



CAD-basierte Methoden zur Unterstützung der Karosseriekonstruktion in der Konzeptphase

Dissertation

zur Erlangung der Würde eines
Doctor technicae (Dr. techn.)

vorgelegt an der

**Fakultät für Maschinenbau und Wirtschaftsingenieurwesen
der Technischen Universität Graz**

von

Alexander Harrich

Graz, im November 2014

Begutachtung:

Univ.-Doz. Dipl.-Ing. Dr.techn. Mario Hirz
Ao.Univ.-Prof. Mag.rer.nat. Dr.techn. Anton Gfrerrer

Kurzfassung

Die Anforderungen an die Entwicklung von Kraftfahrzeugen liegen derzeit in einer steigenden Variantenvielfalt, einer erhöhten Produktkomplexität und einem verbesserten Qualitätsanspruch. Gleichzeitig wird versucht, die Entwicklungskosten und vor allem die Entwicklungszeit zu reduzieren. Um diese Ziele zu erreichen, kommt der frühen Phase des Entwicklungsprozesses eine besondere Bedeutung zu. Hier werden aufbauend auf vordefinierten Produktanforderungen technische Lösungen konzipiert, die dann als Basis für eine anschließende Serienentwicklung dienen. Dabei kann maßgeblich auf die Projektkosten und -zeitpläne Einfluss genommen werden, da sich technische Änderungen –im Gegensatz zu den folgenden Projektabschnitten– rasch und kostengünstig umsetzen lassen. Das Ziel dieser Phase besteht daher in der Erstellung eines ausgereiften Fahrzeugkonzepts, dessen technische Inhalte weitgehend abgesichert und optimiert sind.

Aufgrund der Tatsache, dass Prototypen und Versuchsträger in einer frühen Entwicklungsphase in einem unzureichenden Maße vorhanden sind, kann zumeist nur auf virtuelle Werkzeuge zurückgegriffen werden. Hierbei spielen moderne CAD-Programme, mit denen neben einer reinen Geometrierzeugung durch Integration von wissensbasierten Methoden auch Auslegungs- und Absicherungsschritte durchgeführt werden können, eine entscheidende Rolle. Durch ihren geschickten Einsatz lassen sich effizient Varianten- und Konzeptgeometrien erstellen, die gleichzeitig flexibel und robust angepasst werden können sowie einen hohen technischen Reifegrad aufweisen.

Im Rahmen dieser Arbeit wird eine mögliche Verwendung von wissensbasierten CAD-Methoden in der Konzeptphase der Fahrzeugentwicklung anhand eines Vorgehensmodells ausgearbeitet. Dieses sieht eine Koppelung von funktionaler Absicherung und Auslegung mit einer teilautomatisierten Geometriemodellierung auf Baugruppen- und Bauteilebene vor. Die Wirksamkeit des Vorgehensmodells wird anhand von konkreten Anwendungsbeispielen aus der Karosserieentwicklung evaluiert. Dabei handelt es sich zum einen um die Kinematikauslegung einer Standard-Fahrzeugschürze und zum anderen um die Bauraumauslegung einer Stoßfängerstruktur. Durch die Erstellung des Vorgehensmodells sowie den Einsatz von neuartigen Methoden in der funktionalen Auslegung liefert diese Arbeit einen wesentlichen Beitrag für einen gezielten Einsatz von modernen CAD-Systemen in der Konzeptphase der Fahrzeugentwicklung.

Abstract

Today, automotive development is characterised by increased variant diversity and product complexity, as well as enhanced quality standards. At the same time, there is the challenge of reducing the development costs and even more so the development time. In order to achieve these aims the concept stage of the development process is attributed great importance. In this early phase the technical solutions are conceptualised and serve as a basis for the subsequent production development. Thereby, both the project costs and production time can be significantly optimised, since the concept phase, in contrast to subsequent development stages, offers the opportunity for a rapid and cost-effective implementation of technical modifications. Thus, the goal of the concept phase consists in the realisation of a well-engineered vehicle concept that is already optimised as far as possible.

As prototypes and test carriers are seldom available in an early development phase, virtual tools have a high importance. Among them modern CAD software plays a crucial role. Combining geometry creation with knowledge based methods enables them to support the design at dimensioning and virtual validation tasks. Furthermore, variant and concept geometries can be realised, which are at the same time flexible and robust. In addition these geometries have a high level of maturity.

In this thesis, applications of knowledge based CAD methods in the concept phase of automotive development are worked out by means of a process model. This model allows combined functional validation and design with semi-automated geometry modeling in assembly as well as in part level. The effectiveness of the process model is evaluated by explicit application examples of automotive body development. The examples range from kinematic layout validation of a standard vehicle door to the installation space design of a bumper structure. With the creation of a process model as well as the implementation of new methods in functional design, the thesis essentially caters to a precise practicable application of modern CAD systems in the concept phase of automotive development.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht. Das in „TUGRAZonline“ hochgeladene Textdokument ist mit der vorliegenden Dissertation identisch.

Graz, den 9. November 2014

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als Doktorand und wissenschaftlicher Mitarbeiter am Institut für Fahrzeugtechnik der Technischen Universität Graz im Rahmen eines Forschungsprojekts in Zusammenarbeit mit MAGNA STEYR Fahrzeugtechnik, Graz.

Mein Dank gilt Herrn Univ.-Doz. Dr.techn. Mario Hirz, Institut für Fahrzeugtechnik, der mir die Möglichkeit zur Promotion eröffnete und mich auf dem Weg dorthin begleitete. Für die fachliche Betreuung und die intensive Förderung dieser Arbeit bedanke ich mich herzlich bei ihm.

Gebührender Dank gilt Herrn Ao.Univ.-Prof. Dr.techn. Anton Gferrer, Institut für Geometrie der Technischen Universität Graz, für das Interesse an meiner Arbeit und die freundliche Bereitschaft, die Zweitbegutachtung zu übernehmen.

Des Weiteren bedanke ich mich bei den Projektpartnern der MAGNA STEYR Fahrzeugtechnik für die fortlaufende, freundschaftliche Unterstützung. Für viele wertvolle Diskussionen und fachliche Hinweise bin ich ihnen zu großem Dank verpflichtet.

Ich bedanke mich auch bei all jenen Personen, die direkt oder indirekt an der Entstehung dieser Arbeit mitgewirkt haben. Besonderer Dank gilt den Studenten, deren Ergebnisse im Rahmen von Projekt- und Diplomarbeiten wesentlich zum Gelingen dieser Dissertation beigetragen haben.

Ganz besonderer Dank gebührt meinen Kollegen am Institut für Fahrzeugtechnik für das außerordentlich gute und freundschaftliche Arbeitsklima. Neben den vielen privaten Kontakten, waren es gerade die fachlichen Diskussionen und Anregungen, die wesentlich zum Gelingen dieser Arbeit beigetragen haben.

Abschließend bedanke ich mich bei meinen Freunden und meiner Familie, die mich unentwegt tatkräftig unterstützt haben.

Graz, im November 2014

Alexander Harrich

Abkürzungen und Formelzeichen

Abkürzungen

3D	...	dreidimensional
Abb.	...	Abbildung
Anh.	...	Anhang
API	...	Application Programming Interface
bzw.	...	beziehungsweise
CAx	...	Computer Aided x (x als Platzhalter für die jeweilige Anwendung)
CAD	...	Computer Aided Design
CAM	...	Computer Aided Manufacturing
CAS	...	Computer Aided Styling
CFD	...	Computational Fluid Dynamics
CMS	...	Crash Management System
DMU	...	Digital Mock Up
DOF	...	Degrees Of Freedom
EDV	...	Elektronische Datenverarbeitung
EVA	...	Eingabe-Verarbeitung-Ausgabe
FEM	...	Finite-Elemente-Methode
Gl.	...	Gleichung
Hrsg.	...	Herausgeber
IPO	...	Input-Process-Output
JT	...	Jupiter Tessellation
Kap.	...	Kapitel
KBE	...	Knowledge Based Engineering
MKS	...	Mehrkörper-Simulation
MOKA	...	A methodology and tools oriented to knowledge-based engineering application
PDM	...	Produktdaten-Management
PLM	...	Product Lifecycle Management
PVC	...	Polyvinylchlorid
RCAR	...	Research Council for Automobile Repairs
RPT	...	Rapid Prototyping

sog.	...	sogenannt
STEP	...	Standard for the Exchange of Product Model Data
TPD	...	Technische Produktdokumentation
TU	...	Technische Universität
UDF	...	User Defined Feature
VR	...	Virtual Reality
zgl.	...	zugleich

Lateinische Formelzeichen

\mathbf{a}	...	Ortsvektor zu einem Punkt P_a
\mathbf{A}	...	Matrix, mit deren Hilfe die Optimierung von e in ein Eigenwertproblem übergeführt wird
A_{Px}	...	Auflagefläche des Prallschaums am Querträger
A_{Py}	...	Querschnittsfläche des Prallschaums am Querträger
\mathbf{b}	...	Richtungsvektor der Geraden g
b	...	Gewindesteigung
c_A	...	Begrenzungskurve von S an der A-Säule
c_B	...	Begrenzungskurve von S an der B-Säule
c_c	...	Schnittkurve der Türaußenhaut
c_c^*	...	c_c gedreht um den Winkel γ und gestreckt um den Faktor ξ
c_E	...	Erzeugende für S_S
c_H	...	Bahnkurve einer Schraubbewegung
c_K	...	Begrenzungskurve von S an der Brüstungskante
c_S	...	Trajektorie von S_S
c_T	...	Begrenzungskurve von S an der Dachkante
\mathbf{d}	...	Richtungsvektor der Schraubachse d
d	...	Schraubachse
d_B	...	Optimierte Schraubachse unter Berücksichtigung von c_B
d_S	...	Optimierte Schraubachse unter Berücksichtigung von S
D	...	Determinante
e	...	Zielfunktion: Quadratsumme aller h
E_0	...	Anfangsenergie des Fahrzeugs
E_R	...	Für Verformung des Stauchrohrs notwendige Energie
E_{Rpl}	...	Plastischer Energieanteil von E_R
E_{CMS}	...	Für die Verformung des CMS notwendige Energie
E_P	...	Für die Verformung des Prallschaums notwendige Energie
E_Q	...	Für die Verformung des Querträgers notwendige Energie

E_{Qel}, E_{Qpl}	...	Elastischer und plastischer Energieanteil von E_Q
\mathbf{F}	...	Matrix der äußeren Lasten
E	...	Elastizitätsmodul
f	...	Nebenbedingung bei der Optimierung von e
f_{CMS}	...	Korrekturfaktor um die Verformungsenergie der CMS-Komponenten zu erhalten
$F_P(x_P)$...	Kraftcharakteristik des Prallschaums
$F_{P,i}$...	Kräfte des Prallschaums, welche auf die Knotenpunkte des Querträgers $P_{Q,i}$ wirken
F_{Rpl}	...	Durchschnittskraft beim Falten eines Stauchrohrs
g	...	Bahn- bzw. Flächennormale
\mathbf{g}	...	Richtungsvektor von g
g_B	...	Geraden eines Linienkomplexes \mathcal{L} im Punkt P_B
g_{mn}	...	Gittergeraden für die punktweise Berücksichtigung der Scharniergeometrie bei der Untersuchung der Türeindrehung
G	...	Punkt auf \mathcal{G}
\mathcal{G}	...	Verschneidung von \mathcal{Q} und \mathcal{Q}^*
h_i	...	Abstand von einem G_i zu \mathcal{H}_0
h_{ijk}	...	Halbgerade, auf der alle Scharnierdrehpunkte liegen, die bei einer Drehung von $c_{c,i}$ um β zu einer Kollision mit $P_{0,ij}$ im Punkt $P_{c,ik}$ führen
\mathcal{H}_0	...	Hyperebene
i	...	Index
i_e	...	Obere Schranke von i
I_{Qz}	...	Flächenträgheitsmoment des Querträgers um die z-Achse
j	...	Index
j_e	...	Obere Schranke von j
k	...	Index
k_e	...	Obere Schranke von k
\mathbf{K}	...	Steifigkeitsmatrix
\mathcal{K}	...	Abbildung der Menge aller orientierten Geraden eines dreidimensionalen Raumes \mathbb{R}^3 und einer vierdimensionalen Mannigfaltigkeit von Punkten im \mathbb{R}^6
l_{F1}	...	Minimalabstand zwischen Scharnierdrehpunkten und Türaußenhaut
l_{F2}	...	Freigang zwischen bewegten Bauteilen
l_{F3}	...	Offset von Θ_K um Toleranzen abzubilden
l_{Hx}, l_{Hy}	...	Abstände vom Drehpunkt zum Ende des Scharnierarms in x - und y - Richtung
l_{Hz}	...	Mindestabstand zwischen den Türscharnieren
l_{R1}, l_{R2}	...	Seitenlängen eines rechteckigen Stauchrohrquerschnitts
l_{R3}	...	Dicke eines Stauchrohrquerschnitts
l_{Px}	...	Tiefe des Prallschaums

l_{Qy}, l_{Qz}	...	Länge und Höhe des Querträgers
\mathcal{L}	...	Linienkomplex
m	...	Index
m_e	...	Obere Schranke von m
m_0	...	Fahrzeugmasse
M	...	Allgemeine Schraubbewegung
\mathcal{M}^4	...	Quadratische Hyperfläche (Quadrik)
M_{Qpl}	...	Maximales Biegemoment des Querträgers
M_B	...	Soll-Absenkbewegung der Seitenscheibe ausschließlich unter Berücksichtigung von c_B
M_S	...	Soll-Absenkbewegung der Seitenscheibe ausschließlich unter Berücksichtigung von S
M_{SB}	...	Soll-Absenkbewegung der Seitenscheibe ausschließlich unter Berücksichtigung von S und von c_B
n	...	Index
n_e	...	Obere Schranke von n
\mathcal{O}	...	Koordinatenursprung
\mathbf{p}	...	Ortsvektor zu P_g
$\dot{\mathbf{p}}$...	Die Ableitung von \mathbf{p} entspricht dem Tangentenvektor von c_H
P_{mn}	...	Scharnierdrehpunkte auf g_{mn} , die zu einer Kollision von Scharnier und offener Türe führen
P_0	...	Punkt auf der Störgeometrie
P_a	...	Ein Punkt auf der Schraubachse d
P_B	...	Punkt auf c_B
P_c	...	Endpunkt von \mathbf{r} , der gleichzeitig auf c_c liegt
P_c^*	...	Endpunkt von \mathbf{r}^* , der gleichzeitig auf c_c^* liegt
P_g	...	Punkt auf g
P'_g	...	Fernpunkt von g
P_x, P_y	...	Punkte auf g
P_P	...	Diskretisierungspunkt auf dem Prallschaum
P_Q	...	Diskretisierungspunkt auf dem Querträger
P_W	...	Diskretisierungspunkt auf der Barriere
\mathbb{P}^n	...	n -dimensionaler projektiver Raum
q	...	Exponent für die Beschreibung eines dehnungsabhängigen Werkstoffverhaltens
\mathcal{Q}	...	<i>Plücker</i> -Bedingung
\mathcal{Q}^*	...	Bedingung für <i>normalisierte Plückervektoren</i>
\mathbf{r}	...	Ortsvektor zu c_c
\mathbf{r}^*	...	Drehstreckung von \mathbf{r} um den Winkel γ und mit dem Faktor ξ

S	...	Scheibenfläche
S_{lim}	...	Grenzfläche, welche die P_{mn} miteinander verbindet
S	...	Scheibenfläche
S_S	...	Schraubfläche einer Scheibenfläche entsprechend der Schraubbewegung M_S
S_{SB}	...	Schraubfläche einer Scheibenfläche entsprechend der Schraubbewegung M_{SB}
S_D	...	Design-Scheibenfläche
t	...	Zeit
$T_{\mathbf{o}\tau}$...	Transformation von Σ_τ in Σ_0
\mathbf{u}	...	Verformungsmatrix
\mathbf{v}	...	Geschwindigkeitsvektor
v_0	...	Anfangsgeschwindigkeit
\mathbf{w}	...	Vektor der Winkelgeschwindigkeit
\mathbf{x}	...	Koordinatenrichtung
x_{Rpl}	...	Relativverschiebung des Stauchrohrs
$x_{P,i}$...	Relativverschiebungen des Prallschaums zwischen den $P_{P,i}$ und $P_{Q,i}$
\mathbf{x}_0	...	Einheitsvektor von Σ_0
\mathbf{x}_τ	...	Einheitsvektor von Σ_τ
\mathbf{y}	...	Koordinatenrichtung
\mathbf{y}_0	...	Einheitsvektor von Σ_0
\mathbf{y}_τ	...	Einheitsvektor von Σ_τ
z	...	Koordinatenrichtung

Griechische Formelzeichen

α_x	...	Neigungswinkel der Scharnierachse um die Fahrzeug-x-Achse
α_y	...	Neigungswinkel der Scharnierachse um die Fahrzeug-y-Achse
β	...	Türöffnungswinkel
γ	...	Winkel der Drehstreckung
Γ	...	Menge aller Geraden des dreidimensionalen \mathbb{P}^3
Δ	...	Differenz
ε	...	Dehnung
ε_0	...	Dehnung von 0,2% = Streckgrenze
ζ	...	Biegewinkel
ζ_{pl}	...	Zusätzlicher Biegewinkel bei einem plastischen Gelenk
η	...	Parameter für die Parameterdarstellung von \mathbf{h}
Θ	...	Lösungsraum für die Drehpunkte der Türscharniere
Θ_o, Θ_u	...	Θ besteht aus einem oberen und einem unteren Lösungsraum Θ_o und Θ_u

Θ_G	...	Lösungsraum für die Scharnierdrehpunkte, sodass es in Konstruktionslage zu keiner Überschneidung der Scharniermodelle mit umliegenden Bauteilmodellen kommt
Θ_{Go}, Θ_{Gu}	...	Θ_G besteht aus einem oberen und einem unteren Lösungsraum Θ_{Go} und Θ_{Gu}
Θ_{GK}	...	Schnittmenge von Θ_G und Θ_K
$\Theta_{GKo},$ Θ_{GKu}	...	Θ_{GK} besteht aus einem oberen und einem unteren Lösungsraum Θ_{GKo} und Θ_{GKu}
Θ_K	...	Lösungsraum für die Drehpunkte der Türscharniere, unter Berücksichtigung der Kinematik, jedoch nicht der Scharniergeometrie
$\Theta_{K,i}, \Theta_{K,i}^*$...	Lösungsraum Θ_K der i -ten Schnittebene bzw. der dazugehörige Komplementärbereich
λ	...	Lagrange-Multiplikator
λ^*	...	Kleinster Eigenwert von \mathbf{A}
μ	...	Parameter für die Parameterdarstellung von \mathbf{r}
μ_0, μ_e	...	Untere und obere Schranke von μ
ξ	...	Streckfaktor der Drehstreckung
ρ	...	Beliebiger Faktor
$\sigma(\varepsilon)$...	Spannungs-Dehnungscharakteristik
σ_y	...	Spannung bei der Streckgrenze
Σ_0	...	Inertiales, kartesisches Koordinatensystem
Σ_τ	...	Bewegtes, lokales Koordinatensystem
φ	...	Winkel zwischen Σ_0 und Σ_τ

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Ausgangssituation	3
1.3	Zielsetzung	5
1.4	Gliederung der Arbeit	6
2	Produktentstehung in der Fahrzeugtechnik	8
2.1	Einleitung	8
2.2	Entwicklungsprozess in der Fahrzeugtechnik	8
2.2.1	Prozessmodell in der Produktentstehung	8
2.2.2	Definitionsphase	9
2.2.3	Konzeptphase	10
2.2.4	Vorentwicklungsphase	11
2.2.5	Serienentwicklungsphase	12
2.2.6	Serienproduktionsphase	13
2.3	Trends in der Produktentstehung	13
2.4	Bedeutung der frühen Entwicklungsphase	16
2.5	Zusammenfassung	18
3	Wissensbasierte CAD-Methoden	19
3.1	Einleitung	19
3.2	Wissensbasierte Konstruktion	19
3.3	CAD-Systeme und deren Entwicklung	23
3.3.1	Entwicklungsstufen von CAD-Systemen	23
3.3.2	Konventionelles CAD	25
3.3.3	Parametrisch-assoziatives CAD	25
3.3.4	Featurebasiertes CAD	28
3.3.5	Wissensbasiertes CAD	30
3.3.6	Trends	32
3.4	CAD-Systeme in einem PDM-unterstützten Entwicklungsprozess	34
3.5	Templates als Werkzeuge einer wissensbasierten Produktmodellierung	37
3.5.1	Überblick und Eingrenzung	37
3.5.2	Strukturtemplates	39
3.5.3	Geometrietemplates: Geometriedummies	45
3.5.4	Geometrietemplates: Matingtemplates	46
3.5.5	Funktionstemplates	56

3.5.6	Ansätze für eine Template-unterstützte Entwicklung	59
3.6	Zusammenfassung	60
4	Vorgehensmodell einer CAD-basierten Vorauslegung	62
4.1	Einleitung	62
4.2	Templates in der frühen Entwicklungsphase	62
4.2.1	Templates in der Definitionsphase – Entwurfssysteme für Gesamtfahrzeugkonzepte	62
4.2.2	Herausforderung beim Übergang von der Definitions- in die Konzeptphase	64
4.2.3	Templates in der Konzeptphase	67
4.3	CAD-basierte Vorauslegung mittels Templates	69
4.3.1	Bedarf einer CAD-basierten Vorauslegung in der Konzeptphase	69
4.3.2	Vorgehensmodell für eine CAD-basierte Vorauslegung	71
4.3.3	Vorteile einer CAD-basierten Vorauslegung im Entwicklungsprozess	74
4.3.4	Randbedingungen und Einschränkungen einer CAD-basierten Vorauslegung	77
4.4	Prototypische Umsetzung einer CAD-basierten Vorauslegung	80
4.5	Zusammenfassung	80
5	Kinematikauslegung einer Fahrzeugtüre	82
5.1	Einleitung	82
5.2	Türenentwicklung in der Konzeptphase	82
5.3	Anwendung des Vorgehensmodells	84
5.4	Absicherung einer kollisionsfreien Türöffnung	86
5.4.1	Anforderungen	86
5.4.2	Lösungsweg in drei Stufen	88
5.4.3	Stufe eins: Kollisionsprüfung in der Konstruktionslage	89
5.4.4	Stufe zwei: Lösung des Kinematik-Syntheseproblems	91
5.4.5	Stufe drei: Kollisionsprüfung der Scharniere	98
5.4.6	Bewertung des Lösungswegs	100
5.5	Scheibenabsenkung	102
5.5.1	Anforderungen	102
5.5.2	Lösungsweg in zwei Stufen	105
5.5.3	Approximation durch Schraubflächen	107
5.5.4	Stufe eins: Approximation einer Schraubfläche als Scheibenfläche	109
5.5.5	Stufe zwei: Berechnung einer Solltrajektorie	111
5.5.6	Bewertung des Lösungswegs	113
5.6	Geometriemodellierung eines Türinnenblechs	114
5.7	Zusammenfassung	118
6	Bauraumauslegung einer Stoßfängerstruktur	119
6.1	Einleitung	119
6.2	Entwicklung von Crash Management Systemen in der Konzeptphase	119
6.3	Anwendung des Vorgehensmodells	120

6.4	Dimensionierung eines CMS	122
6.4.1	Anforderungen	122
6.4.2	Umgesetzter Lösungsansatz	125
6.4.3	Beschreibung des Berechnungsmodells	126
6.4.4	Bewertung des Lösungswegs	133
6.5	Geometriemodellierung eines CMS	135
6.6	Zusammenfassung	137
7	Zusammenfassung	138
A	Anhang	141
A.1	Plückerkoordinaten	141
A.2	Lineares Balkenelement	144
	Literaturverzeichnis	147

1 Einleitung

1.1 Motivation

Die Entwicklung von Produkten hat sich mit dem fortschreitenden Einsatz von rechnerunterstützten Methoden wesentlich gewandelt und erweitert. So hat die Verfügbarkeit von gesteigerter Rechnerleistung, Speicherplatz sowie schnellem Datentransfer dazu geführt, dass ein großer Teil der Produktentwicklung mit Hilfe von virtuellen Werkzeugen und Methoden durchgeführt werden kann. Die kontinuierliche Verbesserung von EDV¹-Komponenten lässt sich anschaulich anhand des Moore'schen Gesetzes darstellen, Abb. 1.1. Dieses besagt, dass die Anzahl an Transistoren auf einem Mikrocontroller im Laufe der Zeit exponentiell ansteigt [Mol06]. Wenngleich dieses Gesetz nicht unmittelbar eine Aussage über die Rechenleistung zulässt, so veranschaulicht es dennoch den stetigen Fortschritt im Bereich der Hardware-Entwicklung. Trotz der Endlichkeit dieser Gesetzmäßigkeit und ungeachtet der Herausforderungen durch weitere Miniaturisierungen von Mikroprozessoren, wird eine Beibehaltung dieses Trends für die nächsten Dekaden prognostiziert [II10].

Parallel zur Leistungssteigerung der Hardware haben sich auch die Fähigkeiten der jeweiligen Anwendungen umfassend erweitert. Dies hat dazu geführt, dass Tätigkeiten, die vormals ohne Unterstützung von Computern ausgeübt wurden, durch rechnerunterstützte Methoden (CAx²) ersetzt oder ergänzt wurden. Im Bereich der Konstruktion vollzog sich der Umstieg auf eine rechnerunterstützte Konstruktion (CAD³) in der Industrie vor allem in den 80er- und 90er-Jahren des 20. Jahrhunderts [SK97]. Die Geometrirepräsentation beschränkte sich zu Beginn hauptsächlich auf Drahtgitter- und Flächenelemente. Neben der Aufgabe der Zeichnungserstellung war besonders die Darstellung von Freiformflächen für die Luftfahrt- bzw. Automobilindustrie von großem Interesse [Far02]. Aufbauend auf die 3D⁴-Flächenmodellierung lieferte in weiterer Folge die Modellierung von Volumenkörpern eine eindeutige und umfassende Beschreibung von Körpern. Zusätzlich wurde durch die Anordnung von Bauteilen in Baugruppen und Produktmodellen die Möglichkeit geschaffen, komplexe Produktstrukturen zu generieren, Abb. 1.1.

Moderne CAD-Programme verfügen durch die Integration von Programmier-, Simulations- und Optimierungswerkzeugen über die Fähigkeit, neben einer reinen Geometrieerzeugung, auch

¹Elektronische Daten Verarbeitung

²Computer Aided x (x gilt als Platzhalter für die jeweilige Anwendung)

³Computer Aided Design

⁴dreidimensionale

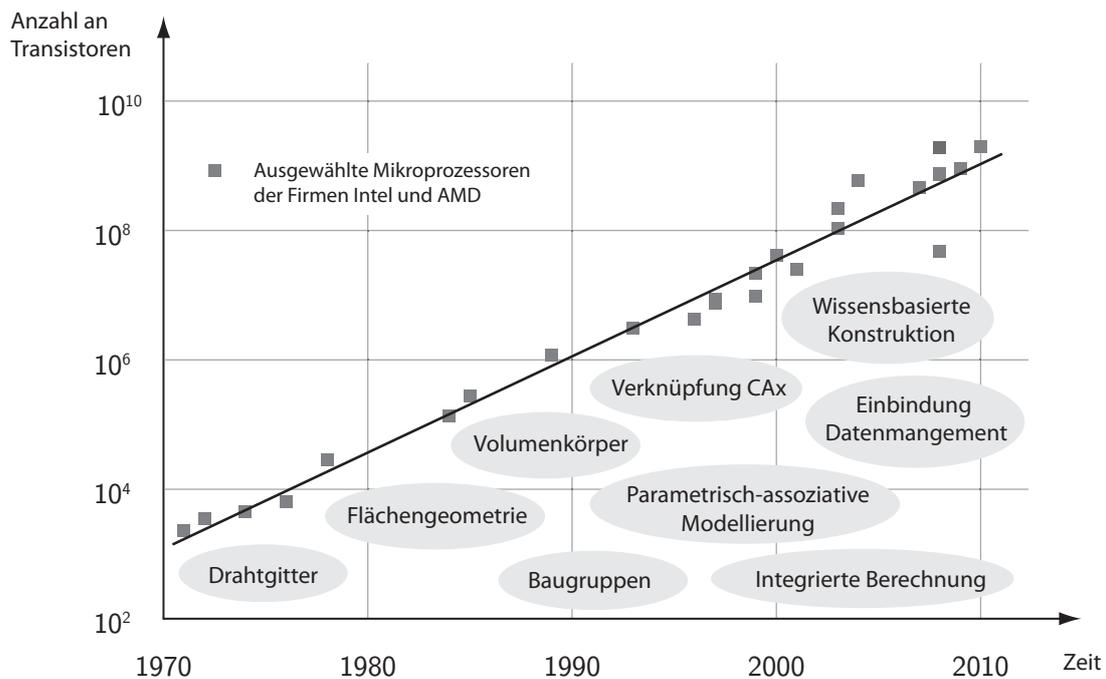


Abbildung 1.1: Entwicklungsschritte von CAD-Programmen vor dem Hintergrund der fortschreitenden Leistungssteigerung von EDV-Komponenten, zum Teil aus [Mey07, Ste07, Hir11].

Auslegungs- und Berechnungsaufgaben zu übernehmen. Außerdem ermöglicht der parametrisch-assoziative Strukturaufbau ein rasches Anpassen der resultierenden Geometrie an neue Eingangsdaten. Dabei werden Geometrielemente durch eine Parameterstruktur beschrieben und auf Basis von *Eltern-Kind*-Beziehungen mit anderen Geometrielementen verknüpft. Mit der Möglichkeit Formelemente (*Features*), Vorlagen (*Templates*) sowie Makro-Routinen zu erstellen und diese in Bibliotheken abzulegen, kann erworbenes Wissen konserviert und bei Bedarf effizient bereitgestellt werden. Außerdem lässt sich durch eine Koppelung mit Datenbank- und Datenmanagementsystemen eine durchgängige Produktstruktur erzielen. Aufgrund dieser Merkmale haben sich CAD-Systeme als unverzichtbarer Bestandteil einer virtuellen Produktentwicklung etabliert [ES09].

Eine bedeutende Herausforderung beim industriellen Einsatz von CAD-Programmen liegt allerdings in der optimalen Nutzung der bereitgestellten Funktionalitäten. So erschwert die Komplexität der eingesetzten Programme die Nutzung der implementierten Funktionen durch den Anwender. Obwohl es zu ständigen Verbesserungen und Anpassungen der Konstruktionsmethoden und -prozesse kommt, werden die vorhandenen Möglichkeiten zumeist nur unzureichend ausgeschöpft. Dieser Umstand ist der Ausgangspunkt von etlichen Ansätzen, die das Ziel verfolgen, den Einsatz von CAD-Systemen effizienter zu gestalten. Bei der Entwicklung und der Umsetzung der resultierenden Methoden nimmt unter anderem die Automobilindustrie eine Vorreiterrolle ein [VWBZ09]. Auch diese Arbeit legt ihren Fokus auf die Fahrzeugentwicklung und berücksichtigt deren Anforderungen.

Eine Vielzahl der ausgearbeiteten Methoden wird dabei in einer frühen Phase der Fahrzeugentwicklung eingesetzt. Diese ist geprägt von unvollständigen Datenständen, sich ständig ändernden Randbedingungen, unterschiedlichen Varianten und einem dynamischen Umfeld. Unter diesen Bedingungen haben moderne CAD-basierte Methoden im Vergleich zu konventionellen Konstruktionsmethoden wesentlich mehr Vorteile, als in einer relativ starren Serienentwicklung, bei der Faktoren wie Lieferantenintegration, Produktionsplanung oder Änderungsmanagement in den Vordergrund treten und es üblicherweise nur mehr zu Detailanpassungen der Produktdaten im CAD-System kommt. Darüber hinaus bestimmt das Resultat einer konzeptionellen Auslegung maßgeblich den gesamten Entwicklungsaufwand, sodass ein effizienter Einsatz von CAD-basierten Methoden in der Konzeptphase eine große Bedeutung für den folgenden Entwicklungsprozess hat. Aus diesen Gründen bietet es sich an, das Schwergewicht der CAD-Methodenentwicklung auf eine frühe Entwicklungsphase zu legen.

1.2 Ausgangssituation

Effiziente, rechnerunterstützte Konstruktionsmethoden zur Auslegung von Fahrzeugkonzepten in einer frühen Entwicklungsphase sind weit vorangeschritten und weisen einen hohen Reifegrad auf. So unterstützen gerade CAD-basierte Entwurfssysteme die Gesamtfahrzeugentwicklung hinsichtlich Bauraumauslegung (*Packaging*), Ergonomiebewertung, Homologation und dienen als Datenlieferant für nachgeschaltete Berechnungen und Detailkonstruktionen. Entsprechende Ansätze mit unterschiedlichen Schwerpunkten wurden beispielsweise von *Niemierski*, *Hänschke*, *Deter* oder *Rosbacher* vorgestellt [Nie88, Hän90, DO95, RHH⁺09]. Die Entwurfssysteme eignen sich für die Erstellung von Fahrzeugkonzepten, die neben einer vereinfachten Geometrie auch globale Entwurfsparameter sowie Produkt- und Prozessinformationen beinhalten. Auf Basis dieser Konzepte können die jeweiligen Subsysteme ausgelegt und detailliert werden, bis schlussendlich ein ausgereiftes Gesamtfahrzeugkonzept für eine folgende Serienkonstruktion zur Verfügung steht. Insbesondere bei der Erstellung und Detaillierung der Konzeptgeometrie kann auf weitreichende Methoden zurückgegriffen werden.

So beschreibt etwa *Mendgen* die Notwendigkeit einer systematischen Vorgehensweise bei der Geometriemodellierung mit parametrisch-assoziativen CAD-Systemen und erarbeitet Grundprinzipien für eine methodische Geometrieerzeugung. Die gewonnenen Erkenntnisse werden unter anderem im Rahmen eines Assistenzsystems zur Modellanalyse umgesetzt [Men99]. *Forsen* verallgemeinert die Ergebnisse und erweitert sie um systemtechnische Ansätze. Auf Basis dieser Betrachtungsweise wird ein Vorgehensmodell erstellt, das als Leitfaden für eine parametrisch-assoziative Konstruktion dient [For03]. Anwendungsorientierte Strategien und Methoden zur Flächenmodellierung werden unter anderem von *Braß*, *Haslauer* und *Brill* zusammengefasst. Alle drei Arbeiten richten sich in erster Linie an CAD-Anwender in der Karosseriekonstruktion. *Braß* geht dabei detailliert auf einzelne Funktionen eines kommerziell verfügbaren CAD-Programms ein [Bra09]. *Haslauer* legt den Fokus auf die Konzeptauslegung und beschreibt ausführlich die Verwendung von *Adaptermodellen* [Has05]. *Brill* gibt Gestaltungsregeln sowohl für die Bauteilkonstruktion als auch für die Konstruktion der entsprechenden Produktionswerkzeuge [Bri06].

Eine Vielzahl von Ansätzen behandelt Bauteil- und Baugruppenmodelle, die umfangreich feature- und wissensbasierte *Templates* zur Geometrieerstellung einsetzen. Dadurch wird eine wiederholte Verwendung von bereits erzeugter Geometrie ermöglicht. Darüber hinaus können Vorlagen neben der Geometrie auch Produkt- und Prozessinformationen beinhalten. *Prieur* beschreibt neben *Features* und *Templates* vor allem die Verwendung von *Skelettmodellen* und erstellt ein generisches Baugruppenmodell für Tiefziehwerkzeuge [Pri06]. *Bossmann* stellt ein Konzept für ein integriertes Produkt- und Prozessmodell auf und ermöglicht damit ein frühzeitiges Einbinden der Produktionsplanung [Bos07]. *Mbang Sama* erstellt einen ganzheitlichen Ansatz zur Modellierung von Karosseriebauteilen im Kontext einer virtuellen Produktentwicklung und setzt dafür umfassend *Features* und *Templates* ein [Mba08]. Die Vorteile einer vorgegebenen Bauteilstruktur im Rahmen eines erweiterten *Startmodells* werden durch *Brockmeyer* ausgeführt [Bro10].

Neben der Geometriemodellierung besteht ein zentrales Anwendungsgebiet von CAD-Programmen in der geometrischen Integration, die unter anderem Bauraumuntersuchungen und Ergonomieauslegungen beinhalten. Dabei werden die Auslegungen mit Hilfe eines zentralen Produktmodells (DMU⁵) durchgeführt, das vor allem die Produktstruktur und die Geometrieprepräsentation eines Gesamtfahrzeugs enthält. Eine Vorgehensweise für einen entwicklungsbegleitenden DMU-Prozess, bei dem es zu einer Koppelung zwischen DMU, Bauteilkonstruktion und Berechnung kommt, wird etwa von *Markworth* erarbeitet [Mar03].

Zusätzlich zur Geometriemodellierung und der geometrischen Integration bieten CAD-Systeme auch die Möglichkeit für eine funktionale Auslegung. Dies kann einerseits durch die Nutzung der bereitgestellten Programmier- und Berechnungsfunktionen geschehen oder andererseits durch das Einbinden von Berechnungsmodellen. Darüber hinaus lassen sich CAD-Programme mit externen Berechnungsprogrammen koppeln, wenngleich dies oftmals einen Bruch in der Datenstruktur nach sich zieht. Im Bereich der Entwurfssysteme für Gesamtfahrzeuge stehen etliche Erweiterungen für eine funktionale Auslegung zur Verfügung, wie beispielsweise von *Rasenack* und *Sander*. Beide verwenden Optimierungsalgorithmen; zum einen um globale Fahrzeugparameter auszulegen [San07] und zum anderen um eine Bauraumauslegung auf Komponentenebene durchzuführen [Ras98]. Auf Bauteil- und Baugruppenebene gibt es ebenfalls Beispiele für eine funktionale Auslegung und Absicherung, in denen vornehmlich *Templates* verwendet werden [Fra08, Fri08]. Diese Anwendungen stellen jedoch die funktionalen Aspekte häufig isoliert dar und verbinden diese nicht mit einer wissensbasierten Geometriemodellierung.

Eine Verknüpfung von funktionaler Auslegung und Absicherung mit einer teilautomatisierten Geometrieerzeugung innerhalb eines CAD-Systems bringt allerdings einige Vorteile mit sich. So entspricht die resultierende Konzeptgeometrie bereits frühzeitig funktionalen Anforderungen, ohne dass auf Ergebnisse von externen Berechnungswerkzeugen zurückgegriffen werden muss. Gleichzeitig kann aufgrund des parametrisch-assoziativen Strukturaufbaus von modernen CAD-Systemen die Konzeptgeometrie rasch an Änderungen der Eingangsdaten angepasst werden. Damit stehen schon in einer frühen Entwicklungsphase ausgereifte und qualitativ hochwertige Geometriedaten zur Verfügung, mit denen der Aufwand für folgende, detailliertere

⁵Digital Mock-Up

Absicherungs- und Optimierungsverfahren reduziert werden kann. Des Weiteren erleichtert die einheitliche Datenstruktur einen durchgängigen Entwicklungsprozess.

Ein Ansatz für eine Koppelung von Geometrierzeugung und funktionaler Absicherung mit Hilfe von *Templates* wird beispielsweise von *Haasis* aufgezeigt [HAR06, HVA08]. Diese Beiträge geben einen überblicksartigen Status von industriellen Anwendungen bei einem Fahrzeughersteller und sprechen Herausforderungen sowie Anforderungen für die Umsetzung einer *Template*-unterstützten Entwicklung an. Ein weiterer Ansatz, der in der Industrie zum Einsatz kommt, stammt von *Gülich*. Allerdings behandelt er die Kombination aus einer Geometrierzeugung und einer funktionalen Absicherung nur am Rande und setzt den Fokus auf die Bereitstellung von Geometriedaten für eine folgende Berechnung [GNR⁺08]. *Berkan* beschreibt ebenfalls Möglichkeiten für eine *Template*-basierte Integration von Geometriemodellierung und Funktionsauslegung. Auch dieser Beitrag hat seinen Ursprung in einem industriellen Projekt und liefert umgesetzte Beispiele einer konzeptionellen Fahrwerksentwicklung [BABS10].

