

Masterthesis

**Untersuchung der Anforderungen generischer Datenmodelle
für den Einsatz mit Supply-Chain-Daten**

Zur Erlangung des akademischen Grades Master of Science (M.Sc.)
Wirtschaftsingenieurwesen

Kevin Pattara Bohn (191646)

Ausgegeben am 10.03.2019

Abgegeben am 09.09.2019

Betreuer:

Prof. Dr.-Ing. Markus Rabe

M. Sc. Astrid Klüter

I Inhaltsverzeichnis

I	Inhaltsverzeichnis	II
II	Abbildungsverzeichnis	IV
III	Tabellenverzeichnis	VI
IV	Abkürzungsverzeichnis	VII
V	Formelverzeichnis	VIII
1	Einleitung.....	1
2	Stand der Technik	3
2.1	Daten im Supply-Chain-Kontext	3
2.1.1	Unterscheidung von Zeichen, Daten, Information und Wissen.....	3
2.1.2	Relevante Supply-Chain-Daten	5
2.2	Referenzmodelle	13
2.2.1	Supply-Chain-Operations-Reference-Modell	13
2.2.2	Informationen im Supply-Chain-Netzwerk.....	15
2.2.3	Vor- und Nachteile eines Referenzmodells.....	16
2.3	Grundlagen des Datenbankentwurfs	17
2.4	Datenbankmodelle	20
2.4.1	Implementative Datenbankmodelle.....	21
2.4.2	Konzeptionelle Datenmodelle	24
2.5	Generisches Datenmodell	36
2.5.1	Rollenkonzepte.....	39
2.5.2	Powertyp-Konzept	45
2.5.3	Strukturschema des Synthesekonzepts.....	47
2.6	Disziplinen der Modellverständlichkeit.....	53
3	Anforderungserhebung und -analyse im Supply-Chain-Umfeld.....	57
3.1	Unterschiede zwischen einem generischen und spezifischen Datenmodell.....	58
3.1.1	Nachteile generischer und spezifischer Datenmodelle.....	59
3.1.2	Vorteile eines generischen und spezifischen Datenmodells.....	61
3.1.3	Allgemeine Anforderungen an ein generisches Datenmodell.....	62
3.2	Anforderungen an das generische Datenmodell.....	63
3.2.1	Entwurfskonzept des generischen Datenmodells.....	64
3.2.2	Sicherstellung der Modellverständlichkeit in Supply Chains.....	67
3.2.3	Prozessschnittstellen in Supply Chains	68

3.2.4	Entwicklung generischer Fachbegriffe	70
3.2.5	Zusammenfassung der Anforderungen.....	75
3.3	Empfehlung und Darstellung eines generischen Datenmodells	79
3.4	Datenorientiertes Kompendium	86
4	Zusammenfassung und Ausblick.....	91
5	Anhang.....	IX
	Literaturverzeichnis	XIX
	Eidesstattliche Versicherung	XXIV

II Abbildungsverzeichnis

Abbildung 2-1: Begriffspyramide [vgl. Bodendorf 2006, S. 1]	4
Abbildung 2-2: Ziele des SCM [vgl. Wellbrock 2015, S. 46]	6
Abbildung 2-3: Modell einer SC [vgl. Sommer 2007, S. 30]	7
Abbildung 2-4: Kernprozesse des SCOR-Modells [vgl. Stölzle und Halsband 2005, S. 14]	15
Abbildung 2-5: Material- und Informationsfluss in einer polyzentrischen SC [vgl. Werner 2017, S. 28].....	16
Abbildung 2-6: Drei-Ebenen-Modell [vgl. Herrmann 2018, S. 6].....	18
Abbildung 2-7: Phasenkonzept des Datenbankentwurfs [vgl. Herrmann 2018, S. 8]	19
Abbildung 2-8: Hierarchisches Datenbankmodell [vgl. Stahlknecht und Hasenkamp 2005, S. 170]	22
Abbildung 2-9: Netzwerk Datenbankmodell [vgl. Stahlknecht und Hasenkamp 2005, S. 172]	22
Abbildung 2-10: Relationales Datenbankmodell [vgl. Geisler 2009, S. 63].....	23
Abbildung 2-11: Objektorientiertes Datenbankmodell [vgl. Geisler 2009, S. 74]	24
Abbildung 2-12: Darstellung eines Beziehungstyps im ERM [vgl. Jarosch 2016, S. 48f; van Randen et al. 2016, S. 10].....	28
Abbildung 2-13: Kardinalität nach Chen im ERM [vgl. Gadatsch 2017, S. 11]	29
Abbildung 2-14: MIN, MAX-Notation im ERM [vgl. Gadatsch 2017, S. 34].....	29
Abbildung 2-15: Aufbau einer Vererbung [vgl. Gadatsch 2017, S. 19; Schneider 2007a, S. 254]	31
Abbildung 2-16: Aggregation einer Beziehung und Entitätstypen [vgl. Schwarze 2000, S. 231]	32
Abbildung 2-17: Notation im ERD [vgl. Gadatsch 2017, S. 10-26]	33
Abbildung 2-18: Darstellung des UML im Klassendiagramms [vgl. van Randen et al. 2016, S. 5-17]	35
Abbildung 2-19: Qualitätskriterien für die Anforderungen zur Konstruktion eines generischen Datenmodells [vgl. Schneider 2007a, S. 412]	36
Abbildung 2-20: Zusammenhänge der Konzepte für die generische Darstellung [vgl. Schneider 2007a, S. 418]	39
Abbildung 2-21: Einfache Rollendarstellung eines Entitätstyps [vgl. Schneider 2007a, S. 258]	40
Abbildung 2-22: Seperate Role Type [vgl. Fowler 1997, S. 4f].....	42
Abbildung 2-23: Role Subtype [vgl. Fowler 1997, S. 6].....	43
Abbildung 2-24: Role Object [vgl. Fowler 1997, S. 14].....	43
Abbildung 2-25: Role Relationship [vgl. Fowler 1997, S. 17]	44
Abbildung 2-26: Anwendungsfälle der Rollenkonzepte [vgl. Fowler 1997, S. 4-17].....	45
Abbildung 2-27: Beziehung eines Powertypen zu einem konkreten Objekttyp [vgl. Schneider 2007a, S. 428]	47
Abbildung 2-28: Beispielhafte Aufbauorganisation eines Unternehmens [vgl. Schneider 2007a, S. 451]	49
Abbildung 2-29: Hierarchischer Aufbau der Fachbegriffe des veränderlichen Verhaltensmuster [vgl. Schneider 2007a, S. 454]	49

Abbildung 2-30: Generisches Datenmodell einer Aufbauorganisation im Konzern [vgl. Schneider 2007a, S. 456]	50
Abbildung 2-31: Kombinationsmöglichkeiten von Organisations-Rollen-Paaren im generischen Datenmodell [vgl. Schneider 2007a, S. 458]	52
Abbildung 2-32: Integrationsfähigkeit des generischen Datenmodells [vgl. Schneider 2007a, S. 462f]	53
Abbildung 3-1: Unterschied zwischen einer generischen und spezifischen Modellbeschreibung.....	58
Abbildung 3-2: Interpretationsspielraum durch spezifische und generische Begriffe	60
Abbildung 3-3: Beispielhafte Teildatenmodelle in Supply Chains.....	66
Abbildung 3-4: Gegenüberstellung der Fachbegriffe verschiedener Kernprozesse	67
Abbildung 3-5: Identifizierende Fachbegriffe im Teildatenmodell	68
Abbildung 3-6: Einflüsse an den Schnittstellen der SC	70
Abbildung 3-7: Hierarchische Anordnung von Fachbegriffen durch die Mengenlehre.....	72
Abbildung 3-8: Beispielhafter Aufbau einer hierarchischen Struktur für generische Begriffe	72
Abbildung 3-9: Die Fachbegriffsmodellierung als Regelkreis	73
Abbildung 3-10: Hierarchischer Aufbau der SC-Begriffe im Rahmen der SC- Netzwerkbeziehung	82
Abbildung 3-11: Generisches Datenmodell für die Netzwerkbeziehung in der Supply Chain	83
Abbildung 3-12: Generisches Datenmodell für die Adresse der SC-Akteure	84
Abbildung 3-13: Zusammenführen von generischen Supply-Chain-Prozessen.....	85
Abbildung 3-14: Das datenorientierte Kompendium als Basis eines generischen Datenmodells.....	87

III Tabellenverzeichnis

Tabelle 2-1:	Stamm- und Bewegungsdaten [vgl. Schuh und Stich 2012, S. 73-79]	8
Tabelle 2-2:	Zusammenfassung der wesentlichen SC-Daten [vgl. Schuh und Stich 2012, S. 73-79; Alicke 2005, S. 85; Kurbel 2016, S. 443-452; Corsten und Gössinger 2008, S. 215-216; Nebl und Corsten 2011, S. 8]	12
Tabelle 2-3:	Entwicklungsstufen des SCOR-Modells [vgl. Werner 2017, S. 71]	13
Tabelle 2-4:	Beispielhafte Darstellung von Attribut und Attributwert [vgl. Jarosch 2016, S. 34]	26
Tabelle 2-5:	Kardinalitäten nach Chen [vgl. Gadatsch 2017, S. 11f; Abts und Mülder 2011, S. 146f]	28
Tabelle 2-6:	Grundtypen der MIN, MAX-Notation [Schneider 2007a, S. 248]	30
Tabelle 2-7:	Kategorisierung zwischen Datenbank- und Datenmodellen	35
Tabelle 2-8:	Kategorisierung der Fachbegriffe nach ihrer Wesensart und Verhaltensmuster [vgl. Schneider 2007a, S. 453]	48
Tabelle 3-1:	Nachteile generischer und spezifischer Datenmodelle	61
Tabelle 3-2:	Vorteile generischer und spezifischer Datenmodelle	62
Tabelle 3-3:	Übersicht der vorläufigen Klassenbezeichnung	80
Tabelle 3-4:	Zuordnung der wesentlichen Fachbegriffe in den Fachbereichen	81
Tabelle 3-5:	SC-Fachbegriffe kategorisiert nach ihrer unveränderlichen Wesensart und ihrem veränderlichen Verhaltensmuster	82
Tabelle 3-6:	Beispielhafte Darstellung eines datenorientierten Kompendiums für ein generisches Datenmodell im Umgang mit SC-Daten	90
Tabelle 5-1:	Zielerreichungsgrad bestehender SCM-Konzepte [vgl. Wellbrock 2015, S. 51]	XVIII

IV Abkürzungsverzeichnis

ATP	Available-to-promise
ANSI	American National Standards Institute
BZW.	Beziehungsweise
CPS	Cyberphysische Systeme
CTP	Capable-to-promise
DB	Datenbank
DBS	Datenbanksystem
DBMS	Datenbankmanagementsystem
DDL	Data Definition Language
DML	Data Manipulation Language
DQL	Data Query Language
DSDL	Data Storage Description Language
EPK	Ereignisgesteuerte Prozesskette
ET	Entitätstyp
ERM	Entity Relation Model
ERP	Enterprise Ressource Planning
IS	Informationssystem
ISe	Informationssysteme
PPS	Produktionsplanung- und Steuerung
SC	Supply Chain
SCs	Supply Chains
SCM	Supply Chain Management
SCOR	Supply-Chain-Operations-Reference-Modell
SPARC	Standards Planning and Requierements Committee
SQL	Structured Query Language
TDM-L	Teildatenmodell-Liefern
TDM-H	Teildatenmodell-Herstellen
UML	Unified Modeling Language
VGL.	Vergleich

V Formelverzeichnis

(1)	$= \{B \subset A \wedge \neg(A \subset B) \wedge A \neq B\}$	70
-----	--	----

1 Einleitung

Infolge der Globalisierung tragen die stark wachsenden Absatz- und Beschaffungsmärkte zu einem Anstieg der Wettbewerbsintensität bei, weshalb im logistischen Netzwerk die Leistungserstellung zunehmend in einer flexiblen Arbeitsteilung sowie einer stärkeren Kooperation der Geschäftspartner erfolgt [vgl. Wellbrock 2015, S. 19]. Diese Netzwerke sind unter dem Begriff Supply Chain (SC, Plural: SCs) bekannt und elementarer Bestandteil der heutigen Prozesslandschaft [vgl. Scheidler 2017, S. 1]. Die intensive Kooperation der SC-Partner erfordert einen hohen Informationsaustausch und wird maßgeblich durch den effizienten Einsatz der Ressource Daten bestimmt, die durch Informationssysteme (IS, Plural: ISe) bereitgestellt werden [vgl. Mielke et al. 2018, S. 270; Moreira et al. 2018, S. 33f]. Als Basis für die Entwicklung sowie Anwendung betrieblicher ISe wird eine Datenbank (DB) benötigt, um die Speicherung und Verfügbarkeit von Daten zu realisieren [vgl. Bodendorf 2006, S. 7]. Als Ausgangspunkt für die Entwicklung einer DB, dient ein konzeptionelles Datenmodell, das die betrieblichen Prozesse durch Fachbegriffe beschreibt und darstellt. Damit das konzeptionelle Datenmodell sich zeitnah und ohne großen Aufwand an wechselnde Bedingungen anpassen kann, wird es generisch modelliert [Schneider 2007a, S. 3]. Der Begriff generisch kennzeichnet dabei Begriffe, die über ihren konkreten Inhalt hinaus eine allgemeine Bedeutung besitzen [vgl. Schneider 2007a, S. 408].

Darauf aufbauend verfolgt diese Arbeit das Ziel, die Anforderungen eines generischen Datenmodells für den Einsatz mit SC-Daten zu untersuchen und zu beschreiben. Hierfür werden wesentliche Anforderungen an generische Datenmodelle, Methoden zur Modellierung von Fachbegriffen sowie zwei Konzepte untersucht, die das Modellieren eines konsensfähig konzeptionellen Datenmodells sicherstellen und parallel dazu eine generische Darstellung ermöglichen. Darüber hinaus werden relevante SC-Daten herausgearbeitet, um die Untersuchungen auf das SC-Umfeld zu reflektieren. Dabei sollen Anforderungen und Methoden ermittelt werden, um generische SC-Fachbegriffe zu entwickeln. Die Anforderungen an die Konzepte sollen sowohl eine konsistente als auch generische Darstellung der SC-Fachbegriffe im generischen Datenmodell sicherstellen. Schlussendlich wird eine Empfehlung für ein generisches Datenmodell ausgesprochen.

Zunächst werden wesentliche SC-Daten herausgearbeitet (Kapitel 2.1.2), die im weiteren Verlauf dazu dienen, um SC-Fachbegriffe zu entwickeln. Zusätzlich wird das SCOR-Modell als Referenzmodell herangezogen (Kapitel 2.2), um die Prozesse in SCs untersuchen zu können. Die Basis der Untersuchung bildet die Anforderungen eines generischen Datenmodells (Kapitel 2.5). Darauf aufbauend werden Methoden und Konzepte unter-

sucht, die die Darstellung eines generischen Datenmodells ermöglichen (Kapitel 2.5.1, Kapitel 2.5.2 und Kapitel 2.5.3). Abschließend folgt die Empfehlung geeigneter Methoden und Konzepte sowie die Darstellung eines generischen Datenmodells (Kapitel 3.3). Hierfür werden aus den zuvor herausgearbeiteten SC-Daten SC-Fachbegriffe entwickelt, um beispielhafte SC-Prozesse im generischen Datenmodell abzubilden. Zuletzt werden Herausforderungen beschrieben sowie ein Ausblick gegeben (Kapitel 4).

2 Stand der Technik

Der Unterschied zwischen der klassischen Lieferkette und einer SC besteht darin, dass bei der SC eine ganzheitliche Betrachtung der Logistikkette zugrunde liegt. Dabei geht es um die Zusammenarbeit aller Unternehmen der unternehmensübergreifenden Wertschöpfungskette, da sie die Zuverlässigkeit innerhalb der SC in entscheidender Weise beeinflussen [vgl. Corsten und Gössinger 2008, S. 96]. Um eine optimale Zusammenarbeit der SC-Partner sicherzustellen, ist der Einsatz eines Datenbanksystems notwendig. Durch ein Datenbanksystem wird es ermöglicht, Daten aus Prozessen (z.B. Logistik, Produktion) zu erfassen, zu verarbeiten und Informationen aus den Daten abzuleiten [vgl. Meier und Kaufmann 2016, S. 2f]. Dieses Kapitel soll die theoretischen Grundlagen untersuchen, die im Rahmen dieser Arbeit wesentlich sind.

2.1 Daten im Supply-Chain-Kontext

In diesem Kapitel soll vorab ein grundlegendes Verständnis über Daten geschaffen werden. Darauffolgend werden wesentliche SC-Daten herausgearbeitet. Hierfür werden die in einer Studie evaluierten Ziele des SCM analog in Betrachtung gezogen, da sich aus den Zielen relevante SC-Daten ableiten lassen. Die evaluierten praxisorientierten Ziele dienen dazu, um diese mit den in der Literatur vorgestellten SC-Daten zu vergleichen. Somit können die herausgearbeiteten SC-Daten nicht nur den theoretischen, sondern auch den praxisorientierten Anforderungen gerecht werden.

2.1.1 Unterscheidung von Zeichen, Daten, Information und Wissen

Die Begriffe *Zeichen*, *Daten*, *Information* und *Wissen* werden im Alltag häufig im selben Kontext verwendet. Dieses Kapitel soll verdeutlichen, dass diese Begriffe aus der Sicht der Informatik klar zu differenzieren sind, da sie hierarchisch aufeinander aufbauen (vgl. Abbildung 2-1) [vgl. Krcmar 2015, S. 11; Gadatsch 2017, S. 1]. Als Ausgangspunkt für eine Begriffsinterpretation dienen *Zeichen*. Zeichen sind Buchstaben, Ziffern, Sonderzeichen und Symbole. Werden die Zeichen nach Syntaxregeln codiert, so darf man von Daten sprechen [vgl. Krcmar 2015, S. 4; Abts und Müller 2011, S. 24]. Durch die Interpretation der Daten, wird diesen eine besondere Bedeutung beigemessen. Das Ergebnis der Dateninterpretation sind Informationen [vgl. Gadatsch 2017, S. 2]. Überdies entsteht Wissen durch die Verknüpfung von Informationen [vgl. Bodendorf 2006, S. 1].

In der Betriebswirtschaftslehre wird eine Information als zweckorientiertes Wissen bezeichnet [vgl. Wittmann 1959, S. 14, zitiert nach Mielke et al. 2018, S. 7]. Im Rahmen dieser Definition ist unter zweckorientiert zu verstehen, dass nur solches Wissen als Information zu definieren ist, welches dazu dient, Entscheidungen oder Handeln vorzubereiten [vgl. Krcmar 2015, S. 5]. Demzufolge handelt es sich bei Informationen um einen Rohstoff für Wissen [vgl. North 2005, S. 33, zitiert nach Mielke et al. 2018, S. 7]. Aus diesem Grund stellt eine Information einen Produktionsfaktor im betrieblichen Leistungserstellungsprozess dar [vgl. Mielke et al. 2018, S. 256].

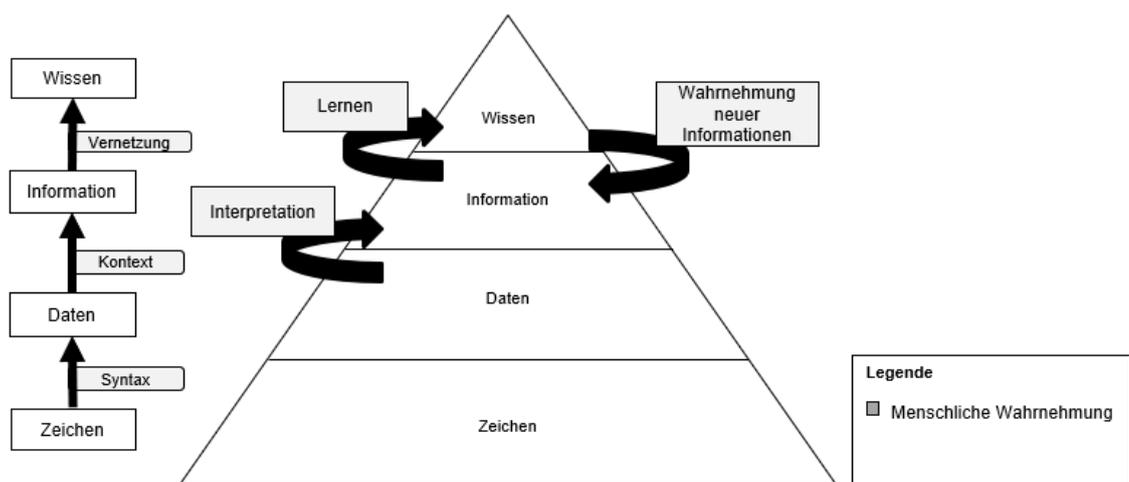


Abbildung 2-1: Begriffspyramide [vgl. Bodendorf 2006, S. 1]

Zusammenfassend ist auszuführen, dass der Wissensstand einer Person davon abhängig ist, was letztendlich in einem Prozess als Information zu verstehen ist [vgl. Alpar et al. 2016, S. 6]. Dies liegt darin begründet, dass je nach Wissensstand, Informationen unterschiedlich interpretiert werden können. Folglich müssen Informationen, bezogen auf Inhalt und Rolle, im Produktionsprozess analysiert werden, bevor sie dann entsprechend definiert werden können [vgl. Mielke et al. 2018, S. 21]. Dahingehend ist die Modellbildung als Interpretationsprozess entscheidend, um die Ressource Information effizient nutzen zu können [vgl. Krcmar 2015, S. 6]. Darüber hinaus ist auszuführen, dass die Grundlage jeder Planung durch Daten gebildet wird, die von einem IS bereitgestellt werden [vgl. Aliche 2005, S. 84 f.; Bretzke 2006, S. 3]. Durch die Planung und Organisation von Daten- und Informationsflüssen können Chancen und Risiken entlang der Wertschöpfungskette frühzeitig erkannt werden, um geeignete strategische Maßnahmen zu entwickeln [vgl. Fank 2001, S. 15].

2.1.2 Relevante Supply-Chain-Daten

Durch die Integration neuer Technologien, nimmt die Geschwindigkeit zu, mit der Daten in Unternehmen produziert werden. Die zunehmende Vielfalt von Datenquellen führt zu *unstrukturierten* und *strukturierten* Daten [vgl. Krcmar 2015, S. 94f].

Strukturierte Daten sind beispielsweise *Warenströme*, *Kundendaten* und *Aufträge*. Diese Daten werden durch ein Enterprise-Resource-Planning (ERP)-System erfasst [vgl. Gadatsch und Landrock 2017, S. 5]. Bei einem ERP-System handelt es sich um eine Software zur Ressourcenplanung in einem Unternehmen [vgl. Bretzke 2006, S. 11].

Unstrukturierte Daten (z.B. Texte, Bilder, Dokumente) können in Form von *Produktionsdaten* auftreten, die in der Produktion durch Sensoren erfasst werden [vgl. Gadatsch und Landrock 2017, S. 5].

Strukturierte Daten, die in einem fest definierten Schema vorliegen, werden nach Modellen wie dem Entity-Relationship-Modell (ERM) modelliert. Alternativ werden auch Klassendiagramme als Vertreter eines Strukturdiagramms von UML verwendet (siehe auch Kapitel 2.3.2) [vgl. Lemke et al. 2017, S. 354]. Um die Anforderungen generischer Datenmodelle im SC-Umfeld beschreiben zu können, werden in diesem Kapitel strukturierte SC-Daten herausgearbeitet. Dadurch können die Prozesse in SCs definiert werden.

Aufgrund der Tatsache, dass sich aus Anforderungen Ziele ableiten lassen und Ziele die Betrachtung auf wesentliche Bereiche fokussieren, werden vorab die Ziele des SCM beschrieben. Dadurch wird die Betrachtung auf wesentliche Bereiche gelenkt, aus denen sich relevante SC-Daten herausarbeiten lassen. Im Rahmen einer eindeutigen Definition zum SCM gehen die Meinungen vieler Autoren teilweise deutlich auseinander. Ein fehlendes einheitliches Verständnis mag der Tatsache geschuldet sein, dass der Ansatz des SCM seine Wurzeln in der Unternehmenspraxis hat [vgl. Wellbrock 2015, S. 20]. Jedoch scheint allgemein akzeptiert, dass SCM eine effiziente Koordination prozessorientierter Planung und Steuerung der Waren-, Informations- und Geldflüsse über den gesamten Wertschöpfungsprozess beschreibt. Zusätzlich ist ein nicht ganz unerheblicher Aspekt des SCM die Berücksichtigung der Beziehungen zwischen den Akteuren [vgl. Werner 2017, S. 5 f.; Bretzke 2006, S.3]. Demzufolge ist auszuführen, dass die Leistungserstellung zunehmend in einer stärkeren Kooperation der Geschäftspartner sowie einer flexiblen Arbeitsteilung erfolgt [vgl. Seufert et al. 2014, S. 16-25]. Hinsichtlich der Zielbeschreibung des SCMs empfiehlt Wellbrock [2015, S. 46] eine Aufteilung in Formal- und Sachziele, um einen höheren Detaillierungsgrad zu erreichen. Formalziele lassen sich als übergeordnete Gesamtziele beschreiben, während Sachziele sich an konkrete Prozesse orientieren. Hierarchisch betrachtet sind Sachziele den Formalzielen untergeordnet und bilden ein Mittel

zur Erreichung der Formalziele [vgl. Wellbrock 2015, S. 46]. In Anlehnung der zuvor beschriebenen Zielarten, werden in Abbildung 2-2 beispielhafte Ziele dargestellt.

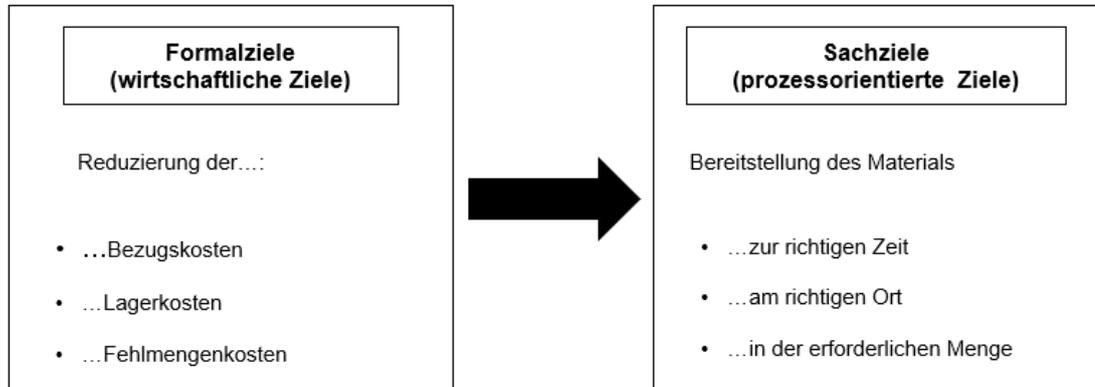


Abbildung 2-2: Ziele des SCM [vgl. Wellbrock 2015, S. 46]

Bei der Definition der SCM-Ziele ist eine Kohärenz zwischen verschiedenen betriebswirtschaftlichen Literaturen erkennbar. Dabei wird dem Formalziel der langfristige Erfolg hinsichtlich der Wettbewerbsfähigkeit der gesamten SC durch die Erschließung von unternehmensübergreifenden Erfolgspotenzialen zugesprochen.

Aus der Basis dieser Definition gehen die Sachzielkategorien *Steigerung des Endkundennutzens*, *Realisierung von Kostenvorteilen*, *Zeitvorteilen*, *Qualitätsvorteilen* und *Flexibilitätsvorteilen* hervor [vgl. Wellbrock 2015, S. 47f]. Um die Praxisrelevanz der Sachziele zu bestätigen, wird im Folgenden das Ergebnis der empirischen Studie der Philipps-Universität Marburg, welches durch den Lehrstuhl für Allgemeine Betriebswirtschaftslehre und Logistik im Jahre 2011 durchgeführt wurde, vorgestellt [vgl. Wellbrock 2015, S. 48f]. Das detaillierte Ergebnis dieser Studie ist aus Anhang A zu entnehmen.

Im Folgenden werden die praxisrelevanten Sachziele aufgeführt, die bei der Evaluation der Studie besonders hervorgestochen sind. Zu nennen sind: *Erhöhung der Produktverfügbarkeit*, *Verbesserung der Flexibilität hinsichtlich der Nachfrageänderung des Endkunden*, *Verkürzung der Durchlaufzeit*, *der möglichst effiziente Einsatz der Ressourcen* sowie *die Verbesserung der Produktqualität*. Diese Ziele können erreicht werden, indem alle wertschöpfenden Prozesse integriert und synchronisiert werden. Zur Erreichung dieser Ziele ist es entscheidend, die Nachfrageentwicklung der nachgelagerten Unternehmen zu kennen. Die Supply-Chain-weite Transparenz ist somit elementar für eine ganzheitliche Optimierung [vgl. Bretzke 2006, S. 3]. Mit Hilfe der zuvor beschriebenen Sachziele werden die SC-Daten herausgearbeitet, die für die Zielerreichung dieser Arbeit wesentlich sind.

Aufgrund der Orientierung an die praxisrelevanten Sachziele ist zusätzlich anzumerken, dass die in dieser Arbeit herausgearbeiteten SC-Daten auch eine Praxisrelevanz erfüllen. Zunächst werden die wertschöpfenden Prozesse der SC-Akteure genauer durchleuchtet. Der wertschöpfende Prozess besteht bei jedem SC-Akteur aus den Teilprozessen Beschaffung, Produktion sowie Vertrieb, was auch als interne SC bezeichnet wird [vgl. Werner 2017, S. 7]. Durch die Berücksichtigung der externen Prozesse, steigt die Komplexität der SC. Allerdings ist die externe Betrachtung aufgrund der hohen Anforderungen an die Unternehmen wesentlich [vgl. Wellbrock 2015, S. 22ff]. Die externe SC bezieht eingangsseitig die Lieferanten und ausgangsseitig die Kunden in das Netzwerk mit ein. Das Netzwerk ist nicht nur auf die direkten Lieferanten und Kunden limitiert, es bezieht ebenso „Lieferanten der Lieferanten“ sowie die „Kunden der Kunden“ mit ein [vgl. Werner 2017, S. 7]. Demnach steht der Erfolg eines Unternehmens in Abhängigkeit des Erfüllungsgrades der kompletten SC, weshalb die Ziele der Netzwerkpartner korrespondieren müssen, um den Anforderungen an Logistiknetzwerken gerecht zu werden [vgl. Yüzcülec et al. 2010, S.81]. Eine Übersicht der anfallenden SC-Prozesse liefert Abbildung 2-3.

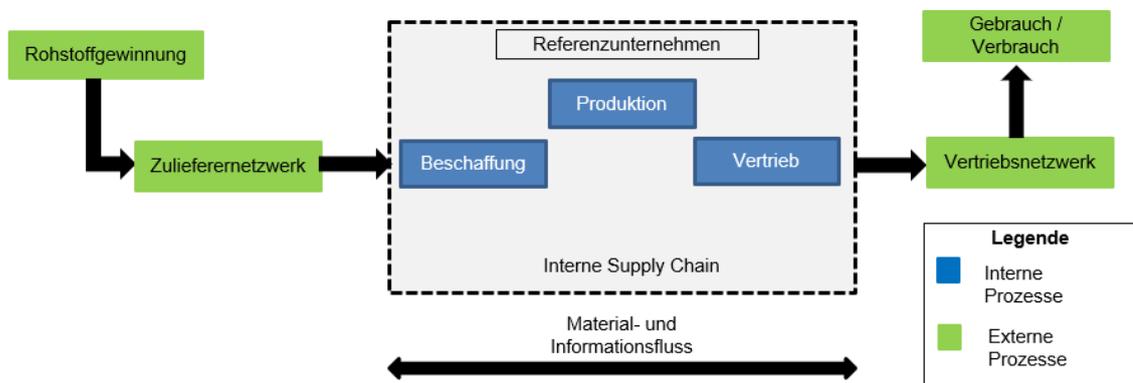


Abbildung 2-3: Modell einer SC [vgl. Sommer 2007, S. 30]

Die interne SC beschreibt die internen Prozesse eines Unternehmens. Diese entsprechen den Aufgaben der internen Produktionsplanungs- und Steuerungssysteme (PPS) [vgl. Schuh und Stich 2012, S. 111ff]. Die in der PPS anfallenden Daten dienen als Grundlage zur Vervollständigung der SC-Daten, die entlang der Wertschöpfungskette auftreten. Aus diesem Grund werden die Daten der PPS im Folgenden näher betrachtet.

Die Abgrenzung nach verschiedenen Datenkategorien ermöglicht es, Daten zu strukturieren und diese mittels eines Datenmanagements den entsprechenden Geschäftsprozessen des Unternehmens zur Verfügung zu stellen [vgl. Weber und Klotz 2017, S. 218 ff.]. Die Daten der PPS werden in *Stamm-* und *Bewegungsdaten* eingeteilt [vgl. Alicke 2005, S. 85;

Schemm 2009, S. 19f]. Stammdaten weisen eine lange Lebensdauer auf und beanspruchen einen umfassenden Pflegeaufwand. Dies ist der Tatsache geschuldet, dass im Planungsprozess permanent auf den Stammdatenbestand zurückgegriffen wird [vgl. Schuh und Stich 2012, S. 73]. Demgegenüber stehen Bewegungsdaten, deren Lebensdauer auf eine bestimmte Zeit begrenzt ist. Dabei wird zwischen zeitbezogenen Daten und unterschiedlichen Statuszuständen unterschieden. Als Beispiel für zeitbezogene Daten ist der Lagerbestand zu einem bestimmten Zeitpunkt zu nennen. Für unterschiedliche Statuszustände sind als Beispiel Produktionsaufträge zu nennen, deren Zustände *gesperrt*, *freigegeben*, *in Bearbeitung* und *abgeschlossen* sein können.

In Tabelle 2-1 sind die relevanten Daten der PPS zusammengefasst, die im weiteren Verlauf der Arbeit näher erläutert werden. Darüber hinaus muss Tabelle 2-1 um eine weitere Ebene erweitert werden. Sowohl Alicke [2005, S. 85] als auch Schuh und Stich [2012, S. 73-79] bezeichnen in ihrer Arbeit die in der dritten Ebene enthaltenen Angaben als Attribute.

Tabelle 2-1: Stamm- und Bewegungsdaten [vgl. Schuh und Stich 2012, S. 73-79]

1. Ebene	2. Ebene	3. Ebene
Stammdaten	Materialdaten	Attribute
	Stücklisten	
	Ressourcendaten	
	Arbeitspläne	
	Kundendaten	
	Lieferantendaten	
Bewegungsdaten	Lagerbestandsdaten	
	Produktionsdaten	
	Betriebsdaten	

Materialdaten werden nach verschiedenen Merkmalen differenziert. Mögliche Unterscheidungen sind die Beschaffungsart (Eigenfertigungsteile, Fremdbezugsteile und Fertigungsteile), der Grad der Bearbeitung (Baugruppen, Einzelteile, Enderzeugnisse), der Verwendungszweck sowie die Verwendungsart. Je nach Materialart kann es mit anderen Daten in Relation stehen. Als Beispiel ist ein Material zu nennen, dass sowohl *eigengefertigt* als auch *fremdbezogen* wird. Hierfür werden sowohl Lieferantendaten als auch Stücklisten sowie Arbeitspläne benötigt. Als Gegenbeispiel ist ein nur *fremdbezogenes Material* zu

nennen, welches Lieferantendaten benötigt, jedoch keine Stücklisten und Arbeitspläne. Den Materialdaten können *Materialnummer, Materialbezeichnung, MIN Bestellmenge, MIN Produktionsmenge, Maße, Gewicht, Zeichnungsnummer, Mengeneinheit, Produktionslosgröße, Alternativmaterialien, Volumen, Gebinde, Gewicht, Kosten* (Preis, Wertschöpfung, Lagerhaltung) als Attribute zugeordnet werden [vgl. Alicke 2005, S. 85; Schuh und Stich 2012, S. 73f].

Stücklisten dienen der Fertigung von Eigenfertigungsteilen und beschreiben die hierarchische Abfolge ihrer Zusammensetzung. Es wird zwischen verschiedene Stücklistenarten, wie zum Beispiel Mengenübersichtsstücklisten, Baukastenstücklisten und Strukturstücklisten unterschieden. Als mögliche Attribute können *optionale Komponenten, Bedarfskoeffizient, Produktbezeichnung* und *Kapazität* genannt werden [vgl. Alicke 2005, S. 85; Schuh und Stich 2012, S. 75].

Durch *Ressourcendaten* werden Informationen über die im Unternehmen vorhandenen Arbeitsplätze sowie das jeweilige Kapazitätsangebot bereitgestellt. Das Kapazitätsangebot lässt sich aus dem Leistungsvermögen einer Maschine, einem Betriebskalender und dem darin integrierten Schichtmodell berechnen. Für die interne Leistungsverrechnung werden in den Datensätzen zusätzlich die Kostenstellennummer und Maschinenstundensätze hinterlegt. Als mögliche Attribute der Ressourcendaten sind *optionale Ressourcen, Maschinenstundensatz, Schichtpläne, Identifizierungsnummer, Kostenstellennummer, Personalqualifikation, Zeitraum der Maschinenzuordnung* und *Lohnkostensatz* zu nennen [vgl. Alicke 2005, S. 85; Schuh und Stich 2012, S. 74f].

Arbeitspläne beschreiben die Produktionsprozesse von Produkten. Die Arbeitspläne werden durch Arbeitsgänge beschrieben, die im jeweiligen Produktionsprozess anfallen. Die Arbeitsgänge werden durch ihre Zeiten definiert. Je mehr Zeitanteile und entsprechende Kostensätze im Arbeitsgang beschrieben werden, desto differenzierter können die Kosten des Arbeitsganges bestimmt werden. Mögliche Attribute sind *Arbeitsplanstatus, Optionale Arbeitspläne, Durchlaufzeit, Rüstzeit, Liegezeit, interne Transportzeit, Standzeit, Personalzeit, Nebenzeit, Übergangszeit, Bearbeitungszeit* [vgl. Schuh und Stich 2012, S. 76].

