Realisierung in Pseudocode eingegangen.

##### 4.7.3.1 Schritt 1: Aufbereitung der Daten

Die Methode `PREPROCESS_INPUT` dient der Aufbereitung der Datenbank  $\mathcal{D}$ , welche einen der Eingangsparameter des Algorithmus darstellt. Die Datenbank beinhaltet Informationen zu sämtlichen Kundentransaktionen  $\mathcal{T}$ , die über den DfC Konfigurator vorgenommen wurden. Eine Transaktion in der Datenbank umfasst die vom Kunden gewählten Eigenschaften mitsamt der zu den Eigenschaften gehörigen Kundenbewertungen. Daneben werden innerhalb der Transaktion sämtliche Leistungsmerkmale erfasst, welche der Kunde basierend auf seiner Eigenschaftenauswahl erworben hat. Basierend auf den Daten wird es einem Unternehmen ermöglicht, Rückschlüsse über Zusammenhänge zwischen Kundenbedürfnissen und tatsächlichen Käufen zu ziehen. Um eine schnellere Ausführung des Algorithmus zu ermöglichen, wird die Datenbank mit Transaktionen in der vorliegenden Methode um zusätzliche Sicherheitsgrade angereichert und als Datenbank  $\mathcal{D}_{preprocess}$  zurückgeliefert. Die resultierende Datenbank  $\mathcal{D}_{preprocess}$  dient in der Folge als Eingabeparameter für alle der nachfolgenden Methoden. Die Berechnung der Sicherheitsgrade erfolgt über die Zugehörigkeitsfunktionen zu jedem f-Item. Sobald zu jedem f-Item der zugehörige Sicherheitsgrad errechnet wurde, ist die Datenbank  $\mathcal{D}_{preprocess}$  vollständig und wird von der Methode als Ergebnis zurückgeliefert. Der erste Schritt zur Aufbereitung der Eingabedaten des Fuzzy-Apriori-Algorithmus ist somit abgeschlossen und es kann begonnen werden, große Objektmengen zu bilden.

**Methode: PREPROCESS\_INPUT**

**Input:** Eine Datenbank  $\mathcal{D}$  mit Transaktionen  $\mathcal{T}$ , zu den Eigenschafts-Items  $p_i$  und Merkmals-Items  $c_i$  gehörige Zugehörigkeitsfunktionen  $\mu_{\mathcal{L}\mathcal{T}_{i,j}}$

**Output:** Eine Datenbank  $\mathcal{D}_{preprocess}$  mit Transaktionen zu Merkmals-Item  $c_{recent}$ , welche die Informationen in Datenbank  $\mathcal{D}$  um die Sicherheitsgrade  $\mu_{\mathcal{L}\mathcal{T}_{i,j}}(x_{p_i})$  der Eigenschafts-Items  $p_i$  basierend auf Kundeneingabe  $x_{p_i}$  zu Eigenschafts-Item  $p_i$  und Zugehörigkeitsfunktionen  $\mu_{\mathcal{L}\mathcal{T}_{i,j}}$  erweitert.

- (1) Selektiere diejenigen Transaktionen  $\mathcal{T}$  aus  $\mathcal{D}$ , in denen  $c_{recent}$  vom Kunden gekauft wurde als  $\mathcal{D}_{c_{recent}}$
- (2)  $\mathcal{D}_{preprocess} = \mathcal{D}_{c_{recent}}$
- (3) Für jede Transaktion  $\mathcal{T}$  in Datenbank  $\mathcal{D}_{c_{recent}}$  {
- (4)     Für jedes Eigenschafts-Item  $p_i$  {
- (5)         Für jede Zugehörigkeitsfunktion  $\mu_{\mathcal{L}\mathcal{T}_{i,j}}$  zu Eigenschafts-Item  $p_i$  {
- (6)             Berechne den Sicherheitsgrad  $\mu_{\mathcal{L}\mathcal{T}_{i,j}}(x_{p_i})$  basierend auf der Kundeneingabe  $x_{p_i}$  und Zugehörigkeitsfunktion  $\mu_{\mathcal{L}\mathcal{T}_{i,j}}$  zu Eigenschafts-Item  $p_i$
- (7)             Speichere  $\mu_{\mathcal{L}\mathcal{T}_{i,j}}(x_{p_i})$  in der Datenbank  $\mathcal{D}_{preprocess}$
- (8)             }
- (9)         }
- (10)     }
- (11) Setze  $k = 1$
- (12) Liefere  $\mathcal{D}_{preprocess}$  zurück

Abbildung 34: Die Methode PREPROCESS\_INPUT des Fuzzy-Apriori-Algorithmus

#### 4.7.3.2 Schritt 2: Generierung großer Objektmengen

Ziel des zweiten Schrittes des Fuzzy-Apriori-Algorithmus ist die Generierung großer Objektmengen. Wie auch im herkömmlichen Apriori-Algorithmus wird zwischen der Erstellung der großen Objektmengen mit einem Element und der Erstellung großer Objektmengen mit  $k$  Elementen, wobei  $k \geq 2$  ist, unterschieden. Der Generierung der großen Objektmengen mit einem Element dient die Methode GENERATE\_FUZZY\_1\_ITEMSETS, welche unter anderem Datenbank  $\mathcal{D}_{preprocess}$  als Eingabeparameter enthält. Die Generierung der großen Objektmengen mit einem Element erfolgt, indem für jedes f-Item der Support und die Standardabweichung errechnet werden. Ist der Support eines f-Items größer als der geforderte Fuzzy-Mindestsupport und gleichzeitig die Standardabweichung kleiner als die geforderte Fuzzy-Mindeststandardabweichung, so wird das f-Item als Element in die große Objektmenge  $\mathcal{L}_1$

**Method:** GENERATE\_FUZZY\_1\_ITEMSETS

**Input:** Eine Datenbank  $\mathcal{D}_{preprocess}$  mit Transaktionen  $\mathcal{T}$ , die Anzahl an Transaktionen in Datenbank  $\mathcal{D}$ , ein vordefinierter Fuzzy-Mindestsupport  $\sigma_{fuzzy-supp}$ , eine vordefinierte Fuzzy-Maximalstandardabweichung  $\sigma_{fuzzy-dev}$

**Output:** Eine große Fuzzy-Itemmenge  $\mathcal{L}_1$

- (1)  $\mathcal{L}_1 = \emptyset$
- (2)  $\mathcal{L}_{all} = \emptyset$
- (3) Für jedes f-Item  $fi$  in Datenbank  $\mathcal{D}_{preprocess}$  {
- (4)      $count = 0$
- (5)     Für jede Transaktion  $\mathcal{T}$  in  $\mathcal{D}_{preprocess}$  {
- (6)          $count = count + \mu_{\mathcal{L}\mathcal{T}_{i,j}}(x_{p_i})$
- (7)     }
- (8)      $fi.sup = count/|\mathcal{D}|$
- (9)      $fi.dev =$  Standardabweichung der Sicherheitsgrade  $\mu_{\mathcal{L}\mathcal{T}_{i,j}}(x_{p_i})$  in  $\mathcal{D}_{preprocess}$
- (10)     Falls  $(fc_k.sup \geq \sigma_{fuzzy-supp})$  und  $(fc_k.dev \leq \sigma_{fuzzy-dev})$  {
- (11)          $\mathcal{L}_1 = \mathcal{L}_1 \cup \{fi\}$
- (12)     }
- (13) }
- (14)  $\mathcal{L}_{all} = \mathcal{L}_1$
- (15) Liefere  $\mathcal{L}_1$  zurück

**Abbildung 35: Die Methode GENERATE\_FUZZY\_1\_ITEMSETS des Fuzzy-Apriori-Algorithmus**

aufgenommen. Der Support errechnet sich dabei über die Summe der Sicherheitsgrade für ein f-Item über alle Transaktionen in Datenbank  $\mathcal{D}_{preprocess}$  hinweg geteilt durch die Gesamtanzahl an Transaktionen in Datenbank  $\mathcal{D}$ , welche als  $|\mathcal{D}|$  bezeichnet wird. Die Berechnung der Standardabweichung wurde im Pseudocode aus Platzgründen nicht explizit dargestellt, erfolgt aber über die herkömmliche Berechnung einer Standardabweichung und wird daher im Weiteren nicht gesondert erläutert. Ergebnis der Methode ist die Menge aller großen Objektmengen mit einem Element  $\mathcal{L}_1$ , die als Methodenergebnis zurückgeliefert wird.

Ist die Generierung der initialen Menge  $\mathcal{L}_1$ , welche alle großen Objektmengen mit einem Element beinhaltet, abgeschlossen, so können alle weiteren großen Objektmengen mit  $k$  Elementen für  $k \geq 2$  erstellt werden. Die Methode GENERATE\_FUZZY\_K\_ITEMSETS geht dabei iterativ vor, d.h. es werden zunächst alle großen Objektmengen mit 2 Elementen gesucht, dann mit 3 Elementen und so weiter. Um von Mengen mit  $k - 1$  Elementen auf Mengen mit  $k$  Elementen zu kommen, werden in der Methode APRIORI\_GEN wie auch im herkömmlichen Apriori-Algorithmus so genannte Kandidatenmengen  $\mathcal{C}_k$  mit  $k$  Elementen gebildet. Diese

**Methode: GENERATE\_FUZZY\_K\_ITEMSETS**

**Input:** Eine Datenbank  $\mathcal{D}_{preprocess}$  mit Transaktionen  $\mathcal{T}$ , die große Fuzzy-Itemmenge  $\mathcal{L}_{k-1}$ , die Anzahl an Transaktionen in Datenbank  $\mathcal{D}$ , ein vordefinierter Fuzzy-Mindestsupport  $\sigma_{fuzzy-supp}$ , eine vordefinierte Fuzzy-Maximalstandardabweichung  $\sigma_{fuzzy-dev}$

**Output:** Eine große Fuzzy-Itemmenge  $\mathcal{L}_k$

- (1) For ( $k = 2, \mathcal{L}_{k-1} \neq \emptyset, k++$ )
- (2)  $\mathcal{C}_k = \text{fuzzy\_apriori\_gen}(\mathcal{L}_{k-1})$
- (3)  $\mathcal{L}_k = \emptyset$
- (4) Für jeden Fuzzy-Kandidaten  $fc_k$  in  $\mathcal{C}_k$  {
- (5) Selektiere diejenigen Transaktionen  $\mathcal{T}$  aus  $\mathcal{D}_{preprocess}$ , in denen jedes f-Item  $fi$  des Fuzzy-Kandidaten  $fc_k$  einen Sicherheitsgrad  $\mu_{\mathcal{L}_{\mathcal{T}_{ij}}}(x_{\mathcal{P}_i})$  größer als Null aufweist, als  $\mathcal{D}_{preprocess}^T$
- (6) Falls ( $\mathcal{D}_{preprocess}^T = \emptyset$ ) {
- (7)  $fc_k.sup = 0$
- (8)  $fc_k.dev = \infty$
- (9) }
- (10) Sonst {
- (11)  $fc_k.sup = \infty$
- (12)  $fc_k.dev = 0$
- (13) Für jedes f-Item  $fi$  im Fuzzy-Kandidaten  $fc_k$  {
- (14)  $count = 0$
- (15) Für jede Transaktion  $\mathcal{T}$  in  $\mathcal{D}_{preprocess}^T$  {
- (16)  $count = count + \mu_{\mathcal{L}_{\mathcal{T}_{ij}}}(x_{\mathcal{P}_i})$
- (17) }
- (18)  $fi.sup = count/|\mathcal{D}|$
- (19)  $fc_k.sup = \min\{fc_k.sup, fi.sup\}$
- (20)  $fi.dev =$  Standardabweichung der Sicherheitsgrade  $\mu_{\mathcal{L}_{\mathcal{T}_{ij}}}(x_{\mathcal{P}_i})$  in  $\mathcal{D}_{preprocess}^T$
- (21)  $fc_k.dev = \max\{fc_k.dev, fi.dev\}$
- (22) }
- (23) }
- (24) Falls ( $fc_k.sup \geq \sigma_{fuzzy-supp}$ ) und ( $fc_k.dev \leq \sigma_{fuzzy-dev}$ ) {
- (25)  $\mathcal{L}_k = \mathcal{L}_k \cup \{fc_k\}$
- (26) }
- (27) }
- (28)  $\mathcal{L}_{all} = \mathcal{L}_{all} \cup \mathcal{L}_k$

Abbildung 36: Die Methode GENERATE\_FUZZY\_K\_ITEMSETS des Fuzzy-Apriori-Algorithmus

basieren auf den großen Objektmengen aus  $\mathcal{L}_{k-1}$  und stellen potentielle große Objektmengen mit  $k$  Elementen dar. Die Methode GENERATE\_FUZZY\_K\_ITEMSETS bricht die Suche nach großen Objektmengen mit  $k$  Elementen ab, falls die große Objektmenge  $\mathcal{L}_{k-1}$  leer ist. Eine Menge mit  $k$  f-Items wird wie auch in der vorangegangenen Methode nur dann eine große Objektmenge, falls Support und Standardabweichung innerhalb der Grenzen des vordefinierten Fuzzy-Mindestsupports und der vordefinierten Fuzzy-Maximalstandardabweichung liegen. Sämtliche großen Objektmengen  $\mathcal{L}_k$  werden wie auch in der vorangegangenen Methode  $\mathcal{L}_k$  in die große Menge  $\mathcal{L}_{all}$  aufgenommen. Die Menge  $\mathcal{L}_{all}$  fasst alle großen Objektmengen zusammen und dient im dritten Schritt des Fuzzy-Apriori-Algorithmus zur Bestimmung der Fuzzy-Assoziationsregeln. Die Bestimmung des Supports sowie der Standardabweichung erfolgt analog zur Berechnung, die innerhalb der Methode GENERATE\_FUZZY\_1\_ITEMSETS vorgestellt wurde.

Zur Erstellung von Kandidatenmengen  $\mathcal{C}_k$  mit  $k$  Elementen basierend auf den großen Objektmengen  $\mathcal{L}_{k-1}$  wird die Untermethode APRIORI\_GEN verwendet. Anstatt für alle möglichen Objektmengen den Support direkt zu berechnen, wird durch APRIORI\_GEN auf Basis bereits

**Methode: FUZZY\_APRIORI\_GEN**

**Input:** Die große Fuzzy-Itemmenge  $\mathcal{L}_{k-1}$

**Output:** Die Kandidatenmenge  $\mathcal{C}_k$  als Obermenge der großen Fuzzy-Itemmenge  $\mathcal{L}_{k-1}$

- (1)  $\mathcal{C}_k = \emptyset$
- (2) Für alle Fuzzy-Kandidaten  $f_{\mathcal{C}_{k-1,a}}, f_{\mathcal{C}_{k-1,b}} \in \mathcal{L}_{k-1}$  mit  $f_{\mathcal{C}_{k-1,a}} \neq f_{\mathcal{C}_{k-1,b}}$  {
- (3) Wenn  $f_{\mathcal{C}_{k-1,a}}$  und  $f_{\mathcal{C}_{k-1,b}}$  haben  $k-2$  gleiche f-Items  $f_i$
- (4)  $f_{\mathcal{C}_{k-1,a}} = \{f_{i_1}, \dots, f_{i_{k-2}}, f_{i_{f_{\mathcal{C}_{k-1,a}}}}\}$
- (5)  $f_{\mathcal{C}_{k-1,b}} = \{f_{i_1}, \dots, f_{i_{k-2}}, f_{i_{f_{\mathcal{C}_{k-1,b}}}}\}$  {
- (6)  $\mathcal{C}_k = \mathcal{C}_k \cup \{f_{i_1}, \dots, f_{i_{k-2}}, f_{i_{f_{\mathcal{C}_{k-1,a}}}}, f_{i_{f_{\mathcal{C}_{k-1,b}}}}\}$
- (7) }
- (8) }
- (9) Für alle Fuzzy-Kandidaten  $f_{\mathcal{C}_k} \in \mathcal{C}_k$  {
- (10) Für alle Teilmengen  $s$  von  $f_{\mathcal{C}_k}$  mit  $(k-1)$  f-Items  $f_i$  {
- (11) Wenn  $s \notin \mathcal{L}_{k-1}$  {
- (12)  $\mathcal{C}_k = \mathcal{C}_k \setminus \{f_{\mathcal{C}_k}\}$
- (13) }
- (14) }
- (15) }
- (16) Liefere  $\mathcal{C}_k$  zurück

Abbildung 37: Die Methode FUZZY\_APRIORI\_GEN des Fuzzy-Apriori-Algorithmus

gefundenen großer Objektmenge eine Menge von Fuzzy-Kandidaten zur weiteren Überprüfung generiert. Dadurch können große Objektmenge mit einer immer größer werdenden Anzahl an Elementen  $k$  gefunden und somit auch komplexe Zusammenhänge erfasst werden. Dabei wird jede Menge der großen Objektmenge  $\mathcal{L}_{k-1}$  als Fuzzy-Kandidat  $f_{\mathcal{C}_{k-1}}$  bezeichnet. Alle Fuzzy-Kandidaten  $f_{\mathcal{C}_{k-1,a}}$  und  $f_{\mathcal{C}_{k-1,b}}$ , welche  $k - 2$  gleiche f-Items aufweisen, fließen in der Folge in die Kandidatenmenge  $\mathcal{C}_k$  (Zeilen 2-8 im Pseudocode). Der Schritt dient der Erstellung aller möglichen Itemkombinationen zu einer großen Menge und wird als Join-Schritt bezeichnet. In dem Join-Schritt folgenden Prune-Schritt (Zeilen 9-15 im Pseudocode) werden alle Itemkombinationen wieder aus der Kandidatenmenge  $\mathcal{C}_k$  ausgesondert, welche Teilmengen aufweisen, die keine großen Objektmenge darstellen. Dies liegt darin begründet, dass Objektmenge mit Teilmengen, die keine großen Objektmenge sind, niemals selbst große Objektmenge sein können. Ein Beweis hierfür liefert (Agrawal & Srikant, 1994). Als Ergebnis von APRIORI\_GEN erhalten wir mit  $\mathcal{C}_k$  alle Objektmenge mit  $k$  Elementen, welche möglicherweise große Objektmenge darstellen. Die Menge  $\mathcal{C}_k$  wird zur weiteren Bearbeitung an die Methode GENERATE\_FUZZY\_K\_ITEMSETS zurückgeliefert.

#### 4.7.3.3 Schritt 3: Generierung der Fuzzy-Assoziationsregeln (FAR)

In Schritt 2 des Fuzzy-Apriori-Algorithmus wurden alle großen Objektmenge mit  $k$  Elementen  $\mathcal{L}_k$  erstellt und in der Gesamtmenge großer Objektmenge  $\mathcal{L}_{all}$  zusammengefasst. Im vorliegenden dritten Schritt werden nun auf Basis von  $\mathcal{L}_{all}$  Fuzzy-Assoziationsregeln erstellt. Anders als im herkömmlichen Apriori-Algorithmus werden nur diejenigen Regeln zurückgeliefert, welche innerhalb der Grenzen gewisser Konfidenzintervalle  $\lambda$  liegen, welche die Zugehörigkeit zu einem linguistischen Term repräsentieren. Zu jedem linguistischen Term existiert daher ein Konfidenzintervall  $\lambda_i = [\lambda_{i,min}, \lambda_{i,max}]$ . Die Gesamtmenge aller Konfidenzintervalle  $\lambda_i$  wird als  $\lambda_{ges}$  bezeichnet. Im Vergleich zum Vorgehen unter Sicherheit wird im vorliegenden Fuzzy-Apriori-Algorithmus die Konfidenz unter Berücksichtigung der Sicherheitsgrade der einzelnen f-Items errechnet. Die grundsätzliche Berechnung der Konfidenz einer Regel als Quotient aus Support der Prämisse einer Regel ( $FAR_X.sup$ : Zeilen 13-21 im Pseudocode) und Support der Gesamtelemente einer Regel ( $FAR_{XY}.sup$ : Zeilen 4-12 im Pseudocode) bleibt erhalten. Die Prämisse einer Regel FAR setzt sich aus allen f-Items einer großen Itemmenge in  $\mathcal{L}_{all}$  sowie den zugehörigen linguistischen Term  $\lambda_i$  zusammen.

Somit ist das Ziel, die implizit in der Datenbank vorhandenen Zusammenhänge zwischen Kundenbedürfnissen und tatsächlichem Kaufverhalten explizit darzustellen, erreicht. Die über den Fuzzy-Apriori-Algorithmus generierten Assoziationsregeln spiegeln die Bedürfnisse und den vom Kunden gewünschten Nutzen wieder. Diese können somit als Anforderungen in die Entwicklungsprozesse des Anbieters rückgeführt oder zur Adaption der Fuzzy-Regelbasis genutzt werden.

**Methode: GENERATE\_FAR**

**Input:** Die große Fuzzy-Itemmenge  $\mathcal{L}_{all}$ , eine Datenbank  $\mathcal{D}$ , eine Menge vordefinierter Fuzzy-Konfidenzintervalle  $\lambda_{ges}$

**Output:** Eine Menge an Fuzzy-Assoziationsregeln  $FAR$  zu Merkmals-Item  $c_{recent}$

- (1) Für alle Fuzzy-Kandidaten  $fc_k \in \mathcal{L}_{all}$  {
- (2)     Erstelle Fuzzy-Assoziationsregel  $FAR_{recent}: fi_1 \cap \dots \cap fi_k \rightarrow c_i$
- (3)      $count = 0$
- (4)     Selektiere diejenigen Transaktionen  $\mathcal{T}$  aus  $\mathcal{D}_{preprocess}$ , in denen jedes f-Item  $fi$  des Fuzzy-Kandidaten  $fc_k$  einen Sicherheitsgrad  $\mu_{\mathcal{L}\mathcal{T}_{ij}}(x_{p_i})$  größer als Null aufweist, als  $\mathcal{D}_{preprocess}^T$
- (5)     Für jedes f-Item  $fi$  im Fuzzy-Kandidaten  $fc_k$  {
- (6)          $count = 0$
- (7)         Für jede Transaktion  $\mathcal{T}$  in  $\mathcal{D}_{preprocess}^T$  {
- (8)              $count = count + \mu_{\mathcal{L}\mathcal{T}_{ij}}(x_{p_i})$
- (9)             }
- (10)          $fi.sup = count/|\mathcal{D}|$
- (11)          $FAR_{XY}.sup = \min\{FAR_{XY}.sup, fi.sup\}$
- (12)     }
- (13)     Selektiere diejenigen Transaktionen  $\mathcal{T}$  aus  $\mathcal{D}$ , in denen jedes f-Item  $fi$  des Fuzzy-Kandidaten  $fc_k$  einen Sicherheitsgrad  $\mu_{\mathcal{L}\mathcal{T}_{ij}}(x_{p_i})$  größer als Null aufweist, als  $\mathcal{D}^T$
- (14)     Für jedes f-Item  $fi$  im Fuzzy-Kandidaten  $fc_k$  {
- (15)          $count = 0$
- (16)         Für jede Transaktion  $\mathcal{T}$  in  $\mathcal{D}^T$  {
- (17)              $count = count + \mu_{\mathcal{L}\mathcal{T}_{ij}}(x_{p_i})$
- (18)             }
- (19)          $fi.sup = count/|\mathcal{D}|$
- (20)          $FAR_X.sup = \min\{FAR_X.sup, fi.sup\}$
- (21)     }
- (22)      $FAR_{recent}.conf = (FAR_X.sup/FAR_{XY}.sup)$
- (23)     Für alle  $\lambda_i \in \lambda_{ges}$  {
- (24)         Falls  $((FAR_{recent}.conf > \lambda_{i,min}) \cap (FAR_{recent}.conf > \lambda_{i,max}))$  {
- (25)              $FAR = FAR \cup \{FAR_{recent}, \lambda_i\}$
- (26)         }
- (27)     }
- (28) }
- (29) Liefere  $FAR$  zurück

**Abbildung 38: Methode GENERATE\_FAR des Fuzzy-Apriori-Algorithmus**

#### 4.7.4 Zusammenfassung

Schritt vier der Methodik ‚Design for Customer‘ hat zum Ziel, das über den DfC Lösungskonfigurator gesammelte Kundenwissen zu analysieren und in Kundenanforderungen zu überführen oder zur Adaption der Fuzzy-Regelbasis zu nutzen. Hierzu sind die in der Kundenwissensdatenbank implizit vorhandenen Kundenbedürfnisse explizit zu machen und als Kundenanforderungen darzustellen. Aus Ermangelung bestehender Algorithmen wurde ein Fuzzy-Apriori-Algorithmus entwickelt, der auf dem herkömmlichen Apriori-Algorithmus zur Generierung von Assoziationsregeln unter Sicherheit aufbaut. Im Vergleich zu diesem erlaubt es der Fuzzy-Apriori-Algorithmus, Zusammenhänge auf Basis eines unscharfen Datenbestandes darzustellen. Ergebnis sind Fuzzy-Assoziationsregeln, die als Kundenanforderungen in die Entwicklungsprozesse des Lösungsanbieters fließen. Das Aufzeigen abgeleiteter Handlungsoptionen basierend auf den identifizierten Kundenanforderungen ist Aufgabe des fünften Schrittes der Methodik ‚Design for Customer‘.

## 4.8 DfC Anforderungsintegration

*Ziel der DfC Anforderungsintegration als fünftem Schritt der Methodik ‚Design for Customer‘ ist die durchgängige Einarbeitung der zuvor identifizierten Kundenanforderungen in die Entwicklungsprozesse des Lösungsanbieters. Basis der Integration ist das Schalenmodell des PSSE nach Weber et al., in das sich die Methodik ‚Design for Customer‘ nahtlos einbindet. Differenzen zwischen durch den Fuzzy-Apriori-Algorithmus identifizierten Assoziationsregeln und der Regelbasis des DfC Lösungskonfigurators lösen Synthese- und Analyseschritte des PSSE Schalenmodells aus. Somit können wandelnde Markttrends, als auch nicht mit dem bestehenden Lösungsportfolio abgedeckte Kundenbedürfnisse als Anforderungen in die Entwicklungsprozesse fließen. Zudem wird die Lernfähigkeit des Fuzzy-Systems sichergestellt.*

Eingabeparameter der Integration von Kundenanforderungen sind die mithilfe des Fuzzy-Apriori-Algorithmus generierten Fuzzy-Assoziationsregeln, welche die Anforderungen der Kunden darstellen. Die Kundenanforderungen leiten sich wie im vorangegangenen Schritt erläutert aus vier Anwendungsfällen ab und können in verschiedene Handlungsoptionen beim Lösungsanbieter münden. Die relevanten Handlungsoptionen sind bereits in Tabelle 11 ersichtlich, sollen der Vollständigkeit halber hier nochmals aufgeführt werden:

- *Fehlerhaftes Mapping der Fuzzy-Regelbasis:* Die Fuzzy-Regelbasis enthält ein fehlerhaftes Mapping zwischen Eigenschaften und Merkmalen. Der DfC Lösungskonfigurator schlägt dem Kunden aufgrund dessen nicht zu dessen Bedürfnissen passende Merkmale vor. Fachkompetente Kunden erkennen den Fehler und erwerben die zu den Eigenschaften passenden Merkmale.
- *Sich wandelnder Markttrend:* Wie im obigen Fall enthält die Fuzzy-Regelbasis ein fehlerhaftes Mapping zwischen Eigenschaften und Merkmalen. Fachkompetente Kunden erkennen den Fehler und erwerben die zu den Eigenschaften passenden Merkmale. In diesem Fall ist das fehlerhafte Mapping auf einen wandelnden Markttrend zurückzuführen, den das Unternehmen noch nicht erkannt hat. Die Unterscheidung zwischen den beiden Fällen kann nicht automatisiert erkannt werden, sondern ist von Experten zu beurteilen.
- *Fehlender Markttrend im Lösungsportfolio:* Der häufige Abbruch des Konfigurationsvorgangs bei der Eingabe ähnlicher Eigenschaften durch mehrere Kunden deutet darauf hin, dass die Bedürfnisse des Kunden durch das vorhandene Lösungsportfolio nicht abgedeckt werden können. Demnach ist ein Markttrend im Lösungsportfolio nicht vorhanden.

### 4.8.1 Vorgehensweise

Nicht jede über den Fuzzy-Apriori-Algorithmus generierte Assoziationsregel muss notwendigerweise zu einem der erläuterten Handlungsoptionen führen. Vielmehr ergibt sich der Handlungsbedarf erst dadurch, dass ein Delta zwischen der im Lösungskonfigurator vorhandenen Fuzzy-Regelbasis und den Fuzzy-Assoziationsregeln als Ausgangsgröße des Fuzzy-Apriori-Algorithmus entsteht. Das Schalenmodell des PSSE nach Weber et al. (Steinbach, Botta, & Weber, 2005) wurde entsprechend angepasst, um diesen Umstand anschaulich illustrieren zu können. Die Integration der Methodik ‚Design for Customer‘ in das Schalenmodell des PSSE ist in Abbildung 39 ersichtlich und soll im Folgenden im Detail erläutert werden.

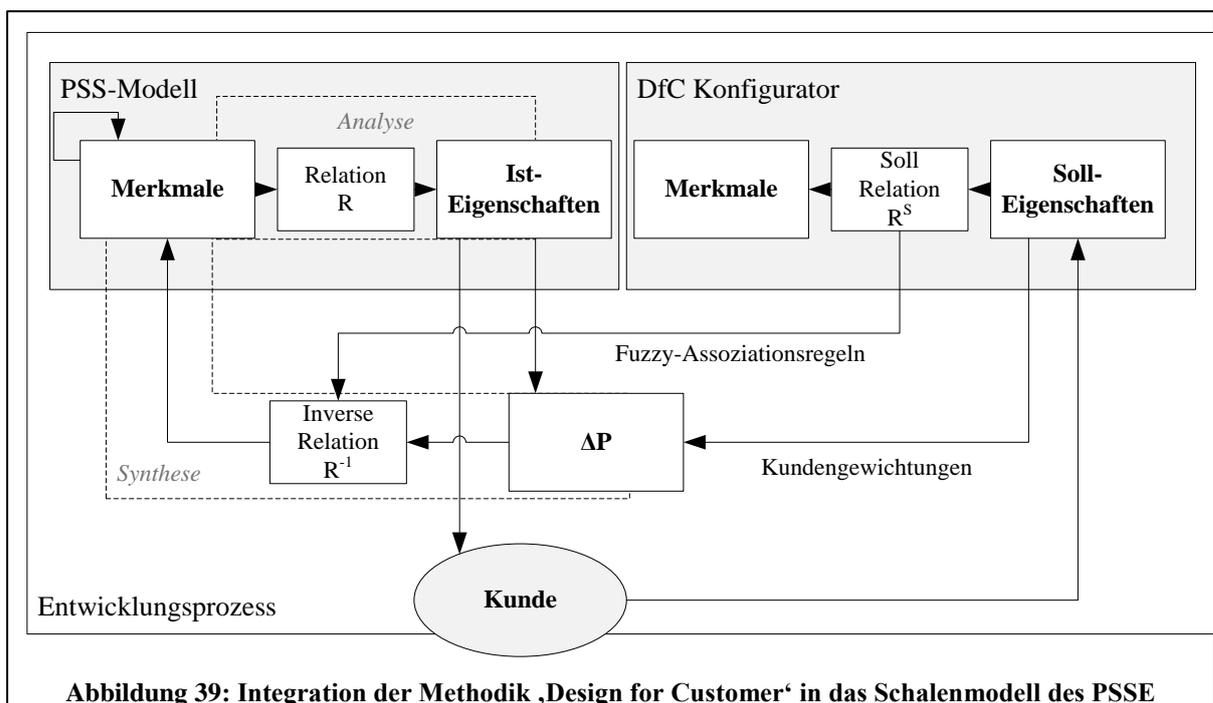


Abbildung 39: Integration der Methodik ‚Design for Customer‘ in das Schalenmodell des PSSE

Ausgangspunkt des ursprünglichen PSSE Modells nach Weber et al. sind die Anforderungen des Kunden, die als Soll-Eigenschaften in das Schalenmodell aufgenommen werden.  $\Delta P$  stellt nach Weber zu jedem Zeitpunkt des Entwicklungsprozesses die aktuelle Differenz zwischen den vom Kunden geforderten Soll-Eigenschaften und den tatsächlich in der Kundenlösung vorhandenen Ist-Eigenschaften dar. Über Syntheseschritte werden einzelne Soll-Eigenschaften dadurch realisiert, dass Merkmale innerhalb der Kundenlösung angepasst oder gänzlich neu entwickelt werden. Durch die Definition neuer oder angepasster Merkmale resultieren Auswirkungen auf die in der Kundenlösung vorhandenen Ist-Eigenschaften, die innerhalb eines Analyseschrittes an die veränderten Merkmale angepasst werden. Die Verknüpfung von Eigenschaften und Merkmalen erfolgt über eine Verknüpfungsrelation, die je nach Richtung Relation  $R$  oder Inverse Relation  $R^{-1}$  genannt wird.