Trotz dieser vorhandenen Ansätze wird die Kombination aus einer geometrischen und einer funktionalen Auslegung innerhalb der CAD-Umgebung oftmals nur im Überblick und aus dem Blickwinkel spezifischer Anwendungen behandelt. Es fehlen daher übergeordnete Methoden, die abseits der angeführten Beispiele für eine erfolgreiche Umsetzung herangezogen werden können. Außerdem wird in der Regel auf eine detaillierte Modellbeschreibung der Funktionsauslegung verzichtet, sodass diese kaum nachvollzogen und überprüft werden kann. Es besteht daher sowohl der Bedarf an einer allgemeineren Betrachtungsweise der Kombination aus Geometrie- und Funktionsauslegung in der frühen Entwicklungsphase eines Fahrzeugs als auch an konkreten Modellbeschreibungen für eine nachvollziehbare Umsetzung.

1.3 Zielsetzung

Das Ziel der vorliegenden Arbeit besteht darin, die Verbindung einer CAD-basierten funktionalen Absicherung mit einer teilautomatisierten Geometrierzeugung unter Einsatz von *Templates* zu beleuchten. Diese Anwendung von wissensbasierter Konstruktion eignet sich insbesondere für den Einsatz in der frühen, konzeptionellen Entwicklungsphase, da hier die Anforderungen an Auslegungswerkzeuge mit einem hohen Grad an Flexibilität bei gleichzeitiger Robustheit hinsichtlich der sich ständig ändernden Randbedingungen am größten sind. Daher soll für die Anwendung in dieser frühen Phase ein Vorgehensmodell erstellt werden, mit dem Ziel die Modellierung von Bauteil- und Baugruppen zu erleichtern und damit den Einsatz einer computerunterstützten Konstruktion effizienter zu gestalten.

Neben der Erstellung des Vorgehensmodells liegt ein wesentliches Ziel dieser Arbeit in der Anwendung des ausgearbeiteten Ansatzes. So soll die Wirksamkeit des Vorgehensmodells anhand von zwei konkreten Beispielen aus der Karosseriekonstruktion dargestellt werden. Bei diesen Anwendungsbeispielen handelt es sich zum einen um die Modellierung eines Türinnenblechs unter Berücksichtigung der Türkinematik und zum anderen um die Vorauslegung und Konstruktion einer Stoßfängerstruktur.

Die Herausforderung bei der Kinematikauslegung einer Standard-Fahrzeuggestüre liegt in der Absicherung des Fahrzeugdesigns. So beeinflusst etwa die Anforderung eines kollisionsfreien Türöffnens die Fahrzeugaußenhaut und den Verlauf sowie die Breite der Türfuge. Ebenfalls gilt es, die Absenkbarkeit der Seitenscheibe abzusichern, die einerseits durch das Design und andererseits durch Vorgaben der Scheibendichtungen beeinflusst wird. Die Ergebnisse der Kinematikauslegung sollen dann als Eingangsdaten für die Erstellung eines Türinnenblechs herangezogen werden.

Das zweite Anwendungsbeispiel –die Vorauslegung einer Stoßfängerstruktur– ist in einer konzeptionellen Entwicklungsphase wesentlich bestimmt von Packaging-Anforderungen. So muss frühzeitig sichergestellt werden, dass genügend Bauraum für Deformationselemente, wie etwa Crashboxen oder Prallschäume, vorgehalten wird. Zusätzlich hat die Verformung des Crashsystems aufgrund von standardisierten Crashtests Einfluss auf die geometrische Integration von umliegenden Bauteilen, wie beispielsweise Kühlerpakete. Durch den Einsatz einer funktionalen Auslegung sollen der notwendige Bauraum für eine Stoßfängerstruktur ermittelt und die einzelnen Bauteile dimensioniert werden.

Wie im Vorgehensmodell vorgeschlagen, soll die Absicherung und Auslegung in beiden Beispielen mit Hilfe von wissensbasierten CAD-Methoden unter Einsatz von innovativen Lösungsansätzen erfolgen. Ein charakteristisches Merkmal dieser Arbeit soll darin bestehen, dass neben der Evaluierung des Vorgehensmodells detailliert auf die Modellbildung der funktionalen Absicherungen eingegangen wird. Auf Basis der erzielten Ergebnisse sollen in weiterer Folge Möglichkeiten für eine teilautomatisierte Geometriemodellierung umgesetzt werden.

1.4 Gliederung der Arbeit

Die Arbeit unterteilt sich in insgesamt acht Kapitel, die in drei aufbauenden Teilen angeordnet sind, Abb. 1.2. Demnach werden im ersten Teil der Stand der Technik erörtert und grundlegende Rahmenbedingungen definiert. Dieser Teil umfasst insgesamt zwei Kapitel, nämlich eine Betrachtung des Entwicklungsprozesses in einer typischen Fahrzeugentwicklung und eine Darstellung von wissensbasierten CAD-Methoden. Es wird eingangs auf die Eigenschaften und Anforderungen einer Fahrzeugentwicklung eingegangen und die Rolle einer frühen, konzeptionellen Entwicklungsphase herausgearbeitet. Des Weiteren werden die Merkmale von modernen wissensbasierten CAD-Systemen dargestellt und Möglichkeiten für einen effizienten Einsatz aufgezeigt. Die beschriebenen Funktionalitäten kommen dann in den folgenden Kapiteln zur Anwendung.

Aus den Anforderungen des Entwicklungsprozesses und den Möglichkeiten von modernen CAD-Systemen leiten sich dann die Merkmale des vorgestellten Vorgehensmodells ab. Dieses Modell sieht eine Kombination aus einer funktionalen Absicherung und einer teilautomatisierten Geometriemodellierung mit wissensbasierten CAD-Methoden vor. Die Beschreibung des Modells erfolgt im zweiten Teil dieser Arbeit und umfasst ein Kapitel.

Einleitung		
Produktentstehung in der Fahrzeugentwicklung	TEIL I	Stand der Technik
Wissensbasierte CAD-Methoden		
Vorgehensmodell einer CAD-basierten Vorauslegung	TEIL II	Konzeption des Vorgehensmodells
Kinematikauslegung einer Fahrzeugtüre	TEIL III	Anwendung und prototypische Umsetzung
Bauraumauslegung einer Stoßfängerstruktur		
Zusammenfassung		

Abbildung 1.2: Gliederung der Arbeit.

Der dritte Teil befasst sich mit der Anwendung und Evaluierung des dargestellten Ansatzes. In zwei Kapiteln werden die Anwendungsbeispiele – die Kinematikauslegung einer Standard-Fahrzeugtüre und die Vorauslegung einer Stoßfängerstruktur – detailliert beschrieben und neben einer umfangreichen Darstellung der funktionalen Modellbildung wird auf die Verfahren für eine wissensbasierte Geometrieerzeugung eingegangen.

Die drei Teile werden ergänzt durch eine Einleitung und eine Zusammenfassung. Hier werden zum einen die Motivation sowie die Ausgangslage dieser Arbeit beschrieben und davon abgeleitet die Forschungsziele definiert. Zum anderen werden die Ergebnisse zusammengefasst, die zentralen Erkenntnisse hervorgehoben und deren Neuheitsgrad herausgearbeitet. Des Weiteren enthält diese Arbeit einen Anhang, der relevante Herleitungen und theoretische Grundlagen beinhaltet. Damit soll ein besseres Verständnis für die eingesetzten Methoden geschaffen werden.

2 Produktentstehung in der Fahrzeugtechnik

2.1 Einleitung

Der Schwerpunkt dieser Arbeit liegt in der Anwendung von wissensbasierten CAD-Methoden in einer frühen Phase der Fahrzeugentwicklung. Diese Phase hat einen wesentlichen Einfluss sowohl auf das resultierende Produkt als auch auf den folgenden Entwicklungsaufwand. Es wird daher versucht, in diesem Projektabschnitt ein möglichst ausgereiftes Fahrzeugkonzept zu erzeugen, das dann als Basis für eine anschließende seriennahe Entwicklung dient und idealerweise für geringe Folgekosten, eine kurze Entwicklungszeit sowie eine hohe Produktqualität sorgt. Eine Maßnahme, um dieses Ziel zu erreichen, liegt im geschickten Einsatz von virtuellen Entwicklungsmethoden. In diesem Kapitel werden übergeordnet die Abläufe einer typischen Fahrzeugentwicklung herausgearbeitet, auf Trends in der Produktentwicklung eingegangen und die Bedeutung der frühen Entwicklungsphase aufgezeigt.

2.2 Entwicklungsprozess in der Fahrzeugtechnik

2.2.1 Prozessmodell in der Produktentstehung

Eine typische Neuentwicklung eines Fahrzeuges lässt sich mittels Meilensteinen in Projektabschnitte unterteilen. Obwohl die Benennungen und Umfänge der Abschnitte bei den diversen Fahrzeugherstellern wie auch in der Literatur vielfach variieren, so ähneln sie sich dennoch in ihrer Struktur und in ihrem Ablauf. Es lässt sich daher ein charakteristischer Prozess definieren, der unabhängig vom jeweiligen Hersteller den Entwicklungsablauf eines neuen Fahrzeuges beschreibt. In Abb. 2.1 wird ein solcher Produktentstehungsprozess beispielhaft skizziert. Demnach startet die eigentliche Produktentwicklung nach Abschluss der Produktforschung und wird anschließend von der Produktherstellung abgelöst [SK97]. Diese Gliederung lässt sich weiter in eine Definitionsphase, eine Konzeptphase, eine Vorentwicklungsphase, eine Serienentwicklungsphase und eine Serienproduktionsphase unterteilen, wobei sich die Unterteilung der Phasen und die dazugehörigen Meilensteine an *Böhme*, *Wiermeier*, *Brockmeyer* und *Hirz* orientieren [Mar03, Böh04, Wie05, Bro10, Hir11].

Neben der Unterteilung der einzelnen Entwicklungsphasen werden in Abb. 2.1 ebenfalls jene Zeitabschnitte angeführt, in denen markante Entwicklungsaufgaben erfolgen. Eine klare Abgrenzung dieser Zeiträume gestaltet sich jedoch als problematisch, da in der industriellen Praxis weder der Start noch das Ende dieser Abschnitte abrupt erfolgen. Des Weiteren führt das Bestreben, die unterschiedlichen Fachbereiche möglichst frühzeitig in den Entwicklungsprozess

einzubinden, zu einer zunehmenden Parallelisierung der jeweiligen Entwicklungsaufgaben. Die Gliederung und Anordnung der Entwicklungsaufgaben erfolgt in Anlehnung an *Hirz*, der mit einer Untersuchung von Entwicklungsprozessen von diversen Fahrzeugherstellern einen charakteristischen Ablauf definiert [Hir11].

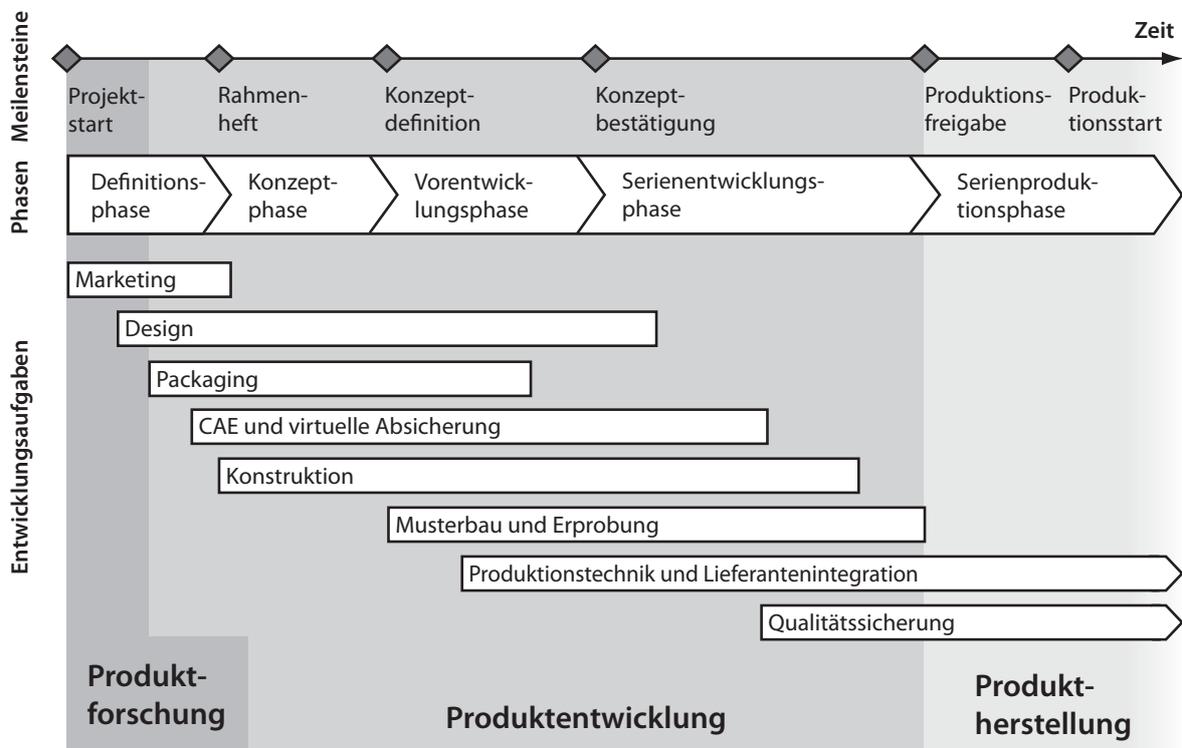


Abbildung 2.1: Exemplarischer Produktentstehungsprozess bei einer Fahrzeugneuentwicklung.

Der hier beschriebene Produktentstehungsprozess ist stellvertretend für eine Vielzahl an Prozessketten und -phasenmodellen, mit denen versucht wird, eine typische Fahrzeugentwicklung darzustellen. Da es jedoch durchaus zu Abweichungen und Unterschieden in den Modellen kommt, darf der hier beschriebene Prozess nicht als allgemeingültig, sondern vielmehr als beispielhaft verstanden werden.

2.2.2 Definitionsphase

Die Definitionsphase kann weiter unterteilt werden in eine Strategieweise und eine Initialphase [For03, San07]. In der Strategieweise werden die Anforderungen an ein neu zu entwickelndes Fahrzeug definiert und in einem Zielkatalog zusammengefasst, wobei der geplante wirtschaftliche Erfolg des Produktes im Vordergrund steht. Der lange betriebswirtschaftliche Planungshorizont macht umfangreiche Marktanalysen notwendig, in denen die Kundenwünsche identifiziert werden und mit deren Hilfe versucht wird, eine gezielte Marktprognose abzugeben. Neben den produktstrategischen Aspekten geben frühe Designstudien dem Fahrzeug eine erste Gestalt.

Der Strategieweise fehlt weitgehend der Bezug zur technischen Machbarkeit und sie kann daher der Produktforschung zugerechnet werden, wogegen in der anschließenden Initialphase die

zuvor ausgearbeiteten Anforderungen untersucht und plausibilisiert werden und diese daher eher einen Teil der Produktentwicklung darstellt. In Abb. 2.1 wird deshalb die übergeordnete Definitionsphase sowohl der Produktforschung als auch der Produktentwicklung zugeordnet.

Ein wesentliches Element der Initialphase besteht im Aufbau eines Maßkonzepts des Gesamtfahrzeugs, mit dem erste Bauraumuntersuchungen, Ergonomiestudien sowie Überprüfungen von gesetzlichen Vorgaben erfolgen. An dieser Stelle des Produktentstehungsprozesses werden zum ersten Mal CAD-Systeme eingesetzt, da eine Vielzahl der verwendeten Entwurfssysteme und -werkzeuge für Gesamtfahrzeugkonzepte in CAD-Programmen integriert sind oder auf diese zurückgreifen. Diese virtuellen Werkzeuge haben mitunter unterschiedliche Schwerpunkte, wie etwa Packaging [RHH⁺09], funktionale Auslegung [San07] oder Gesetzesanforderungen [Pot08].

Des Weiteren ist die Initialphase geprägt von einer Vielzahl an Designentwürfen, aus denen in einer Vorauswahl die aussichtsreichsten Konzepte für die weitere Entwicklung selektiert werden. Die Designkonzepte, deren Ausarbeitung zumeist in konkurrierenden Konzeptteams erfolgt, berücksichtigen schon erste technische Anforderungen aus dem Fahrzeugpackage [Ges01].

Neben der Erstellung eines Maßkonzepts und der Designvorauswahl werden die Anforderungen des Zielkatalogs in der Initialphase plausibilisiert und in einem Rahmenheft dokumentiert. Dieses enthält die wesentlichen Eigenschaften des zu entwickelnden Fahrzeugs. Darunter fallen etwa die Antriebsart, die Anzahl der Sitzplätze, die Karosseriebauart, die Grobabmessungen, das Zielgewicht aber auch Prozesseigenschaften wie beispielsweise der geplante Produktionsstart. Im Laufe des Entwicklungsprozesses entsteht aus dem Rahmenheft ein verbindliches Lastenheft [San07].

2.2.3 Konzeptphase

Die Hauptaufgabe der Konzeptphase besteht darin, die Machbarkeit des Entwicklungsprojekts vor dem Hintergrund der vorgegebenen Ressourcen sicherzustellen. Dies geschieht einerseits durch eine zunehmende Detaillierung des Fahrzeugkonzepts und andererseits mit Hilfe von Simulationen und Berechnungen, mit denen die Anforderungen überprüft und prinzipielle Lösungsvorschläge ausgearbeitet werden. Da in dieser Phase jedoch viele Eingangsdaten noch nicht vorhanden sind oder –wie beispielsweise im Bereich der Geometriemodellierung– parallel dazu erstellt werden, muss häufig auf Daten von Vorgängermodellen, heuristische Annahmen und empirische Erfahrungswerte zurückgegriffen werden. Oftmals reichen in dieser Phase jedoch qualitative Aussagen oder eine Untersuchung der Machbarkeit für eine Variantenauswahl aus.

Während der Konzeptphase kommt es ebenfalls zu einer Weiterentwicklung und Detaillierung des Designs. Das Fahrzeugdesign kann nicht nur unter künstlerischen Gesichtspunkten erstellt werden, sondern muss auch technischen Anforderungen genügen. Dieser Abgleich zwischen dem Design, das in der Konzeptphase schon in Form von Geometriedaten zur Verfügung steht, und den technischen Anforderungen wird auch als *Design-Technik-Konvergenz* bezeichnet und dauert bis zum sog. *Design-Freeze* am Beginn der Serienentwicklung, bei dem dann die Fahrzeuggestalt endgültig festgelegt wird. Im Laufe der Konzeptphase wird die Machbarkeit des

Fahrzeugkonzepts und somit auch des Designs überprüft. Dabei kommt es üblicherweise zu umfangreichen Änderungen und Anpassungen. Da die Designflächen eine wesentliche Eingabe bei der Erstellung einer konzeptionellen Bauteilgeometrie darstellen, verlangen die oftmaligen und umfangreichen Änderungen nach flexiblen Werkzeugen. Gerade im Bereich der Geometrieerstellung können hier parametrisch-assoziative CAD-Programme ihre Vorteile wesentlich zur Geltung bringen. Erschwerend bei der Geometriemodellierung kommt der Umstand hinzu, dass eine Entscheidung, welche Variante des Grobdesigns weiterverfolgt wird, erst im Laufe der Konzeptphase erfolgt und bis dorthin parallel an mehreren Designentwürfen gearbeitet wird [Ges01].

Im Bereich der geometrischen Integration und der Erstellung einer Konzeptgeometrie nimmt der DMU eine entscheidende Rolle ein. Der DMU, dem bereits eine Produktstruktur hinterlegt ist, dient als zentrales Datenmodell, das Grenzflächen sowie Designgeometrien für die Bauteilkonstruktion bereitstellt und mit dem Packaginguntersuchungen durchgeführt werden. In einer *design-in-context*-Arbeitsweise werden die aktuellen, zum Teil noch unfertigen, Konstruktionsstände der einzelnen Bauteile dem DMU zur Verfügung gestellt, sodass dieser einen hohen Grad an Aktualität besitzt [Mar03]. Die Geometriemodellierung an sich ist geprägt von einer Vielzahl an Varianten und verfolgt das Ziel, eine flexible Konzeptgeometrie bereitzustellen, die zwar noch keine vollständige Detaillierungstiefe aufweist, jedoch die wesentliche Gestalt der jeweiligen Baugruppen abbildet.

Während in der Definitionsphase die Entwicklung weitgehend in kleinen Konzeptteams durchgeführt wird, kommt in der Konzeptphase eine Vielzahl von Fachbereichen zum Einsatz. Mit der steigenden Detaillierungstiefe des Fahrzeugkonzepts nimmt auch die Anzahl der beteiligten Mitarbeiter zu. Gleichzeitig sind die Entwicklungsziele aus dem Rahmenheft jedoch vergleichsweise weich und müssen erst überprüft, angepasst und detailliert werden, bis sie schließlich als verbindlich gelten. Außerdem kommt es bei der funktionalen und geometrischen Auslegung sowie in der *Design-Technik-Konvergenz* zu einer Vielzahl an Varianten, Änderungen und Optimierungsschleifen, die zumeist parallel ablaufen. Dies führt zu einer heterogenen Entwicklungsumgebung, die nach flexiblen und robusten Entwicklungsmethoden verlangt.

Am Ende der Konzeptphase steht ein Fahrzeugkonzept zur Verfügung, das alle wesentlichen geometrischen und funktionalen Eigenschaften abbildet. Des Weiteren sind die Anforderungen an das Fahrzeug und die Entwicklungsziele detailliert und auf ihre Machbarkeit überprüft. Dies gilt insbesondere für neue Technologien, die parallel zum Fahrzeug noch bis zur Serienreife entwickelt werden müssen.

2.2.4 Vorentwicklungsphase

Das Ziel der Vorentwicklungsphase besteht darin, die prinzipiellen Lösungsvorschläge der Konzeptphase auszuarbeiten, das erstellte Fahrzeugkonzept abzusichern und das Fahrzeug zu optimieren. Dabei spielen virtuelle Methoden eine entscheidende Rolle, da Prototypen erst im Laufe der Vorentwicklungsphase zur Verfügung stehen. Es kommen insbesondere Simulationswerkzeuge zur Anwendung, mit denen die Eigenschaften des Fahrzeuges untersucht und Vorgaben

überprüft werden. In dieser Phase besitzen die Fahrzeugdaten eine Detaillierungstiefe, mit der sich die Wirksamkeit von Detailverbesserungen untersuchen lässt. So kann beispielsweise das Crashverhalten mit einer hohen Vorhersagegüte berechnet werden und der Einfluss von Details, wie etwa die Versickerung eines Bauteils, untersucht werden. Im Laufe der Vorentwicklungsphase kann zunehmend auf die Ergebnisse von Versuchen auf Baugruppen- und Komponentenebene zurückgegriffen werden.

Im Rahmen der Konstruktion werden die Bauteile verstärkt unter dem Gesichtspunkt der Produktion weiter detailliert. Gleichzeitig dazu erfolgt die Entwicklung der Produktionsanlagen und -werkzeuge. So werden beispielsweise Tiefziehwerkzeuge für Karosseriebauteile bereits auf Basis der vorhandenen Konzeptgeometrie entworfen, da sich die Entwicklung und Herstellung der Werkzeuge als zeitaufwändig erweist.

In dieser Phase kommt es zu einer intensiven Einbeziehung von Entwicklungspartnern, Systemlieferanten und Lieferanten. Während die Erstellung des Fahrzeugkonzepts als eine der Kernaufgaben eines Fahrzeugherstellers gilt, kommt es bei der Entwicklung der Submodule und Komponenten wie auch bei anderen Entwicklungs- und Produktionsumfängen zu Auslagerungen, die laut Prognosen 2015 einen Wertschöpfungsanteil von 77 % erreichen werden [DGHK04]. Diese enge Zusammenarbeit schafft auch einige Herausforderungen in der Projektorganisation und -kommunikation.

Das Resultat der Vorentwicklungsphase besteht in einem abgesicherten und detaillierten Fahrzeugkonzept. So muss mit Abschluss dieser Phase sichergestellt sein, dass alle Anforderungen aus dem Lastenheft erfüllt werden.

2.2.5 Serienentwicklungsphase

Während der Serienentwicklung kommt es zum Einsatz von Prototypenfahrzeugen und Versuchsträgern. Damit fließen Erkenntnisse aus Fahrzeugversuchen und Erprobungsläufen in die Entwicklung ein. Mit dem zunehmenden Einsatz einer virtuellen Entwicklung haben Fahrzeugversuche oftmals die Aufgabe, bereits durchgeführte virtuelle Auslegungen zu überprüfen und zu bestätigen. Durch die umfangreichen virtuellen Absicherungen soll vermieden werden, dass es in dieser Phase aufgrund der Ergebnisse aus Erprobungen zu gravierenden Änderungen am Fahrzeug kommt. Während in der Vergangenheit oftmals mehrere Prototypengenerationen üblich waren, geht der Trend dazu, die Generationen zu verschmelzen und die Aufgaben des Versuchs zunehmend mit virtuellen Methoden abzudecken. Dennoch sind Versuche unverzichtbar, da diese Eingangsdaten für die Simulation liefern, die Gesamtheit von äußeren Einflüssen abdecken sowie subjektive Beurteilungen und davon abgeleitete Abstimmungen zulassen [Deu06, SFBP07].

Mit dem zunehmenden Reifegrad der Konstruktion, die in der Serienentwicklung vor allem produktionstechnische Aspekte berücksichtigt, verringern sich die Freiheitsgrade für mögliche Änderungen massiv. Dieser Umstand wird dadurch verstärkt, dass bereits parallel zur Fahrzeugentwicklung die notwendigen Produktionsanlagen entwickelt und gebaut werden. Des Weiteren

können selbst geringfügige Änderungen am Fahrzeug einen Einfluss auf die geometrische Integration, die Produktion oder auf bereits abgesicherte funktionale Eigenschaften haben. Dadurch wird ein umfangreicher Freigabe- und Änderungsprozess notwendig.

Die Serienentwicklungsphase endet mit der Produktionsfreigabe. Die Produktentwicklung ist abgeschlossen und das Fahrzeug vollständig entwickelt. Gleichzeitig stehen am Ende dieses Entwicklungsabschnitts alle Produktionsanlagen und -werkzeuge zur Verfügung, mit denen dann seriennahe Prototypen hergestellt werden können.

2.2.6 Serienproduktionsphase

Die Serienproduktionsphase wird oftmals in eine Vorserien- und eine Serienphase unterteilt. In der Vorserienphase wird die Produktion schrittweise hochgefahren und es werden sukzessive die Stückzahlen erhöht sowie die Produktionstaktzeiten verkürzt, bis schlussendlich mit dem Produktionsstart die geplante Auslastung erreicht wird. Mit dem Produktionsstart beginnt die Serienphase und damit die eigentliche Serienproduktion. Bei den Fahrzeugen aus der Vorserie handelt es sich in der Regel bereits um Kundenfahrzeuge, die in weiterer Folge verkauft werden. In der industriellen Praxis sind jedoch zu diesem Zeitpunkt – entgegen den Planungen – oftmals nicht alle Entwicklungsziele erreicht. Dies führt üblicherweise zu Nacharbeiten und Ausbesserungen von Schwachstellen. Bei der Vorserienphase kommt es zu einem Übergang von der Produktentwicklung zur Produktherstellung. Da Vorserienfahrzeuge schon an Kunden verkauft werden können, wird die Vorserienphase daher in Abb. 2.1 der Produktherstellung zugeordnet.

Verbesserungen des resultierenden Produkts finden in der Serienbetreuung statt. Diese ist vor allem durch produktions- und qualitätsbedingte Aspekte gekennzeichnet. Darüber hinaus kommt es in der Mitte des Produktionszyklus häufig zu einer umfangreichen Überarbeitung des Fahrzeugs. Bei diesem sog. *Facelift* werden im Wesentlichen die Phasen aus Abb. 2.1 durchlaufen, wenngleich es sich dabei um keine Neuentwicklung sondern vielmehr um eine Weiterentwicklung handelt.

2.3 Trends in der Produktentstehung

Der hier beschriebene Produktentstehungsprozess skizziert einen typischen Entwicklungsablauf, wie er derzeit in der industriellen Praxis vorgefunden wird. Dieser Ablauf unterliegt jedoch einem ständigen Wandel, dessen Ursprung unter anderem im Wettbewerbsdruck liegt, wie er im Kundenmarkt Automobil vorherrscht. Dieser Verdrängungswettbewerb führt zu einer zunehmenden Produktvielfalt und einem erhöhten Qualitätsbewusstsein. Gleichzeitig erhöht sich der Kostendruck bei den Herstellern, obwohl die Anforderungen der Kunden an das resultierende Produkt steigen. Diese Anforderungen führen zu einem vermehrten Einsatz von elektronischen Komponenten, aktiven Systemen, nachrichtentechnischen Anwendungen, unterschiedlichen Werkstoffen und vielen weiteren Maßnahmen, die neben dem gestiegenen Kundennutzen auch zu einer Erhöhung der Produktkomplexität beitragen. Neben den Kundenanforderungen

müssen auch gesetzliche Auflagen und Richtlinien berücksichtigt werden, die mitunter länderspezifisch voneinander abweichen und zu unterschiedlichen Rahmenbedingungen führen. Um die steigende Produkt- und Prozesskomplexität handhaben zu können, kommt es zu ständigen Anpassungen des Produktentstehungsprozesses wie auch der zugehörigen Werkzeuge und Methoden [ES09].

Der strategische Ansatz des *Product-Lifecycle-Managements* (PLM) verfolgt etwa das Ziel, den Produktlebenszyklus ganzheitlich zu betrachten. Es wird versucht, alle beteiligten Instanzen zu integrieren, untereinander abzustimmen und ein resultierendes Produkt von der Produktidee bis zur Produktentsorgung zu beeinflussen. Ein wesentlicher Bereich von PLM umfasst das *Produktdaten-Management* (PDM). Dieses stellt sicher, dass in der Entwicklung auf eine einheitliche und übergreifende Datenbasis zugegriffen werden kann. Moderne PDM-Programme übernehmen neben der reinen Dokumentation auch Aufgaben des Prozess- und Änderungsmanagements, den Strukturaufbau von Daten und dienen als Schnittstelle zu den anwendungsorientierten Programmen für Simulation, Konstruktion oder Projektmanagement. Der Trend geht zu einer zunehmenden Integration von PDM und den zugreifenden Programmen. So kommt es gerade im Bereich der Konstruktion zu einer verstärkten Verschränkung zwischen CAD-Programmen und PDM-Systemen [BR97, WWD03, Abr07, ES09].

Des Weiteren kommt es zu einem vermehrten Zusammenführen der bisher unabhängigen Produktmodelle, die in den einzelnen Domänen, wie Design, Konstruktion, Berechnung oder Fertigungsplanung, verwendet werden. Mit dieser umfassenden PDM-Ausprägung wird die Grundlage für einen *Digitalen Prototypen* geschaffen, mit dem –vergleichbar mit einem realen Prototypen– virtuelle Versuche und Absicherungen durchgeführt werden können [Wan02]. Diese Verschmelzung von unterschiedlichen Daten führt schlussendlich zu einem umfassenden und integrierten digitalen Produktmodell [SK97].

Ein wesentlicher Trend im Entwicklungsprozess liegt in der Globalisierung und der zunehmenden Verzahnung von Fahrzeugherstellern und Lieferanten. Wurde in den 80er-Jahren des 20. Jahrhunderts die Entwicklung eines Fahrzeuges fast ausschließlich durch den Fahrzeughersteller durchgeführt, so konzentrieren sich die Hersteller derzeit auf ihre Kernkompetenzen wie etwa Motoren- und Karosserieentwicklung oder die Konzepterstellung. Die Entwicklung von Baugruppen, die nicht in diese Kernkompetenzen fallen, wird an Systemlieferanten ausgegliedert und die Fahrzeughersteller beschränken sich auf die Vorgabe von Anforderungen und die Integration der resultierenden Module. Neben den Auslagerungen an Lieferanten kommt es auch verstärkt zu einer Zusammenarbeit zwischen den unterschiedlichen Herstellern, die zu gemeinsam entwickelten Fahrzeugplattformen oder Motoren führt. Durch diese intensive und globale Kooperation zwischen Herstellern, Lieferanten, aber auch Kunden kommt den jeweiligen Schnittstellen für den Kommunikations- und Informationsaustausch eine bedeutende Rolle zu. Methoden und Ansätze, die das Ziel verfolgen, eine optimale Zusammenarbeit zwischen allen beteiligten Parteien sicherzustellen, werden unter dem Sammelbegriff des *Collaborative Engineering* zusammengefasst [WdM98, KSV08].

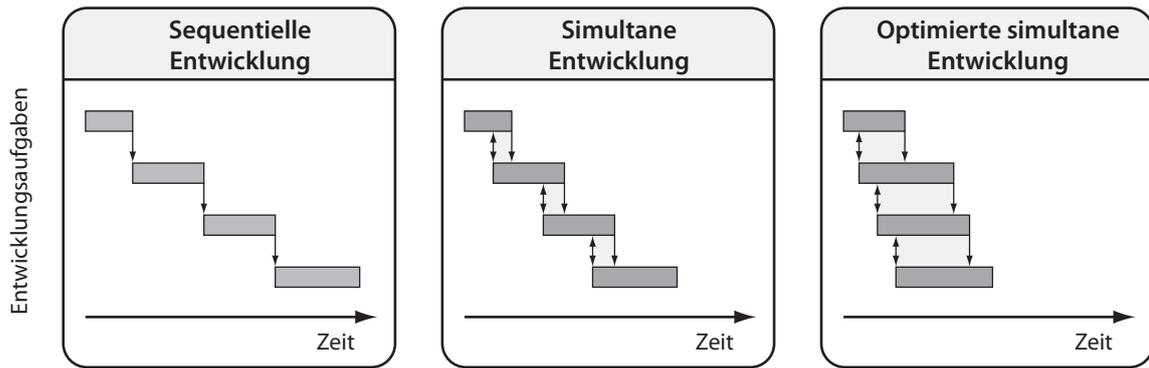


Abbildung 2.2: Entwicklungszeitverkürzung durch *Simultaneous Engineering*.

Ein weiterer Trend, der den Produktentstehungsprozess in den letzten Jahrzehnten wesentlich beeinflusst hat, besteht im *Simultaneous Engineering*⁶. Darunter versteht man die Parallelisierung der Entwicklungsabläufe, mit dem Ziel, die Entwicklungszeit zu reduzieren. Wird in einer konventionellen, sequentiellen Entwicklung mit dem Start einer Entwicklungsaufgabe gewartet, bis der vorangegangene Entwicklungsschritt abgeschlossen ist, so beginnt bei der simultanen Entwicklung eine neue Entwicklungsaufgabe, sobald schon vorläufige Ergebnisse vorhanden sind, Abb. 2.2. Dies führt zu einer Reduktion der gesamten Entwicklungsdauer, wenngleich einzelne Arbeitsschritte mitunter länger dauern, da sich die Eingangsdaten häufig ändern und es zu einem verstärkten Kommunikations- und Abstimmungsaufwand kommt. Dennoch bietet eine Verkürzung der Gesamtentwicklungszeit wesentliche Vorteile. So kann unter anderem der Planungshorizont verringert werden und schneller auf Markterfordernisse und Kundenwünsche reagiert werden. Gleichzeitig kommt es aufgrund der übergreifenden Entwicklung zu einer steigenden Qualität, da schon frühzeitig Anforderungen und Probleme von folgenden Entwicklungsabläufen berücksichtigt werden. Eine simultane Entwicklung benötigt flexible Methoden und Werkzeuge, die gut mit sich ändernden und unvollständigen Eingangsdaten umgehen können, wie dies etwa bei wissensbasierten CAD-Methoden der Fall ist [Soh92, SON06].

Bei dem beschriebenen Produktentstehungsprozess in Kapitel 2.2 handelt es sich um einen *Simultaneous Engineering*-Prozess, wie er derzeit Stand der Technik ist. So wird beispielsweise bereits mit der Planung und Entwicklung der Werkzeuge begonnen, obwohl die Bauteile noch nicht ihre endgültige Form haben. Auch wenn dieser Prozess bereits einen hohen Grad an Parallelisierung aufweist, wird weiterhin versucht, den Ansatz des *Simultaneous Engineering* voranzutreiben, um mit einem optimierten simultanen Entwicklungsprozess eine minimale Entwicklungszeit zu erreichen, Abb. 2.2.

Bei den vorgestellten Trends und Ansätzen in der Produktentwicklung handelt es sich um eine repräsentative Auswahl mit unterschiedlichen Schwerpunkten. Ein wesentlicher Aspekt, den alle Ansätze beinhalten, besteht in der Notwendigkeit für eine umfassende Verwendung von virtuellen Methoden und Werkzeugen. Nur durch ihren gezielten Einsatz lassen sich die Ziele,

⁶Neben der Bezeichnung *Simultaneous Engineering* wird auch der Begriff *Concurrent Engineering* verwendet.

wie etwa eine Verkürzung der Entwicklungszeit sowie eine Verbesserung der Produktqualität, erreichen. Eine besondere Bedeutung wird hierbei der frühen Entwicklungsphase beigemessen.

2.4 Bedeutung der frühen Entwicklungsphase

Die Abgrenzung der frühen Entwicklungsphase gestaltet sich als schwierig, da es in der Industrie wie auch in der Literatur keine einheitliche Abgrenzung zu den folgenden Abschnitten gibt und die frühe Phase für die jeweilige Anwendung oftmals unterschiedlich definiert wird. In dieser Arbeit werden die Definitions- und die Konzeptphase als Bestandteile einer frühen Entwicklungsphase angesehen, während die Vorentwicklungs- und Serienentwicklungsphase als seriennahe Entwicklung bezeichnet werden.

Ungeachtet der unterschiedlichen Abgrenzungen hat die frühe Entwicklungsphase einen großen Einfluss auf die Produktentstehung und die dafür notwendigen Ressourcen. Betrachtet man die Kosten, die bei einer Produktänderung entstünden, so zeigt sich, dass diese mit zunehmendem Projektfortschritt drastisch steigen, Abb. 2.3. Es wird sogar von einem exponentiellen Anstieg der Änderungskosten ausgegangen [EKL07]. Gegenläufig dazu sinken die Freiheitsgrade für Entscheidungen, da wesentliche Produkteigenschaften schon in einer frühen Entwicklungsphase festgelegt werden und Änderungen zu einem späteren Zeitpunkt zu unverhältnismäßig hohen Kosten führen würden. Demgegenüber steigt das Produktwissen in einer frühen Phase vergleichsweise langsam an. Dies führt dazu, dass in dieser Phase der Hauptteil der Kosten und Aufwände definiert wird – *Wozny* geht beispielsweise von 70% der akkumulierten Kosten aus –, wobei gleichzeitig relativ wenig über das Produkt bekannt ist [Woz92].

Die Bedeutung der frühen Phase liegt insbesondere darin, dass charakteristische Produkteigenschaften und die damit entstehenden Kosten und Aufwände definiert werden, gleichzeitig eine hohe Anzahl an Freiheitsgraden vorhanden ist und Änderungen am Produkt schnell und kostengünstig durchgeführt werden können. Die große Abweichung zwischen Kostenfestlegung und Produktwissen macht jedoch eine Stärkung der frühen Phase und eine Erweiterung des Produktwissens notwendig, sodass dieses idealerweise mit den festgelegten Kosten korreliert, Abb. 2.3. Es wird daher versucht, Wissen, das bei einem konventionellen Entwicklungsprozess erst im Laufe der Produktentwicklung erlangt wird, bereits frühzeitig bereitzustellen. Ansätze, die zu einem frühzeitigen Wissenszuwachs führen, werden unter dem Sammelbegriff *Frontloading* zusammengefasst. Da dieser frühzeitige Wissenszuwachs für den betriebswirtschaftlichen Erfolg des resultierenden Produkts bedeutsam ist, haben auch *Frontloading*-Methoden einen hohen unternehmerischen Stellenwert.