Unter dem Begriff *Kundendaten* sind Adressen von ehemaligen, bestehenden oder potenziellen Vertriebsgeschäftspartnern zu verstehen, die je nach Zweck unterschiedlich relevant sind. Da der Vertrieb der internen SC zugeordnet wird (vgl. Abbildung 2-3) und die interne PPS auch als interne SC verstanden werden kann, beziehen sich die Kundendaten auf unternehmensinterne Produktionsbereiche. Als mögliche Attribute können *Versandadresse, Rechnungsadresse, Produkt, Qualitätsnorm* sowie *Zahlungsbedingungen* den Kundenstammdaten zugeordnet werden [vgl. Schuh und Stich 2012, S. 76f].

Lieferantendaten sind kongruent zu Kundenstammdaten. Im Einkaufsbereich werden Beziehungen zusammengefasst, die der Fremdbezugsplanung und -steuerung zugrunde liegen und die Organisationsstruktur abbilden. Als Attribute der Lieferantendaten sind beispielsweise *lieferantenspezifische Materialbezeichnungen, alternative Materialien, Lieferzeiten, Liefermengen* sowie *Verpackungs- und Qualitätsspezifikationen* zu nennen [vgl. Schuh und Stich 2012, S. 77].

Lagerbestandsdaten beschreiben die Material- und Produktflüsse im Prozess. Dargestellt werden Zu- und Abgänge sowie der aktuelle Bestand. Diese Prozesse erfolgen für alle Materialien und Produkte, weshalb zwischen Material- und Produktstammdaten sowie den Bestandskonten ein eindeutiger Zusammenhang besteht. Im Rahmen der Lagerbestandsdaten können die Attribute *Status, Produkt, Bestand* sowie *Zeitstempel* genannt werden [vgl. Schuh und Stich 2012, S. 78].

Produktionsauftragsdaten koordinieren den Auftragsabwicklungsprozess. Auf Basis der vorliegenden Kundenaufträge oder eines prognostizierten Absatzplans können mengen- und zeitmäßige Abstimmungen der Ressourcen erfolgen [vgl. Schuh und Stich 2012, S. 78]. Hinsichtlich der Produktionsauftragsdaten können als mögliche Attribute *Auftragsposition, Status, Produkt, Beschaffungsmenge, Liefermenge, vereinbarter Liefertermin, Repeaterfaktor* und *Zeitstempel* genannt werden [vgl. Aliche 2005, S. 85; Nebl und Corsten 2011, S. 8].

Betriebsdaten beschreiben eine zeitliche, mengenmäßige, kapazitäts- und auftragsbezogene Erfassung von IST-Werten, die innerhalb eines Produktionsprozesses auftreten. Den Betriebsdaten sind die Attribute *Kapazitätsauslastung, Fertigungsfortschritt*, und *Zeitstempel* zuzuordnen [vgl. Schuh und Stich 2012, S. 78f].

Die zuvor beschriebenen Stamm- und Bewegungsdaten beziehen sich auf die interne SC. Diese müssen um die Daten erweitert werden, die die externen Prozesse bzw. externe SC miteinbeziehen. Bei verschiedenen Autoren ist eine Kohärenz hinsichtlich der Notwendigkeit der Erweiterung des PPS auf unternehmensübergreifende SC-Aufgaben zu erkennen. Diesbezüglich erläutert Nyhuis [2012, S. 181f], dass die Produktionslogistik bei allen Planungsschritten in den Vordergrund gestellt werden muss, um über alle Planungsstufen realistische und aufeinander abgestimmte Vorgaben zu ermöglichen. Demzufolge haben SC-Aufgaben aufgrund ihres übergreifenden Charakters Anknüpfungspunkte zu verschiedenen Kernaufgaben der lokalen Planungsebene.

Zusammengefasst lässt sich ausführen, dass alle Daten, die in den Kernaufgaben der internen PPS benötigt werden, auch für die SC-Aufgaben von Bedeutung sind. Darauf

aufbauend ist zu untersuchen, welche Daten in den unternehmensübergreifenden Prozessen anfallen. Hierfür wird auf Kurbel [2016, S. 443-452] und Alicke [2005, S. 85] verwiesen, aus deren Arbeiten eine Kohärenz hinsichtlich der externen Prozessdaten ersichtlich ist. Die Daten der externen Prozesse beschreiben Interaktionen zwischen den SC-Akteuren und werden den Stammdaten zugeordnet. Repräsentativ zu nennen sind *Herstellerdaten*, *Lieferantendaten* sowie Daten über *Transportzeiten*, *-frequenzen* und *-art*.

Auch ist es möglich, die Bewegungsdaten um Daten zu erweitern, die eine vorausschauende Planung im SC-Netzwerk ermöglichen. Zuvor wurde ausgeführt, dass die Leistungserstellung eines Unternehmens zunehmend in einer stärkeren Kooperation der Geschäftspartner sowie einer flexiblen Arbeitsteilung erfolgt, weshalb der Erfolg eines Unternehmens nicht nur bei den eigenen Prozessen liegt, sondern auf die gesamte SC zurückzuführen ist. Um eine flexible Arbeitsteilung zu ermöglichen, werden die Bewegungsdaten mit Daten der Verfügbarkeitsplanung erweitert und den Betriebsdaten zugeordnet. Mit Hilfe dieser Daten soll bestimmt werden, inwieweit Liefertermine und Liefermenge der aktuell angefragten Kundenaufträge erfüllt werden können, um flexibel auf spezifische Geschehnisse reagieren zu können [vgl. Ball et al. 2004, S. 447-483]. Die Rede ist von *Available-to-Promise (ATP)*-Daten sowie *Capable-to-Promise (CTP)*-Daten.

Beim Available-to-Promise-Ansatz wird auf Ebene der Endprodukte analysiert, ob es aufgrund der aktuellen Lagerbestände, der geplanten Lagerzugänge und der bereits bestehenden Lieferverpflichtungen möglich ist, die angefragte Menge zum angefragten Liefertermin auszuliefern [vgl. Corsten und Gössinger 2008, S. 215].

Beim Capable-to-Promise-Ansatz wird geprüft, ob die vom Kunden angefragten Lieferzeiten und -mengen durch die noch frei verfügbare Kapazität erfüllt werden können [vgl. Corsten und Gössinger 2008, S. 216].

Somit sind alle für die Ausarbeitung relevanten SC-Daten herausgearbeitet und in Tabelle 2-2 entsprechend kategorisiert. Darüber hinaus ist anzumerken, dass die zweite und dritte Ebene variabel ist. Dies liegt darin begründet, dass jedes Unternehmen bzw. jeder Datenmodell-Konstrukteur in Anlehnung an seine spezifischen Prozesse die vorkommenden Daten verschieden interpretieren kann (siehe auch Kapitel 2.1.1).

Tabelle 2-2: Zusammenfassung der wesentlichen SC-Daten [vgl. Schuh und Stich 2012, S. 73-79; Alicke 2005, S. 85; Kurbel 2016, S. 443-452; Corsten und Gössinger 2008, S. 215-216; Nebl und Corsten 2011, S. 8]

1. Ebene: Stammdaten	
2. Ebene	3. Ebene
Materialdaten	Materialbezeichnung
	Maße
	Gewicht
	Volumen
	Mengeneinheit
	Optionale Materialien
	Produktionslosgröße
	Materialnummer
	Zeichnungsnummer
	Kosten
	Gebinde
	MIN Bestellmenge
	MAX Produktionsmenge
Stücklisten	Produktbezeichnung
	Optionale Komponente
	Bedarfskoeffizient
	Kapazität
Ressourcendaten	Optionale Ressourcen
	Maschinenstundensatz
	Schichtpläne
Arbeitspläne	Arbeitsplanstatus
	Optionale Arbeitspläne
	Arbeitsgang (Zeiteinheiten)
Kunden- und Lieferantendaten	Versandadresse
	Rechnungsadresse
	Produkt
	Qualitätsnorm
	Zahlungsbedingungen
SC-Daten (Externe Logistik)	Hersteller
	Lieferant
	Optionaler Lieferant
	Externe Transportzeit
	Transportfrequenz
	Transportart

1. Ebene: Bewegungsdaten	
2. Ebene	3. Ebene
Lagerbestandsdaten	Bestand
	Status
	Produkt / Repetierfaktor
	Zeitstempel
Produktionsauftragsdaten	Auftragsposition
	Status
	Produkt
	Beschaffungsmenge
	Vereinbarter Liefertermin
Betriebsdaten	Zeitstempel
	Kapazitätsauslastung
	Fertigungsfortschritt
	Available-to-Promise
	Capable-to-Promise

Legende

Arbeitsgang (Zeiteinheiten)	Durchlaufzeit
	Rüstzeit
	Liegezeit
	Interne Transportzeit
	Standzeit
	Personalzeit
	Nebenzeit
	Übergangszeit
Bearbeitungszeit	

2.2 Referenzmodelle

Grundsätzlich werden Referenzmodelle dazu verwendet, Geschäftsprozesse zu systematisieren und einheitlich zu gestalten [vgl. Poluha 2010, S. 78]. Aus diesem Grund können Referenzmodelle als Vorlage für die Konstruktion von unternehmensspezifischen Datenmodellen verwendet werden [vgl. Krcmar 2015, S. 44; Becker et al. 1997, S.3]. Im Rahmen dieser Arbeit soll das SCOR-Modell dazu verwendet werden, um die Prozesse und Schnittstellen in SCs näher zu untersuchen. Darauf aufbauend sollen Anforderungen untersucht werden, um die Balance zwischen dem Detailierungsgrad und der Wiederverwendbarkeit eines (konzeptionellen) Datenmodells sicherzustellen.

2.2.1 Supply-Chain-Operations-Reference-Modell

Das Supply-Chain-Operations-Reference (SCOR)-Modell stellt einen Ansatz dar, um die Lieferkette einer Organisation zu definieren [vgl. Bolstorff et al. 2007, S. 15]. Durch die Verwendung des SCOR-Modell soll gewährleistet werden, dass allen beteiligten Partnern einer SC eine einheitliche Sprache für die Kommunikation (unternehmensintern und -übergreifend) zur Verfügung gestellt wird [vgl. Werner 2017, S. 70f]. Das SCOR-Modell ist den normativen Modellen zuzuordnen. Ein normatives Modell setzt sich aus einem vordefinierten Ansatz zusammen und beschreibt wie ein Objekt des Modells gesehen wird und wie es sich verhalten sollte [vgl. Bolstorff 2007, S. 16]. Statt des Begriffs des normativen Modells findet sich auch häufig die Bezeichnung des *Referenzmodells*.

Das SCOR-Modell unterliegt einer hierarchischen Struktur, bei dem der Grad der Konkretisierung mit jeder Entwicklungsstufe zunimmt. Diese Entwicklungsstufen sind *Top-Level*, *Configuration-Level*, *Process-Element-Level* und *Implementation-Level* (vgl. Tabelle 2-3).

Tabelle 2-3: Entwicklungsstufen des SCOR-Modells [vgl. Werner 2017, S. 71]

Entwicklungsstufen	Bezeichnung
1. Stufe	Top-Level
2. Stufe	Configuration-Level
3. Stufe	Process-Element-Level
4. Stufe	Implementation-Level

Im Rahmen dieser Arbeit sind für ein grundlegendes Verständnis die Kernprozesse der obersten Entwicklungsstufe des SCOR-Modells ausreichend, weshalb auf die Prozesse der anderen Stufen nicht näher eingegangen wird.

Auf der obersten Stufe unterscheidet das Modell fünf Prozesse, die den Umfang und den Inhalt einer SC definieren. Zu nennen sind *Planen*, *Beschaffen*, *Herstellen*, *Liefern* und *Zurückführen* (vgl. Abbildung 2-4). Im Folgenden werden die Prozesse der Top-Level-Stufe näher erläutert. Repräsentativ ist Werner [2017, S. 71f] auszuführen. Abweichende Quellen werden angegeben.

- *Planen*: Dieser Prozess weist einen gewissen übergeordneten Charakter auf. Er umfasst neben der Planung der Infrastruktur auch die Ressourcenplanung sowie die Ermittlung der Nachfrageanforderungen.
- *Beschaffen*: Der Prozess Beschaffen umfasst alle Beschaffungsaktivitäten. Darunter fallen Aktivitäten, die im Rahmen des Erwerbs, des Erhalts, der Qualitätsprüfung, der Bezahlung und der Bereitstellung des eingehenden Materials anfallen.
- *Herstellen*: Dieser Prozess umfasst die Kapazitätssteuerung, Zwischenlagerung sowie Verpackung und Übergabe an den Vertrieb [vgl. Stözle und Halsband 2005, S. 541].
- *Liefern*: Der Prozess Liefern bezieht sich auf die Distributionsprozesse an der Schnittstelle zum Kunden.
- *Zurückführen*: Der letzte Prozess umfasst alle administrativen Tätigkeiten, die mit der Rückgabe von Rohstoffen (an Lieferanten) oder dem Empfang rückgeführter Fertigung (vom Kunden) verbunden sind.

Die zweite Stufe *Configuration-Level* bildet die SC eines Unternehmens ab. Dazu wird die SC in unterschiedliche Teilprozesse zerlegt. Dabei bilden die fünf Kernprozesse aus der ersten Ebene die Prozesskategorien [vgl. Werner 2017, S. 72].

Auf der dritten Stufe *Process-Element-Level* werden die Prozesskategorien in einzelne Prozesselemente zerlegt. Im Fokus steht die Definition dieser Prozesselemente und die Ermittlung von In- und Output-Beziehungen [vgl. Werner 2017, S. 75].

Als letzte Stufe ist der *Implementation-Level* zu nennen, welcher bei der Implementierung der Prozesselemente unterstützt. Die vierte Stufe ist nicht dem SCOR-Ansatz

direkt zugehörig, da aufgrund der unterschiedlichen Prozessspezifikationen kein einheitliches Konzept definiert werden kann. Jedoch ist die Stufe der Implementierung in ihrer Durchführung wesentlich, da das SCOR-Modell sonst unvollendet bliebe [vgl. Werner 2017, S. 77].

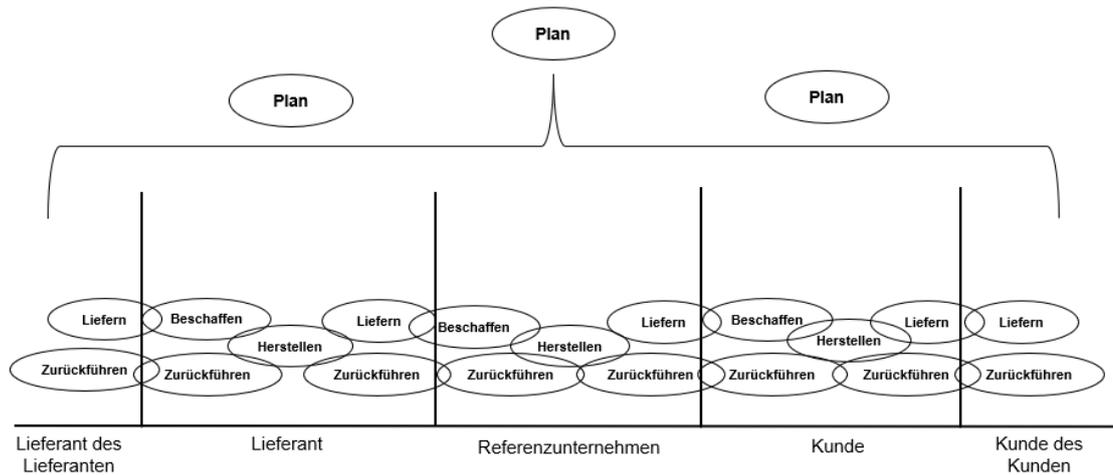


Abbildung 2-4: Kernprozesse des SCOR-Modells [vgl. Stölzle und Halsband 2005, S. 14]

2.2.2 Informationen im Supply-Chain-Netzwerk

Informationen können im Supply-Chain-Netzwerk sowohl stromaufwärts als auch stromabwärts fließen. Die Steuerung einer SC erfolgt primär durch den Endkunden. Die Strategie der Ausrichtung des Angebots an der Nachfrage des Konsumenten ist in der Logistik als *Pull-Prinzip* bekannt. Das Pull-Prinzip sagt aus, dass stromaufwärts der Wertschöpfungskette keine Leistung produziert wird, bevor eine Kunde diese stromabwärts angefordert hat [vgl. Hertel et al. 2005, S. 172f]. Offensichtlich ist, dass Informationen ein Steuerungselement darstellen, welches durch den Verbrauch des Kunden bestimmt wird. Informationen befinden sich in jedem Prozess einer SC. Dabei können diese dem Materialfluss *vorauslaufen*, *begleiten* oder *nachfolgen* [vgl. Pfohl 2004, S. 8]. Die nachfolgende Abbildung (vgl. Abbildung 2-5) liefert ein Beispiel für eine polyzentrische SC.

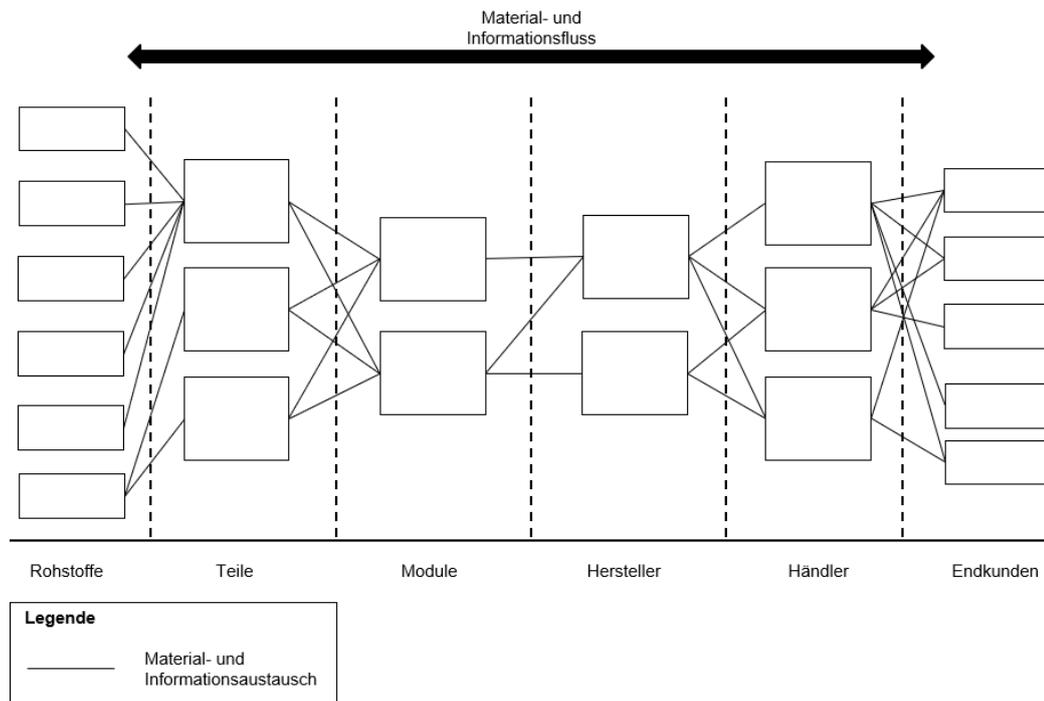


Abbildung 2-5: Material- und Informationsfluss in einer polyzentrischen SC [vgl. Werner 2017, S. 28]

Im Ursprung der Wertschöpfungskette stehen Rohstofflieferanten und Teilelieferanten im wechselseitigen Abhängigkeitsverhältnis miteinander. Die engste Bindung bilden Modullieferanten und Hersteller, die sich wiederum in Austauschprozessen mit dem Handel befinden. Endkunden bilden die letzte Instanz, die in diversen Austauschprozessen mit dem Handel stehen [vgl. Werner 2017, S. 28].

2.2.3 Vor- und Nachteile eines Referenzmodells

Die Wiederverwendung von Referenzmodellen ermöglicht die schnellere Erstellung individueller Modelle. Referenzmodelle stellen eine Vereinheitlichung der Fachbegriffswelt dar, wodurch eine bessere Dokumentation der bestehenden Abläufe sichergestellt wird. Außerdem wird die Identifikation relevanter Prozesse ermöglicht [vgl. Gronau 2010, S. 86]. Das Referenzmodell greift auf bekannte Best-Practice-Methoden zurück, wodurch die Effizienz im Modellierungsprozess gesteigert wird [vgl. Krcmar 2015, S. 28].

Die Standardisierung durch Nutzung von Referenzmodellen kann auch zum Verlust strategischer Wettbewerbsvorteile führen, wenn die Stärken eines Unternehmens bei der Modellerstellung nicht berücksichtigt werden [vgl. Gronau 2010, S. 86]. Darüber hinaus würden sich die verschiedenen logistischen Prozesse zwischen konkurrierenden SC's kaum

unterscheiden. Zusätzlich können *unzureichende Modellanpassungen* zu hohen Kosten führen. Dies ist darauf zurückzuführen, dass eine große Herausforderung darin besteht, zwischen Detailierungsgrad und Wiederverwendbarkeit im (konzeptionellen) Datenmodell abzuwägen [vgl. Krcmar 2015, S. 28].

Zusammenfassend ist auszuführen, dass ein Referenzmodell in erster Linie dazu dienen soll, die allgemeinen Prozesse eines Anwendungsbereichs in einem konzeptionellen Datenmodell darzustellen. Die im SCOR-Modell enthaltenen Prozesse erfüllen einen allgemeingültigen Charakter, weshalb das SCOR-Modell als Konstruktionsvorlage für konzeptionelle Datenmodelle im SC-Umfeld dient. Die durch Referenzmodelle herbeigeführte unzureichende Modellanpassung verlangt eine Darstellung, um der unzureichenden Modellanpassung entgegenzuwirken. Damit das daraus modellierte konzeptionelle Datenmodell wiederverwendet werden kann, ist es notwendig, dass die Anpassung oder Integration neuer Prozesse erlaubt wird. Dies wird durch das generische Datenmodell ermöglicht, welches im Einzelnen unter Kapitel 2.5 beschrieben wird.

2.3 Grundlagen des Datenbankentwurfs

Eine Datenbank ist eine Ansammlung von Daten, die Prozesse der realen Welt, wie beispielsweise die eines Unternehmens beschreiben soll. Das Arbeiten mit einer DB erfordert ein Datenbankmanagementsystem (DBMS). Ein DBMS ist ein Softwaresystem (d.h. eine Sammlung von Programmen), das dem Anwender die Pflege einer DB ermöglicht [vgl. Kudraß 2015, S. 20f]. Dies beinhaltet die Manipulation und Definition von Datenbanken [vgl. Herrmann 2018, S. 5]. Das Verknüpfen aus mehreren Datenbanken und einem DBMS wird als Datenbanksystem (DBS) bezeichnet [vgl. Kudraß 2015, S. 21]. Der grundlegende Aufbau eines DBS kann durch die drei-Ebenen-Architektur beschrieben werden. Dieses Modell wurde 1975 vom Standards Planning and Requirements Committee (SPARC) des American National Standards Institute (ANSI) entwickelt (vgl. Abbildung 2-6) [vgl. Herrmann 2018, S. 5f]. Die im Folgenden aufgeführten Sprachen werden in vielen Literaturen als vom DBMS unterstützte Sprachen genannt.

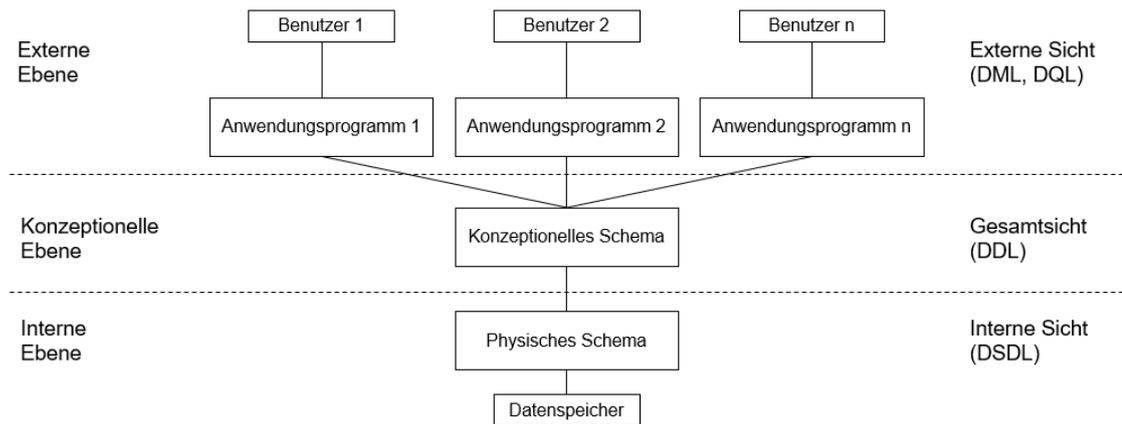


Abbildung 2-6: Drei-Ebenen-Modell [vgl. Herrmann 2018, S. 6]

- Die *externe Ebene* benötigt Sprachen, die dem Anwender einer DB den Umgang mit dem Datenbestand ermöglichen soll. Dabei wird zwischen der *Datenmanipulationssprache* (DML = Data Manipulation Language) und der *Datenabfragesprache* (DQL = Data Query Language) unterschieden. Die Datenmanipulationssprache soll dem Anwender eines Anwendungsprogramms Zugriff auf die Datenbank ermöglichen, um Daten beispielsweise zu ändern, hinzuzufügen oder zu löschen. Die Datenabfragesprache ermöglicht dem Anwender die Abfrage von Daten [vgl. Herrmann 2018, S. 6]. In Abbildung 2-6 wird auf der externen Ebene die Sicht der einzelnen Nutzer auf die Datenbank dargestellt.
- Die *konzeptionelle Ebene* benötigt eine Sprache, mit der sich die Datenstrukturen beschreiben lassen. Mit der Datendefinitionssprache (DDL = Data Definition Language) wird zunächst die Datenstruktur in einem konzeptionellen Datenmodell, beispielsweise durch das Entity-Relationship-Modell (ERM) beschrieben. Diese Ebene beschreibt somit ein konzeptionelles Schema, um die Gesamtsicht aller Daten und deren Beziehung innerhalb der DB zu beschreiben [vgl. Herrmann 2018, S. 6].
- Die *interne Ebene* verwendet eine Speicherbeschreibungssprache (DSDL = Data Storage Description Language), um Zugriffspfade und die Speicherverwaltung der Daten zu beschreiben. Diese Ebene beschreibt somit das physische Schema einer DB [vgl. Herrmann 2018, S. 6].

Darauf aufbauend wird das Phasenkonzept nach Herrmann [2018, S. 8] vorgestellt und beschrieben (vgl. Abbildung 2-7). Das Phasenkonzept beschreibt eine systematische Vor-

gehensweise, um eine DB zu entwerfen. Diesbezüglich führt Herrmann [2018, S. 9] aus, dass das dargestellte Phasenkonzept als Anregung zur Erstellung einer DB dienen soll. Da jedes Projekt zur Erstellung einer DB sich unterscheidet, sind die vorgestellten Phasen nicht als streng sequenzielles Vorgehen zu verstehen, sondern müssen an die eigenen Bedürfnisse angepasst werden.

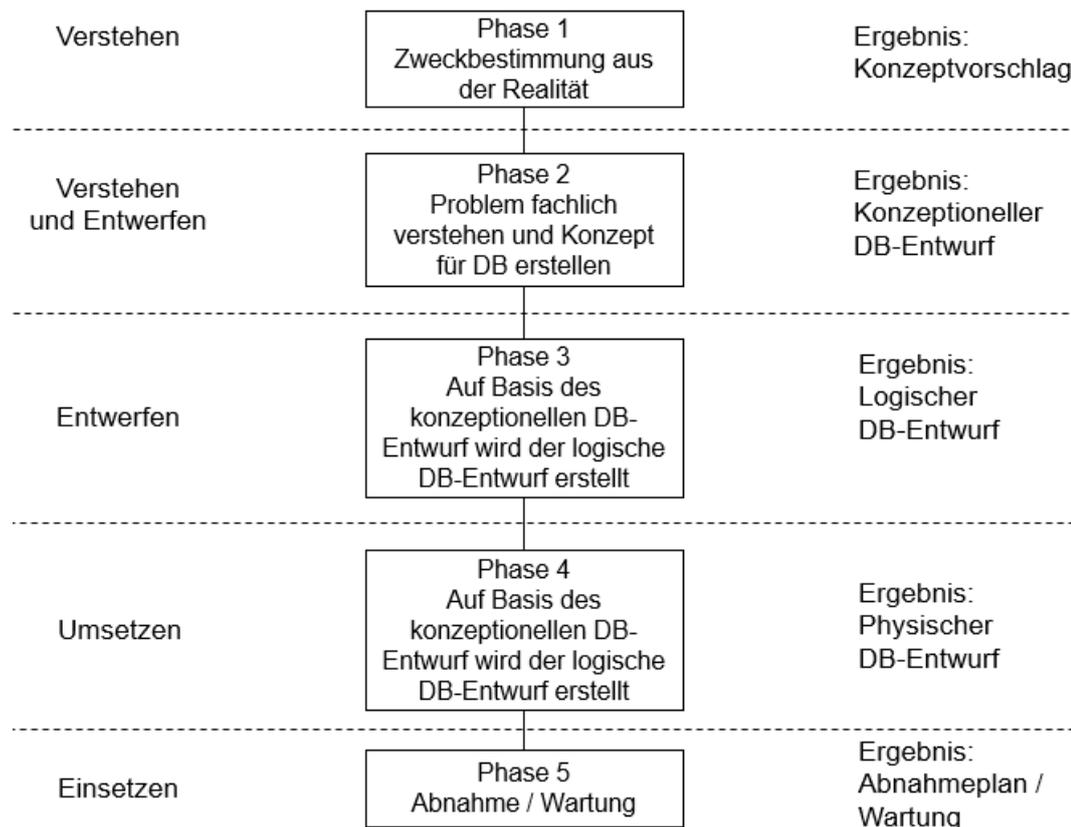


Abbildung 2-7: Phasenkonzept des Datenbankentwurfs [vgl. Herrmann 2018, S. 8]

- Phase 1:* In dieser Phase wird auf Grundlage der Realität festgelegt, für welchen Zweck eine DB entworfen werden soll. Die Zweckbestimmung (Analysephase) bildet die Basis und somit die wichtigste Phase, da letztendlich festgelegt wird, welche Daten als relevant anzusehen sind. Das Ergebnis der ersten Phase ist der Konzeptvorschlag, der das Ziel des Projektes beschreibt [vgl. Herrmann 2018, S. 8].
- Phase 2:* Anschließend wird in Phase zwei ein *konzeptioneller Datenbankentwurf* erstellt. Hierfür werden die bisher abgewickelten Geschäftsprozesse analysiert und ermittelt. Dabei steht im Vordergrund, welche Objekte mit welchen Informationen existieren (z.B. ein Lieferant mit den Informationen Lieferantenummer, Adresse)

und welche Beziehungen und Einschränkungen zwischen den Objekten bestehen. Hierbei kann beispielsweise das Entity-Relationship-Modell (ERM) nach Chen angewendet werden. Der konzeptionelle Datenbankentwurf dient als Grundlage zur Erstellung des logischen Datenbankentwurfs [vgl. Herrmann 2018, S. 8].

- *Phase 3:* Da der konzeptionelle Datenbankentwurf nicht direkt in eine Datenbankstruktur transformiert werden kann, wird aus dem konzeptionellen Datenbankentwurf ein *logischer Datenbankentwurf* entwickelt. Im logischen Datenbankentwurf wird die Art der Speicherung der Daten umgesetzt. Hierfür wird beispielsweise das relationale Datenmodell verwendet, in welchem die Daten in Form von Tabellen abgespeichert werden. Dabei sind spezielle Transformationsregeln zu beachten, die an dieser Stelle nicht weiter ausgeführt werden [vgl. Herrmann 2018, S. 9].
- *Phase 4:* In dieser Phase wird der logische Datenbankentwurf in *einein physischen Datenbankentwurf* überführt. Man spricht hierbei auch von der *Phase der Realisierung*. Dabei wird bei relationalen Datenbanken das relationale Datenmodell mit der Datenbeschreibungssprache *Structured Query Language (SQL)* in eine physische Datenbankstruktur umgesetzt [vgl. Herrmann 2018, S. 9].
- *Phase 5:* Abschließend wird das System eingeführt. Nach der Kundenabnahme erfolgt die Wartung nach einem zuvor definierten Wartungsplan sowie die Betreuung und Schulung der Endbenutzer [vgl. Herrmann 2018, S. 9].

Zusammenfassend ist auszuführen, dass der Datenbankentwurf eine systematische Vorgehensweise verlangt. Die Trennung zwischen einem *konzeptionellen*, *logischen* und *physischen* Datenbankentwurf sorgt für eine Trennung der Beschreibung der Daten in separate Phasen. Dadurch wird eine übersichtliche Entwicklung einer DB in verschiedenen Entwicklungsphasen sichergestellt. Im folgenden Kapitel sollen nun grundlegende Datenmodelle vorgestellt werden.

2.4 Datenbankmodelle

Der Aufbau eines Datenbankmodells erfolgt nach bestimmten Rahmenbedingungen, die von Modell zu Modell unterschiedlich sind [vgl. Geisler 2009, S. 54]. Um eine DB zu entwerfen, müssen geeignete Datenbankmodelle verwendet werden, die einen Realitätsausschnitt möglichst exakt beschreiben. Dabei ist zwischen *abstrakten* und *konkreten* Datenbankmodellen zu unterscheiden [vgl. Kudraß 2015, S. 22]. Resultierend aus der Kategori-

sierung der Datenbankmodelle wird repräsentativ Geisler [2009, S. 54] genannt. Er kategorisiert zwischen *implementativen (konkreten)* und *konzeptionellen (abstrakten)* Datenbankmodellen.

Um im Rahmen dieser Arbeit ein einheitliches Begriffsverständnis aufzubauen sowie die Unterscheidung der oben angeführten Kategorien zu verdeutlichen, wird die Kategorisierung durch die Definition von Jarosch [2016, S. 26] weiter präzisiert. Er führt aus, dass die Strukturbeschreibung der Daten, die auf kein spezielles DBMS Bezug nimmt, als *konzeptionelles Datenmodell* bezeichnet wird. Somit wird im Rahmen dieser Arbeit zwischen den Begriffen *implementatives Datenbankmodell* und *konzeptionelles Datenmodell* unterschieden. In den nachfolgenden Kapiteln werden diese näher beschrieben.

2.4.1 Implementative Datenbankmodelle

Das *implementative Datenbankmodell* ist ein DBMS abhängiges Modell und beschäftigt sich mit der Auslegung der Datenspeicherung in Datenbanken sowie der zu erzeugenden Datenstrukturen und deren Beziehungen zueinander [vgl. Geisler 2009, S. 54; Schwarze 2000, S. 232; Schneider 2007a, S. 292]. Die nachfolgend beschriebenen Datenbankmodelle sind den implementativen Datenbankmodellen zuzuordnen. Dies ist in der Literatur unter anderem bei Geisler [2009, S. 54, S. 67] nachzulesen.

Hierarchisches Datenbankmodell

Im hierarchischen Datenbankmodell werden die Entitäten als Datensätze dargestellt. Dabei beschreibt ein Datensatz eine Zusammenfassung von Daten, die zu einer Entität gehören (z.B. Artikelnummer, Bezeichnung). Die Darstellung erfolgt in Form einer Baumstruktur, in welcher die Datensätze miteinander verknüpft sind (vgl. Abbildung 2-8). Von jedem Datensatz existiert ein eindeutig definierter Weg zum obersten Datensatz, mit Ausnahme der obersten Stufe (Wurzel). Der Pfad vom obersten Datensatz zu den anderen Datensätzen wird als Zugriffspfad bezeichnet. Aus der hierarchischen Struktur ergeben sich zwangsläufig übergeordnete und untergeordnete Beziehungen. In diesem Datenbankmodell werden Entitätstypen, Entitäten und Attribute gleichrangig behandelt [vgl. Stahlknecht und Hasenkamp 2005, S. 170; Mayer 1988, S. 63].

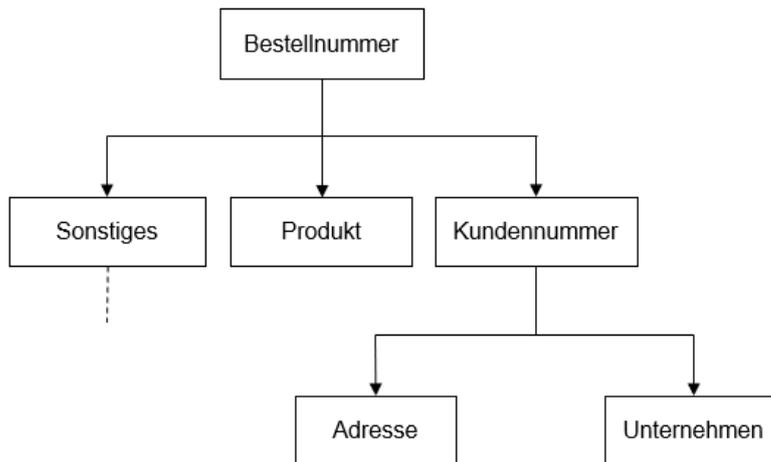


Abbildung 2-8: Hierarchisches Datenbankmodell [vgl. Stahlknecht und Hasenkamp 2005, S. 170]

Netzwerk Datenbankmodell

Der Aufbau eines Netzwerk-Datenbankmodells erscheint komplexer als beim hierarchischen Datenmodell. Von jeder Kante können beliebig viele Verweise ausgehen, weshalb sich die Beziehungen in beliebig viele Richtungen entwickeln können [vgl. Mayer 1988, S. 64]. Demnach kann ein Datensatz beliebig viele Nachfolger haben und im Gegensatz zum hierarchischen Datenmodell mehrere Vorgänger besitzen (vgl. Abbildung 2-9). Auch in diesem Datenbankmodell werden Entitätstypen, Entitäten und Attribute gleichrangig behandelt [vgl. Stahlknecht und Hasenkamp 2005, S. 171f].