Die Methodik ‚Design for Customer‘ bedient sich wie in den Kapiteln 4.3 und 4.4 gezeigt einer Fuzzy-Regelbasis zur Abbildung der beiden Relationen. Grundlage des Lösungskonfigurators bildet Relation  $R$ , welche die Konfiguration basierend auf Kundennutzen und Kundenbedürfnissen anstelle von Produktmerkmalen erlaubt. Für die strukturierte Generierung der Relationen bietet das Vorgehensmodell DfC Unified Process (siehe Kapitel 4.5) praktische Handlungsempfehlungen. Während der Nutzung des Lösungskonfigurators artikulieren Kunden ihre Bedürfnisse und Anforderungen an den Konfigurator, indem sie die für sie relevanten Eigenschaften gewichten und treffende Merkmale auswählen. Basierend auf diesem Kaufverhalten kann eine Soll-Relation  $R^S$  erstellt werden, welche die für den Kunden idealtypische Relation darstellt. Die Erstellung erfolgt mithilfe des Fuzzy-Apriori-Algorithmus zur Generierung von Assoziationsregeln in unscharfen Datenbeständen. Damit stellt  $\Delta P$  nicht nur die aktuelle Differenz zwischen den vom Kunden geforderten Soll-Eigenschaften und den tatsächlich in der Kundenlösung vorhandenen Ist-Eigenschaften dar. Zusätzlich wird ein Abgleich von Relation  $R$  mit der auf Basis der Kundeneingaben entstandenen Relation  $R^S$  möglich. Das hieraus entstehende  $\Delta P$  stellt die Differenz der vom Kunden geforderten Soll-Relation und den tatsächlich in der Kundenlösung vorhandenen Ist-Relation dar. Über Syntheseschritte werden die Soll-Relationen den Handlungsoptionen zugewiesen und fließen als Anforderungen in den Entwicklungsprozess. Die Anforderungen decken dabei wie gefordert die zuvor identifizierten Anwendungsfelder Lernfähigkeit des Fuzzy-Systems sowie das Erkennen sich wandelnder oder fehlender Markttrends im Lösungsportfolio ab.

#### 4.8.2 Zusammenfassung

Die Anforderungsintegration ermöglicht die durchgängige Einarbeitung der zuvor identifizierten Kundenanforderungen in die Entwicklungsprozesse des Lösungsanbieters. Die Methodik ‚Design for Customer‘ wird hierfür in das Schalenmodell des PSSE eingebettet, welches eines der am meisten fortgeschrittenen Vorgehensmodelle zur Entwicklung von Kundenlösungen darstellt. Mithilfe des DfC Lösungskonfigurators erfasste und durch den Fuzzy-Apriori-Algorithmus analysierte Kundenanforderungen können somit in das Lösungsportfolio eingearbeitet werden. Darunter fallen wandelnde Markttrends als auch nicht mit dem bestehenden Lösungsportfolio abgedeckte Kundenbedürfnisse.

Mit der Integration der Kundenanforderungen ist das zweite Hauptziel der Methodik ‚Design for Customer‘ erreicht. Darin werden Ansätze des Anforderungsmanagements speziell für den Bereich der hybriden Wertschöpfung gefordert, die bestehende Interaktionspunkte mit dem Kunden berücksichtigen. Im Rahmen der vorliegenden Arbeit wurde dies durch die kontinuierliche Rückführung von Kundenanforderungen auf Basis des bedürfnisorientierten Lösungskonfigurators als kontinuierliche Bezugsquelle sichergestellt.

## 4.9 Nutzen der Methodik

Der Nutzen der vorgestellten Methodik ‚Design for Customer‘ lässt sich in eine industrielle sowie in eine wissenschaftliche Sichtweise untergliedern. Diese werden im Folgenden dargelegt und in Beziehung zu den in Kapitel 2.4 identifizierten Handlungsbedarfen gesetzt. Somit wird die Zielerreichung der vorliegenden Arbeit demonstriert.

Der industrielle Nutzen der vorliegenden Arbeit besteht darin, Unternehmen die Nutzenpotentiale des Wandels zum Lösungsanbieter in ihrer Gänze ausschöpfen zu lassen. Diese lassen sich in Anlehnung an die ausführliche Darstellung aus Kapitel 2.1.4 wie folgt zusammenfassen:

- Aufbau von Differenzierungspotentialen
- Steigerung der Gewinnmarge
- Dauerhafte Kundenbindung

Durch die stringente Ausrichtung am Kundennutzen erlaubt die vorgestellte Methodik ‚Design for Customer‘ primär die Schaffung einer dauerhaften Kundenbindung sowie den Aufbau von Differenzierungspotential im Vergleich zum Wettbewerb. Sekundär wird die Steigerung der Gewinnmarge ermöglicht, indem die Methodik die Entwicklung eines an Kundenbedürfnisse angepassten Leistungsportfolios erlaubt. Die Sicherstellung dieser Nutzenpotentiale im Rahmen der Methodik gewährleisten folgende für die Industrie relevanten Methoden und Werkzeuge:

*N1-1 Bereitstellung des Konzeptes eines Lösungskonfigurators, der sich am Kundennutzen ausrichtet und sich bestehender Interaktionspunkte mit dem Kunden bedient*

Der Einsatz von Systemen zur Konfiguration wird von der Hälfte der Teilnehmer einer im Rahmen des vorliegenden Forschungsvorhabens durchgeführten Studie (siehe Kapitel 2.3.3) als sehr niedrig bis niedrig eingestuft. Bestehende Systeme zur Konfiguration richten sich jedoch vielmehr am Produkt und weniger am Kundennutzen aus, weshalb neuartige Ansätze benötigt werden. Die Bereitstellung eines nutzenbasierten Konfigurationssystems ermöglicht den Aufbau von Differenzierungspotential im Vergleich zum Wettbewerb sowie die Steigerung der Kundenbindung.

Folgende Handlungsbedarfe werden befriedigt: U2-1, U2-2

*N1-2 Bereitstellung eines Vorgehensmodells, das eine strukturierte Entwicklung des Konfigurators erlaubt*

Die Methodik stellt eine Vorgehensweise bereit, welche Unternehmen die strukturierte Einführung, Wartung und Pflege des nutzenbasierten Konfigurators erlaubt. Das Vorgehensmodell basiert auf dem weitverbreiteten Unified Process aus dem Software

Engineering und nutzt die Unified Modeling Language (UML) zur Formalisierung. Durch die Nutzung bestehender und akzeptierter Methoden und Technologien werden Eintrittsbarrieren verhindert sowie die Lernkurve der beteiligten Personen verkürzt.

Folgende Handlungsbedarfe werden befriedigt: U1-3

##### *N1-3 Bereitstellung von Methodenbaukästen und Guidelines für die ergonomische Gestaltung von IT-Systemen*

Klare Übersichten für die ergonomische Gestaltung von Systemen sind nicht verfügbar. Die Methodik ‚Design for Customer‘ trägt diesem Umstand Rechnung und stellt Werkzeuge bereit, welche Usability-Verantwortliche in ihrer Auswahl unterstützen. Guidelines führen den Nutzer in einem ersten Schritt transparent durch einen projektspezifisch zu beantwortenden Fragekatalog. Die Methodenbaukästen eignen sich darauf aufbauend, um weiterführende Informationen zur Normen- und Methodenauswahl zu erhalten und diese weiter zu verfeinern.

Folgende Handlungsbedarfe werden befriedigt: U1-4

##### *N1-4 Algorithmus zur automatischen Generierung von Kundenanforderungen und allgemein zur Entdeckung von Abhängigkeiten in unscharfen Datenbeständen*

Die in Kapitel 2.3.3 vorgestellte Studie zeigt die Nutzung wenig strukturierter Methoden des Anforderungsmanagements in der Industrie, die mit hohem manuellem Aufwand verbunden sind. Mit der Bereitstellung des Fuzzy-Apriori-Algorithmus wird die Möglichkeit geschaffen, Abhängigkeiten in unscharfen Datenbeständen zu entdecken und Kundenanforderungen abzuleiten.

Folgende Handlungsbedarfe werden befriedigt: U2-3

Aus wissenschaftlicher Sichtweise lässt sich der Nutzen der vorliegenden Arbeit insbesondere durch die Entwicklung neuer sowie durch die Erweiterung bestehender Forschungsansätze darstellen. Diese beziehen sich vorwiegend auf die in Kapitel 1 und 3 analysierten Forschungsfelder. Im Folgenden soll im Detail herausgestellt werden, welche Fortschritte die Methodik ‚Design for Customer‘ aus wissenschaftlicher Sichtweise liefert.

##### *N2-1 Modellierung eines Fuzzy-Reglers zur Verknüpfung von Kundeneigenschaften und Anbietermerkmalen*

Fuzzy-Regler stellen auf Fuzzy-Logik basierende Gesamtsysteme zur Ableitung logischer Schlussfolgerungen dar. Mit dem DfC Fuzzy-Controller wurde ein solcher Fuzzy-Regler entwickelt. Dieser beruht auf der in Kapitel 3.1.3 vorgestellten Architektur und wurde an die Trennung von Kundeneigenschaften und Anbietermerkmalen angepasst.

Folgende Handlungsbedarfe werden befriedigt: U1-1

*N2-2 Erweiterung der Wissensrepräsentationssprache OWL 2 zur Darstellung von Fuzzy-Informationen in Anlehnung an (Bobbilo & Straccia, 2011)*

Bobbilo und Straccia verwenden Annotationen innerhalb der Wissensrepräsentationssprache OWL 2, um Unschärfe zwischen verschiedenen Konzepten einer Ontologie abzubilden. Im Rahmen der vorliegenden Arbeit wird OWL 2 basierend auf diesem Ansatz erweitert, um nicht nur Unschärfe darstellen, sondern auch logische Schlussfolgerungen aus der Unschärfe ableiten zu können. Somit wird mit der DfC Fuzzy-Ontologie ein auf semantischen Technologien beruhender Fuzzy-Regler geschaffen.

Folgende Handlungsbedarfe werden befriedigt: U1-2

*N2-3 IEEE-konforme Analyse und Bewertung bestehender Methoden des Ontology Engineering*

Auf Grundlage des IEEE 1074-1995 Standards für Lebenszyklen der Softwareentwicklung (IEEE, 1996) wurde eine Analyse bestehender Methoden des Ontology Engineering durchgeführt. Basierend auf der Analyse ist eine projektspezifische Bewertung und Auswahl der Methoden möglich.

Folgende Handlungsbedarfe werden befriedigt: U1-3

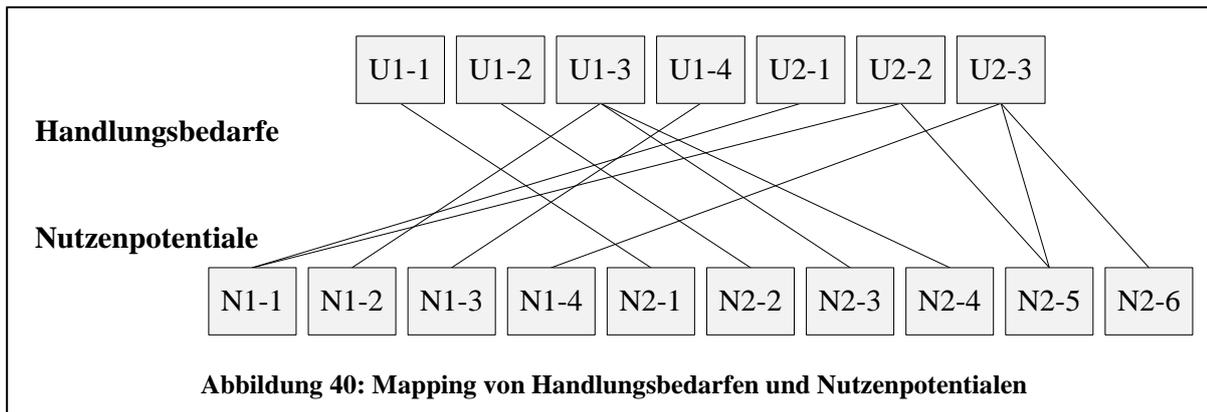
*N2-4 Ableitung einer Best-Practice Methode des Ontology Engineering zur strukturierten Entwicklung, Wartung und Pflege des Lösungskonfigurators*

Die Vergleichsanalyse bestehender Methoden des Ontology Engineering ermöglichte die Ableitung einer Best-Practice-Methode auf Basis von Konzepten bestehender Methoden des Ontology Engineering. Diese floss als DfC Unified Process in die Arbeit ein. Wie alle Methoden des Ontology Engineering hat sie die strukturierte Entwicklung, Wartung und Pflege von Ontologien zum Ziel.

Folgende Handlungsbedarfe werden befriedigt: U1-3

*N2-5 Entwicklung einer Methode zum kontinuierlichen und automatisierten Anforderungsmanagement*

Klassische Methoden des Anforderungsmanagements ermöglichen wie in Kapitel 2.2 gezeigt keine automatisierte Rückführung von Kundenanforderungen aus der Nutzungsphase von Leistungen. Durch das Zusammenspiel der methodischen Schritte der Methodik ‚Design for Customer‘ kann im Lösungskonfigurator erfasstes Kundenverhalten automatisiert, kontinuierlich in Kundenanforderungen überführt werden und in die Entwicklungsprozesse von Unternehmen einfließen. Kernkomponente zur Generierung von Kundenanforderungen stellt der im Rahmen der Arbeit entwickelte



Fuzzy-Apriori-Algorithmus dar.

Folgende Handlungsbedarfe werden befriedigt: U2-2, U2-3

#### N2-6 Entwicklung eines Verfahrens der Assoziationsanalyse bei unscharfen Datenbeständen

Die Analyse des Stands der Technik ergab das Fehlen eines vollständig implementierbaren Konzepts zur Assoziationsanalyse bei unscharfen Datenbeständen. Der zur Generierung von Kundenanforderungen entwickelte Fuzzy-Apriori-Algorithmus ermöglicht im Allgemeinen die Entdeckung von Abhängigkeiten in unscharfen Datenbeständen und schließt somit die Forschungslücke im Bereich des KDD.

Folgende Handlungsbedarfe werden befriedigt: U2-3

Wie bereits in der Beschreibung der einzelnen Nutzenpotentialen ersichtlich, nimmt jeder Nutzen Bezug auf einen in Kapitel 2.4 beschriebenen Handlungsbedarf. Abbildung 40 zeigt anschaulich auf, dass durch die Nutzenpotentialen sämtliche im Rahmen der vorliegenden Arbeit aufgestellten Handlungsbedarfe abgedeckt werden. Die Methodik ‚Design for Customer‘ befriedigt somit den geforderten Handlungsbedarf.

## 4.10 Zusammenfassung

Das vorliegende Kapitel 4 beschreibt die Methodik ‚Design for Customer‘ und bildet somit den Kern der Forschungsarbeit. Mit ihr wird ein Ansatz zur Kundenintegration für Lösungsanbieter bereitgestellt, der sich konsequent am Nutzen des Kunden orientiert und dessen Anforderungen und Wünsche kontinuierlich in die Entwicklungsprozesse rückführt. Basis der Methodik bildet ein bedürfnisorientierter Lösungskonfigurator, der auf der Trennung von Eigenschaften und Merkmalen beruht. Zur Befriedigung der in Kapitel 1 identifizierten Handlungsbedarfe wird die Methodik in 5 methodische Schritte gekapselt.

Schritt 1 dient der strukturierten Entwicklung, Wartung und Pflege des Lösungskonfigurators und stellt hierfür ein Vorgehensmodell bereit, das als Best-Practice-Modell aus bestehenden Ansätzen des Ontology Engineering abgeleitet wurde. Die Grundsätze des bedürfnisorientierten Lösungskonfigurators finden sich in Schritt 2. Darin werden zwei alternative Ansätze zum Aufbau des Lösungskonfigurators vorgestellt. Beide Ansätze verwenden ein Fuzzy-System zur Abbildung von Eigenschaften und Merkmalen. Während der in Abschnitt 4.3 vorgestellte DfC Fuzzy-Controller natürliche Unschärfe in menschlicher Sprache verarbeiten kann, ermöglicht die in Abschnitt 4.4 vorgestellte DfC Fuzzy-Ontologie zusätzlich die Identifikation impliziten Wissens in Kundenaussagen. Der Einsatz einer der beiden Ansätze ist projektspezifisch zu beantworten. Schritt 3 der Methodik stellt sicher, dass nicht nur die methodische Basis, sondern das Gesamtsystem entsprechend den Bedürfnissen des Benutzers gestaltet ist und liefert hierfür Guidelines und Methodenbaukästen des Usability Engineering. Das gewonnene Kundenwissen wird in Schritt 4 hinsichtlich immanenter Kundenanforderungen analysiert, wofür ein Algorithmus zur Fuzzy-Assoziationsanalyse entwickelt wurde. Die identifizierten Kundenanforderungen fließen in Schritt 5 in die Entwicklungsprozesse des Unternehmens ein. Nachfolgend kann wiederum Schritt 1 gestartet werden, um den Lösungskonfigurator an das geänderte Lösungsportfolio anzupassen, womit der Kreislauf geschlossen wird.

Abschließend wurde im vorliegenden Kapitel die Methodik hinsichtlich ihres Zielerfüllungsgrades bewertet. Grundlage hierfür dienen die in Kapitel 1 identifizierten Handlungsbedarfe. Es wurde festgestellt, dass die Methodik sämtlichen Handlungsbedarfen genügt.



## 5 Validierung der Methodik

Kapitel 5 dient der Validierung der im Rahmen des vorliegenden Forschungsvorhabens aufgestellten Methodik ‚Design for Customer‘. Zur Validierung der Methodik wurden zwei Anwendungsszenarien definiert, welche die Methodik in ihrer Gänze abdecken und anhand zweier Beispiele aus der Praxis verständlich aufbereiten.

### 5.1 Validierungskonzept

Die Methodik ‚Design for Customer‘ wird im vorliegenden Kapitel anhand zweier Anwendungsbeispiele hinsichtlich ihrer Praxistauglichkeit und Durchführbarkeit validiert. Die Anwendungsbeispiele fokussieren unterschiedliche Schritte der Methodik und werden durch jeweils einen eigenständigen Softwaredemonstrator unterstützt.

Innerhalb des ersten Anwendungsszenarios wird die Gesamtmethodik basierend auf dem Produkt- und Dienstleistungsportfolio von NAVIGON, einem Anbieter von Navigationslösungen, validiert. Das dargestellte Anwendungsszenario basiert auf öffentlich verfügbaren Informationen, ein Bezug zu einem realen Implementierungsprojekt besteht nicht. Navigationslösungen eignen sich durch ihren Variantenreichtum und die oftmals vorhandene Fachfremdheit der Kunden über einzelne Leistungsbestandteile als treffendes Anwendungsobjekt. Eine Einarbeitung in die Thematik der Navigationslösungen lohnt sich für die Kunden aufgrund des relativ niedrigen Preises vor dem Kauf in der Regel nicht, weshalb die Bereitstellung eines bedürfnisorientierten Konfigurators die Kaufentscheidung vereinfacht. Innerhalb des Anwendungsszenarios wird der DfC Unified Process angewandt, um einen Lösungskonfigurator basierend auf dem in Kapitel 4.3 hergeleiteten DfC Fuzzy-Controller zu entwickeln. Das erfasste Nutzerverhalten wird in der Folge in Kundenanforderungen überführt und auf Basis der DfC Kundenanforderungsgenerierung in die Entwicklungsprozesse des Unternehmens integriert.

Das zweite Anwendungsszenario ist angelehnt an das von der europäischen Kommission innerhalb des siebten Forschungsrahmenprogramms geförderte Projekt e-CUSTOM<sup>6</sup>, welches zum Ziel hat, den Kunden intensiver in die Prozesse von Herstellern zu integrieren und die Lücke zwischen Massenproduktion und Mass Customization zu schließen. Als Validierungsobjekt wurde die Produktfamilie Fiat 500 des Projektpartners Centro Ricerche Fiat

---

<sup>6</sup> A Web-based Collaboration System for Mass Customization, Grant Agreement No. FP7-2010-NMP-ICT-FoF 260067

gewählt. Ziel des zweiten Anwendungsszenarios ist die Validierung der Anwendbarkeit semantischer Technologien zur Erstellung eines Lösungskonfigurators basierend auf den in Kapitel 4.4 erläuterten Erweiterungen. Die Validierung innerhalb dieses Anwendungsszenarios beschränkt sich auf die Entwicklung der DfC Fuzzy-Ontologie sowie deren Realisierung auf Basis der syntaktischen Erweiterungen von OWL 2. Die praktische Anwendbarkeit der Fuzzy-Ontologie als Wissensbasis eines Konfigurators wird im Anschluss durch exemplarische Konfigurationseingaben und deren Auswertung mithilfe eines Fuzzy-Reasoners validiert.

### 5.2 Konsumgüter-Anwendungsszenario

Im ersten Anwendungsszenario wird die Gesamtmethodik auf Basis eines Anwendungsbeispiels aus dem Konsumgüterbereich überprüft. Datengrundlage bildet das Lösungsportfolio des Unternehmens NAVIGON. Zunächst erfolgt in den Abschnitten 5.2.1 und 5.2.2 eine Definition des Anwendungsszenarios sowie ein kurzer Abriss der prototypischen Realisierung durch einen Softwaredemonstrator. Danach wird in Abschnitt 5.2.3 die Methodik ‚Design for Customer‘ anhand des vorgestellten Anwendungsszenarios exemplarisch durchgeführt.

#### 5.2.1 Definition des Anwendungsszenarios

Im ersten Anwendungsszenario wurde als Validierungsobjekt das Produkt- und Dienstleistungsportfolio des Unternehmens NAVIGON, einem Anbieter von Navigationslösungen, gewählt. Das dargestellte Anwendungsszenario basiert auf öffentlich verfügbaren Informationen, ein Bezug zu einem realen Implementierungsprojekt besteht nicht. Das gewählte Anwendungsszenario validiert die Gesamtmethodik ‚Design for Customer‘ und deckt mit Ausnahme des DfC Usability Engineering alle methodischen Schritte ab.

Navigationslösungen sind IT-gestützte technische Systeme, welche mithilfe satellitengestützter Positionsbestimmung und Geoinformationen eine Zielführung zu einem gewählten Ort oder eine Route unter Beachtung gewünschter Kriterien ermöglichen. Navigationslösungen zeichnen sich wie zunehmend alle Konsumgegenstände durch deren Individualisierbarkeit und Variantenreichtum aus. Neben unterschiedlichen mechanischen Bestandteilen wie dem Gehäuse oder der Recheneinheit können vom Kunden unterschiedliche Softwaredienstleistungen gewählt werden, was die Navigationslösung für den Kunden zu einem komplexen Betrachtungsgegenstand macht.

Ziel des Anwendungsszenarios ist es, die Gesamtmethodik ‚Design for Customer‘ hinsichtlich ihrer industriellen Anwendbarkeit zu überprüfen und zu validieren. Hierbei wird zunächst der DfC Unified Process angewandt, um einen Lösungskonfigurator basierend auf dem DfC Fuzzy-

Controller strukturiert zu entwickeln. Das durch die Nutzung erfasste Kundenwissen wird in der Folge in Kundenanforderungen überführt, um die Entwicklung und Bereitstellung von an Kundenbedürfnisse angepassten Leistungen zu ermöglichen. Das Anwendungsszenario greift dabei auf die DfC Kundenanforderungsgenerierung und insbesondere auf den dort vorgestellten Fuzzy-Apriori-Algorithmus zurück.

### 5.2.2 Prototypische Realisierung

Die prototypische Realisierung des Softwaredemonstrators im Konsumgüterbereich umfasst einen Lösungskonfigurator auf Basis des DfC Fuzzy-Controllers sowie den Fuzzy-Apriori-Algorithmus, der die Ableitung von Kundenanforderungen basierend auf dem Nutzerverhalten im Konfigurator ermöglicht.

Das prototypisch realisierte Softwareframework ist nach dem Architekturmuster ‚Model View Controller‘ (MVC) aufgebaut, das zum Ziel hat, die Präsentation einer Software von dessen Logik und Daten zu trennen. Das Modell (Model) enthält die darzustellenden Daten und ist von Präsentation und Steuerungslogik unabhängig. Für die Darstellung der benötigten Daten aus dem Modell in einer Benutzeroberfläche und die Entgegennahme von Benutzerinteraktionen ist die Präsentationsschicht (View) zuständig. Die Steuerungsschicht (Controller) beinhaltet die eigentliche Anwendungslogik. Jede Einzelpräsentation benötigt dabei eine Steuerung, welche für die Weiterverarbeitung der vom Benutzer übergebenen Daten zuständig ist. Der Software-Prototyp basiert auf Microsoft-Technologien, insbesondere dem Microsoft .NET Framework 4.0 und wurde in der Entwicklungsumgebung Visual Studio 2010 entwickelt.

Auf Modellebene dient eine Microsoft SQL Server 2010 Enterprise Edition als zentrale Basis zur Verwaltung der anfallenden Daten. Die Datenbasis speichert das Nutzerverhalten im Lösungskonfigurator, verwaltet die notwendigen Merkmale und Eigenschaften sowie die Regelbasis des Konfigurators.

Die Steuerungsschicht beinhaltet die Anwendungslogik des Lösungskonfigurators und des Fuzzy-Apriori-Algorithmus. Die Steuerungsschicht wurde auf Basis des .NET Framework 4.0 von Microsoft erstellt. Als Programmiersprachen kamen VB.NET als auch C# zum Einsatz. Die Steuerungsschicht ist modular aufgebaut und erlaubt eine Wiederverwendbarkeit der einzelnen die Steuerung beschreibenden Klassen.

Die Benutzeroberfläche basiert auf den ASP.NET Web Forms, welche die Gestaltung dynamischer Weboberflächen gestattet. Um den Benutzern moderne Visualisierungseffekte wie Drag and Drop, 3D-Effekte und Animationen anbieten zu können, wurde zusätzlich Microsoft Silverlight (Version 5.1.10411.0) verwendet, das ebenfalls auf dem .NET Framework basiert. Abbildung 41 zeigt die grafische Benutzeroberfläche des Lösungskonfigurators.

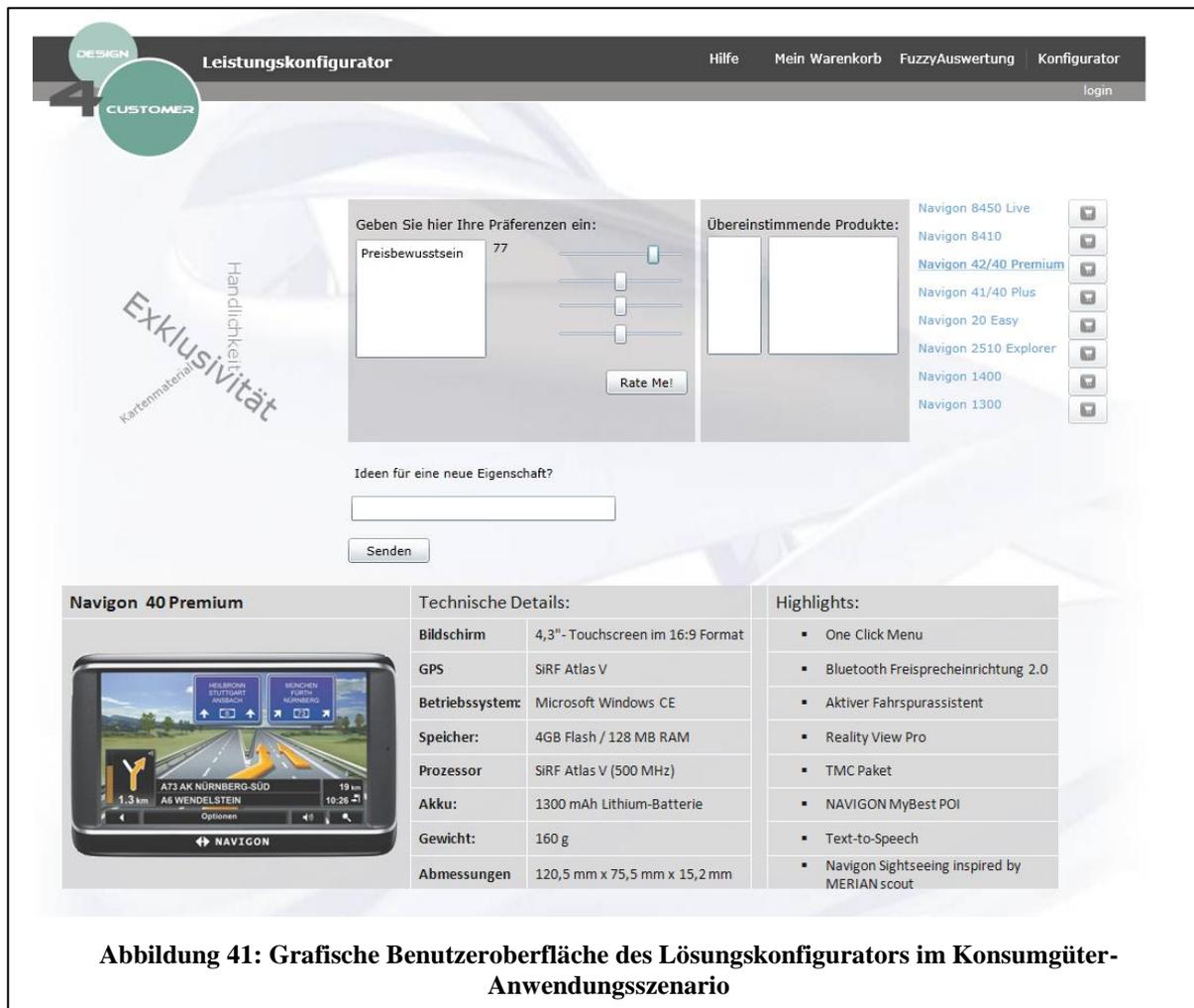


Abbildung 41: Grafische Benutzeroberfläche des Lösungskonfigurators im Konsumgüter-Anwendungsszenario

### 5.2.3 Anwendung der Methodik am Fallbeispiel

Ausgangspunkt des Anwendungsfalles ist der Wunsch des Unternehmens NAVIGON, die Bindung zu seinen Kunden zu steigern und dadurch die operative Gewinnmarge zu erhöhen. Das Unternehmen möchte daher die Einsetzbarkeit der Methodik ‚Design for Customer‘ prüfen.

#### 5.2.3.1 Nutzenanalyse und Entwicklung des DfC Lösungskonfigurators

In einem ersten Schritt wird der DfC Unified Process angewandt, um die grundsätzliche Machbarkeit der Methodik zu überprüfen und eine Nutzenanalyse durchzuführen. Fallen die Studien positiv aus, so ermöglichen die nachfolgenden Workflows des Vorgehensmodells die strukturierte Entwicklung und Bereitstellung der tatsächlichen Softwarelösung.

Zu Beginn des Prozesses findet ein Kick Off Event statt, innerhalb dessen ein Stab aus Vertretern der betroffenen Fachabteilungen gebildet wird. Unter die betroffenen Fachbereiche fallen der Vertrieb und das Marketing als Schnittstellen zu Markt und Kunde, die

Entwicklungsabteilungen des Unternehmens mit tiefgreifender Kenntnis des Leistungsportfolios sowie die IT-Abteilung, welche für die Umsetzung der finalen Softwarelösung zuständig ist. Die Aufgabe des Stabes ist die Überwachung des Projektverlaufes und die zentrale Sammlung der Teilergebnisse des Projektes. Im Zentrum des Prozessverlaufes stehen fünf Workflows, die operative Arbeitsanweisungen für die beteiligten Personen beinhalten.

**Anforderungsworkflow**

Als erste Handlung ordnet der Stab den Anstoß des Anforderungsworkflows an. Der Anforderungsworkflow legt mit der Definition der Szenarien und Geschäftsziele den Grundstein der zu entwickelnden Systemlösung. Der anvisierte Nutzen sowie damit verbundene Kosten lassen sich so bereits früh herausstellen und eine Planungssicherheit erzielen. Im vorliegenden Anwendungsfall treffen die relevanten Fachbereiche einen Konsens über das Anwendungsgebiet des zu definierenden Lösungskonfigurators und definieren den anvisierten Nutzen. Der Anforderungsworkflow wird dabei von den Fachabteilungen Marketing, Vertrieb und Entwicklung getrieben. Die IT-Abteilung hält im Anforderungsworkflow eine Beobachterrolle inne, um ein vereinfachtes Verständnis der definierten Anforderungen zu erlangen. Arbeitsergebnis ist die in Tabelle 12 ersichtliche Anforderungsspezifikation. Auf Basis der dort vorgenommenen Stärken-Schwächen-Analyse entscheidet sich der Stab für die Weiterverfolgung des Projektes. Somit ist der Analyseworkflow abgeschlossen.

**Anforderungsspezifikation des Konsumgüter-Anwendungsszenarios**

Bestimmung des Anwendungsgebietes und Gültigkeitsbereiches	Das Anwendungsgebiet umfasst die vom Unternehmen angebotenen Produktmodelle 8450 Live, 8410, 40 Premium, 40 Plus, Easy Europe 20, 2510 Explorer EU 41, 1400 und 1300. Diese stellen einen repräsentativen Querschnitt des verfügbaren Produktspektrums dar und decken sämtliche Kundengruppen ab. Neben den Kernprodukten werden sämtliche zu den gewählten Produkten zugehörige Dienstleistungspakete in das Anwendungsgebiet aufgenommen. Für das festgelegte Leistungsspektrum mitsamt dessen Merkmalsausprägungen wird ein am Kundennutzen ausgerichteter Lösungskonfigurator bereitgestellt, der die wesentlichen Nutzenkriterien der heterogenen Käuferschichten von modern exklusiv bis hin zu konservativ funktionsorientiert Rechnung trägt.
Definition des Geschäftszweckes	Der fokussierte Geschäftszweck lässt sich in eine externe sowie eine interne Sichtweise untergliedern. Die erste Kategorie dient der Nutzensteigerung beim Kunden (externer Faktor), die durch eine höhere Kundennähe erreicht wird. Folgende KPIs werden fokussiert: <ul style="list-style-type: none"> <li>• Transparenter Informationsabgleich von Nachrichten, die zwischen Kunde und Unternehmen ausgetauscht werden. Durch eine für den Kunden verständliche Präsentation des Leistungsportfolios erhöht sich die Transparenz des erreichbaren Kundennutzens.</li> </ul>

- Steigerung der Kundenzufriedenheit: Obiger Punkt trägt zeitgleich zur Steigerung der Kundenzufriedenheit bei.