Ein wesentlicher Teil dieser *Frontloading*-Methoden wird durch Simulationen abgedeckt. Mit Hilfe von Berechnungen lassen sich frühzeitig Eigenschaften überprüfen, Variantenstudien durchführen, Fehler aufdecken und Lösungen optimieren. Aus Mangel an physischen Prototypen kann in einer frühen Phase zumeist nur auf Berechnungen zurückgegriffen werden, um Produktwissen von einer Neuentwicklung zu erwerben. Eine wesentliche Herausforderung bei Simulationen besteht jedoch im Bereitstellen von Eingangsdaten, da die Güte der Ergebnisse wesentlich von den Eingaben abhängt. Diese sind jedoch in einem frühen Entwicklungsabschnitt zumeist in

einem unvollständigen Umfang und in einer unzureichenden Qualität vorhanden, sodass oftmals nur Abschätzungen oder qualitative Aussagen möglich sind. Um Konzeptvarianten nun vergleichen und einordnen zu können sowie die besten Varianten zu identifizieren, sind daher Kennzahlen wie auch Bewertungs- und Entscheidungsmethoden notwendig.

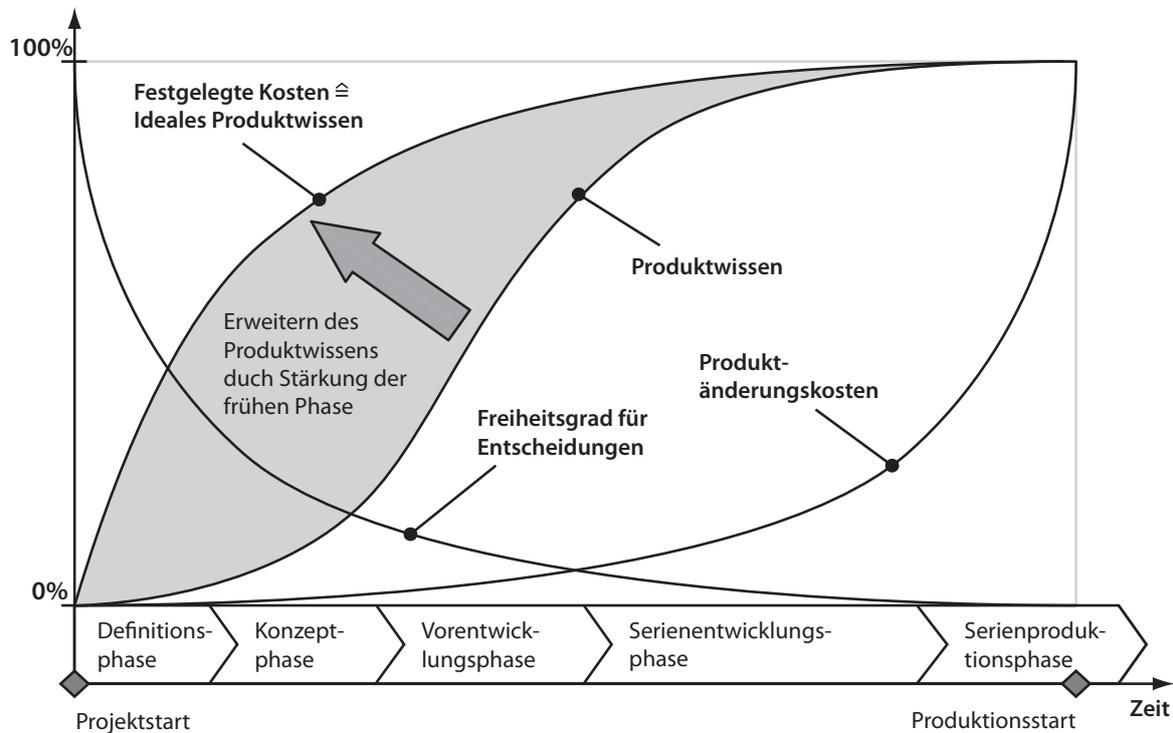


Abbildung 2.3: Schematische Darstellung von Produktwissen, Produktänderungskosten und Freiheitsgrad für Entscheidungen in einem Produktentstehungsprozess, nach [SK97]⁷.

Ansätze, die durchaus den *Frontloading*-Methoden zugerechnet werden können, bestehen etwa im *Design-for-x* und in der *Digitalen Fabrik*. Bei ersterem handelt es sich im Bereich der Konstruktion um eine ganzheitliche Betrachtung und Beeinflussung von Produkteigenschaften. So soll basierend auf Gestaltungsrichtlinien schon frühzeitig auf Eigenschaften wie etwa die Herstellbarkeit oder die Qualität Rücksicht genommen werden. Die *Digitale Fabrik* wird im Bereich der Produktionsplanung herangezogen. Hierzu wird ein virtueller Fabriksprototyp erstellt, mit dem alle wesentlichen Prozesse und Ressourcen abgebildet werden und der zu einer frühzeitigen Beeinflussung des Produktionsaufbaus und der Produktion dient. Die *Digitale Fabrik* wird auf Basis der virtuellen Produktdaten des Fahrzeugs erstellt und bildet eine Schnittstelle zwischen Produktentwicklung und Produktionsplanung [VDI06, BGW11].

Neben den virtuellen Methoden können etwa durch *Rapid Prototyping* (RPT⁸) bereits in einer frühen Phase Prototypen von Bauteilen hergestellt werden. Der Vorteil liegt in der schnellen

⁷Im Gegensatz zu *Spur/Kraus* liegt der Anfangswert des Produktwissens bei 0%, da im Phasenmodell aus Kapitel 2.2 neben der Produktentwicklung und -herstellung auch die Produktforschung inkludiert ist, sodass die Produktentstehung mit der Produktidee beginnt.

⁸Rapid Prototyping

Verfügbarkeit und der Möglichkeit, Versuche durchzuführen. Demgegenüber stehen hohe Kosten und der Umstand, dass die Bauteile zumeist nicht mit seriennahen Herstellungsverfahren erzeugt werden. Dadurch kann nicht immer ausreichend genau auf die späteren Produkteigenschaften geschlossen werden.

Ein zentrales Element des *Frontloading*-Ansatzes besteht in der frühzeitigen Bereitstellung von Expertenwissen. So können beispielsweise Erfahrungen und Kennwerte von Vorgängerprojekten dokumentiert und zur Verfügung gestellt werden. Dieses Expertenwissen kann auch in Form von *Templates* oder Assistenzsystemen, die jeweils bereits erprobte Methoden und Modelle enthalten, vorliegen. Je nach Ausprägung gibt es sowohl *Templates* und Assistenzsysteme für die Berechnung als auch für die Konstruktion, wobei gerade für letzteres wissensbasierte CAD-Methoden zur Anwendung kommen. Speziell bei der Geometriemodellierung in einer frühen Phase spielen diese Ansätze eine entscheidende Rolle, da mit dem hinterlegten Expertenwissen bereits Auslegungen gemacht werden können, ohne etwa auf Simulationsergebnisse warten zu müssen. Darüber hinaus kann mit Hilfe von Bibliotheken auf bereits erstellte und eingesetzte Bauteilgeometrien zurückgegriffen werden. Durch diese Maßnahmen wird der Konstruktionsprozess wesentlich beschleunigt und die Ergebnisqualität verbessert.

2.5 Zusammenfassung

Das vorgestellte Phasenmodell spiegelt den Entwicklungsprozess eines Fahrzeuges wieder, wie er heutzutage in der Industrie umgesetzt wird. Neben einer Abgrenzung und Beschreibung der einzelnen Abschnitte wird vor allem auf die Trends und Strömungen eingegangen, welche die Produktentwicklung aus heutiger Sicht beeinflussen. Diese Trends, wie etwa eine Verkürzung der Entwicklungszeit oder eine Steigerung der Produktkomplexität, machen einen verstärkten Einsatz von virtuellen Methoden und Werkzeugen erforderlich. Ein besonderes Augenmerk liegt hierbei auf der frühen Entwicklungsphase, da diese das resultierende Produkt wie auch die dafür notwendigen Ressourcen entscheidend beeinflusst. Gerade in dieser Phase, die durch ein heterogenes und sich schnell änderndes Entwicklungsumfeld gekennzeichnet ist, bietet sich der vermehrte Einsatz von *Frontloading*-Methoden an. Dadurch lässt sich bereits erworbenes Expertenwissen bereitstellen und damit die konzeptionelle Fahrzeugentwicklung effizienter gestalten.

3 Wissensbasierte CAD-Methoden

3.1 Einleitung

Computerunterstützte Zeichenprogramme haben sich im Laufe der Zeit von digitalen Zeichenbrettern zu hochfunktionalen virtuellen Werkzeugen weiterentwickelt. Neben der reinen Geometriemodellierung steht eine Fülle an zusätzlichen Funktionen zur Verfügung, die unter anderem dazu verwendet werden können, den Konstruktionsprozess zu verbessern. An dieser Stelle wird ausgehend von den Anforderungen einer wissensbasierten Konstruktion auf die Merkmale von modernen CAD-Programmen eingegangen und ihre Rolle im Verbund einer virtuellen Entwicklung beleuchtet. In weiterer Folge werden Methoden für die Anwendung von *Templates*, als wichtige Bestandteile von wissensbasierten CAD-Systemen, beschrieben. Die unterschiedlichen Ausprägungen von *Templates* werden kategorisiert und es wird auf ihre Merkmale und ihre Einsatzmöglichkeiten eingegangen. Die hier angeführten Eigenschaften bilden die Grundlage für die weitere Arbeit, insbesondere bei der prototypischen Umsetzung im Rahmen der Anwendungsbeispiele.

3.2 Wissensbasierte Konstruktion

Die Bereitstellung von Wissen in einer frühen Entwicklungsphase gilt als eine zentrale *Front-loading*-Methode und führt im Bereich der Konstruktion zum Ansatz der *Wissensbasierten Konstruktion* (KBE⁹). Dieser Ansatz stellt eine technikorientierte Form des Wissensmanagements dar, die speziell auf die Produktentwicklung und die damit verbundenen Aufgaben zielt [Str06]. Dabei wird vorhandenes Prozess- und Produktwissen identifiziert, aufbereitet und dem Konstruktionsprozess im Rahmen von Assistenzsystemen zur Verfügung gestellt. Dies kann in einer vielschichtigen Weise geschehen, was dazu führt, dass es eine Fülle an unterschiedlichen Definitionen und Abgrenzungen von KBE gibt. Generell kann KBE als Sammelbegriff für Methoden und Anwendungen herangezogen werden, die auf Basis von Wissen bei Konstruktionstätigkeiten unterstützen oder diese automatisieren [VBGvC12].

KBE-Anwendungen spiegeln firmenspezifisches Wissen wider und sind oftmals auf die Bedürfnisse und Anforderungen des jeweiligen Unternehmens ausgerichtet. Bei einer entsprechenden Umsetzung sorgen sie für einen Wettbewerbsvorteil und können daher den Erfolg eines Unternehmens beeinflussen. Aus diesen Gründen handelt es sich bei KBE-Anwendungen oftmals um individuelle Produkte, die entweder firmenintern oder durch eine externe Auftragsentwicklung

⁹Knowledge Based Engineering

erstellt werden. Kommerziell erhältliche KBE-Anwendungen liefern aufgrund ihrer breiten Verfügbarkeit keinen unmittelbaren Wettbewerbsvorteil und stellen – gleich wie Auftragsentwicklungen – in der Regel nicht die notwendigen Ressourcen für die Einführung und Umsetzung im Unternehmen. Darüber hinaus bestehen kaum Möglichkeiten, die Anwendungen aufgrund von internem Wissen zu verbessern oder zu individualisieren, ohne von den zuvor ausgewählten externen Partnern abhängig zu sein. Allerdings bieten sie im Vergleich zu Eigenentwicklungen in der Regel Kostenvorteile, da sich der Entwicklungsaufwand auf mehrere Abnehmer aufteilt.

Die Entwicklung von KBE-Anwendungen erfolgt – unabhängig davon, ob es sich um eine unternehmensinterne oder -externe Entwicklung handelt – entkoppelt vom eigentlichen Konstruktionsprozess, Abb. 3.1. Die Erstellung beinhaltet die Zieldefinition, die Entwicklung eines virtuellen Prototypen, die Validierung des Prototypes und die Einführung der resultierenden Anwendung [KS95]. Nach der Implementierung kann ein KBE-System in den jeweiligen Konstruktionsprojekten genutzt werden. Idealerweise kommt es nach dem Einsatz einer KBE-Anwendung zu einem Informationsrückfluss, der Rückschlüsse auf die Wirksamkeit der Anwendung zulässt und Möglichkeiten für Verbesserungen aufzeigt. Durch laufende Anpassungen kann sichergestellt werden, dass neue Erkenntnisse, veränderte Anforderungen und gesammelte Erfahrungen entsprechend dem *Best-Practice*-Ansatz ständig in die KBE-Anwendungen einfließen [HAR06].

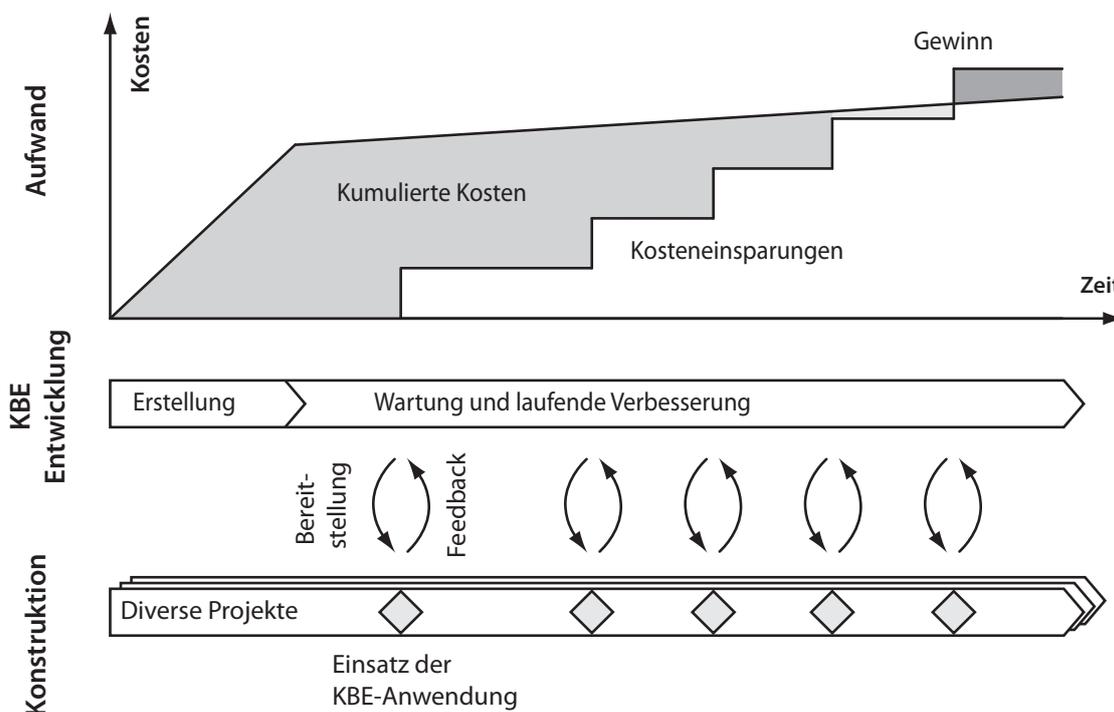


Abbildung 3.1: Phasenmodell für die Entwicklung und den Einsatz von KBE sowie ein Schema der Kostenentwicklung.

Abb. 3.1 zeigt ebenfalls den Kostenverlauf bei der Entwicklung und beim Einsatz einer KBE-Anwendung. Während sich die Erstellung der Anwendung als kostenintensiv erweist, fallen in der laufenden Betreuung vergleichsweise geringe Aufwände an. Gleichzeitig kommt es durch

den Einsatz der KBE-Anwendung in den Konstruktionsprojekten zu Zeit- und Kosteneinsparungen im Vergleich zu einer konventionellen Konstruktion, wodurch sich bei einer mehrmaligen Nutzung die anfänglichen Aufwände amortisieren. In Abb. 3.1 wird jede Einsparung, die durch einen KBE-Einsatz entstanden ist, modellhaft als Stufe dargestellt.

Wenngleich die Umsetzung von KBE-Anwendungen in vielen Fällen zu einer gewünschten Verbesserung des Konstruktionsablaufs führt, so eignen sich nicht alle Konstruktionsaufgaben für den Einsatz von wissensbasierten Konstruktionsmethoden und -werkzeugen. Darüber hinaus sind für eine erfolgreiche Umsetzung diverse Randbedingungen zu beachten. In der folgenden Aufzählung werden, angelehnt an *Stokes*, Voraussetzungen angeführt, unter denen sich eine Umsetzung von KBE nicht als zielführend erweist [SM01]:

- Die Konstruktionsaufgabe kann leicht und schnell manuell ausgeführt werden, sodass eine Automatisierung nur zu einer geringfügigen Zeit- und damit Kostenersparnis führt. In diesem Fall amortisieren sich die anfallenden Kosten in Abb. 3.1 unverhältnismäßig spät oder aufgrund der ständig vorhandenen Wartungs- und Bereitstellungskosten nie.
- Die Technologien des Produktes oder der Konstruktionswerkzeuge ändern sich ständig, sodass eine zu Grunde liegende KBE-Anwendung ebenfalls regelmäßig und umfassend verändert werden müsste. Es besteht die Gefahr, dass die KBE-Anwendung andauernd der Konstruktionspraxis hinterherhinkt und dadurch keinen Vorteil bei der Verwendung bieten kann.
- Das Unternehmen hat nicht den Willen oder die Ressourcen, um KBE-Anwendungen einzuführen und nachhaltig zu betreuen. Derzeit tendieren Betriebe vielfach dazu Methoden- und Softwareentwicklung auszulagern. Im Fall einer wissensbasierten Konstruktion sind jedoch diese Bereiche zumindest in rudimentärer Form unternehmensintern notwendig, um nachhaltig KBE-Anwendungen einzuführen und aufrecht zu erhalten. Selbst wenn die eigentliche Erstellung extern durchgeführt wird, stellt die Identifikation von Wissen und die Integration der Werkzeuge in bestehende Prozesse vorrangig eine innerbetriebliche Aufgabe dar.
- Das Wissen für die konkrete Aufgabe steht nicht zur Verfügung. Dies kann unter anderem bei Konstruktionsumfängen auftreten, die regelmäßig ausgelagert werden, wodurch das zu Grunde liegende Wissen zur Lösung der Aufgabe mitunter veraltet ist oder abhanden kommt. Darüber hinaus besteht über vorhandenes Wissen oftmals ein unvollständiges Bewusstsein, sodass dieses nicht in die Erstellung von KBE-Applikationen einfließt, sondern erst bei aktuellen Problemstellungen zutage tritt [FGSS90]. Dieser Umstand belegt die Wichtigkeit einer ständigen Betreuung und Verbesserung von KBE-Anwendungen.
- Sobald die Konstruktionsaufgabe einen hohen Anteil an Kreativität beinhaltet, ist sie einer wissensbasierten Konstruktion schwer zugänglich. Gerade dieser Punkt sorgt vielfach für Kontroversen, da der kreative Anteil einer Konstruktion oft schwer quantifizierbar ist und generell die Gefahr gesehen wird, dass durch Einsatz von KBE originäre, oft vorteilhafte Merkmale in einer Konstruktion verschwinden und diese einen normteilartigen Charakter erhält [Man08b].

Sollte eine geeignete Konstruktionsaufgabe identifiziert sein, so stehen für die Entwicklung der entsprechenden KBE-Anwendung, abhängig von der Komplexität der Aufgabe, eine Fülle an Methoden und Werkzeugen zur Verfügung. Diese reichen von relativ simplen Automatisierungsroutinen bis hin zu vielschichtigen Entwicklungssystemen, die mitunter Funktionalitäten von externen Programmen nutzen. So lassen sich beispielsweise Simulationsumgebungen oder Optimierungsroutinen integrieren und es kann auf Datenbanken zugegriffen werden. Ein Ansatz, um weitreichende, modellbasierte KBE-Anwendungen systematisch zu entwickeln, besteht in MOKA¹⁰. Diese Vorgehensweise wurde im Rahmen eines öffentlichen Forschungsprojekts aufgestellt und bietet einen KBE-Lebenszyklus mit sechs Phasen, der als Leitfaden für eine KBE-Entwicklung herangezogen werden kann. MOKA umfasst sowohl Methoden als auch Werkzeuge und hat das Ziel, einerseits die KBE-Erstellung effizienter zu gestalten und andererseits als Basis für einen internationalen Standard zu dienen [SM01, Lie04].

Unabhängig von der gewählten Methode zur Erstellung der KBE-Anwendungen und -Assistenzsystemen besteht ein wesentlicher Aspekt für die resultierende Wirksamkeit einer wissensbasierten Konstruktion in der Integration von KBE-Anwendungen in die CAD-Umgebung. Auch in einer wissensbasierten Konstruktion kommen CAD-Systeme als zentrale digitale Konstruktionswerkzeuge zum Einsatz und bieten gerade im Bereich der Geometriemodellierung weitreichende Funktionalitäten. Es wird daher versucht, KBE-Systeme in eine vorhandene CAD-Umgebung einzubinden, wobei der Grad an Integration unter anderem von den Fähigkeiten des CAD-Programms abhängt [Leh89, Str06]. Abb. 3.2 zeigt drei unterschiedliche Integrationsstufen von KBE-Systemen in CAD-Umgebungen: getrennt, gekoppelt und eingebettet. Diese drei Stufen eignen sich generell, um die Verbindung von CAD-Programmen zu weiteren Umgebungen, wie etwa CAE oder CAS, einzuordnen.

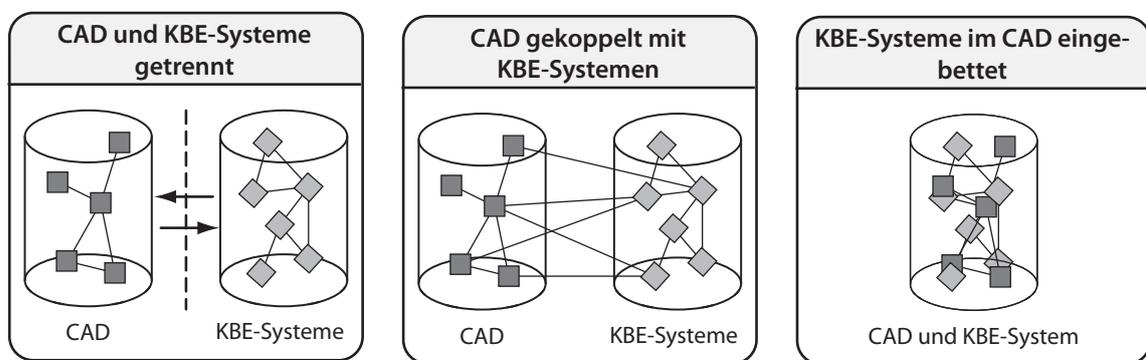


Abbildung 3.2: Integrationsstufen von CAD- und KBE-Systemen, nach [Str06].

Beim geringsten Grad an Integration sind beide Systeme voneinander getrennt und eine Kommunikation erfolgt über einen Austausch von Daten. Als Nachteil kann dabei angesehen werden, dass der Austausch zumeist manuell erfolgt, was den Zeitaufwand und die Fehleranfälligkeit erhöht. Außerdem kann selbst bei Verwendung von standardisierten, neutralen Datenformaten zumeist nicht auf die vollständige Produkt- und Geometriebeschreibung innerhalb des CAD-Systems zugegriffen werden. Ein Vorteil der Trennung von CAD- und KBE-Systemen liegt in

¹⁰A methodology and tools oriented to knowledge-based engineering application

der Unabhängigkeit der Systeme, sodass beispielsweise bei der Verwendung von unterschiedlicher CAD-Software die gleichen KBE-Anwendungen zum Einsatz kommen können.

Eine Koppelung von CAD- und KBE-Anwendung kann einerseits die Nachteile einer getrennten Anbindung vermeiden und andererseits die Leistungsfähigkeit des Gesamtsystems steigern. So kann etwa mit Hilfe von Programmierschnittstellen sowohl auf die internen Leistungsumfänge des CAD-Programms als auch jene des KBE-Systems zugegriffen werden. Dies schafft die Möglichkeiten von bidirektionalen Koppelungen, mit denen beispielsweise Parametervariationen der CAD-Geometrie und darauf aufbauende Optimierungen durchgeführt werden können. Der Gewinn an Benutzerfreundlichkeit und Funktionalität bringt jedoch einen erhöhten Erstellungs- und Wartungsaufwand mit sich. Außerdem sind Anpassungen und Standardisierungen notwendig, um eine KBE-Anwendung mit unterschiedlichen CAD-Programmen zu koppeln.

Bei einer vollständigen Integration von KBE-Systemen in die CAD-Umgebung kann es durchaus zu Vereinfachungen bei der Erstellung einer KBE-Anwendung kommen. Hierbei sind die KBE-Funktionalitäten ein Teil des CAD-Programms und es kommt zu einem Wegfall von Schnittstellen. Sofern die Leistungsumfänge der integrierten KBE-Funktionalitäten ausreichen, kann gänzlich auf externe Systeme verzichtet werden. Nachteilig wirkt sich bei dieser vollständigen Integration die starke Bindung an das CAD-System aus, die eine Abhängigkeit von dem verwendeten CAD-Programm schafft.

Aufgrund der unterschiedlichen Komplexität der jeweiligen Konstruktionsaufgabe sowie von innerbetrieblichen Randbedingungen, wie etwa die zur Verfügung stehenden Softwarepakete, kommt es in der industriellen Praxis zum Einsatz der gesamten dargestellten Bandbreite an Integrationsstufen. Während jedoch ein getrennter Einsatz von CAD- und KBE-Systemen kaum Anforderungen an die jeweiligen Programme nach sich zieht, stellt der Einsatz von gekoppelten oder eingebetteten KBE-Systemen wesentlich höhere Ansprüche an die Systemumwelt. So führt insbesondere die Integration in PDM-Systeme zu einer Herausforderung, da die eingesetzten Datenmanagementsysteme mitunter nur eingeschränkt für die Archivierung und Bereitstellung von komplexen KBE-Anwendungen geeignet sind [CM07]. Gleichzeitig steigen auch die Anforderungen an die Funktionalität des CAD-Systems. Dies ist der Ausgangspunkt für die Entwicklung von wissensbasierten CAD-Programmen.

3.3 CAD-Systeme und deren Entwicklung

3.3.1 Entwicklungsstufen von CAD-Systemen

Um den Anforderungen einer effizienten Geometriemodellierung sowie einer umfassenden Darstellung von übergeordneten Produktdaten gerecht zu werden, haben sich CAD-Programme von digitalen Werkzeugen zur Zeichnungserstellung zu hochfunktionalen, virtuellen Produktentwicklungssystemen gewandelt. Abb. 3.3 zeigt anhand eines einfachen Bauteilmodells die wesentlichen Entwicklungsstufen der CAD-Geometriemodellierung, wobei die einzelnen Stufen auf die vorangegangenen aufbauen und diese erweitern. Angefangen von einer konventionellen

Modellierungstechnik über parametrisch-assoziatives und featurebasiertes CAD liefert wissensbasiertes CAD derzeit die größte Anzahl an Funktionalitäten und besitzt damit verbunden auch die höchste Komplexität [VDI09, Bro10].

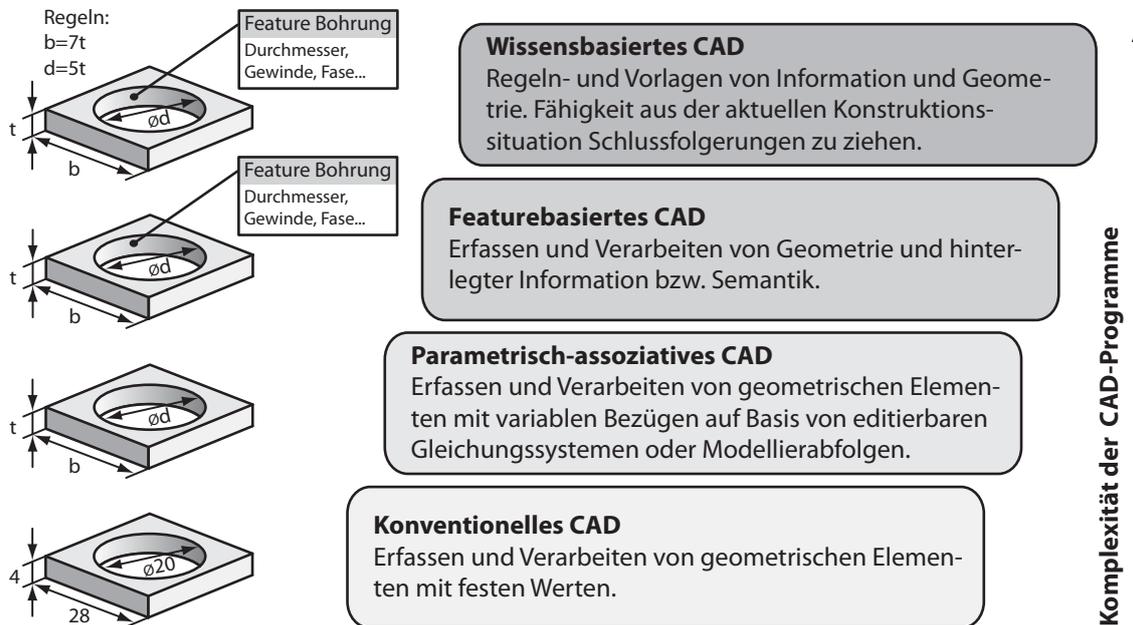


Abbildung 3.3: Entwicklungsstufen der CAD-Geometriemodellierung am Beispiel eines einfachen Bauteilmodells, nach [VDI09].

Die ersten drei Entwicklungsstufen haben einen großen Einfluss auf die Techniken und Vorgehensweisen bei der Geometriemodellierung. Während das konventionelle CAD einem digitalen Zeichenbrett gleichkommt, so ist bei den folgenden Entwicklungsstufen darauf zu achten, dass die resultierende Geometrie parametrigerecht bzw. featuregerecht aufgebaut wird [VDI09]. Wenngleich dies anfänglich zu einem erhöhten Erstellungsaufwand führt, können damit jedoch nachvollziehbare und vor allem leicht änderbare Produktmodelle aufgebaut werden, sodass es im Laufe der Produktentstehung insgesamt zu einer wesentlichen Reduktion des Aufwands kommt [Has06]. Wissensbasiertes CAD ändert nicht vordergründig die Modellierungstechnik der Geometrie, schafft jedoch mit den bereitgestellten Funktionalitäten, wie etwa *Formeln* oder *Regeln*, die Möglichkeit, vorhandenes Wissen zu hinterlegen und damit den Konstruktionsprozess effizienter zu gestalten. Wissensbasierte CAD-Funktionalitäten finden daher bei der eigentlichen Konstruktion kaum Verwendung, sondern werden vielmehr bei der Erstellung von wissensbasierten Anwendungen und Assistenzsystemen herangezogen, die dann dem Konstruktionsprozess zugeführt werden und diesen verbessern.

Konventionelles CAD entspricht dem Entwicklungsstand an Modellierungstechnik, wie er in den 90er-Jahren des 20. Jahrhunderts als Standard in den Konstruktionswerkzeugen der Automobilindustrie eingesetzt wurde. Zu diesem Zeitpunkt waren die wesentlichen Entwicklungen bei der internen Geometrirepräsentation bereits abgeschlossen, sodass sich mit allen vier CAD-Entwicklungsstufen 3D-Geometrie in Form von Kanten-, Flächen- und Volumenkörpern

erzeugen lässt [Men99]. Daher gibt es bei den Entwicklungsstufen zwar Unterschiede in der Struktur und dem Aufbau eines CAD-Modells, die Gestalt der resultierenden Bauteilgeometrie bleibt jedoch in der Regel unverändert und hängt bei komplexen Bauteilgeometrien von den Fähigkeiten des verwendeten Programms ab. So erweist sich beispielsweise die Modellierung von Flächen mit Selbstdurchdringungen genauso wie das Erstellen von Offsetflächen in CAD-Systemen weiterhin als problematisch. Diese Bereiche sind daher nach wie vor Teil von Forschungsarbeiten wie exemplarisch in [PLH⁺05, SFJ06, PEK08].

3.3.2 Konventionelles CAD

Beim konventionellen CAD wird die Strategie einer direkten Geometriemodellierung verfolgt. Dabei werden Geometrielemente bei der Erzeugung anhand von festen Werten definiert. Es besteht keine dauerhafte Abhängigkeit zwischen den einzelnen Geometrieobjekten, sondern jedes Element stellt eine isolierte Einheit dar. Darüber hinaus wird eine explizite Verwendung von Parametern, mit denen die erzeugte Geometrie beschrieben werden kann, nicht unterstützt. Eine Änderung der Geometrie kann daher nicht durch eine Anpassung von Parameterwerten erfolgen, sondern nur mit einem direkten Eingriff in das Modell [VDI09].

Die Modellierungstechnik des konventionellen CAD besitzt den Vorteil, dass der Geometrieaufbau weitgehend unstrukturiert erfolgen kann und sich nicht an der resultierenden Gestalt des Objekts orientieren muss. Dadurch kann in einem frühen Entwicklungsstadium, in der die Produktgestalt noch unzureichend bekannt ist, ein schneller Entwurf erstellt werden, ohne dass auf einen korrekten Strukturaufbau geachtet werden muss und dadurch die Kreativität eingeschränkt wird. Aus diesem Grund kommt eine direkte Modellierung bei computerunterstütztem Design (CAS¹¹) häufig zur Anwendung. Jedoch erweisen sich diese Modelle im Bereich der Konstruktion oftmals als unübersichtlich und daher schwer nachvollziehbar. Deshalb eignen sich Geometriemodelle, die mit konventionellen CAD erstellt wurden, nur eingeschränkt für eine Bearbeitung von mehreren Personen im Sinne des *Collaborative Engineering*. Des Weiteren erfordern konstruktionsbedingte Änderungen im Geometriemodell einen hohen Arbeitsaufwand. So ziehen geringfügige Anpassungen mitunter eine Reihe von Änderungen nach sich, die aufgrund der mangelnden Verknüpfungen von Geometrieelementen manuell erfolgen müssen. Daher wurden konventionelle CAD-Programme bei Fahrzeugherstellern und Systemlieferanten sukzessiv durch Systeme ersetzt, die einen parametrisch-assoziativen Modellierungsansatz verfolgen.

3.3.3 Parametrisch-assoziatives CAD

Ein wesentliches Merkmal eines parametrisch-assoziativen CAD-Systems liegt in der Fähigkeit, variable Eingaben und Beziehungen zwischen Geometrieelementen zu erfassen und zu verarbeiten. Dies erfordert eine duale Repräsentation des CAD-Modells, wobei die primäre Struktur alle Parameter enthält, die für die Definition der einzelnen Geometrieobjekte notwendig sind, während die sekundäre Struktur der Ermittlung und Visualisierung der resultierenden Geometrie dient. Abb. 3.4 zeigt schematisch den Ablauf einer Geometrieerzeugung und die Trennung

¹¹Computer Aided Styling

der primären Parameter- von der sekundären Geometriestruktur. Ebenfalls deutlich erkennbar ist der sequentielle Ablauf der Modellierung, beginnend mit der Zuweisung der Parameterwerte, die für eine vollständige Definition eines Geometrieelements benötigt werden, über die Evaluierung der Eingaben bis hin zur internen Geometrieverarbeitung [Men99, For03].

Eine vollständige Beschreibung eines neuen Geometrieelements umfasst eine Vielzahl von Parametern, wobei die Mehrzahl der Attribute entweder einer programminternen Verarbeitung dienen oder Eigenschaften festlegen, die in der Anwendung nicht von vordergründigem Interesse sind, wie etwa Visualisierungseigenschaften. Gerade letztere werden vom CAD-System weitgehend mit Standardwerten befüllt, um den Benutzer zu entlasten. Diese Standardwerte können nachträglich geändert werden, wohingegen programminterne Attribute zumeist nicht editierbar sind. Diese programminternen Parameter umfassen mitunter auch Parameter, welche entscheidend für die Form der Geometrie sind, allerdings dem Benutzer nicht zugänglich gemacht werden. Dies spielt insbesondere bei Geometrieelementen eine Rolle, die aus einem externen System importiert werden. Aus diesem Grund können etwa Freiformflächen, die in CAS-Programmen erstellt wurden, zwar dargestellt und weiterverarbeitet werden, jedoch lassen sich die zu Grunde liegenden Geometrieparameter und damit die Form vom Benutzer in der Regel nicht mehr editieren.

Ein Charakteristikum von parametrisch-assoziativen CAD besteht in der Möglichkeit, Geometrieelemente neben variablen Parameterwerten ebenfalls durch Verknüpfungen mit bereits bestehenden Geometrieobjekten zu definieren. Dieser Leitgedanke des *Associative Design* führt zu umfangreichen Beziehungen, die sowohl innerhalb eines Bauteils als auch bauteilübergreifend in einer Produktumgebung erstellt werden können. Dabei ergibt sich entsprechend der Verknüpfungsrichtung ein Geflecht aus sog. *Eltern-Kind*-Beziehungen, die zu einer komplexen Modellstruktur führen [For03].

Neben den assoziativen *Eltern-Kind*-Beziehungen können Geometrieobjekte auch durch Zwangsbedingungen (*Constraints*) in einen Kontext zueinander gebracht werden. So können Objekte beispielsweise parallel, koaxial oder unter einem vorgegebenen Winkel zueinander positioniert werden. Der Einsatz von *Constraints* hat sich vor allem bei zweidimensionalen Skizzen und bei der Anordnung von Bauteilen in einer Produktumgebung bewährt. Die festgelegten Zwangsbedingungen führen zu nichtlinearen algebraischen Gleichungssystemen, deren Lösung die Anordnung der Geometrieobjekte beschreibt. Zur Berechnung der Lösungen wird in der Regel eine Kombination von Algorithmen und Strategien herangezogen, die zumeist auf numerische Lösungsverfahren sowie die Graphentheorie setzen. Zwar sind diese Verfahren weit vorangeschritten, dennoch können sich selbst wohlbestimmte Systeme derart komplex erweisen, dass auch moderne CAD-Systeme eine Lösung nicht ermitteln können [BR98b, Hof05].

Wie in Abb. 3.4 dargestellt, erfordern die assoziativen Verknüpfungen und die Zwangsbedingungen in der Primärstruktur eine umfangreiche Konsistenzprüfung des Modells. Im Falle einer konsistenten Eingabe werden die Parameterwerte und Bindungen herangezogen, um mit internen Lösungsverfahren das parametrisierte Modell zu berechnen und der eigentlichen Geometrierzeugung zugänglich zu machen. Dieser Berechnungsdurchlauf ist sowohl nach der initialen Definition eines Geometrieelements als auch nach jeder Änderung von Parameterwerten oder

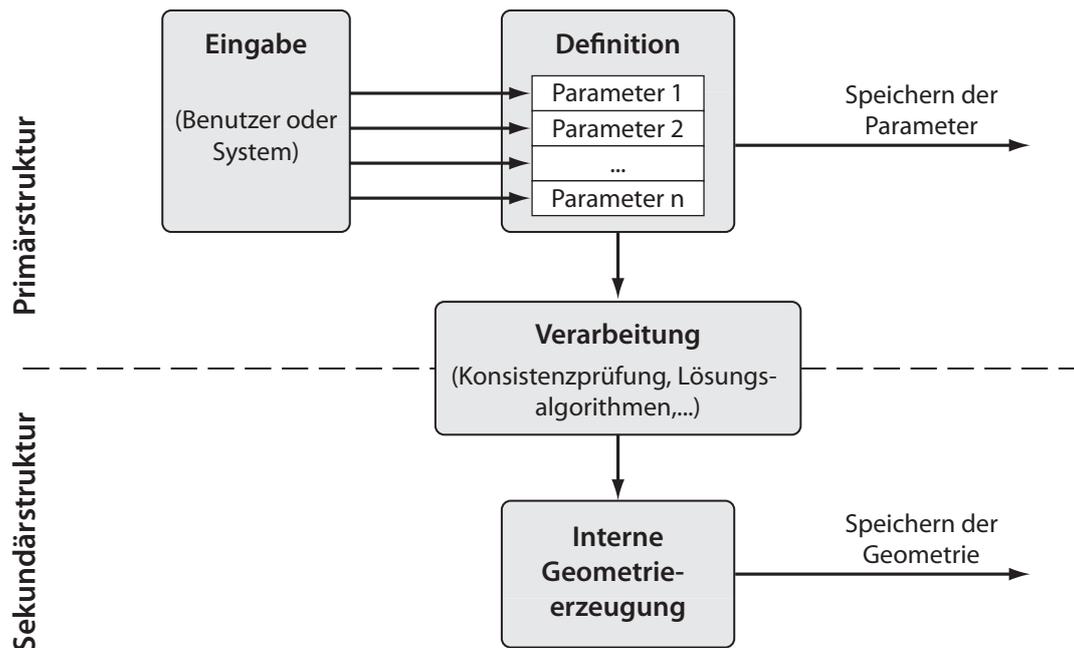


Abbildung 3.4: Duale Repräsentation eines parametrisch-assoziativen CAD-Modells, nach [AH94].