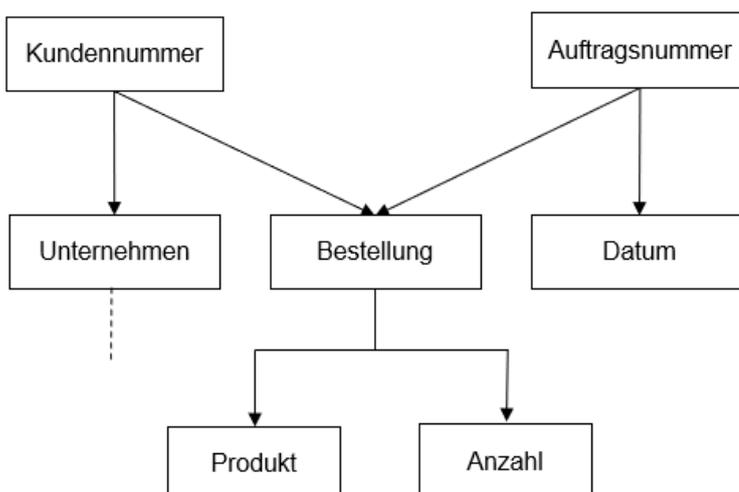


Abbildung 2-9: Netzwerk Datenbankmodell [vgl. Stahlknecht und Hasenkamp 2005, S. 172]

Relationales Datenbankmodell

Das relationale Datenbankmodell wird für die Entwicklung von relationalen DBS verwendet. In diesem Datenbankmodell werden alle Daten in Tabellen mit einer festen Anzahl von Spalten und einer beliebigen Anzahl von Zeilen hinterlegt. Angelehnt an die Chen-Notation, ist die *Tupel* als Entität und die *Relation (Tabelle)* als Entitätstyp zu verstehen. Darüber hinaus ist die Beschreibung von hierarchischen Strukturen sowie Netzwerkstrukturen mit diesem Datenbankmodell möglich [vgl. Stahlknecht und Hasenkamp 2005, S. 172]. Die Verbindung zwischen den Tabellen wird durch sogenannte *Primär- und Fremdschlüssel* realisiert (vgl. Abbildung 2-10). Die Bedeutung des Primärschlüssels kommt dann zum Tragen, wenn man zwei Tabellen in Beziehung setzt. Was bei der Verknüpfung in der einen Tabelle als Primärschlüssel bezeichnet wird, ist in einer weiteren Tabelle der Fremdschlüssel. Dadurch können Datensätze miteinander in Verbindung treten [vgl. Geisler 2009, S. 63].

Der Vorteil des relationalen Datenbankmodells besteht darin, dass dieses im Gegensatz zu den bereits vorgestellten Datenbankmodellen eine strukturelle Unabhängigkeit vorweist (Daten werden in Form von Tabellen abgelegt). Der Anwender der relationalen Datenbank beschäftigt sich ausschließlich mit der logischen Struktur der Daten [vgl. Geisler 2009, S. 64]. Aufgrund der Tatsache, dass Daten in Tabellen abgelegt sind, ist als Nachteil der Mangel an der visuellen Darstellung der Beziehungen zwischen den Daten zu nennen [vgl. Burkert 1995, S 19].

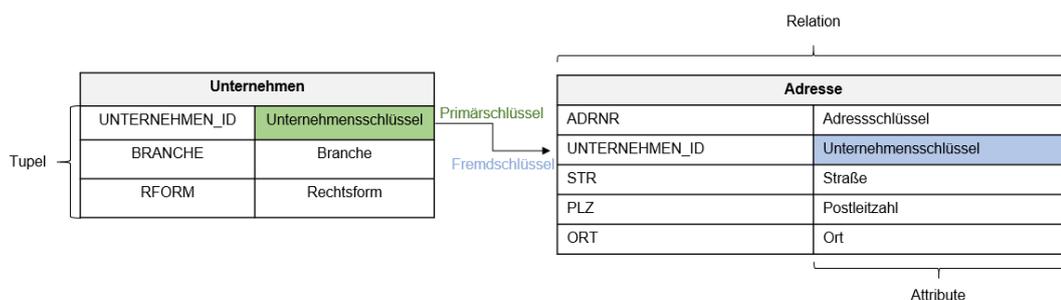


Abbildung 2-10: Relationales Datenbankmodell [vgl. Geisler 2009, S. 63]

Objektorientiertes Datenbankmodell

Beim objektorientierten Datenbankmodell handelt es sich um eine Methode, bei der versucht wird, Sachverhalte mit Objekten zu modellieren. Die Definition der Objekte erfolgt über sogenannte Klassen. Durch die Beschreibung von Objektklassen können komplexe Datenstrukturen in einer DB verwaltet werden. Eine Klasse besteht prinzipiell aus bestimmten *Attributen* und *Methoden*. Die Methode beschreibt hierbei das Verhalten eines Objekts, wodurch die Zustände eines Objekts verändert werden können [vgl. Geisler 2009,

S. 70 - 75]. Im Grunde bilden Klassen einen Bauplan mit festen Attributen und Methoden, aus denen unterschiedliche Objekte gebildet werden können. In Abbildung 2-11 soll dies veranschaulicht werden.

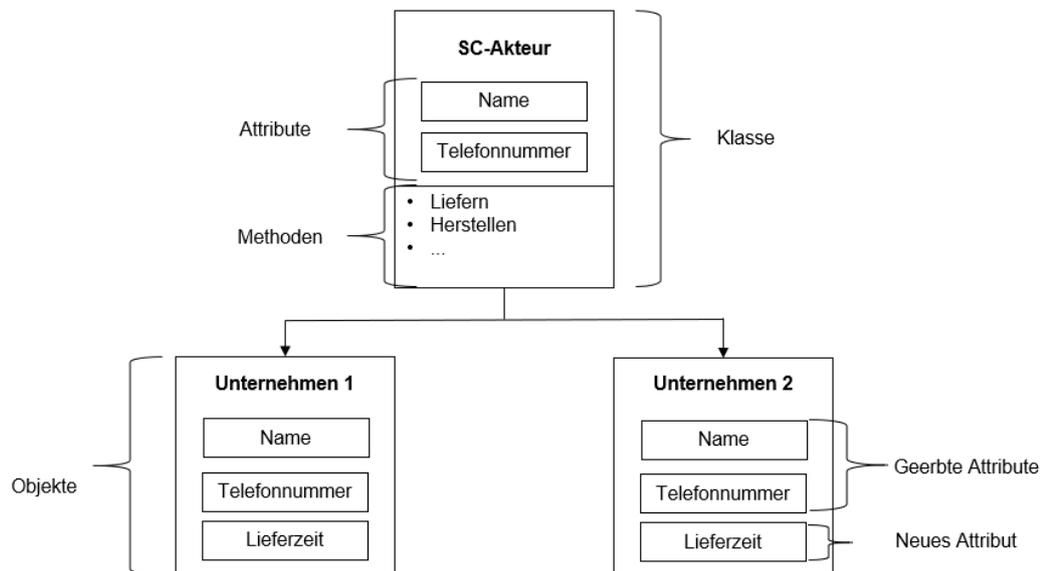


Abbildung 2-11: Objektorientiertes Datenbankmodell [vgl. Geisler 2009, S. 74]

Aus einer Klasse lassen sich Objekte (z.B. Unternehmen 1, Unternehmen 2) erstellen. Alle Objekte haben bestimmte Attribute, wie z.B. der Name des Unternehmens oder die Telefonnummer, welche sich von Objekt zu Objekt unterscheiden. Allerdings können die Objekte die gleichen Methoden (z.B. liefern, herstellen) ausführen. Methoden helfen dabei, auf die Daten eines Objekts zugreifen zu können.

Abschließend ist auszuführen, dass in der Literatur weitere Datenbankmodelle existieren. Allerdings bilden diese Erweiterungen der hier aufgeführten Datenbankmodelle.

2.4.2 Konzeptionelle Datenmodelle

Die informationstechnische Verarbeitung von Daten erfordert eine Strukturierung von Daten. Diese Aufgabe wird als Datenmodellierung bezeichnet. Dabei werden die Daten eines bestimmten Anwendungsbereichs in einem *konzeptionellen Datenmodell* beschrieben [vgl. Lemke et al. 2017, S. 349]. Das konzeptionelle Datenmodell bildet die Grundlage für die Ableitung der bei der Datenspeicherung und Datenmanipulation benötigten Datenstrukturen [vgl. Bodendorf 2006, S. 12]. Als Datenmanipulation sind alle Prozesse gemeint, um

zu einem Datenbestand neue Elemente hinzuzufügen, alte zu löschen oder zu ändern [vgl. Sieberichs und Krüger 1993, S. 609].

Das konzeptionelle Datenmodell ist als Analysemodell zu verstehen, das betriebswirtschaftliche Anforderungen an ein IS beschreibt, um die innerbetriebliche Kommunikation zu fördern. Ein Analysemodell verfolgt die Absicht, eine zweckgerichtete Kommunikation zu ermöglichen [vgl. Schneider 2007a, S. 292]. Dies erfolgt unabhängig von der physischen Datenorganisation, von den spezifischen Erfordernissen des DBMS sowie den einzelnen betrieblichen Anwendungen [vgl. Schwarze 2000, S. 226].

Darauf aufbauend wird Schneider [2007a, S. 152; S. 293] zitiert, der das konzeptionelle Datenmodell ebenfalls als Analysemodell versteht. In seiner Ausführung definiert er ein konzeptionelles Datenmodell als fachbegriffliches Datenmodell zur Beschreibung von Fachbegriffen und ihren Beziehungen. Ein Fachbegriffsmodell lässt sich dahingehend definieren, dass es sich hierbei um ein Modell handelt, das dem konstruierten Ergebnis des Terminologiemanagements entspringt und die klar definierte, konsensfähige Begriffswelt repräsentiert (siehe auch Kapitel 2.4.2) [vgl. Schneider 2007a, S. 285]. Die Aufgaben eines konzeptionellen Datenmodells liegen somit darin, die Beziehungen zwischen den Fachbegriffen und die Rollen, in denen die Fachbegriffe dabei auftreten, zu konkretisieren. Im weiteren Verlauf der Arbeit wird jedoch nicht explizit zwischen Fachbegriff und Begriff unterschieden.

Zunächst werden die Darstellungsmöglichkeiten und Bezeichnungen separat vorgestellt und nicht an ein bestimmtes konzeptionelles Datenmodell angeknüpft. Dadurch wird eine Vergleichsmöglichkeit verschiedener Beschreibungsmittel sichergestellt. Aufgrund der Vielzahl an Literaturen zur Thematik Datenmodell und Datenmodellierung sowie der daraus resultierenden Vielzahl an Definitionen wird zur weiteren Erläuterung der Entity-Relationship (ER)-Formalismus nach Chen gewählt. Dies soll dem Zweck dienen, Irritationen bei der Beschreibung von Begrifflichkeiten zu vermeiden.

Entität

Eine Entität ist ein identifizierbares Objekt der realen Welt und bildet eine zentrale Darstellung in einem konzeptionellen Datenmodell [vgl. Unland und Pernul 2015, S. 34]. Durch eine Typisierung werden Entitäten zu einem übergeordneten *Entitätstypen* zusammengefasst. Darunter ist zu verstehen, dass Entitäten mit ähnlichen Attributen zusammengefasst werden [vgl. Gadatsch 2017, S. 9]. Beispielsweise können in einem konzeptionellen Datenmodell mehrere konkrete LKWs (Entitäten) unter dem Begriff *Transportmittel* (Entitäts-

typ) zusammengefasst werden. Somit lässt sich ausführen, dass Entitäten eine konkrete und Entitätstypen eine generische Bezeichnung darstellen. Entitätstypen werden in Abbildung 2-12 als Rechtecke dargestellt. Die Bezeichnung eines Entitätstyps (z.B. Transportmittel) wird in das Rechteck eingetragen.

Attribut

Merkmale die eine Entität näher beschreiben, werden *Attribute* genannt. Attribute können künstlich erdachte oder tatsächlich beobachtete Merkmale darstellen [vgl. Vossen 2008, S. 61]. Jede Entität besitzt für jedes seiner Attribute einen bestimmten Wert, auch bekannt als *Attributwert*. Ein Attributwert ist das Ergebnis einer präzisierenden Beschreibung eines Attributs [vgl. Unland und Pernul 2015, S. 34]. In Tabelle 2-4 soll der Unterschied zwischen einem Attribut und dem Attributwert verdeutlicht werden. Des Weiteren wird zwischen verschiedenen Arten von Attributen unterschieden, die im Folgenden näher erläutert werden.

Tabelle 2-4: Beispielhafte Darstellung von Attribut und Attributwert [vgl. Jarosch 2016, S. 34]

Attribut	Attributwert
Lieferant	Gummi GmbH
Lieferantenummer	42.42.42
Branche	Reifen
Produkte	Autoreifen, Flugzeugreifen
Telefonnummer	0123456789

An erster Stelle sind *zusammengesetzte Attribute* zu nennen. Diese lassen sich in kleinere Komponenten zerlegen. Im Umkehrschluss kann ausgeführt werden, dass zusammengesetzte Attribute eine Kombination aus mehreren inhaltlich zusammengehörenden Attributen darstellt. Als Beispiel ist als Attribut die *Adresse* zu nennen. Die Adresse selbst bildet das Attribut einer Entität (z.B. Person, Unternehmen). Die Adresse kann jedoch wiederum die Attribute Straße, Hausnummer und Ort enthalten. Den Kontrast dazu, bildet ein *atomares (einfaches) Attribut*. Dieses enthält keine weiteren Attribute. Es lässt sich somit nicht weiter unterteilen. Als Beispiel ist das atomare Attribut *Vorname* zu nennen [vgl. Elmasri und Navathe 2002, S. 63].

Als nächstes sind *einwertige Attribute* zu nennen, die zu einem bestimmten Zeitpunkt nur einen Wert für eine bestimmte Entität annehmen können. Demgegenüber stehen *mehrwertige Attribute*, die mehrere mögliche Werte annehmen können. Als Beispiel für einwer-

tige Attribute ist die Lieferantenummer zu nennen, da jedes Unternehmen in der Regel nur mit einer Lieferantenummer in einer DB hinterlegt ist. Für mehrwertige Attribute ist die Telefonnummer als Beispiel zu nennen, da ein Unternehmen mehrere Telefonnummern haben kann [vgl. Unland und Pernul 2015, S. 34; Voss 2008, S. 62 f.; Gadatsch 2017, S. 21f].

Des Weiteren sind *abgeleitete Attribute* zu nennen. Abgeleitete Attribute werden aus anderen Attributen oder Entitäten berechnet. Bezogen auf eine relationale DB, ist hierbei als Beispiel die Summe aus mehreren Spalten einer Tabelle zu nennen [vgl. Unland und Pernul 2015, S. 34f].

Als letztes sind Attribute mit identifizierenden Merkmalen zu nennen, auch bekannt als *identifizierende Attribute* oder *Schlüsselattribute*. Schlüsselattribute können beispielsweise durch eine fortlaufende Nummerierung generiert werden. Als typisches Beispiel ist die Kundennummer zu nennen [vgl. Unland und Pernul 2015, S. 34; Jarosch 2016, S. 34-38; Gadatsch 2017, S. 10]. Attribute werden in Abbildung 2-12 als ovale Kreise dargestellt.

Beziehung

Eine Beziehung beschreibt das Verhältnis zwischen zwei oder mehreren Entitätstypen [vgl. Voss 2008, S. 303]. Beziehungen entstehen nicht wie Entitätstypen aus sich selbst heraus, sondern sind von der Existenz der beteiligten Entitätstypen abhängig [vgl. Becker und Schütte 2004, S. 87-88]. Ähnlich wie Entitäten können Beziehungen ebenfalls Attribute besitzen. Derartige Attribute stellen Merkmale dar, die erst durch eine Beziehung entstehen [vgl. Jarosch 2016, S. 45ff]. Beziehungen werden in Abbildung 2-12 als Rauten dargestellt.

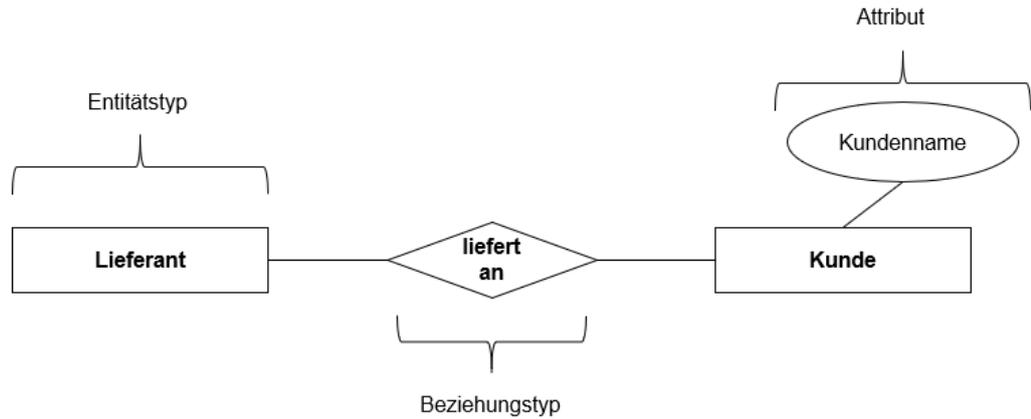


Abbildung 2-12: Darstellung eines Beziehungstyps im ERM [vgl. Jarosch 2016, S. 48f; van Randen et al. 2016, S. 10]

Kardinalität

Die Kardinalität (auch *Multiplizität* genannt) legt über den Beziehungstyp fest, wie viele Entitäten eines Entitätstyps mit Entitäten des anderen Entitätstyps in Beziehung stehen [vgl. Schneider 2007a, S. 243; Herrmann 2018, S. 18]. Kardinalitäten lassen sich in drei Grundformen unterscheiden, wobei M und N für eine beliebige Anzahl größer eins stehen (vgl. Tabelle 2-5) [vgl. Gadatsch 2017, S. 11].

Tabelle 2-5: Kardinalitäten nach Chen [vgl. Gadatsch 2017, S. 11f; Abts und Müller 2011. S. 146f]

Art der Kardinalität	Beispiel
1:1-Beziehung	Ein Lieferant liefert eine Bestellung an einen Kunden.
1:N-Beziehung	Ein Lieferant beliefert mehrere Kunden.
M:N-Beziehung	I) Lieferanten beliefern mehrere Kunden. II) Kunden bestellen bei mehreren Lieferanten.

MIN, MAX-Notation

Mit Bezug auf die in Tabelle 2-5 zusammengefassten Kardinalitäten ist anzumerken, dass dieses Schema unscharf in seiner Beschreibung ist [vgl. Herrmann 2018, S. 33f]. Beispielsweise wäre bei einer 1:N-Beziehung unklar, mit wie vielen Entitäten der Entitätstyp *Lieferanten* mit den Entitäten des Entitätstyps *Kunden* in Verbindung stehen muss. Es wird somit nur eine Einschränkung über die maximale Teilnahme der Entitäten beschrieben (vgl. Abbildung 2-13).



Abbildung 2-13: Kardinalität nach Chen im ERM [vgl. Gadatsch 2017, S. 11]

Um die Beziehung präziser zu formulieren, wird die MIN, MAX-Notation verwendet [vgl. Jarosch 2016, S. 52]. Durch die MIN, MAX-Notation kann zwischen den Entitätstypen *Lieferanten* und *Kunden* Aussage darüber getroffen werden, mit wie vielen Entitäten des Typs *Kunden* eine Entität des Typs *Lieferanten* in Beziehung stehen kann und umgekehrt. Im Folgenden wird die MIN, MAX-Notation beispielhaft anhand Abbildung 2-14 näher beschrieben.

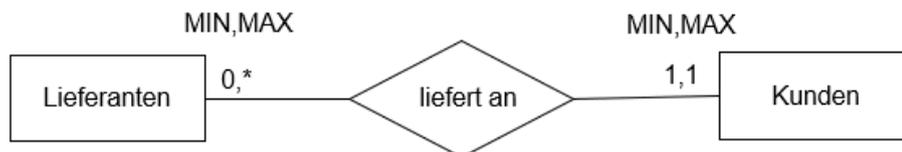


Abbildung 2-14: MIN, MAX-Notation im ERM [vgl. Gadatsch 2017, S. 34]

Der Ausdruck (1,1) sagt aus, dass es für jede Entität des Typs *Kunden* MIN und MAX eine zugeordnete Entität des Typs *Lieferanten* gibt. Umgekehrt sagt der Ausdruck (0,*) aus, dass es keine Einschränkung gibt. Für Entitäten des Typs *Lieferanten* gibt es entweder Null oder beliebig viele zugeordnete Entitäten des Entitätstyps *Kunden*. Somit kann ausgeführt werden, dass die MIN,MAX-Notation eine stärkere Einschränkung (detaillierte Beschreibung) ermöglicht. Aus der MIN, MAX-Notation ergeben sich die vier Grundtypen, die in Tabelle 2-6 zusammengefasst sind.

Tabelle 2-6: Grundtypen der MIN, MAX-Notation [Schneider 2007a, S. 248]

Notation	Beschreibung
(0,1)	Ein Entitätstyp ET-1 <u>kann</u> eine Beziehung zu <u>genau einem</u> Entitätstyp ET-2 haben.
(0,*)	Ein Entitätstyp ET-1 <u>kann</u> eine Beziehung zu <u>einem</u> oder <u>mehreren</u> Entitätstyp(en) ET-2 haben.
(1,1)	Ein Entitätstyp ET-1 <u>muss</u> eine Beziehung zu <u>genau einem</u> Entitätstyp ET-2 haben.
(1,*)	Ein Entitätstyp ET-1 <u>muss</u> eine Beziehung zu <u>einem</u> oder <u>mehreren</u> Entitätstyp(en) ET-2 haben.

Vererbung

Bei der Vererbung erfolgt eine Unterscheidung zwischen einem übergeordneten Entitätstyp (*Supertyp*) und einem untergeordneten Entitätstyp (*Subtyp*) [vgl. Schneider 2007a, S. 252]. Der übergeordnete Entitätstyp umfasst alle Attribute, die für den untergeordneten Entitätstyp wesentlich sind, weshalb in der Literatur auch häufig der Begriff *Basisklasse* fällt. Der Subtyp erbt alle Attribute und Beziehungstypen des Supertyps und kann zusätzlich über weitere Attribute und Beziehungstypen verfügen [vgl. Unland und Pernul 2015, S. 100 f.; Gadatsch 2017, S. 19; Ferstl und Sinz 2013, S. 167]. Häufig fallen in der Literatur die Begriffe *Spezialisierung* und *Generalisierung*, um das zuvor beschriebene Vererbungskonzept zu beschreiben. Im Grunde ist darunter zu verstehen, dass eine Spezialisierung zu Subtypen und eine Generalisierung zu Supertypen führt [vgl. Ferstl und Sinz 2013, S. 166]. In Abbildung 2-15 wird das Prinzip der Vererbung visualisiert.

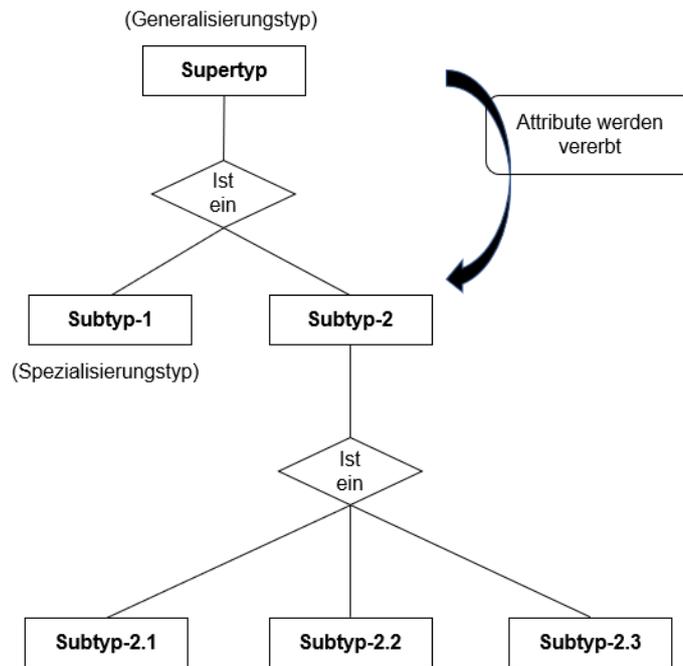


Abbildung 2-15: Aufbau einer Vererbung [vgl. Gadatsch 2017, S. 19; Schneider 2007a, S. 254]

Aggregation

Unter einer Aggregation versteht man ein Konzept, nach dem durch die Kombination von Entitätstypen ein neuer zusammengesetzter Entitätstyp (*Aggregat*) entsteht (vgl. Abbildung 2-16). Die Entitätstypen, die sich zum Aggregat kombinieren, heißen *Komponenten* [vgl. Elmasri und Navathe 2002, S. 126]. Im Grunde stellt die Aggregation eine übergeordnete Ebene dar.

Darüber hinaus ist auszuführen, dass auch ein Beziehungstyp unter Umständen als neuer Entitätstyp verstanden werden kann. Weist ein Beziehungstyp zusätzliche Attribute auf, die inhaltlich durch die Kombination der Entitätstypen charakterisiert sind, so kann dieser Beziehungstyp als neuer Entitätstyp verstanden werden [vgl. Unland und Pernul 2015, S. 69; Schwarze 2000, S. 231; Abts und Müller 2011, S. 149].

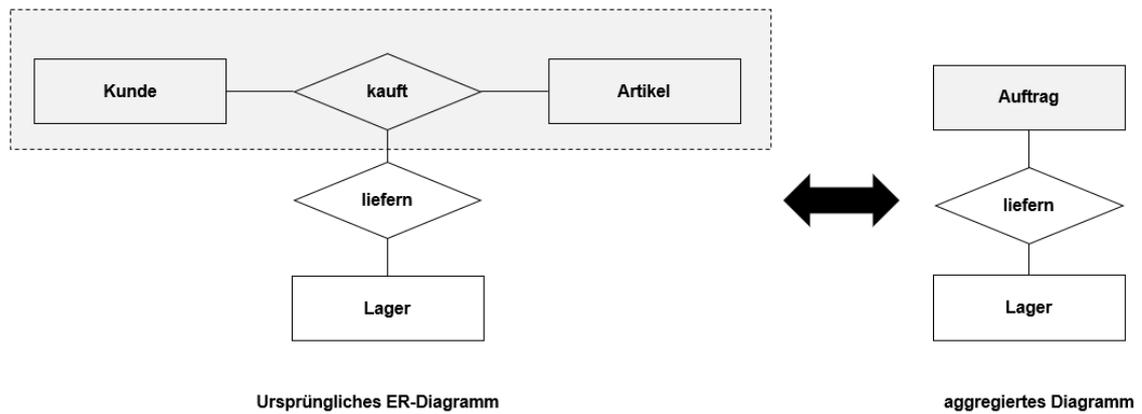


Abbildung 2-16: Aggregation einer Beziehung und Entitätstypen [vgl. Schwarze 2000, S. 231]

Komposition

Eine strenge Form der Aggregation ist die *Komposition* [vgl. Becker et al. 2012, S. 11]. Hierbei sind die Komponenten eines Aggregats von dessen Existenz abhängig. Somit ist der Beziehungstyp zwischen dem Aggregat und den Komponenten auf Seiten der Komponenten mit (0,1) oder (1,1) zu deuten [vgl. Schneider 2007a, S. 249]. Als Beispiel einer Komposition dient eine Rechnung, die sich aus mehreren Rechnungspositionen zusammensetzt. Die Rechnungspositionen sind ohne Rechnung nicht existenzfähig [vgl. van Randen et al. 2016, S. 11f].

Viele dieser Konzepte finden sich in ähnlicher Ausführung auch in Kapitel 2.5 und Kapitel 2.6 wieder, weshalb speziell in diesem Kapitel die Konzepte genauer ausgeführt wurden.

Bei der Auswahl eines Beschreibungsverfahrens für das konzeptionelle Datenmodell muss darauf geachtet werden, welcher Formalismus als angemessen erscheint. So wird das ERM in erster Linie zur Darstellung des relationalen Datenbankmodells verwendet, während die Unified Modeling Language (UML) vornehmlich für das objektorientierte Datenbankmodell verwendet wird [vgl. Krcmar 2015, S. 46]. Im Folgenden sollen die zwei bekannten Vertreter der Datenmodellierung näher beschrieben werden.

Entity-Relationship-Modell

In Anlehnung an die Definition zur Kategorisierung der Datenbankmodelle (siehe auch Kapitel 2.4) sowie nach Aussage einiger Autoren – repräsentativ Geisler [2009, S. 54, S. 67], Schneider [2007a, S. 308f] und Burkert [1995, S. 19] – wird das ERM den konzeptionellen Datenmodellen zugeordnet.

Mit Hilfe des ERM wird ein Entity-Relationship-Diagramm (ERD) erstellt, welches auch für fachfremde Personen verständlich dargestellt werden kann [vgl. Geisler 2009, S. 54]. Somit können in der frühen Phase des Datenbankentwurfs Auftraggeber und Datenbankentwickler verständlicher miteinander kommunizieren (siehe auch Kapitel 2.2). Sobald die Datenstruktur auf semantischer Ebene erfasst wurde, wird dieses in ein *implementatives Datenbankmodell* überführt (siehe auch Kapitel 2.3 und 2.3.1). In der Regel wird das ERM in das relationale Datenbankmodell überführt [vgl. Geisler 2009, S. 68; Schneider 2007a, S. 308]. Im Folgenden werden die grundlegenden Darstellungsmöglichkeiten des ERM dargestellt (vgl. Abbildung 2-17).

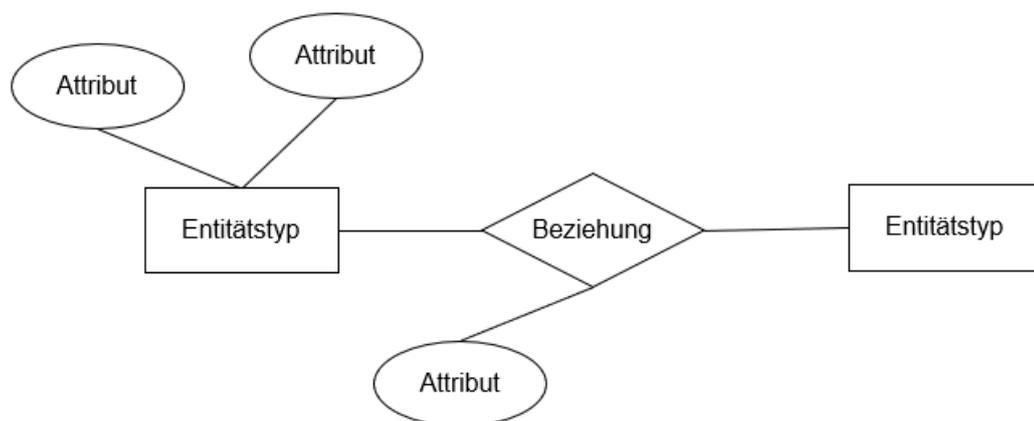


Abbildung 2-17: Notation im ERD [vgl. Gadatsch 2017, S. 10-26]

Unified-Modeling-Language

Unterstein und Matthiessen [2013, S. 160] führen aus, dass das UML nur dem konzeptuellen Entwurf diene und dass der logische Entwurf daraus abzuleiten sei, um die Implementierung von (implementativen) Datenbankmodellen zu ermöglichen. Nach dieser Definition ist somit das UML ähnlich wie das ERM auch den konzeptionellen Datenmodellen zuzuordnen.

Für die Entwicklung objektorientierter Datenbankmodelle ist der ER-Formalismus nur eingeschränkt tauglich. Dies liegt darin begründet, dass der ER-Formalismus die Festlegung der Methoden nicht erlaubt, um das Verhalten eines Objekts zu beschreiben (siehe auch Kapitel 2.3.1). Für diesen Fall wird die *Unified Modeling Language* (UML) gewählt [vgl. Staud 2005, S. 235]. Die UML lässt sich in zwei Beschreibungsarten unterteilen. Unterschieden wird zwischen *Struktur-* und *Verhaltensdiagrammen*, die im Folgenden näher beschrieben werden.

In *Strukturdiagrammen* werden statische, zeitunabhängige Elemente modelliert, die den logischen Aufbau eines Prozesses beschreiben. Hierbei wird zwischen sieben Arten von Strukturdiagrammen unterschieden. Das wohl bekannteste Strukturdiagramm ist das Klassendiagramm, das die Beziehungen, Attribute und Methoden in der objektorientierten Programmiersprache darstellt [vgl. Crezelius 2015, S. 1f].

Ein *Verhaltensdiagramm* ermöglicht eine dynamische Sicht auf ein System und eignet sich zur Darstellung von Prozessen, zeitlichen Abläufen oder Interaktionen zwischen prozessbeteiligten Einheiten. Repräsentativ ist hierbei das Aktivitätsdiagramm zu nennen, welches ähnlich wie bei der ereignisgesteuerten Prozesskette (EPK), kontrollflussorientierte Modelle darstellt [vgl. Crezelius 2015, S. 2f].

Da die Bezeichnungen im UML geringfügig von der Chen-Notation abweichen, werden diese kurz erläutert. In der UML werden Klassendiagramme verwendet, um die Beziehungen, Attribute und Methoden von Klassen zu beschreiben [vgl. Alpar et al. 2016, S. 380]. Eine Klasse ist die Verallgemeinerung gemeinsamer Objekte aus der realen Welt. Als Beispiel ist die Klasse *Lieferanten* zu nennen. Die Klasse Lieferanten beinhaltet alle Informationen, die Lieferanten allgemein haben müssen (z.B. Name, Adresse, Lieferantenummer). Ein Objekt wäre eine konkrete Beschreibung eines Lieferanten (z.B. Lieferant: Mustermann GmbH). In Klassendiagrammen werden Klassen miteinander in Beziehung gesetzt. Diese werden durch ein Rechteck dargestellt, das in die drei Bereiche *Klassenname*, *Attribute* und *Methoden* aufgeteilt ist [vgl. Alpar et al. 2016, S. 384f]. Für eine grobe Übersicht werden auch nur Rechtecke verwendet, die den Klassennamen beinhalten (vgl. Abbildung 2-18). Der Bereich Attribute beinhaltet den Attributnamen (z.B. Lieferantenummer), der durch einen Doppelpunkt von einem Datentyp (z.B. String, Integer) des Attributes getrennt wird. Der letzte Bereich beschreibt die Methoden einer Klasse (auch Operationen genannt). Eine Methode könnte beispielsweise der Begriff *Liefere* sein, wodurch das Verhalten einer Klasse beschrieben wird [vgl. Alpar et al. 2016, S. 390].

Zusammenfassend ist auszuführen, dass die UML eine standardisierte Modellierungssprache für die Konstruktion einer Software darstellt. Es besteht die Möglichkeit, ein ERM in das Klassendiagramm des UML zu überführen [vgl. Meier und Kaufmann 2016, S. 87].

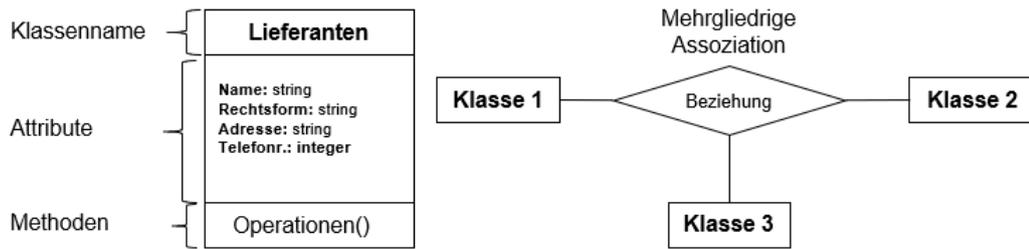


Abbildung 2-18: Darstellung des UML im Klassendiagramms [vgl. van Randen et al. 2016, S. 5-17]

Die zuvor beschriebenen Datenbank- und Datenmodelle sind nachfolgend noch einmal in Tabelle 2-7 kategorisiert zusammengefasst.

Tabelle 2-7: Kategorisierung zwischen Datenbank- und Datenmodellen

Kategorie	
Implementative Datenbankmodelle	Hierarchisches Datenbankmodell
	Netzwerk Datenbankmodell
	Relationales Datenbankmodell
	Objektorientiertes Datenbankmodell
Konzeptionelle Datenmodelle (Datenmodellierung im jeweiligen Formalismus)	Entity-Relationship-Modell
	Unified Modeling Language

Im nachfolgenden Kapitel wird das konzeptionelle Datenmodell in seiner besonderen Form als generisches Datenmodell beschrieben.

In der Literatur äußert sich die Mehrheit der Autoren darüber, dass ein generisches Datenmodell eine besondere Konstruktionsweise eines konzeptionellen Datenmodells darstellt. Allen voran wird Schneider [2007a, S. 406] genannt, der ein generisches Datenmodell als ein spezielles konzeptionelles Datenmodell bezeichnet. Etwas präziser führt Scheller [1993, S. 87] aus, dass ein generisches Datenmodell als anwendungsunabhängiges semantisches Datenmodell zu verstehen ist. Dabei werden vordefinierte Objekte zur Verfügung gestellt, deren Definition anwendungsspezifisch verfeinert werden muss. Reichenberger [2010, S. 182] führt aus, dass die Fachbegriffe in einem generischen Modell nicht spezifisch modelliert sind.