Die zweite Kategorie fokussiert interne Geschäftszwecke:

- Erhöhung des Umsatzes: Bedingt durch Steigerung der Transparenz sowie Erhöhung der Kundenzufriedenheit und engere Kundenbindung wird der Umsatz um 5% gesteigert.
- Steigerung des Verständnisses über den Nutzen des eigenen Lösungsportfolios bzw. dessen einzelner Bestandteile: Die Rückführung der im Kundenverhalten erfassten Antworten tragen zur Verbesserung des eigenen Lösungsportfolios bei.

Definition der Anwendungsfälle

Im Rahmen des angestrebten Lösungskonfigurators wird ein zentraler Anwendungsfall angestrebt, welcher die initiale Konfiguration einer an individuelle Kundenbedürfnisse angepassten Navigationslösung umfasst. Der Konfigurator fokussiert insbesondere ein Kundenklientel, welches die Details des NAVIGON Produktprogramms nicht oder nur ungenügend kennt. Darüber hinaus dient er einer Käufergruppe mit vorhandenem Wissen im Anwendungsbereich zu einer Vorabkonfiguration, welche den Lösungsraum auf eine überschaubare Menge reduziert.

Beide Käufergruppen stellen durch die Artikulation ihrer kundeneigenen Bedürfnisse ihre Anforderungen an die zu konfigurierende Lösung und erhalten auf Basis des so übermittelten Anforderungskatalogs eine individuell spezifizierte Lösungskonfiguration. Die Transformation der vom Kunden geäußerten Eigenschaften in relevante Leistungsmerkmale ist unter Berücksichtigung natürlicher umgangssprachlicher Unschärfe sicherzustellen.

Die Ergebnispräsentation der Leistungsmerkmale soll den Kunden in dessen Entscheidung unterstützen und zu jedem Merkmal einen eindeutigen Sicherheitsgrad aufweisen, welcher die Relevanz des Leistungsmerkmals für die vom Kunden artikulierten Kundenanforderungen verdeutlicht. Somit wird der Kunde in dessen Entscheidungsprozessen unterstützt, ohne mögliche relevante Merkmale gänzlich auszublenden.

Identifikation von Kompetenzfragen

- CQ1: Welches sind die für die identifizierten Kundengruppen relevanten Bedürfnisse, welche an das definierte Leistungsportfolio gestellt werden?
- CQ2: Welche synonymen Begriffe werden von den unterschiedlichen Käufergruppen zur Artikulation ihrer Bedürfnisse verwendet?
- CQ3: Aus welchen Produkt- und Dienstleistungsmerkmalen setzt sich das Leistungsportfolio des Anwendungsgebietes zusammen?
- CQ4: Welcher Zusammenhang besteht zwischen den identifizierten Kundenanforderungen und Anbietermerkmalen?
- CQ5: Wie kann eine Verknüpfungsrelation in geeigneter Form mathematisiert werden?

Stärken-  
Schwächen-

Stärken:

- Kundengruppen mit geringer Kenntnis über das Leistungsportfolio

Analyse	<p>erhalten eine einfache und transparente Lösungskonfiguration.</p> <ul style="list-style-type: none"> <li>• Kundennutzen und Zusammenhang zum eigenen Lösungsportfolio werden für das anbietende Unternehmen aufgrund der Auseinandersetzung mit der Thematik transparent.</li> <li>• Die Bindung des Kunden wird aufgrund der Transparenz erhöht.</li> <li>• Der Umsatz steigert sich aufgrund engerer Kundenbindung und wahrgenommener Differenzierung vom Wettbewerb um 1%.</li> <li>• Durch Betonung des Kundennutzens erhöht sich das Differenzierungspotential im Vergleich zum Wettbewerb.</li> </ul> <p>Schwächen:</p> <ul style="list-style-type: none"> <li>• Die Implementierung und Wartung eines zusätzlichen Lösungskonfigurators verursacht jährliche Zusatzkosten in Höhe von € 50.000.</li> <li>• Fehlerhafte Konfigurationsregeln können zur Abwanderung von Kunden führen.</li> </ul>
---------	--

**Tabelle 12: NAVIGON DfC Unified Process Anforderungsspezifikation**

**Analyseworkflow**

Der Analyseworkflow schließt sich dem Anforderungswflow nach erfolgreich passierter Go-Entscheidung an und wird wiederum durch den Stab initiiert. Nach Rücksprache mit dem Stab werden die Verantwortlichkeiten der einzelnen Prozessschritte zwischen den Fachbereichen aufgeteilt. Zentrale Punkte des Analyseworkflows sind die Analyse der Produktmerkmale sowie die Identifikation der relevanten Kundeneigenschaften.

Für die initiale Erfassung der Kundeneigenschaften sind Vertrieb und Marketing zuständig, während Entwicklung und IT-Abteilung lediglich beratend tätig sind. Der gemeinsam abgeleitete Eigenschaftenglossar dient als Verständnisbasis für die Bedürfnisse und Anforderungen des Kunden. Ein Auszug aus dem Merkmalsglossar ist in Tabelle 13 ersichtlich.

<b>Eigenschaft</b>	<b>Beschreibung</b>
Exklusivität	Exklusivität bedeutet, ein System einem kleinen Personenkreis vorzubehalten, der anspruchsvolle Anforderungen an das System stellt und sich damit von der Masse abheben möchte.
Handlichkeit	Handlichkeit bezieht sich auf die Gebrauchstauglichkeit eines Systems. Insbesondere ist darunter eine meist kleine und einfache Bauweise zu verstehen, weshalb das System leicht zu verwenden ist.
Günstige Nutzung	Abhängig von der Nutzung des Leistungsbündels fallen laufende Kosten beziehungsweise Betriebskosten an. Darunter fallen beispielsweise Kartenupdates oder der Austausch von Verschleißteilen wie des Akkus.

Leistungsfähigkeit	Die Leistungsfähigkeit eines informationstechnischen Systems beschreibt die verrichtete Arbeitsmenge pro Zeiteinheit. Umgangssprachlich werden die Begriffe Geschwindigkeit und Schnelligkeit synonym verwendet.
Geländetauglichkeit	Als Geländetauglichkeit wird die Eigenschaft eines Systems bezeichnet, das auch abseits von Straßen sicher genutzt werden kann.

**Tabelle 13: Auszug aus dem NAVIGON Eigenschaftenglossar**

Die Entwicklungsabteilung erarbeitet im Analyseworkflow die konfigurationsrelevanten Leistungsmerkmale. Marketing, Vertrieb und IT haben eine beratende Funktion inne. Dies gilt insbesondere für die Erstellung des Merkmalsglossars. Dieser definiert die identifizierten Leistungsmerkmale eindeutig und schafft die Grundlage für ein gemeinsames Verständnis. Tabelle 14 gibt einen Auszug aus dem definierten Merkmalsglossar.

<b>Merkmal</b>	<b>Beschreibung</b>
Media Player	Media Player ist ein Gattungsbegriff für eine Gruppe von Softwaresystemen. Mit derartigen Programmen können Medieninhalte wie Video- und Audiodaten wiedergegeben werden.
Preis	Der Preis ist der in Geldeinheiten ausgedrückte Wert eines Gutes oder einer Dienstleistung. Der Begriff Preis ist objektiv und konkret.
Abmessung	Mit Abmessungen, umgangssprachlich Dimensionen, werden in der Regel die relevanten, kennzeichnenden physischen Längenmaße eines Gegenstandes benannt.
Gewicht	Als Gewicht wird umgangssprachlich die Masse eines Gegenstandes verstanden. Die Masse ist eine Eigenschaft der Materie und eine physikalische Grundgröße. Sie wird gemäß dem internationalen Einheitensystem in der Einheit Kilogramm angegeben.
Kartenservice	Kartenservices sind Dienstleistungspakete, welche die Art und Nutzungsdauer des mitgelieferten Kartenmaterials eines Leistungssystems beinhalten.
GPS-Boost (Instant Fix II)	Der GPS Boost ist ein Softwaresystem, das eine Beschleunigung der Satellitenortung ermöglicht und somit eine erhöhte Leistungsfähigkeit der Navigationslösung gewährleistet.
Koordinateneingabe	Die Koordinateneingabe ermöglicht die Suche eines Ortes mithilfe dessen geographischer Koordinaten. Mit den geographischen Koordinaten (geographische Breite und geographische Länge) lässt sich die Lage eines Punktes auf der Erde beschreiben.
Fußgänger- Navigation	Die Fußgänger-Navigation ermöglicht die Navigation zu einem Ort auf für Fußgänger geeigneten Wegen (Wanderwege, Bürgersteig,...).
Live Services	Live Services ermöglichen den Echtzeitzugang der Dienste Stauinformation, Radarstationen, lokales Wetter und Parkinformationen.
Echtglas-Oberfläche	Die Echtglas-Oberfläche ist eine Ausprägung des Bildschirms, reagiert präziser auf Benutzerinteraktion und ist kratzfester als herkömmliche Kunststoffbildschirme.

**Tabelle 14: Auszug aus dem NAVIGON Merkmalsglossar**

Zusätzlich zu den Merkmalen wurde ein Mapping erstellt, welches die Zuordnung der identifizierten Merkmale zu den Leistungen des Unternehmens erlaubt. Tabelle 15 zeigt die Verknüpfungen in tabellarischer Form auf. Das Symbol ● zeigt auf, dass eine Relation zutrifft, Symbol ○ zeigt auf, dass die Relation nicht zutrifft.

Leistung		8450 Live	8410	40 Premium	40 Plus	20 Easy Europe 20	2510 Explorer EU 41	1400	1300
Merkmal									
Preis									
>200€		●	●	○	○	○	○	○	○
<200€		○	○	●	●	●	●	●	●
Abmessung									
<100mm		○	○	○	○	●	●	●	●
>100mm		●	●	●	●	○	○	○	○
Gewicht									
< 150 gr		○	○	○	○	●	●	●	●
Kartenservice									
FreshMaps inklusive		●	●	○	●	○	○	○	○
FreshMaps 80% Rabatt		○	○	●	○	●	●	○	○
GPS-Boost (Instant Fix II)		●	●	○	○	○	○	○	●
Koordinateneingabe		●	●	●	●	○	○	○	○
Fußgänger-Navigation		●	●	●	●	●	●	●	○
Media Player		●	●	○	○	○	○	○	○
Live Services		●	○	○	○	○	○	○	○
Echtglas-Oberfläche		●	●	○	○	○	○	○	○

Tabelle 15: Mapping von Merkmalen und Leistungen

Zeitgleich zur Erstellung der beiden Glossare wird eine geeignete Verknüpfungsrelation zwischen Eigenschaften und Merkmalen bestimmt. Die hierfür verantwortliche IT-Abteilung identifiziert Stärken und Schwächen verschiedener Relationen im Hinblick auf deren spätere IT-basierte Umsetzung. Die Wahl fällt im vorliegenden Anwendungsfall auf eine Fuzzy-basierte Beziehung. Die Ausarbeitung der eigentlichen Regelbasis auf Basis der gewählten Verknüpfungsrelation wird von Vertrieb, Marketing und der Entwicklung geleistet, da diese das notwendige Wissen hinsichtlich Kundenbedürfnissen und zugehörigem Leistungsportfolio aufweisen. Tabelle 16 zeigt einen Auszug einfacher Regeln mit je einer Eigenschaft in der Regelprämisse und einem Merkmal in der Konklusion. Als linguistische Terme werden der Einfachheit halber lediglich die Terme ‚niedrig‘, ‚mittel‘ und ‚hoch‘ zugelassen.

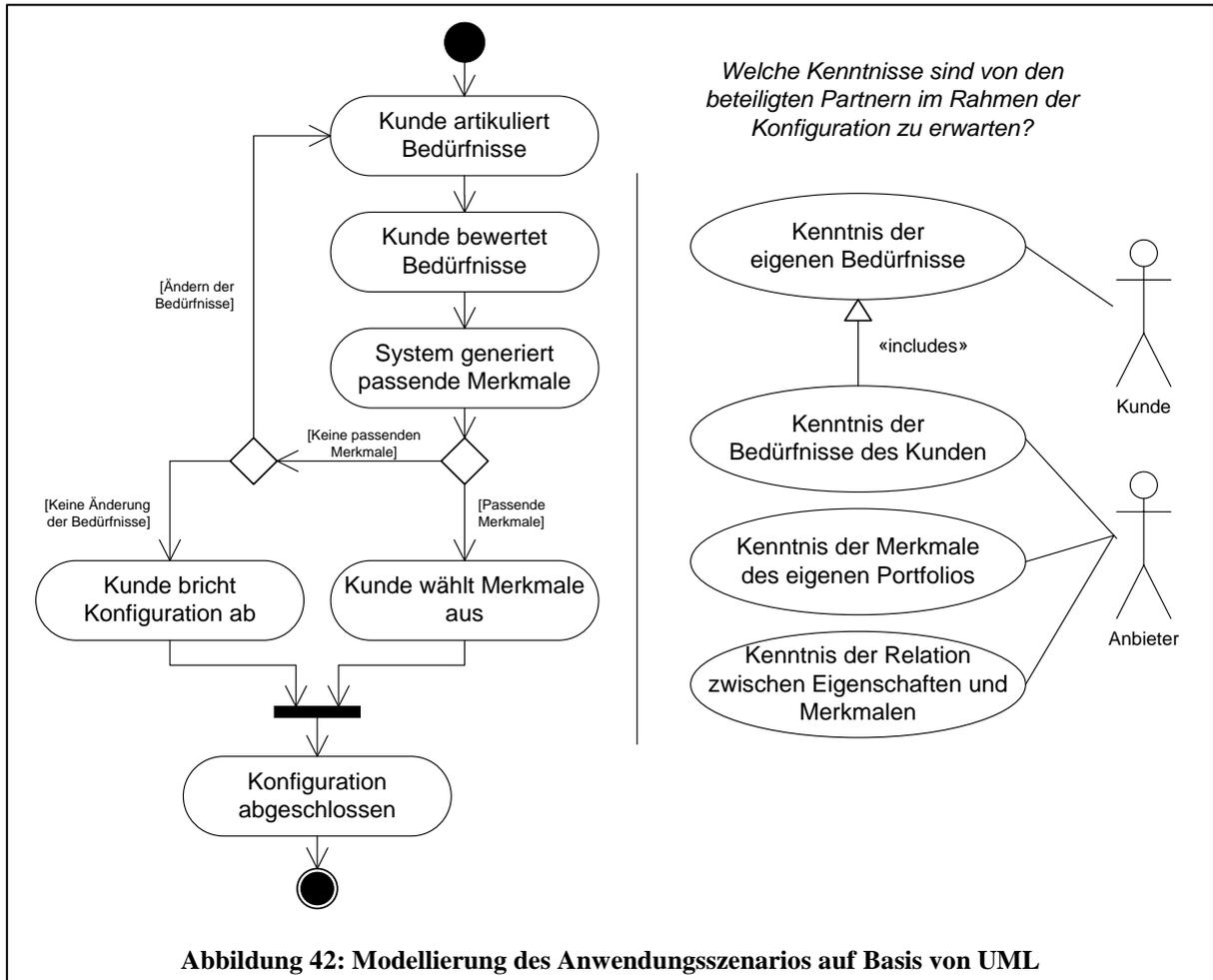
Regelbasis	
$R_1$	Wenn Exklusivität hoch, dann Media Player hoch.
$R_2$	Wenn Exklusivität hoch, dann Echtglas-Oberfläche hoch.
$R_3$	Wenn Handlichkeit hoch, dann Abmessungen <100mm hoch.
$R_4$	Wenn Handlichkeit hoch, dann Gewicht <150 gr hoch.
$R_5$	Wenn Handlichkeit niedrig, dann Abmessungen >100mm hoch.
$R_6$	Wenn Günstige Nutzung hoch, dann FreshMaps inklusive hoch.
$R_7$	Wenn Günstige Nutzung hoch, dann FreshMaps 80% Rabatt mittel.
$R_8$	Wenn Günstige Nutzung niedrig, dann FreshMaps inklusive niedrig.
$R_9$	Wenn Leistungsfähigkeit hoch, dann GPS-Boost (Instant Fix II) hoch.
$R_{10}$	Wenn Leistungsfähigkeit niedrig, dann GPS-Boost (Instant Fix II) niedrig.
$R_{11}$	Wenn Outdoor-Tauglichkeit hoch, dann Abmessungen <100mm hoch.
$R_{12}$	Wenn Outdoor-Tauglichkeit hoch, dann Koordinateneingabe hoch.
$R_{13}$	Wenn Outdoor-Tauglichkeit hoch, dann Fußgänger-Navigation hoch.

Tabelle 16: Auszug aus der NAVIGON Regelbasis

Für die linguistischen Terme sind zudem Zugehörigkeitsfunktionen zu bestimmen, welche die Kundenbedürfnisse mathematisieren und möglichst realistisch abbilden. Wiederum der Einfachheit halber werden einfache triangulare Funktionen verwendet, die wie folgt lauten:

$$\mu_{\text{niedrig}} = \begin{cases} 1, & x \leq 20 \\ 2 - 5x, & 20 < x \leq 40 \\ 0, & x > 40 \end{cases} \quad \mu_{\text{mittel}} = \begin{cases} 0, & x \leq 30, x > 70 \\ 5x - 1.5, & 30 < x \leq 50 \\ 3.5 - 5x, & 50 < x \leq 70 \end{cases}$$

$$\mu_{\text{hoch}} = \begin{cases} 0, & x \leq 60 \\ 5x - 3, & 60 < x \leq 80 \\ 1, & x > 80 \end{cases} \quad [5.1]$$



Mit dem Eigenschaftenglossar, dem Merkmalsglossar sowie der Regelbasis ist die Analysephase der Fachabteilungen Marketing, Vertrieb und Entwicklung abgeschlossen. Auf Grundlage der Arbeitsergebnisse beginnt die IT-Abteilung in der Folge mit der Formalisierung des Anwendungsszenarios. Die entstehenden UML Diagramme dienen als Grundlage der weiteren Diskussion sowie als Ausgangspunkt der nachfolgenden IT-gestützten Abbildung. Zur Schaffung einer allgemeingültig verständlichen Darstellung des Anwendungsszenarios eignen sich insbesondere UML Aktivitätsdiagramme (Abbildung 42 links) sowie UML Anwendungsfalldiagramme (Abbildung 42 rechts).

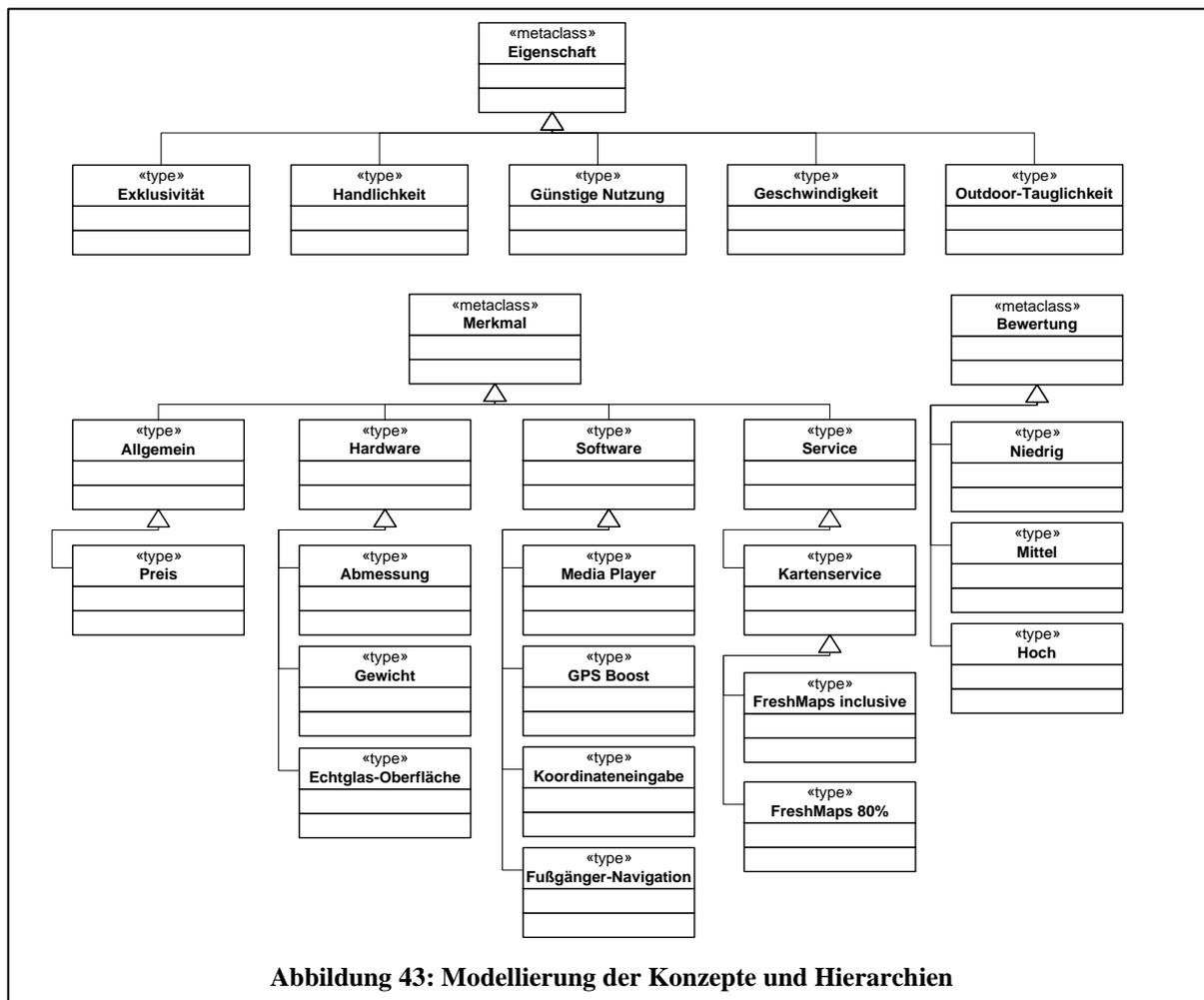


Abbildung 43: Modellierung der Konzepte und Hierarchien

### Designworkflow

Mit der Annahme der UML Diagramme durch die beteiligten Fachabteilungen und den Stab ist der Analyseworkflow abgeschlossen. Im darauffolgenden Designworkflow wird die IT-Abteilung mit der Spezifikation der Systemlösung betraut. Im Sinne der Kontinuität erfolgt diese mithilfe von UML Modellierwerkzeugen. Zur Modellierung der Konzepte sowie Darstellung der Hierarchien und domänenspezifischen Beziehungen eignen sich im Speziellen UML Klassendiagramme. Das Klassendiagramm des vorliegenden Anwendungsfalles ist in Abbildung 43 ersichtlich. Nach Fertigstellung werden die Arbeitsergebnisse einer Vollständigkeitskontrolle unterzogen. Dabei überprüfen alle am Projekt beteiligten Partner die umfangreiche Spezifikation der zuvor definierten Glossare.

Die innerhalb des Designworkflows erarbeiteten Ergebnisse werden zentral dokumentiert und bilden die Basis für die folgende systemgestützte Implementierung. Diese wird von der Wahl der formalen Sprache eingeleitet. Im vorliegenden Anwendungsfall wird ein herkömmliches Fuzzy-System ohne Zuhilfenahme semantischer Technologien implementiert. Die gewählten Technologien sowie weitere die Implementierung betreffende Informationen werden in Kapitel

5.2.2 Prototypische Realisierung detaillierter vorgestellt. Ist die Systemlösung vollständig implementiert, so kann der Testworkflow als letzter Ablauf des DfC Unified Process starten.

### **Testworkflow**

Der finale Testworkflow dient der Ergebnis- und Qualitätssicherung. Neben einer systemorientierten Prüfung hinsichtlich Konsistenz und Korrektheit durch die IT-Abteilung wird eine fachliche Prüfung durchgeführt. Diese stellt einerseits sicher, dass die Glossare vollständig in die Systemlösung übernommen wurden. Andererseits wird die Funktionstauglichkeit der Systemlösung dahingehend überprüft, ob die im Anforderungsworkflow aufgestellten Kompetenzfragen beantwortet werden können. Diese wurden a priori so definiert, dass mit ihrer Hilfe eine Validierung der finalen Systemlösung sichergestellt werden kann. Beispielhaft muss die Systemlösung Kompetenzfrage CQ3 „Aus welchen Produkt- und Dienstleistungsmerkmalen setzt sich das Leistungsportfolio des Anwendungsgebietes zusammen?“ beantworten können.

Mit dem erfolgreichen Abgleich von Anforderungen und implementierter Systemlösung ist die Entwicklung des Lösungskonfigurators abgeschlossen. Der implementierte Stab trifft sich weiterhin in regelmäßigen Abständen, um eine kontinuierliche Qualitätssicherung der Softwarelösung zu gewährleisten. Dabei müssen unter anderem Änderungen der Regelbasis, aber auch Änderungen der analysierten Glossare eingearbeitet werden. Diese werden insbesondere durch die Rückführung von Kundenverhalten auf Basis der Nutzung des Konfigurators geliefert.

### **5.2.3.2 Nutzung des Lösungskonfigurators**

Der zuvor entwickelte Lösungskonfigurator erlaubt eine Konfiguration basierend auf Kundennutzen und Kundenbedürfnissen anstelle von Produktmerkmalen. Grundlage hierfür bildet die Trennung in Kundeneigenschaften und Anbietermerkmale. Der Ablauf eines Konfigurationsvorgangs durch den Kunden wurde bereits in Kapitel 4.3.2 ‚Architektur des DfC Fuzzy-Controllers‘ exemplarisch anhand eines zum vorliegenden Anwendungsszenario passenden Beispiels dargestellt. Aus Gründen der Redundanz wird der Konfigurationsvorgang daher nicht nochmals explizit herausgearbeitet.

### 5.2.3.3 Rückführung von Kundenanforderungen

Das durch den Konfigurationsvorgang erfasste Kundenwissen dient im vorliegenden Anwendungsszenario als kontinuierliche Quelle für das Anforderungsmanagement. Dieser methodische Schritt der Methodik ‚Design for Customer‘ wird insbesondere durch den Fuzzy-Apriori-Algorithmus unterstützt und soll im Folgenden anhand des vorliegenden Anwendungsbeispiels erläutert werden.

Für die Rückführung von Kundenanforderungen aus den Nutzungsinformationen des Konfigurators wurde im Unternehmen ein Prozess installiert, der durch den Fuzzy-Apriori-Algorithmus gestützt wird. Der Algorithmus untersucht die Datenbank der Kundentransaktionen in regelmäßigen Zeitintervallen und generiert als Ergebnis Fuzzy-Assoziationsregeln. Diese werden mit der bestehenden Regelbasis des Konfigurators abgeglichen und daraus ein abgeleiteter Handlungsbedarf für die Entwicklungsprozesse generiert.

Im Folgenden wird die Funktionsweise des Fuzzy-Apriori-Algorithmus beispielhaft für das Produkt ‚NAVIGON 8450 Live‘ durchlaufen. Dieselbe Funktionsweise ergibt sich für jede weitere Leistung, die dem Algorithmus als Suchkriterium eingespeist wird.

Eingabeparameter des Algorithmus umfassen neben der Datenbank  $\mathcal{D}$  des Lösungskonfigurators einen vordefinierten Fuzzy-Mindestsupport  $\sigma_{\text{fuzzy-supp}}$ , eine vordefinierte Fuzzy-Maximalstandardabweichung  $\sigma_{\text{fuzzy-dev}}$  sowie zu jedem linguistischen Term vordefinierte Fuzzy-Konfidenzintervalle  $\lambda_{\text{ges}}$ .

Im vorliegenden Anwendungsfall werden folgende Werte vorgegeben:

- Fuzzy-Mindestsupport  $\sigma_{\text{fuzzy-supp}} = 0.2$
- Fuzzy-Maximalstandardabweichung  $\sigma_{\text{fuzzy-dev}} = 0.3$
- Fuzzy-Konfidenzintervalle  $\lambda_{\text{niedrig}} = [0.8, 1.5]$ ,  $\lambda_{\text{mittel}} = [1.5, 2.5]$ ,  $\lambda_{\text{hoch}} = [2.5, \infty]$

Die Zugehörigkeitsfunktionen zu den linguistischen Termen sind der Fuzzy-Regelbasis des Lösungskonfigurators entnommen und lauten wie folgt:

$$\begin{aligned} \mu_{\text{niedrig}} &= \begin{cases} 1, & x \leq 20 \\ 2 - 5x, & 20 < x \leq 40 \\ 0, & x > 40 \end{cases} \\ \mu_{\text{mittel}} &= \begin{cases} 0, & x \leq 30, x > 70 \\ 5x - 1.5, & 30 < x \leq 50 \\ 3.5 - 5x, & 50 < x \leq 70 \end{cases} \\ \mu_{\text{hoch}} &= \begin{cases} 0, & x \leq 60 \\ 5x - 3, & 60 < x \leq 80 \\ 1, & x > 80 \end{cases} \end{aligned} \quad [5.2]$$

Berechnungsgrundlage des Algorithmus liefert Datenbank  $\mathcal{D}$ , die sämtliches über den Lösungskonfigurator erfasste Nutzerverhalten für das zu untersuchende Leistungsmerkmal speichert. Die in der Datenbank abgebildeten Attribute umfassen einen fortlaufenden eindeutigen Identifikator der Transaktion  $\mathcal{T}$ , vom Kunden gewählte Eigenschaften  $p_i$ , die zugehörige Kundenbewertung  $x_{p_i}$ , sowie das final ausgewählte Leistungsmerkmal  $c_i$ . Die Kundenbewertung  $x_{p_i}$  zu einer Eigenschaft  $p_i$  ist im Intervall  $[0,100]$  abgebildet.  $x_{p_i} = 0$  drückt absolutes Desinteresse und eine Kundenbewertung  $x_{p_i} = 100$  absolutes Interesse gegenüber Kundeneigenschaft  $p_i$  aus. Die Kundenbewertung entstammt der Menge aller natürlicher Zahlen  $\mathbb{N}$  im angegebenen Intervall.

Tabelle 17 stellt eine exemplarische Datenbank  $\mathcal{D}$  dar, in der diejenigen Transaktionen aus der Gesamtdatenbank  $\mathcal{D}_{ges}$  selektiert wurden, bei denen Leistungsmerkmal ‚NAVIGON 8450 Live‘ käuflich erworben wurde. Jede Transaktion steht für eine vom Kunden getätigte Konfigurationstransaktion. Die Datenbank beinhaltet der Einfachheit halber lediglich zwei Kundeneigenschaften  $p_1$  und  $p_2$  mitsamt deren Bewertungen durch die Kunden.  $p_1$  stellt die Eigenschaft Exklusivität und  $p_2$  die Eigenschaft Leistungsfähigkeit dar. Fehlende Werte bei den Bewertungen der Kundeneigenschaften weisen darauf hin, dass der Kunde die Eigenschaften als irrelevant ansieht und daher nicht bewertet hat.

Transaktion $\mathcal{T}$	Exklusivität $p_1$	Leistungsfähigkeit $p_2$	8450 Live $c_i$
T1	80	50	1
T2	55	45	1
T3	50	70	1
T4	80		1
T5		50	1
T6	70	30	1
T7	75	50	1
T8	90		1
T9	50	40	1
T10	80		1
T11		80	1
T12	75	55	1

Tabelle 17: Aus dem Nutzerverhalten ermittelte Datenbank  $\mathcal{D}$

### Schritt 1: Aufbereitung der Datenbank

Im ersten Schritt wird Datenbank  $\mathcal{D}$  mit zusätzlichen Werten aufbereitet, um die Laufzeit des Algorithmus zu optimieren. Ergebnis ist eine neue Datenbank  $\mathcal{D}_{preprocess}$ , die neben den Informationen aus Datenbank  $\mathcal{D}$  um die Sicherheitsgrade der Zugehörigkeitsfunktionen angereichert wurde. Unter Verwendung der in oben vorgestellten Zugehörigkeitsfunktionen ergeben sich die in Tabelle 18 dargestellten Werte für Datenbank  $\mathcal{D}_{preprocess}$ .

Zum besseren Verständnis soll die Berechnung für Transaktion T8 beispielhaft erläutert werden. Der Kunde bewertet die Eigenschaft Exklusivität mit dem unscharfen Wert 90. Diese Kundeneingabe projizieren die Zugehörigkeitsfunktionen auf das Einheitsintervall [0,1]. Der linguistische Term ‚hoch‘ erhält somit den Sicherheitsgrad 1, die anderen beiden linguistischen Terme den Wert 0. Die Eigenschaft Leistungsfähigkeit wird vom Kunden in T8 nicht ausgewählt, daher erfolgt keine Projektion über die Zugehörigkeitsfunktionen, die Tabelleneinträge bleiben leer. Datenbank  $\mathcal{D}_{preprocess}$  dient in der Folge dem eigentlichen Algorithmus als Eingabe und Berechnungsgrundlage.