Verknüpfungen notwendig. Anhand des neu berechneten Modells lässt sich mit dem programm-internen Geometriemodellierer eine resultierende Geometrie erzeugen. Hierfür haben sich vor allem hybride Modellierer durchgesetzt, die sowohl Begrenzungselemente wie Flächen oder Kurven ermitteln als auch Volumenkörper mit Boole'schen Operatoren verknüpfen (*Boundary Representation* bzw. *Constructive Solids Geometry*) [Men99].

Die Eingaben der Primärstruktur wie auch die Ergebnisse der Geometriemodellierung der Sekundärstruktur werden dem Benutzer visualisiert und zur Verfügung gestellt. Während es sich bei der Sekundärstruktur um die resultierende Geometrie handelt, kommen bei der Anzeige der Primärstruktur unterschiedliche Strategien zur Anwendung [Men99, VDI09]. So kann mit einer Modellierungschronologie die zeitliche oder logische Abfolge der Modellierungsschritte zusammengefasst werden. Demgegenüber bildet eine Erzeugnisstruktur eine hierarchische Organisation der einzelnen Objekte. Zusätzlich können die Verknüpfungen der assoziativen *Eltern-Kind*-Beziehungen gekennzeichnet und ausgegeben werden. Während sich die Endgeometrie der Sekundärstruktur auch bei der Verwendung von verschiedenen CAD-Programmen gleich und zwischen den Systemen mit neutralen Datenformaten ausgetauscht werden kann, so haben sich abhängig vom jeweiligen Softwarehersteller unterschiedliche Repräsentationsformen der Primärstruktur entwickelt. Diese sind unter anderem von Programmierstrategien bestimmt und lassen sich selbst bei Verwendung von genormten Datenformaten, die auch die Modellstruktur berücksichtigen, nur eingeschränkt zwischen unterschiedlichen CAD-Systemen austauschen [MBW⁺11].

Ein wesentlicher Vorteil von parametrisch-assoziativen CAD-Systemen gegenüber konventionellen liegt in der durchgängigen Assoziativität der Objekte, die zu dauerhaften Abhängig-

keiten zwischen den einzelnen Geometrieelementen führt. Durch diese Bindungen ist es möglich, den Modellaufbau nachvollziehbar und leicht änderbar zu gestalten. Des Weiteren lassen sich durch Variation von Parameterwerten und den Austausch von Referenzgeometrien parametrisch-assoziative Konstruktionen effizient ändern. Dies erweist sich gerade bei Anpassungs- und Variantenkonstruktionen als ein entscheidender Vorteil gegenüber einer direkten Modellierungstechnik. Jedoch erfordern die verzweigten Abhängigkeiten eine methodische Vorgehensweise bei der Bauteilmodellierung, da anderenfalls selbst bei einfachen Bauteilen eine unübersichtliche und komplexe Modellstruktur entstehen kann. Das gleiche gilt für umfangreiche Baugruppen, wie sie in der Fahrzeugentwicklung üblich sind, da die Abhängigkeiten über Bauteilgrenzen hinweg ein weitreichendes Geflecht von Verknüpfungen erzeugen können [VDI09].

Gerade dieser Umstand erschwert die Migration von einem konventionellen CAD-Programm zu einem parametrisch-assoziativen und erfordert firmenspezifische Standards, Erweiterungen im Datenmanagement sowie einen erhöhten Aufwand für Anwenderschulungen. Aus diesem Grund vollzog sich der Umstieg bei Fahrzeugherstellern und Systemlieferanten trotz der systembedingten Vorteile über mehrere Jahre hinweg [Kir08]. Parallel zur steigenden Verbreitung von parametrisch-assoziativen CAD-Programmen wurden in umfangreichen Arbeiten Methoden und Strategien für eine strukturierte Geometriemodellierung erstellt [Men99, For03], die sich in Kombination mit Erkenntnissen aus der industriellen Praxis zu einer Reihe von Leitfäden und standardisierten Vorgehensweisen etabliert haben [Bra09, Bri06, Has06].

3.3.4 Featurebasiertes CAD

Der Begriff des *Features* wird in unterschiedlichen Bereichen der CAx-Technologien verwendet und bezeichnet ein Funktions- oder Konstruktionselement. Im Gebiet der computerunterstützten Konstruktion hat sich die Definition der *VDI-Richtlinie 2218* durchgesetzt [For03], die ein *Feature* als Aggregation aus Geometrieelementen und/oder Semantik bezeichnet. Dabei wird unter Semantik die interpretierbare Bedeutung des *Features* für die jeweilige Phase im Produktlebenszyklus sowie die damit verbundenen Daten und Informationen verstanden [VDI03, VDI09]. Durch die und/oder-Kombination ergibt sich eine Aufteilung in *Features*, die rein aus Geometrieelementen bestehen, sog. *Form-Features*, *Features* ohne Geometrie und *Features*, die Geometrie und Semantik kombinieren, sog. *Semantik-Features*. Die Verwendung von *Features* reicht von der Geometriemodellierung über die Fertigungs- und Produktionsplanung, die Qualitätssicherung, die Kostenermittlung hin zur Simulation und deckt alle Bereiche ab, die Produktinformationen benötigen. Eine umfassende Auflistung von Arbeiten, die unterschiedliche Einsatzgebiete von *Features* behandeln, wird beispielsweise durch *Bossmann* gegeben [Bos07].

Form-Features haben eine wesentliche Bedeutung für die Geometriemodellierung. Sie bilden eine geschlossene Einheit von Geometrieelementen und vereinfachen die Geometrieberechnung der Sekundärstruktur genauso wie auch die Repräsentation der Primärstruktur eines CAD-Systems. In der Erzeugnisstruktur erscheinen sie gewöhnlicherweise als einzelnes Element, das zwar editiert und verknüpft werden kann, jedoch besteht üblicherweise keine Möglichkeit, auf

die inkludierten Geometrieobjekte direkt zuzugreifen. Eine Veränderung der Gestalt erfolgt für gewöhnlich mit der Parametrierung und den Verknüpfungen des *Features*. Die Entwicklung von *Form-Features* erfolgte parallel zur Entwicklung von parametrisch-assoziativen CAD-Systemen, sodass diese bereits *Form-Features* beinhalten. Typische Beispiele dafür bestehen in Löchern, Nuten, Rippen, Fasen oder Freistichen. Allerdings haben diese den Charakter von Geometrieobjekten, während Funktionselemente innerhalb von featurebasiertem CAD vor allem dazu dienen, in den Produktmodellen zusätzliche Informationen einzubetten, die dann in der weiteren Produktentstehung verarbeitet werden können.

Diese Produktinformationen können laut der Definition in der *VDI-Richtlinie 2218* in Kombination mit Geometrieinformation oder auch separat innerhalb von *Features* abgelegt werden. Für letzteres finden sich typische CAX-Einsatzgebiete unter anderem in der Berechnung und der Fertigungsplanung. Im Bereich der Konstruktion sind diese Features derzeit von untergeordneter Bedeutung, sie kommen jedoch im Ansatz der *Assembly Features* zur Anwendung. Dabei handelt es sich um Informationsträger, die bauteilübergreifende Verbindungsinformationen zur Weiterverarbeitung bereitstellen. Mit diesen Funktionselementen soll die Möglichkeit geschaffen werden, Bauteilverbindungen, wie etwa Verschraubungen, in einer Baugruppe effizient zu koppeln [Vie05].

Ein zentrales Element einer featurebasierten CAD-Modellierung besteht in *Semantik-Features*, die sowohl Geometrie als auch Produktinformationen umfassen. Damit lassen sich jedem Geometrieelement Informationen zuordnen, die für die weitere Produktentstehung von Bedeutung sind. Die *Feature-Technologie* unterstützt somit die Vision eines integrierten Produktdatenmodells, das alle Produktinformationen beinhaltet [SK97]. Die hinterlegten Informationen lassen sich dann mit standardisierten Schnittstellen auslesen und weiter verarbeiten. Zwar können die Attribute der *Features* standardmäßig von CAD-Systemen vorgegeben werden, allerdings sind diese oftmals abhängig von der weiteren Verwendung in der Produktentstehung, wie auch von innerbetrieblichen Rahmenbedingungen. Moderne CAD-Systeme bieten daher die Möglichkeit benutzerdefinierte *Features* (UDF¹²) zu erstellen, die dann entsprechend den jeweiligen Anforderungen definiert, angepasst und in Bibliotheken gespeichert werden können. Aufgrund dieser Funktionalitäten können UDFs als ein Bindeglied zu einer wissensbasierten Konstruktion angesehen werden.

Der wesentliche Vorteil des featurebasierten CAD liegt in der Möglichkeit, dem Geometriemodell weitere Informationen hinzuzufügen und dieses damit zu einem Informationsträger in der Prozesskette zu machen. Jedoch stellt eine durchgängige featurebasierte CAD-Modellierung hohe Anforderungen an die Prozessschritte. So bietet ein featurebasiertes CAD-Modell nur dann Vorteile, wenn die integrierte Information in weiterer Folge ausgelesen und verarbeitet werden kann. Dies erfordert allerdings einen hohen Aufwand bei der Erstellung und Einführung einer featurebasierten Konstruktion sowie bei deren Administration. Aus diesem Grund hat eine umfassende featurebasierte Modellierung, die über eine Verwendung von *Form-Features* hinausgeht, nur in Teilbereichen der Fahrzeugtechnik Einzug gehalten. Ein Anwendungsbeispiel umfasst etwa die computerunterstützte Fertigung von spanabhebend bearbeiteten Bauteilen,

¹²User Defined Feature

die beispielsweise für Tiefziehwerkzeuge oder Motoren verwendet werden [Pri06, Bos07]. Ein weiteres Anwendungsgebiet ist die Modellierung von Karosseriebauteilen unter dem Gesichtspunkt einer anschließenden Produktion [Mba08].

3.3.5 Wissensbasiertes CAD

Wissensbasierte CAD-Programme verfügen über Berechnungsroutinen, Programmierumgebungen und -schnittstellen sowie etliche weitere Funktionalitäten, mit denen Aufgaben abseits der Geometriemodellierung oder der Dokumentation von Produkteigenschaften übernommen werden können. Dadurch lassen sich vielfach Fähigkeiten vereinigen, die anderenfalls eine Vielzahl an separaten Programmen erforderlich machen. Ein wesentlicher Bereich des bereitgestellten Leistungsumfangs zielt darauf ab, KBE-Anwendungen vollständig innerhalb der CAD-Umgebung zu entwickeln und in diese einzubetten. Somit bietet wissensbasiertes CAD die Möglichkeit für eine vollständige Integration von KBE- und CAD-Systemen, Abb. 3.2. Aufgrund dieser Funktionalitäten erhalten wissensbasierte CAD-Programme neben der Aufgabe als Konstruktionswerkzeug ebenfalls die Rolle als Entwicklungsumgebung für die Erstellung von KBE-Applikationen. Abb. 3.5 zeigt die wesentlichen Erweiterungen von wissensbasiertem CAD im Vergleich zu featurebasiertem CAD und ordnet diese entsprechend ihrem vordergründigen Einsatzgebiet –entweder um konstruktive Tätigkeiten zu unterstützen oder um KBE-Anwendungen zu erstellen.

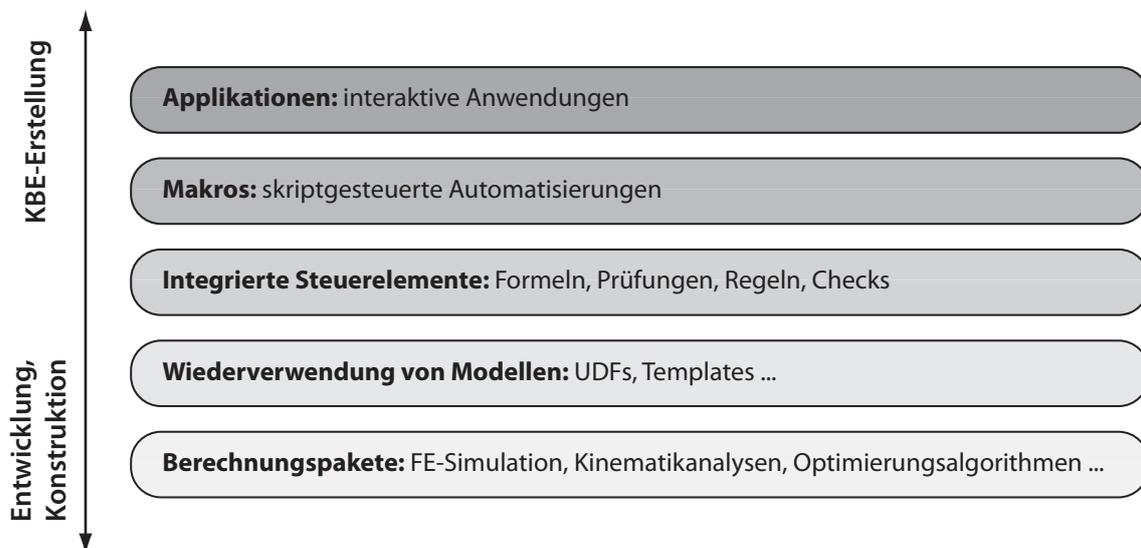


Abbildung 3.5: Funktionalitäten innerhalb des wissensbasierten CAD.

Eine Erweiterung, welche vor allem die Produktentwicklung unterstützen soll, liegt in der Implementierung von zusätzlichen Programmpaketen, mit denen direkt innerhalb der CAD-Umgebung Berechnungen und Simulationen erstellt werden können. Dabei lässt sich eine Fülle an unterschiedlichen Einsatzgebieten abdecken, wobei der Umfang an implementierten Programmpaketen stark vom jeweiligen CAD-Programm abhängig ist. So umfasst eine mögliche Erweiterung Kinematiksimulationen, die der Analyse von Bewegungsabläufen dienen und mit

denen beispielsweise Einbauuntersuchungen durchgeführt werden können. Des Weiteren enthalten CAD-Systeme vielfach virtuelle Modelle von Menschen, die für Ergonomie- und Komfortuntersuchungen herangezogen werden können. Eine häufig implementierte Erweiterung umfasst Berechnungsprogramme, die auf der *Methode der Finiten Elemente* (FEM¹³) beruhen. Damit lassen sich Aussagen über die Festigkeit und das Schwingungsverhalten von Bauteilen und Baugruppen treffen. Außerdem besteht die Möglichkeit, thermische Aspekte, wie etwa die Wärmeausdehnung, zu untersuchen. Ein weiterer Bereich umfasst Programmpakete für Strömungssimulationen, mit denen das Verhalten von Fluiden abgebildet werden kann. Diese Umfänge lassen sich beispielsweise für eine Entwicklung von Rohrgeometrien einsetzen, wie sie etwa in der Fahrzeugklimatisierung verwendet werden. Ebenfalls enthalten sind Optimierungsroutinen, mit denen die Geometrie entsprechend vorgegebener Randbedingungen und Optimierungskriterien verbessert werden kann. Neben den Simulationsmöglichkeiten können wissensbasierte CAD-Systeme ebenfalls Werkzeuge beinhalten, mit denen Prozessschritte, die einer Konstruktion nachgeschaltet sind, unterstützt werden können. So lässt sich innerhalb der CAD-Umgebung etwa der Maschinencode für eine automatisierte mechanische Bearbeitung erstellen oder ein Fabriklayout planen. Dadurch werden Aufgaben einer computerunterstützten Fertigungsentwicklung (CAM¹⁴) übernommen.

Die Erweiterungen von CAD-Systemen hinsichtlich Berechnungs- und Simulationspaketen liefern kein unmittelbares Unterscheidungsmerkmal für wissensbasiertes CAD. Bereits in konventionellen CAD-Systemen finden sich etwa FEM- oder CAM-Applikationen. Dennoch kann dieser vielseitige und umfangreiche Einsatz von zusätzlichen Programmpaketen als ein Spezifikum von wissensbasiertem CAD angesehen werden, da das CAD-System den Charakter eines zentralen Entwicklungswerkzeugs erhält.

Ein wichtiger Faktor, um bereits erworbenes Wissen zu konservieren, besteht in der Fähigkeit, dieses anwenderorientiert aufzubereiten, es zu verwalten und dem Benutzer leicht zugänglich zu machen. Gerade hierfür stellen wissensbasierte CAD-Systeme eine Reihe von Möglichkeiten zur Verfügung, mit denen bestehende Modelle in Vorlagen abgelegt und wiederverwendet werden können. Die Bandbreite reicht, angefangen von einzelnen Geometrieobjekten, die zu UDFs zusammengefasst werden können, über Teilkonstruktionen bis hin zu gesamten Bauteilen und vorgefertigten Produktstrukturen. Durch einen umfassenden Einsatz dieser *Templates* kann bei einer Neukonstruktion auf einen Baukasten von bewährten Lösungen und Methoden zurückgegriffen werden, wodurch sich der Erstellungsaufwand drastisch reduzieren lässt. Im Gegensatz zu parametrisch-assoziativem oder featurebasiertem CAD können die Vorlagen eines wissensbasierten CAD-Systems neben der Geometrie und den Produktdaten eine Reihe von Funktionen enthalten, mit denen Produkteigenschaften überprüft und automatisiert geändert werden können.

Es handelt sich hierbei im Wesentlichen um *Formeln*, *Prüfungen*, *Regeln*, und *Reaktionen*. Mit diesen Steuerelementen lässt sich eine Konstruktionsvorlage aufbereiten, sodass sich die Geometrie entsprechend der aktuellen Konstruktionssituation anpasst und Schlussfolgerungen abgeleitet werden können. Durch den Einsatz von *Formeln* lassen sich die Parameter der Primärstruk-

¹³Finite-Elemente-Methode

¹⁴Computer Aided Manufacturing

tur mathematisch miteinander verknüpfen. Hierzu steht zumeist eine Bibliothek an gängigen Operationen und Funktionen zur Verfügung. Des Weiteren können Produkteigenschaften mit Hilfe von *Prüfungen* kontrolliert werden. Damit lassen sich Eigenschaften, die außerhalb von vorgegebenen Grenzen liegen, ausgeben und auf diese Weise das Erfüllen von Anforderungen an das resultierende Produkt überwachen. Darüber hinaus kann mit *Regeln* und *Reaktionen* die Konstruktion situationsabhängig adaptiert werden. Während sich *Regeln* mittels Gleichungen und Ungleichungen auf vorgegebene Referenzwerte beziehen, berücksichtigen *Reaktionen* Ereignisse und Benutzereingaben, wie etwa die Änderungen eines Parameterwertes oder das Einfügen eines neuen Geometrieobjekts. Ein Beispiel für den Einsatz von *Regeln* besteht etwa in der automatisierten Änderung einer Welle zu einer Hohlwelle, sobald ein Zielgewicht überschritten wird. *Reaktionen* erweisen sich als weitaus komplexer, da sie interaktive Benutzereingaben berücksichtigen können. So lässt sich etwa eine Warnung ausgeben, sollte der Anwender eine Bohrung in einem unzulässigen Bereich des Bauteils einfügen.

Mit den angeführten Steuerelementen lassen sich eine Reihe von wissensbasierten Aufgaben abdecken. Dennoch sind die Möglichkeiten zur Erstellung von wissensbasierten Vorlagen begrenzt und bei komplexen Aufgaben neigt ein intensiver Einsatz von Steuerelementen zu Unübersichtlichkeit. Einen weitaus größeren Umfang an Fähigkeiten liefern in den wissensbasierten CAD-Systemen eingebettete Programmierumgebungen. Damit können etwa wiederkehrende Konstruktionsabläufe gespeichert und im Bedarfsfall abgerufen werden, sodass die hinterlegten Arbeitsschritte automatisiert umgesetzt werden. Mit Hilfe dieser sog. *Makros* kann der Konstruktionsprozess entlastet werden, da der Anwender von monotonen, automatisierbaren Aufgaben entlastet wird und im Idealfall mehr Zeit für kreative, wertschöpfende Tätigkeiten erhält. Neben den *Makros*, die festgelegten Programmsequenzen entsprechen, lassen sich in den Programmierumgebungen auch interaktive Applikationen erstellen oder mit Hilfe von Schnittstellen (API¹⁵) auf externe Softwarepakete zugreifen. Dadurch lassen sich Assistenzsysteme entwickeln, die sowohl auf die internen Leistungsumfänge des CAD-Systems zurückgreifen als auch externe Software-Umfänge einbinden und damit optimal auf die jeweiligen Bedürfnisse abgestimmt werden können.

Mit wissensbasierten CAD-Systemen stehen demnach vielseitige Entwicklungsplattformen für die Erstellung von KBE-Anwendungen zur Verfügung, die eine Fülle an Funktionalitäten liefern. Eine der größten Herausforderungen beim Einsatz liegt jedoch in der Komplexität der Modelle, die sich beim Ausschöpfen der gebotenen Möglichkeiten ergibt. So neigen wissensbasierte Modelle oftmals zu Unübersichtlichkeit und erweisen sich daher als fehleranfällig und schwer wartbar. Wie auch generell bei der Entwicklung von KBE-Anwendungen gilt es abzuwägen, ob die Erstellung einen wirtschaftlichen Nutzen mit sich bringt und sich demnach der Einsatz von wissensbasiertem CAD als sinnvoll erweist.

3.3.6 Trends

Mit wissensbasierten CAD-Systemen steht in der Produktentwicklung ein mächtiges Werkzeug zur Verfügung, das ständigen Verbesserungen und Erweiterung unterworfen ist. Laut *Hirz* ha-

¹⁵Application Programming Interface

ben sich bei Automobilherstellern derzeit drei Anbieter von CAD-Software etabliert, die den Markt von wissensbasierten CAD-Systemen beherrschen [Hir11]¹⁶. Betrachtet man die Entwicklung ihrer CAD-Softwareprodukte in den letzten Jahren, so lassen sich Trends erkennen und für eine zukünftige Entwicklung von CAD-Systemen extrapolieren. Dabei kann es sich jedoch lediglich um einen repräsentativen Ausschnitt von markanten Strömungen handeln, der keinerlei Anspruch auf Vollständigkeit erhebt.

Einer der Trends besteht in der weiteren Integration von zusätzlichen Berechnungs- und Simulationsumfängen, um bestehende Grenzen zu erweitern und zusätzliche Aufgabengebiete abzudecken. So beschränken sich die derzeit implementierten FEM-Pakete oftmals auf ein linear-elastisches Werkstoffverhalten oder der Wechsel zwischen expliziten und impliziten Integrationsverfahren ist nicht möglich. Durch den Wegfall dieser Grenzen können innerhalb der CAD-Umgebung weitaus komplexere Analysen erstellt werden, mit dem Vorteil, dass Schnittstellen zu externen Berechnungsumgebungen entfallen. Darüber hinaus geht der Trend zu einer Integration von mechatronischen Simulationsumfängen, mit denen unter anderem regelungstechnische oder hydraulische Aufgabenstellungen untersucht werden können. Dadurch verschwimmen die Grenzen zwischen Konstruktions- und Berechnungswerkzeug bzw. Synthese- und Analyseumgebung zunehmend. In gleicher Weise werden vermehrt benachbarte Aufgabengebiete, die abseits einer reinen Konstruktionstätigkeit liegen, in die CAD-Umgebung eingebunden. So werden beispielsweise Programmpakete für ein computerunterstütztes Design genauso wie Umfänge für betriebswirtschaftliche Belange in CAD-Programme implementiert. Zusätzlich werden Schnittstellen zu bereits etablierten Softwarepaketen geschaffen und es wird versucht, diese in einem gemeinsamen Layout darzustellen [Sch10b, Ray09].

Eine herausragende Stellung hat hierbei die Zusammenführung von CAD- und PDM-Systemen. Diese Verknüpfung ist derzeit weit vorangeschritten und der Trend geht dazu, dass CAD-Systeme in einem steigenden Ausmaß mit PDM-Systemen vernetzt werden. Dies kann soweit gehen, dass CAD-Programme zwingend ein spezielles PDM-System erfordern und es zu einer Verschmelzung von CAD- und PDM-Umfängen kommt. Aus diesem Grund bieten einerseits Hersteller von CAD-Systemen im zunehmenden Ausmaß PDM-Lösungen an und andererseits tritt bei der Entscheidung der Kunden über den Einsatz eines CAD-Produkts die PDM-Anbindung in den Vordergrund [SW11, Sch10c]. Neben der engen Koppelung von CAD- und Datenmanagement-Systemen kommt es auch zu einer verstärkten Einbindung des Internets. Darunter fällt die globale Verfügbarkeit von Daten im Rahmen des *Collaborative Engineering* auch auf mobilen Endgeräten genauso wie eine vernetzte Nutzung von EDV-Ressourcen (*Cloud Computing*) oder eine Vereinfachung von Administrationstätigkeiten durch automatische Softwareaktualisierungen [Ray09, Sch09, TM10].

Ein weiterer Trend geht zu einer zunehmenden Verwendung von direkter Modellierungstechnik. Dabei wird das Ziel verfolgt, in einer frühen Entwicklungsphase das schnelle und intuitive Skizzieren von Ideen zu ermöglichen. Darüber hinaus sollen innovative Bedienkonzepte eingesetzt

¹⁶Bei den angeführten Firmen handelt es sich um:

Dassault Systèmes, Vélizy-Villacoublay	www.3ds.com
Siemens PLM Software, Plano	www.plm.automation.siemens.com
Parametric Technology, Needham	www.ptc.com

werden, mit denen auch ungeübte Nutzer, die mit einem strukturierten Modellaufbau nicht vertraut sind, rasch Konzepte erstellen und vorhandene Geometrien adaptieren können. Gerade hierbei sollen Berührungsbildschirme oder alternative Eingabegeräte zum Einsatz kommen. Da bei einer direkten Modellierung die Modellierungshistorie keinen Einfluss auf das Ergebnis hat, ist in diesem Zusammenhang auch der Begriff *History free CAD* gebräuchlich. Da jedoch parametrisch-assoziative bzw. featurebasierte Modellierung in der Produktentstehung wesentliche Vorteile besitzt, geht der Trend zu hybriden Systemen, bei denen die verschiedenen Modellierungstechniken kombiniert werden und ein Wechsel zwischen den jeweiligen Modellierungstechniken möglich ist. Unabhängig davon wird versucht, die Bedienbarkeit intuitiver und dadurch angenehmer zu machen. Dies führt zu verbesserten Menüführungen und automatisierten Konstruktionsvorschlägen [Ray09, TM10, CKMH12].

Parallel zu den Weiterentwicklungen der CAD-Programme kommt es auch zu einer ständigen Verbesserung der Hardware. Neben der Leistungssteigerungen bei einzelnen Komponenten, wie etwa bei Grafikkarten, finden auch neue Technologien Anwendung. Während sich in der Vergangenheit im Bereich der Ein- und Ausgabegeräte Standard-EDV-Komponenten gegenüber speziell angepassten Entwicklungen [Sut63], durchgesetzt haben, stehen derzeit neuartige Technologien zur Verfügung, die das Potential für eine Erleichterung der Konstruktionsarbeit besitzen. Dazu zählt unter anderem die 3D-Visualisierung, mit der eine virtuelle realitätsnahe Abbildung (VR¹⁷) von Objekten ohne aufwändige Projektoren möglich ist. Des Weiteren können mit 3D-Druckern *Rapid Prototyping* Bauteile rasch und zu vertretbaren Kosten erstellt werden [Ray09, Sch09, TM10].

Ein Schwachstelle bei den unterschiedlichen CAD-Systemen liegt in der fehlenden Interoperabilität der Systeme. Zwar stehen genormte Datenformate zur Verfügung, jedoch bilden diese in der Regel nicht die vollständige Information ab oder es kommt zu Datenverlusten bei der Konvertierung der unterschiedlichen Datenformate. Auch wenn es viele Bestrebungen und Anstrengungen in der Industrie gibt, einheitliche, neutrale und standardisierte Datenformate zu verwenden, so zeichnet sich derzeit kein umfassender und generell akzeptierter Standard ab. Daher kann auch in Zukunft von einem heterogenen Datengefüge und den damit verbundenen Schnittstellenproblemen ausgegangen werden [TM10].

3.4 CAD-Systeme in einem PDM-unterstützten Entwicklungsprozess

Unabhängig davon, ob CAD-Systeme als Werkzeuge für die Bauteilkonstruktion oder für die Erstellung von KBE-Applikationen verwendet werden, lässt sich ihr Einsatz gerade in großen Unternehmen nicht losgelöst von einem PDM-System betrachten. Zum einen haben PDM-Systeme einen hohen Durchdringungsgrad im Fahrzeugbau und zum anderen werden sie dazu verwendet, Prozesse und Arbeitsabläufe in der Entwicklung und insbesondere in der Konstruktion festzulegen. So liefert ein PDM-unterstützter Entwicklungsprozess Vorgaben und Restriktionen an die Bauteil- und Baugruppenmodellierung, die unter anderem auf eine effiziente Verwaltung

¹⁷Virtual Reality

und eine durchgängige Strukturierung der Produktdaten abzielen. Diese Anforderungen führen zu einer starken Verflechtung zwischen CAD-Systemen, die in ihren Modellen zentrale Produktdaten in Form von Geometrie und Metadaten bereitstellen, und PDM-Programmen, die als übergeordnete Organisationsplattformen fungieren. Darüber hinaus wird das PDM-System als primäre Datenschnittstelle eingesetzt, um die verwalteten Produktdaten an die jeweiligen Folgeprozesse zu übergeben. Es kann somit als Datenrückgrat (*Data Backbone*), das für eine einheitliche und übergreifende Datenbasis innerhalb des Entwicklungsprozesses sorgt, verstanden werden.

Die unterschiedlichen Folgeprozesse sind jedoch nicht an den vollständigen Datensätzen interessiert, sondern lediglich an dem für sie relevanten Teil. Daher können mitunter reduzierte Modelle und angepasste Datensätze aus dem PDM-System abgerufen werden. Abb. 3.6 veranschaulicht diesen Umstand speziell für Geometriedaten, die in CAD-Modellen abgelegt sind. Gleiches gilt ebenfalls für die produkt- und prozessbeschreibenden Zusatzinformationen innerhalb der CAD-Dokumente, mit denen Eigenschaften wie das Material, die Bauteilnummer oder der Ersteller des Dokuments dokumentiert werden. Auch sie sind von unterschiedlichem Interesse für Folgeprozesse und können selektiv bereitgestellt werden. Aufgrund der Vielfältigkeit der verschiedenen Attribute geht Abb. 3.6 allerdings nicht weiter auf diese Metadaten ein.

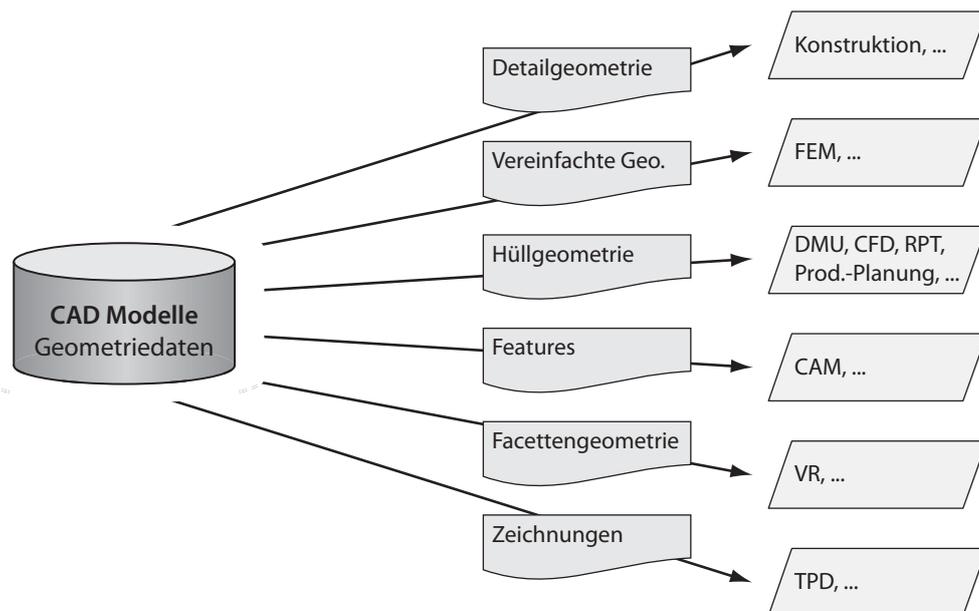


Abbildung 3.6: CAD-Geometrie als zentrale Geometrie-Datenbasis in der Produktentwicklung, nach [Sch02].

Während die Konstruktion zumeist auf detaillierte und vollständige Geometriedaten angewiesen ist und diese auch bearbeitet und verändert, beschränken sich andere CAx-Disziplinen oftmals auf Teilumfänge. So benutzen FEM-Simulationen zumeist bewusst vereinfachte Geometriedaten, da *Form-Features* wie etwa Fasen oder Verrundungen zu detailliert für ein Berechnungsnetz sind und daher deaktiviert werden. Andere Berechnungsdisziplinen, wie etwa

die Strömungssimulation (CFD¹⁸), benötigen keinen vollständigen Strukturaufbau des Geometriemodells, sondern lediglich die äußeren Begrenzungsflächen. Diese Hüllgeometrie genügt ebenfalls den Anforderungen aus DMU-Prozessen, dem *Rapid Prototyping* sowie der Produktionsplanung. In gleicher Weise von Interesse für die Produktion und insbesondere für CAM-Prozesse erweisen sich *Features*, da diese fertigungsrelevante Informationen enthalten können. Im Bereich der VR-Darstellung kommt oftmals eine tessellierte Facettengeometrie mit Grafikeigenschaften wie Texturen und Reflexionen zum Einsatz. Ein weiterer Teil der Geometriedaten besteht in Zeichnungen und Schnitten, die für eine technische Produktdokumentation (TPD¹⁹) benötigt werden.

Durch die unterschiedlichen Anforderungen der Folgeprozesse an den Modellaufbau und -inhalt eines CAD-Dokuments, werden die Daten mitunter in verschiedenen Ausprägungen parallel abgelegt. Aufgrund der bereits angeführten Problematik einer fehlenden Interoperabilität zwischen den CAx-Systemen kommen gerade hierzu neutrale, nicht-proprietäre Datenaustauschformate zur Anwendung. Diese haben den Vorteil, dass sie von einer Vielzahl an Systemen verarbeitet werden können und daher die Anzahl an unterschiedlichen Konvertierungen zwischen den einzelnen CAx-Systemen wesentlich minimieren [GA90]. Derzeit stehen diverse Formate zur Verfügung, darunter auch das STEP²⁰- oder das JT²¹-Format, die beide eine hohe Verbreitung in der Fahrzeugtechnik aufweisen [Int94, Int12]. Das STEP-Format verfolgt das Ziel, eine eindeutige Produktrepräsentation zu schaffen, mit deren Hilfe die Informationen eines gesamten Produktlebenszyklus abgelegt werden können [And93]. Das Format ist genormt und bietet die Möglichkeit, Produktdaten, wie etwa Produktstruktur, Materialdaten oder Toleranzen, aber auch funktionale Zusammenhänge, gesamtheitlich zu beschreiben. Demgegenüber handelt es sich beim JT-Format um einen Industriestandard, der seinen Fokus auf die Visualisierung von Geometriedaten legt. Es können allerdings auch Metadaten gespeichert werden. Der wesentliche Unterschied zum STEP-Format besteht in einer besseren Speicherausnutzung und in der Möglichkeit, eine bidirektionale Assoziativität der gespeicherten Elemente mit dem Ursprungsdocument darzustellen. Damit kann eine JT-Datei bei einer Änderung des Quellmodells leicht aktualisiert werden, was wiederum Vorteile bei der Verwaltung in einem PDM-System bringt [Ste07].

Neben der Datendokumentation und -bereitstellung werden PDM-Systeme zusätzlich als Werkzeuge für das Projektmanagement verwendet. So können die Anwender mit projektspezifischen Berechtigungen versehen werden und damit ihr Verantwortungsbereich abgegrenzt werden. Des Weiteren ermöglichen die verwalteten Daten Rückschlüsse auf das Erreichen von Zielvorgaben hinsichtlich Kosten, Terminen oder produktbezogenen Eigenschaften, wie dem Gewicht. Darüber hinaus können PDM-Systeme für eine Verwaltung von Varianten und Versionen herangezogen werden. Gerade an dieser Stelle stößt eine Datenorganisation, die über kein übergeordnetes Datenmanagement verfügt, an ihre Grenzen. So sind unterschiedliche Produktkonfigurationen, wie sie etwa bei verschiedenen Fahrzeugvarianten vorkommen, ohne PDM-System nur mit einem unverhältnismäßig großen Aufwand darstellbar. Gleiches gilt für die Versionierung, mit

¹⁸Computational Fluid Dynamics

¹⁹Technische Produktdokumentation

²⁰Standard for the Exchange of Product Model Data

²¹Jupiter Tessellation

deren Hilfe ständig aktuelle Datenstände zur Verfügung gestellt werden können. Außerdem lässt sich die Datenreife mit Hilfe von Attributen feststellen und dadurch der Freigabe- und Änderungsprozess unterstützen [Jun05, ES09].

Ein weiteres Gebiet des Datenmanagements liegt in der unterschiedlichen Strukturierung des Datenmaterials. So unterscheidet sich beispielsweise die Produktstruktur im CAD-System, die Interesse an der räumlichen Positionierung hat, von der Fertigungsstruktur, die beispielsweise eine Fügefolge beachtet. Gleiches gilt für die Stücklistenstruktur, deren Verwaltung eng mit dem PDM-System gekoppelt ist. Auch hier treten andere Aspekte, wie etwa das Variantenmanagement oder eine Fertigungsplanung, in den Vordergrund. Ungeachtet der unterschiedlichen Strukturierungen und trotz der großen Anzahl an Varianten sowie der unterschiedlichen Bauweisen vermeiden PDM-Systeme eine Redundanz im Datenmaterial. Demnach ist etwa jedes Bauteil eindeutig durch seine Bauteilnummer bestimmt und kann dadurch identifiziert sowie in weiterer Folge einer beliebigen Produktkonfiguration zugeordnet werden [Bur08].

Aufgrund der Vorteile einer leichten Beherrschbarkeit von großen und umfangreichen Datenmengen sowie einer leichten Zugänglichkeit von Produktinformationen haben sich PDM-Systeme insbesondere in großen Unternehmen durchgesetzt. Für die Modellierung von CAD-Daten bedeutet ein PDM-basierter Entwicklungsprozess eine verstärkte Berücksichtigung von Anforderungen und Vorgaben, die eine Archivierung und Organisation der Produktdaten ermöglichen und erleichtern. Darüber hinaus können insbesondere wissensbasierte Fähigkeiten von CAD-Systemen eingeschränkt sein, sofern es zu keiner engen Koppelung zwischen CAD- und PDM-System kommt. So erweisen sich beispielsweise Konstruktionstabellen oder bauteilübergreifende *Regeln* und *Reaktionen* aber auch *Constraints* zur Bauteilpositionierung als problematisch. Um diese Einschränkungen zu verkleinern, bietet sich eine sorgfältige Abstimmung zwischen CAD- und PDM-Systemen an. Außerdem bilden derzeitige Lücken einen Hauptantrieb für eine zukünftige engere Verflechtung dieser Systeme.