2.5 Generisches Datenmodell

Datenmodelle müssen sich als flexibel genug erweisen, um sich den Veränderungen zeitnah und kostengünstig anpassen zu können [vgl. Schneider 2007a, S. 3]. Generische Datenmodelle müssen bestimmte Erwartungen erfüllen, um ihren Allgemeingültigkeitsanspruch gerecht zu werden. Die Anforderungen zur Konstruktion eines generischen Datenmodells leiten sich aus den folgenden Qualitätskriterien ab (vgl. Abbildung 2-19). Die dort aufgeführten Anforderungen stellen daher einen wesentlichen Bestandteil zur Erfüllung der generischen Darstellung eines konzeptionellen Datenmodells dar [vgl. Schneider 2007a, S. 400; S. 412].

Qualitätskriterien	
Kategorie 1: Modellgestaltung	Kategorie 2: Änderbar- und Verwendbarkeit eines Modells
Vollständigkeit	Flexibilität
Korrektheit	Stabilität
Minimalität	Modularität
Verständlichkeit	Wiederverwendbarkeit
Schlichtheit	Integrationsfähigkeit

Abbildung 2-19: Qualitätskriterien für die Anforderungen zur Konstruktion eines generischen Datenmodells [vgl. Schneider 2007a, S. 412]

Die in Abbildung 2-19 aufgeführten Kategorien sind so zu verstehen, dass sich die Auflistung in Spalte „Kategorie 1“ hauptsächlich auf die allgemeine Modellgestaltung eines konzeptionellen Datenmodells bezieht, während die Auflistung der Spalte „Kategorie 2“ zusätzlich die Veränderbarkeit eines konzeptionellen Datenmodells beschreibt, wodurch sich ein generisches Datenmodell auszeichnet [vgl. Schneider 2007a, S. 412]. Im Folgenden werden die einzelnen Anforderungen näher beschrieben:

- Die *Vollständigkeit* ist die Grundlage aller wichtigen Bestandteile der Anwendungsbereiche eines konzeptionellen Datenmodells. Daher obliegt die Vollständigkeit des einzelnen Datenmodells jedem Hersteller desselbigen [vgl. Schneider 2007a, S. 398].

- Unter *Korrektheit* ist, ein Vernunftgeleitetes und zweckausgerichtetes Denken und Handeln zu verstehen. Dies beinhaltet die absichtliche Auswahl von Gründen sowie die Entscheidung für Gründe, die als vernünftig gelten, um bestimmte Ziele zu erreichen. Dies kann je nach Anwendungsbereich und je nachdem, was man als vernünftig betrachtet, unterschiedliche Bedeutung haben [vgl. Schneider 2007a, S. 398].
- Die *Minimalität* eines Datenmodells umfasst die für jedes zu erstellende konzeptionelle Datenmodell notwendigen Inhalte. Dies bedeutet, dass jedes Datenmodell individuell für sich zu betrachten ist und um seine Funktionalität zu erhalten, nicht verändert werden kann [vgl. Schneider 2007a, S. 398].
- Unter *Verständlichkeit* eines Datenmodells ist zu verstehen, dass jeder Anwender (auch aus unterschiedlichen Fachbereichen) den Inhalt eines Datenmodells verstehen kann [vgl. Schneider 2007a, S. 398].
- Die *Schlichtheit* eines Datenmodells umfasst die zum notwendigen Verständnis wichtigsten Inhalte eines Datenmodells [vgl. Schneider 2007a, S. 399].
- Die *Flexibilität* sagt aus, dass jedes Datenmodell für sich jederzeit in seiner Anforderung anpassungsfähig sein soll und muss [vgl. Schneider 2007a, S. 399].
- Die *Stabilität* bedeutet, dass das auf dem Datenmodell basierende Grundgerüst nicht verändert werden darf und dauerhaft sein muss [vgl. Schneider 2007a, S. 399].
- Die *Modularität* bedeutet, dass das Datenmodell in einzelne Bereiche aufgeteilt werden kann, ohne im Ganzen verändert zu werden [vgl. Schneider 2007a, S. 400].
- Die *Wiederverwendbarkeit* bedeutet, dass das jeweilige Datenmodell in seiner Grundkonzeption auch in anderen Bereichen angewendet werden kann.
- Die *Integrationsfähigkeit* bedeutet, dass Verbindungen geschaffen werden müssen, um Teildatenmodelle zu einem Ganzen zusammenzuführen [vgl. Schneider 2007a, S. 400].

Der Fokus für die generische Gestaltung eines Datenmodells liegt in den Anforderungen der zweiten Kategorie. Diese Kategorie beschreibt nicht nur wie ein Datenmodell aufgebaut ist, sondern auch, wie sich ein Datenmodell verhält. Darauf aufbauend führt Schneider [2007a, S. 413f] aus, dass die Anforderungen der Flexibilität und Stabilität als übergeordnete Anforderungen anzusehen sind, aus denen sich die anderen Anforderungen der zweiten Kategorie ableiten lassen bzw. sich daraus ergeben. Ein Datenmodell, das zugleich flexibel und stabil ist, ist auch wiederverwendbar, integrationsfähig und modular. Dabei ermöglicht die Flexibilität eine Veränderung im Datenmodell durch einfache Datenmodellmodifikationen und die Stabilität eine Veränderung ohne eine Datenmodellmodifikation.

Somit lässt sich ausführen, dass ein generisches Datenmodell nur den Aspekt der Allgemeingültigkeit erfüllen kann, wenn es sich als flexibel und stabil erweist. Aus dieser Erkenntnis folgt, dass ein generisches Datenmodell sich direkt durch die Flexibilität sowie Stabilität und indirekt durch Wiederverwendbarkeit, Integrationsfähigkeit und Modularität auszeichnet [vgl. Schneider 2007a, S. 416].

Zusammenfassend ist auszuführen, dass ein generisches Datenmodell in der frühen DB-Entwurfsphase zum Einsatz kommt (siehe auch Kapitel 2.3 und Kapitel 2.4.2). Die Wiederverwendung eines konzeptionellen Datenmodells erfordert eine generische Darstellung. Hierzu wird eine Methode, die eine generische Konstruktion unterstützt.

In den folgenden Kapiteln werden zwei Konzepte vorgestellt, die in Kombination die generische Darstellung eines konzeptionellen Datenmodells ermöglichen. In der Literatur wird die Kombination dieser beiden Konzepte auch als *Synthesekonzept* bezeichnet. Die Rede ist hierbei vom *Powertyp-Konzept* und dem *Rollenkonzept* (engl. role pattern) [vgl. Schneider 2007a, S. 418]. Powertypen sorgen für die Flexibilität und Stabilität eines generischen Datenmodells. Darüber hinaus ermöglichen Powertypen die einfache Integration von Fachbegriffen, wodurch eine Teilanforderung für die Übernahme von Fachbegriffen ins Datenmodell ermöglicht wird. In Kombination mit dem Powertyp-Konzept ist das Rollenkonzept hauptsächlich für die Beachtung der Diversifikation der Fachbegriffe zuständig. Dabei wird auch eine Teilanforderung erfüllt, um Fachbegriffe ins Datenmodell aufzunehmen [vgl. Schneider 2007a, S. 418]. In Abbildung 2-20 soll das Synthesekonzept dargestellt werden. Dabei ist ersichtlich, welches Konzept, die zuvor beschriebenen Anforderungen erfüllt.

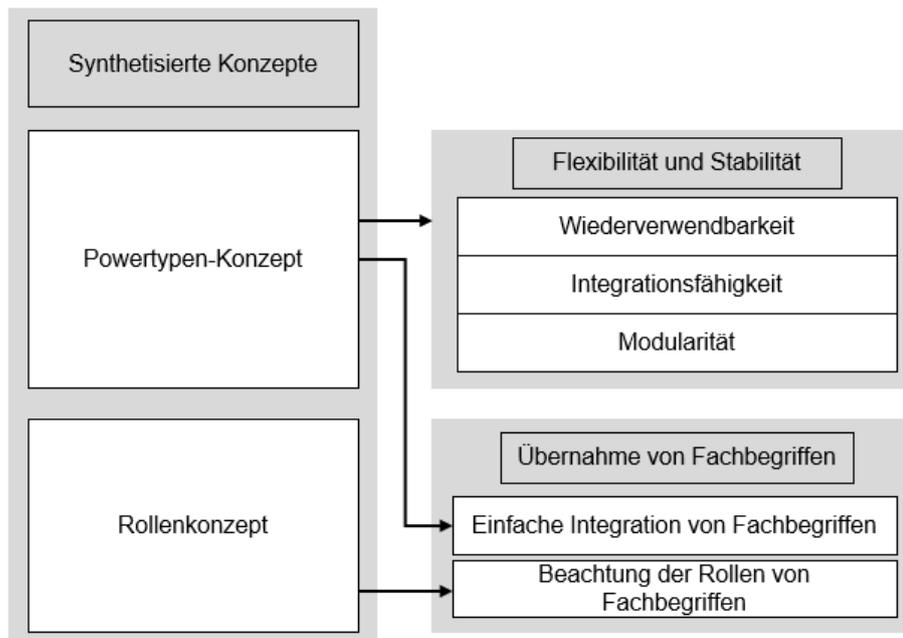


Abbildung 2-20: Zusammenhänge der Konzepte für die generische Darstellung [vgl. Schneider 2007a, S. 418]

2.5.1 Rollenkonzepte

Im alltagssprachlichen Gebrauch beschreibt eine Rolle ein spezifisches Verhaltensmuster, das aus einer Situation heraus entsteht. Rollenkonzepte beschreiben eine Methode, um die Integration von Fachbegriffen ins Datenmodell zu ermöglichen. Schneider [2007a, S. 403] führt diesbezüglich aus, dass die Überlegung der Übernahme von expliziten Fachbegriffen gepaart mit dem Bewusstsein der zwei wesentlichen Aspekte eines Gegenstands hilfreich ist, um die Diskurswelt in ontologisch fundierter Form richtig aufzufassen.

Um das Rollenkonzept anwenden zu können, müssen die zu verwendeten Fachbegriffe nach ihrer *unveränderlichen Wesensart* und ihrem *veränderlichen Verhaltensmuster* kategorisiert werden. Das veränderliche Verhaltensmuster zeigt, wie sich ein Objekttyp in spezifischen Situationen verhält. Beispielsweise kann eine Person analog die Rollen Kunde und Lieferant annehmen [vgl. Schneider 2007a, S. 258f]. Das Nachfolgende Beispiel (vgl. Abbildung 2-21) verdeutlicht die Unterscheidung zwischen *Wesensart* und *Verhaltensmuster*. Dabei stellt der Entitätstyp Person die unveränderliche Wesensart dar, da sie unabhängig von den Beziehungen existieren kann. Aus dem Entitätstyp Person resultieren verschiedene Verhaltensmuster, wie zum Beispiel Lieferant oder Kunde. Dabei kann der Lieferant wiederum verschiedene Verhaltensmuster annehmen, welches durch seine Produktpalette definiert werden könnte.

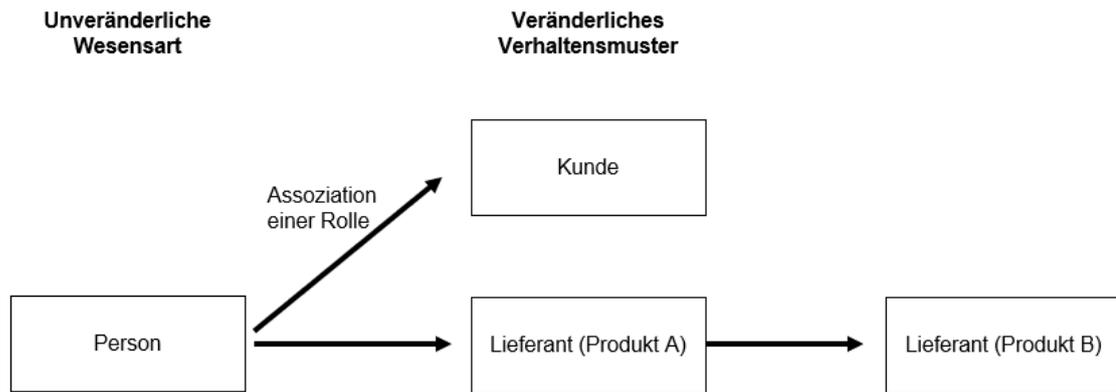


Abbildung 2-21: Einfache Rollendarstellung eines Entitätstyps [vgl. Schneider 2007a, S. 258]

Unabhängig von der Auswahl einer geeigneten Darstellungsmethode ist bei verschiedenen Autoren klar erkennbar, dass ein Entitätstyp nicht nach der Rolle benannt werden sollte, die er in einer bestimmten Beziehung einnimmt. Vielmehr sollte hier die zugrunde liegende Natur des Entitätstyps beschrieben werden [vgl. West 1996, S. 1]. Eine Rolle resultiert somit aus der veränderlichen Assoziation eines Entitätstyps. Dieses Phänomen bezeichnet Thelemann [1996, S. 175] als die Veränderbarkeit der Typzugehörigkeit während der Lebensdauer einer Entität. Dahchour et al. [2002, S. 644-648] beschreiben ein Rollenkonzept durch einen generischen Ansatz. Sie unterscheiden zwischen *object class* und einer *role class*. Die *object class* repräsentiert die grundlegende Bezeichnung beziehungsweise den unveränderlichen natürlichen Typ einer Entität. Die *role class* hingegen beschreibt ein veränderliches Verhaltensmuster. Darauf aufbauend stellt Schneider [2007a, S. 270f] sieben kontextunabhängige Anforderungen auf, die für den Umgang mit Rollen in einem generischen Datenmodell wesentlich und entsprechend zu berücksichtigen sind.

- *Trennung Wesensart und Verhaltensmuster*: Die strikte Trennung zwischen Wesensart und Verhaltensmuster ist zu fordern, auch wenn Synonyme die Abgrenzung nicht immer eindeutig ermöglichen. In der Regel sei eine strikte Trennung immer anwendbar. Jedoch obliegt die Entscheidung beim Modellierer, ob ein Gegenstand als unveränderliche Wesensart (natürlicher Typ) oder als veränderliches Verhaltensmuster (Rolle) im Datenmodell dargestellt wird.
- *Wesensart- oder Rollenbezug der Rolle*: Rollen stehen grundsätzlich in Relation mit anderen Rollen oder Gegenständen.

- *Wesensartige Mehrfacheinnahme und -ablegung einer oder mehrerer Rollen:* Ein Gegenstand kann zu einem bestimmten Zeitpunkt mehrere Rollen annehmen oder ablegen.
- *Existenzunabhängigkeit der Rolle:* Rollen können unabhängig von Gegebenheiten existieren, weshalb diese im Vorfeld einer Assoziation beschrieben werden können.
- *Einsatzabhängigkeit einer Rolle:* Die Rollen eines Gegenstands zeigen sich erst in Ausübung einer Assoziation. Hierbei ist zur Konkretisierung einer Assoziation zwischen *statischer* und *dynamischer* Assoziation zu unterscheiden. Dabei beschreiben statische Assoziationen strukturelle Zusammenhänge zwischen den Objekttypen, während dynamische Assoziationen die Einsatzabhängigkeit einer Rolle bestimmen. Mit Bezug auf die zuvor erläuterten Kategorisierungen eines Gegenstandes nach der unveränderlichen Wesensart und dem veränderlichen Verhaltensmuster ist im Kontext ist auszuführen, dass bei einer dynamischen Assoziation ein Gegenstand (unveränderliche Wesensart) in seiner Rolle (veränderliches Verhaltensmuster) mit einem Gegenstand (unveränderliche Wesensart) in dessen Rolle (veränderliches Verhaltensmuster) eine Verbindung eingeht.
- *Hierarchisierung der Rolle:* Diese Anforderung sagt aus, dass Rollentypen in Vererbungsrelationen stehen können.
- *Einnahme- und Ablegungsunabhängigkeit zwischen den Rollen:* Alle Rollen können unabhängig voneinander angenommen und wieder abgelegt werden.

Auf Grundlage der oben aufgeführten Rollen, wird im Folgenden auf die Arbeit von Fowler [1997, S. 2, S. 2] verwiesen. Er beschreibt fünf Konzepte zur Darstellung von *Rollentypen*. Zu differenzieren sind *Single Role Type*, *Seperate Role Type*, *Role Subtype*, *Role Object* und *Role Relationship*.

Der *Single Role Type* wird für einfache Differenzierungen zwischen Entitätstypen verwendet, die nicht weiter spezifiziert werden. Hierbei werden in einem Entitätstyp sämtliche Rollendaten zusammengefasst. Bei diesem Typ kann es sich um einen Rollentyp handeln, der einen Namen aufweist, welcher die Rolle bezeichnet [vgl. Fowler 1997, S. 4; Schneider 2007a, S. 430].

Beim *Separate Role Type* werden die Rollen nicht in einem Supertyp zusammengefasst, sondern in jedem Rollentyp wiederholt aufgezählt. Schwierigkeiten entstehen, wenn ein

Entitätstyp analog mehrere Rollen annimmt. Als Beispiel ist ein Unternehmen zu nennen, das zeitgleich Lieferant und Kunde ist. Für jede Rolle muss ein separater Entitätstyp erstellt werden (Redundanzproblem). Zusätzlich ist schwer zu definieren, ob sich bestimmte Daten auf den gleichen Gegenstand beziehen (vgl. Abbildung 2-22) [vgl. Fowler 1997, S. 4; Schneider 2007a, S. 430].

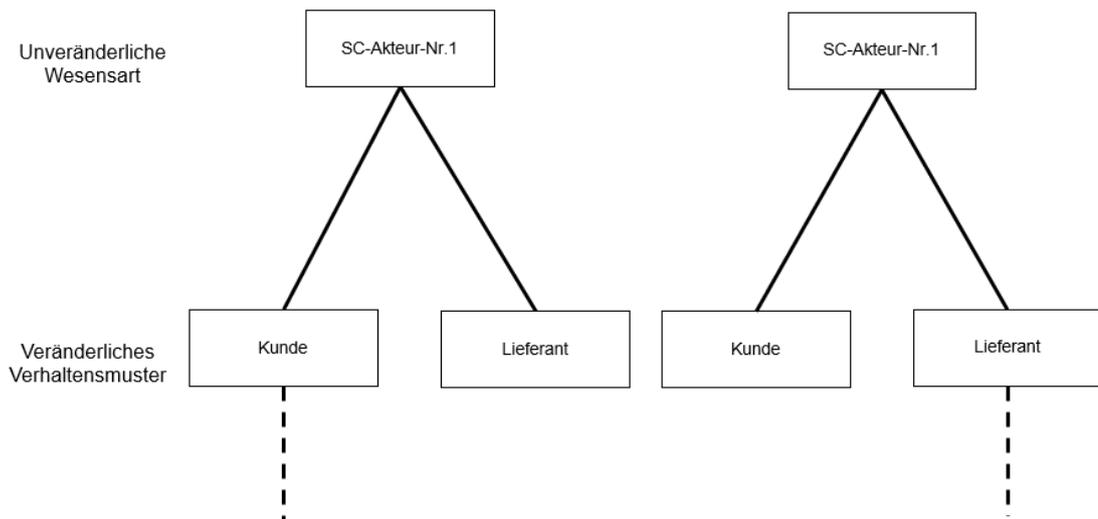


Abbildung 2-22: Separate Role Type [vgl. Fowler 1997, S. 4f]

Besitzen die Entitätstypen keine signifikanten Unterschiede, so findet der *Role Subtype* Verwendung. Dieses Rollenkonzept umgeht das Redundanzproblem des Separate Role Type, indem durch Vererbungsmechanismen (siehe auch Kapitel 2.2) gemeinsame Rollenmerkmale zu einem Supertyp zusammengefasst werden. In den untergeordneten Ebenen werden den Subtypen weitere Rollenmerkmale zugeordnet, um diese zu spezifizieren (vgl. Abbildung 2-23) [vgl. Fowler 1997, S. 5; Schneider 2007a, S. 431].

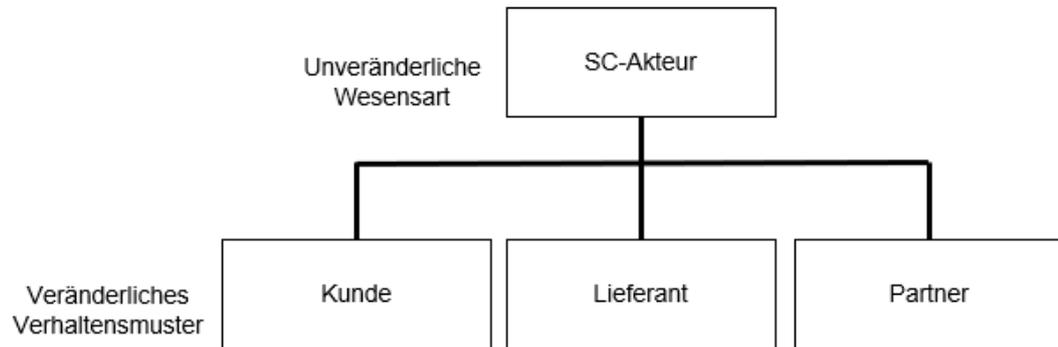


Abbildung 2-23: Role Subtype [vgl. Fowler 1997, S. 6]

Die Alternative zum Role Subtype bildet das *Role Object*. Der wesentliche Unterschied besteht darin, dass die Rollen nicht in einem Vererbungsmechanismus zusammengefasst sind. In diesem Konzept werden die Rollentypen mit einem natürlichen Entitätstypen in Relation gesetzt (vgl. Abbildung 2-24) [vgl. Fowler 1997, S. 13f; Schneider 2007a, S. 431f].

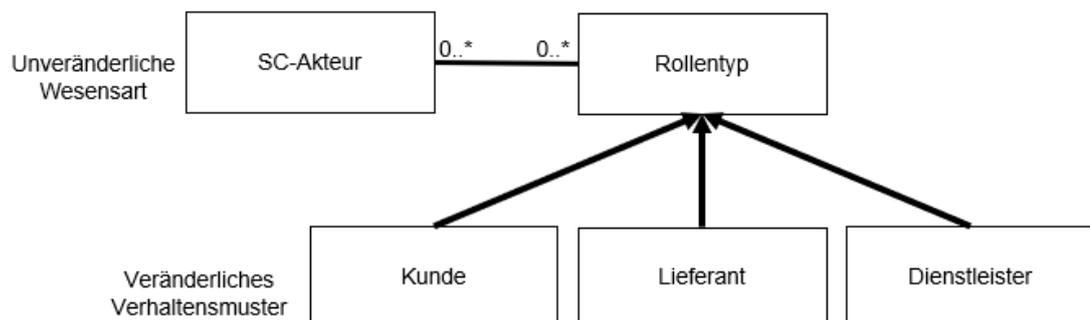


Abbildung 2-24: Role Object [vgl. Fowler 1997, S. 14]

Als letztes Rollenkonzept ist das *Role Relationship* zu nennen. Dieses Rollenkonzept wird verwendet, wenn eine Entität mehr als eine Rolle in verschiedenen Gruppen einer Organisation innehat. Der Unterschied zum Role Object besteht somit darin, dass diesem Pattern der Begriff „Gruppe“ hinzugefügt wird. Dadurch wird nicht nur die Rolle eines Entitätstyps beschrieben, sondern auch der Bezug zur Gruppe hergestellt, bei der diese Rolle ausgeübt wird (vgl. Abbildung 2-25).

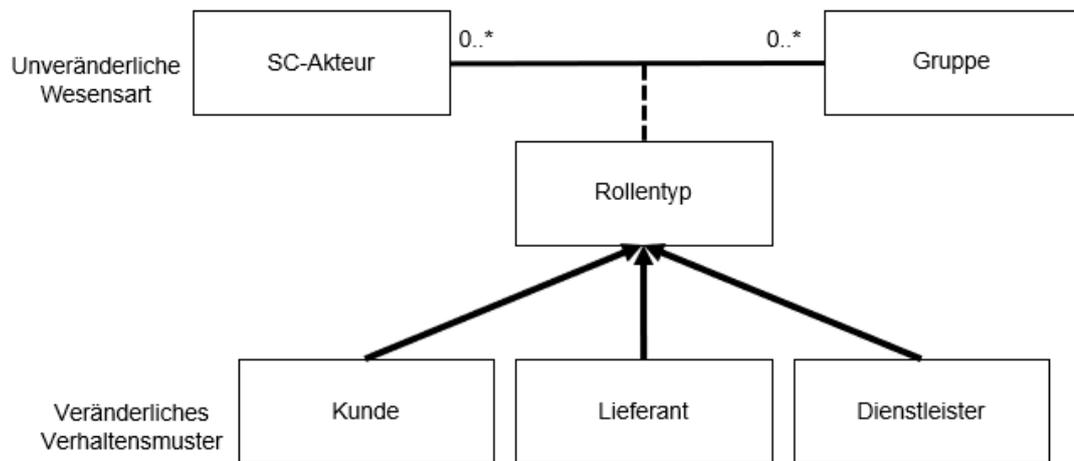


Abbildung 2-25: Role Relationship [vgl. Fowler 1997, S. 17]

Die fehlende Berücksichtigung eines Rollenkonzepts führt zu einer sogenannten Objekt-Reklassifikations-Anomalie (Redundanzproblem). Dieser Sachverhalt tritt auf, falls ein Entitätstyp in einer Beziehung mehrfach erzeugt wird, weil z.B. bei der Modellierung nicht erkannt wird, dass es sich um ein veränderliches Verhaltensmuster und nicht um eine neue Wesensart handelt [vgl. Chu und Zhang 1997, S. 258]. Im Grunde wird dieser Sachverhalt auch durch Abbildung 2-22 beschrieben. Sobald eine Person einen Rollenwechsel durchführt und dies nicht als solches erkannt wird, so werden zwei separate Entitätstypen modelliert [vgl. Schneider 2007a, S. 262]. Dadurch entstehen die Entitätstypen Person-Kunde und Person-Lieferant, obwohl sich an der Person selbst nichts geändert hat. Aus verschiedenen Literaturen geht hervor, dass ein Klassendiagramm als geeignete Methode der Rollenbeschreibung zu zählen ist (siehe auch Kapitel 2.3.2). Dies liegt darin begründet, dass in „einfachen Beziehungen“ nur am Endergebnis ersichtlich ist, welches Verhaltensmuster ein Entitätstyp annimmt. Dadurch gehen die Rollen in ihrer Detailliertheit in den Beziehungen aufgrund der fehlenden Rollenberücksichtigung unter. Anders sieht es in Klassendiagrammen aus, bei dem Rollen in ihrer konzeptionellen Gesamtheit als Klasse mit ihren Attributen explizit Berücksichtigung finden [vgl. Schneider 2007a, S. 260]. Nach Fowler [1997, S. 17ff] sind die Rollenkonzepte je nach Komplexität der Daten und dem daraus resultierenden Detailierungsgrad anzuwenden. In Abbildung 2-26 werden Anforderungen aufgezählt, die für das jeweilige Rollenkonzept in einem spezifischen Anwendungsfall sprechen.

Single Role Type	Seperate Role Type	Role Subtype
<ul style="list-style-type: none"> • Rollen die nicht weiter explizit ausgeführt werden müssen. • Entitäten die nicht explizit unterschieden werden müssen. 	<ul style="list-style-type: none"> • Viele Entitäten im Datenmodell, deren Rollen getrennt voneinander betrachtet werden können. 	<ul style="list-style-type: none"> • Wenig Rollen im Datenmodell. • Wenig neue Rollen im Datenmodell. • Feste Vorgabe der zu realisierenden Rollen.
Role Object	Role Relationship	
<ul style="list-style-type: none"> • Viele Rollen im Datenmodell. • Viele neue Rollen im Datenmodell. 	<ul style="list-style-type: none"> • Wenn ein elementarer Entitätstyp dieselbe Rolle bei verschiedenen Entitätstypen ausführt. 	

Abbildung 2-26: Anwendungsfälle der Rollenkonzepte [vgl. Fowler 1997, S. 4-17]

Abschließend ist auszuführen, dass das Rollenkonzept für eine generische Darstellung im konzeptionellen Datenmodell als wesentlich zu erachten ist. Dies liegt darin begründet, dass ein Rollenkonzept die generische Darstellung dahingehend fördert, dass erst bei einer entstehenden Beziehung ein Sachverhalt spezifiziert wird. Somit bleibt das konzeptionelle Datenmodell in seiner ursprünglichen Form allgemein, also generisch.

2.5.2 Powertyp-Konzept

In der Regel werden die Fachbegriffe eines Anwendungsbereichs in einem konzeptionellen Datenmodell durch Entitätstypen (Objekttypen, Klassen), Beziehungstypen und Attributen repräsentiert (siehe auch Kapitel 2.4.2). Eine Methode der generischen Darstellung ist, bestimmte Fachbegriffe auf der Ebene der Attributwerte anzusiedeln, um dadurch eine flexiblere Anpassung an sich verändernden Gegebenheiten zu ermöglichen [vgl. Schneider 2007b, S. 178]. Die Fachbegriffe, die auf Attributwertebene anzusiedeln sind, entsprechen den Fachbegriffen des *veränderlichen Verhaltensmusters* (siehe auch Kapitel 2.5.1). Dadurch bleibt die vorhandene Strukturumgebung trotz Integration neuer Fachbegriffe stabil. Jedoch setzt dies voraus, dass Änderungen in der Grundstruktur selbst nicht zu erwarten sind oder Anpassungen problemlos durchgeführt werden können [vgl. Schneider 2007a, S. 419]. Unter dem Begriff Strukturumgebung sind die beschriebenen Prozesse im konzeptionellen Datenmodell zu verstehen.

Um Fachbegriffe auf Attributwertebene anzusiedeln, ist eine inhaltliche Zweiteilung des konzeptionellen Datenmodells anzustreben [vgl. Schneider 2007a, S. 419]. Diese Zweiteilung entspricht einer Aufteilung des Datenmodells in eine Typ-Ausprägungs-Ebene sowie in eine Typ-Typ-Ebene [vgl. Schneider 2007a, S. 419]. Die Typ-Typ-Ebene beinhaltet sogenannte *Powertypen*, die der konkreten Bedeutung anderer Objekttypen, die auf Typ-Ausprägungs-Ebene angesiedelt sind, dienen. [vgl. Schneider 2007a, S. 420].

Durch die Verwendung von Powertypen können generische Darstellungen realisiert werden und ermöglichen die einfache Integration von Fachbegriffen ins Datenmodell. Darüber hinaus ist ein wesentlicher Vorteil eines Powertyps, dass ein Blick in die Powertypen ausreicht, um einen Überblick über die im Datenmodell umgesetzten Fachbegriffe zu erhalten. Somit wird ein Fachbegriff als „Attribut“ (Ausprägung) in einem Powertyp aufgenommen. Aus diesem Grund ist es möglich, dass weitere Fachbegriffe hinzugefügt werden können. Reichen die vorhandenen konkreten Objekttypen zur Detaillierung der Fachbegriffe nicht aus, so lassen sich einfach weitere Objekttypen vom jeweils vorhandenen konkreten Objekttypen ableiten. Dies setzt jedoch eine konstante Strukturgebung der Powertypen voraus. Im Rahmen dieser Arbeit wird zur Beschreibung eines Powertyps Schneider [2007a, S. 421ff] zitiert:

„Ein Powertyp ist ein Objekttyp, der Fachbegriffe auf Ausprägungsebene aufnimmt und damit die die Fachbegriffe konkretisierenden Objekttypen repräsentiert“.

Der Vorteil der inhaltlichen Zweiteilung eines generischen Datenmodells durch die Verwendung von Powertypen liegt darin, dass ein Blick in die Powertypen ausreicht, um eine Übersicht über die im konzeptionellen Datenmodell verwendeten Fachbegriffe zu erhalten. Somit wird deutlich, dass die konkreten Bezeichnungen der Objekttypen im Powertyp zusammengefasst sind. Weitere Fachbegriffe können dem Powertypen hinzugefügt werden. Darüber hinaus ermöglicht diese Darstellung eine Überprüfung, ob ein Fachbegriff in die bestehende Strukturumgebung integrierbar ist. [vgl. Schneider 2007a, S. 422]. In Abbildung 2-27 soll die Beziehung eines Powertypen zu einem konkreten Objekttypen verdeutlicht werden.

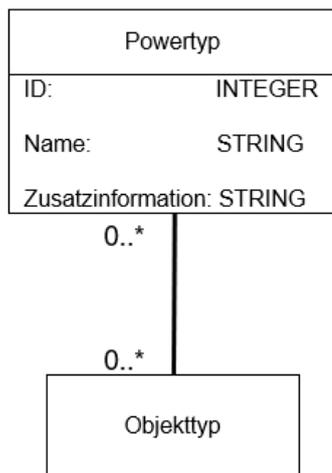


Abbildung 2-27: Beziehung eines Powertypen zu einem konkreten Objekttyp [vgl. Schneider 2007a, S. 428]

Ein Powertyp ist die Beschreibung für einen konkreten Objekttyp. Das Attribut *Zusatzinformation* erlaubt die Angabe erläuternder Informationen zum Fachbegriff und ist optional. Das Attribut *Name* nimmt die Bezeichnungen der zugeordneten Fachbegriffe auf [vgl. Schneider 2007a, S. 425; S. 428].

2.5.3 Strukturschema des Synthesekonzepts

In diesem Kapitel soll das Synthesekonzept zur generischen Darstellung eines Datenmodells anhand eines Beispiels näher erläutert werden. Das in diesem Kapitel beschriebene generische Datenmodellkonzept bezieht sich auf die Arbeit von Schneider [2007a, 451-456].

Aus Gründen der Übersichtlichkeit wurde im generischen Datenmodell auf die Zuordnung von Attributen verzichtet. Das generische Datenmodell resultiert aus der Kombination der beiden zuvor genannten Konzepte (Rollenkonzept und Powertyp-Konzept). Das in Abbildung 2-30 dargestellte generische Datenmodell repräsentiert eine auf UML basierte „Aufbauorganisation“ eines Konzerns. Im Folgenden wird das methodische Vorgehen beschrieben, das die für die Darstellung des generischen Datenmodells durchzuführen ist. Vorab ist jedoch anzumerken, dass die Darstellung des generischen Datenmodells von der subjektiven Einschätzung des Modellierers abhängt. Die modellierten Begriffe sowie ihr hierarchischer Aufbau können sich je nach Verständnis über einen Anwendungsbereich unterscheiden (siehe auch Kapitel 2.1.1).

Zunächst werden die begrifflichen Zusammenhänge einer Aufbauorganisation im Unternehmen festgelegt. In der Literatur existiert eine Vielzahl an Methoden, die eine schematische Vorgehensweise zur Entwicklung von Begriffen beschreiben (z.B. Fachbegriffsmodell). Allerdings haben diese Methoden alle eins gemeinsam. Sie sind von dem Verständnis eines Modellierers über einen Anwendungsbereich abhängig.

Die Aufbauorganisation eines Unternehmens besteht aus den Ebenen *Unternehmensbereich*, *Geschäftsbereich*, *Niederlassung* und *Kostenstelle*. Diese Begriffe entsprechen der unveränderlichen Wesensart eines Gegenstands [vgl. Schneider 2007a, S. 454]. Damit im späteren Datenmodell der hierarchische Aufbau dieser Ebenen beschrieben werden kann, werden die Rollen „*Übergeordnet*“ und „*Untergeordnet*“ als Begriffe gewählt, die das veränderliche Verhaltensmuster eines Gegenstands beschreiben. In Tabelle 2-8 werden die zuvor beschriebenen Begriffe zusammengefasst und nach ihrer Wesensart und Verhaltensmuster kategorisiert.

Tabelle 2-8: Kategorisierung der Fachbegriffe nach ihrer Wesensart und Verhaltensmuster [vgl. Schneider 2007a, S. 453]

Kategorie	Fachbegriff
Wesensart	Unternehmensbereich
Wesensart	Geschäftsbereich
Wesensart	Kostenstelle
Wesensart	Niederlassung
Verhaltensmuster	Übergeordnet
Verhaltensmuster	Untergeordnet

Die Ebenen *Unternehmensbereich*, *Geschäftsbereich*, *Niederlassung* und *Kostenstelle* lassen sich in dem übergeordneten Begriff *Organisation* zusammenfassen. Da neben der Organisation auch Personen in einem Unternehmen eine Rolle spielen, können die Begriffe *Organisation* und *Person* im übergeordneten Begriff *Partei* zusammengefasst werden [vgl. Schneider 2007a, S. 454]. Die in Abbildung 2-28 dargestellten Fachbegriffe stellen Fachbegriffe dar, die die *unveränderliche Wesensart* eines Gegenstands benennen.