Transaktion $\mathcal{J}$	Exklusivität $p_1$			Leistungsfähigkeit $p_2$			8450 Live
	Niedrig	Mittel	Hoch	Niedrig	Mittel	Hoch	$c_i$
T1	0	0	1	0	1	0	1
T2	0	0,75	0	0	0,75	0	1
T3	0	1	0	0	0	0,5	1
T4	0	0	1				1
T5				0	1	0	1
T6	0	0	0,5	0,5	0	0	1
T7	0	0	0,75	0	1	0	1
T8	0	0	1				1
T9	0	1	0	0	0,5	0	1
T10	0	0	1				1
T11				0	0	1	1
T12	0	0	0,75	0	0,75	0	1

Tabelle 18: Aufbereitete Datenbank  $\mathcal{D}_{preprocess}$

## Schritt 2: Generierung großer Objektmengen

Im zweiten Schritt des Algorithmus werden große Objektmengen generiert, die innerhalb gewisser vordefinierter Grenzen liegen. Die Grenzen werden beim Fuzzy-Apriori-Algorithmus durch die Eingabeparameter Fuzzy-Mindestsupport  $\sigma_{fuzzy-supp}$  sowie Fuzzy-Standardabweichung  $\sigma_{fuzzy-dev}$  festgelegt. Der Algorithmus geht dabei iterativ vor d.h. es werden zunächst alle großen Objektmengen mit einem Element gesucht, dann mit zwei Elementen und so weiter. Ein Element entspricht dabei einem f-Item, das ein Paar aus Eigenschaft  $p_i$  und zur Eigenschaft gehörigem linguistischem Term darstellt.

Die Berechnung folgt der Methode GENERATE\_FUZZY\_1\_ITEMSETS (siehe Kapitel 4.7.3.2). Tabelle 19 zeigt das Ergebnis der Berechnung großer Objektmengen mit einem Element auf Basis der Berechnungsgrundlage von Datenbank  $\mathcal{D}_{preprocess}$  (siehe Tabelle 18). Darin ist neben den relevanten Werten Support und Standardabweichung auch die Hilfsvariable Count angegeben. Die innerhalb der geforderten Grenzen von  $\sigma_{fuzzy-supp}$  und  $\sigma_{fuzzy-dev}$  liegenden Werte für Support und Standardabweichung sind in der Tabelle kursiv hervorgehoben.

Beispielhaft sei die Berechnung für das f-Item (Exklusivität, mittel) gezeigt. Der Support errechnet sich über die Summe der Sicherheitsgrade für ein f-Item über alle Transaktionen in Datenbank  $\mathcal{D}_{preprocess}$  hinweg geteilt durch die Gesamtanzahl an Transaktionen in Datenbank  $\mathcal{D}$ . Die Summe der Sicherheitsgrade wird in der Hilfsvariable Count gespeichert. Diese ergibt sich aus den in Tabelle 18 verfügbaren Daten zu  $0,75 + 1 + 1 = 2,75$ . Geteilt durch die Anzahl an Transaktionen in Datenbank  $\mathcal{D}$  beläuft sich der Support für f-Item  $F_2 = (\text{Exklusivität}, \text{mittel})$  auf 0,229 und liegt damit innerhalb der Grenzen des vorgegebenen Mindestsupports  $\sigma_{fuzzy-supp}$ . Die Standardabweichung ergibt sich aus der Wurzel der Varianz der Stichprobe oder mathematisch ausgedrückt:

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x - \bar{x})^2} \quad [5.3]$$

Dabei steht Variable  $x$  für diejenigen Sicherheitsgrade des f-Items, die einen Wert größer Null aufweisen und  $\bar{x}$  für den Mittelwert der Variable  $x$ . Für das vorliegende f-Item  $F_2 = (\text{Exklusivität}, \text{mittel})$  ergibt sich  $\bar{x}$  zu  $\bar{x} = \frac{1}{3} * (0,75 + 1 + 1) = 0,917$  und die Standardabweichung  $\sigma$  zu 0,144. Somit liegen sowohl Support als auch Standardabweichung innerhalb der vorgegebenen Grenzen von  $\sigma_{fuzzy-supp}$  und  $\sigma_{fuzzy-dev}$ .  $F_2 = (\text{Exklusivität}, \text{mittel})$  ist große Objektmenge und wird in die Menge großer Objektmengen  $\mathcal{L}_1$  aufgenommen.

Nr.	f-Item		Count	Support	Standardabw.
$F_1$	Exklusivität	Niedrig	0,00	0,000	-
$F_2$	Exklusivität	Mittel	2,75	0,229	0,144
$F_3$	Exklusivität	Hoch	6,00	0,500	0,197
$F_4$	Leistungsfähigkeit	Niedrig	0,50	0,042	-
$F_5$	Leistungsfähigkeit	Mittel	5,00	0,417	0,204
$F_6$	Leistungsfähigkeit	Hoch	1,50	0,125	0,354

Tabelle 19: Berechnung von Support und Standardabweichung für  $\mathcal{L}_1$

Insgesamt ergibt sich  $\mathcal{L}_1$  nach Tabelle 19 zu:

$$\mathcal{L}_1 = \{(\text{Exklusivität}, \text{mittel}), (\text{Exklusivität}, \text{hoch}), (\text{Leistungsfähigkeit}, \text{mittel})\}$$

$\mathcal{L}_1$  wird der Menge aller großer Objektmengen  $\mathcal{L}_{all}$  hinzugefügt.

Damit ist die Suche großer Objektmengen mit einem Element abgeschlossen und es kann begonnen werden, große Objektmengen mit zwei Elementen zu generieren. Das im Folgenden beschriebene Vorgehen folgt der Beschreibung der Methode GENERATE\_FUZZY\_K\_ITEMSETS, die im Konzeptteil dieser Arbeit (Kapitel 4.7.3.2) verfügbar ist. Dabei müssen wie von Agrawal und Srikant (Agrawal & Srikant, 1994) gezeigt alle Teilmengen großer Objektmengen selbst große

Objektmengen darstellen. Es genügt daher, die Kandidatenmenge mit zwei Elementen  $\mathcal{C}_2$  auf der Menge großer Objektmengen  $\mathcal{L}_1$  aufzubauen. Die Kandidaten in  $\mathcal{C}_2$  ergeben sich als Permutation aller großen Objektmengen mit einem Element, die unterschiedliche linguistische Variablen aufweisen. Die Kandidatenmenge  $\mathcal{C}_2$  beinhaltet demnach die Kandidaten aller möglichen großen Objektmengen mit zwei Elementen. Analog geht die Methode FUZZY\_APRIORI\_GEN vor.  $\mathcal{C}_2$  enthält zwei Mengen und ergibt sich aus  $\mathcal{L}_1$  zu:

$$\mathcal{C}_2 = \left\{ \left\{ (Exklusivität, mittel), (Leistungsfähigkeit, mittel) \right\}, \left\{ (Exklusivität, hoch), (Leistungsfähigkeit, mittel) \right\} \right\}$$

Für jede der beiden Mengen in  $\mathcal{C}_2$  ist analog zu großen Objektmengen mit einem Element zu überprüfen, ob der Support und die Standardabweichung innerhalb der geforderten Grenzen liegen. Die Grenzen werden für jedes f-Item separat überprüft. Tabelle 20 gibt den Sachverhalt zur Berechnung großer Objektmengen mit zwei Elementen wieder.

Beispielhaft wird die Berechnung für den Kandidaten  $\mathcal{C}_{2,1} = \{(Exklusivität, mittel), (Leistungsfähigkeit, mittel)\}$  erläutert. Zunächst werden alle Transaktionen, in denen beide f-Items einen Sicherheitsgrad größer Null aufweisen, selektiert. Im vorliegenden Beispiel sind dies die Transaktionen T2 und T9. Die Summe der Sicherheitsgrade für f-Item  $(Exklusivität, mittel)$  ergibt  $0,75 + 1 = 1,75$  und wird durch die Gesamtanzahl an Transaktionen in  $\mathcal{D}$  geteilt. Die Berechnung der Standardabweichung folgt dem Vorgehen, das zur Berechnung der großen Objektmengen mit einem Element verwendet wurde. Für den Kandidaten  $\mathcal{C}_{2,1}$  liegen die Standardabweichungen beider Elemente in den geforderten Grenzen. Der Support beider Elemente ist jedoch geringer als der geforderte Mindestsupport  $\sigma_{fuzzy-supp} = 0.2$ . Der Kandidat  $\mathcal{C}_{2,1}$  ist daher keine große Objektmenge. Die Berechnung für  $\mathcal{C}_{2,2}$  ergibt hingegen die Zugehörigkeit als große Objektmenge mit zwei Elementen.

Die Menge großer Objektmengen für zwei Elemente besteht aus Kandidat  $\mathcal{C}_{2,2}$  und ergibt sich zu:

$$\mathcal{L}_2 = \left\{ \left( (Exklusivität, hoch), (Leistungsfähigkeit, mittel) \right) \right\}$$

$\mathcal{L}_2$  wird der Menge aller großen Objektmengen  $\mathcal{L}_{all}$  hinzugefügt.

Menge	f-Item	Count	Support	Standardabw.
$\mathcal{C}_{2,1}$	(Exklusivität, mittel)	1,75	0,146	0,177
	(Leistungsfähigkeit, mittel)	1,25	0,104	0,177
$\mathcal{C}_{2,2}$	(Exklusivität, hoch)	2,5	0,208	0,144
	(Leistungsfähigkeit, mittel)	2,75	0,229	0,144

Tabelle 20: Berechnung von Support und Standardabweichung für  $\mathcal{L}_2$

In der Folge ist die Kandidatenmenge mit drei Elementen  $\mathcal{C}_3$  zu bilden, um daraus die große Objektmenge  $\mathcal{L}_3$  abzuleiten. Die Kandidatenmenge mit drei Elementen  $\mathcal{C}_3$  entsteht aus der Permutation großer Objektmengen in  $\mathcal{L}_2$ . Da Objektmenge  $\mathcal{L}_2$  aus lediglich einer großen Menge besteht, ist  $\mathcal{C}_3 = \emptyset$ . Der Algorithmus kann damit keine weiteren großen Objektmengen finden und schließt Schritt 2 ab.

### Schritt 3: Generierung der Fuzzy-Assoziationsregeln

In Schritt drei werden auf Basis der Menge aller großer Objektmengen  $\mathcal{L}_{all}$  Fuzzy-Assoziationsregeln erstellt. Eine beispielhafte Regel kann wie folgt aussehen: „Wenn Exklusivität hoch, dann 8450 Live hoch.“

Ziel ist es, lediglich diejenigen Regeln zurückzuliefern, die eine gewisse Stärke und einen gewissen Vertrauensgrad aufweisen. Ähnlich zum herkömmlichen Apriori-Algorithmus werden daher nur diejenigen Regeln zurückgeliefert, welche innerhalb der Grenzen vorgegebener Konfidenzintervalle liegen. Die Konfidenz einer Regel errechnet sich aus dem Support der großen Objektmenge in Datenbasis  $\mathcal{D}$  geteilt durch den Support der großen Objektmenge in der Gesamtdatenbasis  $\mathcal{D}_{ges}$ . Die Prämisse der Regel basiert auf den großen Objektmengen und besteht aus  $n$  f-Items mit  $n \in \mathbb{N}$ , während sich die Konklusion aus einem Merkmals-Item  $c_i$  und zum Merkmals-Item gehörigen linguistischen Term  $\mathcal{L}\mathcal{T}_{i,j}$  zusammensetzt. Die Prämisse der Regel wird  $\mathcal{X}$ , die Konklusion  $\mathcal{Y}$  genannt.

Basis zur Identifikation der Fuzzy-Assoziationsregeln bilden die in  $\mathcal{L}_{all}$  identifizierten großen Objektmengen, die in Tabelle 21 aggregiert zusammengefasst sind.

ID	Große Objektmenge	Count(x)
$G_1$	{(Exklusivität, mittel)}	22,25
$G_2$	{(Exklusivität, hoch)}	17,5
$G_3$	{(Leistungsfähigkeit, mittel)}	15
$G_4$	{(Exklusivität, mittel) (Leistungsfähigkeit, mittel)}	13,75

Tabelle 21: Identifizierte große Objektmengen

Zur Berechnung der Fuzzy-Assoziationsregeln zum vorgegebenen Anwendungsfall sind einige zusätzliche Annahmen zu treffen, ohne die die Berechnung nicht durchgeführt werden kann. Diese werden im Folgenden vorgestellt.

Die Gesamtdatenbasis  $\mathcal{D}_{ges}$  besteht aus insgesamt 107 Kundentransaktionen. 12 der dort vorhandenen Transaktionen führen zum Kauf des Leistungsmerkmals NAVIGON 8450 Live und ergeben die bereits bekannte Teildatenbasis  $\mathcal{D}$ . Eine Gesamtdarstellung der Werte in Gesamtdatenbasis  $\mathcal{D}_{ges}$  ist aus Platzgründen nicht möglich. Notwendig für die Berechnung ist

lediglich die Kenntnis der Summe der Sicherheitsgrade für sämtliche großen Objektmengen in Gesamtdatenbasis  $\mathcal{D}_{ges}$ . Diese wird analog zu den vorherigen Schritten durchgeführt und ist in Tabelle 21 als Hilfsvariable  $Count(x)$  ersichtlich. Die Beschreibung der zusätzlichen Annahmen ist somit abgeschlossen und die Berechnung der Fuzzy-Assoziationsregeln kann erfolgen.

Beispielhaft wird die Berechnung von Fuzzy-Assoziationsregeln für die große Objektmenge  $G_2 = \{(Exklusivität, hoch)\}$  gezeigt. Die Berechnung folgt der Methode `GENERATE_FAR`. Hierfür ist zunächst der Support in Datenbasis  $\mathcal{D}$  zu bestimmen, der den Support der Gesamtregel – also  $support(\mathcal{X} \rightarrow \mathcal{Y})$  – ausdrückt. Darauffolgend wird der Support des Prämissenteils der Regel  $support(\mathcal{X})$  bestimmt. Letzterer wird in Gesamtdatenbasis  $\mathcal{D}_{ges}$  errechnet, da die Bedingung, dass Merkmal Y käuflich erworben wurde, nicht gilt.

Der Support der Gesamtregel  $support(\mathcal{X} \rightarrow \mathcal{Y})$  errechnet sich wie folgt: Die Summe der Sicherheitsgrade für ein f-Item über alle Transaktionen in Datenbank  $\mathcal{D}_{preprocess}$  hinweg geteilt durch die Gesamtanzahl an Transaktionen in Datenbank  $\mathcal{D}_{preprocess}$ . Für die große Objektmenge  $G_2 = \{(Exklusivität, hoch)\}$  und Merkmals-Item ‚NAVIGON 8450 Live‘ ergibt sich die Summe der Sicherheitsgrade zu  $1 + 1 + 0,5 + 0,75 + 1 + 1 + 0,75 = 6$ , der Support  $support((Exklusivität, hoch) \rightarrow NAVIGON\ 8450\ Live)$  ergibt sich zu 0,5.

Der Support des Prämissenteils der Regel  $support(\mathcal{X})$  errechnet sich analog zum  $support(\mathcal{X} \rightarrow \mathcal{Y})$ . Grundlage hierfür bilden jedoch die Werte in Datenbasis  $\mathcal{D}_{ges}$ . Die Summe der Sicherheitsgrade kann Tabelle 21 entnommen werden und hat den Wert 17,5. Geteilt durch die Gesamtanzahl an Transaktionen  $\mathcal{D}_{ges}$ , die mit 107 vorgegeben wurde, beläuft sich der Support  $support(Exklusivität, hoch)$  auf 0,164. Die Konfidenz ergibt sich aus dem Bruch der ermittelten Support-Werte zu

$$\lambda = \frac{support((Exklusivität, hoch) \rightarrow NAVIGON\ 8450\ Live)}{support(Exklusivität, hoch)} = \frac{0,5}{0,164} = 3,057$$

Die Ergebnisse der Berechnungen für sämtliche Assoziationsregeln ist in Tabelle 22 dargestellt. Neben den Berechnungen der einzelnen Supportwerte und der Endgröße Konfidenz sind auch die Hilfsvariablen  $count$  als Summe der Sicherheitsgrade aufgeführt.

ID	$count(\mathcal{X} \rightarrow \mathcal{Y})$	$support(\mathcal{X} \rightarrow \mathcal{Y})$	$count(\mathcal{X})$	$support(\mathcal{X})$	$conf(\mathcal{X} \rightarrow \mathcal{Y})$
$G_1$	2,75	0,229	22,25	0,208	1,102
$G_2$	6	0,500	17,5	0,164	3,057
$G_3$	5	0,417	15	0,140	2,972
$G_4$	2,5	0,208	13,75	0,129	1,621

**Tabelle 22: Berechnung der Konfidenz**

Die bislang aus den großen Objektmengen entstandenen Regeln haben beispielsweise die Form „Wenn Exklusivität hoch, dann NAVIGON 8450 Live“. Der Konklusion der Regel fehlt also ein linguistischer Term, um die Regel zu einer vollständigen Fuzzy-Assoziationsregel zu vervollständigen. Dies erfolgt über den Abgleich der errechneten Konfidenzwerte (siehe Tabelle 22) mit den vorgegebenen Konfidenzintervallen der linguistischen Terme, die als Eingangswerte dem Algorithmus zur Verfügung gestellt wurden. Liegen die errechneten Konfidenzen der erstellten Regeln innerhalb der vorgegebenen Grenzen einer der Konfidenzintervalle, so besitzt die Regel den geforderten Vertrauensgrad und sie wird als bestätigt zurückgeliefert. Zudem wird der Regel der linguistische Term, innerhalb dessen Intervallgrenzen die Regel liegt, hinzugefügt.

Im vorliegenden Beispiel wurden die Fuzzy-Konfidenzintervalle

- $\lambda_{niedrig} = [0.75, 1.5]$
- $\lambda_{mittel} = [1.5, 2.5]$
- $\lambda_{hoch} = [2.5, \infty]$

definiert. Die Regel  $G_2$  „Wenn Exklusivität hoch, dann NAVIGON 8450 Live“ besitzt die Konfidenz 3.057 und liegt damit innerhalb der Grenzen des Konfidenzintervalls  $\lambda_{hoch} = [2.5, \infty]$  zum linguistischen Term ‚hoch‘. Die vollständige Regel

„Wenn Exklusivität hoch, dann NAVIGON 8450 Live hoch.“

wird als Fuzzy-Assoziationsregel zurückgeliefert. Die Gesamtheit aller Fuzzy-Assoziationsregeln, die im vorliegenden Beispiel erstellt werden, ist in Tabelle 23 ersichtlich.

ID	Regel
$R_{1,}$	Wenn Exklusivität mittel, dann 8450 Live niedrig.
$R_{2,}$	Wenn Exklusivität hoch, dann 8450 Live hoch.
$R_{3,}$	Wenn Leistungsfähigkeit mittel, dann 8450 Live hoch.
$R_{4,}$	Wenn Exklusivität mittel und Leistungsfähigkeit mittel, dann 8450 Live mittel.

**Tabelle 23: Generierte Fuzzy-Assoziationsregeln**

Die bislang generierten Regeln geben den Zusammenhang zwischen Eigenschaften und Sachbeziehungswise Dienstleistungen an. Sollen die Eigenschaften in Verhältnis zu den identifizierten Merkmalen gesetzt werden, so erfolgt dies mit der aufgestellten Mappingtabelle zwischen Merkmalen und Leistungen (siehe Tabelle 15). Auf Basis der Mappingtabelle kann die Regel

„Wenn Leistungsfähigkeit mittel, dann 8450 Live hoch.“

übersetzt werden in:

„Wenn Leistungsfähigkeit mittel, dann GPS-Boost (Instant Fix II) hoch.“

Das Ziel, Fuzzy-Assoziationsregeln zwischen Eigenschaften und Merkmalen zu generieren, ist somit erreicht. Die Regeln bilden in der Folge Basis für die Generierung neuer oder angepasster Kundenanforderungen.

### Integration in das Anforderungsmanagement

Ziel der Anforderungsintegration ist die durchgängige Einarbeitung der zuvor identifizierten Kundenanforderungen in die Entwicklungsprozesse des Lösungsanbieters. Basis der Integration ist das Schalenmodell des PSSE nach Weber et al. (Weber, Steinbach, & Botta, 2004), in das sich die vom Unternehmen NAVIGON etablierte Methodik nahtlos einbindet. Differenzen zwischen durch den Fuzzy-Apriori-Algorithmus identifizierten Assoziationsregeln und der Regelbasis des Lösungskonfigurators lösen Synthese- und Analyseschritte des PSSE Schalenmodells aus. Somit können wandelnde Markttrends, als auch nicht mit dem bestehenden Lösungsportfolio abgedeckte Kundenbedürfnisse als Anforderungen in die Entwicklungsprozesse fließen. Zudem wird die Lernfähigkeit des Fuzzy-Systems sichergestellt.

Die im Lösungskonfigurator hinterlegte Regelbasis  $R$  wird dabei mit den im Fuzzy-Apriori-Algorithmus identifizierten Regeln  $R^S$  abgeglichen. Die Regelbasis  $R^S$  stellt die aus Sicht des Kunden wahrgenommene Soll-Relation des Leistungsportfolios und somit dessen Anforderungen dar. Die tatsächlich im Leistungsportfolio des Unternehmens vorhandene Relation wird von Relation  $R$  wiedergegeben. Die Differenz  $\Delta P$  zwischen beiden Relationen ermöglicht die Ableitung von Handlungsoptionen für das Anforderungsmanagement.

Folgende drei Beispiele aus dem vorliegenden Anwendungsfall sollen die Vorgehensweise verdeutlichen.

Vergleich	Ist-Relation $R$	Soll-Relation $R^S$
$V_1$	Wenn Exklusivität hoch, dann Echtglas-Oberfläche hoch.	Wenn Exklusivität hoch, dann Echtglas-Oberfläche hoch.
$V_2$	Wenn Leistungsfähigkeit hoch, dann GPS-Boost (Instant Fix II) hoch.	Wenn Leistungsfähigkeit mittel, dann GPS-Boost (Instant Fix II) hoch.
$V_3$	Wenn Leistungsfähigkeit hoch, dann GPS-Boost (Instant Fix II) hoch.	Wenn Leistungsfähigkeit hoch, dann Konfigurationsabbruch hoch.

Abbildung 44: Vergleich von Soll-Relation und Ist-Relation zur Generierung von Anforderungen

- In Vergleich  $V_1$  besteht kein Delta zwischen Ist-Relation  $R$  und Soll-Relation  $R^S$ . Vielmehr stützt die mithilfe des Fuzzy-Apriori-Algorithmus ermittelte Regel die im

Lösungskonfigurator vorhandene Relation. Die Regel weiß eine hohe Konfidenz von 3.057 auf, was für deren Stärke und somit einer hohen Sicherheit spricht.

- In Vergleich  $V_2$  wird ein wandelnder Markttrend beobachtet. Statt einer hoher Leistungsfähigkeit deckt das Leistungsmerkmal ‚GPS-Boost (Instant Fix II)‘ nur noch eine mittlere Leistungsfähigkeit ab. Die Konfidenz von 2.972 liegt nur unwesentlich unter dem Wert von Vergleich  $V_1$  und weißt somit ebenfalls einen hohen Sicherheitsgrad auf.
- Die Regel der Soll-Relation  $R^S$  aus Vergleich  $V_3$  gibt eine hohe Abbruchwahrscheinlichkeit im Konfigurator wieder. Diese gilt für den Fall, dass vom Kunden die Eigenschaft Leistungsfähigkeit als hoch bewertet wurde. Assoziationsregeln, die Abbrüche beinhalten, ermöglichen das Aufspüren von nicht im Leistungsportfolio enthaltenen Markttrends. Die Regel wurde nicht in obigem Anwendungsbeispiel hergeleitet, sondern in einem separaten Durchlauf des Fuzzy-Apriori-Algorithmus festgestellt. In obigem Beispiel ist das Merkmal ‚GPS-Boost (Instant Fix II)‘ nicht mehr in der Lage, eine hohe Leistungsfähigkeit zu garantieren (siehe auch Vergleich  $V_2$ ). Der dritte Vergleich untermauert diese Schlussfolgerung und zeigt auf, dass der Wunsch nach einer hohen Leistungsfähigkeit häufig zu einem Abbruch der Konfiguration führt.

Wie gezeigt ergeben die Vergleiche 2 und 3 ein Delta zwischen Ist-Regelbasis  $R$  und Soll-Regelbasis  $R^S$  und werden in der Folge als Kundenanforderungen in die Entwicklungsprozesse integriert. Die Integration ist als Syntheseschritt im Schalenmodell nach PSSE von Weber abgebildet und kann in der Literatur (Weber, Steinbach, & Botta, 2004), (Steinbach, Botta, & Weber, 2005) (Botta, 2007) nachvollzogen werden.

### 5.3 Automotive-Anwendungsszenario

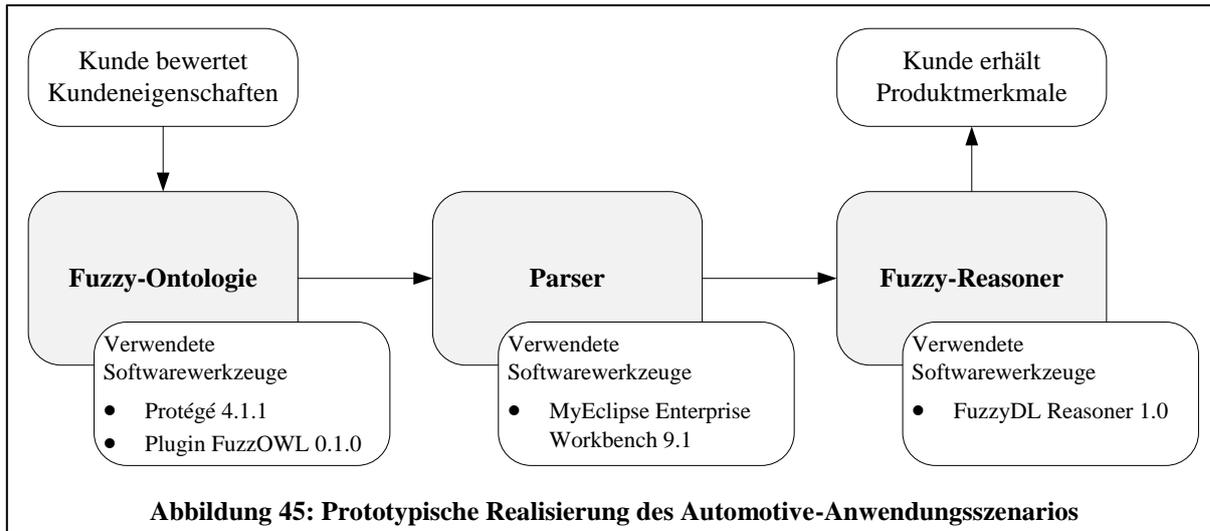
Das zweite Anwendungsszenario basiert auf dem Produktspektrum des Automobilherstellers Fiat und wurde im Rahmen des EU-Forschungsprojektes e-CUSTOM erstellt. Ziel des zweiten Anwendungsszenarios ist die Validierung der Anwendbarkeit semantischer Technologien zur Erstellung eines Lösungskonfigurators basierend auf den in Kapitel 4.4 ‚DfC Fuzzy-Ontologie‘ erläuterten Erweiterungen und Syntax. Wie auch im ersten Anwendungsszenario erfolgen zunächst eine Definition des Anwendungsszenarios und ein kurzer Abriss der IT-basierten prototypischen Realisierung (Abschnitte 5.3.1 und 5.3.2), bevor die Methodik in Abschnitt 5.3.3 anhand eines Fallbeispiels aus der Praxis verdeutlicht wird.

#### 5.3.1 Definition des Anwendungsszenarios

Als Validierungsobjekt innerhalb des ersten Anwendungsszenarios wurde die Produktmodellfamilie Fiat 500 des e-CUSTOM Projektpartners Centro Ricerche Fiat gewählt. Innerhalb des Szenarios soll ein Konfigurator erstellt werden, welcher es Kunden basierend auf der Eingabe von Eigenschaften erlaubt, die dazu passenden Produktmerkmale zu konfigurieren. Dabei sollen auch implizite Kundeneigenschaften identifiziert werden können.

Der Fiat 500 ist ein Kleinwagenmodell des italienischen Herstellers Fiat S.p.A. Die seit 2007 gefertigte aktuelle Baureihe des Fiat 500 wird anders als deren Vorgängermodelle nicht als Einstiegsmodell, sondern als Lifestyle-Produkt nach Vorbild des BMW Mini vertrieben. Wie in der Automobilbranche üblich, können die Modelle entsprechend den Wünschen des Kunden individualisiert und die Aussage des Fahrzeugs somit an den Charakter des Käufers angepasst werden. Die Anpassung des Fahrzeugs erfolgt dabei auf Basis eines Portfolios an Produktmerkmalen, welches sich für einen Kunden ohne Kenntnisse des Produktes intransparent und komplex darstellt. Im Falle des Fiat 500 stehen drei generelle Ausstattungspakete, eine Auswahl mehrerer Verbrennungsmotoren sowie sonstige Ausstattungen zur Verfügung. Die Wahl eines Ausstattungspaketes wirkt sich dabei erheblich auf die Verfügbarkeit der weiteren Produktmerkmale aus. Aufgrund des relativ hohen Kaufpreises wird ein Kunde im vorliegenden Anwendungsszenario eher bereit sein, sich in die Themenstellung einzuarbeiten als bei niedrigpreisigen Produkten oder Lösungen. Die Bereitstellung eines eigenschaftsbasierten Konfigurators ist im vorliegenden Anwendungsszenario daher als Vor-Konfigurator zu sehen, der den Entscheidungsraum des Kunden auf eine überschaubare Menge begrenzt.

Im Rahmen des vorliegenden Anwendungsszenarios wurde ein solcher Konfigurator erstellt und dessen Funktionalität basierend auf einer Kundeneingabe überprüft. Ziel des Anwendungsszenarios ist es, die Anwendbarkeit der DfC Fuzzy-Ontologie als Alternative zum DfC Fuzzy-Controller zu überprüfen und zu validieren.



### 5.3.2 Prototypische Realisierung

Die prototypische Realisierung des Softwaredemonstrators umfasst im vorliegenden Anwendungsszenario ein Konfigurationssystem auf Basis der im Konzept erläuterten syntaktischen Erweiterungen von OWL 2. Bei der Realisierung wurde auf die Entwicklung einer Anwenderoberfläche verzichtet. Die Beschreibung der prototypischen Realisierung legt stattdessen ihren Schwerpunkt auf die Entwicklung der DfC Fuzzy-Ontologie sowie deren Anbindung an einen bestehenden Fuzzy-Reasoner. Die Umsetzung umfasst demzufolge drei Module, welche sich in die DfC Fuzzy-Ontologie, den für die Zerlegung und Umwandlung in ein für die Weiterverarbeitung brauchbares Format zuständigen Parser sowie den Fuzzy-Reasoner untergliedert. Die DfC Fuzzy-Ontologie umfasst die Kundeneigenschaften und Produktmerkmale beschreibende Wissensbasis des Konfigurators mitsamt den sie verbindenden Regeln. Innerhalb der DfC Fuzzy-Ontologie werden zudem kundenspezifische Werte basierend auf einer Kundeneingabe gespeichert. Die Auswertung der Kundeneingabe hinsichtlich der Wissensbasis übernimmt ein Fuzzy-Reasoner, innerhalb dessen nicht nur explizit vorhandene Kundenaussagen, sondern auch implizit geäußertes Kundenwissen ausgewertet werden kann. Für die Überführung der Fuzzy-Ontologie in ein für den Fuzzy-Reasoner brauchbares Eingabeformat wurde ein Parsingalgorithmus entwickelt und zwischen DfC Fuzzy-Ontologie und Fuzzy-Reasoner geschaltet. Abbildung 45 veranschaulicht den Aufbau des Softwaredemonstrators.

Aus softwaretechnischer Sicht wurde bei der Umsetzung einerseits auf bereits bestehende Java-basierte Anwendungen zurückgegriffen und andererseits die Programmiersprache Java für Eigenentwicklungen verwendet, so dass der Demonstrator aufgrund der durchgängigen Verwendung von Java-Technologien plattform- sowie systemunabhängig ist.

Die Modellierung der Fuzzy-Ontologie erfolgte mithilfe der Open Source Anwendung Protégé, die gleichzeitig als Wissensdatenbank-System dient. Die Plattform wird gemeinschaftlich von der Universität Stanford sowie der Universität Manchester weiterentwickelt und erlaubt eine Erweiterung und individuelle Anpassung basierend auf einem Plugin-Konzept. Protégé ermöglicht die Entwicklung sowie den Export von Ontologien in einer Vielzahl an Formaten wie RDF(S), OWL und XML Schema. Da die im vorliegenden Anwendungsfall entwickelte Fuzzy-Ontologie auf syntaktischen Erweiterungen von OWL 2 basiert, sind die Entwicklung und der Export in einem anderen Format als OWL 2 jedoch nicht sinnvoll. Für die Entwicklung der Fuzzy-Ontologie im Rahmen des vorliegenden Anwendungsszenarios wurde die Version 4.1.1 (Build 231) verwendet, insbesondere kam das Protégé Plugin FuzzOWL in der Version 0.1.0 zum Einsatz. Das erwähnte Plugin ermöglicht es, Fuzzy-Annotationen zu OWL 2 in der grafischen Oberfläche von Protégé zu erstellen und zu verwalten. Hierbei sei erwähnt, dass die Entwicklung der Fuzzy-Ontologie auch innerhalb eines herkömmlichen Text-Editors erfolgen kann. Die Plattform Protégé bietet jedoch den Vorteil der intuitiveren Benutzerführung, sowie der grafischen Veranschaulichung der entwickelten Ontologien.

Als Fuzzy-Reasoner kommt die ebenfalls frei verfügbare Anwendung FuzzyDL zum Einsatz, welche eine auf Java basierende Lösung zur Ableitung von Schlussfolgerungen für die Beschreibungslogik *SHIF* darstellt. Dabei erweitert der Reasoner die Beschreibungslogik *SHIF* um fuzzy-logische Ausdrücke und ermöglicht somit die Ableitung von Schlussfolgerungen für unscharfe Informationen. Von den drei identifizierten Fuzzy-Basislogiken wird die Logik nach Łukasiewicz sowie die aus den Basislogiken ableitbare Zadeh-Logik unterstützt. Die Anwendung ist frei in der Version 1.0 verfügbar und ermöglicht auch das Schlussfolgern nicht unscharfen Wissens ohne Fuzzy-Bestandteile.