3.5 Templates als Werkzeuge einer wissensbasierten Produktmodellierung

3.5.1 Überblick und Eingrenzung

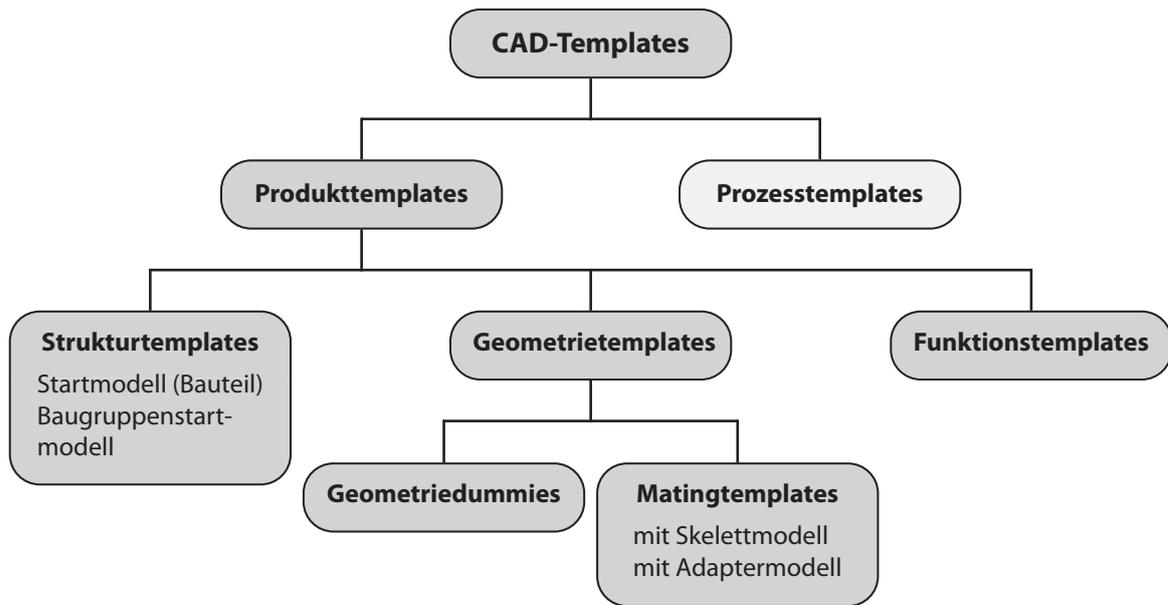
Wissensbasiertes CAD, als derzeitiger Stand der Technik im Bereich der CAD-Systeme, beinhaltet einerseits eine Vielzahl an Funktionalitäten, mit denen Entwicklungs- und Konstruktionsaufgaben übernommen werden können und dient andererseits als Plattform für die Erstellung von KBE-Applikationen. Aufgrund der Komplexität von wissensbasiertem CAD haben sich Methoden etabliert, die das Ziel verfolgen, die bereitgestellten Funktionalitäten besser nutzbar zu machen und in weiterer Folge für einen effizienteren Einsatz im Produktentstehungsprozess zu sorgen. Die Entwicklung dieser Methoden erfolgte parallel zur Entwicklung der eingesetzten CAD-Systeme aus Abb. 3.3. Dies hat als Konsequenz, dass die jeweiligen Strategien nicht zwingend nur für wissensbasierte CAD-Systeme gültig sind, sondern auch in vorangegangenen CAD-Entwicklungsstufen angewandt werden können. So konnten etwa wesentliche Fortschritte

in der Methodenentwicklung mit der Einführung von parametrisch-assoziativen CAD erreicht werden, da dieses ein strukturiertes Vorgehen bei der Modellierung erforderte und daher nach neuartigen Modellierungsmethoden verlangte [Men99, For03]. Die hierbei entwickelten Ansätze haben auch in einem wissensbasierten CAD ihre Gültigkeit und werden ständig angepasst und erweitert.

Die Methoden und Werkzeuge für eine wissensbasierte Produktmodellierung und für den Einsatz von wissensbasiertem CAD sind mannigfaltig und haben verschiedene Schwerpunkte. Beschränkt man sich auf Methoden, die im Rahmen des *Frontloading*-Ansatzes Expertenwissen in einer frühen Entwicklungsphase bereitstellen, so kommen in der Automobilindustrie vor allem Werkzeuge zum Einsatz, die auf *Templates* beruhen, wobei diese *Templates* durchaus auf *Features* zurückgreifen sowie mit Assistenzsystemen gekoppelt sein können [For03, Has05, HAR06, Pri06, HVA08, Mba08, Bro10]. Auch in dieser Arbeit wird der Fokus auf *Templates*-basierte Strategien gelegt, da diese sich gut für eine effiziente, wissensbasierte Produktmodellierung eignen und in weiterer Folge für eine Stärkung der frühen Phase sorgen.

Der Begriff des *Templates* kann als Vorlage übersetzt werden und findet neben einer computerunterstützten Konstruktion ebenfalls in anderen CAx-Domänen, wie auch generell in der Informatik, eine weite Verbreitung. Im Kontext von CAD wird in der vorliegenden Arbeit ein *Template* in Anlehnung an *Forsen* als vorgefertigte Modellstruktur oder Modellierungssequenz definiert, die in ihrer Gesamtheit oder in Teilen wiederverwendet werden kann [For03]. Des Weiteren werden CAD-basierte *Templates* in *Produkttemplates* und *Prozesstemplates* unterteilt, Abb. 3.7. Während *Produkttemplates* zur Erzeugung von Produktdaten zum Einsatz kommen, dienen *Prozesstemplates* zur Verbesserung von Abläufen und sind dabei unter anderem ein Werkzeug im Anforderungs- und Änderungsmanagement. Die vorliegende Arbeit konzentriert sich ausschließlich auf *Produkttemplates* und unterteilt diese –wie auch in Abb. 3.7 dargestellt– in weitere Unterkategorien.

Die Kategorie der *Strukturtemplates* fasst Vorlagen zusammen, welche die Anordnung von untergeordneten Elementen vorgeben und somit ein Gerüst für die Produktmodellierung bereitstellen. Typische Vertreter von *Strukturtemplates* bestehen auf Bauteilebene aus *Startmodellen*. Im Falle von Baugruppen, die bis zum Umfang eines Gesamtfahrzeugs reichen können, sind es *Baugruppenstartmodelle*. Im Gegensatz dazu dient die Gruppe der *Geometriemplates* vor allem einer Wiederverwendung von Geometriemodellen und Modellierungssequenzen. Diese Kategorie lässt sich weiter unterteilen in die Rubrik der *Geometriedummies* und der *Matingtemplates*. Erstere stellen Geometrien zur Verfügung, die weitgehend unverändert bleiben, insbesondere Hüllgeometrien von Übernahme- oder Normteilen. Im Gegensatz dazu beinhalten *Matingtemplates* adaptierbare und steuerbare Geometrievorlagen. Hierunter fallen *Templates* mit *Skelettmodellen*, die drahtgitterartige Geometrieelemente zur Modellsteuerung verwenden und *Templates* mit *Adaptermodellen*, die Bauteilgeometrien an Begrenzungs- und Anschlussgeometrien anpassen können. In einer weiteren Kategorie innerhalb der *Produkttemplates* decken *Funktionstemplates* die Aufgaben einer funktionalen Absicherung und Auslegung ab. Dabei greifen sie oftmals auf Steuerelemente wie etwa *Formeln* oder *Regeln* sowie Berechnungs- und Simulationsumfänge innerhalb der CAD-Umgebung zurück.

Abbildung 3.7: Kategorien von *CAD-Templates*.

Die Bezeichnungen und der Umfang der einzelnen Kategorien können in der Literatur durchaus variieren. So unterteilt etwa *Haasis* lediglich zwischen geometrienerzeugenden und funktionsabsichernden *Templates* und bezeichnet letztere als *Study-Templates*. Des Weiteren unterscheidet er die *Templates* entsprechend ihres Detaillierungsgrads und definiert demzufolge unterschiedliche Ausprägungen [HAR06]. Unabhängig davon steht mit der Verwendung von *Templates* ein mächtiges Werkzeug für eine effiziente Produktmodellierung zur Verfügung. In weiterer Folge wird hier auf die beschriebenen Templatekategorien aus Abb. 3.7 detaillierter eingegangen und es werden Strategien sowie Methoden für deren Einsatz aufgezeigt.

3.5.2 Strukturtemplates

Startmodell

Bei einem *Startmodell* handelt es sich um eine Vorlage, die auf Bauteilebene eine vorgegebene Struktur bereitstellt. Diese Bauteilstruktur legt die Anordnung der einzelnen Modellelemente fest und fußt üblicherweise auf betriebsinternen Konventionen zur Erstellung von CAD-Modellen. Dabei gehen diese Richtlinien in der Regel über die bloße Anordnung der Geometrieobjekte hinaus und legen beispielsweise graphische Attribute oder die Nomenklatur der einzelnen Elemente fest [Bro10]. Üblicherweise handelt es sich hierbei um stringente Vorgaben, die im Rahmen von automatisierten Modellprüfungen kontrolliert werden, um die Modellqualität sicherzustellen. Dadurch soll gewährleistet werden, dass die Bauteilmodelle für eine nachfolgende Verwendung, wie etwa eine Berechnung oder eine Fertigungsplanung, geeignet sind und nicht zusätzlich nachbearbeitet werden müssen. Darüber hinaus erleichtert ein durchgängiger einheitlicher Modellaufbau sowohl eine Bearbeitung durch mehrere Personen als

auch die Verwendung von Automatisierungsroutinen, die auf benötigte Informationen definiert zugreifen können [Ste07].

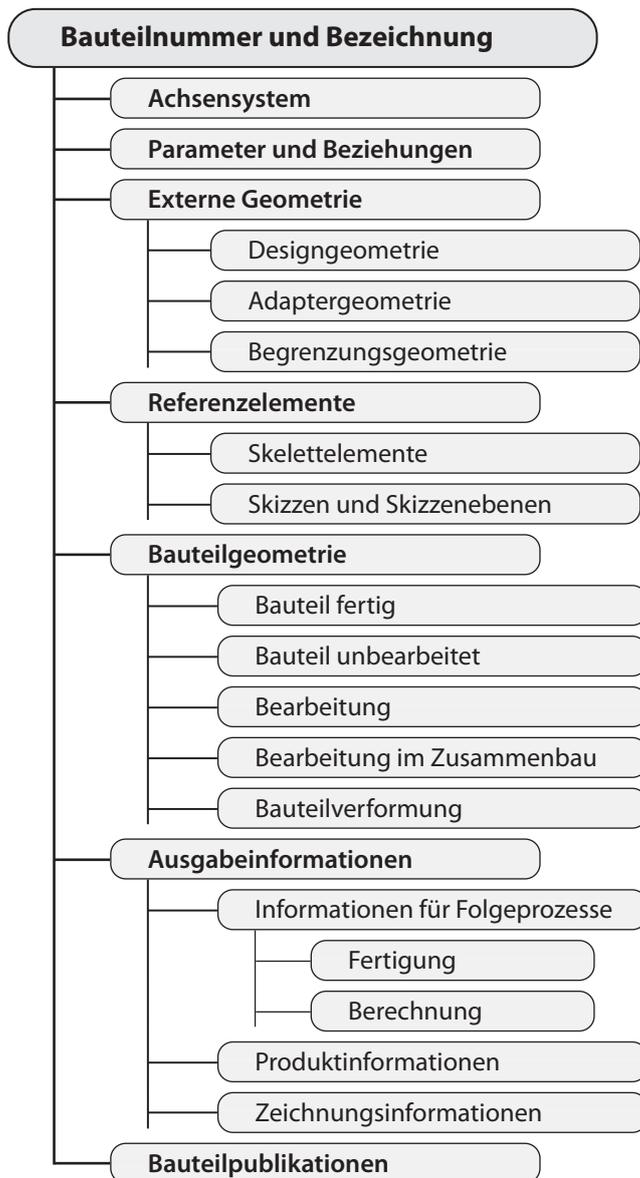


Abbildung 3.8: Typischer Aufbau eines Startmodells, nach [Bri06, HGL08].

Da *Startmodelle* die Konstruktions- und Modellierungsstrategien des jeweiligen Fahrzeugherstellers unterstützen sollen, haben sich in der Automobilindustrie aufgrund der unterschiedlichen innerbetrieblichen Rahmenbedingungen eine Vielzahl an verschiedenen *Startmodellen* etabliert. Wenngleich es Bestrebungen gibt, die *Startmodelle* wie auch die Konstruktionsmethodik zu vereinheitlichen und als Industriestandard festzulegen [KB10], gilt eine vollständige Harmonisierung aufgrund der heterogenen Systemlandschaft bei den verwendeten CAD-Systemen derzeit als unwahrscheinlich. Unabhängig von den jeweiligen firmenspezifischen Unterschieden lässt sich jedoch eine typische Grundstruktur eines Startmodells definieren. Abb. 3.8 zeigt beispielhaft den Aufbau eines *Startmodells* in Anlehnung an Brill und Hirz [Bri06, HGL08].

Wird ein neues Modell auf Basis des *Startmodells* erstellt, erhält es eine neue Bauteilnummer sowie eine unverwechselbare Bezeichnung. Mit dieser eindeutigen Deklaration kann es innerhalb eines PDM-gestützten Entwicklungsprozesses, wie er in der Automobilindustrie üblich ist, identifiziert und zugeordnet werden. Ebenfalls vorgegeben wird ein Achsensystem, das als Referenz dient und bei Fahrzeugen zumeist in der Mitte der vorderen Radaufhängung positioniert wird, um somit einem genormten Bezugssystem zu entsprechen [Int78]. Die Modellierung der einzelnen Komponenten erfolgt – sofern dies nicht innerhalb von Submodulen, wie etwa der Antriebs-einheit, gegenteilig definiert wird – in Konstruktionslage. Somit stimmt das jeweilige Achsen-system üblicherweise mit dem Fahrzeugreferenzsystem überein [HHR11]. Die Bauteilparameter und die resultierenden Beziehungen zwischen Parametern und Konstruktionselementen werden unter einer eigenen Rubrik abgelegt. In dieser Kategorie werden ebenfalls Standardparameter, die etwa Materialeigenschaften oder das Bauteilgewicht beschreiben, vorgegeben und können aufgrund der festgelegten Nomenklatur automatisiert ausgelesen werden. Die strukturierte und einheitliche Organisation der Parameter liefert darüber hinaus eine Grundvoraussetzung für eine durchgängige, parametrisierte Produktmodellierung [Böh04]. Des Weiteren können gerade in diesem Bereich des Startmodells Berechnungsroutinen und Automatismen integriert sein. So wird beispielsweise das Bauteilgewicht aufgrund der eingegebenen Materialparameter und der resultierenden Geometrie ermittelt.

Des Weiteren bietet das vorgestellte *Startmodell* eine Kategorie für externe Geometrielemente. Darunter fallen beispielsweise Designvorgaben, wie etwa die Fahrzeugaußenhaut, die zumeist in einer externen CAS-Umgebung erstellt wird und dann unter anderem als Vorgabe für die Entwicklung von Karosseriebauteilen dient. Ebenfalls enthalten sind Adapter- und Begren-zungselemente, mit denen ein definierter Bauraum festgelegt wird. Dies umfasst Geometrie von unmittelbar angrenzenden Teilen, wie etwa Flansche, aber auch Bauraum für bewegte Teile, wie etwa eine Heckklappe, genauso wie Produktionsmittel, wie beispielsweise Gehänge-aufnahmen für Karosseriebauteile. Eine weitere Klasse innerhalb des *Startmodells* wird durch Referenzelemente abgedeckt. Diese bestehen im wesentlichen aus Drahtgitterelementen sowie Skizzen mit den dazugehörigen Skizzenebenen und können für die Steuerung der Bauteilgeo-metrie verwendet werden.

Die bisher aufgezählten Klassen werden hauptsächlich als Eingangs- und Steuerelemente verwendet, während die eigentliche Bauteilgeometrie in einer eigenen Rubrik abgelegt wird. Diese enthält sowohl die resultierende Geometrie des fertig bearbeiteten als auch des unbearbeiteten Bauteils sowie Geometrielemente der Bearbeitung, wie etwa Beschnittflächen oder Löcher. Darüber hinaus kann hier weitere bearbeitungsrelevante Bauteilgeometrie angeordnet werden. Darunter fallen beispielsweise Geometrie für Bearbeitungsschritte, die in einem zusammenge-bauten Zustand erfolgen, oder verformte Bauteilgeometrie, wie sie etwa beim Tiefziehen mit einer starken Rückfederung vorkommt.

Für den Produktentstehungsprozess wichtige Kategorien im *Startmodell* umfassen die Ausga-beinformationen sowie die Bauteilpublikationen. Während letztere für die Repräsentation im DMU und für Bauteilverknüpfungen in der Produktstruktur von Interesse sind, liefert die Ru-brik Ausgabeinformationen einen Datencontainer für nachfolgende Prozessschritte. So können

hier beispielsweise berechnungs- oder produktionsrelevante Informationen abgelegt werden. Darunter fallen unter anderem Toleranzinformationen, Netze für eine FEM-Berechnung oder die Fügefolge bei geschweißten Bauteilen. Zusätzliche Produktinformationen, wie etwa Trägheitseigenschaften, sind in den Ausgabeinformationen ebenfalls enthalten wie Informationen für eine Zeichnungsableitung oder eine Bemaßung innerhalb des 3D-Modells. Dabei gewinnt gerade letztere immer mehr an Bedeutung, da im Bereich der Produktdokumentation der Trend weg von einer klassischen 2D-Zeichnung und hin zu einem 3D-Produktmodell geht, das ebenfalls alle Bemaßungsinformationen beinhaltet [QRP⁺10].

Die hier beschriebene Bauteilstruktur eines Startmodells kann prinzipiell für jedes Bauteil herangezogen werden. Dennoch kann es in der industriellen Praxis vorkommen, dass sich *Startmodelle* abhängig vom Bauteiltyp geringfügig voneinander unterscheiden. So können sowohl spezielle Startmodelle für flächenbasierte Tiefziehteile als auch für Gussteile, die sich hauptsächlich aus Volumenkörpern zusammensetzen, eingesetzt werden. Diese Trennung vermeidet, dass die Bauteilstruktur zu viele ungenutzte Einträge enthält und dadurch unübersichtlich wird. Die *Startmodelle* beschränken sich auf die Vorgabe einer allgemeingültigen Struktur und enthalten daher keine Geometrielemente. Ausnahmen bilden universelle Referenzelemente, wie etwa Geraden, welche die Entformrichtungen in tiefgezogenen Bauteilen repräsentieren, oder Flächen, die als Platzhalter die Werkzeugtrennung von Gussteilen berücksichtigen.

Das *Startmodell* bietet somit eine Strukturvorgabe, innerhalb welcher alle Geometrie- und Metadaten eines CAD-Bauteilmodells abgelegt werden können. Dadurch kann sichergestellt werden, dass Bauteilmodelle einheitlich, übersichtlich, nachvollziehbar und für Folgeprozesse leicht weiterverarbeitbar sind. Die Einhaltung der Vorgaben für das Befüllen des *Startmodells* sowie das Umsetzen der jeweiligen Konstruktionsrichtlinien ist allerdings in einer frühen Phase nur bedingt sinnvoll. So verhindert einerseits die Fülle an Vorgaben eine flexible Arbeitsweise und andererseits kann eine Vielzahl der vorgegebenen Attribute und Kategorien nicht befüllt werden, da die benötigten Informationen aufgrund des groben Detaillierungsgrads noch nicht vorhanden sind. Dies wird auch durch *Bergholz* bestätigt. Er zeigt, dass der Einsatz von Startmodellen erst bei komplexen Bauteilmodellen, die umfangreiche Informationen beinhalten, vorteilhaft ist [BS07]. Aus diesem Grund erfolgt die Konzeptarbeit oftmals in separaten Konzeptmodellen, deren Ergebnisse dann in einem seriennahen Entwicklungsprozess in *Startmodelle* übergeführt werden [Has05]. Bei dieser Ergebnistransformation kommt es mitunter zu einem Bruch in der Parametrik, sodass die Konzeptgeometrie zwar eine Vorlage liefert, die Konstruktion des Serienbauteils jedoch entkoppelt von der Konzeptkonstruktion erfolgt [GNR⁺08]. Unabhängig davon haben sich *Startmodelle* gerade in einer seriennahen Entwicklung als wichtiges Element einer wissensbasierten Konstruktion etabliert.

Baugruppenstartmodell

Die einzelnen Bauteilmodelle lassen sich innerhalb des CAD-Systems in Baugruppen zusammenfassen und in einer hierarchischen Produktstruktur anordnen. Die Produktstruktur wird zumeist in einer PDM-Umgebung festgelegt und ist eng verbunden mit dem Mengengerüst des Produkts, das unter anderem für die Kosten- und Gewichtsermittlung herangezogen wird. Die

Gliederung der Produktstruktur folgt üblicherweise dem funktionalen Aufbau des Produkts und orientiert sich an Grenzen innerhalb der Organisationsstruktur. So lässt sich ein Fahrzeugmodell in seiner ersten Unterebene beispielsweise in Bereiche wie Antriebsstrang, Karosserie oder Fahrwerk unterteilen [Böh04].

Im Gegensatz zu Bauteilmodellen enthalten Baugruppen in der Regel keine Geometrie, sondern umfassen hauptsächlich Verknüpfungs- und Lageinformationen der inkludierten Komponenten. Bei den Komponenten kann es sich sowohl um Bauteile als auch um Baugruppen handeln, die wiederum untergeordnete Komponenten enthalten. Dadurch entsteht eine verschachtelte Struktur, wie sie beispielhaft in Abb. 3.9 dargestellt wird. Bei der Positionierung der einzelnen Komponenten stehen unterschiedliche Strategien zur Auswahl [HHR11]. So lassen sich Unterbaugruppen und Bauteile mit *Constraints* miteinander verknüpfen und dadurch innerhalb einer übergeordneten Baugruppe gezielt zueinander positionieren. Bei dieser Methode eignet sich der Einsatz eines Skelettbauteils, das als zentrales Referenzelement dient, sodass Verknüpfungen nur zwischen den Komponenten und dem Skelettbauteil zulässig sind. Dadurch lassen sich Zirkelbezüge verhindern und die Berechnung der jeweiligen Positionen erfolgt wesentlich stabiler. In der Baugruppe der Abb. 3.9 sind entsprechende *Skelettmodelle* als eigene Kategorie enthalten. Die notwendigen *Constraints* sowie zusätzliche Parameter werden zentral im Bereich Parameter und Beziehungen abgelegt. Ein Vorteil bei der Verwendung von *Constraints* zum Positionieren von Komponenten liegt in der Möglichkeit, die Parameterwerte der *Constraints* zu variieren und dadurch beispielsweise die Kinematik einer Baugruppe zu untersuchen.

Die Rubrik der Parameter kann neben einer Bauteilpositionierung ebenfalls für eine baugruppenübergreifende Parametrierung genutzt werden. So können Parameterwerte ausgehend von einer übergeordneten Gliederungsebene hierarchisch an Unterbaugruppen und Bauteile weitergegeben werden [Böh04]. Dies schafft die Möglichkeit einer durchgängigen Parametrierung, die allerdings innerhalb einer Serienentwicklung als kritisch zu bewerten ist. So führt eine vollständige Parametrierung zu einer großen Anzahl an Parameterwerten und Abhängigkeiten, die über Verantwortungsbereiche hinweggehen und schwer handhabbar werden können.

Bei komplexen, umfangreichen und daher unter Mitarbeit von mehreren Personen entwickelten Produkten kommt oftmals eine Positionierungsstrategie zur Anwendung, bei der die Bauteile nicht mit *Constraints* miteinander verknüpft, sondern bereits in ihrer finalen Lage konstruiert werden. Dabei dienen angrenzende Bauteilgeometrien als externe Eingangsdaten für die Konstruktion eines neuen Bauteils, das dann bei Änderungen dieser Begrenzungs- und Adaptergeometrien ebenfalls adaptiert werden muss. Jedoch kann bei einer Bauteilerstellung in Konstruktionslage auf eine Positionierung innerhalb einer Baugruppe verzichtet werden, weil alle Bauteile sich üblicherweise auf ein globales Referenzkoordinatensystem beziehen und daher bereits die korrekte Position innehaben.

In der industriellen Praxis kommen, abhängig von der jeweiligen Aufgabenstellung, beide Methoden zur Anwendung. So findet sich bei Baugruppen mit einer tendenziell geringeren Komplexität und bei welchen das Interesse besteht, funktionale Aspekte durch eine Variation von Parameterwerten zu untersuchen, oftmals eine Positionierung mittels Skeletten. Ein typisches Beispiel hierfür liefert der Ventiltrieb eines Verbrennungsmotors oder die Radaufhängung eines

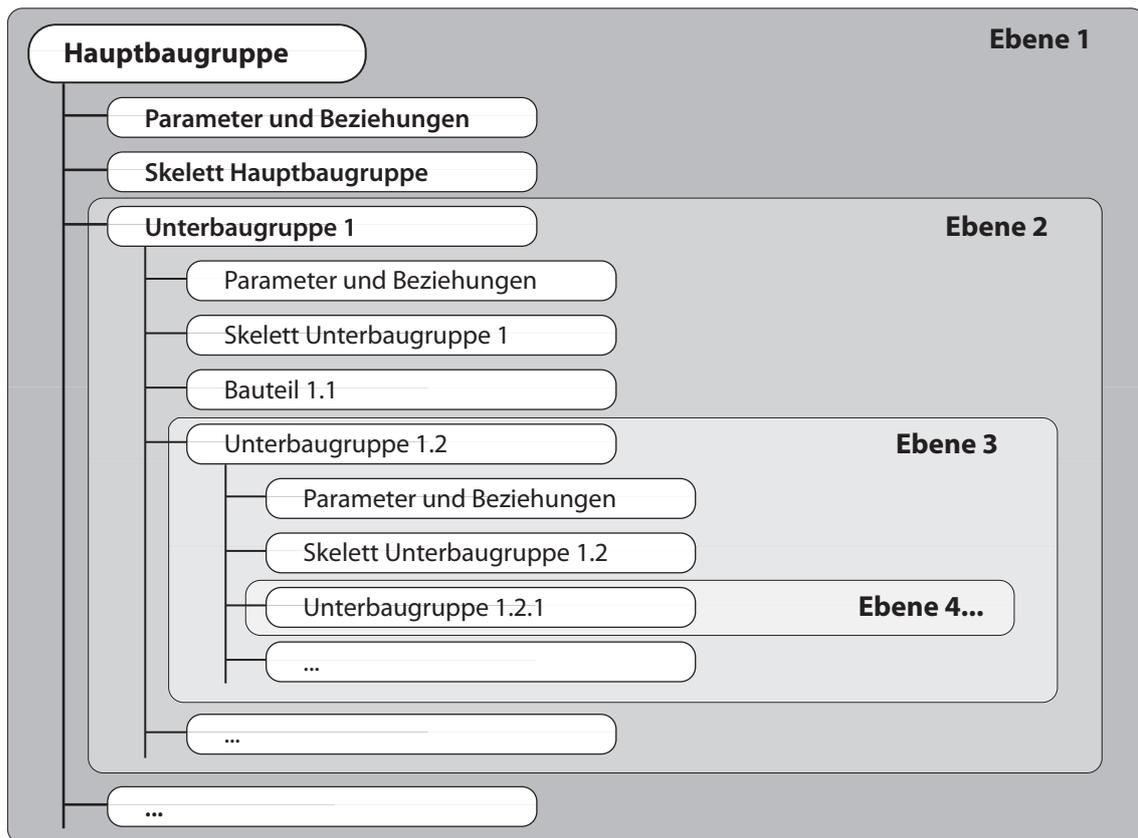


Abbildung 3.9: Typischer Aufbau einer Baugruppenstruktur, nach [HHR11].

Fahrwerks. Bei umfangreichen Baugruppen bietet eine fixe Positionierung, gerade auch wegen einer Vereinfachung von DMU- oder PDM-Prozessen, Vorteile. Darunter fallen beispielsweise Baugruppen wie die Karosserie oder Interieurumfänge sowie Konstruktionsumfänge in einer seriennahen Entwicklung.

Aufgrund der Tatsache, dass sich die Struktur von Fahrzeugen aber auch von fahrzeugtypischen Komponenten, wie etwa einer Antriebseinheit, oftmals nur geringfügig ändert, lässt sich vergleichbar mit einem *Startmodell* eine vorgegebene definierte Produktstruktur festlegen. Diese orientiert sich vor allem an der Organisationsstruktur des Unternehmens, da üblicherweise Grenzen innerhalb von Baugruppen gleichzeitig auch mit den organisationsbedingten Verantwortungs- und Kompetenzbereichen korrelieren. So steht am Start einer seriennahen Entwicklung in der Regel eine vorgegebene Produktstruktur in einem *Baugruppenstartmodell* zur Verfügung, bei dem Platzhalter für die einzelnen Unterbaugruppen und Einzelteile vorgesehen sind. Dieses *Baugruppenstartmodell* kann dann mit den Ergebnissen einer frühen Entwicklungsphase befüllt werden. Dabei besteht eine enge Verknüpfung mit PDM- und DMU-Prozessen, wie etwa der Stücklistenstellung oder einem Management von Produktvarianten. Ein *Baugruppenstartmodell* für eine seriennahe Entwicklung verzichtet allerdings oftmals bewusst auf skelettartige Steuerelemente und -bauteile sowie die darauf fußenden assoziativen Verknüpfungen, weil gerade die Gesamtfahrzeugentwicklung in dieser Phase stark von PDM-Prozessen

sowie Anforderungen des Änderungsmanagements und der Produktion beeinflusst wird. Hierbei erweisen sich weitreichende, baugruppenübergreifende Verknüpfungen des öfteren als störend und als organisatorisch schwer handhabbar.

Des Weiteren lassen sich *Baugruppenstartmodelle* auch innerhalb einer frühen Entwicklungsphase einsetzen. Allerdings orientieren sich diese konzeptionellen *Baugruppenstartmodelle* an den hier vorherrschenden Herausforderungen wie einer Vielzahl an unterschiedlichen Varianten und sich schnell ändernden Arbeitsständen. Daher wird häufig auf eine Positionierung der einzelnen Komponenten mit Hilfe von skelettartigen Steuermodellen zurückgegriffen, mit denen sich relativ einfach Parametervariationen oder Variantenstudien erstellen lassen. Hierdurch enthalten konzeptionelle *Baugruppenstartmodelle* eine definierte Produktstruktur, aber auch produktspezifische Steuerelemente, wie beispielsweise bei einem Verbrennungsmotor die Kurbelwellen- und die Zylinderachsen. An dieser Stelle kommt es zu Berührungspunkten mit der Kategorie der *Matingtemplates*, die ebenfalls Steuerelemente enthalten. Der wesentliche Unterschied zwischen diesen Kategorien liegt darin, dass ein *Baugruppenstartmodell* zwar skelettartige Steuerbauteile enthalten kann, allerdings keine Bauteilgeometrie. Aufgrund dessen beschränken sich *Baugruppenstartmodelle* auf eine Strukturvorgabe und dazugehörige Steuerelemente, während *Matingtemplates* zusätzlich auch Bauteilgeometrie beinhalten.

3.5.3 Geometrietemplates: Geometriedummies

Innerhalb der Kategorie der *Geometrietemplates* fassen *Geometriedummies* alle Modelle zusammen, deren Geometrie nicht adaptier- oder steuerbar ist, aber dennoch dem *Template*-Ansatz folgend in unterschiedlichen Entwicklungsprojekten wiederverwendet wird. Diese Bauteile werden in der Regel unverändert und ohne zusätzlichen Modellierungsaufwand in eine Baugruppenstruktur eingefügt und in weiterer Folge lediglich positioniert. Daher eignen sich im Besonderen Übernahmeteile für den Einsatz als Komponenten einer Modellwiederverwendung als *Geometriedummies*. Da die Geometrie dieser Bauteile nicht verändert werden soll, ist es auch nicht notwendig, dass die Bauteilmodelle eine vollständige Information bezüglich der Geometriemodellierung enthalten. Daher liefern die Modelle mitunter keinerlei Informationen über den zu Grunde liegenden Modellaufbau, sondern beinhalten lediglich die äußeren Konturen der Komponenten. Diese Hüllgeometrie genügt jedoch für eine geometrische Integration innerhalb eines DMUs ebenso wie für Einbauuntersuchungen oder etwa die Konstruktion von Montagehilfsmitteln. Ein weiterer Grund für die Verwendung von Hüllgeometrien liegt im Schutz von betriebsinternem Wissen. So stellen gerade externe Lieferanten und Entwicklungspartner von Submodulen, wie beispielsweise von Getrieben oder von Fahrwerksdämpfern, häufig nur die Außenkonturen bzw. eine vereinfachte Geometrie der Komponenten zur Verfügung.

Ein weitere Anwendung liefern Normteile, die ebenfalls einzeln als *Geometriedummy* gespeichert werden können. Da sich gewisse Gruppen von Normteilen, wie Schrauben oder Muttern, nur geringfügig voneinander unterscheiden, besteht für die Ablage von Normteilen ebenfalls die Option eines parametrisierten *Masterbauteils*. Dabei handelt es sich um ein vollständig parametrisiertes Bauteil, das sich anhand von Eingabeparametern relativ einfach steuern lässt.

Dadurch kann mit einem einzigen Bauteilmodell eine breite Palette an Varianten eines Normteils durch Ändern der hinterlegten Parameterwerte dargestellt werden. Aufgrund dieser Eigenschaften handelt es sich bei einem *Masterbauteil* allerdings nicht um einen Vertreter der *Geometriedummies*. Für die Verwaltung und die Bereitstellung von *Masterbauteilen* stehen unter anderem systeminterne Bibliotheksfunktionen von CAD-Programmen zur Verfügung. Jedoch eignen sich diese Kataloge nur bedingt für einen PDM-gestützten Entwicklungsprozess, der eine redundante Ablage von Bauteilen verhindert und Attribute, wie beispielsweise eine eindeutige Bauteilnummer benötigt. Eine weitere Möglichkeit für die Bereitstellung von Norm- und Übernahmeteilen liefern Internetplattformen. In diesem Rahmen kann zumeist in umfangreichen Katalogen von potentiellen Lieferanten recherchiert und eine passende Komponente samt CAD-Modell ausgewählt werden. Nichtsdestotrotz müssen auch in diesem Fall die Bauteilmodelle vorangehend in ein PDM-System integriert werden.

Ein weiteres Gebiet für die Anwendung von *Geometriedummies* stellt die Schnittgeometrie von Referenzbauteilen dar. Für diese Vorlagen werden charakteristische Schnitte von Bauteilen aus Vorgänger- und Konkurrenzmodellen herangezogen, die hauptsächlich in einer frühen Entwicklungsphase als Eingangsinformation für die Konstruktion von neuen Bauteilen dienen. Hierbei handelt es sich oftmals um Schnitte an markanten Stellen von Bauteilen und -gruppen. Da die extrahierten Schnitte nicht zwangsläufig mit den Schnittgeometrien übereinstimmen, die für die Geometrieerstellung verwendet werden, sind die extrahierten Schnitte mitunter nicht parametrisiert. Trotzdem können diese *Schnitt-Geometriedummies* als Grundlage für eine weitere Konstruktion verwendet werden – entweder um direkt Geometrie darauf aufzubauen oder um sie als Wissensträger und damit als Vorbild für neue Bauteile einzusetzen, ohne dass die eingefügte Schnittgeometrie für den Modellaufbau herangezogen wird. Gerade in der Karosseriekonstruktion findet diese Vorgangsweise häufige Anwendung [Has05, Bra10].

Der Einsatz von *Geometriedummies* führt gerade bei Norm- und Übernahmeteilen zu einer Modularisierung der Konstruktion, wodurch bei einer weitreichenden Umsetzung auf einen Baukasten aus Geometriemodellen zurückgegriffen werden kann. Ein wichtiger Gesichtspunkt beim Einsatz dieser Modelle liegt allerdings in der Organisation der verfügbaren Modelle, sodass diese schnell gefunden und zielgerichtet eingesetzt werden können. Hierfür eignen sich neben den Möglichkeiten des PDM-Systems ebenfalls datenbankbasierte Informationsportale. Ein weiterer Aspekt für eine erfolgreiche Verwendung von *Geometriedummies* liegt in der Aufbereitung der Modelle. Da die Modelle in der Regel nicht verändert werden, bietet es sich an, die Daten in neutralen Datenformaten abzulegen. Dadurch können Inkompatibilitäten aufgrund von Versionsänderungen des verwendeten CAD-Programms oder eines Wechsels des CAD-Systems vermieden werden.

3.5.4 Geometrietemplates: Matingtemplates

Aufbau und Eigenschaften von Matingtemplates

Matingtemplates umfassen – gleich wie *Geometriedummies* – Vorlagen für Modellierungssequenzen wie auch für vollständige Geometriemodelle. Der wesentliche Unterschied zwischen diesen

Kategorien liegt im Ziel der *Matingtemplates*, die Gestalt der *Template*-Geometrie aufgrund von äußeren Eingaben automatisch anzupassen. Diese Eingaben sind im Allgemeinen Verknüpfungen zu Geometrieelementen sowie Parameterwerte, die beide innerhalb der *Matingtemplates* mittels Hilfsmodellen für eine Geometriesteuerung und -adaption herangezogen werden. *Matingtemplates* enthalten daher sowohl Hilfsmodelle als auch eine darauf aufbauende, anpassungsfähige Geometrie. Eine Herausforderung beim Einsatz von *Matingtemplates* liegt darin, für ein möglichst breites Feld an potentiellen Eingaben gültige Ergebnisse zu liefern und damit Fehler in der Modellkonsistenz zu verhindern. Nur dadurch kann sichergestellt werden, dass die *Templates* ohne umfangreiche Nacharbeiten initialisiert oder in ein vorhandenes Modell eingebunden und adaptiert werden können.

Um dieser Anforderung zu genügen, bietet es sich an, die *Templates* stringent entsprechend dem EVA- bzw. IPO-Prinzip²² aufzubauen. Dieses Prinzip hat seinen Ursprung in der Software-Entwicklung und liefert ein durchgängiges Schema für eine Programmstruktur. Hierbei wird ein Programm in drei Bereiche aufgeteilt: die Eingabe (Input), die Verarbeitung (Process) und die Ausgabe (Output). Dieses Prinzip wird im Rahmen einer modularen Programmierung in allen Programmebenen angewandt, sodass Unterprozeduren, die im Verarbeitungsbereich aufgerufen werden, ebenfalls entsprechend der EVA-Systematik aufgebaut sind. Hierdurch wird eine durchgehende Struktur geschaffen, deren Vorteile in der leichten Austauschbarkeit der jeweiligen Module liegen. Der Ansatz lässt sich auch in den Bereich der CAD-Modellierung übertragen und liefert eine durchgängige, transparente und vor allem modulare Modellgliederung. Daher liegt es nahe, diesen Aufbau nicht nur für *Matingtemplates* sondern generell für CAD-Modelle anzuwenden, die durch diese methodische Vorgehensweise den gestalterischen Grundregeln einer eindeutigen, einfachen und sicheren Modellierung genügen können. Aus diesem Grund sehen beispielsweise die Bauteil- und Baugruppenvorlagen der *Strukturtemplates* eine Gliederung vor, bei der Ein- und Ausgaben sowie die eigentliche Geometriemodellierung in getrennten Rubriken abgelegt werden. *Strukturtemplates* selbst enthalten keine adaptierbare Geometrie, jedoch kann mit dieser Gliederung die Geometriemodellierung entsprechend dem EVA-Schemas unterstützt werden [Men99, For03].

Abb. 3.10 zeigt schematisch den Aufbau einer EVA-Struktur für ein *Matingtemplate* auf Bauteilebene und greift bei der Darstellung auf eine systemtechnische Betrachtungsweise zurück. Unter diesen Gesichtspunkten analysiert *Forsen* die methodischen Vorgehensweisen bei der Konstruktion mit einem parametrisch-assoziativen CAD-System und geht dabei insbesondere auf die Gliederung von adaptierbaren Geometrievorlagen ein [For03]. So lässt sich ein *Matingtemplate* als System interpretieren, das über seine Systemgrenzen hinaus durch Ein- und Ausgabereaktionen in Beziehung mit seiner Umwelt steht. Im Fall eines Bauteil-*Templates* besteht die Umwelt üblicherweise aus einer CAD-Baugruppe, in der die Vorlage eingebettet ist. Bei den Ein- und Ausgabereaktionen handelt es sich – wie eingangs beschrieben – in der Regel um Verknüpfungen zu Geometrieobjekten und Parameterwerten.