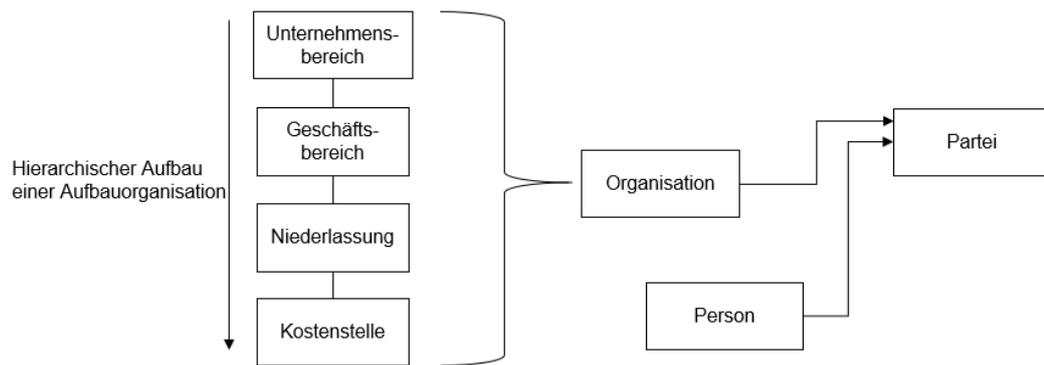


Abbildung 2-28: Beispielhafte Aufbauorganisation eines Unternehmens [vgl. Schneider 2007a, S. 451]

Entsprechend der Rollen, die diese Fachbegriffe in einer Aufbauorganisation einnehmen können, wurden auch die Fachbegriffe, die das *veränderliche Verhaltensmuster* eines Gegenstands benennen, in einer hierarchischen Struktur festgelegt (vgl. Abbildung 2-29)

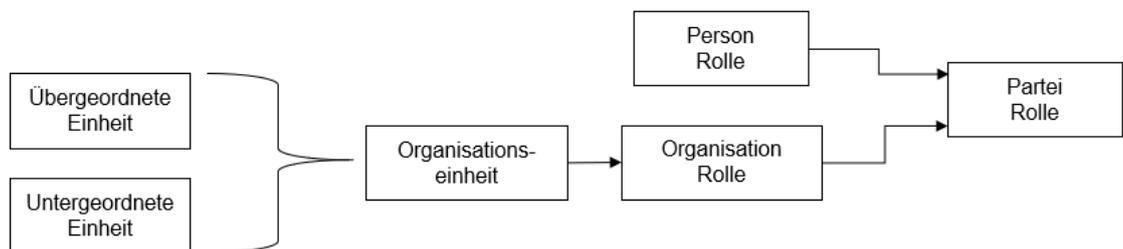


Abbildung 2-29: Hierarchischer Aufbau der Fachbegriffe des veränderlichen Verhaltensmuster [vgl. Schneider 2007a, S. 454]

Im nächsten Schritt wird für dieses Beispiel eine Beziehung festgelegt, die den Zusammenhang zwischen den Begriffen *Partei* und *Partei Rolle* beschreibt. Hierfür wird der Begriff *Partei Assoziation* gewählt. Des Weiteren müssen die Powertypen festgelegt werden, die eine grundlegende Struktur in diesem Datenmodell festlegen. Diese werden aus den Fachbegriffen entwickelt, die der höchsten hierarchischen Stufe der Fachbegriffe entspringen. Der Begriff *Partei* führt zum Powertyp *Partei Typ*, der Begriff *Partei Rolle* führt zum Powertyp *Partei Rollen Typ* und der Begriff *Partei Assoziation* führt zum Powertyp *Partei Assoziation Typ* [vgl. Schneider 2007a, S. 455]. In Abbildung 2-30 wird das generische Datenmodell der „Aufbauorganisation“ dargestellt.

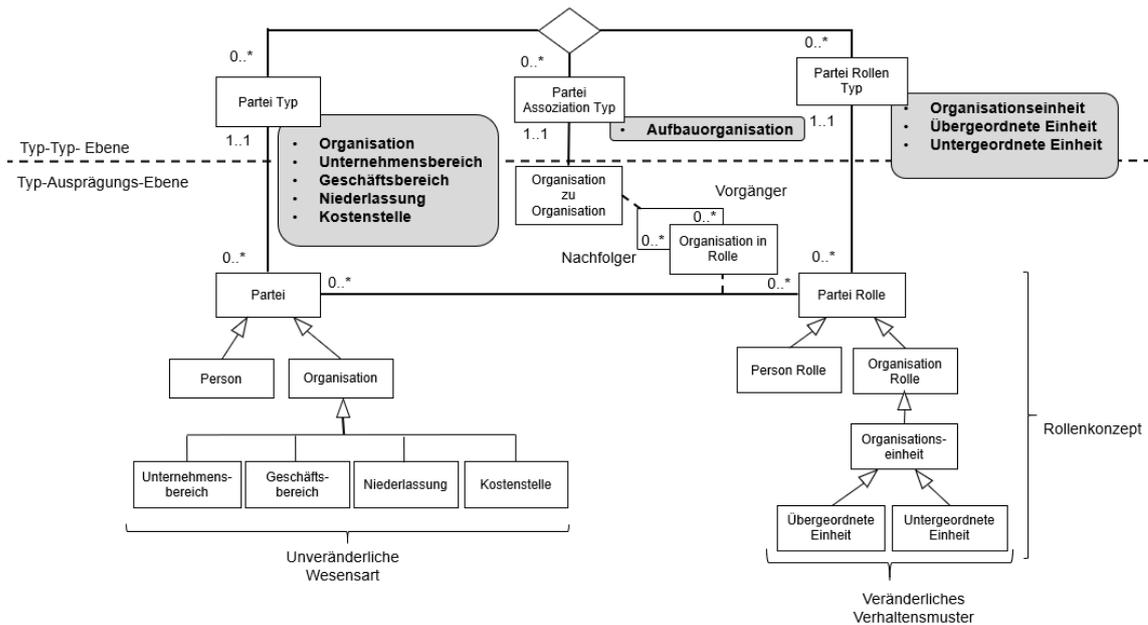


Abbildung 2-30: Generisches Datenmodell einer Aufbauorganisation im Konzern [vgl. Schneider 2007a, S. 456]

Die folgenden Erläuterungen zu den in Abbildung 2-30 dargestellten Sachverhalte beziehen sich auf die Ausarbeitung von Schneider [2007a, 456ff].

Auf der Typ-Typ-Ebene besteht das generische Datenmodell aus den Powertypen *Partei Typ*, *Partei Assoziation Typ* und *Partei Rollen Typ*. Die vordefinierten Ausprägungen der Powertypen lassen sich aus den grau unterlegten Aufzählungen ablesen. Dabei verfügt der *Partei Typ* über den Eintrag „Organisation“, der *Partei Assoziation Typ* über „Aufbauorganisation“ und der *Partei Rollen Typ* über „Organisationseinheit“, „Übergeordnete Einheit“ und „Untergeordnete Einheit“. Durch die mehrstellige Assoziation zwischen den Powertypen ist die Möglichkeit gegeben, auf Typ-Typ-Ebene Regeln festzulegen [vgl. Schneider 2007a, S. 456].

Beispielsweise kann durch die Verbindung des *Partei Assoziation Typs* „Aufbauorganisation“ mit dem *Partei Typ* „Unternehmensbereich“ und dem *Partei Rollen Typ* „Übergeordnete Einheit“ festgelegt werden, dass in der konkreten Zusammenführung auf der Typ-Ausprägungs-Ebene im Rahmen einer „Aufbauorganisation“ ein „Unternehmensbereich“ nur als übergeordnete Einheit erscheinen kann [vgl. Schneider 2007a, S. 456].

Auf der Typ-Ausprägungs-Ebene sind die Klassen *Partei* sowie ihrer Spezialisierung „Organisation“ und die Klasse *Partei Rolle* sowie ihrer Spezialisierungen „Organisation Rolle“, „Organisationseinheit“, „Übergeordnete Einheit“ und „Untergeordnete Einheit“ angegeben.

Die Spezialisierungen der „Organisation Rolle“ sind nur dann relevant, sofern die Objekttypkonkretisierungen der entsprechenden Fachbegriffe weitere Attribute für über- bzw. untergeordnete Einheiten benötigt. Vorgenanntes Beispiel setzt voraus, dass in den Powertypen definierte Fachbegriffe Konkretisierungen in den Objekttypen auf Typ-Ausprägungs-Ebene erfahren [vgl. Schneider 2007a, S. 456].

Weiterhin befinden sich auf der Typ-Ausprägungs-Ebene die Assoziationsklassen *Organisation in Rolle* sowie *Organisation zu Organisation*. Die in diesem Modell auf der Typ-Typ-Ebene universell aufgezeichnete Kardinalität (0..*) am Powertyp ist im generischen Datenmodell der „Aufbauorganisation“ zu einer (1..1) existenzabhängigen Assoziation geworden [vgl. Schneider 2007a, S. 457].

Ein Eintrag aus *Partei* ist genau durch einen Fachbegriff aus *Partei Typ*, ein Eintrag aus *Organisation zu Organisation* durch genau einen Fachbegriff aus *Partei Assoziation Typ* und ein Eintrag aus *Organisation Rolle* durch genau einen Eintrag aus *Partei Rollen Typ* typisiert. Die Modellierung des Rollenkonzepts wurde daher aus Gründen der Übersichtlichkeit in Abbildung 2-28 entsprechend dargestellt [vgl. Schneider 2007a, S. 457].

Durch diese Darstellung ist die Grundlage für ein Datenmodell geschaffen, dass eine flexible Anpassung an die gewünschte Aufbauorganisation erlaubt. Das in Abbildung 2-30 dargestellte generische Datenmodell einer Aufbauorganisation ist nur möglich, da festgestellt wurde, dass es sich bei dem Begriff *Organisation* um die *unveränderliche Wesensart* einer Organisation und bei dem Begriff *Organisationseinheit*, um das *veränderliche Verhaltensmuster* einer Organisation innerhalb der Aufbauorganisation handelt. Die Verhaltensmuster „Übergeordnete“ und „Untergeordnete“ Einheit stehen für die konkrete Stellung (Vorgänger / Nachfolger) einer Organisation in der Aufbauorganisation. Eine Organisation kann einerseits als übergeordnete Organisationseinheit, andererseits als untergeordnete Organisationseinheit dargestellt werden. Unabhängig von ihrer Rolle bleibt sie aber unverändert eine Organisation [vgl. Schneider 2007a, S. 457].

Die Assoziationsklasse *Organisation zu Organisation* beschreibt grundsätzlich Assoziationen zwischen Organisationen in ihren Rollenausübungen. Eine weitere Assoziation unterhält *Organisation zu Organisation* zum Powertyp *Partei Assoziation Typ*. Der Beziehungstyp „Aufbauorganisation“ im Powertyp *Partei Assoziation Typ* kennzeichnet Organisationen in ihren diversen Rollen (übergeordnete / untergeordnete Einheit) [vgl. Schneider 2007a, S. 458].

Die in dieser Abbildung (vgl. Abbildung 2-30) beschriebene Aufbauorganisation wird durch die Zusammenführung einzelner Organisationen gebildet, die im *Partei Typ* und im *Partei*

Rollen Typ in ihren jeweiligen Rollenausübungen spezifiziert sind [vgl. Schneider 2007a, S. 458].

Die Assoziationsklasse *Organisation zu Organisation* steht in einer rekursiven Beziehung, um auszudrücken, dass eine bestimmte Organisation in ihrer Rollenausübung mit einer anderen Organisation in ihrer Rollenausübung in einer Assoziation steht. Die daraus resultierende Assoziation ist als Assoziationsklasse *Organisation zu Organisation* realisiert [vgl. Schneider 2007a, S. 458].

Die nachfolgende Abbildung (vgl. Abbildung 2-31) verdeutlicht die möglichen Rollen-Paare „Organisation“ aus dem in Abbildung 2-30 dargestellten generischen Datenmodell einer „Aufbauorganisation“. Für die Gegenüberstellung der Organisations-Rollen-Paaren, werden konkrete Organisationen z.B. Unternehmensbereich 1 (UB1), Geschäftsbereich 1 (GB1), Geschäftsbereich (GB 2) und Niederlassung 1 (NL 1) der Klasse *Partei* unter der Spezialisierung *Organisation* zugeordnet [vgl. Schneider 2007a, S. 458].

Partei Assoziation Typ	Organisation 1	Organisation Rolle 1	Organisation 2	Organisation Rolle 2
Aufbauorganisation	UB 1	Übergeord. Einheit	GB 1	Untergeord. Einheit
Aufbauorganisation	UB 1	Übergeord. Einheit	GB 2	Untergeord. Einheit
Aufbauorganisation	GB 1	Übergeord. Einheit	NL 1	Untergeord. Einheit
Aufbauorganisation	GB 2	Übergeord. Einheit	NL 1	Untergeord. Einheit

Abbildung 2-31: Kombinationsmöglichkeiten von Organisations-Rollen-Paaren im generischen Datenmodell [vgl. Schneider 2007a, S. 458]

Darüber hinaus ist das integrieren weiterer generischer Datenmodelle in der Regel unproblematisch (vgl. Abbildung 2-32). Dies soll in der folgenden Abbildung verdeutlicht werden. Dabei wird das „Generische Datenmodell 1“, um das „Generische Datenmodell 2“ erweitert, wodurch die Kommunikationsmechanismen *Telekommunikation* und *E-Mail* in den bestehenden Prozess in „Generisches Datenmodell 1“ integriert wurden [vgl. Schneider 2007a, S. 462ff].

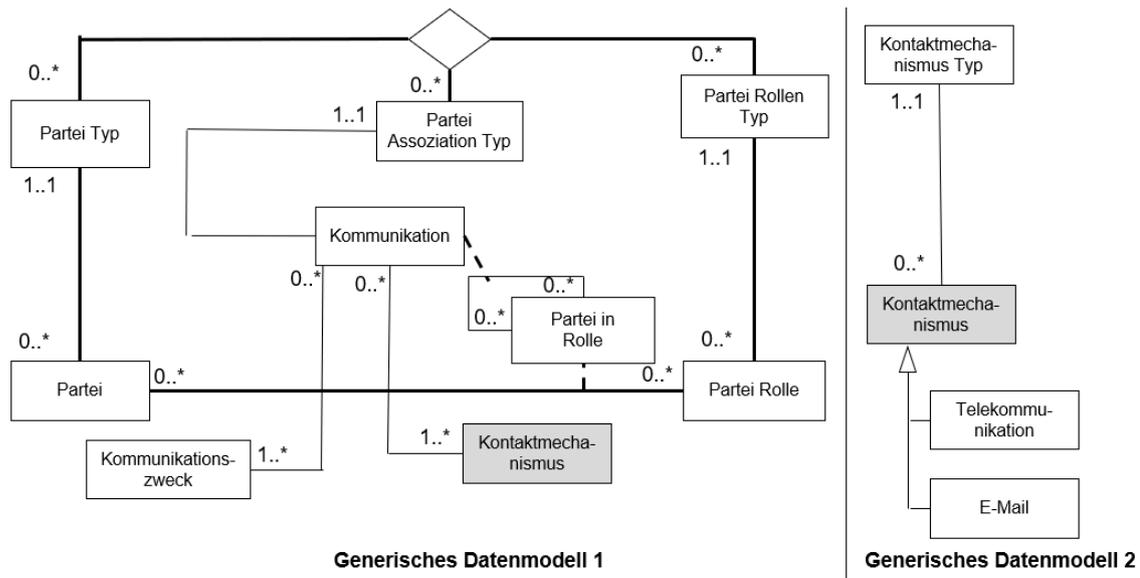


Abbildung 2-32: Integrationsfähigkeit des generischen Datenmodells [vgl. Schneider 2007a, S. 462f]

Abschließend ist auszuführen, dass sich in diesem Konzept keine universell gültigen Strukturierungsvorschläge für konkrete Objekttypen in der Typ-Ausprägungs-Ebene festlegen lassen, da diese von der kontextabhängigen Interpretation eines Fachbegriffs in einem Anwendungsbereich abhängig sind. Somit kann im Vorfeld keine Aussage zur Strukturierung konkreter Objekttypen auf der Typ-Ausprägungs-Ebene festgelegt werden [vgl. Schneider 2007a, S. 427]. Dies trifft auf Powertypen nicht zu, weshalb ein Powertyp aus den Attributen ID, Name und einer optionalen Zusatzinformation besteht, um auf konkrete Objekttypen zu verweisen [vgl. Schneider 2007a, S. 428; S.450]. Im Rahmen dieser Arbeit soll dieses Synthesekonzept auf das SC-Umfeld angewendet werden (siehe auch Kapitel 3.3), um zu untersuchen, welche Herausforderungen sich bei der Modellierung von generischen Datenmodellen mit SC-Daten ergeben.

2.6 Disziplinen der Modellverständlichkeit

Da konzeptionelle Datenmodelle die Beziehung zwischen Fachbegriffen beschreiben (siehe auch Kapitel 2.4.2), müssen Anforderungen untersucht werden, die die Modellverständlichkeit sicherstellen, um Fehlinterpretationen zu vermeiden. Aus diesem Grund werden in diesem Kapitel die Disziplinen der Ontologie sowie des Terminologiemanagements untersucht.

Ontologie

Die Ontologie ist eine Disziplin, um die Realität korrekt und allgemeingültig beschreiben zu können. Die Ontologie in der Informatik befasst sich mit der Darstellung von Begriffen und der zwischen ihnen bestehenden Beziehungen in einem bestimmten Anwendungsbereich [vgl. Dengel 2012, S. 64ff]. Somit ist offensichtlich, dass der Fokus dieser Disziplin darin besteht, ein einheitliches Verständnis sicherzustellen. Die Verständlichkeit eines Modells kann durch vier Kommunikationsszenarien beschrieben werden, die aufgrund unterschiedlich zugrunde liegender Ontologien hervorgehen. Zu nennen sind *Konsens*, *Korrespondenz*, *Konflikte* und *Kontraste* [vgl. Dengel 2012, S. 137f].

- Der *Konsens* beschreibt ein Szenario, bei dem dieselbe Konzeptualisierung sowie gleiche Begriffe verwendet werden [vgl. Dengel 2012, S. 137].
- Die *Korrespondenz* beschreibt ein Szenario, bei dem dieselbe Konzeptualisierung, jedoch unterschiedliche Begriffe verwendet werden [vgl. Dengel 2012, S. 138].
- *Konflikte* treten auf, bei dem einheitliche Begriffe, jedoch unterschiedliche Konzeptualisierungen verwendet werden [vgl. Dengel 2012, S. 138].
- *Kontraste* liegen vor, wenn unterschiedliche Begriffe sowie Konzeptualisierungen verwendet werden [vgl. Dengel 2012, S. 138].

Um einen effektiven Austausch und einen gemeinsamen *Konsens* zu ermöglichen, konzentriert sich die Ontologie auf das Schaffen einer für die Kommunikationsteilnehmer gemeinsamen Konzeptualisierung. Hierfür muss das zur erfolgreichen Kommunikation nötige Wissen über Begriffe und Zusammenhänge beschrieben werden. Darauf aufbauend wird Reichenberger [2010, S. 6] zitiert. Er beschreibt wesentliche Anforderungen, um die einheitliche Verständlichkeit in einem Modell sicherzustellen. Dabei unterscheidet er zwischen dem *Prinzip der Objektidentität*, der *Trennung von Objekt und Benennung* sowie der *Redundanzfreiheit durch Ableitung von Informationen*.

- Das *Prinzip der Objektidentität* sagt aus, dass jedes Objekt nur einmal im semantischen Netzwerk vertreten sein darf. Zusätzlich kommen alle Informationen zu diesem Objekt an einer Stelle zusammen. Somit wird das Zerstreuen sowie redundant angelegte Informationen vermieden [vgl. Reichenberger 2010, S. 7ff].
- Die *Trennung von Objekt und Benennung* baut auf der Objektidentität auf. Hierbei soll durch sprachliche Varianten auf ein Objekt zurückgeführt werden, um redundant ange-

legte Objekte zu vermeiden. Das Objekt wird mit so vielen Synonymen ausgestattet wie nötig, um Informationen auf dieses Objekt zurückzuführen [vgl. Reichenberger 2010, S. 9f]. Als Synonym sind mehrdeutige Begriffe zu verstehen, die den gleichen Gegenstand oder Sachverhalt beschreiben. Als Beispiel wird der Begriff *Verkauf* genannt. Dieser Begriff könnte auch durch die Begriffe *Absatz* oder *Umsatz* beschrieben werden. Eine andere Form der mehrdeutigen Begriffe bilden Homonyme. Diese müssen vermieden werden, um Fehlinterpretationen zu vermeiden. Ein klassisches Beispiel für ein Homonym ist der Begriff *Rad*. Darunter könnte ein Fahrrad oder das Rad eines Fahrzeugs verstanden werden [vgl. Reichenberger 2010, S. 6].

- Das Prinzip der *Redundanzfreiheit in Ableitungen* sagt aus, dass Informationen aus anderen Informationen abzuleiten sind, bevor diese modelliert werden. Dadurch ergibt sich eine hierarchische Struktur, bei dem Sachverhalte auf bestimmte Informationen zurückgeführt werden können [vgl. Reichenberger 2010, S. 10f].

Darüber hinaus führt Reichenberger [2010, S. 12f] aus, dass eine Thematisierung der zu modellierenden Bereiche essenziell ist, um eine Übersicht über die Verfügbarkeit von Informationen zu erhalten. Zusätzlich wird es ermöglicht, die Informationen ihren Bereichen, aus denen sie entstammen, zuordnen zu können.

Terminologiemangement

Das Terminologiemangement beschäftigt sich mit der Planung, Steuerung und Organisation verwendeter Wörter und Fachausdrücke [vgl. Schneider 2007a, S. 283]. Das methodische Vorgehen des Terminologiemagements wird durch die *Begriffsanalyse* und *Rekonstruktion* sowie die *Begriffsnormierung* und *Begriffsdefinition* beschrieben. Jedoch ist hierbei offen, auf welche Art und Weise zu Fachbegriffen gelangt wird. In Anlehnung an die Ontologie verfolgt das Terminologiemangement das Ziel, eine klar definierte, konsistente, konsolidierte sowie konsensfähige Begriffswelt zu schaffen [vgl. Hellmuth 1997, S. 52f]. Dazu ist eine genaue Untersuchung verschiedener Fachbereiche notwendig, um die dort beinhalteten Daten und Informationen in wesentliche Fachbegriffe umzuwandeln, die zur Beschreibung des Datenmodells dienen [vgl. Schneider 2007a, S 441].

In der Phase der *Begriffsanalyse* und *Rekonstruktion* werden die Fachbegriffe für einen bestimmten Prozess analysiert und eine vorläufige Definition erarbeitet. Abschließend wird eine Rekonstruktion der Fachbegriffe durchgeführt, um eine eindeutige Begriffswelt zu konstruieren [vgl. Hellmuth 1997, S. 52f].

Die *Begriffsnormierung* vereinheitlicht die Begriffswelt, so dass zwischen den Begriffen in einem Modell ein Konsens besteht [vgl. Hellmuth 1997, S. 58ff].

Darauf aufbauend führt Schneider [2007a, S. 442] aus, dass bei der Entwicklung von Fachbegriffen eine Kategorisierung notwendig ist, um dem generischen Konstruktionsrahmen Folge leisten zu können. Nach seiner Methode ist ein Fachbegriff nach zwei wesentlich Aspekten zu differenzieren. Hierbei wird zwischen einer *unveränderbaren Wesensart* sowie einem *veränderlichen Verhaltensmuster* unterschieden. Die Fachbegriffe der unveränderbaren Wesensart können unabhängig von anderen Fachbegriffen im Modell existieren. Fachbegriffe, die das veränderliche Verhaltensmuster beschreiben, können dies nicht. Diese sind an den Fachbegriffen der unveränderbaren Wesensart existenzgebunden.

Zusammenfassend ist auszuführen, dass die Qualität eines konzeptionellen Datenmodells sowohl durch die klar definierten Fachbegriffe als auch durch ihre Darstellung im Modell bestimmt wird. Da die Art und Weise offengelegt ist wie generische Fachbegriffe entwickelt werden können, wird im Rahmen dieser Arbeit eine Methode herausgearbeitet, um generische Fachbegriffe zu entwickeln (siehe auch Kapitel 3.2.4). Als Orientierung dienen hierbei das methodische Vorgehen des Terminologiemanagements, die Anforderungen des Rollenkonzepts sowie die Anforderungen der Ontologie.

3 Anforderungserhebung und -analyse im Supply-Chain-Umfeld

Dieses Kapitel verfolgt das Ziel, die zuvor beschriebenen Methoden und Konzepte eines generischen Datenmodells im SC-Kontext zu untersuchen und Anforderungen daraus abzuleiten. Im Folgenden werden die Ziele der einzelnen Kapitel erläutert.

Zunächst wird in Kapitel 3.1 untersucht, welche Unterschiede sich bei der Verwendung von generischen oder spezifischen Fachbegriffen in einem konzeptionellen Datenmodell ergeben. Dabei soll festgestellt werden, welche Vorteile generisch beschriebene Datenmodelle gegenüber spezifisch beschriebene Datenmodell aufweisen und welche Anforderungen sich für generische Datenmodelle aus dem Vergleich ergeben.

In Kapitel 3.2 werden die in Kapitel 2.5 beschriebenen Anforderungen an das generische Datenmodell auf das SC-Umfeld reflektiert. Daraus sollen Anforderungen für das SC-Umfeld abgeleitet werden, die für die Konstruktion eines generischen Datenmodells für den Einsatz mit SC-Daten zu berücksichtigen sind. Zusätzlich werden die in Kapitel 2.5 beschriebenen Methoden und Konzepte im Rahmen des SC-Umfeld untersucht, woraus sich weitere Anforderungen ableiten. Dabei soll das in Kapitel 2.2.1 beschriebene SCOR-Modell dazu dienen, um die Betrachtung auf die allgemeinen Prozesse einer SC zu lenken.

In Kapitel 3.3 wird die Empfehlung eines geeigneten Rollenkonzepts sowie einer geeigneten Modellierungssprache für das im SC-Umfeld zu realisierende generische Datenmodell ausgesprochen. Darauf aufbauend werden einige beispielhafte SC-Prozesse in einem generischen Datenmodell dargestellt. Die Basis bildet zum einen das in Kapitel 2.5.3 beschriebene Synthesekonzept zur Konstruktion eines generischen Datenmodells sowie die in Kapitel 2.1.2 herausgearbeiteten SC-Daten, um einige SC-Fachbegriffe zu modellieren. Die in Kapitel 3.2.5 zusammengefassten Anforderungen zur Konstruktion eines generischen Datenmodells für den Einsatz mit SC-Daten sollen dabei als Orientierungshilfe dienen. Daraus resultierend sollen Herausforderungen beschrieben werden.

In Kapitel 3.4 wird das datenorientierte Kompendium für das SC-Umfeld vorgestellt, welches für die Konstruktion eines generischen Datenmodells für den Einsatz mit SC-Daten empfohlen wird. Im datenorientierten Kompendium werden normierte SC-Fachbegriffe aufgeführt, wodurch sichergestellt werden soll, dass jeder Modellierer über das gleiche Verständnis über die zu verwendeten SC-Fachbegriffe verfügt.

3.1 Unterschiede zwischen einem generischen und spezifischen Datenmodell

In Kapitel 2.4.2 wurde erläutert, dass ein generisches Datenmodell eine Darstellungsart des konzeptionellen Datenmodells entspricht und als Analysemodell zur Beschreibung von Fachbegriffen und ihren Beziehungen dient. Im Rahmen der konzeptionellen Datenmodellierung können dabei die verwendeten Begriffe generischer oder spezifischer gewählt werden. Diese Unterscheidung soll auf Abbildung 3-1 durch die Begriffe *Mitarbeiter* und *Produktionsart* verdeutlicht werden. Vorab ist anzumerken, dass die Darstellung auf Abbildung 3-1 keiner speziellen Modellierungssprache angehört (z.B. ERM, UML). Es sollen lediglich die gewählten Begriffe im Vordergrund dieser Betrachtung stehen.

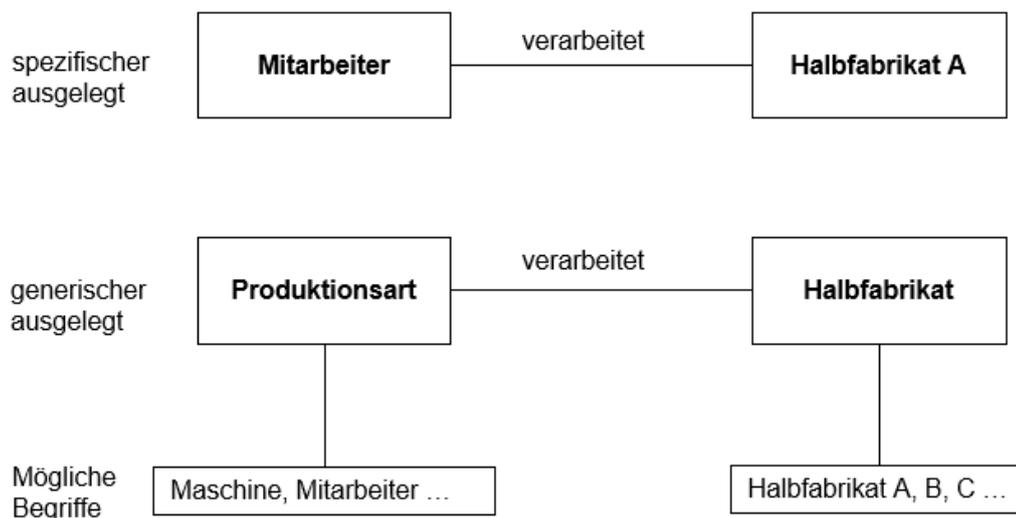


Abbildung 3-1: Unterschied zwischen einer generischen und spezifischen Modellbeschreibung

Auf Abbildung 3-1 ist ersichtlich, dass der gewählte Begriff *Mitarbeiter* einen Prozess spezifischer beschreibt, als der Begriff *Produktionsart*. Dies liegt darin begründet, dass der Begriff *Mitarbeiter* einen Vorgang im Prozess beschreibt, der ausschließlich durch Menschen durchgeführt wird. Hingegen erlaubt der Begriff *Produktionsart*, die Interpretation weiterer Begriffe, wodurch beispielsweise der Vorgang in einem Prozess auch durch Maschinen durchgeführt werden könnte. Somit wird deutlich, dass eine generische Beschreibung eine Anpassung in einem Modell ermöglicht. Im Grunde wird ein allgemeiner (generischer) Begriff entwickelt, der mehrere Begriffe umfasst und anwendungsspezifisch verfeinert werden kann. Durch jede weitere Spezifizierung eines Begriffs, wird dadurch die Interpretationsmöglichkeit eingeschränkt. Dies kann so lange fortgeführt werden, bis ein Begriff konkret wird. Somit würde beispielsweise statt der Begriff *Mitarbeiter*, der Name

einer Person im Prozess beschrieben werden. Dies führt zur Annahme, dass generische Begriffe einen hierarchischen Aufbau folgen. Des Weiteren ist anzumerken, dass der Begriff *Mitarbeiter* nicht zwangsläufig als „nicht generisch“ abzustempeln ist, da z.B. nicht festgelegt wurde, ob ein interner oder externer Mitarbeiter (z.B. Leiharbeiter) den Prozess durchführt.

Darüber hinaus ist auszuführen, dass die Darstellungsart eines konzeptionellen Datenmodells nach der Auslegung der Begriffe (generisch / spezifisch) gestaltet wird. Dies liegt darin begründet, dass bei spezifischen Datenmodellen die Prozesse konkretisiert werden, wodurch kein weiterer Interpretationsspielraum mehr gegeben ist. Ein Prozess wurde somit auf eine bestimmte Sache festgelegt. Demgegenüber ist das generische Datenmodell allgemein gehalten und dient als Basis für die Beschreibung verschiedener Prozesse.

In den nachfolgenden Kapiteln sollen die Vor- und Nachteile untersucht werden, die generische und spezifische Fachbegriffe in einem konzeptionellen Datenmodell mit sich bringen.

3.1.1 Nachteile generischer und spezifischer Datenmodelle

Aufgrund einer generischen Beschreibung besteht die Gefahr von Missverständnissen. Im Rahmen der Konstruktion eines konzeptionellen Datenmodells, stellt die hohe Verständlichkeit definitiv ein wesentliches Qualitätskriterium dar (siehe auch Kapitel 2.5). Die Begründung hierfür ist recht trivial. Mit Bezug zu Kapitel 2.4.2 wurde vorangeführt, dass ein konzeptionelles Datenmodell der Beschreibung von Fachbegriffen und ihren Beziehungen dienen soll. Somit wird unterschwellig die Verständlichkeit eines konzeptionellen Datenmodells vorausgesetzt. Andernfalls würde das konzeptionelle Datenmodell seine Daseinsberechtigung als Analysemodell verlieren. In der folgenden Abbildung (vgl. Abbildung 3-2) sollen die Schwächen einer generischen Beschreibung verdeutlicht werden.

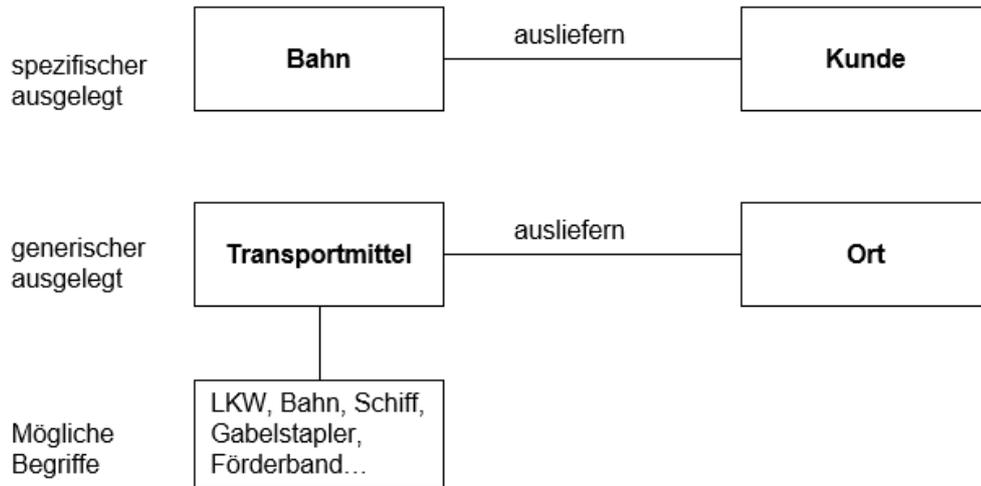


Abbildung 3-2: Interpretationsspielraum durch spezifische und generische Begriffe

In der spezifischeren ausgelegten Beziehung wird beschrieben, dass der Transport von beispielsweise Gütern, nur durch die Bahn an den Kunden erfolgt. Andere Transportmöglichkeiten oder eine Auslieferung innerhalb eines Konzerns, würden in dieser Beschreibung außen vor bleiben. Der Prozess ist somit recht deutlich formuliert. Die generischere ausgelegte Beziehung würde die Auswahl beliebiger Transportmittel ermöglichen. Allerdings ist an dieser Stelle unklar, welche Transportmittel explizit gemeint sein können. So wäre es denkbar, dass unter dem Begriff Transportmittel die interne Beförderung von Gütern durch beispielsweise Gabelstapler zu verstehen ist oder die externe Beförderung von Gütern durch beispielsweise LKWs. In übersichtlichen Darstellungen mag der Sachverhalt noch klar sein, wie Prozesse zu interpretieren sind. Allerdings wird dies in komplexen Prozessen (Strukturen) wie bei einer SC nicht mehr möglich sein.

Somit ist die Konstruktion eines konzeptionellen Datenmodells mit der Herausforderung verbunden, wie spezifisch die Fachbegriffe auszulegen sind. Je spezifischer das konzeptionelle Datenmodell ausgelegt ist, desto höher ist seine Verständlichkeit. Jedoch sinkt mit steigender Spezifikation der Fachbegriffe die Integrierbarkeit (Anpassbarkeit) neuer Prozesse sowie die Wiederverwendbarkeit eines konzeptionellen Datenmodells. Darüber hinaus ist noch auszuführen, dass mit der steigenden spezifischen Auslegung der Fachbegriffe wirtschaftliche Vorteile verloren gehen könnten. Der Verlust der wirtschaftlichen Vorteile ist darin zu begründen, dass im Falle sich verändernden Gegebenheiten neue Datenmodelle für bestimmte Sachverhalte entwickelt werden müssten. Die Nachteile generischer und spezifischer Datenmodelle werden in Tabelle 3-1 zusammengefasst.

Tabelle 3-1: Nachteile generischer und spezifischer Datenmodelle

Generisches Datenmodell	Spezifisches Datenmodell
<ul style="list-style-type: none"> • Fehlinterpretationen möglich • Verlust der Verständlichkeit 	<ul style="list-style-type: none"> • Verlust wirtschaftlicher Vorteile • Prozesse können nicht flexibel beschrieben werden

Abschließend ist auszuführen, dass durch generische Fachbegriffe die Prozesse nicht richtig definiert werden könnten und dies somit zu Fehlinterpretationen führt. Die Nachteile der spezifischen Fachbegriffe führen eher dazu, dass die Prozesse im spezifischen Datenmodell schwer oder gar nicht angepasst werden könnten.

3.1.2 Vorteile eines generischen und spezifischen Datenmodells

Datenmodelle müssen sich als flexibel genug erweisen, um sich den Veränderungen zeitnah und kostengünstig anpassen zu können. Dies erfordert wiederverwendbare Datenmodelle. Genau diese Anforderungen können generische Datenmodelle erfüllen, welches auf generisch ausgelegte Fachbegriffe zurückzuführen ist. Durch die Wiederverwendbarkeit können generische Datenmodelle einem Unternehmen monetäre Vorteile verschaffen (Kostensparnis). Dies kann dadurch begründet werden, dass durch die Wiederverwendbarkeit die Langlebigkeit eines konzeptionellen Datenmodells sichergestellt wird. Die Wiederverwendbarkeit eines konzeptionellen Datenmodell setzt die Integrierfähigkeit von beispielsweise neuen Prozessen im konzeptionellen Datenmodell voraus. Diese Aspekte wurden unter anderem bereits in Kapitel 2.5 als Anforderungen für ein generisches Datenmodell beschrieben.

Durch die Verwendung spezifischer Fachbegriffe wird eine hohe semantische Verständlichkeit im Datenmodell sichergestellt. Genau dies ist die Stärke eines spezifischen Datenmodells. Die hohe Verständlichkeit eines konzeptionellen Datenmodells wurde bereits zuvor als wesentliches Qualitätskriterium für ein konzeptionelles Datenmodell festgestellt (siehe auch Kapitel 3.1.1). Die Vorteile generischer und spezifischer Datenmodelle werden in Tabelle 3-2 zusammengefasst.