Der für die Überführung der Wissensdatenbank in das Eingabeformat des FuzzyDL-Reasoners zuständige Parser wurde aus Gründen der Softwarehomogenität ebenfalls auf Basis der Programmiersprache Java entwickelt. Dabei basiert der Parser auf der Grundlage eines im Paket des FuzzOWL-Plugin für Protégé enthaltenen Parsers, der an die in Kapitel 4.4 definierte syntaktische Erweiterung von OWL 2 angepasst wurde. Als Entwicklungsumgebung kam die MyEclipse Enterprise Workbench in der Version 9.1 zum Einsatz.

### 5.3.3 Anwendung der Methodik am Fallbeispiel

Zur Sicherstellung einer strukturierten Entwicklung der DfC Fuzzy-Ontologie kam der DfC Unified Process zum Einsatz. Auf die detaillierte Darstellung des Vorgehensmodells soll im vorliegenden Anwendungsszenario verzichtet werden, da dieses bereits in Kapitel 5.2 gegeben wurde. Im Folgenden wird lediglich der darin entstandene Eigenschaftenglossar, der

Merkmalsglossar, sowie ein Auszug der Regelbasis beschrieben, die dem besseren Verständnis des Anwendungsszenarios dienen. Im Eigenschaftenglossar werden fünf für das Anwendungsszenario beispielhafte Eigenschaften beschrieben.

<b>Eigenschaft</b>	<b>Beschreibung</b>
Fahrleistung	Unter Fahrleistung wird eine Gruppe von Kennzahlen der Motorisierung eines Fahrzeugs verstanden, welche sich unter anderem aus Höchstgeschwindigkeit, Beschleunigung und maximalem Drehmoment zusammensetzt.
Fahrzeugaussage	Die Fahrzeugaussage ist das Maß, innerhalb dessen das Fahrzeug den Charakter und die Lebenseinstellung des Fahrers widerspiegeln soll.
Nutzungshäufigkeit	Als Nutzungshäufigkeit bezeichnet man die Summe der von Fahrzeugen zurückgelegten Strecke in Kilometern (km) unabhängig von der beförderten Menge von Verkehrsobjekten in einem vorgegebenen Zeitraum.
Preisbewusstsein	Das Preisbewusstsein ist das Preisempfinden eines Käufers. Es ist das Maß, zu dem die Käufer und Verbraucher beim Kauf eines Produkts auf den Preis achten.
Umweltfreundlichkeit	Die Umweltfreundlichkeit misst sich in Ressourceneinsatz je zurückgelegte Distanz in Kilometern.

**Tabelle 24: Eigenschaftenglossar des Automotive-Anwendungsszenarios**

Der Merkmalsglossar umfasst diverse Motorisierungen, die drei verfügbaren Ausstattungspakete sowie zwei exemplarische Ausstattungsmerkmale.

<b>Merkmal</b>	<b>Beschreibung</b>
1.2 8V	Der Motor stellt die Standardmotorisierung des Fiat 500 mit erhöhtem Ressourcenverbrauch und niedriger Fahrleistung dar.
0.9 TwinAir	Der 0.9 Twin Air Motor ist ein ressourcenschonender Benzinmotor mit niedriger Fahrleistung.
1.3 Multijet	Die höchste Motorisierung des Fiat 500 mit durchschnittlichen Umwelteigenschaften.
Paket Pop	Paket Pop ist die Bezeichnung des grundlegenden Ausstattungspaketes des Fiat 500.
Paket Lounge	Das Ausstattungspaket Lounge betont den Lifestyle-Charakter des Fiat 500 mit hochwertigen Materialien.
Paket TwinAir	Ausstattungspaket Twin Air ist die sportliche Variante des Fiat 500 mit umweltfreundlichen Benzinmotoren.
Hifi Interscope	Das Hifi Paket Interscope ist ein Car-Hifi-Soundsystem, welches im Vergleich zum serienmäßig verarbeiteten Soundsystem eine erhöhte Klangqualität bietet.
SkyDome	Das Ausstattungsmerkmal Sky Dome ist ein elektronisch gesteuertes Glasschiebedachsystem, welches die Lücke zum 500c Cabrio schließt.

**Tabelle 25: Merkmalsglossar des Automotive-Anwendungsszenarios**

Die Regelbasis verknüpft Eigenschaften und Merkmale. Nachfolgende Tabelle 26 gibt einen Auszug aus der identifizierten Regelbasis.

<b>Regelbasis</b>	
$R_1$	Wenn Fahrleistung hoch, dann 1.3_Multijet hoch.
$R_2$	Wenn Fahrleistung niedrig, dann 1.2_8V hoch.
$R_3$	Wenn Fahrleistung niedrig, dann 0.9_TwinAir hoch.
$R_4$	Wenn Umweltfreundlichkeit hoch, dann 1.3_Multijet mittel.
$R_5$	Wenn Umweltfreundlichkeit hoch, dann 0.9_TwinAir hoch.
$R_6$	Wenn Umweltfreundlichkeit hoch, dann 1.2_8V niedrig.
$R_7$	Wenn Nutzungshäufigkeit hoch, dann 1.3_Multijet hoch.
$R_8$	Wenn Fahrzeugaussage wichtig, dann Hifi_Interscope hoch.
$R_9$	Wenn Umweltfreundlichkeit hoch, dann Hifi_Interscope niedrig.
$R_{10}$	Wenn Umweltfreundlichkeit hoch, dann SkyDome hoch.
$R_{11}$	Wenn Fahrleistung hoch, dann SkyDome hoch.

**Tabelle 26: Regelbasis des Automotive-Anwendungsszenarios**

Auf Grundlage der vorgestellten Glossaren und der Regelbasis wurde die Fuzzy-Ontologie basierend auf den in Kapitel 4.4.4 vorgestellten Syntax definiert. Eine grafische Repräsentation der Ontologie ist in Abbildung 47 ersichtlich. Die Abbildung zeigt die Konzepte der Fuzzy-Ontologie sowie deren Verknüpfungen untereinander als Netzwerk an. Darüber hinaus sind drei Instanzen ersichtlich, die Bewertungen von Kundeneigenschaften durch einen Kunden im Rahmen der Konfiguration repräsentieren. So ist die Kundeneigenschaft Nutzungshäufigkeit (Konzept `customer_usage_frequency`) exemplarisch mit dem Wert 15 bewertet.

Der Fuzzy-Reasoner ist in der Folge dafür zuständig, die Wissensbasis auszuwerten und die zu den Kundenbewertungen passenden Leistungsmerkmale zurückzuliefern. Der Fuzzy-Reasoner übernimmt somit die Funktion des Fuzzy-Reglers. Den Eingabeparameter des Fuzzy-Reasoners liefert die zuvor definierte Fuzzy-Ontologie als Wissensbasis. Diese wird durch den Parser in ein für den Fuzzy-Reasoner verarbeitbares Format umgewandelt. Als Ausgabe erhält man die zu den gewählten Kundeneigenschaften passenden Leistungsmerkmale mitsamt deren Sicherheitsgraden. Die Ausgabe erfolgt in der Java Konsole der Eclipse Entwicklungsumgebung. In Abbildung 46 ist ersichtlich, dass von den drei Ausstattungspaketen Pop, Lounge und TwinAir das Paket Pop mit einem Sicherheitsgrad von 90 als am Höchsten bewertet wird. Der Kunde wird sich am sichersten für dieses Paket entscheiden, da unter anderem die Nutzungshäufigkeit als sehr gering bewertet wurde.

Die Anwendbarkeit der Fuzzy-Ontologie als Wissensbasis sowie des Fuzzy-Reasoners als Lösungskonfigurator der Methodik ‚Design for Customer‘ wurde hiermit gezeigt.

Die grafische Repräsentation der Fuzzy-Ontologie zeigt die Konzepte und Instanzen der Fuzzy-Ontologie sowie deren Verknüpfungen untereinander als Netzwerk an.

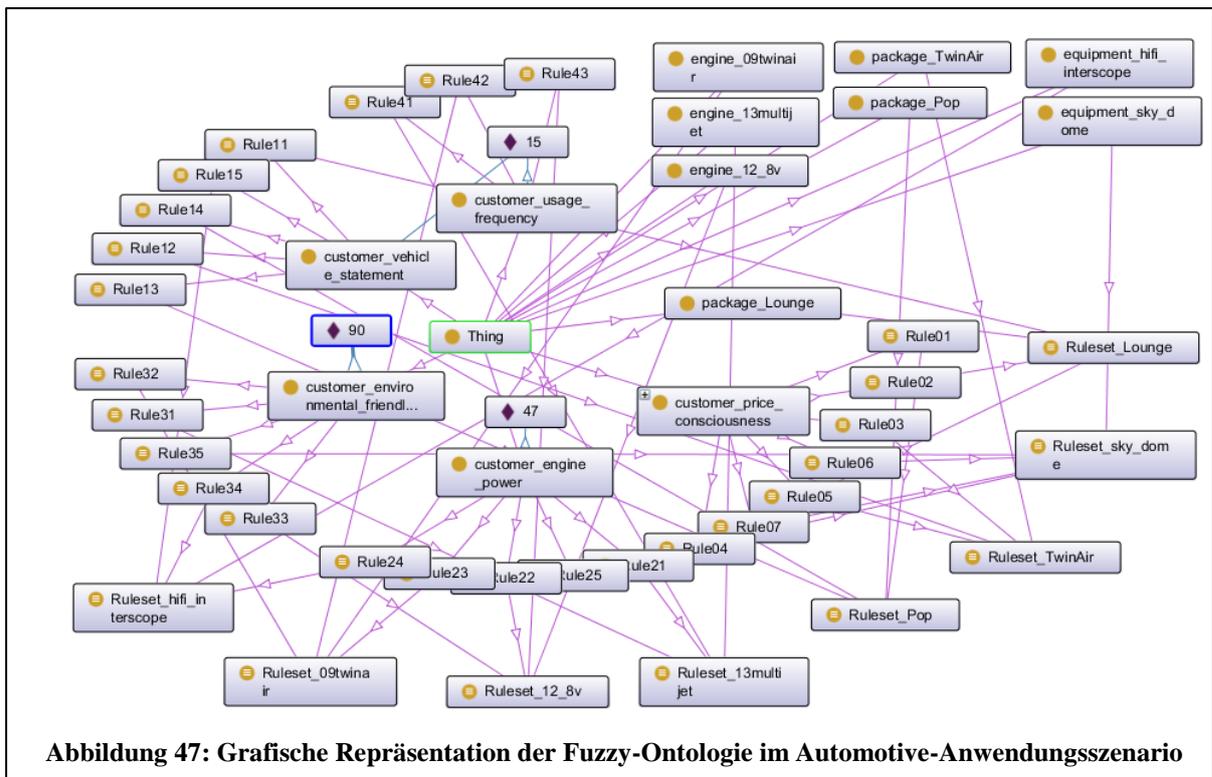


Abbildung 47: Grafische Repräsentation der Fuzzy-Ontologie im Automotive-Anwendungsszenario

Die Ausgabe des Fuzzy-Reasoners ist in der Java-Konsole der Entwicklungsumgebung Eclipse ersichtlich. Von den drei Ausstattungspaketen Pop, Lounge und TwinAir das Paket Pop mit einem Sicherheitsgrad von 90 als am Höchsten bewertet wird.

```

System.out.println("Anzahl an vorhandenen Queries:" + queries.size());
for ( int i = 0; i < queries.size(); i ++ ) {
    System.out.println("Query Nummer " + i + " : " + queries.get(i).toString());
}
System.out.println();

// Nachdem KB und Queries generiert worden sind, wird logische Inferenz gestartet
kb.solveKB();

for ( int i = 0; i < queries.size(); i ++ ) {

    // Löse eine Query i und schreibe die Ergebnisse in result
    Solution result = queries.get(i).solve(kb);

    // Schreibe das Ergebnis auf die Konsole

Anzahl an vorhandenen Queries:8
Query Nummer 0: Middle of the maxima defuzzification of feature package_Lounge for instance customer_choice =
Query Nummer 1: Middle of the maxima defuzzification of feature package_Pop for instance customer_choice =
Query Nummer 2: Middle of the maxima defuzzification of feature package_TwinAir for instance customer_choice =
Query Nummer 3: Middle of the maxima defuzzification of feature engine_12_8v for instance customer_choice =
Query Nummer 4: Middle of the maxima defuzzification of feature engine_13multiJet for instance customer_choice =
Query Nummer 5: Middle of the maxima defuzzification of feature engine_09twinair for instance customer_choice =
Query Nummer 6: Middle of the maxima defuzzification of feature equipment_hifi_interscope for instance customer_choice =
Query Nummer 7: Middle of the maxima defuzzification of feature equipment_sky_dome for instance customer_choice =

Middle of the maxima defuzzification of feature package_Lounge for instance customer_choice = 10.0
Language: ALCF (D)

Middle of the maxima defuzzification of feature package_Pop for instance customer_choice = 90.0
Language: ALCF (D)

Middle of the maxima defuzzification of feature package_TwinAir for instance customer_choice = 10.0
Language: ALCF (D)

```

Abbildung 46: Ausgabe des Fuzzy-Reasoners im Automotive-Anwendungsszenario

### 5.4 Zusammenfassung

Das Kapitel 5 dient der Validierung der im Rahmen des vorliegenden Forschungsvorhabens aufgestellten Methodik ‚Design for Customer‘. Zur Validierung wurden zwei Anwendungsszenarien definiert, welche die Methodik in ihrer Gänze abdecken und anhand zweier Beispiele aus der Praxis verständlich aufbereiten.

Im ersten Anwendungsszenario wurde die Gesamtmethodik auf Basis eines Beispiels aus dem Konsumgüterbereich überprüft. Grundlage des Anwendungsszenarios liefert das Leistungsspektrum des Unternehmens NAVIGON als Anbieter von Navigationslösungen. Unter Zuhilfenahme des DfC Unified Process wird ein Lösungskonfigurator hergeleitet, der auf dem DfC Fuzzy-Controller beruht und die Konfiguration basierend auf den Bedürfnissen und Anforderungen des Kunden erlaubt. Das so gewonnene Nutzerverhalten wird mithilfe des Fuzzy-Apriori-Algorithmus aus dem Schritt DfC Kundenanforderungsmanagement in Kundenanforderungen überführt. Durch die nahtlose Integration in die Entwicklungsprozesse des Anbieters wird die kontinuierliche Anpassung eines Lösungsportfolios an sich wandelnde Kundenbedürfnisse erlaubt.

Das zweite Anwendungsszenario basiert auf dem Produktspektrum des Automobilherstellers Fiat und wurde im Rahmen des EU-Forschungsprojektes e-CUSTOM erstellt. Ziel des zweiten Anwendungsszenarios war die Validierung der Anwendbarkeit semantischer Technologien zur Erstellung eines Lösungskonfigurators basierend auf den in Kapitel 4.4 erläuterten Erweiterungen und Syntax als Alternative zum DfC Fuzzy-Controller. Im Anwendungsfall konnte die Nutzbarkeit der aufgestellten Erweiterungen auf Basis einer prototypischen Implementierung gezeigt werden.

## 6 Zusammenfassung und Ausblick

Das vorliegende Kapitel fasst die Inhalte und Ergebnisse der Arbeit übersichtlich zusammen, gibt einen Ausblick, wie die Industrie beim erfolgreichen Wandel zum Lösungsanbieter unterstützt werden kann und schließt mit der Identifikation des weiteren Forschungsbedarfs.

### 6.1 Zusammenfassung

Das vorliegende Forschungsvorhaben stellt die Methodik ‚Design for Customer‘ zur Verfügung. Diese ermöglicht die kontinuierliche Anpassung eines Lösungsportfolios an sich wandelnde Kundenbedürfnisse basierend auf einem bedürfnisorientierten Lösungskonfigurator.

Der Handlungsbedarf des Forschungsvorhabens ergab sich aus einer empirischen Marktuntersuchung sowie einer intensiven Analyse des bestehenden Stands der Technik in den Bereichen hybride Wertschöpfung sowie Anforderungsmanagement. In diesem Kontext stellt die Marktstudie die bislang größte Untersuchung zum Thema hybride Wertschöpfung im deutschsprachigen Raum dar. In ihr wurde deutlich, dass der Markt für Kundenlösungen als attraktiver Wachstumsmarkt angesehen wird, die methodische und insbesondere operative Unterstützung bislang jedoch nur unbefriedigend gewährleistet ist. Insbesondere hinsichtlich der Integration von Kundenanforderungen zeigte sich das Vorherrschen altbewährter und wenig strukturierter Methoden, die von den Unternehmen als unzureichend bezeichnet werden.

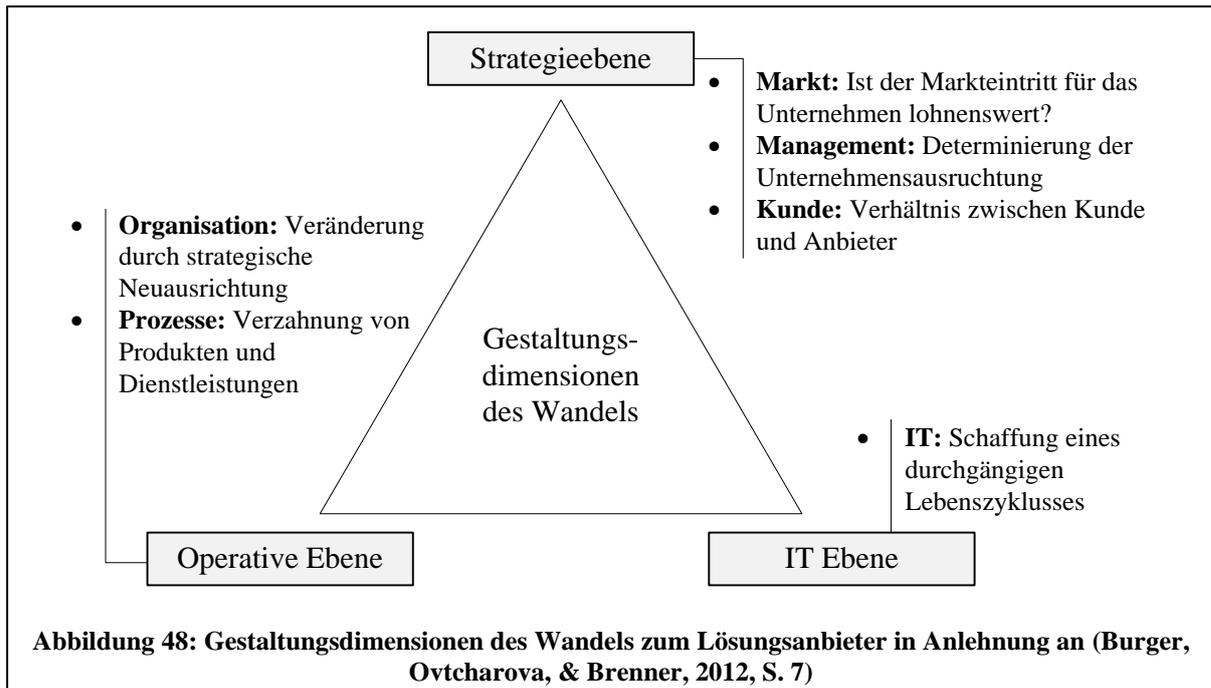
In der Folge wurden bestehende Ansätze aus Wissenschaft und Technik mit dem Ziel analysiert, den Handlungsbedarf mit bereits bestehenden Techniken und Werkzeugen möglichst umfassend abdecken zu können. Zur Verknüpfung von Kunden- und Anbietersicht im Lösungskonfigurator wurden ausgewählte Methoden der Künstlichen Intelligenz einer Stärken-Schwächen-Analyse unterzogen. Fuzzy-Systeme wurden als aussichtsreiche Kandidaten identifiziert, die eigenschaftsbasierte Sprache des Kunden mit der merkmalsbasierten Sprache des Anbieters zu verbinden. Um auch implizites Kundenwissen erfassen und in die Konfigurationsentscheidung einfließen zu lassen, wurden semantische Technologien untersucht. Fuzzy-Ontologien verbinden dabei die Vorteile von Fuzzy-Systemen und semantischen Technologien miteinander. Zur Rückführung des Kundenwissens wurden Methoden des Knowledge Discovery in Databases vorgestellt. Dabei wurde insbesondere auf Methoden der Assoziationsanalyse eingegangen, die Abhängigkeiten zwischen den untersuchten Objekten des Datenbestandes beschreiben. Die somit identifizierten Ansätze bilden die Grundlage zur Erstellung der eigentlichen Methodik und werden darin weiterentwickelt und an die spezifische Problemstellung angepasst.

Kern des vorliegenden Forschungsvorhabens stellt die Methodik ‚Design for Customer‘ dar. Die Methodik ermöglicht dem Kunden die Konfiguration basierend auf Kundennutzen und Kundenbedürfnissen anstelle von Produktmerkmalen. Der Anbieter auf der anderen Seite wird in die Lage versetzt, kontinuierlich die Anforderungen seiner Kunden erfassen und in die Entwicklungsprozesse rückführen zu können. Somit wird die Anpassung des Lösungsportfolios an sich wandelnde Kundenbedürfnisse gewährleistet. Zur Zielerreichung wird die Methodik ‚Design for Customer‘ in fünf methodische Schritte gekapselt, welche die aufgestellten Handlungsbedarfe in ihrer Gänze abdecken. Der erste methodische Schritt liefert ein Vorgehensmodell, das die strukturierte Entwicklung des geforderten bedürfnisorientierten Lösungskonfigurators erlaubt. In Schritt zwei werden zwei alternative Ansätze zur Verknüpfung von Kundeneigenschaften und Leistungsmerkmalen vorgestellt, welche das Rückgrat des Lösungskonfigurators bilden. Zur Sicherstellung, dass nicht nur die methodische Basis des Ansatzes, sondern das Gesamtsystem den Bedürfnissen des Benutzers entsprechend gestaltet ist, liefert der dritte Schritt der Methodik Methoden- und Werkzeugbaukästen des Usability Engineering. Für die Analyse und nachgelagerte Rückführung von Kundenbedürfnissen in die Entwicklungsprozesse von Lösungsanbietern und somit ein kontinuierliches Anforderungsmanagement sorgen die Schritte vier und fünf der Methodik. Ein Algorithmus zur Assoziationsanalyse ermöglicht die Identifikation von Anforderungen, die über das Prozessmodell nach Weber et al. (Weber, Steinbach, & Botta, 2004) in die Entwicklungsprozesse von Anbietern integriert werden.

Um die Anwendbarkeit der aufgestellten Methodik unter praktischen Gesichtspunkten zu demonstrieren, wurden zwei Anwendungsszenarien aus unterschiedlichen Industriezweigen definiert. Im ersten Konsumgüter-Anwendungsszenario wird die Gesamtmethodik anhand eines Leistungsportfolios für Navigationslösungen demonstriert. Navigationslösungen eignen sich durch ihren Variantenreichtum hinsichtlich Sach- und Dienstleistungsbestandteilen und die oftmals vorhandene Fachfremdheit der Kunden über einzelne Leistungsbestandteile als treffendes Anwendungsobjekt. Das zweite Anwendungsszenario mit Fokus auf die Automotive-Industrie ist angelehnt an das von der europäischen Kommission innerhalb des siebten Forschungsrahmenprogrammes geförderte Projekt e-CUSTOM<sup>7</sup>, welches zum Ziel hat, den Kunden intensiver in die Prozesse von Herstellern zu integrieren und die Lücke zwischen Massenproduktion und Mass Customization zu schließen. Ziel des zweiten Anwendungsszenarios ist die Validierung der Anwendbarkeit semantischer Technologien zur Erstellung eines Lösungskonfigurators.

---

<sup>7</sup> A Web-based Collaboration System for Mass Customization, Grant Agreement No. FP7-2010-NMP-ICT-FoF 260067



## 6.2 Ausblick

In der Arbeit wurde aufgezeigt, dass zunehmender Wettbewerbsdruck, insbesondere durch Niedriglohnländer sowie die Internationalisierung der Märkte Unternehmen vor neue Herausforderungen stellen. Die Stärke deutscher Industrieunternehmen, sich vorrangig über technische Produkteigenschaften vor ausländischer Konkurrenz abzugrenzen, wird zunehmend erschwert.

Eine Möglichkeit, diesen Herausforderungen zu begegnen, ist die Vermarktung von Kundenlösung als Kombination aus Produkten und Dienstleistungen, welche am Markt als integrierte Leistungsbündel zur Bewältigung kundenspezifischer Probleme angeboten werden. Vorwiegend in der High-Tech und Electronics-Branche profilieren sich Unternehmen wie General Electric, HP, IBM, Nokia oder Rolls Royce mit Kundenlösungen bereits nachweislich erfolgreich im Wettbewerb (Davies, Brady, & Hobday, 2006, S. 39). Von solchen Best Practices können Unternehmen weniger reifer Branchen lernen.

Der Wandel vom produzierenden Unternehmen hin zum Lösungsanbieter erfordert jedoch ein tiefgreifendes Umdenken und ist für Unternehmen mit zahlreichen Anstrengungen verbunden, da er einer Anpassung des Gesamtunternehmens bedarf. Der Wandel erstreckt sich angefangen von der Strategieebene über die operative Ebene bis hin zur unterstützenden IT-Infrastruktur (siehe Abbildung 48) und ist mit monetären, zeitlichen und ressourcenbedingten Aufwendungen verbunden. Eine Auswahl an Methoden, welche den Wandel innerhalb der unterschiedlichen

Gestaltungsebenen unterstützen, wird beispielsweise von Burger et al. (Burger, Ovtcharova, & Brenner, 2012) vorgestellt. So stellt die im Rahmen dieser Dissertation entwickelte Methodik ‚Design for Customer‘ lediglich einen Baustein dar, welcher Industrieunternehmen darin unterstützt, den Wandel zum Lösungsanbieter erfolgreich zu gestalten. Die vorgestellte Methodik darf daher nicht losgelöst, sondern in Kombination mit weiteren Ansätzen gesehen werden. Erst durch deren Zusammenwirken werden die Nutzenpotentiale des Wandels zum Lösungsanbieter ausgeschöpft.

Als große Herausforderung für die Zukunft wurde bei der Bearbeitung der Dissertation die große Diskrepanz zwischen wissenschaftlicher Theorie und industrieller Praxis deutlich. Um dieser entgegen zu wirken müssen bestehende Forschungsergebnisse in verstärktem Maße über Transferprojekte in die Industrie getragen werden. Ferner wurde offensichtlich, dass der Industrie einfach anwendbare Methoden und Werkzeuge fehlen, welche sie auf Arbeitsebene zu einem vereinfachten Wandel befähigen. Dies kann über die Bereitstellung von Methodenbaukästen als Ergänzung generischer Vorgehensmodelle erfolgen, die unternehmensspezifisch an spezifische Randbedingungen angepasst werden können.

Um den Rahmen dieser Arbeit nicht zu sprengen, wurden Abgrenzungen hinsichtlich des Aufgabenbereichs gemacht. Über den Umfang dieser Arbeit hinausgehender Forschungsbedarf wird in den folgenden Punkten gesehen:

- Es besteht ein allgemeiner Forschungsbedarf hinsichtlich Methoden und Systemen zur Konfiguration. Das Themenfeld wurde trotz zunehmender Individualisierung bislang nur unzureichend bearbeitet, womit nicht ersichtlich ist, welchen Problemstellungen Unternehmen entgegenblicken und welchen weiteren Handlungsbedarf sie im Bereich Konfiguration sehen.
- Die vorliegende Arbeit fokussierte sich darauf, Zusammenhänge zwischen den Kundennutzen beschreibenden Eigenschaften und den strukturbeschreibenden Merkmalen herauszuarbeiten. Obwohl mithilfe eines Apriori-Algorithmus grundsätzlich auch Zusammenhänge innerhalb der Eigenschaften und innerhalb der Merkmale aufgefunden werden können, wurden diese im Rahmen der vorliegenden Arbeit nicht explizit betrachtet. Eine solche integrierte Betrachtung schafft jedoch zusätzliches Wissen über Zusammenhänge und Auswirkungen.
- Zur kosteneffizienten Bereitstellung kundenspezifischer Lösungskonfigurationen strukturieren aktuell viele Unternehmen des deutschen Maschinen- und Anlagenbaus ihr Leistungsportfolio in modular vordefinierte Strukturen. Der modulare Aufbau von Leistungsmerkmalen im Sinne eines Baukastensystems wurde innerhalb dieser Arbeit nicht betrachtet. Die Rückführung von Kundenwissen in bestehende Baukastenstrukturen ist die logische Konsequenz und bedarf einer gesonderten Betrachtung.

- Es existieren keine Industriestandards zur Strukturierung und Kategorisierung von Anforderungen. Analog zu ECI@ss als branchenübergreifendem Standard für die Klassifizierung und Beschreibung von Produkten und Dienstleistungen besteht der Bedarf an einem Standard für Anforderungen im Allgemeinen und Kundenanforderungen im Speziellen. Im Rahmen der vorliegenden Arbeit wurden die Anforderungen des Kunden anhand eigener Kriterien kategorisiert, was in der Praxis eine unzureichende Möglichkeit der Vergleichbarkeit und Kommunikation zulässt.



## Glossar

<i>Anforderung</i>	Eine Anforderung ist eine Aussage über eine zu erfüllende Eigenschaft oder zu erbringende Leistung eines Produktes, Prozesses oder der am Prozess beteiligten Personen.
<i>Anforderungsmanagement</i>	Das Anforderungsmanagement ist ein kooperativer, iterativer, inkrementeller Prozess, dessen Ziel es ist zu gewährleisten, dass alle relevanten Anforderungen bekannt und in dem erforderlichen Detaillierungsgrad verstanden sind, die involvierten Stakeholder eine ausreichende Übereinstimmung über die bekannten Anforderungen erzielen, alle Anforderungen konform zu den Dokumentationsvorschriften dokumentiert bzw. konform zu den Spezifikationsvorschriften spezifiziert sind.
<i>Apriori-Algorithmus</i>	Der Apriori-Algorithmus ist ein Verfahren zur Assoziationsanalyse, der häufig gemeinsam auftretende Objekte aus einem Datenbestand extrahiert. Ergebnis der Anwendung solcher Algorithmen sind Assoziationsregeln, die signifikante Abhängigkeiten zwischen den untersuchten Objekten des Datenbestandes beschreiben.
<i>Assoziationsanalyse</i>	Die Assoziationsanalyse befasst sich mit der Entdeckung von Abhängigkeiten zwischen einzelnen Elementen eines Datenbestandes. Meist wird die Warenkorbanalyse als typisches Beispiel für Assoziationsanalysen herangezogen.
<i>Bayes'sches Netz</i>	Bayes'sche Netze bauen auf dem beschriebenen Bayes Theorem auf und stellen einen Ansatz zur Repräsentation probabilistischer Zusammenhänge mithilfe graphischer Modelle dar.
<i>Dienstleistung</i>	In Abgrenzung zu Sachleistungen (materielle Güter) spricht man bei Dienstleistungen von immateriellen Gütern. Als typische Merkmale gelten die fehlende Lagerbarkeit, die Integration des externen Faktors sowie die Gleichzeitigkeit von Produktion und Verbrauch (Uno-actu-Prinzip).
<i>Eigenschaft</i>	Eigenschaften beschreiben das Verhalten einer Kundenlösung und können vom Entwickler nicht direkt beeinflusst werden. Die vom Kunden artikulierten Anforderungen beziehen sich meist auf die Eigenschaften einer Kundenlösung. Beispiele für Eigenschaften sind Sicherheit, Ästhetik oder Umwelteigenschaften.

<i>Ergonomie</i>	Ergonomie befasst sich mit dem Verständnis der Wechselwirkungen zwischen menschlichen und anderen Elementen eines Systems mit dem Ziel, das Wohlbefinden des Menschen und die Leistung des Gesamtsystems zu optimieren.
<i>Fuzzy-Logik</i>	Die Fuzzy-Logik ist eine Theorie, die zur Modellierung von Unschärfe in umgangssprachlichen Beschreibungen entwickelt wurde und das System der booleschen Logik auf alle reellen Zahlen im Intervall [0,1] erweitert.
<i>Fuzzy-Ontologie</i>	Fuzzy-Ontologien erweitern herkömmliche Ontologien um die Darstellbarkeit menschlicher Unschärfe.
<i>Fuzzy-Regler</i>	Fuzzy-Regler oder Fuzzy-Controller sind auf Fuzzy-Logik basierende Gesamtsysteme zur Ableitung logischer Schlussfolgerungen, die aus den Teilschritten Fuzzifizierung, Inferenz und Defuzzifizierung bestehen.
<i>Hybride Wertschöpfung</i>	Die hybride Wertschöpfung befasst sich mit der Entwicklung und Bereitstellung von Kundenlösungen als an Kundenwünsche angepasste Kombinationen aus Sach- und Dienstleistungen.
<i>Knowledge Discovery in Databases</i>	Knowledge Discovery in Databases (KDD) ist ein Ansatz der Datenanalyse und zielt darauf ab, in umfangreichen Datenbeständen implizit vorhandenes Wissen zu entdecken und explizit zu machen.
<i>Künstliche Intelligenz</i>	Die Künstliche Intelligenz (KI) als Informatikdisziplin beschäftigt sich mit der Modellierung intelligenter Maschinen und der Nachahmung menschlichen Verhaltens am Computer.
<i>Künstliches neuronales Netz</i>	Künstliche neuronale Netze (KNN) stellen Modelle biologischer neuronaler Netze dar, welche aus einer Ansammlung einzelner Neuronen bestehen, die miteinander vernetzt Reizsignale verstärken oder dämpfen.
<i>Kundenlösung</i>	Eine Kundenlösung ist eine Kombination aus Sach- und Dienstleistungen, die speziell an die Kundenwünsche angepasst werden können und welche eng miteinander verbunden sind.
<i>Linguistischer Term</i>	Die Ausprägungen eines modellierten Konzeptes (linguistische Variable) werden in der Fuzzy-Logik als linguistische Terme bezeichnet.
<i>Linguistische Variable</i>	In der Fuzzy-Logik werden die modellierten Konzepte als linguistische Variable bezeichnet. In der Methodik ‚Design for Customer‘ werden sie von den Kundeneigenschaften und Anbietermerkmalen repräsentiert.
<i>Merkmal</i>	Merkmale beschreiben die Struktur, Gestalt und Beschaffenheit einer Kundenlösung und können vom Entwickler direkt beeinflusst werden.