Sowohl die Ein- als auch die Ausgaben werden entsprechend dem EVA-Schema zentral in einer Ein- bzw. Ausgabeebene zusammengefasst. Dort werden die Informationen in schnittstellenar-

²²Eingabe-Verarbeitung-Ausgabe bzw. Input-Process-Output Prinzip

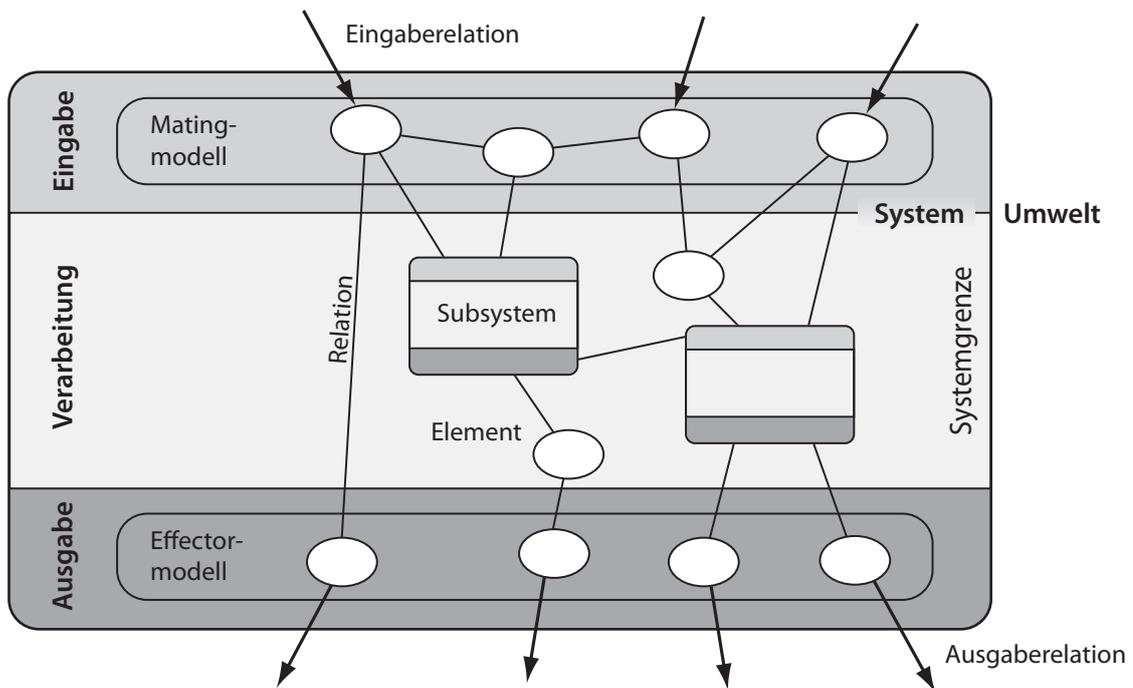


Abbildung 3.10: Schematischer Aufbau eines *Matingtemplates* aus einer systemtechnischen Betrachtungsweise, nach [For03].

tigen Hilfsmodellen für eine weitere Verarbeitung aufbereitet. *Forsen* bezeichnet diese schnittstellenartigen Hilfsmodelle für Eingaben als *Mating-* und für Ausgaben als *Effectormodelle*. Innerhalb der *Matingmodelle* werden einerseits die Eingaben für eine übergeordnete Geometrie-adaption aufbereitet und andererseits enthalten *Matingmodelle* skelettartige Führungselemente für eine zentrale Geometriesteuerung. Darüber hinaus sind üblicherweise Dummyeingaben enthalten, um das System vollständig oder teilweise entkoppelt von seiner Umwelt betrachten zu können. Dies kann notwendig sein, wenn – wie es in einer frühen Entwicklungsphase oftmals der Fall ist – nicht alle erforderlichen Eingaben bekannt sind. Die Dummyelemente schließen ebenfalls Automatismen ein, mit denen anhand einer bekannten Eingabe auf eine weitere, noch nicht vorhandene Eingabe geschlossen werden kann. So können beispielsweise zwei Längen in ein vorgegebenes Verhältnis zueinander angenommen werden, solange nur eine der beiden bekannt ist [Men99, For03].

Die eigentliche Bearbeitung der Eingaben erfolgt innerhalb der Verarbeitungsebene des Systems. Hier kommt es im Fall eines Bauteil-*Templates* zur Geometrieerzeugung des gewünschten Bauteils. Dabei bietet es sich an, charakteristische Geometriesequenzen einer durchgängigen Gliederung folgend ebenfalls mit einer Eingabe-, Verarbeitungs- und Ausgabeebene zu strukturieren. In Abb. 3.10 sind diese Geometriesequenzen als Subsysteme dargestellt, während einzelne Geometrieoperationen als Elemente symbolisiert werden. Die Verbindung zwischen den Elementen und den Subsystemen erfolgt über Relationen, die im wesentlichen assoziative Verknüpfungen darstellen. Die finale Geometrie wird, wie weitere für die Umwelt interessante Objekte, gesammelt im *Effectormodell* abgelegt. Demzufolge dient der *Effector* als zentraler Datencontainer, der neben Parametern hauptsächlich Geometrieelemente enthält.

In der vorliegenden Arbeit wird unter dem Begriff des *Matingtemplates* eine vollständige Vorlage für adaptierbare Geometrie, bestehend aus Eingabe-, Verarbeitungs- und Ausgabebene verstanden. Damit enthält ein *Matingtemplate* alle Elemente, die in Abb. 3.10 innerhalb der Systemgrenzen liegen. Demgegenüber umfasst ein *Matingmodell* lediglich ein schnittstellenartiges Hilfsmodell in der Eingabebene, das als Bindeglied zur Systemumwelt dient, und somit ein Bestandteil eines übergeordneten *Matingtemplates* ist. Des Weiteren lässt sich die Kategorie des *Matingmodells* abhängig von den Verknüpfungen mit der Systemumwelt –also abhängig davon, wie das übergeordnete *Matingtemplate* in die Systemumwelt eingebunden ist– in ein *Adapter-* oder in ein *Skelettmodell* unterteilen.

Adapter- und Skelettmodelle

Adaptermodelle werden vornehmlich als Bindeglied zwischen Bauteilen verwendet, die direkt aneinander grenzen und zumeist auch eine funktionale Schnittstelle besitzen. So kann beispielsweise ein *Adaptermodell* zur Koppelung eines Motor- und eines Getriebemodells dienen, die aneinander geflanscht sind. Demgegenüber sorgt ein *Skelettmodell* für die innere Systemstruktur eines übergeordneten Modells und kann als Gerüst für die enthaltenen Elemente verstanden werden. Ein Beispiel dafür liefert das Skelett eines Verbrennungsmotors, an dem sich alle Komponenten, wie etwa der Kurbel- oder ein Ventiltrieb, orientieren und welches die Vorgabe für die Konstruktion dieser Komponenten liefert [Bos07].

Diese Unterschiede in der Anwendung führen zu einer grundsätzlich verschiedenen Einbindung der *Adapter-* und *Skelettmodelle* innerhalb einer übergeordneten Gliederungsebene. Während *Adaptermodelle* als Schnittstelle für Eingaben einer gleichen wie auch einer übergeordneten Ebene dienen, verarbeiten *Skelettmodelle* ausschließlich Eingaben einer übergeordneten Ebene. Demnach stellt ein *Skelettmodell* einen Spezialfall eines *Adaptermodells* dar. Abb. 3.11 veranschaulicht diese Unterschiede und zeigt für den jeweiligen Anwendungsfall eine typische Baugruppenstruktur mit zwei Untererebenen, die allesamt entsprechend des EVA-Ansatzes aufgebaut sind. Auch die Schnittstellen der jeweiligen Bauteile sind durchgängig entweder als Adapter oder als Skelett aufgebaut. Zwischen den einzelnen Komponenten sind ebenfalls jene Verknüpfungen zu Geometrieobjekten und Parameterwerten eingezeichnet, die ausschließlich für eine Anpassung der Geometrie sorgen. Auf Parametrierungsstrategien für Produkteigenschaften, die etwa Material oder den Fertigungsprozess beschreiben, wird hier nicht weiter eingegangen. In Abb. 3.11 ist ersichtlich, dass die Verknüpfungen innerhalb einer Struktur von *Skelettmodellen* streng hierarchisch aufgebaut sind. So enthält ein *Skelettmodell* einer übergeordneten Ebene alle notwendigen Eingaben für die untergeordneten *Skelettmodelle*. Damit stellen *Skelettmodelle* ein wirksames Hilfsmittel für eine erzeugnisorientierte (*Top-down*) Modellierungsstrategie bereit.

Bei dieser Modellierungsstrategie folgt die Produktstruktur einer methodischen Vorgehensweise bei der „vom Groben ins Feine“ zuerst ein Konzept erstellt wird, das dann weiter entworfen, ausgearbeitet und detailliert wird [VDI93, VDI09]. So werden ausgehend von einer groben Funktionsstruktur, die sich in einem Hauptskelett abbilden lässt, erst nach und nach die Details innerhalb von Unterbaugruppen und Bauteilen erstellt. Diese Reihenfolge findet sich im

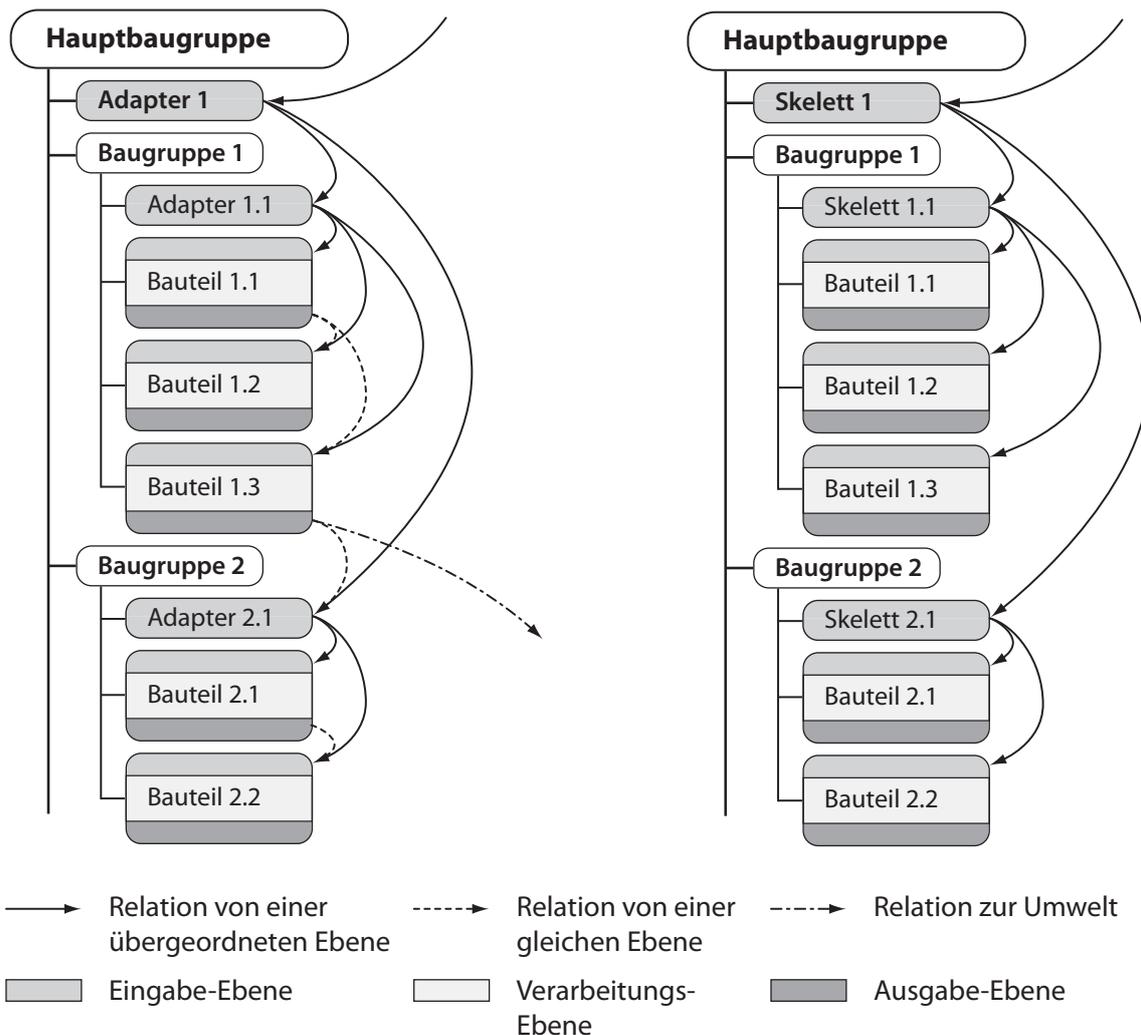


Abbildung 3.11: Verknüpfungen, die für eine Geometrieadaptation innerhalb eines *Matingtemplates* verwendet werden (links *Adaptermodelle*, rechts *Skelettmodelle*), nach [Bra09].

Wesentlichen auch in einem typischen Entwicklungsprozess eines Fahrzeuges. Eine Grundvoraussetzung für die Anwendung einer *Top-down*-Modellierungsstrategie liegt allerdings in der genauen Kenntnis der zu Grunde liegenden Produktarchitektur, was bei einem Fahrzeug und dessen Komponenten zumeist der Fall ist [VDI09]. Der entscheidende Vorteil einer *Top-down*-Modellierung, die durch *Skelettmodelle* unterstützt wird, liegt in einer Durchgängigkeit der Verknüpfungen, die eine gute Übersichtlichkeit und Nachvollziehbarkeit gewährleistet. Jedoch kommt es bei einem zunehmenden Detaillierungsgrad und umfangreichen parallelen Arbeiten an unterschiedlichen Bauteilen zumeist zu einer Aufteilung der Konstruktion in Teilbereiche [Vic06]. Dadurch wird die Entwicklung eines Bauteils oftmals durch die parallele Konstruktion von angrenzenden Komponenten beeinflusst. Dies führt an dieser Stelle zu einer Kombination aus einer ergebnisorientierten und einer bauteilorientierten Modellierungsstrategie.

Die bauteilorientierte oder *Bottom-up*-Modellierungsstrategie ist als Gegenstück eines *Top-down*-Ansatzes anzusehen. Bei der *Bottom-up*-Modellierungsstrategie liegen Bauteilmodelle vor, die dann zu einem Produkt zusammengefügt oder in ein solches integriert werden. Diese Modelle können beim Vorhandensein von Übernahmeteilen selbst in einer frühen Entwicklungsphase schon in einer detaillierten Form vorliegen. Die Entwicklung ist somit getrieben von der Eingliederung bereits definierter Komponenten, die dadurch umliegende Bauteile beeinflussen. Diese Beeinflussungen können aber auch bei der parallelen Entwicklung von Bauteilen auftreten. So haben beispielsweise Anpassungen bei einer Fahrzeugseitenwand Auswirkungen auf die Gestalt einer Türe und umgekehrt. Gerade bei diesen Wechselwirkungen, wie sie eben bei *Bottom-up*-Modellierungsstrategien auftreten können, liefern *Adaptermodelle* ein bewährtes Werkzeug für eine effiziente Modellierung der resultierenden Geometrie.

So bieten *Adaptermodelle* – im Gegensatz zu *Skelettmodellen* – die Möglichkeit für Verknüpfungen innerhalb einer gleichen Gliederungsebene wie auch zu einer Systemumwelt. Abb. 3.11 zeigt dazu eine repräsentative Anordnung von *Adaptermodellen* sowie ihre Verknüpfungen innerhalb einer Baugruppenstruktur. Als ein Beispiel für eine Relation innerhalb einer Baugruppe dienen die Ergebnisse von Bauteil 1.1 – beispielsweise ein Karosserieunterboden –, die als Eingaben für die Geometrieerzeugung von Bauteil 1.3 – eine Seitenwand – herangezogen werden. Ähnlich dazu sind Bauteil 1.3 und Adapter 2.1 verknüpft, jedoch handelt es sich hierbei um eine Verbindung zwischen Baugruppen, wie etwa einer Karosserie und einem Türrohbau. Hierbei ist zu beachten, dass innerhalb von Baugruppen in der Regel keine eigenen *Effector*-Bauteile vorhanden sind, sondern die resultierende Baugruppen-Geometrie zumeist im CAD-System als sog. *publizierte Geometrie* bereitgestellt wird. Diese systeminternen Geometrielemente haben die Funktion eines *Effectors*, sind in Abb. 3.11 jedoch nicht separat dargestellt, sondern der Ausgangsebene eines Bauteils gleichgesetzt. Ein Beispiel für eine Relation zu einer Systemumwelt liefert Bauteil 1.3, dessen Ergebnisse für die Geometriemodellierung an einer anderen Stelle herangezogen werden.

Der umfassende Einsatz von *Adaptermodellen* führt aufgrund der entstehenden Abhängigkeiten und Verknüpfungen zu einem weitreichenden Beziehungsgeflecht. Es besteht dabei die Gefahr der Unübersichtlichkeit sowie von Zirkelbezügen. Diese Abhängigkeiten stellen daher auch eine Herausforderung für das Verknüpfungsmanagement von CAD-Systemen dar, das unter anderem die Aktualität der Eingaben gewährleistet. So kann beispielsweise in Abb. 3.11 die Baugruppe 2 nicht getrennt von Baugruppe 1 betrachtet werden, da letztere Eingaben enthält, die von der Baugruppe 2 benötigt werden. Diese Abhängigkeiten wirken außerdem nur unidirektional, wodurch die Gestalt einer Baugruppengeometrie die andere beeinflusst, jedoch nicht umgekehrt. Daher muss bei angrenzenden Bauteilen, die mit einem Adapter verbunden sind, klar geregelt sein, welches Bauteil die Eingaben für das *Adaptermodell* und in weiterer Folge für das davon abhängige Bauteil liefert. Aufgrund dessen ist eine sorgfältige Auswahl von Abhängigkeiten und eine klare Aufteilung von Verantwortlichkeiten zu treffen. So werden gerade in einer seriennahen Entwicklung vorhandene Verknüpfungen bewusst aufgebrochen, um weitreichende Änderungen nicht automatisch über die Grenzen der Verantwortlichkeiten hinweg umzusetzen. Daher geht in der Regel eine intensive Abstimmung zwischen den verantwortlichen Stellen einer Anpassung der Adaptergeometrie voraus.

In der industriellen Praxis werden *Adapter-* und *Skelettmodelle* je nach Aufgabenstellung sowohl getrennt als auch in Kombination verwendet. Darüber hinaus kann durch eine Veränderung der Verknüpfungen prinzipiell zwischen den Typen gewechselt werden. Dennoch erweist sich, abhängig von der jeweiligen Anwendung, zumeist eines der beiden Konzepte als vorteilhaft. Die Rubrik der *Matingtemplates* wird in Abb. 3.7 daher in *Templates* aufgeteilt, die in ihrer obersten Gliederungsebene entweder ein *Adapter-* oder ein *Skelettmodell* besitzen. Entsprechend der angeführten Definition eines *Templates*, das Vorlagen beginnend mit Modellierungssequenzen bis hin zu umfangreichen Baugruppen einschließt, lässt sich die Kategorie der *Matingtemplates* weitgehend auf alle Modellebenen anwenden. Diese Ebenen umfassen im Wesentlichen Geometriesequenzen, Bauteile und Baugruppen. Aufgrund der Besonderheiten innerhalb der einzelnen Ebenen sowie deren unterschiedlichen Einsatzzweck haben auch deren *Matingtemplates* spezifische Ausprägungen.

Geometriesequenzen als *Matingtemplates*

Geometriesequenzen sind die kleinste Einheit, die sich in ein *Matingtemplate* transformieren lässt. Diese umschließen eine Reihenfolge an Modellelementen und -operationen innerhalb eines Bauteils, die einen Teilbereich eines Geometriemodells beschreiben. Viele wissensbasierte CAD-Systeme besitzen die Fähigkeit, einzelne Modellabfolgen zusammenzufassen, abzulegen und sie damit einer Wiederverwendung bereitzustellen. Im Sinne einer wissensbasierten Konstruktion eignen sich hierfür vor allem wiederkehrende Konstruktionselemente, die nicht nur bei einer spezifischen Komponente verwendet werden, sondern sich in mehreren unterschiedlichen Bauteilen wiederfinden. Als typische Beispiele im Bereich der Karosseriekonstruktion gelten dafür unter anderem Sicken, Flansche oder Verprägungen.

Bei der Erstellung des *Matingtemplates* sind jene Eingaben zu definieren, welche in der Modellsequenz für die weitere Verarbeitung benötigt werden. Sobald eine *Template*-Geometriesequenz in ein vorhandenes Bauteilmodell eingefügt wird, muss eine Initialisierung der Eingabeverknüpfungen erfolgen. Dabei entscheidet es sich, inwieweit sich die Eingaben der Geometriesequenz entsprechend eines *Adapter-* oder eines *Skelettmodells* verhalten. Nach dem Initialisieren einer *Template*-Geometriesequenz werden alle enthaltenen Objekte entsprechend der vorgegebenen Ursprungssequenz in die Bauteilstruktur integriert. Hierin besteht ein wesentlicher Unterschied zu UDFs. Diese können ebenfalls dazu verwendet werden, aus Modellabfolgen Vorlagen zu erstellen. Jedoch besteht bei der Anwendung von UDFs in der Regel keine Möglichkeit, auf die zu Grunde liegenden Geometrieobjekte direkt zuzugreifen. Ein UDF erscheint – dem *Feature*-Ansatz folgend – als einzelnes Element in der Modellstruktur und kann lediglich durch Ändern der Eingabeparameter und -verknüpfungen angepasst, jedoch in seinem Aufbau nicht verändert werden. Der Vorteil eines UDFs im Vergleich zu einer *Template*-Geometriesequenz besteht allerdings in der Fähigkeit, zusätzliche, über Geometrieinformation hinausgehende Attribute, zu enthalten.

Bei der Organisation von *Template*-Geometriesequenzen und UDFs kann oftmals auf eine systeminterne Bibliotheksfunktion der CAD-Programme zurückgegriffen werden. Damit lassen sich die enthaltenen Modellabfolgen strukturieren, kategorisieren und direkt in der CAD-

Umgebung einsetzen. Neben den Bibliotheken besteht eine weitere Option für die Verwaltung von Geometriesequenzen – wie auch für andere Vorlagen – in der direkten Integration in ein PDM-System. Hier können die Ursprungsmodelle abgelegt werden und bei Bedarf lassen sich die jeweiligen Modelle öffnen und die enthaltenen Geometriesequenzen in ein Bauteil einfügen. Die Wahl der Organisationsmethode hängt in hohem Maße von den Fähigkeiten des verwendeten CAD-Systems wie auch von innerbetrieblichen Randbedingungen ab.

Bauteile als Matingtemplates

Matingtemplates auf Bauteilebene bilden das Rückgrat einer *Template*-basierten Geometrierstellung, da Geometriedaten innerhalb eines CAD-Systems vorrangig in Bauteilmodellen abgelegt werden. Ein Bauteilmodell kann als zentraler Datenträger für Geometrie wie auch für Produktdaten angesehen werden. Darüber hinaus besteht gerade auf Bauteilebene ein großes Interesse, bewährte Lösungen auch bei neuen Komponenten anzuwenden, wobei sich prinzipiell ein entsprechend aufbereitetes parametrisch-assoziatives Bauteil auch in ein *Matingtemplate* umwandeln lässt. Es sind lediglich die entsprechenden Ein- und Ausgabeschnittstellen zu erstellen und für die Eingaben Dummyelemente zu inkludieren, wodurch das Bauteil entkoppelt von der Systemumwelt betrachtet werden kann. Darüber hinaus ist die Modellstabilität sicherzustellen, um für den angestrebten Wertebereich keine fehlerhaften Ergebnisse zu erhalten. Die Ablage eines *Matingtemplates* auf Bauteilebene kann in einem PDM-System erfolgen, wobei bauteilspezifische Eigenschaften, wie etwa eine Bauteilnummer, anzupassen sind, sobald die Vorlage für eine neue Komponente verwendet wird. Die Herausforderungen bei einer systematischen Erstellung von *Matingtemplates* auf Bauteilebene unterscheiden sich kaum von der generellen Problematik beim Einsatz von Vorlagen. Diese liegen unter anderem im Erstellungsaufwand, in der Archivierung und Wartung der Modelle, der Dokumentation sowie dem Sicherstellen einer großen Bandbreite an Eingaben.

Baugruppen als Matingtemplates

Matingtemplates auf Baugruppenebene dienen üblicherweise als Vorlage für eine Anordnung von Bauteilen, die in einem Zusammenbau miteinander verbunden sind. Diese Vorlagen stellen sowohl die Baugruppenstruktur als auch die Produktdaten innerhalb von Bauteilmodellen zur Verfügung. Dabei kommt in allen enthaltenen Ebenen, wie etwa in Unterbaugruppen oder in Bauteilen, der EVA-Ansatz konsequent zum Einsatz. *Matingtemplates* auf Baugruppenebene enthalten demnach fast ausschließlich Bauteil-*Matingtemplates*, in denen auch sämtliche Geometrieinformationen enthalten sind. Die vorgegebenen Baugruppen beschränken sich daher vordergründig auf die Anordnung und Organisation der enthaltenen Bauteile und können diese auch in Unterbaugruppen zusammenfassen. Eine typische Anwendung eines Baugruppen-*Matingtemplates* besteht in der Karosseriekonstruktion in weitgehend geschlossenen Konstruktionsbereichen wie beispielsweise einem Unterboden, einer Seitenwand oder einer Heckklappe. Ein Beispiel für ein *Matingtemplate* auf Baugruppenebene liefert Abb. 3.11. Dort kann die dargestellte Hauptbaugruppe als ein *Matingtemplate* verstanden werden, da in der obersten

Ebene entweder ein *Skelett*- oder ein *Adaptermodell* die Eingangsdaten zusammenfasst und an die darunterliegenden Ebenen weiterleitet.

Ein Spezifikum von Baugruppentemplates besteht in der Verwaltung der *Skelett*- und *Adaptermodelle*, die als eigene Bauteile in der Baugruppenstruktur integriert sind. Diese übernehmen die Funktion von Schnittstellen, enthalten jedoch selbst keine zusätzliche Produktinformation und scheinen demzufolge auch nicht als eigene Bauteile beispielsweise in Stücklisten oder in der Produktionsplanung auf. Dennoch müssen sie gegebenenfalls in ein PDM-System eingebunden und gewartet werden. Des Weiteren stellen diese *Skelett*- und *Adaptermodelle* eine Sollbruchstelle innerhalb einer weitverzweigten, assoziativen Baugruppenstruktur dar. Gerade an diesen Stellen bietet es sich aus Gründen der Handhabbarkeit bei einem Übergang von einer konzeptionellen zu einer seriennahen Entwicklung an, vorhandene Verknüpfungen aufzutrennen. So erschweren unter anderem ein verstärkt eingreifendes Änderungsmanagement oder das zunehmende Auslagern von Konstruktionstätigkeiten an Lieferanten ein Beibehalten von bauteilübergreifenden Abhängigkeiten.

Einsatz von Matingtemplates

Der umfangreiche Einsatz von *Matingtemplates* in allen Ebenen, beginnend bei Geometriesequenzen bis hin zu komplexen Baugruppen, bietet vielerlei Möglichkeiten und Vorteile. Einer dieser Vorteile besteht beispielsweise in den vorgegebenen Schnittstellen. So lassen sich *Matingtemplates* mit gleichen Schnittstellen rasch austauschen, weshalb leicht zwischen unterschiedlichen Varianten gewechselt werden kann. Abb. 3.12 veranschaulicht diesen Umstand und verwendet wiederum eine systemtechnische Darstellung, wobei es sich bei dem System um ein Bauteil-*Template* mit insgesamt fünf Geometriesequenzen als Subsysteme handelt. Es ist ersichtlich, dass zwar alle Eingabeebenen der Subsysteme mit Eingangsdaten befüllt werden, aber nur die Ausgabebenen der Subsysteme 2 und B mit dem *Effectormodell* des Systems verbunden sind und folglich für die resultierende Bauteilgeometrie sorgen. Durch Ändern der Relationen lassen sich jedoch eine Vielzahl an Geometrievarianten erstellen. Kombiniert man also im System von Abb. 3.12 jeweils eines der Subsysteme 1, 2 oder 3 mit einem Subsystem A oder B, können insgesamt sechs Bauteilvarianten erstellt werden. Darüber hinaus lassen sich weitere Konstruktionsvarianten durch Einfügen und Verknüpfen von neuen Subsystemen vergleichsweise einfach erstellen, wodurch die Variantenvielfalt rasch erhöht werden kann.

Gerade in einer frühen Entwicklungsphase, in der eine Vielzahl an Varianten untersucht wird, bietet sich also ein weitreichender Einsatz von *Matingtemplates* an. Dabei enthalten *Matingtemplates* zumeist einen relativ hohen Detaillierungsgrad, sodass die frühzeitige Erstellung von einer hochwertigen Konzeptgeometrie unterstützt wird. Auf eine allzu hohe und für eine frühe Phase unnötige Detaillierung, wie sie in einem Serienbauteil vorzufinden ist, wird jedoch in der Regel bewusst verzichtet. Aufgrund dessen bleiben die *Templates* universell einsetzbar und projektspezifische Detailkonstruktionen, die zumeist auch nicht auf andere Entwicklungsprojekte übertragbar sind, erfolgen erst in einer seriennahen Entwicklung. Daher werden *Matingtemplates* vor allem in einer frühen Entwicklungsphase für eine Variantenauswahl verwendet und dienen dann als Basis für die Serienentwicklung.

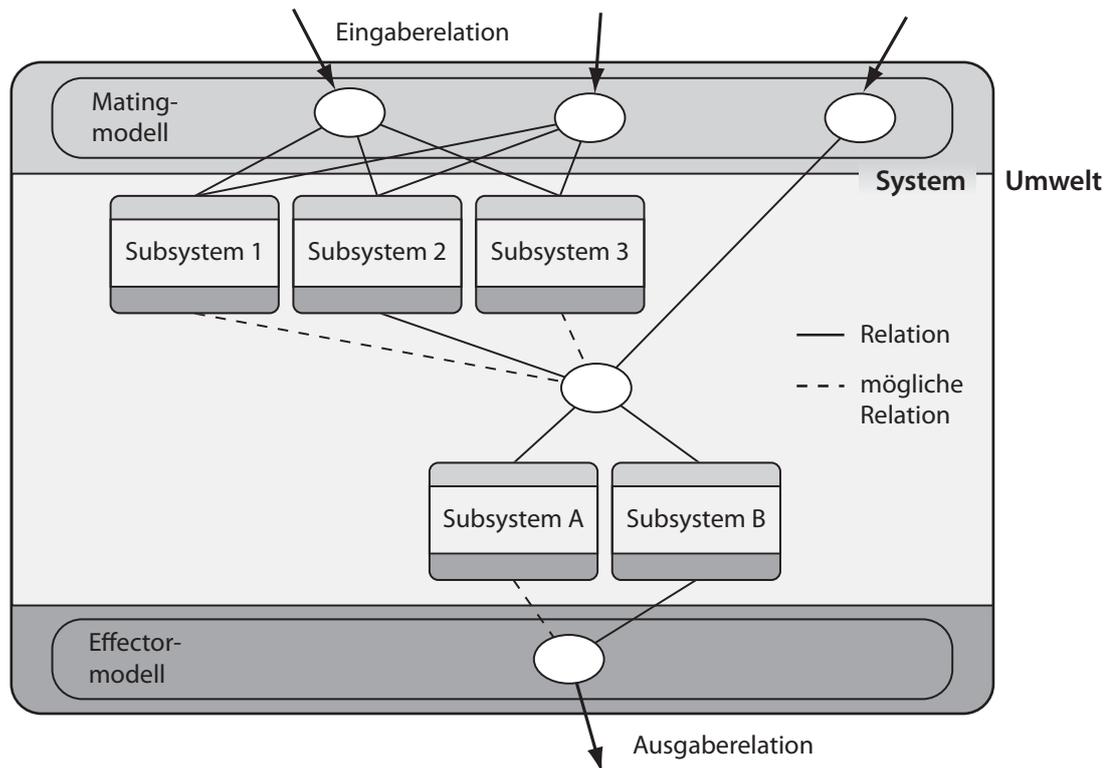


Abbildung 3.12: Möglichkeiten für die Darstellung von Varianten durch den Einsatz von *Matingtemplates*.

Abhängig davon, ob ein verwendetes *Matingtemplate* gleichzeitig auch den Strukturanforderungen eines *Startmodells* entspricht, kann das Modell von einer frühen Entwicklungsphase bis in die Serienentwicklung beibehalten werden. Anderenfalls muss die Konzeptgeometrie in ein neues *Startmodell* überführt werden, was zumeist einen Bruch in der Parametrik nach sich zieht und gegebenenfalls zu einer neuerlichen Modellierung der Bauteilgeometrie in einem *Startmodell* führt. Ein weiterer Aspekt beim Übergang in eine seriennahe Entwicklung liegt im bereits angeführten Auftrennen der assoziativen Beziehungen innerhalb der Produktstruktur. So kann bei einer konsequenten Anwendung des EVA-Ansatzes und einem umfangreichen Einsatz von *Matingtemplates* ein Produktmodell mit weitreichenden Verknüpfungen und Abhängigkeiten entstehen. Dieser Umstand erweist sich in einer frühen Entwicklungsphase als vorteilhaft, da die Produktkomplexität vergleichsweise gering ist, jedoch gleichzeitig eine hohe Variabilität gefordert wird. So passt sich eine Konzeptgeometrie rasch an veränderte Eingaben – wie etwa ein neues Fahrzeugdesign – an und es lassen sich viele Varianten untersuchen. Im Laufe des Entwicklungsprozesses sinken jedoch die Anforderungen hinsichtlich einer Modellvariabilität, der Detaillierungsgrad steigt und infolge umfangreicher parallel ablaufender Entwicklungstätigkeiten treten Prozessanforderungen in den Vordergrund.

Matingtemplates leisten aufgrund ihrer Eigenschaften gerade in einer frühen Entwicklungsphase einen wesentlichen Beitrag für eine wissensbasierte Konstruktion im Rahmen des *Frontloading*-Ansatzes. So liefern die *Templates* einen Baukasten an erprobten Geometriemodellen, die steuerbar und adaptierbar sind. Da sich *Matingtemplates* allerdings als komplex und umfangreich

erweisen können, ist eine wichtige Voraussetzung für ihre effiziente Anwendung die Bereitstellung einer Unterstützung für den jeweiligen Anwender. Diese kann von einer Dokumentation bis hin zu umfangreichen Assistenzsystemen und interaktiven Hilfeportalen reichen [Arn08]. Des Weiteren ist aufgrund der zum Teil hohen Komplexität der *Matingtemplates* ein großes Augenmerk auf ihre Archivierung, Verwaltung und Wartung zu legen. So spielt gerade die Integration in ein PDM-System eine entscheidende Rolle, da hierdurch eine zentrale Datenablage geschaffen wird. [HAR06].

3.5.5 Funktionstemplates

Die Kategorie der *Funktionstemplates* fasst alle CAD-basierten Vorlagen zusammen, mit denen Vorgaben abgesichert werden oder mit denen eine Produktauslegung erfolgt. Die Vorgaben können sowohl von funktionalen Anforderungen als auch von Anforderungen aus Folgeprozessen, wie etwa der Produktion, stammen. Das CAD-System wird in diesen Fällen weniger als Konstruktions- und Dokumentationswerkzeug, sondern vielmehr als ein integriertes Auslegungswerkzeug genutzt, das vielfach auf die Funktionalitäten wissensbasierter CAD-Systeme zurückgreift. Die Komplexität der verschiedenen *Funktionstemplates* kann allerdings stark variieren. So reicht die Bandbreite von vergleichsweise simplen Berechnungen bis hin zu komplexen *Template*-basierten Strukturen samt dazugehörigen Software-Applikationen und Assistenzsystemen. Diese Applikationen beinhalten neben einer Parametersteuerung und einer Bereitstellung von Berechnungsalgorithmen zumeist auch eine graphische Benutzeroberfläche. Dadurch vereinfacht sich die Anwendung der *Templates*, was wiederum für eine hohe Akzeptanz bei den Benutzern sorgt. Aus diesem Grund verwenden gerade komplexe und umfangreiche *Funktionstemplates* Schnittstellen wie APIs und greifen auf Software-Applikationen zurück.

Dadurch lässt sich der Funktionsumfang der *Templates* massiv ausdehnen und es kann auf Fähigkeiten zurückgegriffen werden, die innerhalb des CAD-Systems nicht vorhanden sind. Demnach stehen den *Funktionstemplates* die gleichen Formen einer Koppelung zwischen einem *Template* und einem externen Programm zur Verfügung, wie sie auch schon für CAD-Programme und KBE-Systeme in Abb. 3.2 gezeigt werden. So gibt es *Templates*, die vollständig mit den Funktionalitäten des CAD-Systems auskommen, *Templates*, die über eine Schnittstelle mit externen Programmen verbunden sind, und *Templates*, bei denen zumindest Teile einer Absicherung extern erfolgen, es allerdings zu keinem automatischen Datenaustausch kommt.

Der Umfang eines *Funktionstemplates*, der außerhalb des CAD-Systems dargestellt wird, kann stark variieren. Dies kann soweit führen, dass die Hauptfunktionalitäten nicht mehr durch das CAD-System, sondern durch externe Programme bereitgestellt werden. Dies erschwert eine klare Abgrenzung der *Funktionstemplates* zu Software-Applikationen, welche das CAD-System als Hilfsprogramm verwenden und lediglich auf dessen Funktionen zurückgreifen. In dieser Arbeit besteht daher ein Merkmal von *Funktionstemplates* darin, dass das CAD-System die zentrale Benutzerschnittstelle liefert. Demnach sollen neben den Eingangsdaten vor allem die Ergebnisse im CAD-System vorliegen und einer Weiterverarbeitung zugänglich gemacht werden. Einen typischen Aufbau eines *Funktionstemplates*, wie es in den Anwendungsbeispielen in den folgenden Kapitel eingesetzt wird, zeigt Abb. 3.13.

Das dargestellte *Template* ist über ein Assistenzsystem mit externen Programmen gekoppelt und besitzt Schnittstellen für die Eingabedaten und die Ergebnisdokumentation. Die Eingaben von einem *Funktionstemplate* umfassen üblicherweise Geometrieobjekte und Parameterwerte. Gerade die Geometriedaten stammen in der Regel aus der CAD-Umgebung und werden direkt in das *Funktionstemplate* eingefügt, während die Parameterwerte ihren Ursprung auch in externen Datenquellen haben können. Außerdem können Parameterwerte ebenfalls von Benutzereingaben stammen, die üblicherweise mit Hilfe einer Benutzeroberfläche getätigt werden. Wie auch schon bei den *Matingtemplates* bietet es sich an, im *Funktionstemplate* Dummyelemente und Standardwerte für notwendige Eingaben vorzusehen, um im Falle von unvollständigen Eingaben darauf zurückgreifen zu können. Das *Funktionstemplate* selbst teilt sich in zwei Bereiche auf: die konventionelle CAD-Umgebung mit allen Bauteil- und Baugruppenelementen und das Assistenzsystem. Neben der Bereitstellung von einer Benutzeroberfläche und von Berechnungsalgorithmen dient das Assistenzsystem in Abb. 3.13 ebenfalls für eine Koppelung mit externen Programmen. Gerade der Datentransfer zwischen CAD-Umgebung und der Systemumwelt sowie die Ablaufsteuerung einer Auslegung wird vom Assistenzsystem durchgeführt. Dadurch nimmt das Assistenzsystem eine zentrale Rolle ein und verbindet das *Funktionstemplate* zum einen mit dem Anwender und zum anderen mit der Systemumwelt. Außerdem sorgt es für eine Aufbereitung und Ausgabe der Ergebnisse. Diese umfassen einerseits Geometriedaten und andererseits Parameterwerte, die auch abseits einer CAD-Umgebung als Berichte oder Konstruktionstabellen abgelegt werden können.