Tabelle 3-2: Vorteile generischer und spezifischer Datenmodelle

Generisches Datenmodell	Spezifisches Datenmodell
<ul style="list-style-type: none"> • Flexibilität • Langlebigkeit • Kostenersparnis 	<ul style="list-style-type: none"> • Hohe Verständlichkeit • Hohe Datenmodellqualität

Nach vorangegangener Unterscheidung eines generischen und spezifischen Datenmodells, kann im Rahmen dieser Arbeit abschließend festgestellt werden, dass ein spezifisches Datenmodell das Resultat eines generischen Datenmodells ist, dessen vordefinierte Fachbegriffe verfeinert werden.

Abschließend ist auszuführen, dass die in Kapitel 3.1.1 und Kapitel 3.1.2 zusammengefassten Vor- und Nachteile im folgenden Kapitel verwendet werden, um *allgemeine Anforderungen* für ein generisches Datenmodell im SC-Umfeld festzulegen. Dadurch kann rückwirkend geprüft werden, ob die in Kapitel 3.2 herausgearbeiteten methodisch-spezifischen Anforderungen, den allgemeinen Anforderungen Folge leisten können.

3.1.3 Allgemeine Anforderungen an ein generisches Datenmodell

Hervorgehend aus der Unterscheidung zwischen einem generischen und spezifischen Datenmodell bzw. zwischen generischen und spezifisch ausgelegten Fachbegriffen in einem konzeptionellen Datenmodell, ist somit für generische Datenmodelle eine Lösung hinsichtlich der *Modellverständlichkeit* zu fordern. Offensichtlich ist, dass bei den Begriffen zur Beschreibung von Prozessen, ein zu großer Interpretationsspielraum zu Fehlinterpretationen im generischen Datenmodell führen kann. Demzufolge muss eine einheitliche Konzeptualisierung sowie einheitliche Begriffe verwendet werden, um die Verständlichkeit eines generischen Datenmodells sicherzustellen. In Anlehnung der vorangeführten Unterscheidung zwischen generischen und spezifischen Datenmodellen sowie den beschriebenen Anordnungen an ein generisches Datenmodell aus Kapitel 2.5, werden im Folgenden *allgemeine Anforderungen an ein generisches Datenmodell im SC-Umfeld* aufgestellt. Die allgemeinen Anforderungen dienen als Orientierungshilfe zur Konstruktion von generischen Datenmodellen im SC-Umfeld.

- Beschreibung des Datenmodellzwecks

- SC-Fachbegriffsmodellierung als Vorstufe der Datenmodellierung
 - Beschreibung wesentlicher SC-Fachbegriffe
 - Festlegen des Interpretationsspielraums von SC-Fachbegriffen
 - Entwickeln hierarchischer (generischer) SC-Fachbegriffe

Die Beschreibung des Datenmodellzwecks dient der ergebnisorientierten Darstellung des generischen Datenmodells. Darauf aufbauend werden wesentliche SC-Fachbegriffe festgelegt, die die Prozesse in SCs realitätsnah im generischen Datenmodell wiedergeben müssen. Dies verlangt eine Methode, um sowohl generische SC-Fachbegriffe zu entwickeln als auch den Interpretationsspielraum von SC-Fachbegriffen zu minimieren, um Fehlinterpretationen zu vermeiden (siehe auch Kapitel 3.2.2). Hierfür wird das Führen eines *datenorientierten Kompendiums* für SCs empfohlen. Das Kompendium beinhaltet domänenspezifische Fachbegriffe, die zum modellieren generischer Datenmodelle herangezogen werden können (siehe auch Kapitel 3.4).

Im folgenden Kapitel werden die Methoden zur Modellverständlichkeit (siehe auch Kapitel 2.6) und Konzepte für die generische Darstellung (siehe auch Kapitel 2.5) auf das SC-Umfeld übertragen und untersucht. Das Ergebnis dieser Untersuchung führt zu einer Beschreibung der Anforderungen an ein generisches Datenmodell für den Einsatz mit SC-Daten.

3.2 Anforderungen an das generische Datenmodell

In diesem Kapitel sollen nun die methodenspezifischen Anforderungen ermittelt werden. Dafür werden die in Kapitel 2.5 und Kapitel 2.6 beschriebenen Methoden auf das SC-Umfeld reflektiert. Darauf aufbauend, werden die Anforderungen an generischen Datenmodellen für das SC-Umfeld abgeleitet.

3.2.1 Entwurfskonzept des generischen Datenmodells

Der Entwurf eines generischen Datenmodells für die Organisationsstruktur eines Unternehmens kann für die Entwicklung eines Informationssystem in der Produktionsplanung oder für ein unternehmensübergreifendes logistisches Informationssystem verwendet werden. Damit diese Informationssysteme miteinander interagieren können, muss in Hinblick der Entwicklung eines generischen Datenmodells eine einheitliche Ontologie vorherrschen. In Anlehnung an Kapitel 2.6 ist ein *Konsens* in der generischen Datenmodellkonstruktion zu fordern. Darunter ist zu verstehen, dass in der Entwicklung eines generischen Datenmodells dieselbe Konzeptualisierung sowie einheitliche Fachbegriffe verwendet werden müssen. Dadurch wird der Wissensaustausch zwischen verschiedenen Informationssystemen erleichtert. Als Beispiel ist der Wissensaustausch zwischen der Produktionsplanung und der Logistik zu nennen. Fragt das Informationssystem der Produktionsplanung nach allen möglichen Ressourcen mit spezifischen Attributen, so sollte für beide Informationssysteme das gleiche Verständnis zugrunde liegen, was eine Ressource ist. Somit ist für generische Datenmodelle im SC-Umfeld eine einheitliche Semantik zu fordern, die sowohl von den Vertragspartnern in einem SC-Netzwerk als auch von Maschinen richtig interpretiert werden kann. Diese Anforderungen sind auch aus Kapitel 2.5 zu entnehmen. Die Rede ist hierbei von der *Verständlichkeit* und der *Korrektheit*.

Um eine präzise Übersicht über die Informationen in den Prozessen zu erlangen, wurde in Kapitel 2.6 gefordert, die zu modellierenden Bereiche zu thematisieren. Somit wird eine Übersicht über die Verfügbarkeit von Informationen sichergestellt. Darüber hinaus wurde in Kapitel 2.5 als Anforderung an ein generisches Datenmodells die *Modularität* beschrieben. Diese besagt, dass ein Datenmodell in einzelne Bereiche aufgeteilt werden kann, ohne im Ganzen verändert zu werden. Aus diesem Grund wird im Folgenden nach einer Möglichkeit gesucht, eine SC in entsprechende Teildatenmodelle zu kategorisieren.

Aus der Thematisierung eines konzeptionellen Datenmodells in verschiedene Bereiche, resultieren Teildatenmodelle. Bei einem Teildatenmodell handelt es sich um ein konzeptionelles Datenmodell, dass nach einem spezifischen Themenbereich modelliert ist. Dabei kann ein Teildatenmodell im SC-Umfeld beispielsweise den Produktionsprozess eines Unternehmens oder den logistischen Prozess der vorgelagerten SC-Partner abbilden. In Anlehnung an die vorangeführte Beschreibung eines Teildatenmodells soll im Folgenden die in diesem Kapitel zentrale Begrifflichkeit des Teildatenmodells geklärt werden.

Ein *Teildatenmodell* entspricht einem konzeptionellen Datenmodell, dessen Diskurswelt auf einen spezifischen Bereich beschränkt wird. Teildatenmodelle sind keine eigenständigen Datenmodelle, sondern stellen einen Ausschnitt aus dem gesamten konzeptionellen

Datenmodell dar. Somit dient ein Teildatenmodell dazu, die Komplexität im Modellierungsprozess zu reduzieren. Alle Teildatenmodelle müssen dieselbe Konzeptualisierung aufweisen. Dadurch wird gewährleistet, dass sie untereinander konsistent sind. Das Teildatenmodell kann sowohl durch generische als auch spezifische Fachbegriffe repräsentiert werden.

Darauf aufbauend ist die Überlegung anzustellen, in welche Bereiche eine SC zu kategorisieren ist, um entsprechende Teildatenmodelle zu modellieren. Des Weiteren muss untersucht werden, wie Teildatenmodelle in SCs, die sowohl prozessübergreifende als auch unternehmensübergreifende Prozesse beschreiben, miteinander verknüpft werden können. Dies entspricht der Anforderung der *Integrationsfähigkeit* (siehe auch Kapitel 2.5).

Zur Unterstützung, um entsprechende Teildatenmodelle im SC-Umfeld modellieren zu können, wird ein Referenzmodell verwendet. Die Rede ist hierbei vom SCOR-Modell. Das SCOR-Modell liefert einen Ansatz, um die Prozesse in SCs definieren zu können, wodurch allen Partnern einer SC eine einheitliche Sprache ermöglicht wird (siehe auch Kapitel 2.2 und Kapitel 2.2.1).

Das SCOR-Modell beschreibt fünf Kernprozesse, die eine SC in verschiedene Bereiche kategorisiert bzw. thematisiert. Zu nennen sind die Kernprozesse *Planen*, *Beschaffen*, *Herstellen*, *Liefern* und *Zurückführen*. Aufgrund der Tatsache, dass der Kernprozess *Planen* im SCOR-Modell organisatorische Maßnahmen beschreibt, sind lediglich im Rahmen der Fachbegriffsmodellierung die Prozesse *Beschaffen*, *Herstellen*, *Liefern* und *Zurückführen* relevant, um die wertschöpfenden Prozesse im generischen Datenmodell darstellen zu können. Die Entwicklung der einzelnen Fachbegriffe findet separat in den jeweiligen Kernprozessen statt. Dadurch ist die Komplexität auf ein Minimum reduziert, da die Entwicklung der Fachbegriffe sich vorerst nur auf die eigenen Prozesse fokussiert.

Somit kann den Anforderungen der *Korrektheit* sowie der *Verständlichkeit* Folge geleistet werden. Dies ist darin zu begründen, dass ein Umfeld geschaffen wurde, indem eine einheitliche SC-Fachsprache für bestimmte Prozesse entwickelt bzw. festgelegt werden kann. Darüber hinaus kann der Anforderung der *Modularität* Folge geleistet werden, da die Prozesse einer SC durch jeweilige Teildatenmodelle repräsentiert werden können (siehe auch Kapitel 2.5).

Zusätzlich kann der Anforderung der *Schlichtheit* sowie der *Minimalität* Folge geleistet werden. Die Aufteilung einer SC in mehrere Teildatenmodelle trägt dazu bei, dass die einzelnen Teildatenmodelle nur die zum notwendigen Verständnis wichtigsten Inhalte eines zu beschreibenden Prozesses umfassen. Dies hat den Vorteil, dass die Komplexität im

Modellierungsprozess aufgrund einer besseren Übersicht über die Prozesse reduziert wird. Darüber hinaus können die Prozessänderungen auf bestimmte Bereiche zurückgeführt werden, wodurch eine zielgerichtete Analyse der Prozesse sichergestellt wird. Zusätzlich können Modelländerungen vorgenommen werden, ohne die Modellintegrität des gesamten konzeptionellen Datenmodells zu gefährden, wodurch auch der Anforderung der *Integrierfähigkeit* Folge geleistet werden kann (siehe auch Kapitel 2.5).

Auf Abbildung 3-3 wird beispielhaft die Verknüpfung zwischen zwei Teildatenmodellen einer SC dargestellt. Hierbei handelt es sich um die Teildatenmodelle Herstellen (TDM-H) und Liefern (TDM-L), wobei jedes Teildatenmodell für sich eigene Teildatenmodelle beinhalten kann (z.B. TDM-L1, TDM-L2), die einzelne Prozesse beschreiben.

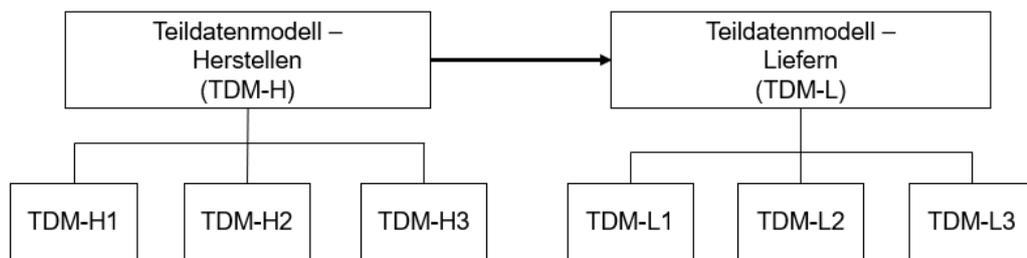


Abbildung 3-3: Beispielhafte Teildatenmodelle in Supply Chains

Zusammenfassend ist auszuführen, dass die Anforderung zur *Beschreibung des Datenmodellzwecks* (siehe auch Kapitel 3.1.3) durch das Modellieren von Teildatenmodellen ermöglicht wird. Grund dafür ist, dass die Komplexität der zu gestaltenden Prozesse auf ein Minimum reduziert wird und dadurch der Fokus im jeweiligen Fachbereich liegt. Darüber hinaus wird aufgrund der Kategorisierung der SC in verschiedene Kernprozesse eine Einschränkung des Interpretationsspielraums von Fachbegriffen ermöglicht. Das anschließende Verknüpfen der Teildatenmodelle erfordert eine Anpassung der Fachbegriffe. Dies soll im folgenden Kapitel näher beschrieben werden. Abschließend können in diesem Kapitel folgende Anforderungen zusammengefasst werden:

- Eine SC ist in Kernprozesse zu unterteilen, um die Komplexität im Entwicklungsprozess eines generischen Datenmodells zu reduzieren. Das Ergebnis dieser Kategorisierung sind Teildatenmodelle aus den Bereichen *Beschaffen*, *Herstellen*, *Liefern* und *Zurückführen*.

- Das Erstellen von Teildatenmodellen wird gefordert, um den Interpretationsspielraum der zu modellierenden Fachbegriffe auf die wesentliche Prozesse zu beschränken.

3.2.2 Sicherstellung der Modellverständlichkeit in Supply Chains

Um den Anforderungen der *Korrektheit* und der *Verständlichkeit* (siehe auch Kapitel 2.5) sowie der *Anforderung der Objektidentität* (siehe auch Kapitel 2.6) Folge leisten zu können, müssen bei der Verknüpfung der Teildatenmodelle eventuelle Anpassungen an den Fachbegriffen vorgenommen werden, um die Identität der einzelnen Fachbegriffe zu wahren. Dieser Sachverhalt soll im Folgenden und an Abbildung 3-4 näher beschrieben werden.

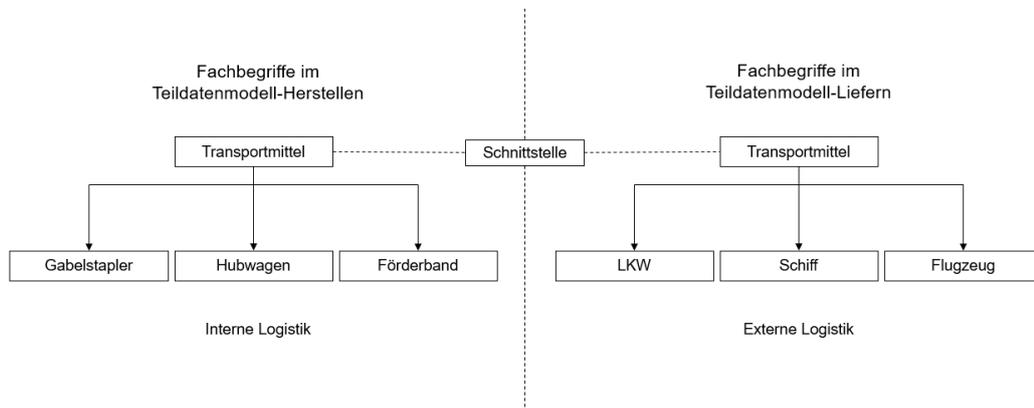


Abbildung 3-4: Gegenüberstellung der Fachbegriffe verschiedener Kernprozesse

Der generische Begriff *Transportmittel* kann sowohl für das *Teildatenmodell-Liefern* als auch für das *Teildatenmodell-Herstellen* in ihren jeweils eigenen Prozessen eindeutig sein. Im Bereich *Liefern* würde man unter dem Begriff *Transportmittel* beispielsweise LKW, Schiff und Flugzeug verstehen, womit *externe logistische Vorgänge* beschrieben werden. Im Bereich *Herstellen* würde man darunter *interne logistische Vorgänge* verstehen, wie z.B. Gabelstapler, Hubwagen und Förderbänder. Somit würde das Verknüpfen dieser beiden Teildatenmodelle zu Fehlinterpretationen führen können, sofern keine Anpassungen vorgenommen werden. In Abbildung 3-5 werden einige beispielhafte Fachbegriffe dargestellt, wodurch Objekte durch bestimmte Fachbegriffe in generischen Datenmodellen bezeichnet werden können. Diese Beispiele sollen verdeutlichen, dass die generische Beschreibung von Fachbegriffen mit Bedacht zu wählen ist.

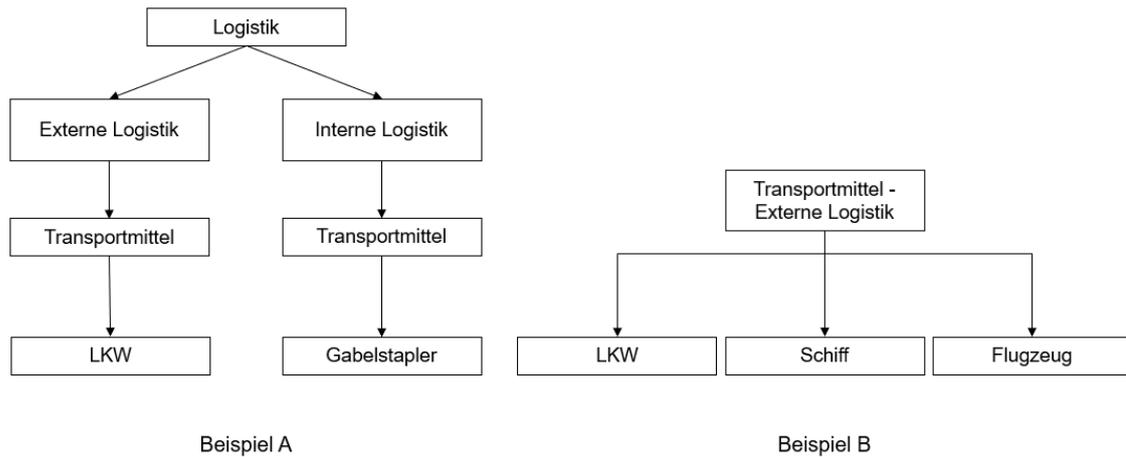


Abbildung 3-5: Identifizierende Fachbegriffe im Teildatenmodell

Mit Bezug zu Abbildung 3-5 liefert Beispiel A einen generischeren Ansatz zur Beschreibung der Prozesse. Jedoch führt dies bei der Verknüpfung von verschiedenen Teildatenmodellen nach wie vor zu Fehlinterpretationen. Dies ist der Tatsache geschuldet, dass der Begriff *Logistik* zu generisch erscheint und erst auf den nächsten Ebenen erkannt wird, welcher Prozess entsteht. Die elegantere Lösung bietet Beispiel B. Der übergeordnete Begriff *Transportmittel - Externe Logistik* ist spezifischer in der Beschreibung eines Prozesses und zeitgleich generisch genug, um sich bei Veränderungen in logistischen Prozessen flexibel anpassen zu können. Im Grunde bildet der *übergeordnete Fachbegriff* aus Beispiel B eine Kombination der beiden *untergeordneten Fachbegriffe* aus Beispiel A (Externe Logistik / Transportmittel). Abschließend können folgende Anforderungen zusammengefasst werden:

- Vermeidung der Mehrfachverwendung eines Fachbegriffs, bei der Verknüpfung von Teildatenmodellen.
- Die Verknüpfung der Teildatenmodelle fordert eine Anpassung der verwendeten Fachbegriffe. Besonders ist die Unterscheidung ähnlicher generischer Fachbegriffe sicherzustellen. Die Anpassung eines Fachbegriffs zur eindeutigen Identität kann durch die Kombination zusätzlicher Fachbegriffe gekennzeichnet werden.

3.2.3 Prozessschnittstellen in Supply Chains

Um Prozessschnittstellen in SCs zu untersuchen, wird auf Abbildung 2-4 verwiesen. Dort ist ersichtlich, dass die Kernprozesse *Beschaffen*, *Liefern* und *Zurückführen* jeweils eine Schnittstelle zu anderen SC-Partnern bzw. externen Prozessen bilden. Lediglich der Kern-

prozess *Herstellen* ist unternehmensintern angesiedelt und bleibt unberührt von den externen Prozessen.

Aus dieser Erkenntnis folgt, dass der Kernprozess *Herstellen* demzufolge durch langfristig fixe Prozesse beschrieben werden kann und somit die zu modellierenden Fachbegriffe spezifischer ausgerichtet werden können. Die Begründung liegt darin, dass ein Herstellungsprozess in der Regel unabhängig von direkten äußeren Einflüssen durchgeführt werden kann. Solange Rohstoffe zur Verfügung gestellt werden, kann der Produktionsprozess ohne Probleme fortgeführt werden. Von welchem Lieferanten die Rohstoffe bezogen werden, ist in diesem Kernprozess letztendlich egal, da der Fokus bei der Verarbeitung von Rohstoffen liegt. Zudem ist auch unwahrscheinlich, dass der interne Produktionsprozess einer häufigen Veränderung unterliegt.

Durch die Erschließung neuer Vertragspartner im SC-Netzwerk und sich dadurch neu ergebender Transportwege erhöht sich die Gefahr, dass sich an den Schnittstellen der Kernprozesse *Beschaffen*, *Liefern* und *Zurückführen* häufig sich verändernde Prozesse ergeben können, so dass hier eine weitsichtige Vorausplanung, die flexibel gestaltet ist, angedacht werden sollte.

Darüber hinaus ist zu erwähnen, dass Störfaktoren, die den konventionellen Prozess im SC-Netzwerk behindern, nicht zwangsläufig den Partnern im SC-Netzwerk zuzuschreiben sind. Es können durchaus auch politische Rahmenbedingungen dazu führen, dass eine flexible Anpassung durchgeführt werden muss. Im Folgenden wird dies durch Abbildung 3-6 verdeutlicht und näher beschrieben.

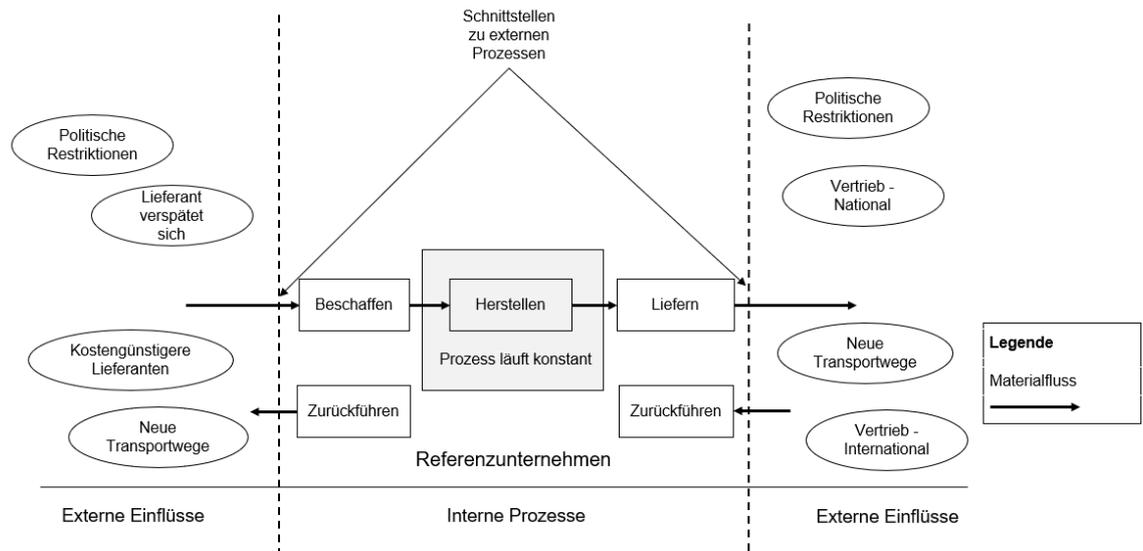


Abbildung 3-6: Einflüsse an den Schnittstellen der SC

Auf Abbildung 3-6 soll verdeutlicht werden, dass Unternehmen bzw. SC-Netzwerke wenig Kontrolle auf äußere Einflüsse haben. Beispielsweise können politische Rahmenbedingungen dazu führen, dass neue logistische Prozesse festgelegt werden müssen. Darauf aufbauend wird ausgeführt, dass im Grunde die Prozessschnittstellen im konzeptionellen Datenmodell generischer auszurichten sind, die häufig bzw. in erster Instanz den äußeren Einflüssen ausgesetzt sind. Somit wird zum einen die Möglichkeit offenbart, neue Prozesse an den Schnittstellen zu integrieren bzw. zu verändern und zum anderen eine Fehlinterpretation der Prozesse vermieden, die nicht in erster Instanz von den äußeren Einflüssen beeinträchtigt werden. Demnach können beispielsweise Prozesse im Bereich Herstellen spezifischer ausgerichtet werden, um Fehlinterpretationen zu vermeiden, da bei diesen Prozessen in der Regel abzusehen ist, wann und in welcher Form eine Anpassung durchgeführt werden muss. Abschließend können folgende Anforderungen zusammengefasst werden:

- Überprüfung der SC-Prozesse, die in Kontakt zu äußeren Einflüssen stehen
- SC-Prozessschnittstellen, die in erster Instanz den äußeren Einflüssen ausgesetzt sind, sind entsprechend generischer zu modellieren.

3.2.4 Entwicklung generischer Fachbegriffe

Die in Kapitel 2.5.1 beschriebenen Rollenkonzepte bauen auf einer hierarchischen Struktur auf. Aus diesem Grund soll in diesem Kapitel ein Schema offeriert werden, um logische

hierarchische Fachbegriffe modellieren zu können. Im Folgenden soll beispielhaft durch eine „Personenliste“ beschrieben werden, wie ein hierarchischer Aufbau von Fachbegriffen erfolgen sollte.

Angenommen, die auf der Liste beschriebenen Personen müssten kategorisiert werden, dann würden Kategorien (Begriffe) wie z.B. *Mitarbeiter*, *Kunde* oder *Lieferant* genannt werden (vgl. Abbildung 3-7). Reflektiert in ein generisches Datenmodell, ist jede *konkrete Person* ein Individuum. *Mitarbeiter*, *Kunden* und *Lieferanten* entsprechen Begriffe, die eine konkrete Person einer bestimmten Gruppe zuordnet. Diese Begriffe sind dem „übergeordneten“ Begriff *Personen* untergeordnet. Damit eine logische hierarchische Struktur entwickelt werden kann, ist diesbezüglich eine explizite Definition notwendig. Hierfür könnte man sich an der *Mengenlehre der Mathematik* orientieren, um entsprechende Definitionen zu entwickeln. Im Folgenden wird beispielhaft am Begriff *Kunde* eine Definition durchgeführt.

„Wenn alle Kunden auch Personen sind, nicht aber alle Personen auch Kunden, dann ist der Begriff Kunde ein *untergeordneter* Begriff von Personen“.

Für $A = \text{Personen}$ und $B = \text{Kunden}$, kann dies folgendermaßen mathematisch formuliert werden:

$$(1) \quad = \{B \subset A \wedge \neg(A \subset B) \wedge A \neq B\}$$

Diese mathematische Beschreibung sagt aus, dass B eine echte Teilmenge von A, jedoch A keine Teilmenge von B ist. Zusätzlich ist A nicht identisch mit B. Auf Abbildung 3-7, wird dieser Sachverhalt grafisch dargestellt.

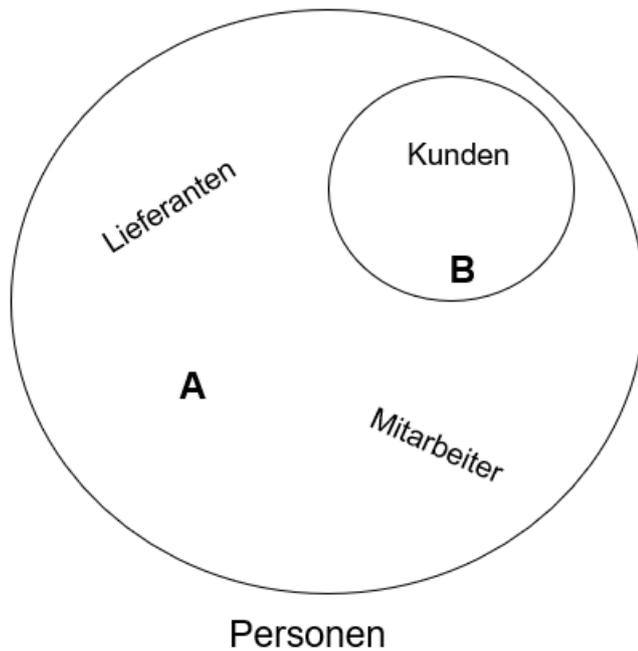


Abbildung 3-7: Hierarchische Anordnung von Fachbegriffen durch die Mengenlehre

Zusätzlich ist ausführen, dass die oben beschriebene mathematische Beschreibung eine hierarchische Struktur impliziert, da B ein Teil von A, jedoch A kein Teil von B darstellt (vgl. Abbildung 3-7). Anders betrachtet, wenn A theoretisch gesehen auch ein Teil von B darstellen könnte, dann würde keine logische hierarchische Abstufung dieser Begriffe zustande kommen. Demnach müssten neue Begriffe untersucht werden, die eine hierarchische Abstufung erlauben. Führt man dieses Definitionsschema auch mit den Begriffen Mitarbeiter und Lieferant durch, so könnten die Fachbegriffe wie beispielhaft in Abbildung 3-8 dargestellt werden.

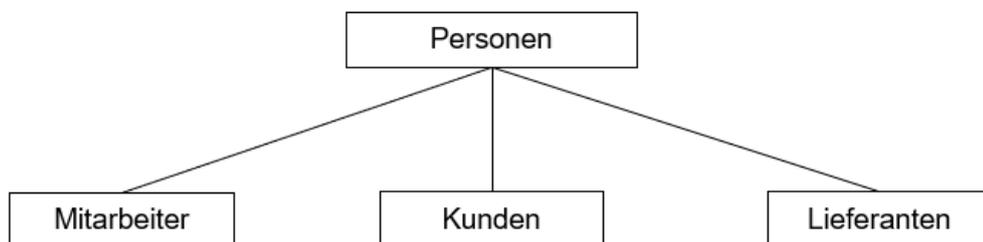


Abbildung 3-8: Beispielhafter Aufbau einer hierarchischen Struktur für generische Begriffe

Im Grunde ist die hierarchische Abstufung abhängig von den verwendeten Begriffen und von der subjektiven Einschätzung des Modellierers. Allerdings ist zu beachten, dass in

einem konzeptionellen Datenmodell, vor allem wenn es generisch modelliert ist, die gewählten Begriffe klar unterscheidbar sein müssen. Dies wurde bereits in Kapitel 3.1.2 im Rahmen der Verknüpfung von Teildatenmodellen festgestellt.

Mit Bezug zum *Terminologiemanagement* (siehe auch Kapitel 2.6) wurde ein methodisches Vorgehen beschrieben, um Begriffe für ein spezielles Umfeld zu modellieren. Dadurch soll in einem konzeptionellen Datenmodell der Konsens zwischen den verwendeten Fachbegriffen sichergestellt werden. Im Prinzip kann dieses methodische Vorgehen durch einen Regelkreis dargestellt werden (vgl. Abbildung 3-9).

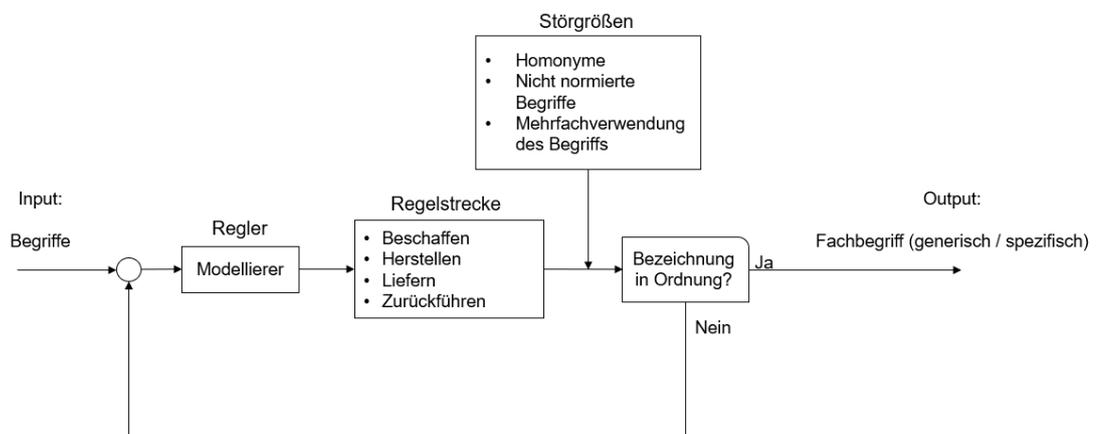


Abbildung 3-9: Die Fachbegriffsmodellierung als Regelkreis

Der Regelkreis wurde gemäß den zuvor beschriebenen spezifischen Anforderungen für das SC-Umfeld sowie nach den in Kapitel 2.6 beschriebenen methodischen Vorgehen des Terminologiemanagements visualisiert, um SC-Fachbegriffe zu modellieren. Nach dem in Abbildung 3-9 dargestellten Regelkreis, gelangen durch den Input individuelle Begriffe in den Regelkreis, die durch den *Regler* (z.B. Modellierer, Manager) definiert werden. Dabei beschreibt die *Regelstrecke*, für welchen Bereich der zu modellierende Fachbegriff vorgesehen ist. Die Regelstrecke orientiert sich an den in Kapitel 2.2.1 beschriebenen Kernprozessen des SCOR-Modells. Dementsprechend werden die modellierten Fachbegriffe dem jeweiligen Kernprozess zugeordnet. Die *Störgröße* beschreibt verschiedene Zustände eines Begriffs, die sowohl gleichzeitig als auch separat vorkommen können und die Verständlichkeit des generischen Datenmodells gefährden. Repräsentativ sind *Homonyme*, *nicht normierte Begriffe* sowie die *Mehrfachverwendung von Fachbegriffen* im generischen Datenmodell zu nennen. An der Stelle der Rückführung wird überprüft, ob der modellierte Fachbegriff den Anforderungen genügt. Sind alle Störgrößen beseitigt, so kann der Fach-

begriff ins *datenorientierte Kompendium* aufgenommen werden (siehe auch Kapitel 3.4). Ansonsten durchläuft der Begriff so lange die Regelstrecke, bis alle Störgrößen beseitigt sind.

Abschließend werden die wesentliche Anforderungen zusammengefasst, die im Rahmen der Entwicklung von hierarchischen SC-Fachbegriffen wesentlich sind:

- Homonyme sind im Rahmen der Entwicklung von normierten Fachbegriffen zu vermeiden. Grund dafür ist, dass Homonyme Begriffe mit unterschiedlicher Bedeutung beschreiben. Als Beispiel ist der Begriff „Name“ zu nennen. Dieser Begriff kann als Projektname, Ortsname, Vorname oder Nachname aufgefasst werden.
- Normierte Fachbegriffe sind für die gesamte SC zu fordern, um die Verständlichkeit der Prozesse sicherzustellen.
- Fachbegriffe werden in den jeweiligen Kernprozessen (Beschaffen, Herstellen, Liefern, Zurückführen) modelliert, um die Komplexität in der Entwicklung von Fachbegriffen zu reduzieren. Der Fokus liegt somit vorerst im jeweiligen Fachbereich, um die Prozesse explizit beschreiben zu können bzw. eine geeignete Bezeichnung zu entwickeln. Darauf aufbauend kann eine generischere Bezeichnung des Fachbegriffs erfolgen, welche durch einen hierarchischen Aufbau der Fachbegriffe realisiert werden kann. Um Fachbegriffe hierarchisch logisch aufzubauen, werden die Kenntnisse aus der Mengenlehre hinzugezogen.
- Fachbegriffe sollten nur einmalig im generischen Datenmodell vorkommen, um eine Unterscheidbarkeit der Begriffe sicherzustellen. Als Beispiel ist der Begriff Transportmittel zu nennen. Dieser Begriff ist sowohl in der Produktion als auch in der Logistik jeweils eindeutig. Erfolgt jedoch eine zeitgleiche Betrachtung der Produktion und der Logistik, so wird der Begriff unscharf. Somit muss der Begriff angepasst werden. Dabei empfiehlt es sich, den jeweiligen Begriff durch seinen Fachbereich zu definieren, z.B. „Transportmittel – Produktion“.
- Fachbegriffe sind im datenorientierten Kompendium mit Synonymen zu verknüpfen, um die eindeutige Identität eines Fachbegriffs sicherzustellen. Darüber hinaus empfiehlt es sich, generische Fachbegriffe im Plural zu modellieren, somit wird impliziert, dass eine weitere Unterteilung folgt. Dementsprechend sind spezifische Fachbegriffe im Singular zu modellieren.

Zusammenfassend ist auszuführen, dass dieses Kapitel eine Methode offeriert, um eine logisch hierarchische Abstufung der Begriffe zu ermöglichen, wodurch generische Fachbegriffe entwickelt werden können. Weiterhin wurde in Anlehnung der Methode des Terminologiemanagements eine methodische Vorgehensweise beschrieben, die die Entwicklung von SC-Fachbegriffen beschreibt. Darüber hinaus wurde festgestellt, dass die Bezeichnung von Begriffen durch Attribute gekennzeichnet wird.

Abschließend ist auszuführen, dass die modellierten SC-Begriffe über längere Zeit weder in ihrer Bezeichnung noch in ihrer Bedeutung verändert werden sollten. Dies liegt darin begründet, dass bei jeder Änderung die gesamte SC eine Anpassung durchführen muss, um die Verständlichkeit der Prozesse sicherzustellen.