---

	Beispiele für Merkmale sind Geometrie, Materialien, oder Dimensionen.
<i>Methode</i>	Unter einer Methode wird ein planmäßiges, auf einem Regelsystem beruhendes Vorgehen verstanden, das der Erreichung von vordefinierten Zielen und der Unterstützung der Denk- und Arbeitsweise der jeweiligen Benutzer dient.
<i>Ontologie</i>	Unter einer Ontologie wird ein Konstrukt verstanden, das Wissen über ein bestimmtes Gebiet repräsentiert und den Wissensaustausch innerhalb dieses Gebiets ermöglicht.
<i>Ontology Engineering</i>	Das Ontology Engineering beschäftigt sich mit der strukturierten Entwicklung, Wartung und Pflege von Ontologien. Es hat das Ziel, Domänenwissen innerhalb einer Gruppe strukturiert zu formalisieren, um den Wissensaustausch zwischen Menschen und Informationssystemen zu ermöglichen.
<i>Sachleistung</i>	Sachleistungen oder auch Produkte werden in die Elementarfaktoren Rohstoffe, Produktions-/Betriebsmittel und Verbrauchsgüter eingeteilt. Zu Sachleistungen zählen Material, gefertigte Vorprodukte und gefertigte Endprodukte.
<i>Semantische Technologien</i>	Semantische Technologien sind Grundlagentechnologien, welche die Bereitstellung von Informationen mitsamt ihrer Bedeutung gestatten, so dass eine Verarbeitung durch Maschinen ermöglicht wird. Zu diesen gehören neben Wissensrepräsentationssprachen für Ontologien auch Methoden und Werkzeuge zu deren Erstellung, Wartung und Anwendung.
<i>Usability Engineering</i>	Das Usability Engineering ist ein Teilprozess der Entwicklung und Gestaltung technischer Systeme und ergänzt das klassische Software Engineering um ergonomische Aspekte.
<i>Validierung</i>	Bestätigen aufgrund einer Untersuchung und durch Bereitstellung eines Nachweises, dass die besonderen Forderungen für einen speziellen beabsichtigten Gebrauch erfüllt worden sind.
<i>Web Ontology Language</i>	Die Web Ontology Language (OWL) ist eine Sprache zur Wissensrepräsentation, die auf formaler Logik basiert und somit den Zugriff auf implizites Wissen ermöglicht.



## Anhang A. Methodiken des Ontology Engineering

Das Ontology Engineering beschäftigt sich mit der strukturierten Entwicklung, Wartung und Pflege von Ontologien. Trotz der engen Verwandtschaft zur reinen Datenmodellierung geht es wesentlich über diese hinaus. Die Informatik definiert das Ontology Engineering als den Zweig des Knowledge Engineering, welcher das Ziel hat Domänenwissen innerhalb einer Gruppe strukturiert zu formalisieren, um den Wissensaustausch zwischen Menschen und Informationssystemen zu ermöglichen (Jarrar & Meersman, 2002, S. 1238) (Gomez-Perez, Fernandez-Lopez, & Corcho-Garcia, 2003, S. V). Das Knowledge Engineering selbst ist ein Bestandteil des Knowledge Management, das sich dediziert mit der Kontrolle und Verwaltung des im Unternehmen vorhandenen Wissens auseinandersetzt, um eine vereinfachte Erreichung der Unternehmensziele zu ermöglichen (van der Spek & Spijkervet, 2005, S. 25-26).

Im Folgenden sollen bestehende Ansätze des Ontology Engineering vorgestellt und bezüglich ihrer Tauglichkeit für die Modellierung von Kundeneigenschaften hin untersucht werden. In der Untersuchung wurden sämtliche bestehende Ansätze des Ontology Engineering einbezogen. Mehrere Ansätze werden im Rahmen der vorliegenden Arbeit jedoch nicht näher beleuchtet, da diese keinen Bezug zum vorliegenden Forschungsvorhaben zulassen.

Methodik	Datum	Veröffentlichung	Begründung der Nichteignung
Cyc	1990	(Lenat & Guha, 1990)	Die sehr rudimentäre Methodik beschreibt die Erstellung der Cyc Datenbank.
DOGMA	2002	(Jarrar & Meersman, 2002)	Fokus auf kontinuierlich wachsenden, verteilten Ontologien auf derselben Basis.
HCOME	2006	(Kotis & Vouros, 2006)	Einbindung von Ontologien in die operativen Alltagsprozesse von Wissensingenieuren.
KACTUS	1994	(Wielinga, Jansweijer, Anjewierden, & Vam Harmelen, 1994)	Weiterverwendung bestehender Ontologien, Wissenswiederverwendung in verschiedenen komplexen technischen Systemen.
KBSI IDEF5	1994	(Knowledge Based Systems, Inc. (KBSI), 1994)	Methodik benutzt keine standardisierte Terminologie und findet keine Verwendung.
ONIONS	1999	(Steve & Gangemi, 1996)	Werkzeug zum Ontologie-Merging
Ontology Dev. 101	2001	(Noy & McGuinness, 2001)	Techniklastige Guideline zur Erstellung von Ontologien für Einsteiger
SENSUS	1997	(Swartout, Ramesh, Knight, & Russ, 1997)	Halbautomatische Ableitung von Domänen-Ontologien aus einer Upper-Ontologie.
Uschold und King	1995	(Uschold & King, 1995)	Basiert auf denselben Annahmen wie Grüninger und Fox und ist weniger detailliert.

**Tabelle 27: Nicht weiter betrachtete Methoden des Ontology Engineering**

Im Weiteren werden die verbleibenden Methoden aus dem Gebiet des Ontology Engineering dargestellt und schließlich einer Bewertung hinsichtlich des aktuellen Forschungsvorhabens unterzogen. Untersucht werden die Methoden CommonKADS, DILIGENT, DynamOnt, Grüninger und Fox, METHONTOLOGY, On-To-Knowledge sowie UPON. Tabelle 28 gibt einen Überblick der betrachteten Methoden.

Name	Datum	Veröffentlichung
CommonKADS	1992	(Schreiber, Wielinga, de Hoog, Akkermans, & Van De Velde, 1994)
DILIGENT	2005	(Tempich C. , Pinto, Sure, & Staab, 2005)
DynamOnt	2006	(Gruber, Westenthaler, & Gahleitner, 2006)
Grüninger und Fox	1995	(Grüninger & Fox, 1995)
METHONTOLOGY	1997	(Fernandez & Gomez-Perez, 1997)
On-To-Knowledge	2002	(Sure, Staab, & Studer, 2002)
UPON	2009	(De Nicola, Missikoff, & Navigli, 2009)

**Tabelle 28: Untersuchte Methoden des Ontology Engineering**

## A.1 CommonKADS

*CommonKADS ist die erste Methodik, welche die Abbildung von verteiltem Domänenwissen als Modellierungsaktivität begreift. Als Metamethodik liefert CommonKADS keinen bestimmten Prozess, sondern stellt wiederverwendbare Metamodelle für spezifische Anwendungsfälle bereit.*

Die Methodik CommonKADS (Schreiber, Wielinga, de Hoog, Akkermans, & Van De Velde, 1994) fußt auf einem zyklischen Prozess, dessen Prozessschritte innerhalb eines Zyklus beliebig oft wiederholt werden können. Der Gesamtprozess gliedert sich dabei in zwei Unterprozesse, zum einen in die eigentliche Entwicklung der Modelle und zum anderen in begleitende Projektmanagementaktivitäten. Der Entwicklungsprozess geht dabei in einer zyklischen, risikogetriebenen Art ähnlich zu Böhms Spiralmodell vor (Schreiber, Wielinga, de Hoog, Akkermans, & Van De Velde, 1994, S. 28).

Die Autoren betonen, dass CommonKADS keinen bestimmten Prozess beschreibt, sondern stattdessen Metamodelle für spezifische Modellierungstätigkeiten bereitstellt (Schreiber, Wielinga, de Hoog, Akkermans, & Van De Velde, 1994, S. 33). Um die organisatorische Umwelt eines wissensbasierten Systems zu beschreiben, stehen vier Modelle (Organisationsmodell, Aufgabenmodell, Agentenmodell und Kommunikationsmodell) zur Verfügung. Daneben bietet ein viergeteiltes Modell die Möglichkeit, Wissen in generischer Form abbilden zu können. Dieses gliedert sich in statisches Wissen, Inferenzwissen, Aufgabenwissen und strategisches Wissen (Dahlem & Hahn, 2009).

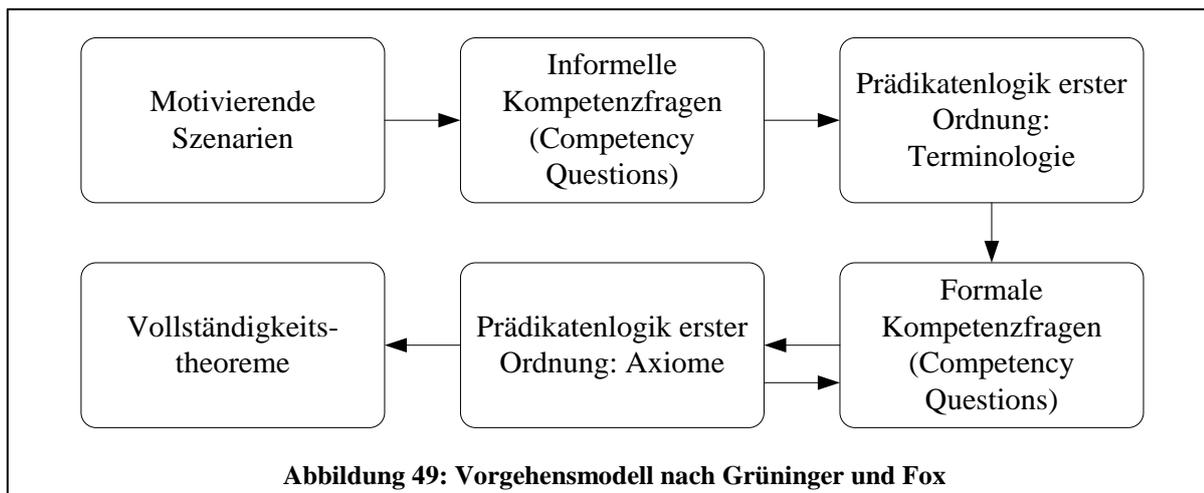
**Fazit:** CommonKADS ist eine Metamethodik zur Erstellung wissensbasierter Systeme. Die Methode ist gut dokumentiert und bietet viele wiederverwendbare Methoden (Gruber, Westenthaler, & Gahleitner, 2006, S. 256), die in den weiteren im Rahmen dieses Kapitels vorgestellten Methodiken Verwendung finden. Als nachteilig sind das Fehlen einer formalen Tiefe sowie die ungenügende Unterstützung des Erfassungs- und Modellierungsprozesses zu nennen.

## A.2 Grüninger und Fox

*Die Methodik von Grüninger und Fox liefert ein Grundgerüst zur Erstellung und Weiterentwicklung von Ontologien mit dem Ziel der Abbildung eines Unternehmens sowie dessen Prozessen. In der Methodik werden informelle Beschreibungen in prädikatenlogische Ausdrücke umgewandelt.*

Basierend auf den Ergebnissen des Forschungsprojektes TOVE (Toronto Virtual Enterprise) wurde von Grüninger und Fox eine Methodik zur Erstellung und Bewertung von Ontologien veröffentlicht (Grüninger & Fox, 1995). Die Methodik wurde im Rahmen des Projektes zur Erstellung der TOVE Ontologien verwendet, welche das Fundament einer Designumgebung zur Erstellung von Unternehmensmodellen bilden (Gomez-Perez, Fernandez-Lopez, & Corcho-Garcia, 2003, S. 119). Das Modell nach Grüninger und Fox baut als erstes Modell auf so genannten Kompetenzfragen (Competency Questions) auf, welche die Anforderungen an die Ontologie darstellen und die finale Ontologie beantworten können muss. Die Autoren betonen, dass solch ein Vorgehen eine präzisere Beurteilung verschiedener Ontologieentwürfe erlaubt (Grüninger & Fox, 1995, S. 6.10). Die Formalisierung der Ontologie erfolgt unter Zuhilfenahme der Prädikatenlogik erster Ordnung. Der vorgeschlagene Prozess gliedert sich wie auch Abbildung 49 entnehmbar in sechs Prozessschritte.

Der erste Schritt liefert *Szenarien*, welche die Erstellung der Ontologie motivieren, in Form von Problemstellungen und Beispielen. In einem zweiten Schritt werden *informelle Kompetenzfragen* aus den entstandenen Szenarien gewonnen. Diese können als Anforderungen an die Ontologie verstanden werden. Aus den Kompetenzfragen wird in einem dritten Schritt die für die Erstellung der Ontologie benötigte *Terminologie* (Konzepte und Rollen) extrahiert und in Prädikatenlogik formalisiert. Basierend auf den informellen Kompetenzfragen und der im vorherigen Schritt gewonnenen Terminologie dient der vierte Schritt zur Generierung *formeller Kompetenzfragen*. Die im fünften Schritt entstehenden *Axiome* definieren die Terminologie der Ontologie sowie Randbedingungen (Constraints) ihrer Nutzung. Sie werden ebenfalls in prädikatenlogischen Ausdrücken formuliert. Die *Vollständigkeit der Ontologie* wird im sechsten Schritt auf Basis der formalen Kompetenzfragen auf Vollständigkeit hin überprüft.



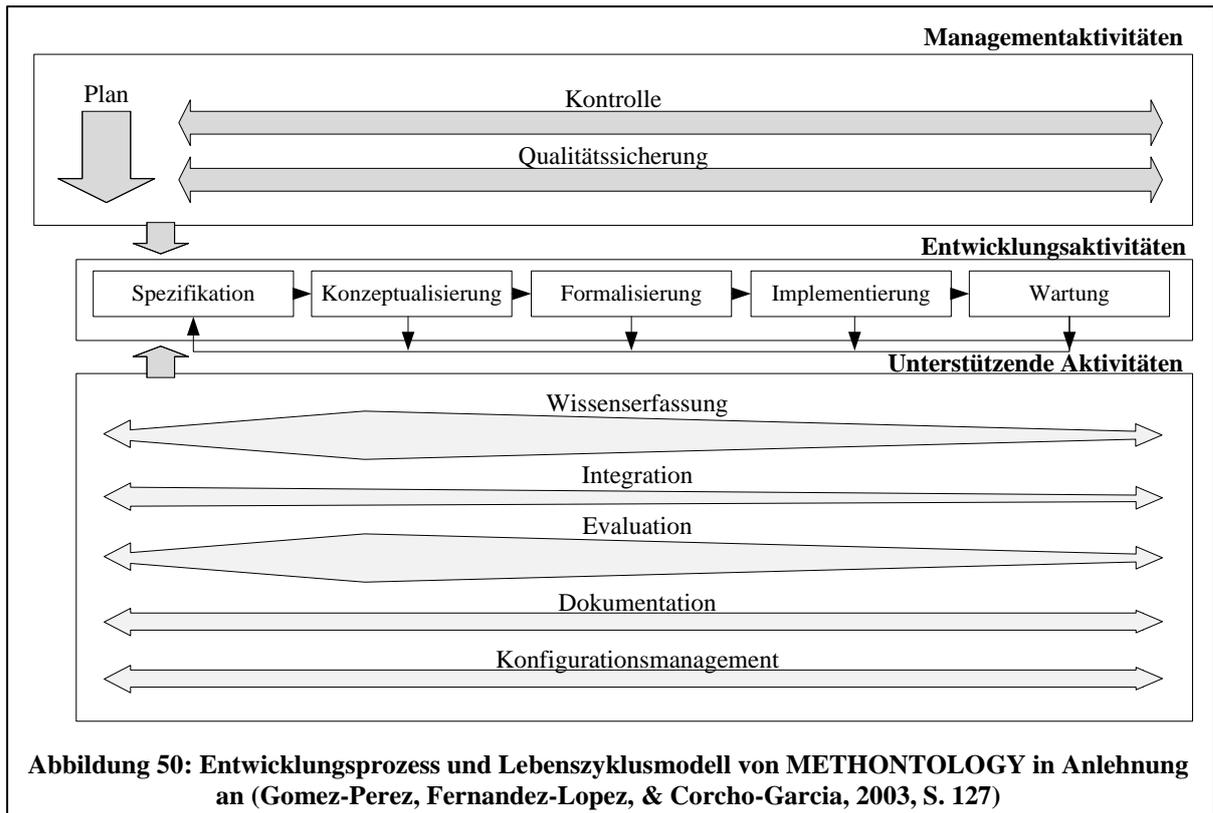
**Fazit:** Grüninger und Fox schlagen eine äußerst formale Methodik vor, welche sich der Robustheit der klassischen Logik bedient. Die Methodik hat eine klare Struktur, versäumt es jedoch, eine Beschreibung in ausreichender Detailtiefe zu liefern (Dahlem & Hahn, 2009, S. 4). Eine Beschreibung, in welcher Form das Domänenwissen erfasst und strukturiert werden kann wird ebenso wenig geliefert wie die Darstellung unterstützender Aktivitäten.

### A.3 METHONTOLOGY

*Die von Fernandez und Gomez-Perez entwickelte Methodik METHONTOLOGY ermöglicht die Entwicklung von Ontologien auf Wissensebene.*

METHONTOLOGY (Fernandez & Gomez-Perez, 1997) proklamiert einen Entwicklungszyklus für Ontologien auf Basis sich kontinuierlich weiterentwickelnder Prototypen (Evolving prototypes) (Gomez-Perez, Fernandez-Lopez, & Corcho-Garcia, 2003, S. 126). Jeder Zyklus beginnt mit einem Projektplan (Schedule). Die Methodik folgt für jeden Prototypen einem Lebenszyklus, der sechs Schritte umfasst.

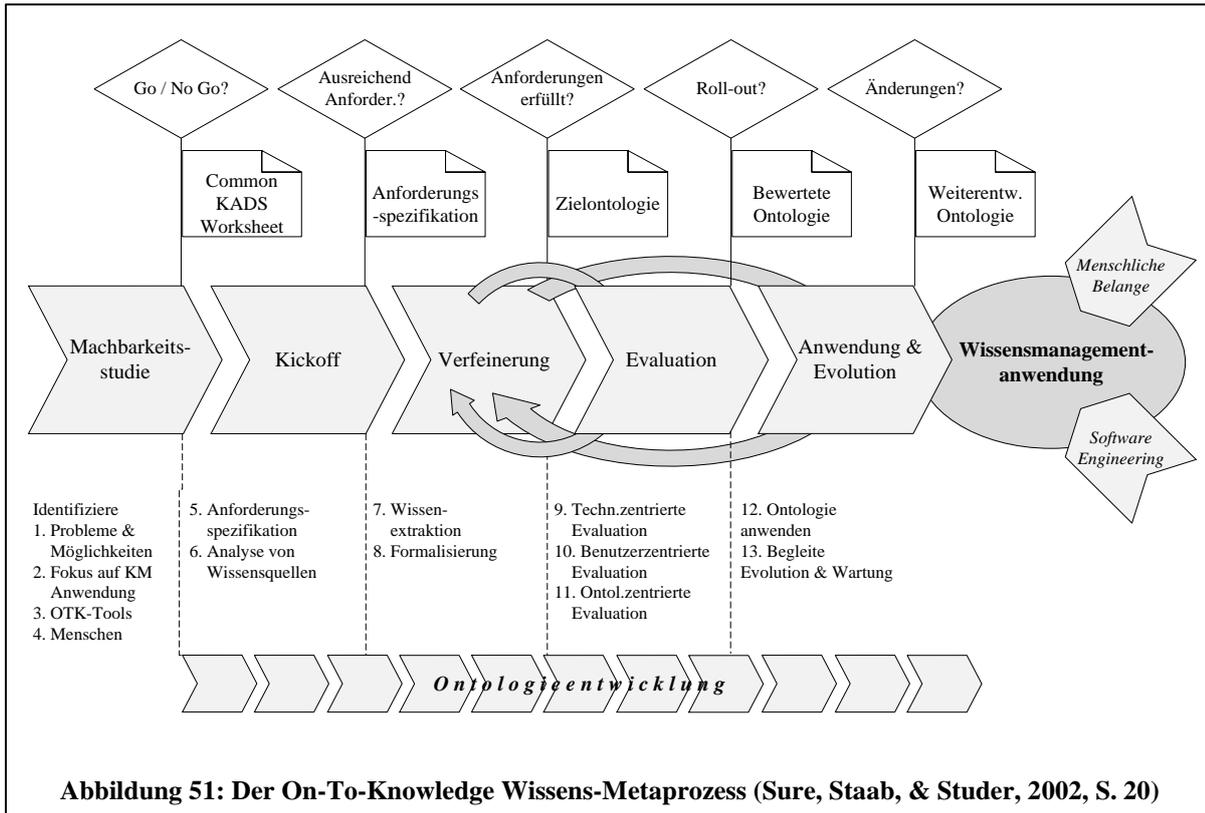
Innerhalb des Prozessschrittes der *Spezifikation* werden die Nutzungsabsicht, Endbenutzer, Formalitätsgrad und Gültigkeitsbereich der Ontologie definiert. Für die Wissensakquise werden Kreativitätstechniken wie Brainstorming, Expertengespräche, Textanalysen und Evaluationen ähnlicher Ontologien empfohlen. Innerhalb des zweiten Prozessschrittes der *Konzeptualisierung* wird aus dem gesammelten Wissen ein konzeptuelles Modell erstellt. Dieses konzeptuelle Modell besteht aus mehreren Tabellen und Graphen, die als so genannte IRs (Intermediate Representation) gespeichert werden. Der Fokus der Methodik liegt innerhalb dieses Schrittes. Im dritten Schritt der *Formalisierung* wird das konzeptuelle Modell in ein formales Modell umgewandelt und im vierten Schritt der *Integration* auf Schnittmengen mit bestehenden Ontologien hin überprüft. Die rechnergestützte *Implementierung* stellt den fünften Schritt dar, in



der das formale Modell in eine rechnerverarbeitbare Sprache transferiert wird. Als Beispiel nennen die Autoren Sprachen wie Ontolingua oder LOOM (Gomez-Perez, Fernandez-Lopez, & Corcho-Garcia, 2003, S. 131). Die sechste Phase befasst sich mit der kontinuierlichen *Wartung* der Ontologie.

Parallel zu den Entwicklungsaktivitäten werden unterstützende Maßnahmen zur kontinuierlichen Qualitäts-, und Projektkontrolle gefordert. Die Methodik basiert auf dem IEEE Standard für Softwareentwicklungsprozesse (IEEE, 1996) und wird von der FIPA für die Entwicklung von Ontologien empfohlen (Gomez-Perez, Fernandez-Lopez, & Corcho-Garcia, 2003, S. 125).

**Fazit:** Der Lebenszyklus von METHONTOLOGY basiert auf kontinuierlich wachsenden Prototypen und Techniken zur Unterstützung der einzelnen Entwicklungsphasen. Die eigentliche Entwicklung wird genauso betrachtet wie Aktivitäten des Projektmanagements. Insbesondere die klare Struktur sowie die detaillierte Beschreibung der Konzeptualisierung sind hervorzuheben. Die Wissenserfassung als unterstützende Maßnahme wird kontinuierlich durchgeführt, was zu kontinuierlichen Änderungen der Anforderungen führen kann.



## A.4 On-To-Knowledge

*On-To-Knowledge zielt darauf ab, ontologiebasierte Anwendungen des Wissensmanagements in Unternehmen einzuführen und zu pflegen. Die Methodik wurde im Rahmen eines gleichnamigen EU-Forschungsprojektes entwickelt und hat eine starke Betonung auf Projektmanagementaktivitäten im Bereich der Ontology Engineering.*

Die Methodik On-To-Knowledge (OTK) (Sure, Staab, & Studer, 2002) gliedert sich in zwei Hauptprozesse, welche unterschiedliche Anwendungsszenarien beleuchten. Der erste Hauptprozess (Knowledge Meta Process) hat die Einführung und Pflege neuer Anwendungen des Wissensmanagement (WM) im Fokus, während sich der zweite Hauptprozess (Knowledge Process) mit der Handhabung bereits bestehender WM Anwendungen befasst. Im Rahmen dieser Arbeit wird lediglich auf den ersten Hauptprozess eingegangen. Dieser untergliedert sich weiter in fünf Prozessschritte, welche weitere Schritte beinhalten. Jeder Prozessschritt enthält Informationen über dessen Inhalte und involvierte Personengruppen. Zusätzlich werden die im Rahmen des Prozessschrittes zu erreichende Resultate sowie zu treffende Entscheidungen/ Meilensteine beleuchtet.

Der erste Prozessschritt *Feasibility Study* befasst sich mit der Definition des Schwerpunktes der zu erstellenden WM Anwendung sowie der Auswahl der betroffenen Personen. Im zweiten

Prozessschritt *Kickoff* werden die Anforderungen an die Ontologie aufgenommen und semiformal in Form von MindMaps gespeichert. Das *Refinement* schließt sich an die Kickoff Phase an und beinhaltet die Formalisierung der aufgenommenen Konzepte sowie die Erstellung einer prototypischen Ontologie. Diese wird im vierten Schritt, der *Evaluation*, auf technologische, benutzerspezifische und ontologische Gesichtspunkte hin überprüft. Im letzten Prozessschritt *Application & Evolution* folgt die Übergabe der Anwendung an die Nutzer sowie deren kontinuierliche Überarbeitung.

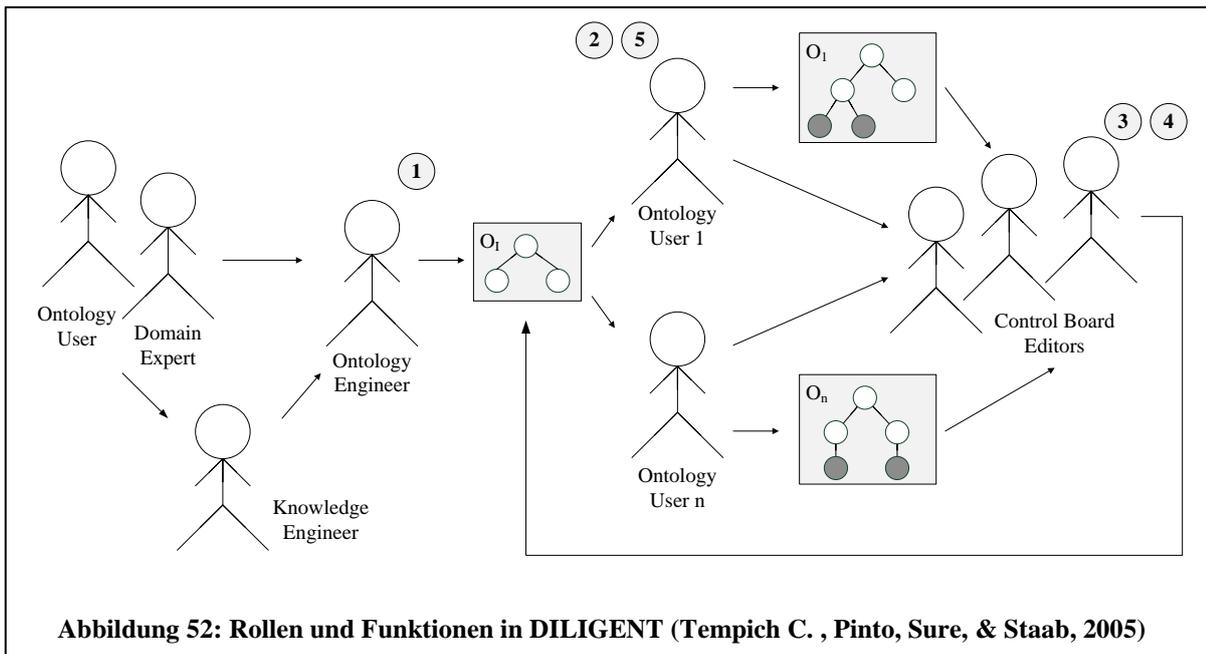
**Fazit:** Die Methodik besitzt eine klare Struktur und einen wohldefinierten Lebenszyklus (Dahlem & Hahn, 2009, S. 7). Zur Unterstützung der Nutzer werden formale Aspekte des Ontology Engineering durch grafische Repräsentationsformen wie MindMaps versteckt. Die Stärke der Methodik liegt in ihrer Betonung auf professionellen Projektmanagementaktivitäten. Details zur Wissenserfassung und Konzeptualisierung werden in weiteren Methodiken wie METHONTOLOGY oder UPON jedoch weitaus tiefgreifender behandelt.

## A.5 DILIGENT

*Im Gegensatz zu anderen Methodiken fokussiert DILIGENT die nutzerzentrierte kontinuierliche (Weiter-)Entwicklung von Ontologien. DILIGENT beschreibt die Ontologieentwicklung als Prozess, der Wissen als bewegliches Ziel betrachtet und einer verteilten Gruppe von Nutzern die Möglichkeit gibt, dieses Wissen über Argumentation formal zu extrahieren.*

Die Methodik DILIGENT (Tempich C. , Pinto, Sure, & Staab, 2005) entstand im Rahmen des EU Forschungsprojektes SEKT, innerhalb dessen semantische Technologien für das Next Generation Knowledge Management entwickelt und instrumentalisiert wurden. Der Methodik liegt die Problemstellung zugrunde, Wissensrepräsentationen für verteilte Nutzergruppen bereitzustellen. Aufgrund der starken Ausrichtung auf den Nutzer findet innerhalb der Methodik eine Trennung in mehrere beteiligte Akteure (Ontology User, Domain Expert, Knowledge Engineers, Ontology Engineers, Control Board Editors) statt.

Der Prozess gliedert sich in die fünf Prozessschritte Build, Local Adaption, Analysis, Revise und Local Update (Vrandecic, Pinto, Tempich, & Sure, 2005, S. 87). Innerhalb des *Build*-Schrittes wird von den Gruppen Ontology User, Domain Expert, Knowledge Engineers, Ontology Engineers eine initiale Ontologie erstellt. Im Vergleich zu anderen Methodiken wird nicht vorausgesetzt, dass diese bereits im ersten Schritt vollständig definiert wird. Im zweiten Schritt finden durch die Nutzer aufgrund deren spezieller Bedürfnisse lokale Anpassungen (*Local Adaptions*) statt. Das Control Board analysiert innerhalb des folgenden *Analysis*-



Prozessschrittes die lokalen Ontologien mit dem Ziel, Gemeinsamkeiten innerhalb der lokalen Ontologien aufzuspüren. Selbiges gilt bei auftretenden Change Requests der verschiedenen Benutzergruppen. Im Prozessschritt *Revise* wird die gemeinsame Ontologie so angepasst, dass ein geringerer Grad lokaler Anpassung vonnöten ist. Nach der Anpassung der zentralen Ontologie ist durch die lokalen Nutzer ein *Local Update* durchzuführen.

Hervorzuheben ist die Methode, mithilfe derer die Konzepte und Rollen der gemeinsamen DILIGENT Ontologie extrahiert werden. Eine auf der IBIS Methodik (Buckingham Shum, Motta, & Domingue, 2002) basierende Vorgehensweise erlaubt es, Diskussionen zwischen den verteilten beteiligten Nutzergruppen formal abzubilden. Eine Eingrenzung der Argumente auf die Typen Elaboration, Evaluation, Justification, Contrast, Alternative, Example sowie Counter Example erlaubt dabei eine zielführendere Kommunikation (Tempich C. , Pinto, Sure, & Staab, 2005, S. 245). Dies begründet sich in der vorhergegangenen Erkenntnis, dass eine Eingrenzung der Flexibilität einer Diskussion zu schnelleren Ergebnissen führt (Pinto, Tempich, Staab, & Sure, 2004, S. 397) (Tempich, Pinto, & Sure, Evaluating DILIGENT Ontology Engineering in a Legal Case Study, 2005, S. 336).

**Fazit:** DILIGENT hebt sich von anderen Methodiken durch die Fokussierung von Nutzern und Domänenexperten anstelle von Wissens- und Softwarearchitekten ab. Während unterstützende Aktivitäten des Projektmanagements nicht betrachtet werden, wird besonderer Wert auf die Extraktion von Konsenswissen und kollaborative Prozesse gelegt. Daneben favorisiert der Ansatz die kontinuierliche Entwicklung von Domänenwissen anstelle direkt eine möglichst optimale Lösung zu fordern.