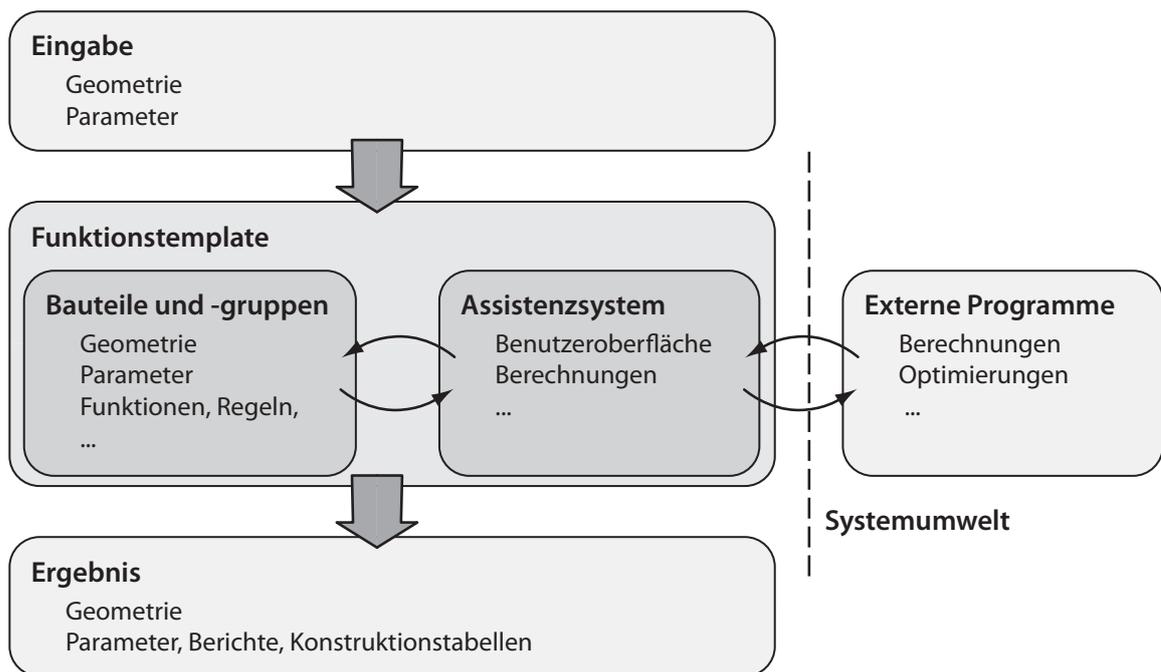


Abbildung 3.13: Beispielhafte Anordnung eines *Funktionstemplates*.

Gerade die Weiterverarbeitung der Geometriedaten stellt eine Domäne der CAD-Systeme dar. So können skelettartige Ergebnismodelle mit Hilfe von *Matingtemplates* für eine automatisierte Geometrierzeugung herangezogen werden. Außerdem dienen Hüllgeometrien als Eingangs-

daten für die Dimensionierung und Positionierung von Bauteilen. Diese Grenzflächen haben ihren Ursprung beispielsweise in Einbau- und Ergonomieuntersuchungen sowie in gesetzlichen Restriktionen oder in kinematischen Analysen.

Funktionstemplates decken ein weites Gebiet an konstruktionsnahen Auslegungswerkzeugen ab, was zu einer Vielzahl an unterschiedlichen Ausprägungen führt. Als ein repräsentativer Ausschnitt werden hier aktuelle Ansätze von *Skarka*, *Krish*, *Hirz* und *Stadler* angeführt, um die Bandbreite an unterschiedlichen Konzepten darzustellen. So geht *Skarka* im Besonderen auf die MOKA-Systematik zur Erstellung von wissensbasierten Konstruktionsvorlagen ein und erstellt generische Spannmittel [Ska07]. *Krish* legt seinen Fokus auf die frühe Phase, um dem Anwender eine Vielzahl an möglichen Konstruktionsvorschlägen zu liefern und zeigt Anwendungsbeispiele für Mobiltelefone und Schreibtische [Kri11]. *Hirz* stellt eine *Template*-basierte Applikation für die konzeptionelle Gesamtfahrzeugentwicklung vor. Neben Aufgaben einer geometrischen Integration und einer Verwaltung durch datenbankartige Strukturen werden ebenfalls konzeptionelle Absicherungen und Auslegungen durchgeführt [Hir11]. *Stadler* zeigt anhand eines Auslegungswerkzeugs für die Konzeptauslegung von Fahrzeugsitzen die Vorteile eines integrierten Auslegungswerkzeuges, welches Schnittstellen zu externen Programmen wie Simulationswerkzeugen, Datenbanken und PDM-Systemen aufweist. Dadurch kann der Anwender bei seinen Entwicklungstätigkeiten unterstützt werden, ohne weitere externe Programme bedienen zu müssen [SH14].

Funktionstemplates führen dem Entwicklungsprozess in hohem Ausmaß vorhandenes Wissen zu und leisten dadurch einen wesentlichen Beitrag für eine wissensbasierte Konstruktion. Ein zusätzlicher Aspekt bei der Erstellung und beim Einsatz von *Funktionstemplates* besteht allerdings im Schutz des bereitgestellten Wissens. CAD-Systeme bieten daher vielfach die Möglichkeit, die vorhandenen Funktionen, Algorithmen und Programmabläufe zu maskieren. Dadurch kann der Anwender die Funktionen zwar verwenden, hat jedoch keinen Zugriff auf das zu Grunde liegende Wissen. Dieser Prozess kann soweit gehen, dass aus *Funktionstemplates* und deren Assistenzsystemen eigene Software-Pakete erstellt werden. Dies geschieht im Besonderen, wenn die Auslegungswerkzeuge kommerziell verfügbar sind oder von externen Auftragnehmern erstellt werden. Aktuelle Ansätze zum Thema Wissensschutz von CAD-Modellen werden unter anderem durch *Mantwill* zusammengefasst [Man08a].

Da *Funktionstemplates* oftmals einen hohen Grad an Komplexität aufweisen, ist großer Wert auf deren Dokumentation, Wartung und Verwaltung zu legen. Außerdem kommen die unterschiedlichen *Funktionstemplates* im Allgemeinen losgelöst von einer stücklistenrelevanten Produktstruktur zum Einsatz. Sie dienen daher nicht der Dokumentation von Produktdaten, sondern vielmehr als entwicklungsbegleitende Werkzeuge, welche Auslegungen durchführen sowie Änderungen überprüfen und absichern. Hierbei kann der Einsatz von *Funktionstemplates* prinzipiell in jeder Entwicklungsphase erfolgen. Allerdings besteht gerade in einer frühen Entwicklungsphase –dem *Frontloading*-Ansatz folgend– ein vermehrter Bedarf an einer konstruktionsnahen Auslegung, sodass *Funktionstemplates* im erhöhten Ausmaß in diesem Zeitabschnitt der Produktentwicklung zum Einsatz kommen.

3.5.6 Ansätze für eine Template-unterstützte Entwicklung

Mit den in Abschnitt 3.5 beschriebenen Kategorien an *Templates* lässt sich die Produktentwicklung in vielen Belangen unterstützen und die Konstruktion unter Zuhilfenahme von KBE-Methoden effizienter gestalten. So liefern *Templates* bewährte Lösungen, mit denen sowohl die Produktqualität als auch die Prozessstabilität erhöht werden. Außerdem verringert sich durch eine *Template*-unterstützte Entwicklung vielfach die Komplexität von Konstruktionstätigkeiten, was zu einer Entlastung des Anwenders und einer damit verbundenen Reduktion von Fehlern führt. Des Weiteren berücksichtigen *Templates* die Anforderungen der einzelnen Entwicklungsphasen und beziehen nachgelagerte Prozesse, wie etwa die Produktion, mit ein.

Aufgrund der angeführten Vorteile finden *Templates* eine breite Verwendung in der Fahrzeugentwicklung. Nichtsdestoweniger wird versucht, die Wirksamkeit durch eine Verbesserung von Methoden, Werkzeugen und Prozessen weiter zu erhöhen. Dies geschieht unter anderem auf Grundlage von aktuellen Forschungsergebnissen und Ansätzen, die das Ziel verfolgen, den Einsatz von *Templates* zu optimieren. Bei diesen Konzepten werden einerseits Teilbereiche der unterschiedlichen *Template*-Kategorien herausgegriffen und isoliert betrachtet sowie andererseits eine gesamtheitliche Herangehensweise untersucht.

So unterstreicht *Mbang Sama* die Bedeutung von *Templates* und *Features* für die Entwicklung von Karosseriebauteilen. In seiner ganzheitlichen Betrachtungsweise liefern *Templates* und *Features* wichtige Bausteine für einen modernen Entwicklungsprozess, bei dem es zu einer engen Verzahnung zwischen Produktentwicklung und Prozessplanung kommt [Mba08]. Ein übergeordneter Ansatz für den Einsatz von *Templates* in der Fahrzeugentwicklung wird außerdem von *Haasis* ausgearbeitet. Er klassifiziert unterschiedliche *Template*-Kategorien, geht auf deren Anwendung ein und behandelt Aspekte der Modellierung und Archivierung. Darüber hinaus beschreibt er einen *Template*-basierten Konstruktionsprozess [HAR06, HVA08]. *Prieur* erstellt ein Konzept für die Konstruktion von Tiefziehwerkzeugen, bei dem er Funktionselemente und *Templates* einsetzt. In seiner Arbeit geht er außerdem auf die Anordnung und Verknüpfung dieser Elemente ein und beleuchtet ebenfalls die Systemimplementierung seines Ansatzes [Pri06].

Alle drei angeführten Arbeiten widmen sich einer Verbesserung der Produktentwicklung mit Hilfe von *Templates*, allerdings liegt das Hauptaugenmerk nicht auf einer Anwendung von *Funktionstemplates*. Dessen ungeachtet liefern gerade *Funktionstemplates* ein Mittel zur frühzeitigen Produktgestaltung im Sinne des *Frontloadings*. Abb. 3.14 verdeutlicht diesen Umstand und vergleicht die unterschiedlichen *Template*-Kategorien. So wird einerseits die Eignung für den Wissenstransfer in die frühe Entwicklungsphase betrachtet und andererseits die Modellkomplexität der *Templates* verglichen. Der Grad an Komplexität korreliert mit dem Aufwand für die Methodenentwicklung sowie den Anforderungen an die Funktionalitäten des eingesetzten CAD-Systems. Werden die *Strukturtemplates* betrachtet, zeigt sich, dass sie für keinen wesentlichen Wissenstransfer in die frühe Phase sorgen. Ihre Vorteile kommen vor allem in einer seriennahen Entwicklung zum Tragen, während die Verwendung in einer frühen Entwicklungsphase zu keiner wesentlichen Stärkung dieses Abschnitts im Sinne des *Frontloadings* führt. Demgegenüber eignen sich *Geometrietemplates* besser als *Frontloading*-Werkzeuge, wobei deren Komplexität

und der Aufwand für die *Template*-Erstellung variiert. So benötigt die Generierung von *Geometriedummies* wenige Ressourcen während sich *Matingtemplates* durchaus als weitverzweigt erweisen können und daher hohe Anforderungen an ihre Modellierung stellen. Gleiches gilt auch für *Funktionstemplates*. Deren Komplexität und Modellierungsaufwand variiert ebenfalls stark und kann gerade bei *Funktionstemplates* äußerst umfangreich sein. Nichtsdestotrotz bieten sie im Vergleich zu den übrigen *Template*-Kategorien ein großes Potential für den Wissenstransfer in die frühe Entwicklungsphase.

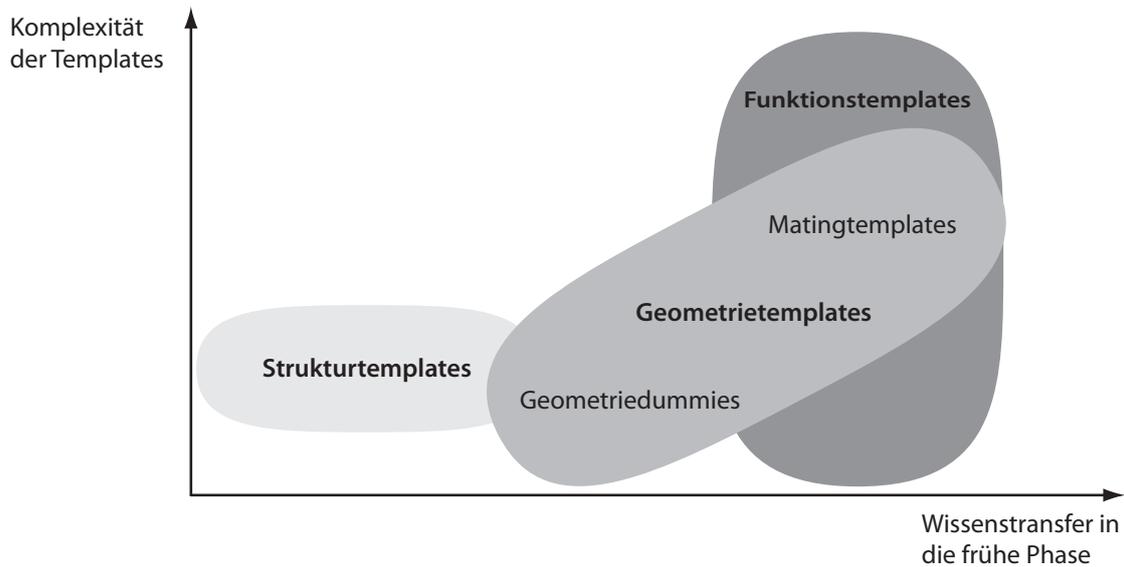


Abbildung 3.14: Bedeutung der *Template*-Kategorien für die frühe Entwicklungsphase.

Um die Produktentwicklung in der frühen Phase durch den Einsatz von *Templates* zu verbessern, wird daher im folgenden Kapitel auf eine Koppelung von *Funktionstemplates* mit anderen *Template*-Kategorien, insbesondere *Matingtemplates*, eingegangen. Das Hauptaugenmerk liegt dabei auf dem Übergang von der Definitionsphase, die mit relativ geringen personellen Ressourcen auskommt und sich auf die Ebene der Gesamtfahrzeugentwicklung beschränkt, zur Konzeptphase, bei der es zu einer starken Erweiterung des beteiligten Personenkreises und einer Konzeptentwicklung auf Komponentenebene kommt. Hier besteht ein großer Bedarf an einer konstruktionsnahen Auslegung und Absicherung von Komponenten sowie einer Geometrieerzeugung mittels Vorlagen. Die Ergebnisse werden in weiterer Folge im Rahmen eines Vorgehensmodell für eine *Template*-basierte Produktentwicklung in der Konzeptphase zusammengefasst.

3.6 Zusammenfassung

Die wissensbasierte Konstruktion liefert Methoden und Werkzeuge, mit denen in der Produktentwicklung frühzeitig Expertenwissen zur Verfügung steht und stellt daher ein zentrales *Front-loading*-Element dar. Dieser Abschnitt behandelt eingangs den Ansatz der wissensbasierten Konstruktion und geht auf die Eigenschaften sowie die Voraussetzungen für eine erfolgreiche

Umsetzung ein. Als ein wesentliches Element für den wirksamen Einsatz werden hierbei die Fähigkeiten des eingesetzten CAD-Systems angesehen. Gerade CAD-Programme haben in den letzten Dekaden ihre Funktionalitäten umfassend erweitert und können somit eine wissensbasierte Konstruktion weitreichend unterstützen. Daher betrachtet dieser Abschnitt ebenfalls die Entwicklung der CAD-Systeme und gibt einen Überblick auf die bereitgestellten Funktionalitäten. Eine dieser Fähigkeiten besteht in der Erstellung von *Templates*, die als Vorlagen vorhandenes Wissen speichern und aufbereiten. Diese Eigenschaften machen *Templates* zu einem wichtigen Werkzeug für den Einsatz von wissensbasierter Konstruktion.

In diesem Kapitel werden die unterschiedlichen Ausprägungen von *Templates* kategorisiert und es wird auf die Eigenschaften der jeweiligen Kategorie eingegangen. Die Kategorien reichen von *Strukturtemplates*, die eine Modellgliederung vorgeben, über *Geometrietemplates*, die entweder fixe oder variable Geometriemodelle umfassen, bis hin zu *Funktionstemplates*, die für eine funktionale Auslegung und Absicherung herangezogen werden. Gerade letztere eignen sich im Besonderen für einen Wissenstransfer in die frühe Entwicklungsphase. In der weiteren Arbeit wird daher speziell auf eine Koppelung von *Geometrie-* und *Funktionstemplates* als ein Werkzeug für einen wissensbasierten, *Frontloading*-getriebenen Entwicklungsprozess eingegangen.

4 Vorgehensmodell einer CAD-basierten Vorauslegung

4.1 Einleitung

In der frühen Entwicklungsphase stellt insbesondere der Übergang von der Definitions- in die Konzeptphase eine Herausforderung für einen effizienten Entwicklungsprozess dar. So kommt es zu einem massiven Anstieg an involvierten Personen, der mit einem Zuwachs an Datenmaterial, einem steigenden Detaillierungsgrad und einer damit verbundenen wachsenden Produktkomplexität einhergeht. Allerdings kann die Qualität und der Reifegrad der Daten nicht mit dem starken Zuwachs an Datenmaterial schritthalten. Dies erfordert einen vermehrten Abstimmungsaufwand und führt zu oftmaligen Änderungsschleifen. Daher verlangt diese Phase nach flexiblen und gleichzeitig robusten virtuellen Werkzeugen, die eine Konzeptentwicklung unterstützen und mit sich ständig ändernden Randbedingungen zurecht kommen. Gerade für diese Anforderungen erweist sich der Einsatz von *Templates* samt dazugehörigen Assistenzsystemen als besonders geeignet. So kann etwa mit Hilfe von *Funktionstemplates* im Sinne des *Frontloadings* Expertenwissen bereitgestellt werden und eine konstruktionsnahe Auslegung von Produkteigenschaften unterstützt werden. Anhand dieser vorläufigen Funktionsauslegung lassen sich mit Hilfe von *Geometrietemplates* ebenfalls detaillierte Geometriemodelle erstellen, die dann als hochwertige Eingangsdaten für weitere Absicherungen und Optimierungen dienen. In diesem Kapitel wird der Ansatz einer Kombination aus einer *Template*-basierten Vorauslegung und einer automatisierten Geometrieerstellung aufgegriffen und anhand eines Vorgehensmodells für die Konzeptphase aufbereitet.

4.2 Templates in der frühen Entwicklungsphase

4.2.1 Templates in der Definitionsphase – Entwurfssysteme für Gesamtfahrzeugkonzepte

Das Hauptziel der frühen Entwicklungsphase besteht in der Ausarbeitung eines Konzepts, welches alle wesentlichen geometrischen und funktionalen Eigenschaften abbildet. In diesem Konzept werden Fahrzeugeigenschaften, wie die Sitzanzahl, festgelegt, die Gewichtsverteilung definiert oder bereits bestehende Module, wie etwa Motoren und Getriebe, integriert. Darüber hinaus lassen sich anhand einer vorläufigen Stückliste Kosten- und Gewichtsziele nachvollziehen und überprüfen. Im Bezug auf die Karosserie stellt insbesondere das Fahrzeugpackage

samt Maß- und Designkonzept eine wesentliche Eingabe für die weiteren Entwicklungsschritte dar. So lässt sich darauf aufbauend eine detaillierte Konzeptgeometrie erstellen, welche bereits Anforderungen hinsichtlich Herstellbarkeit berücksichtigt. Des Weiteren werden in der frühen Entwicklungsphase die Entwicklungsziele definiert, auf ihre Machbarkeit überprüft und in einem verbindlichen Lastenheft zusammengefasst. Dies gilt insbesondere für funktionale Inhalte sowie neuartige Technologien, die parallel zum Fahrzeug noch bis zur Serienreife entwickelt werden müssen, Kap 2.2.

Trotz der großen Bedeutung der frühen Phase für den gesamten Entwicklungsprozess, Kap. 2.4, gestaltet sich die Erreichung dieser Ziele als herausfordernd. So ist gerade die frühe Entwicklungsphase gekennzeichnet durch eine Vielzahl an unterschiedlichen Variantenstudien, unvollständigen Datenständen und sich ständig ändernden Randbedingungen. Es wird daher in diversen Iterationsschleifen versucht, ein möglichst ausgereiftes und optimiertes Konzept zu erstellen. Dabei müssen gesetzliche Anforderungen, Kundenwünsche und innerbetriebliche Rahmenbedingungen unter dem Gesichtspunkt einer wirtschaftlichen Umsetzbarkeit beachtet werden. Dies führt zu einer heterogenen und dynamischen Entwicklungslandschaft, die hohe Ansprüche an die Flexibilität und die Robustheit von Prozessen, Werkzeugen und Methoden stellt. Darüber hinaus erfordert die Entwicklung in der frühen Phase einen vermehrten Organisations- und Kommunikationsaufwand [HSHL10].

Gerade unter diesen Bedingungen können virtuelle Entwicklungswerkzeuge ihre Stärken ausspielen. Demnach stellt die frühe Entwicklungsphase eine Domäne der diversen CAx-Disziplinen, allen voran des CAS, des CAE oder des CAD, dar. Diese Computer-unterstützten Systeme dienen unter anderem der Umsetzung von *Frontloading*-Ansätzen, mit denen der Entwicklungsprozess effizienter gestaltet werden kann. Fokussiert man im Bereich der Karosserieentwicklung auf Werkzeuge für die Definitionsphase, haben allen voran Entwurfssysteme für Gesamtfahrzeuge einen hohen Reifegrad erlangt. Hierbei handelt es sich um Software-Applikationen oder im Fall von CAD-Anwendungen zumeist um *Template*-basierte Assistenzsysteme, mit deren Hilfe vor allem Package-Auslegungen, Ergonomiebewertungen oder Homologationsuntersuchungen durchgeführt werden. Außerdem liefern sie Eingangsdaten für nachfolgende Berechnungen sowie Randbedingungen für das Fahrzeugdesign. Vertreter dieser Entwurfssysteme für Gesamtfahrzeuge werden unter anderem durch *Potthoff*, *Hänschke* oder *Rosbacher* beschrieben [Pot08, HM08, RHH⁺09].

So stellt *Potthoff* ein kommerziell erhältliches Entwurfssystem²³ vor, das seinen Fokus auf die Homologation und der damit verbundenen Abbildung gesetzlicher Anforderungen legt. Es umfasst unter anderem Sichtfelder, Crashbarrieren sowie Randbedingungen für die Positionierung von Komponenten wie Rückspiegel oder Scheinwerfer. Darüber hinaus ist es vollständig in ein CAD-System eingebettet, sodass eine einfache Integration einer Fahrzeug-Konzeptgeometrie ermöglicht wird [Pot08]. *Hänschke* beschreibt ebenfalls ein kommerziell erhältliches Entwurfssystem²⁴, welches seinen Schwerpunkt jedoch auf die funktionale Auslegung von Karosseriestrukturen legt. So wird eine parametrische Karosseriestruktur mit Hilfe von Profilen aufgebaut und damit einer FE-Berechnung sowie in weiterer Folge einer Optimierung zugänglich

²³CAVA, Transcat PLM GmbH, Karlsruhe

²⁴SFE Concept, Gesellschaft für Strukturanalyse in Forschung und Entwicklung mbH, Berlin

gemacht. Zusätzlich finden Bauraumanforderungen unter Zuhilfenahme vereinfachter Volumenkörper Berücksichtigung. Das Entwurfssystem selbst ist ein eigenständiges Programmpaket, es hat allerdings umfassende Schnittstellen zu kommerziellen CAD-Programmen und kann dort eingebunden werden [Hän90, HM08]. *Rossbacher* zeigt ein übergeordnetes Entwurfssystem²⁵, mit dem das Maßkonzept eines Fahrzeugs erstellt werden kann und das sich insbesondere für eine geometrische Integration sowie für die *Design-Technik-Konvergenz* eignet. Des Weiteren sind Module für eine funktionale Auslegung auf Gesamtfahrzeugebene implementiert. Das Entwurfssystem ist ebenfalls vollständig in ein CAD-System eingebettet und enthält Schnittstellen zu den voran beschriebenen Werkzeugen sowie zu weiteren externen Systemen, wie etwa Datenbanken [RHH⁺09].

Die beschriebenen Entwurfssysteme eignen sich insbesondere für die Entwicklungsaufgaben in der Definitionsphase. Dieser Abschnitt konzentriert sich vor allem auf die gesamtheitliche Gestaltung eines Fahrzeugkonzepts. In gleicher Weise beschäftigen sich auch die Entwurfssysteme vornehmlich mit der Konzeptentwicklung auf Gesamtfahrzeugebene. Sie unterstützen hierbei die geometrische Integration, welche die Bauraum- und Ergonomieauslegung unter einem ständigen Abgleich mit Design- und Gesetzesvorgaben beinhaltet und liefern Eingangsgrößen für parallel ablaufende funktionale Auslegungen. Da sich die Entwurfssysteme weitgehend auf diese Aufgaben beschränken, eignen sie sich nur eingeschränkt für Konstruktionstätigkeiten auf Modul- und Bauteilebene. Sie liefern zwar Vorgaben und Randbedingungen für die Entwicklung innerhalb dieser untergeordneten Gliederungsebenen, ihr eigentliches Anwendungsgebiet liegt jedoch weiterhin in der Gesamtfahrzeugentwicklung, die sie auch über eine Definitionsphase hinaus begleiten. Daher besteht beim Wechsel von der Definitions- in die Konzeptphase zusätzlich zu den Entwurfssystemen ein Bedarf an entwicklungsbegleitenden Auslegungswerkzeugen, da dieser Wechsel mit einem massiven Anstieg von Konstruktionsaufgaben auf Modul- und Bauteilebene einhergeht. Auch in diesem Bereich bietet sich der Einsatz von *Template*-basierten Entwicklungswerkzeugen an, die zum einen auf den Ergebnissen eines Gesamtfahrzeugentwurfs aufbauen und zum anderen auf die Charakteristika und Anforderungen der Konzeptphase abgestimmt sind.

4.2.2 Herausforderung beim Übergang von der Definitions- in die Konzeptphase

Eine wesentliche Herausforderung beim Wechsel von der Definitions- in die Konzeptphase besteht im starken Zuwachs von beteiligten Fachabteilungen und damit von benötigten Personalressourcen. Während die Entwicklung bis zu diesem Meilenstein üblicherweise in relativ kleinen Konzeptteams erfolgt, erhöht sich der Personalaufwand im Laufe der Konzeptphase massiv. Dies hat zur Folge, dass die am Entwicklungsprojekt involvierten Personen in der Regel in einer umfassenden Projektstruktur integriert werden, um den gestiegenen Aufwand für die Abstimmung und Koordination der einzelnen Stellen beherrschen zu können. Der zunehmende Personalaufwand wird vor allem durch einen erheblichen Anstieg der Struktur- und Detailtiefe des Produkts notwendig. Diese Zunahme an Produktwissen geht ebenfalls mit einem massiv steigenden Datenvolumen einher und stellt Anforderungen an das Datenmanagement [Ges01].

²⁵ *ConceptCar*, Institut für Fahrzeugtechnik an der TU Graz

Abb. 4.1 veranschaulicht den Datenzuwachs anhand einer typischen Produktstruktur eines Gesamtfahrzeugs und ordnet die dazugehörigen Gliederungsebenen der Definitions- sowie der Konzeptphase zu. So kommt die Definitionsphase mit vergleichsweise wenigen Komponenten und einer flachen Gliederung aus. Mit zunehmendem Projektfortschritt werden die Modelle allerdings detaillierter und Module, die zuvor mit einem vereinfachten Modell dargestellt wurden, gliedern sich nun in mehrere Komponenten. Dieser Prozess setzt sich fort, bis zum Abschluss der Konzeptphase alle Komponenten in ihre tatsächlichen Einzelteile aufgeschlüsselt sind und die Produktstruktur größtenteils feststeht. Wird davon ausgegangen, dass jedes Modell einer Modulebene sich in eine bestimmte Anzahl an Komponenten einer untergeordneten Ebene aufteilt, so steigt die Anzahl an Komponenten innerhalb dieses Detaillierungsprozesses näherungsweise exponentiell an. Dieser modellhafte Anstieg an Komponenten gilt nicht für Baugruppen, die als Übernahmeteile, bereits in einer frühen Entwicklungsphase bis in ihre Einzelteile definiert sind, wie beispielsweise Motoren oder Getriebe.

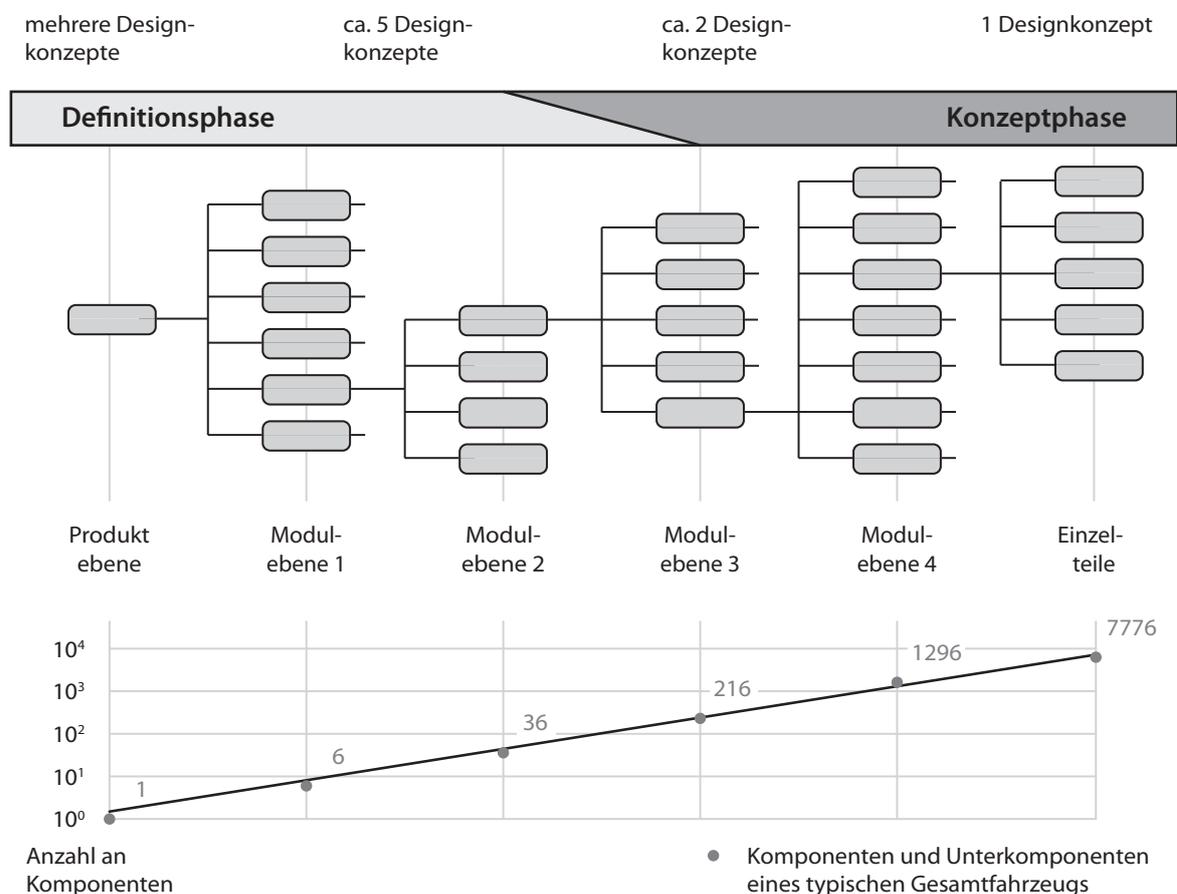


Abbildung 4.1: Anstieg der Anzahl der Komponenten im Laufe der Entwicklung, zum Teil aus [Ges01, Her07].

Der dargestellte Entwicklungsablauf entspricht im Wesentlichen einer *Top-down*-Modellierungsstrategie, die ausgehend von den Ergebnissen einer übergeordneten Gliederungsebene auf die Eigenschaften der enthaltenen Komponenten schließt. Demnach beschäftigt sich die Definitionsphase vor allem mit Themenstellungen auf Ebene des Gesamtfahrzeugs sowie der enthaltenen

Hauptbaugruppen, während in der Konzeptphase der Detaillierungsgrad bis hin zum Einzelteil wächst. Dadurch kommt es in der Konzeptphase zu einem massiven Anstieg an Konstruktions-tätigkeiten, die zunehmend von Funktions- und Produktionsanforderungen getrieben werden. Neben der Modellverfeinerung durch die Aufteilung der Module in ihre Einzelteile steigen daher auch die Ansprüche an die Modellqualität. Die Geometriemodelle werden daher um Details wie etwa Verrundungen ergänzt und berücksichtigen bereits Herstellverfahren, wie beispielsweise Entformschrägen für Tiefziehteile oder Schweißpunkte für Bleche. Parallel zur detaillierteren Geometriemodellierung steigt gleichfalls die Modelltiefe in der Funktionsauslegung. Dadurch können einerseits die Vorhersagequalität erhöht und andererseits zusätzliche Einflüsse untersucht werden. Darüber hinaus steigt in der Konzeptphase ebenfalls die Anzahl an untersuchten Lastfällen, sodass der Aufwand für die Funktionsauslegungen und Berechnungen zunimmt.

Neben der steigenden Produktkomplexität aufgrund der zunehmenden Modelltiefe und des höher werdenden Detaillierungsgrades ist die Konzeptphase ebenfalls durch eine Vielzahl an unterschiedlichen Variantenstudien geprägt. So stehen bei der Ausarbeitung von technischen Lösungen in der Regel mehrere Konzepte zur Auswahl. Diese unterschiedlichen Varianten gilt es zu konkretisieren, zu bewerten und den aussichtsreichsten Ansatz weiter zu verfolgen. Daher kommt es gerade in der frühen Entwicklungsphase, die noch relativ viele Freiheitsgrade für Varianten bietet, zu einem parallelen Bearbeiten von konkurrierenden Konzepten. Im Laufe des Entwicklungsprozesses sinkt dann die Anzahl der Varianten, während es gleichzeitig zu einer Festlegung der Produkteigenschaften kommt. Besonders markant zeigt sich die kontinuierliche Verringerung an Konzepten im Fahrzeugdesign. Obwohl es bis zum endgültigen *Design-Freeze* noch zu Detailänderungen und Optimierungen kommt, wird im Laufe der Definitions- und Konzeptphase ausgehend von mehreren Varianten ein finales Designkonzept ausgewählt, Abb. 4.1.

Eine weitere Charakteristik der Konzeptphase besteht in den sich schnell ändernden Datenständen. Angesichts der Aufgliederung von Entwicklungsaufgaben, die von verschiedenen Personen und Abteilungen parallel bearbeitet werden, kommt es zu oftmaligen Modifikationen von Modellen. Erschwerend wirkt sich auch das Arbeiten mit verschiedenen Varianten aus, welches ebenfalls mit häufigen Modelländerungen verbunden ist. Darüber hinaus führt die vorherrschende *Simultaneous Engineering*-Arbeitsweise zu einer zusätzlichen Parallelisierung des Entwicklungsprozesses, da Entwicklungstätigkeiten bereits frühzeitig und auf Basis von vorläufigen oder unvollständigen Eingangsdaten starten. Diese Umstände führen speziell in der Konzeptphase zu einer Vielzahl an unausgereiften Daten. Aus diesem Grund kommt es selbst bei einer zusammenbauorientierten *Design-in-Context*-Arbeitsweise im Konstruktionsprozess vielfach zu Anpassungen und Änderungen der CAD-Modelle sowie zu einem erhöhten Abstimmungsaufwand.

Die angeführten Eigenschaften der Konzeptphase führen gerade im Bereich der Konstruktion zu einem schnell wechselnden Umfeld, mit sich vielfach ändernden Eingangsdaten und einem parallelen Bearbeiten von Varianten. Dennoch werden beträchtliche Ansprüche an die Resultate gestellt. Diese sollen einen hohen Reifegrad besitzen und sich infolgedessen in der weiteren Entwicklung nur mehr geringfügig ändern. Um diesen Ansprüchen zu genügen, kommt wissensbasierten Konstruktionswerkzeugen und insbesondere *Templates* eine große Bedeutung

zu. Gerade letztere bieten die Möglichkeit für eine effiziente und vor allem anpassungsfähige Arbeitsweise bei Konstruktionstätigkeiten. Aus diesem Grunde kommen *Templates* in den Konstruktionsprozessen der Konzeptphase zunehmend zum Einsatz. Allerdings handelt es sich bei den eingesetzten *Templates* zumeist um nicht-kommerziell verfügbare Eigenentwicklungen der Industrie. Daher können die Qualität der *Templates* und in weiterer Folge auch die Einsatzmethoden in den jeweiligen Unternehmen stark voneinander abweichen. Außerdem führt die wachsende Detailtiefe in der Konzeptphase zu einem breiten Betätigungsfeld, sodass im Vergleich zur Definitionsphase weitaus mehr unterschiedliche *Templates* zur Verfügung stehen.

4.2.3 Templates in der Konzeptphase

In der Konzeptphase können prinzipiell alle *Template*-Kategorien verwendet werden, um den Konstruktionsprozess zu unterstützen. Jedoch eignen sich insbesondere *Geometrie-* und *Funktionstemplates* für einen Wissenstransfer in diese Phase, Abb. 3.14. Es wird zwar versucht, durch den Einsatz von *Strukturtemplates* im Laufe der Konzeptphase ebenfalls frühzeitig einen festgelegten Modellaufbau sicherzustellen, aber aufgrund der sich schnell ändernden und unausgereiften Daten wird oftmals mehr Wert auf einen flexiblen und schlanken Modellaufbau gelegt als auf die stringenten Vorgaben der *Strukturtemplates*. Diesen Vorgaben wird zumeist erst gegen Ende der Konzeptphase sowie während der Vorentwicklungsphase Folge geleistet, da zu diesem Zeitpunkt bereits alle wesentlichen Variantenentscheidungen getroffen sind und der Fokus vermehrt auf Detailoptimierungen und die Produktion gelegt wird. In diesen Abschnitten steigt der Nutzen und somit auch die Bedeutung von *Strukturtemplates*.

Im Bereich der *Geometrietemplates* können gerade *Matingtemplates* in der Konzeptphase ihre Vorteile ausspielen. So lassen sich mit *Skelett-* oder *Adaptermodellen* einerseits Eingangsdaten rasch ändern und andererseits können Varianten leicht erstellt und ausgetauscht werden. Dadurch bieten *Matingtemplates* die erforderliche Flexibilität und decken bei entsprechender Aufbereitung eine weite Bandbreite an möglichen Eingaben ab. Infolgedessen haben *Geometrietemplates* und insbesondere *Matingtemplates* einen hohen Durchdringungsgrad in der Konzeptphase und bieten ein breites Spektrum an Anwendungen. Beispiele für industriell eingesetzte *Geometrietemplates* liefern unter anderem die Arbeiten von *Haslauer*, *Haasis* oder *Reuter* [Has05, HAR06, RG08]. Bei allen drei Ansätzen wird der Modellierungsaufwand in der Geometrieerstellung durch den Einsatz von flexiblen Vorlagen reduziert. Diese Arbeiten zeigen ebenfalls, dass die Verwendung von *Geometrietemplates* in der Industrie weit verbreitet und vorangeschritten ist.

Im Gegensatz dazu haben sich *Funktionstemplates* in einem geringeren Ausmaß im Entwicklungsprozess der Konzeptphase etabliert und haben in diesem Abschnitt, obwohl sie für den Einsatz in einer frühen Entwicklungsphase prädestiniert sind, einen niedrigeren Durchdringungsgrad als *Geometrietemplates*. So kommen abseits der bereits angeführten Gesamtfahrzeug-Entwurfssysteme zwar durchaus Beispiele für entwicklungsbegleitende Auslegungswerkzeuge, wie etwa durch *Franzen* oder *Berkan* beschrieben [Fra08, BABS10], zum Einsatz, jedoch bleiben diese Anwendungen im Vergleich zu den *Geometrietemplates* in der Minderheit.