3.2.5 Zusammenfassung der Anforderungen

In diesem Kapitel werden die zuvor herausgearbeiteten Anforderungen noch einmal in übersichtlicher Form zusammengefasst. Weiterhin wird noch beschrieben, wie den Anforderungen aus Kapitel 2.5 Folge geleistet werden konnte.

Die Beschreibung des Datenmodellzwecks (siehe auch Kapitel 3.1.3) dient der Übersicht des zu modellierenden Prozesses, womit der Anforderung der *Vollständigkeit* Folge geleistet werden soll. In Kapitel 2.5 wurde erläutert, dass diese Anforderung im Grunde von der subjektiven Einschätzung des Modellierers abhängig ist. Durch die Beschreibung des Datenmodellzwecks, soll der Fokus auf wesentlich zu modellierende Prozesse gelegt werden.

Darüber hinaus wurde in Kapitel 2.6 beschrieben, dass eine Thematisierung der zu modellierenden Bereiche zu fordern ist, um eine Übersicht über die Verfügbarkeit von Informationen zu erhalten. Dies führt im Grunde dazu, dass im SC-Umfeld die zu modellierenden Bereiche in verschiedene Kategorien eingeteilt werden. Im Rahmen dieser Arbeit wurde das SCOR-Modell (siehe auch Kapitel 2.2.1) dazu verwendet, um eine SC in wesentliche Kernprozesse aufzuteilen.

Dies führt zu mehreren Teildatenmodellen, die bestimmte Prozesse übersichtlich beschreiben (siehe auch Kapitel 3.2.1). Dadurch konnte den Anforderungen der Verständlichkeit und Korrektheit Folge geleistet werden. Durch das beschreiben mehrerer übersichtlicher Teildatenmodelle kann somit der Inhalt eines Modells für jeden Anwender verständlich beschrieben werden. Durch die Beschreibung des Datenmodellzwecks für jedes Teildatenmodell sollte somit auch die Verständlichkeit sichergestellt werden. Darüber hin-

aus können weniger komplexe Datenmodelle einfacher auf ihre Korrektheit der zu beschreibenden Prozesse geprüft werden. Dies führt auch zur Erfüllung der Anforderung der *Schlichtheit* sowie *Minimalität*. Die Aufteilung einer SC in mehrere Teildatenmodelle trägt dazu bei, dass die einzelnen Teildatenmodelle nur die zum notwendigen Verständnis wichtigsten Inhalte eines zu beschreibenden Prozesses umfassen.

Zusätzlich hat dies den Vorteil, dass einfacher Modelländerungen realisiert werden können, ohne die Modellintegrität des ganzen Datenmodells zu gefährden. Somit kann der Anforderung der *Integrierfähigkeit* Folge geleistet werden, da einfache und übersichtliche SC-Prozesse definitiv das Verbinden mehrerer Teildatenmodelle zu einem Ganzen fördern.

Bei der Integration von Teildatenmodellen wurde darauf hingewiesen, dass wohlmöglich eine Anpassung der Fachbegriffe erforderlich ist, um die Modellverständlichkeit sicherzustellen. Aus diesem Grund wurde die Anforderung aufgestellt, dass zu prüfen ist, ob die Mehrfachverwendung eines Fachbegriffs vorliegt. Dabei kann die Anpassung des Fachbegriffes zur eindeutigen Identität durch die Kombination zusätzlicher Fachbegriffe gekennzeichnet werden.

Zusätzlich wurde festgestellt, dass die externen Schnittstellen Bereiche kennzeichnen, auf die Unternehmen keinen direkt Einfluss haben (siehe auch Kapitel 3.2.3). Aus diesem Grund muss bei der Modellierung darauf besonders geachtet werden, dass erkannt wird, welche Prozessschnittstellen in erster Instanz den äußeren Einflüssen ausgesetzt sind, um diese entsprechend generischer modellieren zu können, um sich an verändernde Gegebenheiten flexibel anzupassen.

Im Rahmen der generischen Fachbegriffsmodellierung wurde angemerkt, dass das Modellieren von Fachbegriffen von der subjektiven Einschätzung des Modellierers abhängt (siehe auch Kapitel 2.6). Darauf aufbauend wurden Methoden beschrieben, die das Modellieren von SC-Fachbegriffen ermöglicht (siehe auch Kapitel 3.2.4).

Im Folgenden werden die herausgearbeiteten Anforderungen an ein generisches Datenmodelle für den Einsatz mit SC-Daten zusammengefasst:

Kapitel 3.1.3

- Beschreibung des Datenmodellzwecks
- SC-Fachbegriffsmodellierung als Vorstufe der Datenmodellierung
 - Beschreibung wesentlicher SC-Fachbegriffe
 - Festlegen des Interpretationsspielraums von SC-Fachbegriffen
 - Entwickeln hierarchischer (generischer) SC-Fachbegriffe

Kapitel 3.2.1

- Eine SC ist in Kernprozesse zu unterteilen, um die Komplexität im Entwicklungsprozess eines generischen Datenmodells zu reduzieren. Das Ergebnis dieser Kategorisierung sind Teildatenmodelle aus den Bereichen *Beschaffen*, *Herstellen*, *Liefern* und *Zurückführen*.
- Das Erstellen von Teildatenmodellen wird gefordert, um den Interpretationsspielraum der zu modellierenden Fachbegriffe auf die wesentliche Prozesse zu beschränken.

Kapitel 3.2.2

- Vermeidung der Mehrfachverwendung eines Fachbegriffs, bei der Verknüpfung von Teildatenmodellen.
- Die Verknüpfung der Teildatenmodelle fordert eine Anpassung der verwendeten Fachbegriffe. Besonders ist die Unterscheidung ähnlicher generischer Fachbegriffe sicherzustellen. Die Anpassung eines Fachbegriffs zur eindeutigen Identität kann durch die Kombination zusätzlicher Fachbegriffe gekennzeichnet werden.

Kapitel 3.2.3

- Überprüfung der SC-Prozesse, die in Kontakt zu äußeren Einflüssen stehen
- SC-Prozessschnittstellen, die in erster Instanz den äußeren Einflüssen ausgesetzt sind, sind entsprechend generischer zu modellieren.

Kapitel 3.2.4

- Homonyme sind im Rahmen der Entwicklung von normierten Fachbegriffen zu vermeiden. Grund dafür ist, dass Homonyme Begriffe mit unterschiedlicher Bedeutung beschreiben. Als Beispiel ist der Begriff „Name“ zu nennen. Dieser Begriff kann als Projektname, Ortsname, Vorname oder Nachname aufgefasst werden.
- Normierte Fachbegriffe sind für die gesamte SC zu fordern, um die Verständlichkeit der Prozesse sicherzustellen.
- Fachbegriffe werden in den jeweiligen Kernprozessen (Beschaffen, Herstellen, Liefern, Zurückführen) modelliert, um die Komplexität in der Entwicklung von Fachbegriffen zu reduzieren. Der Fokus liegt somit vorerst im jeweiligen Fachbereich, um die Prozesse explizit beschreiben zu können bzw. eine geeignete Bezeichnung zu entwickeln. Darauf aufbauend kann eine generischere Bezeichnung des Fachbegriffs erfolgen, welche durch einen hierarchischen Aufbau der Fachbegriffe realisiert werden kann. Um Fachbegriffe hierarchisch logisch aufzubauen, werden die Kenntnisse aus der Mengenlehre hinzugezogen.
- Fachbegriffe sollten nur einmalig im generischen Datenmodell vorkommen, um eine Unterscheidbarkeit der Begriffe sicherzustellen. Als Beispiel ist der Begriff Transportmittel zu nennen. Dieser Begriff ist sowohl in der Produktion als auch in der Logistik jeweils eindeutig. Erfolgt jedoch eine zeitgleiche Betrachtung der Produktion und der Logistik, so wird der Begriff unscharf. Somit muss der Begriff angepasst werden. Dabei empfiehlt es sich, den jeweiligen Begriff durch seinen Fachbereich zu definieren, z.B. „Transportmittel – Produktion“.
- Fachbegriffe sind im datenorientierten Kompendium mit Synonymen zu verknüpfen, um die eindeutige Identität eines Fachbegriffs sicherzustellen. Darüber hinaus

empfiehlt es sich, generische Fachbegriffe im Plural zu modellieren, somit wird impliziert, dass eine weitere Unterteilung folgt. Dementsprechend sind spezifische Fachbegriffe im Singular zu modellieren.

3.3 Empfehlung und Darstellung eines generischen Datenmodells

Das in Kapitel 2.5.3 vorgestellte Synthesekonzept wird für die Erstellung eines generischen Datenmodells empfohlen, da sich dieses Modell flexibel in seiner Anwendung zur Beschreibung von SC-Prozessen eignet. Im Folgenden liegt demnach der Fokus, welche Modellierungssprache sich am besten eignet sowie welches Rollenkonzept im generischen Datenmodell Verwendung findet.

Bereits in Kapitel 2.5.1 wurde erläutert, dass das Klassendiagramm der UML am besten für die Darstellung des Rollenkonzepts geeignet ist. Dies wurde damit begründet, dass die Darstellung in Klassendiagrammen es ermöglicht, Rollen zu identifizieren, bevor eine explizite Beziehung beschrieben wird (siehe auch Kapitel 2.5.1). Darüber hinaus wäre eine Darstellung im ERM unvorteilhaft, sobald die Prozesse komplexer werden. Dies liegt darin begründet, dass in diesem Modell die Attribute als „ovale Kreise“ dargestellt werden und dadurch die Übersichtlichkeit bzw. die Transparenz der modellierten Prozesse leidet. Aus diesem Grund wird als Modellierungssprache für die Realisierung eines generischen Datenmodell das Klassendiagramm des UML empfohlen.

Darüber hinaus muss für das SC-Umfeld das passende Rollenkonzept (Rollenmuster) gewählt werden, welches im synthetisierten Konzept (siehe auch Kapitel 2.5.3) Anwendung findet. Da in SCs sowohl viele bestehende als auch viele neue Rollen beschrieben werden müssen, wird die Verwendung des Role Objects empfohlen (siehe auch Kapitel 2.5.1).

Im weiteren Verlauf des Kapitels sollen unter Berücksichtigung der in Kapitel 3.2.5 zusammengefassten Anforderungen sowie dem in Kapitel 2.5.3 beschriebenen Synthesekonzept zur Modellierung eines generischen Datenmodells beispielhafte SC-Prozesse visualisiert werden. Dadurch soll untersucht werden, welche Herausforderungen sich bei der Modellierung von generischen Datenmodellen mit SC-Daten ergeben.

Aus Gründen der Übersichtlichkeit wird in den folgenden generischen Datenmodellen auf die Angabe von Attributen verzichtet. Im Vordergrund sollte die generische Struktur der Datenmodelle stehen, realisiert durch Powertypen und dem Rollenkonzept sowie die mögliche Verknüpfung der einzelnen generischen Datenmodelle im SC-Umfeld.

Für die Darstellung der Rollen, ist das Rollenmuster *Role Object* (Darstellungsart des Rollenkonzepts) im SC-Umfeld am besten geeignet. Dies wurde damit begründet, dass dieses Rollenmuster am besten für Datenmodelle in Frage kommt, bei denen sowohl viele bestehende als auch viele neue Rollen beschrieben werden müssen (vgl. Abbildung 2-26). Da dies im SC-Umfeld definitiv der Fall ist, findet dieses Rollenmuster im synthetisierten Konzept (siehe auch Kapitel 2.5.3) Anwendung.

In Kapitel 2.5.1 wurde bereits ausgeführt, dass eine Kategorisierung eines Fachbegriffs nach seiner *unveränderlichen Wesensart* und seinem *veränderlichen Verhaltensmuster* durchzuführen ist, wodurch ein Rollenkonzept impliziert wird, dass letztendlich den generischen Konstruktionsrahmen im konzeptionellen Datenmodell ermöglicht. Zunächst werden die in Tabelle 2-2 zusammengefassten SC-Daten herangezogen, um SC-Fachbegriffe zu entwickeln. Diesbezüglich ist anzumerken, dass die Entscheidung, ob ein Fachbegriff als Entitätstyp (Klasse), Beziehungstyp oder als Attribut dargestellt werden soll, von der subjektiven Einschätzung des Modellierers abhängt. Darüber hinaus ist noch anzumerken, dass dies auch für die zu beschreibenden Rollen in einem generischen Datenmodell gilt. Dies ist auf die dem Modellierer zugrunde liegenden Kenntnisse eines Prozesses zurückzuführen (siehe auch Kapitel 2.1.1). Gemäß der in Kapitel 3.2.4 beschriebenen Vorgehensweise werden aus den ermittelten Stamm- und Bewegungsdaten zunächst vorläufige Fachbegriffe abgeleitet (vgl. Tabelle 3-3).

Tabelle 3-3: Übersicht der vorläufigen Klassenbezeichnung

Bezeichnung aus der Literatur	Fachbegriffe für das generische Datenmodelle
Materialdaten	Produkte
Stücklisten	Stücklisten
Ressourcendaten	Ressourcen
Arbeitspläne	Arbeitspläne
Kundendaten	Kunden
Lieferantendaten	Lieferant
SC-Daten	Interne Logistik
	Externe Logistik
Lagerbestandsdaten	Lagebestand
Produktionsauftragsdaten	Aufträge
Betriebsdaten	Kapazität

Anschließend werden die erstellten Fachbegriffe ihren Fachbereichen zugeordnet (vgl. Tabelle 3-4). Die Aufteilung der Fachbegriffe erfolgt gemäß den in Kapitel 3.2.1 und Kapitel 3.2.2 herausgearbeiteten Anforderungen für das SC-Umfeld. Dabei werden die vorläufig erstellten Fachbegriffe ihren jeweiligen Fachbereichen zugeordnet. Diese Zuordnung verringert die Komplexität hinsichtlich der Entwicklung von Fachbegriffen. Dies liegt darin begründet, dass die Fachbegriffe völlig losgelöst von den Prozessen anderer Fachbereiche entwickelt werden können.

Tabelle 3-4: Zuordnung der wesentlichen Fachbegriffe in den Fachbereichen

Beschaffen	Herstellen	Liefern
Lieferant	Stücklisten	Kunden
Externe Logistik	Ressourcen	Aufträge
Interne Logistik	Arbeitspläne	Externe Logistik
Lagerbestand	Produkte	Interne Logistik
Produkte		Kapazität
		Produkte

Beiläufig ist noch anzumerken, dass nicht zwangsläufig alle in Tabelle 2-2 aufgelisteten Daten sich für den Datenbankentwurf im Rahmen einer SC eignen. Grund dafür ist, dass jeder SC-Partner als Lieferant des nachfolgenden SC-Partners agiert (siehe auch Kapitel 2.2.1). Somit würde die realisierte DB jedem nachfolgenden SC-Partner unternehmensinterne Informationen seines Lieferanten offenlegen, weshalb in der Regel keine Daten verwendet werden sollten, die den Wettbewerbsvorteil eines SC-Partners mindern könnten. In der Praxis muss jedes Unternehmen für sich entscheiden, welche Daten verwendet werden sollten. Im Rahmen dieser Arbeit ist dies jedoch kein Untersuchungsgegenstand, weshalb nicht näher darauf eingegangen wird. Allerdings sollte es vollständigshalber erwähnt werden.

In Anlehnung an die in Kapitel 3.1.3 aufgestellte Anforderung des „Datenmodellzwecks“ werden die zu beschreibenden Prozesse festgelegt. Im Folgenden soll ein generisches Datenmodell dargestellt werden, dass die Netzwerkbeziehung zwischen den SC-Akteuren beschreibt. Hierfür werden die Fachbegriffe aus den Bereich „Beschaffen“ (Fachbegriff: *Lieferant*) gewählt. Um den zu beschreibenden Prozess zu präzisieren, wird der Begriff *Lieferant* um die SC-Begriffe *Rohstofflieferant*, *Teilelieferant* und *Komponentenlieferant* erweitert. Diese Begriffe wurden für das zu beschreibende Beispiel aus der Netzwerkbeziehung in Abbildung 2-5 entnommen.

Die Rollen, die dabei ausgeübt werden, sind „vorgelagert“ und „nachgelagert“ (Lieferant). In der folgenden Tabelle (vgl. Tabelle 3-5) werden die ausgewählten SC-Begriffe nach ihrer unveränderlichen Wesensart und ihrem veränderlichen Verhaltensmuster kategorisiert.

Tabelle 3-5: SC-Fachbegriffe kategorisiert nach ihrer unveränderlichen Wesensart und ihrem veränderlichen Verhaltensmuster

Kategorie	Fachbegriff
Wesensart	Rohstofflieferant
Wesensart	Teilelieferant
Wesensart	Komponentenlieferant
Verhaltensmuster	Vorgelagert
Verhaltensmuster	Nachgelagert

Um die in Tabelle 3-5 aufgeführten Begriffe der Wesensart in einem übergeordneten Begriff zusammenzufassen, wird der Begriff *SC-Akteur* gewählt. Entsprechend für die Begriffe der Verhaltensmuster, wird der Begriff *SC-Akteur Rolle* gewählt. Nach der in Kapitel 3.2.4 beschriebenen Methode zur Festlegung der Hierarchisierung von Fachbegriffen, kann entsprechend geprüft werden, wie die Fachbegriffe aufeinander aufzubauen sind. Somit ergibt sich folgender hierarchischer Aufbau (vgl. Abbildung 3-10).

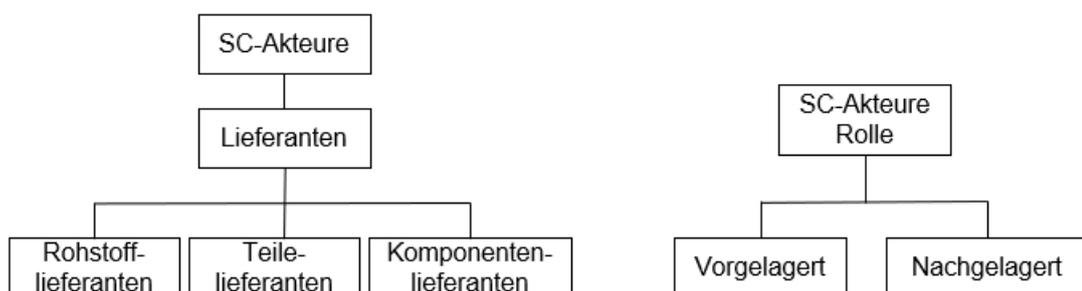


Abbildung 3-10: Hierarchischer Aufbau der SC-Begriffe im Rahmen der SC-Netzwerkbeziehung

Die Assoziation, die den Zusammenhang zwischen den SC-Akteuren beschreibt, wird *SC-Akteure Assoziation* genannt. Somit stehen drei Fachbegriffe fest, die zugleich im generischen Datenmodell als Powertypen aufgeführt werden. Zu nennen sind:

- Der Begriff *SC-Akteure* führt zum Powertyp *SC-Akteure Typ*
- Der Begriff *SC-Akteure Rolle* führt zum Powertyp *SC-Akteure Rolle Typ*
- Der Begriff *SC-Akteure Assoziation* führt zum Powertyp *SC-Akteure Assoziation Typ*

Daraus folgt ein generisches Datenmodell, das die Beziehung zwischen den SC-Akteuren (Lieferanten) beschreibt. Aus diesem Datenmodell soll hervorgehen, welcher Lieferant einem jeweiligen SC-Akteur vor- oder nachgelagert ist.

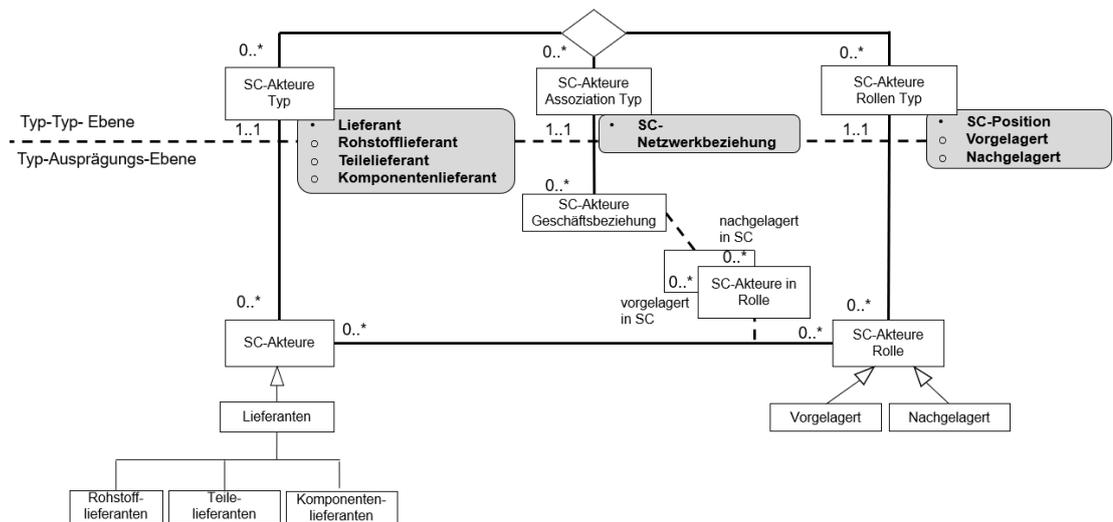


Abbildung 3-11: Generisches Datenmodell für die Netzwerkbeziehung in der Supply Chain

In Anlehnung an den in Abbildung 3-11 beschriebenen Prozess, wäre die Integration eines weiteren generischen Datenmodells möglich, um diesen um wesentliche Daten zu erweitern (siehe auch Kapitel 2.5.3). In Abbildung 3-12 werden Prozesse beschrieben, bei denen verschiedene Orte (z.B. Stadt, Land) eine Rolle (z.B. Abholadresse, Versandadresse) annehmen können.

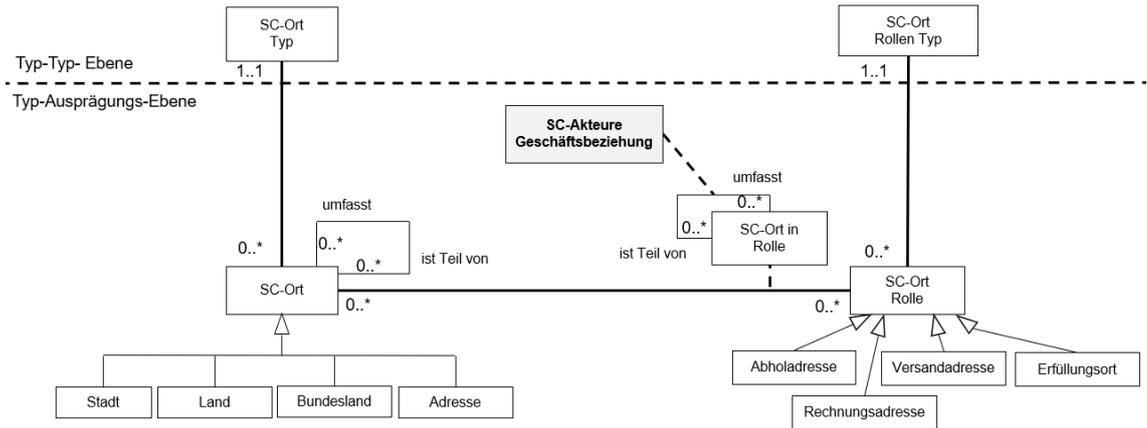


Abbildung 3-12: Generisches Datenmodell für die Adresse der SC-Akteure

Die Fachbegriffe *Stadt*, *Land*, *Bundesland* und *Adresse* werden dem übergeordneten Fachbegriff *SC-Ort* zugeordnet und beschreiben die unveränderliche Wesensart eines Gegenstands. Die Fachbegriffe *Abholadresse*, *Rechnungsadresse*, *Versandadresse* und *Erfüllungsort* werden dem übergeordneten Fachbegriff *SC-Ort Rolle* zugeordnet und beschreiben das veränderliche Verhaltensmuster eines Gegenstands. Aus den übergeordneten Klassen ergeben sich die Powertypen *SC-Ort Typ* und *SC-Ort Rollen Typ*. Die dort enthaltenden Daten können in Abbildung 3-13 um die Angabe logistischer Daten (z.B. Versandadresse, Abholadresse) erweitert werden. Des Weiteren ist auszuführen, dass die „rekursive“ Beziehung des Begriffs *SC-Ort* darin begründet liegt, dass die Begriffe *Stadt*, *Land*, *Bundesland* und *Adresse* in ihrer Rollenausübung in einer Assoziation stehen (siehe auch Kapitel 2.5.3).

Die Erweiterung bzw. Anpassung des Prozesses wird in Abbildung 3-13 durch den Fachbegriff *SC-Ort* repräsentiert.

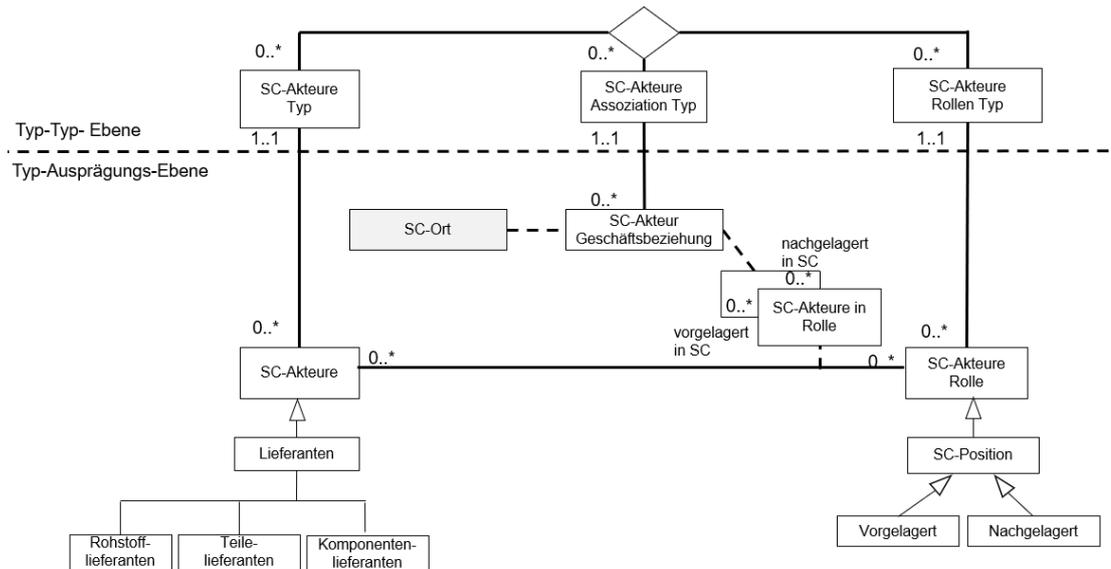


Abbildung 3-13: Zusammenführen von generischen Supply-Chain-Prozessen

Somit könnten Prozesse beschrieben werden, wodurch festgestellt werden kann, in welcher geografischen Lage sich die Lieferanten befinden und wie ihre Beziehung zueinander ist. Beispielsweise könnte durch den beschriebenen Prozess festgestellt werden, wie weit das SC-Netzwerk aufgespannt ist und ob weitere Distributionszentren erstellt werden müssen, um die logistischen Prozesse zu optimieren. Durch die leichte Erweiterbarkeit des in dieser Arbeit vorgestellten Konzepts (siehe auch Kapitel 2.5.3) wäre die Integration weiterer generischer Datenmodelle möglich.

Zusammenfassend ist auszuführen, dass die Struktur dieses generischen Datenmodellkonzepts für jeden zu realisierenden Prozess ähnlich aufgebaut ist und im Grunde jeder Prozess beschrieben werden kann. Wesentlich ist die Aufteilung des generischen Datenmodells in die *Typ-Typ-Ebene* sowie in eine *Typ-Ausprägungs-Ebene*. Auf den jeweiligen Ebenen werden die Powertypen (Typ-Typ-Ebene) sowie die Klassen (Typ-Ausprägungs-Ebene) festgelegt. Ebenso ermöglicht dieser Aufbau die Integration weiterer generischer Datenmodelle, um die Prozesse entsprechend anzupassen oder bestehende Prozesse mit neuen Informationen zu versorgen.

Die Herausforderung bei der Erstellung eines generischen Datenmodells für den Einsatz mit SC-Daten liegt bei der Festlegung der zu modellierenden Prozesse. Anders als bei Datenmodellen die spezifischer modelliert werden, sind generische Datenmodelle mit Weitsicht zu modellieren. Darunter ist zu verstehen, dass ein generisches Datenmodell dahingehend zu modellieren ist, dass es sich flexibel an sich verändernde Gegebenheiten anpassen kann. Dies setzt weitreichende Kenntnisse in den SC-Prozessen voraus, um

entsprechende Schnittstellen modellieren zu können, die die Anpassung oder Integration von Prozessen erlaubt. Allerdings ist die Integration weiterer Prozesse als unkritisch anzusehen (siehe auch Kapitel 2.5.3).

Darüber hinaus wird nach wie vor die Annahme getroffen, dass die größte Herausforderung eines generischen Datenmodells in seiner Modellverständlichkeit liegt (siehe auch Kapitel 3.1.3). Die Verwendung von allgemeinen (generischen) Fachbegriffen führt dazu, dass ein anderes Modellverständnis vorausgesetzt wird, um die dargestellten Prozesse zu verstehen, als das bei Modellen der Fall ist, die durch spezifische Fachbegriffe beschrieben werden. Zusätzlich können Begriffe verschieden interpretiert werden. Hierdurch besteht die Möglichkeit, dass Prozesse fehlinterpretiert werden können oder im Rahmen der Fachbegriffsmodellierung unterschiedliche hierarchische aufgebaute Begriffe entstehen können, was wiederum Einfluss auf die Modellverständlichkeit hat.

Aus diesem Grund ist die Verwendung eines datenorientierten Kompendiums im Rahmen der generischen Datenmodellierung für den Einsatz mit SC-Daten unabdingbar. Beispielsweise muss das datenorientierte Kompendium den hierarchischen Aufbau von Fachbegriffen beinhalten, um die Modellverständlichkeit für das SC-Netzwerk sicherzustellen. Die Anforderungen an das datenorientierten Kompendiums werden im nachfolgenden Kapitel ausführlicher beschrieben.

3.4 Datenorientiertes Kompendium

Fachbegriffe dienen zur Vereinheitlichung der Begriffswelt und werden auf Klassen (Entitätstypen, Objekttypen), Beziehungstypen oder Attribute abgebildet, um in einem konzeptionellen Datenmodell Prozesse zu beschreiben. Dieser Zusammenhang ist auf Abbildung 3-14 verdeutlicht und repräsentiert die Zuordnung der Fachbegriffe zu den Datenmodellierungskonzepten.

Die Fachbegriffsmodellierung ist wesentlich für das eindeutige Verständnis eines Anwendungsbereichs und entscheidend für die Kommunikation. Dabei beschreibt das Terminologiemanagement ein methodisches Vorgehen, um Fachbegriffe zu entwickeln (siehe auch Kapitel 2.6). Darauf aufbauend wurde in Kapitel 3.2.4 eine Methode vorgestellt, um generische Fachbegriffe zu entwickeln.

Die Ergebnisse der Fachbegriffsmodellierung lassen sich im datenorientierten Kompendium zusammenfassen. Das datenorientierte Kompendium dient der Übersicht der einheit-

lich festgelegten Fachbegriffswelt sowie die bestehenden Beziehungen zwischen den Fachbegriffen.

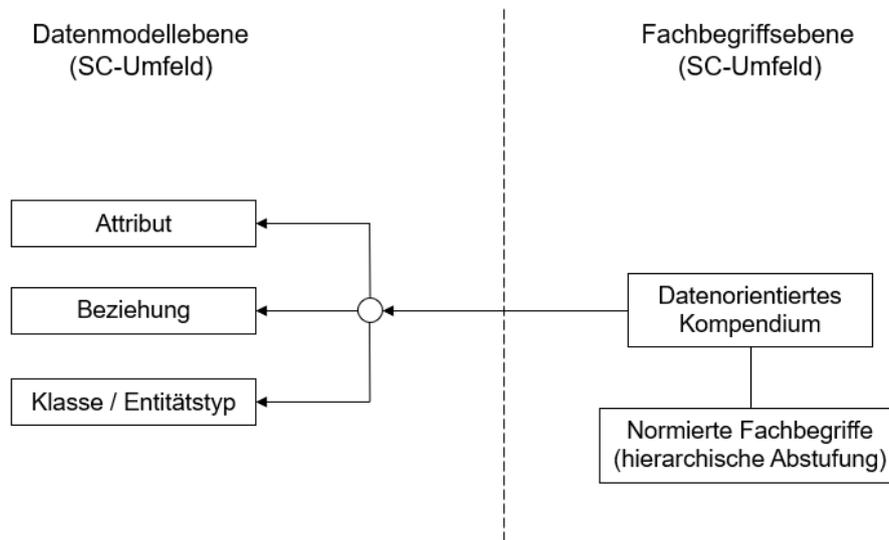


Abbildung 3-14: Das datenorientierte Kompendium als Basis eines generischen Datenmodells

Aufgrund der Tatsache, dass ein generisches Datenmodell eine Schwäche hinsichtlich der Modellverständlichkeit aufweist, ist für das generische Datenmodell ein datenorientiertes Kompendium zu fordern (siehe auch Kapitel 3.1.3 und Kapitel 3.3). Somit bildet das datenorientierte Kompendium die Basis zur Konstruktion eines generischen Datenmodells, weshalb die Fachbegriffsmodellierung als Vorstufe der generischen Datenmodellierung anzusehen ist. Im Rahmen zur Modellierung von SC-Fachbegriffen, sind folgende Anforderungen an das datenorientierte Kompendium zu nennen:

- Abgestimmte Fachbegriffe mit SC-Partnern.
- Darstellung von begrifflichen Zusammenhängen.
- Verknüpfung der Fachbegriffe mit Synonymen.
- Festlegung eines „Hauptbegriffs“ sofern verschiedene Synonyme ein Objekt beschreiben könnten.
- Aufteilung der Fachbegriffe nach ihrem veränderlichen Verhaltensmuster sowie ihrer unveränderlichen Wesensart.

- Generische Fachbegriffe sind im Plural zu modellieren. Dementsprechend sind spezifische Fachbegriffe im Singular zu modellieren.

Die zu entwickelnden Fachbegriffe werden im datenorientierten Kompendium gemäß einer SC den Bereichen *Beschaffen*, *Herstellen*, *Liefern* und *Zurückführen* zugeordnet (siehe auch Kapitel 3.2.1). Durch die Kategorisierung in verschiedene SC-Bereiche (siehe auch Kapitel 3.2.1), wird der Interpretationsspielraum der zu modellierenden SC-Fachbegriffe weitestgehend eingeschränkt und der Fokus auf den jeweiligen Prozess gelenkt.

Voraussetzung für die Verwendung eines datenorientierten Kompendiums ist die Abstimmung der zur grundlegenden Definition der wesentlichen Begrifflichkeiten mit anderen SC-Partnern. Dadurch soll die hohe Modellverständlichkeit sichergestellt werden.

Die Darstellung von begrifflichen Zusammenhängen ermöglicht vorab für das spätere Datenmodell eine Übersicht, über die Beziehungen zwischen den Fachbegriffen. Zusätzlich müssen Fachbegriffe mit Synonymen verknüpft werden, um die eindeutige Identität eines Fachbegriffs sicherzustellen. Diesbezüglich ist ein „Hauptbegriff“ festzulegen, sofern verschiedene Synonyme ein Objekt beschreiben könnten. Um eine generische Beschreibung im Datenmodell zu ermöglichen, wodurch verschiedene Prozesse definiert werden können, ist eine Aufteilung der Fachbegriffe nach ihrem veränderlichen Verhaltensmuster sowie ihrer unveränderlichen Wesensart durchzuführen. Als letztes ist auszuführen, dass generische Fachbegriffe im Plural zu modellieren sind. Somit wird impliziert, dass weitere Unterteilungen möglich sind. Dementsprechend sind spezifische Fachbegriffe im Singular zu modellieren.

Zusammenfassend ist auszuführen, dass das datenorientierte Kompendium die Vereinheitlichung der Fachbegriffe fördert, um eine Verbesserung der Kommunikation sowohl unternehmensintern als auch unternehmensübergreifend sicherzustellen. Durch die gezielte Auseinandersetzung mit den zu verwendeten Fachbegriffen, werden Homonyme vermieden. Dies führt zu einer besseren Verständlichkeit sowie Verbesserung der Zusammenarbeit zwischen den SC-Partnern, welches auf das bessere Verständnis der Prozesse zurückzuführen ist. Im Folgenden wird beispielhaft ein datenorientiertes Kompendium für generische Datenmodelle im SC-Umfeld dargestellt (vgl. Tabelle 3-6) und beschrieben.

Das datenorientierte Kompendium ist in die SC-Fachbereiche (z.B. *Beschaffen*, *Herstellen*) aufgeteilt. Dabei beschreibt die *Position* eine fortlaufende Nummer, um auf den jeweiligen Fachbegriff zu verweisen.

Die Kategorie *Synonyme* dient dazu, einen Gegenstand eindeutig durch Fachbegriffe zu beschreiben. Darüber hinaus muss bei einer größeren Auswahl an Begrifflichkeiten ein „Hauptbegriff“ festgelegt werden.

Die Kategorie *Beschreibung* dient dem Zweck, um einen Begriff eindeutig zu identifizieren. Hierfür muss eine Definition festgelegt werden, die für jeden Anwender verständlich ist. Die gewählte Definition muss zum Ausdruck bringen, was unter dem jeweiligen Begriff explizit zu verstehen ist.