## A.6 DynamOnt

*DynamOnt zielt darauf ab, Domänenexperten bei der Erstellung formaler Wissensmodelle zu unterstützen. Statt ein Prozessmodell zur Verfügung zu stellen, können in DynamOnt auf Basis eines Klassifikationsmodells (3D Matrix) Fragen abgeleitet werden, die zu einer zielführenderen Ontologieerstellung führen.*

DynamOnt ist eine im Rahmen des gleichnamigen österreichischen Forschungsprojektes entstandene Methodik (Gruber, Westenthaler, & Gahleitner, 2006) zur schnellen Erstellung prototypischer Ontologien. Hauptziel ist es, Domänenexperten - welche keine IT-Spezialisten oder Ontologieexperten sind - ein Werkzeug zur schnellen Kategorisierung und Segmentierung beliebiger Wissensräume zur Hand zu geben. Dies erfolgt über ein mehrdimensionales Modell (3D Modell), das Ontologien anhand dreier Klassifizierungskriterien klassifiziert. Die *Ausdrucksstärke* als erstes Klassifizierungskriterium kann von einer einfachen Begriffsliste bis hin zu Ontologien mit ausdrucksstarken Randbedingungen reichen. Abhängig davon ist eine geeignete Repräsentationssprache zu wählen. Die Frage nach der Zielgruppe der Ontologie führt zum zweiten Klassifizierungskriterium (Akzeptanz), welches Aufschluss über die Akzeptanz der zu erstellenden Ontologie gibt. Die dritte Dimension gibt den Anwendungsbereich des Modells wieder, angefangen von spezifischen Erweiterungen bestehender Modelle bis hin zu Top-Level Ontologien. Das Kriterium gibt hierbei Aufschluss über die Granularität der zu verwendenden Konzepte und Rollen.

**Fazit:** DynamOnt ist weniger eine Methodik im eigentlichen Sinn, welche einen Prozessmodell der Ontologieerstellung enthält. Daneben fehlt es der Beschreibung der Methodik an Detailtiefe und formaler Unterstützung (Dahlem & Hahn, 2009, S. 4). Stattdessen enthält DynamOnt mit dem 3D Modell ein Werkzeug zur schnellen Klassifizierung eines Wissensraums, das im Rahmen der Ontologieerstellung dafür geeignet ist, einen schnellen ersten Überblick über die Problemstellung zu erhalten.

## A.7 UPON

*UPON ist eine anwendungsfallgetriebene, iterative und inkrementelle Methodik zur Erstellung großer Ontologien. Die Methodik basiert auf dem weitverbreiteten Unified Process (UP) aus dem Software Engineering, sowie auf der Unified Modeling Language (UML) zum Ontologieentwurf.*

Die Methodik UPON (De Nicola, Missikoff, & Navigli, 2009) entstand im Rahmen des von der Europäischen Union geförderten Athena Projektes, innerhalb dessen die Methodik erstmalig validiert wurde. Ausgangspunkt der Entwicklung von UPON war die Erkenntnis, dass sich keine

Methodik des Ontology Engineering als Referenzmethodik etablieren konnte. UPON begegnet diesem Umstand, indem es auf bewährten und weitverbreiteten Prozessen und Technologien aufsetzt. Somit sollen Eintrittsbarrieren verhindert sowie die Lernkurve der beteiligten Personen verkürzt werden. Der Unified Process (Jacobson, M., Jonson, & Övergaard, 1992) bildet das Rahmenwerk der Methodik, während UML den Entwurf sowie die Konzeption der Ontologie unterstützt. Die Nützlichkeit der UML als Werkzeug des Ontologieentwurfs wird in (Guizzardi, Herre, & Wagner, 2002) erläutert.

Analog zum UP werden in UPON *Zyklen, Phasen, Iterationen* und *Workflows* verwendet. Jeder Zyklus besteht aus vier Phasen (*Inception, Elaboration, Construction* und *Transition*) und führt zur Freigabe einer neuen Ontologieversion. Jede Phase wird weiter untergliedert in Iterationen. Innerhalb jeder Iteration laufen wiederum fünf Workflows (*Requirements, Analysis, Design, Implementation* und *Test*) ab. Frühe Zyklen fokussieren dabei hauptsächlich den Requirements Workflow und die Inception Phase, während späte Zyklen ihr Augenmerk auf die Workflows Implementation und Test sowie auf die Transition Phase richten. Workflows und Phasen können demzufolge als orthogonal zueinander laufend betrachtet werden, da der Beitrag eines Workflows von der Iteration einer Phase abhängig ist (De Nicola, Missikoff, & Navigli, 2009, S. 259).

Die Workflows in UPON beschreiben einzelne Arbeitsschritte, die sukzessive abuarbeiten sind und am Ende jedes Workflows zu einem oder mehreren Ergebnissen führen. Ziel des *Requirementsworkflows* ist es, Anforderungen wie den Geschäftsnutzen und den Anwendungsbereich zu definieren. Der Requirementsworkflow untergliedert sich in sechs Unterschritte, die eine Menge von Ergebnisdokumenten liefern. Diese beinhalten die Kompetenzfragen, Anwendungsmodelle sowie ein Applikationslexikon. Der *Analysisworkflow* beinhaltet die konzeptuelle Analyse der zuvor identifizierten Anforderungen. In vier Unterschritten wird ein Domänenlexikon erstellt und mit dem Applikationslexikon zu einem Referenzglossar erweitert. Ziel des Designworkflows ist es, zu den im Referenzglossar enthaltenen Einträgen eine semantische Struktur zu erstellen. Dazu wird die linguistische Dimension verlassen und in die konzeptuelle Dimension eingestiegen (De Nicola, Missikoff, & Navigli, 2009, S. 266). Im vierten Schritt, dem *Implementation Workflow*, wird die Ontologie in einer maschinenlesbaren formalen Sprache implementiert. Der finale *Test Workflow* dient schlussendlich dazu, die semantische und pragmatische Qualität der Ontologie zu verifizieren. Eine ausführlichere Darstellung des hier nur kurz umrissenen Sachverhaltes findet sich in (De Nicola, Missikoff, & Navigli, 2009).

**Fazit:** UPON bedient sich als einzige Methodik des Ontology Engineering bereits bestehender und akzeptierter Prozesse und Technologien. Insbesondere durch die Verwendung von UML können lange Einarbeitungszeiten in unbekannte Technologien und Methoden

unterbunden werden. Zur Dokumentation und Versionierung werden UML Softwarewerkzeuge wie Microsoft Visio und Rational Rose, die leicht verfügbar sind, vorgeschlagen. Nach Aussage der Autoren liegt insbesondere hier der Vorteil zu METHONTOLOGY, das grob dieselben Entwicklungsprozesse proklamiert (De Nicola, Missikoff, & Navigli, 2009, S. 273). Der Prozess wird in einer hohen Detailtiefe beschrieben, beinhaltet jedoch keine Aussagen zu unterstützenden Aktivitäten des Projektmanagements. Einziger Schwachpunkt liegt in der relativ komplexen Methodik, die oftmals auch dem Unified Process zugesprochen wird.

## A.8 Vergleich bestehender Ansätze des Ontology Engineering

Die im Laufe dieses Kapitels dargestellten Methodiken werden im Folgenden einer komparativen Beurteilung unterzogen. Ein zur Bewertung mehrerer Methodiken des Ontology Engineering geeignetes Rahmenwerk auf Basis des IEEE 1074-1995 Standards für Lebenszyklen der Softwareentwicklung (IEEE, 1996) stellen Fernández und Gómez-Pérez (Fernandez & Gomez-Perez, 1997, S. 109 ff.) vor. Das Framework wurde darauffolgend von De Nicola et al. (De Nicola, Missikoff, & Navigli, 2009, S. 272 ff.) weiterverwendet. Im Rahmen dieser Arbeit wird ein auf diesen Arbeiten beruhendes Rahmenwerk aufgebaut.

Das auf den Kontext von Ontologien angepasste Rahmenwerk unterscheidet drei Hauptkategorien für Prozesse.

- **Prozesse des Ontologiemanagements:** Beinhalten diejenigen Tätigkeiten, welche die Erstellung eines Projektrahmenwerks für den gesamten Ontologielebenszyklus betreffen.
- **Prozesse der Ontologieentwicklung:** Die Aktivitäten lassen sich weiter in Tätigkeiten vor der eigentlichen Entwicklung, die tatsächliche Entwicklung betreffend und schließlich in der Entwicklung nachgelagerte Tätigkeiten untergliedern.
- **Integrale Prozesse:** Beinhalten eine Reihe von Aktivitäten, die zeitgleich zu den entwicklungsorientierten Tätigkeiten ablaufen, ohne die die Ontologieentwicklung jedoch nicht stattfinden könnte.

Tabelle 29 zeigt die Ergebnisse der auf dem IEEE 1074-1995 Standard basierende Vergleichsanalyse auf. Analog zu De Nicola et al. (De Nicola, Missikoff, & Navigli, 2009, S. 272 ff.) erfolgt eine Bewertung in durch die Methodik *vollständig unterstützte Kriterien* (●), *nicht unterstützte Kriterien* ( ) und *teilweise unterstützte Kriterien* (◐). Eine teilweise Unterstützung liegt dann vor, wenn das Kriterium in der Methodik zwar erwähnt, aber nur ungenügend erläutert wird oder die Praxistauglichkeit nur bedingt möglich ist.

Aus der Übersicht wird deutlich, dass keine bestehende Methodik die geforderten Kriterien zur Gänze erfüllt. Stattdessen setzen die Methodiken unterschiedliche Schwerpunkte. Während

On-To-Knowledge seinen Fokus auf die begleitenden Prozesse des Projektmanagements legt, unterstützt DILIGENT mit seiner Argumentationsmethodik primär die Wissensakquise und Anforderungserhebung in verteilten Gruppen. Das Klassifikationsmodell von DynamOnt ermöglicht insbesondere detaillierte Untersuchungen im Rahmen der Vor-Entwicklung und lässt andere Gesichtspunkte außer Acht. Bei den allgemeinen Methodiken zeichnen sich insbesondere METHONTOLOGY und UPON durch eine breite Abdeckung der geforderten Kriterien aus. UPON besitzt hierbei den Vorteil, auf bestehenden und weitverbreiteten Methoden und Technologien aufzubauen, um Eintrittsbarrieren zu verhindern.

IEEE 1074-1995 Standardprozesse			CommonKADS	Grüniger und Fox	METHONTOLOGY	On-To-Knowledge	DILIGENT	DynamOnt	UPON
Projektmanagement-Prozesse	Projektinitiierung			●		●			
	Monitoring und Kontrolle		●	●	●	●			
	Qualitätsmanagement		●		●			●	●
Ontologieentwicklungsorientierte Prozesse	Vor-Entwicklung	Umweltstudie						●	●
		Machbarkeitsstudie				●		●	
	Entwicklung	Anforderungen	●	●	●	●	●	●	●
		Design	●	●	●	●	●	●	●
		Implementierung	●	●	●	●			●
	Nach-Entwicklung	Installation							
Betrieb									
Support									
Wartung				●	●	●		●	
Stilllegung									
Integrale Prozesse	Wissenserfassung			●		●		●	
	Evaluation		●		●	●	●	●	
	Konfigurationsmgmt.			●					
	Dokumentation			●		●		●	
	Training							●	

Tabelle 29: Vergleichsanalyse von Methoden des Ontology Engineering auf Basis IEEE 1074-1995

## Abbildungsverzeichnis

Abbildung 1: Aufbau der Arbeit .....	6
Abbildung 2: Prozess der Erbringung einer Sachleistung .....	10
Abbildung 3: Prozess der Erbringung einer Dienstleistung .....	10
Abbildung 4: Dimensionen einer Dienstleistung. ....	12
Abbildung 5: Kernelemente einer Kundenlösung .....	15
Abbildung 6: Marktvolumen und Ebit-Margen hybrider Lösungsanbieter .....	16
Abbildung 7: Schalenmodell des PSSE in Anlehnung an .....	20
Abbildung 8: Ordnungsrahmen der PSS-Entwicklungsmethodik .....	21
Abbildung 9: Generisches Vorgehensmodell für den HLB-Entwicklungsprozess .....	22
Abbildung 10: Klassifizierung von Methoden des Anforderungsmanagements .....	25
Abbildung 11: Methoden der Anforderungsermittlung .....	26
Abbildung 12: Methoden der Anforderungsanalyse .....	29
Abbildung 13: Methoden der Anforderungsspezifikation .....	31
Abbildung 14: Die Bedeutung reiner Produkte schwindet .....	34
Abbildung 15: Der Markt fordert verstärkt individuelle Lösungen .....	35
Abbildung 16: Der Kunde wird als Initiator von Leistungsverbesserungen gesehen .....	35
Abbildung 17: Nutzung veralteter Methoden zur Identifikation von Kundenanforderungen .....	36
Abbildung 18: Schematischer Aufbau eines Neuronalen Netzes .....	45
Abbildung 19: Architektur eines Fuzzy-Reglers .....	48
Abbildung 20: Eine einfache Beispielontologie .....	52
Abbildung 21: Aufgabenstellungen und Lösungsverfahren im KDD .....	61
Abbildung 22: Schematischer Aufbau der Methodik Design for Customer .....	69
Abbildung 23: Schema von OWL 2 Annotationen .....	84
Abbildung 24: Schematischer Aufbau des DfC Unified Process .....	99
Abbildung 25: Übersicht über den Anforderungsworkflow .....	101
Abbildung 26: Übersicht über den Analyseworkflow .....	102
Abbildung 27: Übersicht über den Designworkflow .....	104
Abbildung 28: Übersicht über den Implementierungsworkflow .....	105
Abbildung 29: Übersicht über den Testworkflow .....	106
Abbildung 30: Guidelines zum Usability-Projektmanagement .....	111
Abbildung 31: Guideline der Usability-Anforderungserfassung .....	114
Abbildung 32: Guideline der Usability-Evaluation .....	116
Abbildung 33: Schematischer Grundaufbau des Fuzzy-Apriori-Algorithmus .....	125
Abbildung 34: Die Methode PREPROCESS_INPUT des Fuzzy-Apriori-Algorithmus .....	127
Abbildung 35: Die Methode GENERATE_FUZZY_1_ITEMSETS des Fuzzy-Apriori-Algorithmus .....	128

Abbildung 36: Die Methode GENERATE_FUZZY_K_ITEMSETS des Fuzzy-Apriori-Algorithmus ....	129
Abbildung 37: Die Methode FUZZY_APRIORI_GEN des Fuzzy-Apriori-Algorithmus .....	130
Abbildung 38: Methode GENERATE_FAR des Fuzzy-Apriori-Algorithmus .....	132
Abbildung 39: Integration der Methodik ‚Design for Customer‘ in das Schalenmodell des PSSE.	135
Abbildung 40: Mapping von Handlungsbedarfen und Nutzenpotentialen .....	140
Abbildung 41: Grafische Benutzeroberfläche des Lösungskonfigurators .....	146
Abbildung 42: Modellierung des Anwendungsszenarios auf Basis von UML .....	153
Abbildung 43: Modellierung der Konzepte und Hierarchien.....	154
Abbildung 44: Vergleich von Soll- und Ist-Relation zur Generierung von Anforderungen .....	164
Abbildung 45: Prototypische Realisierung des Automotive-Anwendungsszenarios .....	167
Abbildung 46: Ausgabe des Fuzzy-Reasoners im Automotive-Anwendungsszenario.....	171
Abbildung 47: Grafische Repräsentation der Fuzzy-Ontologie .....	171
Abbildung 48: Gestaltungsdimensionen des Wandels zum Lösungsanbieter .....	175
Abbildung 49: Vorgehensmodell nach Grüninger und Fox .....	186
Abbildung 50: Entwicklungsprozess und Lebenszyklusmodell.....	187
Abbildung 51: Der On-To-Knowledge Wissens-Metaprozess .....	188
Abbildung 52: Rollen und Funktionen in DILIGENT .....	190

## Tabellenverzeichnis

Tabelle 1: Stärken-Schwächen Analyse Bayes'scher Netze.....	44
Tabelle 2: : Stärken-Schwächen Analyse Künstlicher Neuronaler Netze .....	47
Tabelle 3: : Stärken-Schwächen Analyse von Fuzzy-Systemen.....	49
Tabelle 4: Analyse bestehender Ansätze zur Nutzung der Fuzzy-Logik.....	55
Tabelle 5: Methoden des Ontology Engineering.....	56
Tabelle 6: Vergleichsanalyse von Methoden des Ontology Engineering nach IEEE 1074-1995 .....	59
Tabelle 7: Verfahren der Assoziationsanalyse .....	63
Tabelle 8: Existierende Implementierungen von Fuzzy-Reasonern .....	96
Tabelle 9: Methodenbaukasten der Usability-Anforderungserfassung .....	115
Tabelle 10: Methodenbaukasten der Usability-Evaluation .....	117
Tabelle 11: Im Lösungskonfigurator erfasstes Kaufverhalten und abgeleiteter Handlungsbedarf...	121
Tabelle 12: NAVIGON DfC Unified Process Anforderungsspezifikation .....	149
Tabelle 13: Auszug aus dem NAVIGON Eigenschaftenglossar .....	150
Tabelle 14: Auszug aus dem NAVIGON Merkmalsglossar .....	150
Tabelle 15: Mapping von Merkmalen und Leistungen .....	151
Tabelle 16: Auszug aus der NAVIGON Regelbasis .....	152
Tabelle 17: Aus dem Nutzerverhalten ermittelte Datenbank <b>D</b> .....	157
Tabelle 18: Aufbereitete Datenbank <b>Dpreprocess</b> .....	158
Tabelle 19: Berechnung von Support und Standardabweichung für <b>L1</b> .....	159
Tabelle 20: Berechnung von Support und Standardabweichung für <b>L2</b> .....	160
Tabelle 21: Identifizierte große Objektmengen.....	161
Tabelle 22: Berechnung der Konfidenz .....	162
Tabelle 23: Generierte Fuzzy-Assoziationsregeln.....	163
Tabelle 24: Eigenschaftenglossar des Automotive-Anwendungsszenarios .....	169
Tabelle 25: Merkmalsglossar des Automotive-Anwendungsszenarios .....	169
Tabelle 26: Regelbasis des Automotive-Anwendungsszenarios .....	170
Tabelle 27: Nicht weiter betrachtete Methoden des Ontology Engineering.....	183
Tabelle 28: Untersuchte Methoden des Ontology Engineering .....	184
Tabelle 29: Vergleichsanalyse von Methoden des Ontology Engineering nach IEEE 1074-1995 ..	194



## Literaturverzeichnis

- Adamy, J. (2007). *Fuzzy Logik, Neuronale Netze und Evolutionäre Algorithmen (2. Ausg.)*. Aachen: Shaker Verlag.
- Agrawal, R., & Srikant, R. (1994). *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB*. Fast algorithms for mining association rules in large databases, (S. 487-499). Santiago, Chile.
- Akao, Y. (1990). *Quality Function Deployment: Quality Function Deployment - Integrating Customer Requirements into Product Design (Bd. 17)*.
- Alexander, I. F., & Stevens, R. (2002). *Writing Better Requirements*. Reading, Massachusetts: Addison-Wesley.
- Allemang, D., & Hendler, J. (2008). *Semantic web for the working ontologist: modeling in RDF, RDFS and OWL*. Burlington, USA: Elsevier.
- Aurum, A., & Wohlin, C. (2005). *Engineering and Managing Software Requirements*. Berlin, Heidelberg: Springer.
- Baader, F., Calvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P. F. (2003). *The Description Logic Handbook: Theory, Implementation, Applications*. Cambridge, UK: Cambridge University Press.
- Barwise, J., & Etchemendy, J. (2005). *Sprache, Beweis und Logik, Band 1: Aussagen- und Prädikatenlogik*. Paderborn: Mentis-Verlag.
- Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Amsterdam: Addison-Wesley Longman.
- Becker, J., Beverungen, D., & Knackstedt, R. (2008). *Proceedings of the 41st Hawai'i International Conference on System Sciences (HICSS-41)*. Reference Models and Modeling Languages for Product-Service Systems - Status-Quo and Perspectives for Further Research, (S. 105-115). Waikoloa, Hawai'i, USA.
- Becker, J., Beverungen, D., Knackstedt, R., & Müller, O. (2008). *Proceedings of the GI-Tagung Modellierung, Workshop Dienstleistungsmodellierung*. Konzeption einer Modellierungstechnik

zur tool-unterstützten Modellierung, Konfiguration und Bewertung hybrider Leistungsbündel, (S. 45–62). Berlin.

Becker, J., Beverungen, D., Knackstedt, R., Glauner, C., Stypmann, M., Rosenkranz, C., . . . Leimeister, J. (2008). *Ordnungsrahmen für die hybride Wertschöpfung*. In O. Thomas, & M. Nüttgens, *Dienstleistungsmodellierung - Methoden, Werkzeuge und Branchenlösungen* (S. 109–128.). Berlin Heidelberg: Physica.

Beekmann, F., & Chamoni, P. (2006). *Verfahren des Data Mining*. In P. Chamoni, & P. Gluchowski, *Analytische Informationssysteme - Business Intelligence-Technologien und -Anwendungen* (3. Ausg.). Berlin: Springer.

Berners-Lee, T., Hendler, J., & Lassila, O. (5 2001). *The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities*. *Scientific American*, 284(5), S. 34-43.

Beverungen, D., Knackstedt, R., & Müller, O. (2008). *Entwicklung Serviceorientierter Architekturen zur Integration von Produktion und Dienstleistung: Eine Konzeptionsmethode und ihre Anwendung am Beispiel des Recyclings elektronischer Geräte*. *Wirtschaftsinformatik*, 50(3), S. 220-234.

Beyer, H. R., & Holtzblatt, K. (Mai 1995). *Apprenticing with the customer*. *Communications of the ACM*, 38(5), S. 45-52.

Bobbilo, F., & Straccia, U. (2008). *Proceedings of the 17th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2008)*. fuzzyDL: An Expressive Fuzzy Description Logic Reasoner, (S. 923-930). Hong Kong, China.

Bobbilo, F., & Straccia, U. (2011). *Fuzzy Ontology Representation using OWL 2*. *International Journal of Approximate Reasoning*,(52), S. 1073-1094.

Bobillo, F., Delgado, M., & Gomez-Romero, J. (2007). *Proceedings of the 3rd ISWC Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2007)*. Optimizing the crisp representation of the fuzzy description logic SROIQ (S. 189-206). Springer.

Böhmman, T., & Krcmar, H. (2007). *Hybride Produkte: Merkmale und Herausforderungen*. In M. Bruhn, & B. Stauss, *Wertschöpfungsprozesse bei Dienstleistungen* (S. 239-255). Wiesbaden : Gabler.

Botta, C. (2007). *Rahmenkonzept zur Entwicklung von Product-Service Systems: Product-Service Systems Engineering*. Lohmar: Eul.

Bray, I. K. (2002). *An Introduction to Requirements Engineering*. Reading: Addison-Wesley.

- Breitner, C., Christoph, A., & Herzog, U. (Februar 1996). *Abhängigkeit von der IT-Abteilung aufgehoben: Data Warehouse als Schlüssel zur Bewältigung der Informationsflut*. Computerwoche, 1.
- Bruhn, M. (2006). *Markteinführung von Dienstleistungen - Vom Prototyp zum marktfähigen Produkt*. In H. Bullinger, & A. Scheer, *Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen* (S. 227-248). Berlin Heidelberg: Springer.
- Buckingham Shum, S., Motta, E., & Domingue, J. (2002). *HypACoM 2002: Facilitating Hypertext-Augmented Collaborative Modeling*. *ACM Hypertext '02 Workshop*, 11.-12. Juni 2002. Augmenting design deliberation with compendium: The case of collaborative ontology design. University Maryland, MD, USA.
- Bullinger, H.-J., & Scheer, A.-W. (2006). *Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen*. Berlin, Heidelberg: Springer.
- Burchill, G., & Fine, C. (1997). *Time versus Market Orientation in Product Concept Development: Empirically-based Theory Generation*. *Management Science*, 43(4), S. 465–478.
- Burger, A., & Ovtcharova, J. (2011). *Marktstudie und Reifegradmodelle für den Wandel zum Lösungsanbieter*. *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 5, S. 286.
- Burger, A., & Ovtcharova, J. (2012). *Endanwender-Usability als kritischer Erfolgsfaktor moderner Anwendungssysteme*. *ERP-Management - Zeitschrift für unternehmensweite Anwendungssysteme*(4), S. 28-31.
- Burger, A., & Ovtcharova, J. (2013). *Design for Customer - Sustainable Customer Integration based upon a Customer-driven Solution Configurator*. In Y. Shimonura, & K. Kimita, *The Philosopher's Stone for Sustainability* (S. 263-268). Berlin Heidelberg: Springer.
- Burger, A., Bittel, V., Awad, R., & Ovtcharova, J. (2011). *MECIT International Conference on "Applied Information and Communications Technology"*. Design for Customer – Sustainable Customer Integration into the Development Processes of Product-Service System Providers. 22-23 March 2011, Muscat, Oman.
- Burger, A., Bittel, V., Awad, R., Stanev, S., & Ovtcharova, J. (2010). *3rd International Workshop "Innovation in Information Technologies: Theory and Practice"*. Customer-Oriented Integration and Individualization Aspects Enabling the Transition from Manufacturer to Solution Provider. 06.-10.09.2010, Dresden, Germany.
- Burger, A., Ovtcharova, J., & Brenner, M. (2012). *Vom Produzenten zum Lösungsanbieter - Bestandsaufnahme und Entwicklungsoptionen für Industrieunternehmen*. *Industry Report*.

- Burton-Jones, A., Storey, V., Sugumaran, V., & Ahluwalia, P. (2005). *A semiotic metrics suite for assessing the quality of ontologies*. *Data & Knowledge Engineering*, 55(1), S. 84-102.
- Chen, P. P.-S. (1976). *The Entity-Relationship Specification - Toward a Unified View of Date*. *ACM Transactions on Database Systems*, 1(1), S. 9-36.
- Chini, L. (1976). *Marketing für gewerbliche Dienstleistungsunternehmen. Entscheidungen über den Individualisierungsgrad von Dienstleistungen*. *Internationales Gewerbearchiv*(42), S. 1-10.
- Chroust, G. (1992). *Modelle der Software-Entwicklung*. München, Wien: Oldenbourg.
- Clausing, D. (1994). *Total Quality Development. A step-by-step guide to world-class concurrent engineering*. New York, USA: ASME Press.
- Conrad, J. (2010). *Semantische Netze zur Erfassung und Verarbeitung von Informationen und Wissen in der Produktentwicklung (Bd. 49)*. Saarbrücken: Universität des Saarlandes.
- Corsten, H., & Gössinger, R. (2007). *Dienstleistungsmanagement (5., überarb. u. erw. Aufl. Ausg.)*. München et al.: Oldenbourg.
- Dahlem, N., & Hahn, A. (2009). *Proceedings of the 2009 Americas Conference on Information Systems, AMCIS*. User-friendly ontology creation methodologies - a survey, (S. Paper 117).
- Davies, A. (2004). *Moving base into high-value integrated solutions: a value stream approach*. *Industrial and Corporate Change*, 13(5), S. 727-756.
- Davies, A., & Brady, T. (2000). *Organisational Capabilities and Learning in Complex Product Systems: Towards Repeatable Solutions*. *Research Policy*, 29(7/8), S. 931-953.
- Davies, A., Brady, T., & Hobday, M. (2006). *Charting a path towards integrated solutions*. *MIT Sloan Management Review*, 47(3), S. 39-48.
- Davis, A. M. (1990). *Software Requirements: Analysis & Specification*. Englewood-Cliffs: Prentice-Hall.
- de Bono, E. (2000). *Six Thinking Hats*. Great Britain: Penguin Books.
- De Nicola, A., Missikoff, M., & Navigli, R. (April 2009). *A software engineering approach to ontology building*. *Information Systems*, 34, S. 258-275.
- Degle, R. (2005). *Ontologie-basierte Beschreibung digitaler Lernmaterialien*. Norderstedt: GRIN Verlag.
- DeMarco, T. (1979). *Structured Analysis and system specification*. New-York: Prentice-Hall.
- Deutsches Institut für Normung e.V. (2009). *DIN PAS 1094*. Berlin: Beuth.

- Dilts, R. B. (2003). *Modeling mit NLP. Das Trainingshandbuch zum NLP-Modeling- Prozess. Paderborn: Junfermann.*
- DIN EN ISO 6385. (2004). *Grundsätze der Ergonomie für die Gestaltung von Arbeitssystemen. Deutsche Fassung EN ISO 6385. Berlin: Beuth Verlag.*
- DIN EN ISO 9241. (2006). *Ergonomie der Mensch-System-Interaktion. Berlin: Beuth Verlag.*
- Djouadi, Y., Redaoui, S., & Amroun, K. (2007). *IEEE international fuzzy systems conference. Mining fuzzy association rules from uncertain data, (S. 1-6). London.*
- Donabedian, A. (1966). *Evaluating the quality of medical care. Milbank Memorial Fund Quarterly, 44(3), S. 166-203.*
- Düsing, R. (2006). *Knowledge Discovery in Databases - Begriff, Forschungsgebiet, Prozess und System. In P. Chameni, & P. Gluchowski, Analytische Informationssysteme - Business Intelligence-Technologien und -Anwendungen (3. Ausg.). Berlin: Springer.*
- Ebert, C. (2009). *Systematisches Requirements Engineering: Anforderungen ermitteln, spezifizieren, analysieren und verwalten. Heidelberg: dpunkt.*
- Engelhardt, W. H., Kleinaltenkamp, M., & Reckenfelderbäumer, M. (1992). *Dienstleistungen als Absatzobjekt. Veröffentlichungen des Instituts für Unternehmensführung und Unternehmensforschung, Arbeitsbericht Nr. 52.*
- Engelhardt, W. H., Kleinaltenkamp, M., & Reckenfelderbäumer, M. (1993). *Leistungsbündel als Absatzobjekte – Ein Ansatz zur Überwindung der Dichotomie von Sach- und Dienstleistungen. Zeitschrift für betriebswirtschaftliche Forschung, 45(5), S. 395-426.*
- Ericson, Å., Müller, P., Larsson, T., & Stark, R. (2009). *Proceedings of the 1st CIRP Industrial Product-Service Systems (IPS2) Conference. Product-Service Systems – From Customer Needs to Requirements in Early Development Phases, (S. 62-67). Cranfield University.*
- Essler, W., Martínez Cruzado, R., & Labude, J. (2001). *Grundzüge der Logik: Das logische Schließen. Frankfurt am Main: Vittorio Klostermann.*
- Ester, M., & Sander, J. (2009). *Knowledge Discovery in Databases: Techniken und Anwendungen. Berlin: Springer.*
- Fährnich, K.-P., & Opitz, M. (2006). *Service Engineering – Entwicklungspfad und Bild einer jungen Disziplin. In H.-J. Bullinger, & A.-W. Scheer, Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen (S. 85-113). Berlin, Heidelberg: Springer.*
- Fernandez, M., & Gomez-Perez, A. (1997). *AAAI-97 Spring Symposium on Ontological Engineering, 24.-26. März 1997. METHONTOLOGY: From Ontological Art Towards Ontological Engineering, (S. 33-40). Stanford University.*

*Franke, H.-J., Grein, G., & Türck, E. (Hrsg.). (2011). Anforderungsmanagement für kundenindividuelle Produkte. Methodische Lösungsansätze - Praxisorientierte Tools - Industrielle Anwendungen. Ergebnisse der BMBF-Projekte DIALOG und KOMSOLV. Karlsruhe: Shaker Verlag.*

*Gebauer, H., Hildenbrand, K., & Fleisch, E. (2006). Servicestrategien für die Industrie. Harvard Business Manager, 27(5), S. 47-55.*

*Gerstner, L. (2002). Who Says Elephants Can't Dance? Inside IBM's Historic Turnaround. New York: HarperCollins Publishers.*

*Gomez-Perez, A., Fernandez-Lopez, M., & Corcho-Garcia, O. (2003). Ontological Engineering - With examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. New York: Springer Verlag.*

*Görz, G., Rollinger, C.-R., & Schneeberger, J. (2003). Handbuch der Künstlichen Intelligenz (4. Ausg.). (G. Görz, C.-R. Rollinger, & J. Schneeberger, Hrsg.) München: Oldenbourg Wissenschaftsverlag.*

*Gräßle, M., Thomas, O., Fellmann, M., & Krumeich, M. (2010). MKWI 2010 - Multikonferenz Wirtschaftsinformatik 2010. Vorgehensmodelle des Product-Service Systems Engineering - Überblick, Klassifikation und Vergleich, (S. 2031 - 2042). Goettingen.*

*Green, P., & DeSarbo, W. (1978). Additive Decomposition of Perceptions Data via Conjoint Analysis. Journal of Consumer Research, 5(1), S. 58-65.*

*Grieble, O., & Scheer, A.-W. (2000). Grundlagen des Benchmarkings öffentlicher Dienstleistungen. (A.-W. Scheer, Hrsg.) Veröffentlichungen des Instituts für Wirtschaftsinformatik, Nr. 166.*

*Gruber, A., Westenthaler, R., & Gahleitner, E. (2006). Proceedings of 6th International Conference on Knowledge Management (I-KNOW '06), 6.-8. September 2006. Supporting Domain Experts in Creating Formal Knowledge Models (Ontologies), (S. 252-260). Graz, Österreich.*

*Gruber, T. (1993). A translation approach to portable ontology specification. Knowledge Acquisition, 5(2), S. 1999-2000.*

*Grüninger, M., & Fox, M. (1995). Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing IJCAI. Methodology for the Design and Evaluation of Ontologies, 95, S. 6.1-6.10. Toronto, Kanada.*

*Guarino, N., & Giaretta, P. (1995). Ontologies and Knowledge Bases: Towards a Terminological Classification. In N. Mars, Towards Very Large Knowledge Bases: Knowledge*

- Building and Knowledge Sharing (KBKS'95), University of Twente, Enschede, Holland (S. 25-32). Amsterdam: IOS Press.
- Guizzardi, G., Herre, H., & Wagner, G. (2002). *First International Conference on Ontologies, Databases and Application of Semantics*. Towards ontological foundations for UML conceptual models, (S. 1100-1117). Irvine, CA, USA.
- Günther, C. (2001). *Das Management industrieller Dienstleistungen: Determinanten, Gestaltung und Erfolgsauswirkungen*. Wiesbaden: Gabler.
- Gutenberg, E. (1983). *Grundlagen der Betriebswirtschaftslehre: Erster Band – Die Produktion (24. Auflage Ausg.)*. Berlin: Springer.
- Haarslev, V., Pai, H., & Shiri, N. (2007). *Proceedings of the 2007 International Workshop on Description Logics (DL-2007)*. Optimizing tableau reasoning in ALC extended with uncertainty, (S. 307-314). Brixen, Italien.
- Habiballa, H. (2007). *Proceedings of the 5th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2007)*. Resolution strategies for fuzzy description logic, (S. 27–36). Ostrava, Tschechien.
- Hagemann, S. (2008). *Maßzahlen für die Assoziationsanalyse im Data Mining*. Hamburg: *Diplomica*.
- Hájek, P. (2005). *Making fuzzy description logics more expressive*. *Fuzzy Sets Systems*, 154(1), S. 1-15.
- Hamilton, R., & Koukova, N. (2008). *Choosing options for products: the effects of mixed bundling on consumers' inferences and choices*. *Journal of the Academy of Marketing Science*, 36(3), S. 423-433.
- Han, J., & Kamber, W. (2012). *Data mining: concepts and techniques (3 Ausg.)*. San Francisco, USA: Morgan Kaufmann.
- Hanna, M., & Newman, W. (2006). *Integrated operations management: a supply chain perspective*. South-Western College Pub; 2 edition.
- Hippner, H., & Wilde, K. (2008). *Data Mining im CRM*. In S. Helmke, U. M.F., & W. Dangelmaier, *Effektives Customer Relationship Management - Instrumente - Einführungskonzepte - Organisation*. Gabler.
- Hitzler, P., Krötzsch, M., Rudolph, S., & Sure, Y. (2008). *Semantic Web - Grundlagen (1 Ausg.)*. Heidelberg: Springer.
- Hoffmann, W. (1999). *Objektorientiertes Qualitätssystem—Referenzmodell und Realisierungsansätze*. PhD Thesis. University of Saarbrücken: Gabler.

Homburg, C., Faßnacht, M., & Günther, C. (2002). *Erfolgreiche Umsetzung dienstleistungsorientierter Strategien von Industriegüterunternehmen*. Zeitschrift für betriebswirtschaftliche Forschung, 54(6), S. 487-508.

Hong, T., Kuo, C., & Wang, S. (2004). *A fuzzy AprioriTid mining algorithm with reduced computational time*. Applied Soft Computing, 5(1), S. 1-10.

Hu, Y., Chen, R., & Tzeng, G. (2003). *Discovering fuzzy association rules using fuzzy partition methods*. Knowledge Based Systems, 16(3), S. 137-147.

Huber, R. J. (1992). *Die Nachfrage nach Dienstleistungen, Volkswirtschaftliche Forschungsergebnisse (Bd. 15)*. Hamburg: Verlag Dr. Kovac.

IBM Deutschland GmbH. (2010). *Global CEO Study: Unternehmensführung in einer komplexen Welt*. Ehningen: IBM Institute for Business Value.

IEEE. (1996). *IEEE Standard for Developing Software Life Cycle Processes: IEEE Std 1074-1995*. New York: IEEE Computer Society.

IEEE. (1998). *IEEE Recommended Practise for Software Requirements Specifications*. New York: The Institute of Electrical and Electronics Engineers.

Jacobson, I., M., C., Jonson, P., & Övergaard, G. (1992). *Object-Oriented Software Engineering - A Use Case Driven Approach*. Wokingham, England: Addison-Wesley.

Jarrar, M. (2006). *Proceedings of the 15th International Conference on World Wide Web (WWW'06)*. Position paper: towards the notion of gloss, and the adoption of linguistic resources in formal ontology engineering (S. 497-503). Edinburgh, Schottland: ACM Press.

Jarrar, M., & Meersman, R. (2002). *Proceedings of the International Conference on Ontologies, Databases and Applications of Semantics (ODBase 02)*. Formal Ontology Engineering in the DOGMA Approach. 2519, S. 1238 - 1254. Springer Verlag.

Jensen, F., Olesen, K., & Andersen, S. (1990). *An algebra of Bayesian belief universes for knowledge-based systems*. Networks, 20(5), S. 637-659.

Jiao, J. (., & Chen, C.-H. (2006). *Customer Requirement Management in Product Development: A Review of Research Issues*. Concurrent Engineering: Research and Application, 14(3), S. 173-185.

Jiao, J., & Chen, C.-H. (2006). *Customer Requirement Management in Product Development: A Review of Research Issues*. Concurrent Engineering: Research and Applications, 14(3), S. 173-185.

Johansson, J., Krishnamurthy, C., & Schlissberg, H. (2003). *Solving the Solutions Problem*. The McKinsey Quarterly, 40(3), S. 116-125.

- John, I., & Dörr, J. (2003). *Extraktion von Produktfamilienanforderungen aus Benutzendokumentation*. IESE-Report Nr. 111.03/D.
- Joint US/EU Agent Markup Language Committee. (27. 3 2001). DAML+OIL language specification. Abgerufen am 16. 10 2011 von <http://www.daml.org/2001/03/daml+oil-index.html>
- Jungk, H. (2007). *Informationsmanagement zur Planung und Verfolgung von Produktlebenszyklen*. Berlin: Fraunhofer IRB Verlag.
- Kano, N., Seraku, N., Takahashi, F., & Tsuji, S. (1984). *Attractive Quality and Must-be Quality*. The Japan Society for Quality Control, 14(2), S. 39–48.
- Kano, N., Tsuji, S., Seraku, N., & Takahashi, F. (1984). *Attractive Quality and Must-be Quality*. Quality- The Journal of the Japanese Society for Quality Control, 14(2), S. 39-44.
- Karlsson, J., Olsson, S., & Ryan, K. (1997). *Improved Practical Support for Large-scale Requirements Prioritising*. Requirements Engineering, 2(1), S. 51-60.
- Kaulio, M. (1998). *Customer, consumer and user involvement in product development: A framework and a review of selected methods*. Total Quality Management, 9(1), S. 141-149.
- Kawakita, J. (1975). *The KJ Method - A Scientific Approach to Problem Solving*. Tokio, Japan: Kawatika Research Institute.
- Kellner, H. (2002). *Kreativität im Projekt*. München: Hanser.
- Knowledge Based Systems, Inc. (KBSI). (1994). *The IDEF5 Ontology Description Capture Method Overview*. Cape Girardeau, Texas: KBSI Report.
- Koch, K. (1999). *Grundprinzipien der Bayes-Statistik*. In F. Krumm, & V. Schwarze, Quo vadis geodesia...? Schriftenreihe der Institute des Studiengangs Geodäsie und Geoinformatik (Bde. Nr. 1999.6-1). Stuttgart: Universität Stuttgart.
- Koch, K. (2000). *Einführung in die Bayes-Statistik*. Heidelberg: Springer Verlag.
- Konstantopoulos, S., & Apostolikas, G. (2007). *Proceedings of the 2007 OTM Confederated international conference on On the move to meaningful internet systems*. Fuzzy-DL reasoning over unknown fuzzy degrees (S. 1312–1318). Springer.
- Kotis, K., & Vouros, G. (07 2006). *Human-Centered Ontology Engineering: the HCOME Methodology*. International Journal of Knowledge and Information Systems (KAIS), 10, S. 109-131.
- Kotonya, G., & Sommerville, I. (1998). *Requirements Engineering: Processes and Techniques*. New York: John Wiley Sons.
- Kovitz, B. L. (1998). *Practical Software Requirements: A Manual of Content and Style*. Manning.

- Kramer, O. (2009). *Fuzzy Logik*. In O. Kramer, O. Günther, W. Karl, W. Lienhart, & K. Zeppenfeld (Hrsg.), *Computational Intelligence - Eine Einführung* (S. 75-99).
- Krishnan, V., & Ulrich, K. (2001). *Product Development Decisions: A Review of the Literature*. *Management Science*, 47(1), S. 1-21.
- Kruse, R., Gebhardt, J., & Klawonn, F. (1995). *Fuzzy-Systeme (2. überarbeitete Auflage Ausg.)*. Stuttgart: Teubner.
- Laarhoven, P. J., & Pedrycz, W. (1983). *A Fuzzy Extension of Saaty's Priority Theory*. *Fuzzy Sets and Systems*(11), S. 229-242.
- Langeard, E. (1991). *Grundfragen des Dienstleistungsmarketing*. *Marketing - Zeitschrift für Forschung und Praxis*(4), S. 233-240.
- Lauritzen, S., & Spiegelhalter, D. (1988). *Local computations with probabilities on graphical structures and their application to expert systems*. *Journal of Royal Statistical Society*, 50(2), S. 157-224.
- Leimeister, J., & Glauner, C. (2008). *Hybride Produkte – Einordnung und Herausforderungen für die Wirtschaftsinformatik*. *Wirtschaftsinformatik*, 50(3), S. 248–251.
- Lenat, D., & Guha, R. (1990). *Building Large Knowledge-based Systems: Representation and Inference in the Cyc Project*. Boston, Massachusetts: Addison-Wesley.
- Lepar, V., & Shenoy, P. (1999). *A Comparison of Lauritzen-Spiegelhalter, Hugin, and Shenoy-Shafer Architectures for Computing Marginals of Probability Distributions*. (G. Cooper, & S. Moral, Hrsg.) *Uncertainty in Artificial Intelligence*, 14, S. 328-337.
- Lukasiewicz, T., & Straccia, U. (2008). *Managing uncertainty and vagueness in description logics for the Semantic Web*. *International Journal on Web Semantics: Science, Services and Agents on the World Wide Web*(6), S. 291-308.
- Mayhew, D. (1999). *The Usability Engineering Lifecycle*. *Practitioner's Handbook for User Interface Design*. San Francisco: Morgan Kaufmann.
- McAloone, T., & Andreasen, M. (2004). *Proceedings of the 8th International Design Conference – DESIGN 2004*. In M. D. (Hrsg.), *Design for Utility, sustainability and societal virtues: Developing Product Service Systems*, (S. 1545 - 1552). Dubrovnik (Kroatien).
- McKinsey Corporation. (2005). *The McKinsey Global Survey of Business Executives*. *The McKinsey Quarterly*, 42(1), 58-65.
- Meffert, H., & Bruhn, M. (2009). *Dienstleistungsmarketing. Grundlagen - Konzepte - Methoden (6. Ausg.)*. Wiesbaden: Gabler.

- Meier, H., & Uhlmann, E. (2012). *Hybride Leistungsbündel - Ein neues Produktverständnis*. In H. Meier, & E. Uhlmann, *Integrierte Industrielle Sach- und Dienstleistungen* (S. 1-22). Berlin Heidelberg: Springer.
- Meier, H., Uhlmann, E., & Kortmann, D. (2005). *Hybride Leistungsbündel, Nutzenorientiertes Produktverständnis durch interferierende Sach- und Dienstleistungen*. *wt Werkstatttechnik Online*, 95(7/8), S. 528–532.
- Meiren, T., & Barth, T. (2002). *Service Engineering in Unternehmen umsetzen – Leitfaden für die Entwicklung von Dienstleistungen*. Stuttgart: Fraunhofer IRB Verlag.
- Mertens, P. (2001). *Integrierte Informationsverarbeitung 1—Operative Systeme in der Industrie* (13. Ausg.). Wiesbaden: Gabler.
- Meyer, A., & Blümelhuber, C. (1994). *Interdependenzen zwischen Absatz und Produktion in Dienstleistungsunternehmen und ihre Auswirkungen auf konzeptionelle Fragen des Absatzmarketing*. In H. Corsten, & W. Hilke, *Dienstleistungsproduktion* (Bd. 42, S. 5-42). Wiesbaden: Gabler.
- Mont, O. (2002). *Journal of Cleaner Production*. Clarifying the concept of product-service systems, 10, S. 237-245.
- Müller, J. (2011). *Strukturbasierte Verifikation von BPMN-Modellen*. Wiesbaden: Vieweg+Teubner.
- Müller, P., & Blessing, L. (2007). *Entwicklungsprozesse hybrider Leistungsbündel - Erweiterter Handlungsraum für den HLB-Entwickler*. *wt Werkstatttechnik online*(7/8), S. 516-521.
- Müller, P., & Stark, R. (2010). *A Generic PSS Development Process Model Based on Theory and an Empirical Study*. In M. D., S. M., P. N., & B. N., *Proceedings of the 11th International Design Conference - DESIGN 2010* (S. 361-370). Dubrovnik: The Design Society.
- Nagamachi, M. (1989). *Kansei Engineering*. Tokyo: Kaibundo Publisher.
- Nagamachi, M. (2011). *Kansei/affective engineering*. Boca Raton, USA: CRC Press.
- Nauck, D., Borgelt, C., Klawonn, F., & Kruse, R. (2003). *Neuro-Fuzzy-Systeme: Von den Grundlagen künstlicher Neuroner Netze zur Kopplung mit Fuzzy-Systemen* (3. Auflage Ausg.). Wiesbaden: Vieweg+Teubner Verlag.
- Neches, R., Fikes, R., Finin, T., Gruber, T., Senator, T., & Swartout, W. (1991). *Enabling technology for knowledge sharing*. *AI Magazine*, 12(3), S. 36-56.
- Neubach, M. (2010). *Wissensbasierte Rückführung von Produktnutzungsinformationen in die Produktentwicklung im Rahmen einer Product Lifecycle Management (PLM)-Lösung*. Bochum: Shaker Verlag.

- Nguyen, H., Müller, P., & Stark, R. (2013). *Transformation towards an IPS2 business: A deployment approach for process-based PSS development projects*. In Y. Shimonura, & K. Kimita, *The Philosopher's Stone for Sustainability* (S. 251-256). Berlin Heidelberg: Springer.
- Noy, N., & McGuinness, D. (2001). *Ontology Development 101: A Guide to Creating Your First Ontology*. Knowledge Systems, AI Laboratory: Stanford University.
- Nüttgens, M., & Rump, F. J. (2002). *Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK)*. In J. Desel, & M. Weske (Hrsg.), *Promise 2002 - Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen*. 21, S. 64-77. Bonn: LNI.
- Nüttgens, M., Heckmann, M., & Luzius, M. J. (1998). *Service Engineering Rahmenkonzept*. IM – Fachzeitschrift für Information, Management und Consulting(13), S. 14-19.
- Ovtcharova, J. (1997). *A Framework for Feature-Based Product Design - Fundamental Principles, System concepts, Applications*. Düsseldorf: VDI Verlag.
- Ovtcharova, J., & Burger, A. (Mai 2011). *Marktstudie und Reifegradmodelle für den Wandel zum Lösungsanbieter*. Berliner Kreis News(16), S. 14.
- Ovtcharova, J., Burger, A., & Kilger, C. (Februar 2012). *Vom Produzenten zum Lösungsanbieter - Eine empirische Studie deckt unternehmerische Handlungsbedarfe auf*. WiGeP News(1), S. 2-3.
- Pearl, J. (1996). *Fusion, propagation and structuring in belief networks*. Artificial Intelligence(29), S. 241-288.
- Petersohn, H. (2005). *Data Mining - Verfahren, Prozesse, Anwendungsarchitektur*. Oldenbourg Wissenschaftsverlag: München.
- Pinto, H., Tempich, C., Staab, S., & Sure, Y. (2004). *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004), 22.-27. August*. In R. De Mántaras, & L. Saitta (Hrsg.), *Diligent: Towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies* (S. 393–397). Valencia, Spanien: IOS Press.
- Pohl, K. (1994). *The Three Dimensions of Requirements Engineering: A Framework and its Applications* (Bd. 19).
- Pohl, K. (2007). *Requirements Engineering: Grundlagen, Prinzipien, Techniken*. Heidelberg: dpunkt.
- Pohl, K., & Rupp, C. (2009). *Basiswissen Requirements Engineering: Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level*. dpunkt Verlag.

- Pohl, K., & Rupp, C. (2009). *Basiswissen Requirements Engineering: Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level*. Heidelberg: dpunkt.
- Pohl, K., Böckle, G., & van der Linden, F. (2005). *Software Product Line Engineering - Foundations, Principles, and Techniques*. Berlin, Heidelberg, New York: Springer-Verlag.
- Rai, A., & Sambamurthy, V. (December 2006). *The Growth of Interest in Services Management: Opportunities for Information Systems Scholars*. *Information Systems Research*, 17(4), S. 327–331.
- Regnell, B., Runeson, P., & Thelin, T. (2000). *Are the Perspectives Really Different? - Further Experimentation on Scenario-Based Reading of Requirements*. *Empirical Software Engineering*, 5(4), 331-356.
- Rexfelt, O., Ornäs, A., & V.H. (2008). *Consumer acceptance of product-service systems - Relative advantages and uncertainty reductions*. In O. Rexfelt (Hrsg.), *User-centred design and technology-mediated services identifying and adressing challenges by analysing activities*. Göteborg: Chalmers University of Technology.
- Robertson, J., & Robertson, S. (2010). *Volere: Requirements Ressources*. Abgerufen am 19. 12 2011 von <http://www.volere.co.uk/template.htm>
- Roland Berger Strategy Consultants. (2008). *Maschinen- und Anlagenbau im Abschwung*. Berlin.
- Runkler, T. (2009). *Data Mining: Methoden und Algorithmen intelligenter Datenanalyse*. Wiesbaden: Vieweg+Teubner.
- Rupp, C. (2004). *Requirements-Engineering und -Management (3 Ausg.)*. Carl Hanser Verlag.
- Rupp, C., & SOPHISTen. (2009). *Requirements-Engineering und -Management - Professionelle, iterative Anforderungsanalyse für die Praxis*. Heidelberg: Hanser-Verlag.
- Rzehorz, C. (1998). *Wissensbasierte Anforderungsentwicklung auf der Basis eines integrierten Produktmodells*. Aachen: Shaker Verlag.
- Saaty, T. (1980). *The Analytic Hierarchy Process*. New York: McGraw Hill Higher Education.
- Sachs, J., Helbig, M., Herrmann, R., Welge, R., & Bollow, E. (2010). *Ambient Assisted Living - AAL - 3. Deutscher AAL-Kongress*. Merkmalsextraktion und semantische Integration von Ultrabreitband-Sensoren zur Erkennung von Notfällen. Berlin: VDE Verlag.
- Sakao, T., Panshef, V., & Dörsam, E. (2009). *Addressing Uncertainty of PSS for Value-Chain Oriented Service Development*. In T. Sakao, & M. Lindahl, *Introduction to Product/Service-System Design (S. 137-156)*. Londong: Springer.

- Sanchez, D., & Tettamanzi, A. (2004). *Generalizing quantification in fuzzy description logics*. In B. Reusch, Computational Intelligence, Theory and Applications - International Conference 8th Fuzzy Days in Dortmund (S. 81-88). Dortmund: Springer.
- Sarodnick, F., & Brau, H. (2006). *Methoden der Usability Evaluation - Wissenschaftliche Grundlagen und praktische Anwendung (1. Ausg.)*. Bern: Hans Huber Verlag.
- Säuberlich, F. (2000). *KDD und Data Mining als Hilfsmittel zur Entscheidungsunterstützung*. Frankfurt: Peter Lang Verlag.
- Sawhney, M. (2006). *Going Beyond the Product: Defining, Designing and Delivering Customer Solutions*. In R. Lusch, & S. Vargo, *Toward a Service-Dominant Logic of Marketing: Dialogue, Debate, and Directions* (S. 365-380). New York: Armonk.
- Schawel, C., & Billing, F. (2009). *Top 100 Management Tools: Das wichtigste Buch eines Managers*. Wiesbaden: Gabler.
- Scheer, A., Griebler, O., & Klein, R. (2006). *Modellbasiertes Dienstleistungsmanagement*. In H.-J. Bullinger, & A.-W. Scheer, *Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen* (S. 19-51). Berlin Heidelberg New York: Springer.
- Scheer, C. (2006). *Kundenorientierter Produktkonfigurator: Erweiterung des Produktkonfigurator-Konzeptes zur Vermeidung kundeninitiiertem Prozessabbrüche bei Präferenzlosigkeit und Sonderwünschen in der Produktspezifikation*. Berlin: Logos-Verlag.
- Schildheuer, G. (1998). *Konzeption eines objektorientierten Referenzmodells zur Planung und Gestaltung eines umfassenden Qualitätsinformationssystems*. PhD Thesis. University of Bochum.
- Schmitz, G. (2008). *Multikonferenz Wirtschaftsinformatik 2008*. In M. Bichler, T. Hess, H. Krcmar, U. Lechner, F. Matthes, A. Picot, . . . P. Wolf (Hrsg.), *Der wahrgenommene Wert hybrider Produkte: Konzeptionelle Grundlagen und Komponenten* (S. 665-683). Berlin: GITO-Verlag.
- Schreiber, G., Wielinga, B., de Hoog, R., Akkermans, H., & Van De Velde, W. (Dezember 1994). *CommonKADS: A Comprehensive Methodology for KBS Development*. IEEE Expert: Intelligent Systems and Their Applications, 9(6), S. 28—37.
- Schubert, V., Rogalski, S., & Ovtcharova, J. (2011). *Feedback-unterstützte Harmonisierung der Kunden- und Herstellersicht*. In H.-J. Franke, G. Grein, & E. Türck (Hrsg.), *Anforderungsmanagement für kundenindividuelle Produkte. Methodische Lösungsansätze - Praxisorientierte Tools - Industrielle Anwendungen. Ergebnisse der BMBF-Projekte DIALOG und KOMSOLV* (S. 33-44). Karlsruhe: Shaker Verlag.

- Schubert, V., Wicaksono, H., & Rogalski, S. (2011). A Product Life-Cycle Oriented Approach for Knowledge-Based Product Configuration Systems. In A. Bernard (Hrsg.), Global Product Development. Proceedings of the 20th CIRP Design Conference (S. 357-362). Heidelberg, London, New York: Springer Verlag.*
- Schulte, S. (2007). Integration von Kundenfeedback in die Produktentwicklung zur Optimierung der Kundenzufriedenheit. Bochum: Shaker Verlag.*
- Schweitzer, M. (1993). Produktion. In W. Wittmann (Hrsg.), Enzyklopädie der Betriebswirtschaftslehre (Bd. 2: Handwörterbuch der Betriebswirtschaft. 5. Auflage, S. 3328–3347). Stuttgart: Schäffer-Poeschel.*
- Seegy, U. (2009). Dienstleistungskompetenz im Maschinen- und Anlagenbau - Eine Untersuchung wesentlicher Handlungspotenziale und ihrer Auswirkungen. Wiesbaden: Gabler.*
- Shenoy, P., & Shafer, G. (1986). Propagating belief functions using local computation. IEEE Expert, 1(3), S. 43-52.*
- Sommerville, I., & Sawyer, P. (1997). Requirements Engineering: A Good Practise Guide. New York: Wiley.*
- Son, S. (2002). Einsatz von Bayesschen Netzwerken zur Simulation von Unternehmensentscheidungen. Frankfurt am Main: Johann Wolfgang Goethe-Universität.*
- Stark, R., & Müller, P. (2012). HLB-Entwicklungsmethodik – generischer Entwicklungsprozess, Generierung von Anforderungen und Absicherung hybrider Leistungsbündel. In H. Meier, & E. Uhlmann, Integrierte Industrielle Sach- und Dienstleistungen (S. 37-60). Berlin Heidelberg: Springer.*
- Stauss, B., & Bruhn, M. (2007). Hybride Produkte: Merkmale und Herausforderungen. In T. Böhm, & H. Krcmar, Wertschöpfungsprozesse bei Dienstleistungen (S. 239–255). Wiesbaden: Gabler.*
- Steinbach, M. (2005). Systematische Gestaltung von Product-Service-Systems. Phd Thesis. Saarbrücken: Universität des Saarlandes.*
- Steinbach, M., Botta, C., & Weber, C. (2005). Integrierte Entwicklung von Product-Service Systems. werkstatttechnik online, 95(7/8), S. 546-553.*
- Steve, G., & Gangemi, A. (1996). Proceedings of the 10th Knowledge Acquisition Workshop - KAW'96, November 9-14th. ONIONS Methodology and the Ontological Commitment of Medical Ontology ON8.5. Banff, Canada.*
- Stoilos, G., Simou, N., Stamou, G., & Kollias, S. (2006). Uncertainty and the Semantic Web. IEEE Intelligent Systems, 21(5), S. 84–87.*

Straccia, U. (1998). *Proceedings AAAI 1998. A fuzzy description logic (S. 594–599)*. Madison, USA: AAAI Press / MIT Press .

Stuckenschmidt, H. (2011). *Ontologien - Konzepte, Technologien und Anwendungen*. Heidelberg: Springer.

Studer, R., Benjamins, V., & Fensel, D. (1998). *Knowledge Engineering: Principles and Methods 25(1-2)*. IEEE Transactions on Data and Knowledge Engineering, S. 161-197.

Sturm, F., & Bading, A. (2008). *Capital goods manufacturers as solutions providers—survey on the state of change*. Wirtschaftsinformatik, 50(3), S. 174-186.

Sure, Y., Staab, S., & Studer, R. (Dezember 2002). *Methodology for Development and Employment of Ontology Based Knowledge Management Applications*. SIGMOD Record, 31, S. 18-23.

Swartout, B., Ramesh, P., Knight, K., & Russ, T. (1997). *AAAI Symposium on Ontological Engineering, AAAI-97 Spring Symposium Series. Toward Distributed Use of Large-Scale Ontologies, (S. 138-148)*. Banff, California.

Tempich, C., Pinto, H., & Sure, Y. V. (2005). *Evaluating DILIGENT Ontology Engineering in a Legal Case Study*. In P. Casanovas, P. Noriega, D. Bourcier, & V. Benjamins, IVR 22nd World Congress - Law and Justice in a Global Society (S. 330-337). Granada, Spanien: University of Granada.

Tempich, C., Pinto, H., Sure, Y., & Staab, S. (2005). *An Argumentation Ontology for Distributed, Loosely-controlled and evolving Engineering processes of ontologies (DILIGENT)*. In A. Gómez-Pérez, & J. Euzenat, The Semantic Web: Research and Applications, Second European Semantic Web Conference, ESWC 2005, 29. Mai - . Juni (S. 241-256). Kreta, Griechenland.

Thomas, O., & Nüttgens, M. (2009). *Dienstleistungsmodellierung - Methoden, Werkzeuge und Branchenlösungen*. Heidelberg: Springer Verlag.

Thomas, O., & Scheer, W. (2003). *Referenzmodell-basiertes (Reverse-) Customizing von Dienstleistungsinformationssystemen*. Saarbrücken: Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI).

Thomas, O., Walter, P., & Loos, P. (2008). *Product-Service Systems: Konstruktion und Anwendung einer Entwicklungsmethodik*. Wirtschaftsinformatik, 50(3), S. 208-219.

Thomas, O., Walter, P., & Loos, P. (2010). *Konstruktion und Anwendung einer Entwicklungsmethodik für Product-Service Systems* . In O. Thomas, P. Loos, & M. Nüttgens, Hybride Wertschöpfung - Mobile Anwendungssysteme für effiziente Dienstleistungsprozesse im technischen Kundendienst (S. 61-81). Heidelberg Dordrecht London New York: Springer.

- Tresp, C., & Molitor, R. (1998). *Proceedings ECAI 1998*. A description logic for vague knowledge (S. 361–365). Brighton, UK: Wiley & Sons.
- Tu, J. (November 1996). *Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes*. *Journal of Clinical Epidemiology*, 49(11), S. 1225-1231.
- Tuli, K., Kohli, A., & Bharadwaj, S. (2007). *Rethinking Customer Solutions: From Product Bundles to Relational Processes*. *Journal of Marketing*, 71(3), S. 1-17.
- Tzikas, D., & Likas, A. (2010). *Artificial Neural Networks - ICANN 2010*. An incremental Bayesian Approach for Training Multilayer Perceptrons (S. 87-96). Thessaloniki, Griechenland: Springer.
- Ulrich, H. (1981). *Die Betriebswirtschaftslehre als anwendungsorientierte Sozialwissenschaft*. In N. Geist, & R. (. Köhler, *Die Führung des Betriebes*. Stuttgart: C.E Pöschel Verlag.
- Ulrich, K., & Eppinger, S. (2008). *Product Design and Development*. Boston, USA: McGraw-Hill Inc.
- Uschold, M., & Grüninger, M. (1996). *Ontologies: principles, methods and applications*. *The Knowledge Engineering Review*, 11(2), S. 93-136.
- Uschold, M., & King, M. (1995). *IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing*. In D. Skuce (Hrsg.), *Towards a Methodology for Building Ontologies*, (S. 6.1-6.10). Montreal, Kanada.
- van der Spek, R., & Spijkervet, A. (2005). *Knowledge Management: Dealing Intelligently with Knowledge*. Utrecht: The Knowledge Management Network.
- Van Laarhoven, P., & Pedrycz, W. (1983). *Fuzzy Extension of Saaty's Priority Theory*. *Fuzzy Sets and Systems*, 11(3), S. 229–242.
- Verein Deutscher Ingenieure. (1993). *Richtlinie 2221 - Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte*. Düsseldorf: VDI.
- Verein Deutscher Ingenieure. (2004). *Richtlinie 2206 - Entwicklungsmethodik für mechatronische Systeme*. Düsseldorf: VDI.
- Volkert, S. (2004). *Wissensrepräsentation in Customer Relationship Management-Anwendungssystemen und ökonomischen Analysen*. Hamburg: Verlag Dr. Kovac.
- Vrandečić, D., Pinto, S., Tempich, C., & Sure, Y. (2005). *The DILIGENT Knowledge Processes*. *Journal of Knowledge Management*, 9(5), S. 85-96.

Wallace, D. (2007). Knowledge Management: Historical and Cross-Disciplinary Themes. London: Libraries Unlimited Inc.

Wandke, H., Oed, R., Metzker, E., Ballegooy, M. v., & Nitschke, J. (2001). Die Entwicklung von User Interfaces als arbeitswissenschaftlicher Prozess und seine Unerstützung durch Software-Werkzeuge. Zeitschrift für Arbeitswissenschaft, 55(2), S. 79-93.

Wanke, S. (2010). Neue Konzepte zur Verwaltung und Bereitstellung von Lösungen im Produktentwicklungsprozess - CPM/PDD-Lösungsmuster als Grundlage eines verhaltensbeschreibenden Lösungskataloges (Bd. 48). Saarbrücken: Universität des Saarlandes.

Weber, C., Steinbach, M., & Botta, C. (2004). International Design Conference - Design 2004. Modelling of Product-Service Systems (PSS) Based on the PDD Approach, (S. 547-554). Dubrovnik,.

Weng, C., & Chen, Y. (2010). Mining fuzzy association rules from uncertain data. Knowledge and Information Systems, 23(2), S. 129-152.

Wieggers, K. E. (2005). Software-Requirements. Redmond, Washington: Microsoft Press.

Wielinga, B. S., Jansweijer, W., Anjewierden, A., & Vam Harmelen, F. (1994). "Framework and Formalism for Expressing Ontologies", KACTUS Project Deliverable DO1b.1. Amsterdam: University of Amsterdam.

Windahl, C., Andersson, P., Berggren, C., & Nehler, C. (2004). Manufacturing Firms and Integrated Solutions: Characteristics and Implications. European Journal of Innovation Management, 7(3), S. 218-228.

Witte, R. (2002). Architektur von Fuzzy-Informationssystemen zur Repräsentation und Verarbeitung unscharfer Daten. Karlsruhe: Universitätsbibliothek Karlsruhe.

Wittig, F. (2002). Maschinelles Lernen Bayes'scher Netze in benutzeradaptiven Systemen. Saarbrücken: Universität Saarbrücken - Fakultät 6 - Naturwissenschaftlich-Technische Fakultät.

Wöhe, G. (2010). Einführung in die allgemeine Betriebswirtschaftslehre. München: Vahlen.

Woiseschläger, D., Backhaus, C., & Michaelis, M. (2009). Solution Selling: Wie der Wandel zum Lösungsanbieter gelingt. In M. Borchert, E. Heinen, & Z.-R. K., Systematische Gestaltung von Leistungen und Prozessen in KMU - Voraussetzung für erfolgreiche Internationalisierung von Dienstleistungen (S. 57-78). Ingolstadt: Verlag Dr. Jochem Heizmann.

World Wide Web Consortium. (10. 2 2004). OWL Web Ontology Language Overview. Abgerufen am 16. 10 2011 von <http://www.w3.org/TR/owl-features/>

---

*World Wide Web Consortium. (27. 10 2009). W3C Recommendation. Abgerufen am 16. 10 2011 von OWL 2 Web Ontology Language Document Overview: <http://www.w3.org/TR/owl2-overview/>*

*Yen, J. (1991). Proceedings IJCAI 1991. Generalizing term subsumption languages to fuzzy logic (S. 472-477). Sydney: Morgan Kaufmann.*

*Zadeh, L. (1965). Fuzzy sets. Information and Control, 8, S. 338–353.*

*Zehnter, C., Burger, A., & Ovtcharova, J. (2012). Key-Performance-Analyse von Methoden des Anforderungsmanagements. Karlsruhe: KIT Scientific Publishing.*

*Zerbst, S. (2002). 3D Spieleprogrammierung mit DirectX in C/C++ (Bd. 2). Braunschweig: Books on Demand GmbH.*

*Zöller-Greer, P. (2010). Künstliche Intelligenz - Grundlagen und Anwendungen (2. Ausg.). Wächtersbach: Composita Verlag.*