Die Kategorien *Wesensart / Verhaltensmuster* und *Hierarchische Anordnung* dienen dem Zweck, eine normierte hierarchische Begriffswelt aufzubauen. Bereits zuvor wurde beschrieben, dass das Modellieren von Fachbegriffen von der subjektiven Einschätzung eines Modellierers abhängt (siehe auch Kapitel 3.3). Um die Modellverständlichkeit für alle Anwender eines generischen Datenmodells sicherzustellen, werden im datenorientierten Kompendium hierarchische Fachbegriffe vorab definiert. Die Hierarchische Anordnung beschreibt in welcher Reihenfolge die Fachbegriffe hierarchisch aufgebaut sind. In Anlehnung an Tabelle 3-6 ist der Fachbegriff *SC-Akteur* auf der höchsten Stufe. Die Fachbegriffe *Rohstofflieferant* und *Teilelieferant* sind beide auf der gleichen Ebene dem Fachbegriff *SC-Akteur* untergeordnet. Dies wird durch die Klammer (5,7) ausgedrückt.

Darüber hinaus können den Fachbegriffen *Attribute* zugewiesen werden. Es ist auch durchaus denkbar eine Kategorie einzurichten, um Powertypen festzulegen. Die Kategorie *Aktualität* gibt darüber Auskunft, wann ein Fachbegriff zuletzt überarbeitet wurde.

Tabelle 3-6: Beispielhafte Darstellung eines datenorientierten Kompendiums für ein generisches Datenmodell im Umgang mit SC-Daten

Fachbereich: Beschaffen							
Position	Fachbegriff	Synonyme	Beschreibung	Wesensart / Verhaltensmuster	Hierarchische Anordnung	Attribute	Aktualität
A							
1							
2							
3							
4							
R							
5	Rohstofflieferant		Im SC-Netzwerk den Teilelieferanten vorgelagert.	Verhaltensmuster			06.09.2018
S							
6	SC-Akteur	SC-Partner	Fasst die Teilnehmer im SC-Netzwerk zusammen	Wesensart	Position: 8 , (5,7)	Name, ID,	06.09.2018
T							
7	Teilelieferant		Im SC-Netzwerk den Rohstofflieferanten nachgelagert.	Verhaltensmuster			06.09.2018

4 Zusammenfassung und Ausblick

Das Ziel dieser vorliegenden Masterarbeit ist es die Anforderungen generischer Datenmodelle für den Einsatz mit SC-Daten zu untersuchen.

Bezugnehmend darauf wurde zu Beginn dieser Arbeit eine Zielsetzung sowie eine Vorgehensweise formuliert und ausgearbeitet. Im Anschluss daran wurden wesentliche SC-Daten herausgearbeitet. Dabei wurden die Themengebiete „Ziele des SCM“ und „Unterscheidung von Zeichen, Daten, Information und Wissen“ näher erläutert, um SC-Daten herauszuarbeiten, die eine Praxisrelevanz erfüllen.

Darauf folgte die Untersuchung von Referenzmodellen. Im Rahmen dieser Ausarbeitung wurde vornehmlich das SCOR-Modell untersucht. Hierbei wurde untersucht, welche grundlegenden Prozesse in SCs zu berücksichtigen sind.

Zur weiteren Untersuchung der Aufgabenstellung wurden sowohl die Grundlagen des Datenbankentwurfsprozesses untersucht als auch wesentliche Datenbankmodelle. Dies gab darüber Aufschluss, wie der allgemeine Entwicklungsprozess von Datenbanken aussieht und in welche Entwicklungsphase ein generisches Datenmodell einzuordnen ist. Anschließend wurden die Anforderungen eines generischen Datenmodells herausgearbeitet. Darauf aufbauend wurden Methoden und Konzepte untersucht, um ein konsensfähig generisches Datenmodell zu modellieren. Hierbei wurden die Disziplinen der Ontologie und des Terminologiemanagements untersucht. Im Fokus dieser Disziplinen steht die Darstellung von Fachbegriffen im konzeptionellen Datenmodell, um die Modellverständlichkeit sicherzustellen. Zusätzlich wurde aus der Untersuchung von Rollenkonzepten (engl. role-patterns) und Powertypen ein Synthesekonzept vorgestellt. Das Synthesekonzept basiert auf Grundlage des Rollenkonzepts sowie die Verwendung von Powertypen, um eine generische Darstellung eines konzeptionellen Datenmodells zu ermöglichen.

Anschließend wurden die zuvor beschriebenen Anforderungen, Methoden und das Synthesekonzept auf das SC-Umfeld reflektiert. Dadurch ließen sich Anforderungen für generische Datenmodelle für den Einsatz mit SC-Daten ableiten. Darüber hinaus wurden die Schwächen eines generischen Datenmodells herausgestellt, um zusätzliche Anforderungen herauszuarbeiten und zu untersuchen. Die Anforderungen sollen dabei aussagen, wie die Modellverständlichkeit im SC-Umfeld sichergestellt werden kann, wie generische SC-Fachbegriffe entwickelt werden können und welche Maßnahmen durchzuführen sind, um die generische Darstellung für ein konzeptionelles Datenmodell im SC-Umfeld realisieren

zu können. Aus den zuvor herausgearbeiteten SC-Daten wurden SC-Fachbegriffe entwickelt, um beispielhaft ein generisches Datenmodell durch SC-Prozesse darzustellen.

Die Schwierigkeit dieser Arbeit lag darin, wie den Anforderungen eines generischen Datenmodells Folge geleistet werden kann, um SC-Daten weitestgehend für die Darstellung im generischen Datenmodell vorzubereiten bzw. die Methode für SC-Daten zu ermöglichen. Durch das SCOR-Modell konnten allgemeine Prozesse herangezogen werden, um die Anforderungen eines generischen Datenmodells auf das SC-Umfeld zu übertragen. Allerdings sind hierbei genauer zu beschreibende Prozesse erforderlich, um diese generisch sinnvoll darzustellen. Jene Prozesse können von Unternehmen zu Unternehmen oder von Branche zu Branche unterschiedlich sein, weshalb sich kein allgemeingültig generisches Datenmodell darstellen lässt. Demzufolge ist für jedes SC-Netzwerk ein auf seine Bedürfnisse zugeschnittenes generisches Datenmodell zu entwickeln.

Als Fazit lässt sich festhalten, dass das Ziel erfüllt wurde und dass die herausgearbeiteten Anforderungen an ein generisches Datenmodell für den Einsatz mit SC-Daten herangezogen werden können, um einen Datenmodellierer hinsichtlich der auszuführenden Tätigkeiten zur Erstellung eines generischen Datenmodells für SC-Prozesse zu sensibilisieren. Somit sind die in dieser Arbeit beschriebenen Anforderungen an konkrete Prozesse anzupassen. Dementsprechend wurde hiermit die Vorarbeit geleistet, um generische Datenmodelle zu erstellen, die in einzelnen Projekten als Entwurfsmuster benutzt werden können.

5 Anhang

Begriffslexikon

Begriff	Beschreibung
A	
Aggregat	Das Ergebnis aus der Kombination mehrerer Entitätstypen zu einem neuen Entitätstypen wird als Aggregat bezeichnet [vgl. Schneider 2007a, S. 249].
Aggregation	Unter einer Aggregation versteht man ein Konzept, nach dem durch die Kombination von Entitätstypen ein neuer zusammengesetzter Entitätstyp (<i>Aggregat</i>) entsteht [vgl. Elmasri und Navathe 2002, S. 126]
Anforderung	Bedingung, Kriterium an einen Sachverhalt [vgl. Duden-Online, o.V. o.J., o.S.].
Arbeitsgang	Die Arbeitspläne werden durch Arbeitsgänge beschrieben, die im jeweiligen Produktionsprozess anfallen. Die Arbeitsgänge werden durch ihre Zeiten definiert [vgl. Schuh und Stich 2012, S. 76].
Aspekt	Blickrichtung, Blickwinkel, Betrachtungsweise, Gesichtspunkt [vgl. Duden-Online, o.V. o.J., o.S.].
Attribut	Merkmale, die beispielsweise eine Klasse näher beschreiben, werden <i>Attribute</i> genannt. Attribute können künstlich erdacht oder tatsächlich beobachtete Merkmale darstellen [vgl. Unland und Pernul 2015, S. 34].
Ausprägung	Eine Ausprägung beschreibt eine bestimmte Erscheinungsform eines Gegenstandes [vgl. Schneider 2007a, S. 456].

B	
Begriff	Ausdruck, Wort [vgl. Duden-Online, o.V. o.J., o.S.]. (Im Verlauf der Arbeit wird nicht explizit zwischen Fachbegriff und Begriff unterschieden).
Begriffsmodellierung	Das Erstellen einer Bezeichnung (Begriffs) für beispielsweise eine Klasse. Wird ein Begriff für ein Fachbereich erstellt, so ist auch die Rede von einer Fachbegriffsmodellierung. Das Erstellen von Begriffen wird durch das Terminologiemanagement beschrieben [vgl. Hellmuth 1997, S. 52f].
Teildatenmodell	Bei einem Teildatenmodell handelt es sich um ein konzeptionelles Datenmodell, das nach einem spezifischen Themenbereich modelliert ist. Dabei können die Inhalte generisch oder spezifisch ausgerichtet sein (vgl. Kapitel 3.2.1).
Beziehung	Eine Beziehung beschreibt auf sachlogischer Ebene das Verhältnis zwischen zwei oder mehreren Entitätstypen oder Entitäten [vgl. Voss 2008, S. 303].
Beziehungstyp	Eine Beziehung auf (Entitäts)typebene wird als <i>Beziehungstyp</i> bezeichnet [Herrmann 2018, S. 18].
D	
Datenmanipulation	Als Datenmanipulation sind alle Prozesse gemeint, um zu einem Datenbestand neue Elemente hinzuzufügen, alte zu löschen oder zu ändern [vgl. Sieberichs und Krüger 1993, S. 609].
Datenmodellierung	Die vereinfachte Beschreibung eines realen Systems. Ein Aufgabengebiet das sich der Konstruktion von Datenmodellen widmet [vgl. Schneider 2007a, S. 281].

Datenmodellqualität	Die Datenmodellqualität beschreibt den Grad der Übereinstimmung zwischen Anspruch und Erwartung. Im Sinne der Datenmodellqualität werden Ansprüche (Anforderungen) aufgestellt, die dazu führen, ein Datenmodell mit einer bestimmten Vorstellung zu entwickeln [vgl. Schneider 2007a, S. 317].
Datensatz	In einem Datensatz sind zusammenhängende Daten festgehalten (z.B. Artikelnummer und Artikelname) [vgl. Gabler Wirtschaftslexikon, Siepermann o.J., o.S.].
Datenstruktur	Daten die in einem logischen Zusammenhang stehen [vgl. Gabler Wirtschaftslexikon, Siepermann o.J., o.S.].
Diskurswelt	Ein zu modellierender Ausschnitt. Teil der Realität.
E	
Eigenfertigungsteile	Erzeugnisse aus eigener Fertigung.
Entity-Relationship-Datenmodell	Eine Modellierungssprache zur Erstellung von konzeptionellen Datenmodellen. Im Kontext dieser Arbeit wird das ERM aufgrund der Definition einiger Autoren als <i>konzeptionelles Datenmodell</i> bezeichnet [vgl. Geisler 2009, S. 54].
Entität	Ein konkreter oder abstrakter Gegenstand [vgl. Unland und Pernul 2015, S. 34].
Entitätstyp	Ein <i>Entitätstyp</i> ist eine zentrale Darstellung der datenorientierten Sichtweise. Entitäten mit ähnlichen Attributen werden zu einem übergeordneten Entitätstyp zusammengefasst [vgl. Gadatsch 2017, S. 9].
Erfolgspotenzialen	Beschreibt eine Fähigkeit, dauerhaft Erfolge zu erzielen (z.B. im Vergleich zum Wettbewerb ein überlegeneres Logistiknetzwerk) [vgl. Gabler Wirtschaftslexikon, Weber o.J., o.S.].

F	
Fachbegriff	Ausdruck, Wort (Begriff) in einem spezifischen Tätigkeitsfeld [vgl. Duden-Online, o.V. o.J., o.S.]. (Im Verlauf der Arbeit wird nicht explizit zwischen Fachbegriff und Begriff unterschieden)
Formalismus	Eine Sprache mit präzisen Regeln.
Formalziel	Übergeordnetes Ziel, an denen sich Sachziele auszurichten haben [vgl. Wellbrock 2015, S. 46].
G	
Gegenstand	Nicht näher beschriebene Sache, Ding. Neutraler Ausdruck einer Sache [vgl. Duden-Online, o.V. o.J., o.S.].
Generalisierung	Bei der Generalisierung ist der Verlauf von verschiedenen untergeordneten Entitätstypen zu einem übergeordneten Entitätstyp [vgl. Ferstl und Sinz 2013, S. 166].
Generisch	Allgemein, unspezifiziert [vgl. Schneider 2007a, S. 408].
Generisches Datenmodell	Allgemeines, unspezifiziertes konzeptionelles Datenmodell [vgl. Schneider 2007a, S. 406].
H	
Hierarchische Struktur	Die Anordnung von Objekten in einer Hierarchie impliziert eine Rangordnung, nach der Objekte geordnet werden.
Homonym	Als Homonym bezeichnet man ein Wort, das für verschiedene Begriffe steht [vgl. Reichenberger 2010, S. 6].
I	
Implementative Datenbankmodelle	Dieses Datenbankmodell beschreibt ein (DBMS) abhängiges Datenmodell und beschäftigt sich mit der Auslegung der Datenspeicherung in Datenbanken sowie der zu erzeugenden Datenstrukturen [vgl. Geisler 2009, S. 54].

Informationssystem	Ein System zur Speicherung, Wiedergewinnung und Verarbeitung von Informationen. Wird als Synonym verwendet für computergestützte Anwendungssysteme [vgl. Krcmar 2015, S. 7ff].
K	
Kanten	Verbindungen zwischen beispielsweise Entitätstypen, die durch die Angabe einer Kardinalität die Beziehung präzisieren [vgl. Mayer 1988, S. 64].
Kardinalität	Das Verhältnis zwischen den Entitäten zweier verschiedener Entitätstypen, der durch einen Beziehungstyp beschrieben wird [Herrmann 2018, S. 18].
Klasse (Diagramm, Datenmodell)	In der UML werden Klassendiagramme verwendet, um die Beziehungen, Attribute und Methoden von Klassen zu beschreiben [vgl. Alpar et al. 2016, S. 384f].
Kohärenz	Zusammenhang, Abstimmung [vgl. Duden-Online, o.V. o.J., o.S.].
Komponente	Bestandteil, Element eines Ganzen [vgl. Duden-Online, o.V. o.J., o.S.].
Komposition	Eine strenge Form der Aggregation ist die <i>Komposition</i> . Hierbei sind die Komponenten eines Aggregats von dessen Existenz abhängig [vgl. Becker et al. 2012, S. 11].
Konsistent (Konsens)	Zusammenhängend in der Gedankenführung, z.B. konsistente Begriffe [vgl. Dengel 2012, S. 137].
Konzept	Entwurf, Rohfassung. Im weitesten Sinne „Plan“ [vgl. Duden-Online, o.V. o.J., o.S.].
Konzeptionelles Datenmodell	Ein konzeptionelles Datenmodell ist ein fachbegriffliches Datenmodell zur Beschreibung von Fachbegriffen und ihren Beziehungen [vgl. Schneider 2007a, S. 152; S. 293].

M	
Merkmal	Charakteristisches, unterscheidendes Zeichen [vgl. Duden-Online, o.V. o.J., o.S.].
Methode	Eine Methode beschreibt die Art der Durchführung [vgl. Duden-Online, o.V. o.J., o.S.].
Modellintegrität	Die Bezeichnung eines korrekten Inhalts im Modell.
(Daten)Modellierungskonzepte	Entwürfe, die zur Darstellung eines Modells verhelfen (z.B. Entitätstypen, Klassen, Attribute).
N	
Netzwerkstruktur	Verbund mehrerer Entitäten (Gestaltungsmöglichkeiten sind „freier“ als bei einer hierarchischen Struktur) [vgl. Mayer 1988, S. 64].
Normatives Modell	Ein normatives Modell setzt sich aus einem vordefinierten Ansatz zusammen und beschreibt wie ein Objekt des Modells gesehen wird und wie es sich verhalten sollte [vgl. Bolstorff 2007, S. 16].
O	
Objekt (Datenmodell)	Ein Objekt ist ein konkretes Merkmal einer Klasse. Im Prinzip können Objekte mit Entitäten verglichen werden.
Objekttyp	Ein Objekttyp ist vergleichbar mit dem Entitätstyp. Objekte mit ähnlichen Attributen werden zu einem übergeordneten Objekttyp zusammengefasst (siehe auch Entitätstyp).
Ontologie	Die Ontologie in der Informatik befasst sich mit der Darstellung von Begriffen und der zwischen ihnen bestehenden Beziehungen in einem bestimmten Anwendungsbereich [vgl. Dengel 2012, S. 64ff].
Organisationsstruktur	Unternehmensgliederung [vgl. Gabler Wirtschaftslexikon, Schewe o.J., o.S.].

P	
Polyzentrische-Supply-Chain	In SC-Netzwerken gibt es keine Führung des Netzwerks. Daher verteilt sich die Organisation von unternehmensübergreifenden Prozessen gleichberechtigt über alle SC-Teilnehmer. Dieses Netzwerk zeichnet sich durch komplexe und kooperative Beziehungen der SC-Partner aus [vgl. Werner 2017, S. 27ff].
Prozesselemente	Das Prozesselement beschreibt die kleinste Einheit in einem Prozess [vgl. Werner 2017, S. 75f].
Prozessspezifikation	Die detaillierte Beschreibung eines Vorgangs.
Q	
Qualitätskriterium	Die zu erfüllenden Bedingungen, um ein Endergebnis zu erzielen.
R	
Redundanz	Der Begriff Redundanz beschreibt eine Einheit (z.B. Daten, Information) die mehrfach vorhanden ist [vgl. Chu und Zhang 1997, S. 258].
Referenzmodell	Referenzmodelle werden dazu verwendet, Geschäftsprozesse zu systematisieren und einheitlich zu gestalten. Das Referenzmodell dient der Konstruktionsunterstützung von spezifischen Modellen [vgl. Krcmar 2015, S. 44].
S	
Sachziel	Konkretes Handeln bei der Ausübung verschiedener Funktionen [vgl. Wellbrock 2015, S. 46].
Semantik	Die Theorie über die Bedeutung von Zeichen. Zeichen können dabei verschiedene Symbole, Zahlen oder Buchstaben darstellen (z.B. Darstellungen im ERM) [vgl. Gabler Wirtschaftslexikon, Thommen o.J., o.S.].

Semantisches Datenmodell	Das semantische Datenmodell beschreibt im Rahmen der Datenmodellierung eine abstrakte, formale Beschreibung und Darstellung eines Ausschnitts der „realen Welt“ [vgl. Scheller 1993, S. 87].
Spezialisierung (Vererbung)	Bei der Spezialisierung ist der Verlauf von einem übergeordneten Entitätstyp zu verschiedenen untergeordneten Entitätstypen [vgl. Ferstl und Sinz 2013, S. 166].
Spezifisches Datenmodell	Ein Datenmodell das in Hinblick eines Sachverhaltes genauer beschrieben ist (siehe auch Kapitel 3.1.2).
Strukturelle Unabhängigkeit	Befreit von einer visuellen Darstellung (z.B. relationale Datenmodelle)
Störfaktoren	Einflüsse die Prozesse behindern.
Subtyp und Supertyp	Bei der Vererbung erfolgt eine Unterscheidung zwischen einem übergeordneten Entitätstyp (<i>Supertyp</i>) und einem untergeordneten Entitätstyp (<i>Subtyp</i>) [vgl. Schneider 2007a, S. 252].
Synonym	Synonyme sind Ausdrücke und Zeichen, die den gleichen oder einen sehr ähnlichen Bedeutungsumfang haben [vgl. Reichenberger 2010, S. 6].
Syntaxregeln	Beschreibt Regeln, nach denen Zeichen gebildet werden [vgl. Gabler Wirtschaftslexikon, Lackes o.J., o.S.].
System	Ein System beschreibt die Menge zusammenhängender Elemente, die durch Relationen verknüpft sind [vgl. Gabler Wirtschaftslexikon, Gillenkirch o.J., o.S.].
U	
Unified Modelling Language	Eine Modellierungssprache zur Erstellung von konzeptionellen Datenmodellen [vgl. Unterstein und Matthiessen 2013, S. 160].

V	
Vererbung	Die Vererbung beschreibt ein Konzept, bei dem Vererbungshierarchien von Entitätstypen geschaffen werden [vgl. Schneider 2007a, S. 252].
Z	
Zweckorientiertes Wissen	Unter zweckorientiertes Wissen ist zu verstehen, dass nur solches Wissen als Information zu definieren ist, das dazu dient, Entscheidungen oder Handeln vorzubereiten [vgl. Krcmar 2015, S.5].

A Zielerreichungsgrad bestehender Supply-Chain-Management-Konzepte

Die empirische Studie über den Zielerreichungsgrad bestehender SCM-Konzepte wurde 2011 durch den Lehrstuhl für Allgemeine Betriebswirtschaftslehre und Logistik der Philipps-Universität Marburg durchgeführt (vgl. Tabelle 5-1) [vgl. Wellbrock 2015, S. 48f]. Die Bewertung erfolgt auf einer Skala von eins (sehr gering) bis fünf (sehr groß). Aus der Evaluation geht hervor, dass die *Erhöhung Produktverfügbarkeit* mit 4,33 Punkten eine hohe Relevanz für Unternehmen darstellt. Dies liegt darin begründet, dass die Erhöhung des Kundennutzens die Zufriedenheit der Kunden steigert und somit auch die Bindung an die beteiligten Unternehmen. Die Steigerung des Endkundenutzens spielt eine tragende Rolle im SCM und stellt die Wettbewerbsfähigkeit der gesamten SC sicher [vgl. Handlbauer et al. 2003, S. 8-10]. Demgegenüber bildet das Schlusslicht die *Verbesserung der Produktqualität* mit einer mittelmäßigen Relevanz.

Tabelle 5-1: Zielerreichungsgrad bestehender SCM-Konzepte [vgl. Wellbrock 2015, S. 51]

	Ziele des Supply-Chain-Management	Ergebnisse aus der deskriptiven Statistik
		Mittelwert
Endkundenutzen	Erhöhung der Produktverfügbarkeit	4,33
	Erhöhung der kundenspezifischen Individualität der Produkte	3,35
	Verbesserung des Logistikservices	3,98
Kosten	Optimierung der Transportkosten	3,83
	Abbau der Material- und Warenbestände	4,07
	Möglichst effizienter Einsatz der Ressourcen	4,30
	Reduzierung der Administrations- und Planungskosten	3,44
	Reduzierung der Transaktionskosten	3,46
	Reduzierung der Forschungs- und Entwicklungskosten	2,57
Zeit	Verkürzung der Durchlaufzeit	4,26
	Verkürzung der Forschungs- und Entwicklungszeit	3,01
	Verkürzung der Wiederbeschaffungszeit	3,72
	Verkürzung der Reaktionszeit auf Nachfrageänderungen (Time-to-react)	4,03
Qualität	Verbesserung der Produktqualität	3,67
	Erhöhung des Innovationsgrades der Produkte	3,49
Flexibilität	Verbesserung der Flexibilität gegenüber externen Einflussfaktoren	3,90
	Verbesserung der Flexibilität gegenüber Nachfrageänderungen der Endkunden	4,06
	Verbesserung des Weiterentwicklungspotenzials der Supply Chain	4,05

Literaturverzeichnis

- Abts, Dietmar; Müller, Wilhelm: Grundkurs Wirtschaftsinformatik. Eine kompakte und praxisorientierte Einführung. 7. Aufl., Wiesbaden: Vieweg + Teubner, 2011.
- Alpar, Paul; Alt, Rainer; Bensberg, Frank; Grob, Heinz Lothar; Weimann, Peter; Winter, Robert: Anwendungsorientierte Wirtschaftsinformatik. Strategische Planung, Entwicklung und Nutzung von Informationssystemen. 8. Aufl., Wiesbaden: Springer Vieweg, 2016.
- Ball, Michael O.; Chen C.-Y.; Zhao, Z.-Y.: Available to Promise. In: Simchi-Levi, David; Wu, David; Shen, Zuo-Jun: Handbook of Quantitative Supply Chain Analysis. Modeling in the E-Business Era. 1. Aufl., New York: Springer Science+Business, 2004.
- Becker, Jörg; Probandt, Wolfgang; Vering, Oliver: Grundsätze ordnungsmäßiger Modellierung. Konzeption und Praxisbeispiel für ein effizientes Prozessmanagement. Berlin: Springer-Verlag, 2012.
- Becker, Jörg; Schütte, Reinhard: Handelsinformationssysteme. Domänenorientierte Einführung in die Wirtschaftsinformatik. 2. Aufl., Frankfurt am Main: Redline Wirtschaft, 2004.
- Becker, Jörg; Rosemann, Michael; Schütte, Reinhard: Entwicklungsstand und Entwicklungsperspektiven der Referenzmodellierung. Arbeitsbericht Nr. 52. Münster: Institut für Wirtschaftsinformatik der WWU Münster, 1997.
- Bodendorf, Freimut: Daten- und Wissensmanagement. Zweite, aktualisierte und erweiterte Auflage. Berlin: Springer-Verlag, 2006.
- Bolstorff, Peter A.; Rosenbaum, Robert G.; Poluha, Rolf G.: Spitzenleistungen im Supply Chain Management. Ein Praxishandbuch zur Optimierung mit SCOR. Berlin, Heidelberg: Springer-Verlag, 2007.
- Bretzke, Wolf-Rüdiger: IT-Systeme im Supply Chain Management. 2006. URL: http://www.bretzke-online.de/downloads3/IT-Systeme_im_SCM.pdf. (letzter Abruf am 08.09.2019).
- Burkert, Bernd: Ein Framework generischer objekt-orientierter Basiskonzepte zur Werkzeug-Integration in unterschiedliche konzeptionelle Datenmodelle. Paderborn: Universität-Gesamthochschule Paderborn, 1995.
- Chu, Wesley W.; Zhang, Guogen: Associations and Roles in Object-Oriented Modeling. In: Embley, David W.; Goldstein, Robert C.: Conceptual Modeling – ER '97. 16th International Conference on Conceptual Modeling, Los Angeles, California, USA November 3-5 1997 Proceedings. Berlin, Heidelberg: Springer, 1997.

- Corsten, Hans; Gössinger, Ralf: Einführung in das Supply Chain Management. 2. Aufl., München: Oldenbourg Wissenschaftsverlag, 2008.
- Crezelius, Susanne; Hallbauer, Martin; Weltring, Rainer: PROMIDIS Handlungsleitfaden. Produktivitätsmanagement für industrielle Dienstleistungen stärken. 2015. URL: <https://www.inf.uni-hamburg.de/de/inst/ab/itmc/research/completed/promidis/instrumente/service-katalog>. (letzter Abruf am 08.09.2019).
- Dahchour, Mohamed; Pirotte Alain; Zimányi, Esteban: A Generic Role Model for Dynamic Objects. In: Banks Pidduck, Anne; Mylopoulos, John; Woo, Carson C.; Ozsu M. Tamer (Hrsg.): Advanced Information Systems Engineering. Berlin, Heidelberg: Springer, 2002.
- Dengel, Andreas: Semantische Technologien. Grundlagen – Konzepte – Anwendungen. Heidelberg: Spektrum Akademischer Verlag, 2012.
- Elmasri, R.; Navathe, S. B.: Grundlagen von Datenbanksystemen. Bachelorausgabe. 3. Aufl., München: Pearson Education Limited, 2002.
- Fank, Matthias: Einführung in das Informationsmanagement. Grundlagen, Methoden, Konzepte. 2. Aufl., München; Wien: Oldenbourg Wissenschaftsverlag, 2001.
- Ferstl, Otto K.; Sinz, Elmar J.: Grundlagen der Wirtschaftsinformatik. 7. Aufl., München: Oldenbourg Verlag, 2013.
- Fowler, Martin: Dealing with Roles: 1997.
URL: <https://martinfowler.com/apsupp/roles.pdf> (letzter Abruf am 08.09.2019).
- Gadatsch, Andreas: Datenmodellierung für Einsteiger. Einführung in die Entity-Relationship-Modellierung und das Relationenmodell. Wiesbaden: Springer Vieweg, 2017.
- Gadatsch, Andreas; Landrock, Holm: Big Data für Einsteiger. Entwicklung und Umsetzung datengetriebener Geschäftsmodelle. Wiesbaden: Springer Vieweg, 2017.
- Geisler, Frank: Datenbanken. Grundlagen und Design. 3. Aufl., Heidelberg, München, Landsberg, Frechen, Hamburg: mitp-Verlag, 2009.
- Gronau, Norbert: Einführung in das Informationsmanagement. Berlin: Gito, 2010.
- Handlbauer, Gernot; Hinterhuber Hans H.; Matzler, Kurt: Kundenzufriedenheit durch Kernkompetenzen. Eigene Potenziale erkennen, entwickeln, umsetzen. Wiesbaden: Gabler Verlag, 2003.
- Hellmuth, Thomas W.: Terminologiemanagement: Aspekte einer effizienten Kommunikation in der computerunterstützten Informationsverarbeitung: 1997.
URL: <https://d-nb.info/957056079/34> (letzter Abruf am 08.09.2019).
- Herrmann, Frank: Datenorganisation und Datenbanken. Praxisorientierte Übung mit MS Access 2016. Wiesbaden: Springer Vieweg, 2018.

- Hertel, Joachim; Zentes, Joachim; Schramm-Klein, Hanna: Supply-Chain-Management und Warenwirtschaftssysteme im Handel. Berlin, Heidelberg: Springer-Verlag, 2005.
- Jarosch, Helmut: Grundkurs Datenbankentwurf. Eine beispielorientierte Einführung für Studierende und Praktiker. 4. Aufl., Wiesbaden: Springer Vieweg, 2016.
- Kudraß, Thomas: Taschenbuch Datenbanken. 2. Aufl., München: Carl Hanser Verlag GmbH & Co. KG, 2015
- Kurbel, Karl; Enterprise Resource Planning und Supply Chain Management in der Industrie. Von MRP bis Industrie 4.0. 8. Aufl., Berlin, Boston: Walter de Gruyter, 2016.
- Krcmar, Helmut: Einführung in das Informationsmanagement. 2. Aufl., Berlin, Heidelberg: 2015.
- Lemke, Claudia; Brenner, Walter; Kirchner, Kathrin: Einführung in die Wirtschaftsinformatik. Band 2: Gestalten des digitalen Zeitalters. Springer Gabler, 2017.
- Meier, Andreas; Kaufmann, Michael: SQL- & NoSQL-Datenbanken. 8. Aufl., Berlin, Heidelberg: Springer Vieweg, 2016.
- Mielke, Michael; Hildebrand, Knut; Gebauer, Marcus; Hinrichs, Holger: Daten- und Informationsqualität. Auf dem Weg zur Information Excellence. 4. Aufl., Wiesbaden: Springer Vieweg, 2018.
- Moreira, António Carrizo; Ferreira, Luís Miguel D. F.; Zimmermann, Ricardo A.: Innovation and Supply Chain Management. Relationship, Collaboration and Strategies. Cham (CH): Springer International Publishing AG, 2018.
- Nebf, Theodor; Corsten Hans (Hrsg.): Produktionswirtschaft. 7. Aufl., München: Oldenbourg Wissenschaftsverlag, 2011.
- North, K.: Wissensorientierte Unternehmensführung: Wertschöpfung durch Wissen. Wiesbaden: Gabler, 2005.
- Pfohl, Hans-Christian: Logistiksysteme. Betriebswirtschaftliche Grundlagen. Berlin, Heidelberg: Springer-Verlag, 2004.
- Poluha, Rolf G.: Quintessenz des Supply Chain Managements. Was Sie wirklich über Ihre Prozesse in Beschaffung, Fertigung, Lagerung und Logistik wissen müssen. Berlin, Heidelberg: Springer-Verlag, 2010.
- Reichenberger, Klaus: Kompendium semantische Netze. Konzepte, Technologie, Modellierung. Berlin, Heidelberg: Springer-Verlag, 2010.
- Resch, Olaf: Einführung in das IT-Management. Grundlagen, Umsetzung, Best Practice. 3. Aufl., Berlin: Erich Schmidt Verlag, 2013.
- Scheidler, Anne Antonia: Methode zur Erschließung von Wissen aus Datenmustern in Supply-Chain-Datenbanken. Göttingen: Cuvillier Verlag, 2017.

- Scheller, Angela: Informationsmodellierung für verteilte Anwendungen auf Basis standardisierter Datenmodelle. Berlin: Oldenbourg Verlag, 1993.
- Schemm, J. W.: Zwischenbetriebliches Stammdatenmanagement. Lösungen für die Datensynchronisation zwischen Handel und Konsumgüterindustrie. Berlin, Heidelberg: Springer, 2009.
- Schneider, Stephan: Konstruktion generischer Datenmodelle auf fachkonzeptioneller Ebene im betrieblichen Anwendungskontext. Methode und Studie. Aachen: Shaker-Verlag, 2007a.
- Schneider, Stephan: Ein Ansatz zur Konstruktion generischer (Unternehmens-) Datenmodelle auf fachkonzeptioneller Ebene. In: Gesellschaft für Informatik e. V. Stand: 2007b.
URL: <http://cs.emis.de/LNI/Proceedings/Proceedings90/GI-Proceedings-90-8.pdf> (letzter Abruf am 08.09.2019).
- Schuh, Günther; Stich, Volker (Hrsg.): Produktionsplanung und -steuerung 1. Grundlagen der PPS. 4. Aufl., Berlin, Heidelberg: Springer Vieweg, 2012.
- Schwarze, Jochen: Einführung in die Wirtschaftsinformatik. 5. Aufl., Herne, Berlin: Neue Wirtschafts-Briefe Verlag, 2000.
- Seufert, A.; Heinen, M.; Muth, A.: Information Rules: Die neue Anatomie der Entscheidung. In Controlling & Management Review. Wiesbaden: Springer, 2014.
- Sieberichs, Dagmar; Krüger, Hans-Joachim: Die Datenmanipulation. In: Vieweg Software-Trainer Microsoft Access für Windows. Wiesbaden: Vieweg+Teubner Verlag 1993.
- Sommer, Peggy: Umweltfokussiertes Supply Chain Management. Am Beispiel des Lebensmittelsektors. 1. Aufl., Wiesbaden: Deutscher Universitäts-Verlag, 2007.
- Stahlknecht, Peter; Hasenkamp, Ulrich: Einführung in die Wirtschaftsinformatik. 11. Aufl., Berlin, Heidelberg, New York: Springer, 2005.
- Staud, Josef L.: Datenmodellierung und Datenbankentwurf. Ein Vergleich aktueller Methoden. Berlin, Heidelberg, New York: Springer, 2005.
- Stölzle, Wolfgang; Halsband, Elias: Controlling-Lexikon. Das Supply Chain Operations Reference (SCOR)-Modell. In: Controlling. Zeitschrift für erfolgsorientierte Unternehmenssteuerung. 2005. URL: <https://www.alexandria.unisg.ch/22397/1/St%C3%B6lzle%20Halsband%202005.pdf> (letzter Abruf am 08.09.2019).
- Thelemann, Sven: Dissertation -- Semantische Anreicherung eines Datenmodells für komplexe Objekte. Kassel, Universität Kassel, 1996.
- Unland, Rainer; Pernul, Günther: Datenbanken im Einsatz. Analyse, Modellbildung und Umsetzung. Berlin: De Gruyter, 2015.

- Unterstein und Matthiessen: Anwendungsentwicklung mit Datenbanken. 5. Aufl., Berlin, Heidelberg: Springer Vieweg, 2013.
- van Randen, Hendrik Jan; Bercker, Christian; Fiendl, Julian: Einführung in UML. Analyse und Entwurf von Software. Wiesbaden: Springer Vieweg, 2016.
- Vossen, Gottfried: Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme. 5. Aufl., München: Oldenbourg Wissenschaftsverlag, 2008.
- Weber, Kristin; Klotz, Michael: Informations- und Datenmanagement, in: Bergmann, Robert; Tiemeyer, Ernst (Hrsg.): Handbuch IT-Management. Konzepte, Methoden, Lösungen und Arbeitshilfen für die Praxis. 6. Aufl., München: Carl Hanser Verlag, 2017.
- Wellbrock, Wanja: Innovative Supply-Chain-Management-Konzepte. Branchenübergreifende Bedarfsanalyse sowie Konzipierung eines Entwicklungsprozessmodells. Wiesbaden: Springer, 2015.
- Werner, Hartmut: Supply Chain Management. Grundlagen, Strategien, Instrumente und Controlling. 6. Aktualisierte Auflage, Wiesbaden: Gabler, 2017.
- West, Matthew: Developing High Quality Data Models. 1996.
URL: <http://www.matthew-west.org.uk/Documents/princ03.pdf> (letzter Abruf am 08.09.2019).
- Wittmann, W.: Unternehmung und unvollkommene Information: unternehmerische Voraussicht, Ungewissheit und Planung. Köln: Westdt. Verlag, 1959.
- Yüzgülec, G.; Witthaut, M.; Hellingrath, B.: Auswirkungen des Information Sharings in Supply Chains. In: Engelhardt-Nowitzki, C. (Hrsg.): Supply Chain Network Management. Gestaltungskonzepte und Stand der praktischen Anwendung. Wiesbaden: Gabler, 2010.

Eidesstattliche Versicherung

Name, Vorname

Matr.-Nr.

Ich versichere hiermit an Eides statt, dass ich die vorliegende Arbeit mit dem Titel:

selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Ort, Datum

Unterschrift

*Nichtzutreffendes bitte streichen

Belehrung:

Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50.000,00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden. (§ 63 Abs. 5 Hochschulgesetz - HG -)

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird gfls. elektronische Vergleichswerkzeuge (wie z.B. die Software „turnitin“) zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen:

Ort, Datum

Unterschrift