Ein Grund für ihre seltenere Verwendung liegt in der zum Teil aufwändigen Erstellung und Wartung der *Funktionstemplates*. Außerdem konkurrieren CAD-basierte *Templates* mit CAE-Werkzeugen und -Softwareapplikationen, mit denen ebenfalls Funktionsauslegungen sowie Auslegungen für Folgeprozesse durchgeführt werden können. Diese Berechnungsprogramme sind bei komplexen Aufgaben, wie etwa Betriebsfestigkeits- oder Crashsimulationen, den Simulationspaketen innerhalb der CAD-Programme in der Regel überlegen. Außerdem bieten sie häufig keine speziellen Schnittstellen zu CAD-Systemen, sodass die Einbindung in *Funktionstemplates* erschwert wird. Neben diesen systembedingten Abgrenzungen zwischen Konstruktions- und Berechnungswerkzeugen – die trotz der Bestrebungen für eine engere Verbindung dieser beiden Domänen immer noch vorhanden sind – kommt es in der Konzeptphase zumeist auch zu einer organisatorischen und prozessbedingten Trennung von Geometrieerzeugung und funktionaler Auslegung. Während beide Aufgaben in der Definitionsphase oftmals innerhalb eines Konzeptteams bearbeitet werden, kommt es beim Übergang in die Konzeptphase in der Regel zu einer Aufspaltung der Entwicklungsaufgaben, die dann in verschiedenen Fachbereichen bearbeitet werden. Infolgedessen übernehmen üblicherweise die Konstruktionsabteilungen die Geometriemodellierung und die Berechnungsabteilungen die Funktionsabsicherung und -auslegung. Darüber hinaus beginnt bereits die Produktionsplanung, die fertigungsrelevante Entwicklungstätigkeiten und Auslegungen durchführt. Da jedoch im Gegensatz zum Konstruktionsbereich bei Tätigkeiten in der Funktionsauslegung und in der Produktionsplanung in viel geringerem Ausmaß CAD-Systeme, sondern andere CAX-Werkzeuge zum Einsatz kommen, besteht bei diesen Anwendungen ein geringeres Interesse am Einsatz von CAD-basierten *Funktionstemplates*. Außerdem wird eine Koppelung von CAD- und CAX-Werkzeugen im Rahmen von *Funktionstemplates* oftmals aufgrund von innerbetrieblichen Barrieren hinsichtlich des Software- und Lizenzmanagements erschwert.

Nichtsdestoweniger liefern *Funktionstemplates* gemeinsam mit *Geometrietemplates* die Werkzeuge, um in effizienter Art und Weise eine ausgereifte Konzeptgeometrie zu erstellen. Gerade die Kombination dieser *Template*-Kategorien kann dabei einen wichtigen Beitrag für eine konstruktionsbegleitende Auslegung und Absicherung von Produkteigenschaften leisten. Ein wesentlicher Vorteil liegt in der Qualität der resultierenden Konzeptgeometrien, die bereits funktionalen Anforderungen entsprechen, ohne dass auf Ergebnisse von externen Berechnungen zurückgegriffen werden muss. Demzufolge können der vermehrte Einsatz von *Funktionstemplates* sowie die Verbindung von *Funktions-* und *Geometrietemplates* als Methoden angesehen werden, mit denen sich die Lücke zwischen Geometriemodellierung und Funktionsauslegung in der Konzeptphase schließen lässt. Dadurch kann der Bedarf an einer CAD-basierten Vorauslegung, der insbesondere in diesem Abschnitt der Produktentwicklung besonders hoch ist, effizient gedeckt werden.

4.3 CAD-basierte Vorauslegung mittels Templates

4.3.1 Bedarf einer CAD-basierten Vorauslegung in der Konzeptphase

Die organisatorische Aufteilung der Entwicklungstätigkeiten in unterschiedliche Fachbereiche, wie etwa einen Konstruktions- oder einen Berechnungsbereich, wird am Beginn der Konzeptphase infolge des starken Zuwachses an unterschiedlichen und immer detaillierter werdenden Aufgaben notwendig. Dennoch erweist sich diese Trennung aufgrund des vorherrschenden Entwicklungsumfelds, mit dem massiven Anstieg an Komponenten, dem steigenden Detaillierungsgrad sowie der Vielzahl an Varianten, mitunter als problematisch. So können sowohl die zusätzlichen organisatorischen Schnittstellen als auch die dynamischen Bedingungen der Konzeptphase trotz eines PDM-unterstützten Entwicklungsprozesses zu unterschiedlichen Datenständen in den jeweiligen Fachbereichen führen. Gerade die gegenseitigen Abhängigkeiten zwischen Konstruktion, Berechnung, Design und Packaging resultieren während dieser frühen Phase in parallelen Entwicklungen, deren Ergebnisse abgeglichen und aufeinander abgestimmt werden müssen. Eine entscheidende Rolle nimmt hierbei der Konstruktionsbereich ein, dessen Produktmodelle neben den Geometriedaten eine umfassende Beschreibung der Produkteigenschaften enthalten und die dadurch als zentrale Datenmodelle verwendet werden können.

In Folge des sich rasch ändernden Entwicklungsumfelds und der nebeneinander ablaufenden Auslegungen befinden sich die Produkt- und Geometriedaten in einem ständigen Wandel, was wiederum eine Synchronisation von parallelen Entwicklungen erschwert. Abb. 4.2 veranschaulicht diese Problematik für den Teilumfang eines Produktes anhand eines *Simultaneous Engineering*-Entwicklungsprozesses, wie er in der Konzeptphase üblich ist. Beispielhaft wird eine Geometrieauslegung auf Modulebene betrachtet, die für gewöhnlich im Konstruktionsbereich mit Hilfe von CAD-Werkzeugen durchgeführt wird. Die Funktionsauslegung erfolgt im Zuge der Berechnung und benötigt unter anderem Geometrieinformationen als Eingangsdaten. Für die Auslegung eines Lastfalls werden also die CAD-Daten zu einem Zeitpunkt t_1 für die Simulationen und Optimierungen herangezogen. Bis diese Auslegungsschritte jedoch Ergebnisse liefern, die dann in den Konstruktionsprozess zurückgeführt werden und dort für allfällige Geometrieadaptationen sorgen, vergeht die Bearbeitungszeit Δt . Diese Zeitspanne wird üblicherweise für die Modellerstellung (*Preprocessing*), die eigentliche Berechnung (*Solving*) und die Auswertung der Ergebnisse (*Postprocessing*) benötigt und hängt maßgeblich von der Komplexität der Aufgabenstellung ab [Her07].

Während im Konstruktionsprozess auf die Ergebnisse aus der Funktionsauslegung gewartet wird, kommt es zu einem regen Austausch zwischen den Konstruktionsumfängen und der restlichen Produktentwicklung. So benötigen einerseits andere Entwicklungsbereiche aktuelle Geometriedaten, um dort beispielsweise weitere Lastfälle abzusichern oder um eine geometrische Integration durchzuführen. Andererseits fließen neue Anforderungen aus dem Packaging, dem Design, der Konstruktion von umliegenden Bauteilen oder Ergebnisse von anderen Funktionsauslegungen in die Geometriemodellierung ein, sodass sich die Bauteilgeometrie ständig weiterentwickelt. Aufgrund der vergleichsweise hohen Freiheitsgrade in der Konzeptphase können die Anpassungen des Geometriemodells durchaus umfangreich sein. Dies hat zur Folge, dass die

Berechnungsergebnisse auf Basis des Geometriestands von t_1 aufgrund der sich fortschreitend verändernden Geometriedaten bei ihrer Rückführung zum Zeitpunkt $t_1 + \Delta t$ zumeist schon veraltet sind und bei massiven Änderungen der Geometrie sogar unbrauchbar werden. Das Gleiche gilt für eine neuerliche Auslegung anhand eines aktuelleren Geometriestands bei t_2 . Zwar kann die Bearbeitungszeit Δt im *Preprocessing* durch ein bereits vorhandenes Simulationsmodell reduziert werden, jedoch ändert das nichts an der Problematik des Zeitversatzes zwischen der Bereitstellung von Geometriedaten und der Ergebnisrückführung. So hat sich die Geometrie bis zur Ergebnisrückführung bei $t_2 + \Delta t$ ebenfalls weiterentwickelt [Her07].

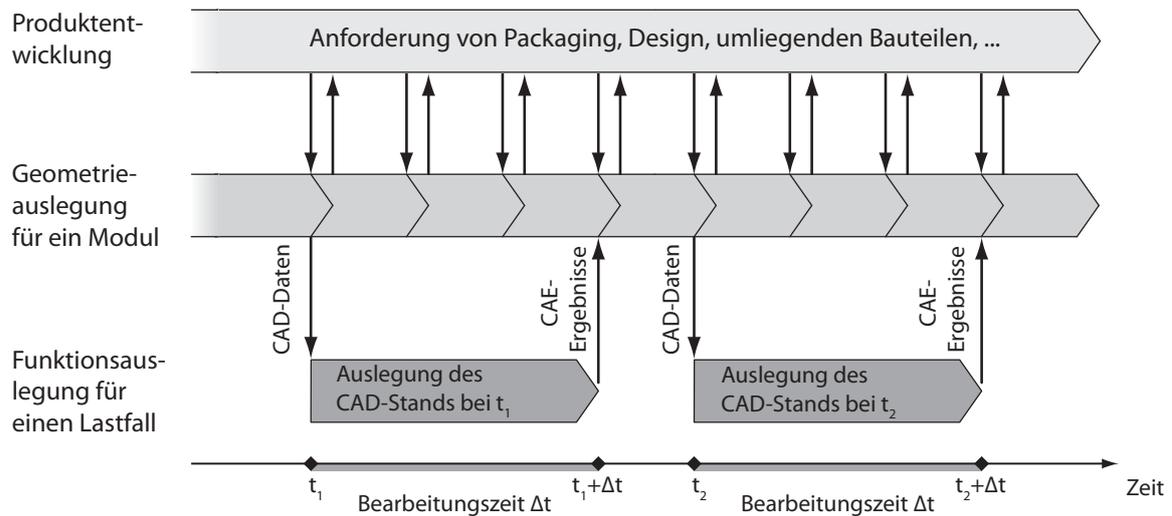


Abbildung 4.2: Parallel ablaufenden Geometrie- und Funktionsauslegung in der Konzeptphase, nach [Her07]²⁶.

Der Zeitversatz zwischen der Übergabe von vorläufigen Geometrieinformationen und dem Informationsrückfluss aus Auslegungen und Absicherungen ist charakteristisch für eine parallel ablaufende Geometrie- und Funktionsentwicklung. Besonders nachteilig wirkt sich dieser Zeitverzug in der Konzeptphase aus, da es hier zu oftmaligen und mitunter umfangreichen Adaptionen des Produkts kommt. So gelten in diesem Abschnitt vergleichsweise weiche Vorgaben aus dem Rahmenheft, die einen breiten Spielraum für Änderungen zulassen. Gleichmaßen kann es zu großen Veränderungen des Fahrzeugdesigns kommen, die wiederum einen beträchtlichen Einfluss auf die zu Grunde liegenden Geometriedaten haben. Außerdem führen Variantenuntersuchungen und die fortschreitende Detaillierung zu einer hohen Änderungsrate des Produkts. Es kann daher nicht davon ausgegangen werden, dass sich die Geometriedaten während einer Funktionsauslegung in einem nur vernachlässigbaren Ausmaß ändern. Diese oftmaligen Anpassungen sind in weiterer Folge auch der Ausgangspunkt für unterschiedliche Entwicklungsstände in den diversen, zumeist nicht gleichzeitig startenden Auslegungsschleifen. Im Gegensatz dazu kommt es in der seriennahen Entwicklung einerseits nur mehr zu geringfügigen Geometrie Anpassungen und andererseits vollziehen sich diese Adaptionen in längeren Zeiträumen, sodass Berechnungsergebnisse bei ihrer Rückführung in den Konstruktionspro-

²⁶In Abb. 4.2 werden die Geometrie Anpassungen aufgrund geänderter Anforderungen als Pfeilspitze im Strang „Geometrieauslegung für ein Modul“ dargestellt.

zess oftmals nicht aufgrund von massiven Überarbeitungen der Geometrie veraltet sind. Die Herausforderungen der seriennahen Entwicklung liegen weniger in kurzfristigen und umfangreichen Änderungen der Geometriedaten sondern vielmehr in der Integration von Lieferanten oder der Fokussierung auf produktionstechnische Aspekte. Überholte und demnach eingeschränkt brauchbare Berechnungs- und Auslegungsergebnisse infolge von Geometrieänderungen treten somit vermehrt in einem frühen Entwicklungsstadium auf.

Die hier beschriebene Problematik einer zeitverzögerten Ergebnisrückführung im Spannungsfeld einer parallel ablaufenden Geometrie- und Funktionsentwicklung in der Konzeptphase ist aus derzeitiger Sicht nicht zufriedenstellend gelöst. Zwar gibt es – wie beispielsweise durch *Herfeld* beschrieben – eine Vielzahl an Ansätzen, die das Ziel verfolgen, mit verbesserten Prozessen, Methoden und Werkzeugen die Geometriemodellierung und die Funktionsauslegung stärker aneinander zu binden und besser zu synchronisieren, allerdings kann die vorherrschende Lücke nicht vollständig geschlossen werden [Her07]. Demnach besteht gerade im Konstruktionsprozess infolge der sich schnell ändernden Datenbasis weiterhin der Bedarf an Auslegungsmethoden, mit denen in kurzer Zeit ausreichend genaue Ergebnisse vorliegen. Diese werden benötigt, um die Auswirkungen von veränderten Anforderungen oder neuen Eingangsdaten, wie etwa einem adaptierten Fahrzeugdesign, umgehend bewerten zu können und mögliche Schwachstellen frühzeitig zu identifizieren. In diesem Punkt reichen gerade für Variantenentscheidungen in der Regel funktional abgesicherte qualitative Aussagen aus. Speziell für die Forderung nach schnellen, konstruktionsnahen und aussagekräftigen Ergebnissen erweisen sich konventionelle Auslegungsschleifen jedoch als zu langwierig und träge. Demgegenüber bietet der Ansatz einer CAD-basierten Vorauslegung eine mögliche Verbesserung der derzeitigen Situation. Hierbei werden die Fähigkeiten von modernen, wissensbasierten CAD-Systemen und insbesondere von *Templates* genutzt, um mit Hilfe von wissensbasierten Konstruktionsmethoden sowohl eine rasche CAD-basierte Funktionsauslegung als auch eine effiziente Geometriemodellierung zu ermöglichen.

4.3.2 Vorgehensmodell für eine CAD-basierte Vorauslegung

Der Grundgedanke der CAD-basierten Vorauslegung besteht darin, dass in der Konzeptphase die üblicherweise scharfe Trennlinie zwischen Funktionsauslegung- und Geometriemodellierung aufgebrochen wird und in den Konstruktionsprozess neben der Erstellung von Geometriedaten auch eine konstruktionsbegleitende, vorläufige Funktionsauslegung und -absicherung integriert wird. Die Ergebnisse dieser Auslegungen liefern gleichzeitig die Grundlage für Geometriedaten, die somit schon in dieser Phase funktionalen Anforderungen genügen. Diese hochwertigen Geometriemodelle dienen weiterhin als Eingangsdaten für Folgeprozesse, wozu vor allem die konventionelle Funktionsauslegung in den Berechnungsabteilungen zählt. Aus diesem Grund wird auch bewusst der Begriff der Vorauslegung gewählt, da es zwar zu einer vorläufigen Auslegung und Absicherung kommt, die Hauptverantwortung für eine funktionale Auslegung jedoch weiterhin innerhalb der Berechnungsabteilungen liegt. Dadurch stellt die CAD-basierte Vorauslegung in weiten Teilen eine Ergänzung und keine radikale Veränderung des konventionellen Entwicklungsprozesses dar. Der vorliegende Ansatz kann vielmehr als eine Anwendung von

wissensbasierten Konstruktionsmethoden angesehen werden, welche bei der hier vorgestellten Umsetzung umfassend auf *Templates* als Wissensträger zurückgreift.

Der im Zuge der vorliegenden Arbeit entwickelte Ablauf einer CAD-basierten Vorauslegung wird in Abb. 4.3 anhand eines Vorgehensmodells veranschaulicht. Generell bilden Vorgehensmodelle wichtige Elemente einer Handlungsfolge ab, die zum einen als Hilfsmittel zum Planen und Kontrollieren von Prozessen dienen und zum anderen ein systematisches Vorgehen abbilden [Lin06]. Das hier vorgestellte Vorgehensmodell kann somit als Leitfaden für den Ablauf einer CAD-basierten Vorauslegung angesehen werden und unterteilt sich in vier Phasen: die Bereitstellung von Eingangsdaten, die funktionale Auslegung, die Geometriemodellierung und die Veröffentlichung der Ergebnisse. Die Werkzeuge für die funktionale und geometrische Auslegung umfassen *Templates*, die abseits des eigentlichen Entwicklungsprozesses vorhandenes Wissen zur Verfügung stellen. Die Erstellung, Wartung und Bereitstellung der *Templates* wird in der Regel von einer zentralen, projektunabhängigen Organisationseinheit übernommen, in der zumeist auch eine übergeordnete Entwicklung von konstruktionsspezifischen Werkzeugen und Methoden erfolgt. Die Bedienung der *Templates* obliegt dem Anwender, der zusätzlich zum formalisierten Wissen innerhalb der *Templates* Erfahrung und Kreativität einbringt. Dadurch sorgt er unter anderem für eine Plausibilisierung der Ergebniswerte sowie für eigenständige Konstruktionslösungen.

Werden die einzelnen Phasen des Vorgehensmodells betrachtet, so steht am Beginn einer Vorauslegung die Bereitstellung der Eingangsdaten. Hierbei kann es sich prinzipiell um CAD-Daten oder um externe Daten, die außerhalb eines CAD-Systems vorliegen, handeln. Unter CAD-Daten werden jene Informationen verstanden, die üblicherweise in einem CAD-Programm erzeugt werden, wie etwa Geometriedaten oder Parameter. Außerdem werden Daten eingeschlossen, die einen externen Ursprung haben, jedoch üblicherweise in ein CAD-System importiert werden und dort vorliegen, wie etwa die Designflächen eines Fahrzeuges. Demgegenüber umfassen externe Daten alle weiteren Informationen, die nicht innerhalb der CAD-Umgebung zur Verfügung stehen. Hierunter fallen unter anderem Normen, Kennwerte, wie etwa die Wärmeleitfähigkeit eines Werkstoffs, Zielvorgaben oder Gesetzesvorschriften. Die Daten werden als Eingaben für die Funktionsauslegung in der zweiten Phase benötigt und unterscheiden sich abhängig von der jeweiligen Auslegung. Eine wesentliche Herausforderung in der Konzeptphase besteht oftmals im Umstand, dass die benötigten Eingabedaten nur unvollständig vorliegen. Aus diesem Grund muss gegebenenfalls auf das Wissen von vorangegangenen Projekten oder die vorhandene Erfahrung des Anwenders zurückgegriffen werden, um mit Annahmen und Standardwerten lückenhafte Eingaben zu ergänzen.

Sobald ein vollständiger Satz an Eingangsdaten bereitsteht, kann eine Auslegung unter Zuhilfenahme von *Funktionstemplates* innerhalb der CAD-Systemumgebung durchgeführt werden. Die *Templates* enthalten Expertenwissen und Modelle, die mit wenigen und oftmals ungenauen Eingabedaten effizient hinreichend genaue Ergebnisse erzeugen. Die Ergebnisse dieser funktionalen Vorauslegung können sowohl Geometrielemente, wie beispielsweise ein Bauteilskelett, aber auch Parameterwerte oder Kennfelder umfassen. Zur Erleichterung des Arbeitsablaufs sind *Funktionstemplates* oftmals mit Assistenzsystemen gekoppelt, die den Anwender unterstützen. Nach Abschluss einer Funktionsauslegung können die gewonnenen Erkenntnisse zum

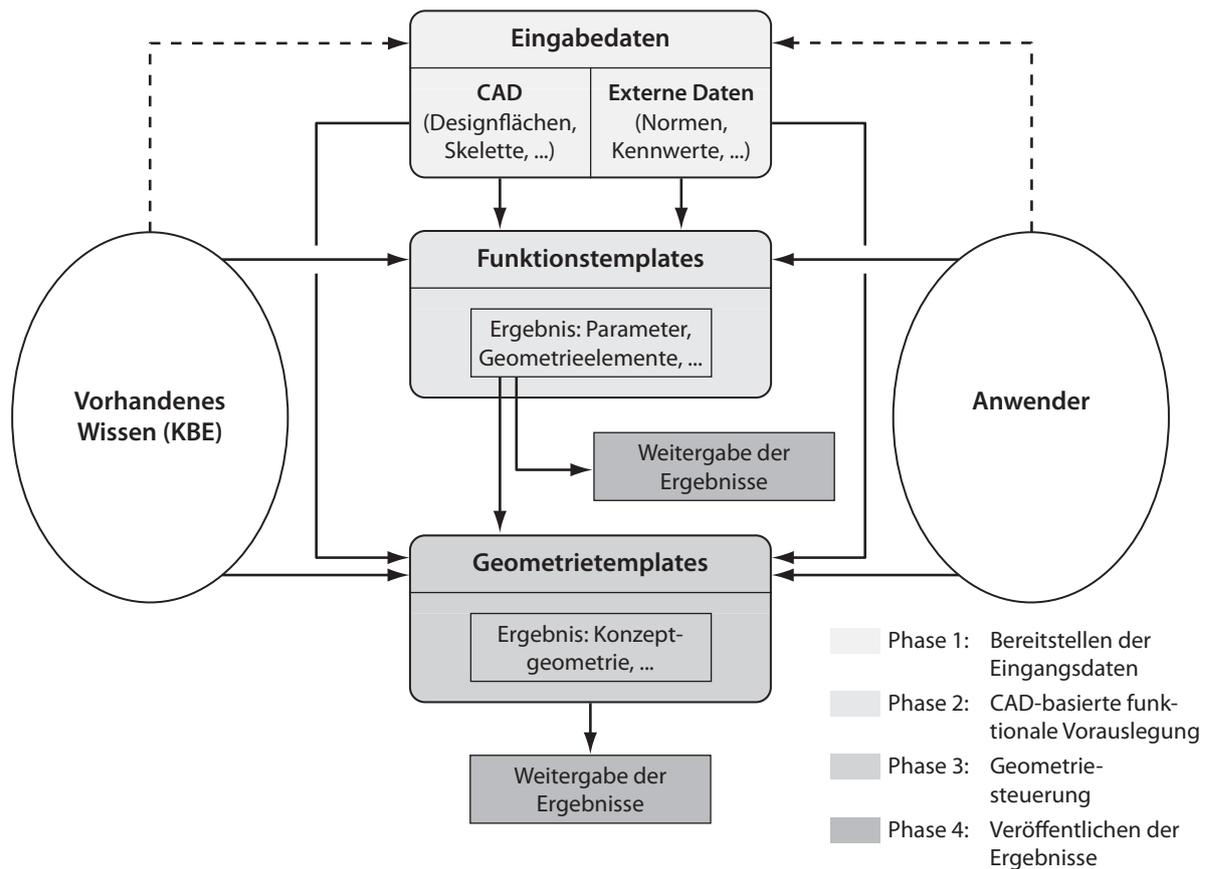


Abbildung 4.3: Vorgehensmodell einer CAD-basierten Vorauslegung unter Zuhilfenahme von *Templates*.

einen dem Entwicklungsprozess zur Verfügung gestellt und zum anderen als Basis für die folgende Geometriemodellierung genutzt werden.

Die Erstellung von Geometriedaten geschieht in der dritten Phase des Vorgehensmodells aus Abb. 4.3. Dieser Abschnitt ist vergleichbar mit einem konventionellen Entwicklungsablauf in der Konzeptphase, bei dem anhand von Vorgaben und Eingangsdaten ein Geometriemodell erstellt wird. Der wesentliche Vorteil im Vergleich zu einer konventionellen Entwicklung liegt im Einbeziehen der Ergebnisse aus der vorangegangenen Funktionsauslegung, sodass die funktionale Qualität der Geometriemodelle in diesem Entwicklungsstadium erhöht wird. Die Modellierung der Geometrie lässt sich sehr erfolgreich mit *Geometrietemplates* unterstützen. Diese liefern Vorlagen für eine Vielzahl an bewährten Bauteilgeometrien und lassen sich effizient steuern und adaptieren. Darüber hinaus können die Skelettgeometrien der *Template*-basierten Funktionsauslegung als direkte Eingaben für die *Skelett*- und *Adaptermodelle* der *Matingtemplates* übernommen werden. Dadurch lässt sich die Geometrie rasch an veränderte Eingangsdaten anpassen. Als Resultat der Geometriemodellierung steht der weiteren Produktentwicklung somit eine hochwertige und anpassungsfähige Konzeptgeometrie zur Verfügung.

Die letzte Phase des Vorgehensmodells umfasst die Veröffentlichung der Ergebnisse und kann sowohl nach der Funktionsauslegung als auch im Anschluss an die Geometriemodellierung erfol-

gen. So liegen nach der funktionalen Auslegung schon erste Resultate vor, mit denen beispielsweise geänderte Eingaben, wie ein adaptiertes Design, bewertet und Schwachstellen identifiziert werden können. Demnach stehen an dieser Stelle bereits Erkenntnisse zur Verfügung, die bei einem konventionellen Entwicklungsprozess erst durch oftmals langwierige Funktionsauslegungen im Berechnungsprozess erzeugt werden. Gerade dieser frühzeitige Informationsgewinn stellt einen der wesentlichen Vorteile einer CAD-basierten Vorauslegung dar. Zusätzlich kann nach Abschluss der Geometriemodellierung auf die Konzeptgeometrie zugegriffen werden und die Modelle können Folgeprozessen, wie etwa dem Berechnungsprozess, zugeführt werden. Da die Geometriedaten bereits auf den vorliegenden Resultaten der vorläufigen Funktionsauslegungen aufbauen, erhöht sich die Wahrscheinlichkeit, dass beispielsweise die Ergebnisse einer Berechnungsschleife nur mehr geringfügige Änderungen an der Geometrie notwendig machen. Im Idealfall werden die Erkenntnisse der Vorauslegung bestätigt und es erfolgt kein Änderungsbedarf. Inwieweit dieser Idealfall erreicht wird, hängt stark von der Ergebnisqualität der Vorauslegung ab, die wiederum wesentlich durch die Modellbildung und die Berechnungsmethode bestimmt wird. So muss bei der Entwicklung von *Funktionstemplates* mitunter auf vereinfachte Modelle oder empirisch ermittelte Daten zurückgegriffen werden, um den Erstellungsaufwand nicht unverhältnismäßig hoch ansteigen zu lassen. Dadurch können zwar qualitative Aussagen getroffen und frühzeitig Schwachstellen erkannt werden, eine Überprüfung der Ergebnisse durch Berechnungsschleifen und im späteren Entwicklungsprozess durch Versuchsergebnisse ist jedoch weiterhin notwendig.

Ein weiterer Aspekt bei der Bereitstellung von Ergebnissen liegt im Informationsrückfluss an jene Organisationseinheit, die für die Entwicklung und Wartung der *Templates* verantwortlich ist. Wie generell bei Werkzeugen für eine wissensbasierte Konstruktion liefern Erkenntnisse aus der Anwendung die Grundlage für die Verbesserung und die Aktualität der *Templates*. Darüber hinaus lässt sich bei einer Rückführung von Ergebnissen aus der Vorauslegung und aus der Überprüfung in einer darauf folgenden Berechnungs- oder Versuchsschleife die Aussagequalität der Funktionsauslegung evaluieren.

Der vorgestellte Ansatz einer CAD-basierten Vorauslegung bietet im Konstruktionsprozess die Möglichkeit, mit Hilfe von wissensbasierten Konstruktionsmethoden vorzeitig Erkenntnisse aus funktionalen Fragestellungen zu erlangen. Damit kann dem Bedarf einer konstruktionsnahen Auslegung in der Konzeptphase begegnet werden. Darüber hinaus wird ein durchgängiger Prozess geschaffen, der zu einer flexiblen und hochwertigen Konzeptgeometrie führt. Die CAD-basierte Vorauslegung bietet demnach eine Reihe von Vorteilen. Allerdings beschränkt sich das Anwendungsgebiet dieses Ansatzes größtenteils auf die Konzeptphase. Daher werden an dieser Stelle zum einen die Vorteile der CAD-basierten Vorauslegung beleuchtet und zum anderen wird auf die Randbedingungen und Grenzen dieser Vorgehensweise eingegangen.

4.3.3 Vorteile einer CAD-basierten Vorauslegung im Entwicklungsprozess

Der Einsatz einer CAD-basierten Vorauslegung bietet im Vergleich zu einer konventionellen Produktentwicklung eine Vielzahl an Vorteilen und Verbesserungen. Hierzu zählt ein frühzeitiger Zuwachs an Produktwissen, der zumindest qualitative Aussagen zulässt und mit dem

potentielle Schwachstellen ersichtlich werden. Außerdem verbessert sich die funktionale Qualität der Geometriemodelle, die als Ausgangsdaten für weitere Auslegungen zur Verfügung stehen. Aus der Sichtweise des Entwicklungsprozesses kann unter diesen Voraussetzungen die Entwicklungszeit reduziert werden, ohne dass es zu Minderungen der Qualität kommt. Diese Vorteile werden bei der Gegenüberstellung eines konventionellen Entwicklungsablaufs und eines Entwicklungsprozesses, bei dem es zur Anwendung einer CAD-basierten Vorauslegung kommt, verdeutlicht.

Abb. 4.4 zeigt demnach einen typischen Ablauf in einem konventionellen Entwicklungsprozess und legt seinen Fokus auf eine Modulebene. Der dargestellte Prozess beginnt zum Zeitpunkt t_1 mit einer neuen Anforderung aus einer übergeordneten Ebene der Produktentwicklung. Ausgehend von dieser Anforderung wird die Bearbeitungszeit Δt benötigt, an deren Ende es zu einer Rückführung von Erkenntnissen aus der Modulentwicklung kommt. Die Entwicklung auf Modulebene gliedert sich in die Anpassung der CAD-Daten, die in den Konstruktionsabteilungen erfolgt, und eine anschließende Funktionsauslegung in den Berechnungsabteilungen. Die Geometriemodellierung stützt sich üblicherweise auf die Erfahrung und Kreativität des Bearbeiters und wird aufgrund der Trennung zwischen Geometriemodellierung und Funktionsauslegung nur selten durch konstruktionsbegleitende Auslegungswerkzeuge unterstützt. Daher liegt es zumeist im Ermessen des Konstrukteurs, inwieweit sich eine neue Anforderung in den Geometriedaten niederschlägt. Nach Änderungen der CAD-Daten lassen sich lediglich Schwachstellen bezüglich der geometrischen Integration bewerten; die Auswirkung auf die Funktion muss in weiterer Folge in einer Auslegungsschleife überprüft werden. Nach der Funktionsauslegung zum Zeitpunkt $t_1 + \Delta t$ stehen die Ergebnisse der Auslegungsschleife zur Verfügung und bilden die Entscheidungsgrundlage für die weitere Entwicklung. So kann beispielsweise die Geometrie, entsprechend den Erkenntnissen aus dem Berechnungsprozess angepasst werden oder eine Änderung der Anforderung wird notwendig, da die Erfüllung von Vorgaben nicht möglich ist. In jedem Fall werden zu diesem Zeitpunkt die Auswirkungen der neuen Anforderung sowohl aus geometrischer als auch aus funktionaler Sicht deutlich.

Wie eingangs in Kap. 4.3.1 behandelt, liegt eine Schwachstelle dieses konventionellen Entwicklungsprozesses in der langen Bearbeitungszeit Δt . Demgegenüber kann mit einer CAD-basierten Vorauslegung Δt drastisch reduziert werden. Diese Verbesserung lässt sich in Abb. 4.5 erkennen. Dort wird wiederum ein Entwicklungsprozess auf Modulebene dargestellt, der mit einer neuen Anforderung beaufschlagt wird. Im Gegensatz zu einer konventionellen Entwicklung kommt es bereits während des Konstruktionsprozesses zu einer CAD-basierten funktionalen Vorauslegung, die frühzeitig Erkenntnisse liefert. Demnach lassen sich die Bearbeitungszeit für erste Ergebnisse reduzieren und anhand dieser Ergebnisse Maßnahmen ergreifen. Zwar ist die Ergebnisqualität im Vergleich zu einer konventionellen Auslegungsschleife geringer, allerdings besteht ein wesentlicher Zeitvorteil. So können parallele Entwicklungsabläufe bereits auf die Ergebnisse der CAD-basierten funktionalen Vorauslegung zugreifen und mögliche gegenläufige Entwicklungen an anderen Stellen vermieden werden. Außerdem können durch diese frühzeitige Vorauslegung Berechnungsschleifen eingespart werden, falls die vorliegenden Ergebnisse eine Anpassung der Anforderungen notwendig machen. Anderenfalls kann – wie beispielhaft in Abb. 4.5 dargestellt – weiterhin mit einer Geometriemodellierung und einer konventionel-

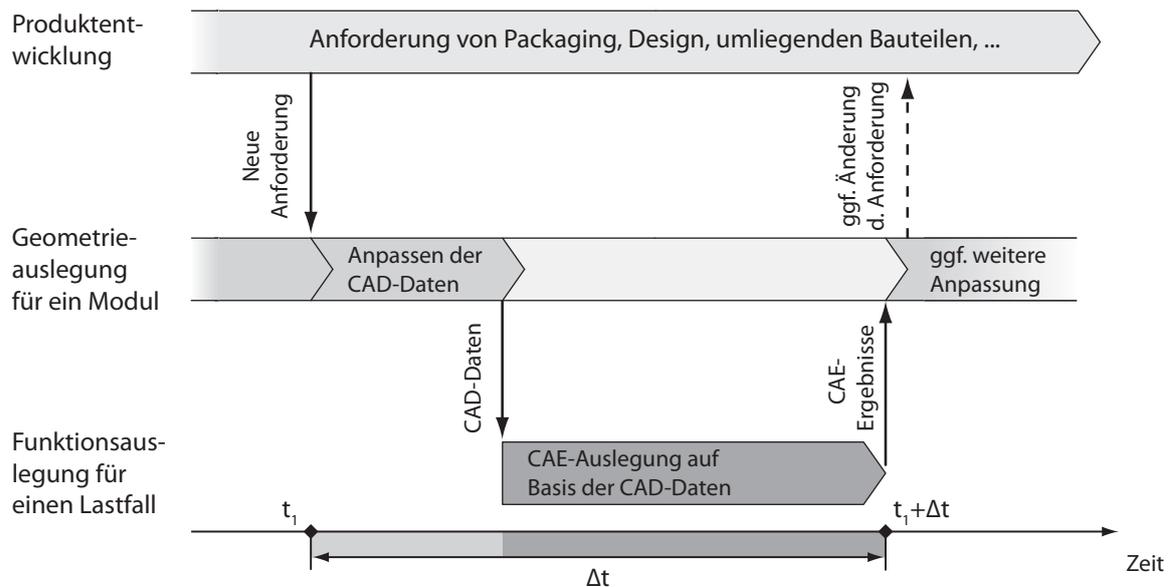


Abbildung 4.4: Konventioneller Entwicklungsprozess mit separater Geometriemodellierung und Funktionsauslegung.

len Auslegungsschleife fortgefahen werden. Mit der Berechnungsschleife im Anschluss an eine Vorauslegung sollen vor allem die vorliegenden Ergebnisse überprüft und gegebenenfalls optimiert werden. Durch die zusätzliche Vorauslegung verlängert sich demnach zwar die Zeitspanne vom Auftreten einer neuen Anforderung bis zum Vorliegen der Simulationsergebnisse aus dem Berechnungsprozess, allerdings kann bereits frühzeitig auf die Resultate und Erkenntnisse der Vorauslegung zugegriffen werden. Darüber hinaus weisen die CAD-Daten vor dem Berechnungsdurchlauf eine höhere funktionale Qualität auf, sodass zumeist ein geringerer Änderungsbedarf besteht und beispielsweise auf zusätzliche Optimierungsschleifen verzichtet werden kann.

Ein weiterer Vorteil der CAD-basierten Vorauslegung besteht in der Verwendung des CAD-Systems als vorherrschendes Auslegungswerkzeug. So verweilt der typische Anwender aus dem Konstruktionsbereich in einer ihm vertrauten CAD-Umgebung, wodurch keine zusätzliche Kenntnis weiterer Programme erforderlich ist. Demnach führt die Verwendung von gewohnten Benutzeroberflächen zu einer erhöhten Akzeptanz in der Anwendung und erleichtert die Implementierung. Die Einführung von Werkzeugen für eine CAD-basierte Vorauslegung umfasst neben dem Einpflegen der notwendigen *Templates* in die Systemlandschaft vor allem auch die Schulung der Mitarbeiter. Da die Konstrukteure in der Regel mit CAD-Systemen vertraut sind, können sich die Schulungsinhalte auf anwendungsspezifische Aspekte konzentrieren. An dieser Stelle ist besonderer Wert auf die Vermittlung von Hintergrundwissen zu legen, welches insbesondere die Ergebnisinterpretation der funktionalen Vorauslegung ermöglicht. So ist es innerhalb der Konstruktionsabteilungen in Folge der zusätzlichen Aufgaben notwendig, entsprechendes Wissen aufzubauen. Dadurch wird einerseits der Konstruktionsbereich aufgewertet und andererseits führt das zusätzliche Wissen zu einem erhöhten Systemverständnis.

Der Einsatz einer CAD-basierten Vorauslegung bietet demnach eine Reihe von Vorteilen und erweist sich insbesondere in der Konzeptphase als eine Möglichkeit, die Produktentwicklung ef-

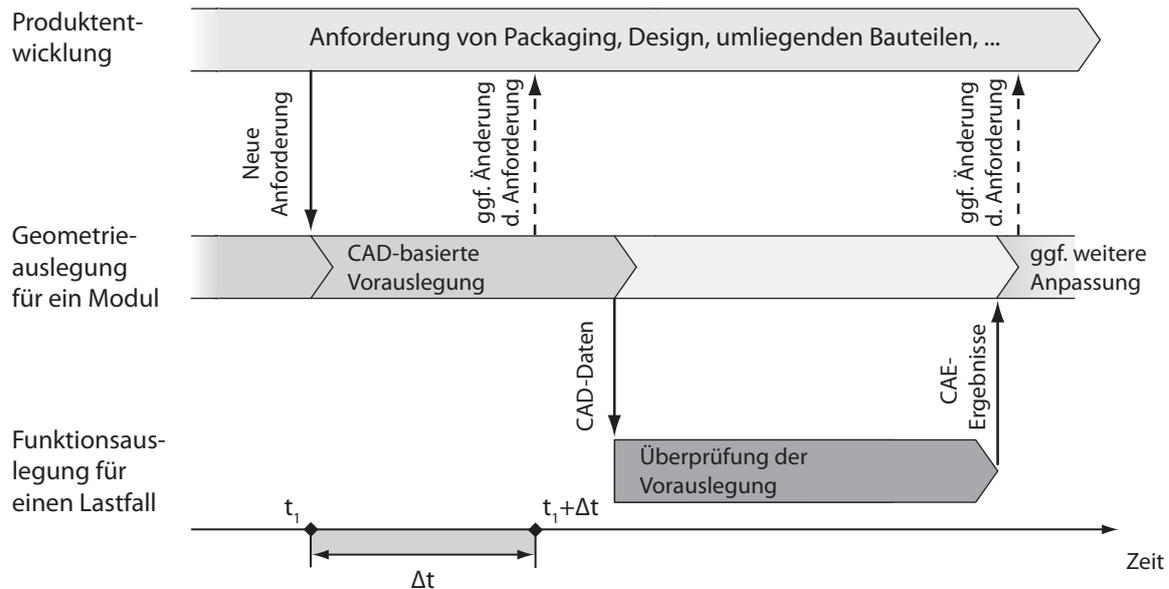


Abbildung 4.5: Entwicklungsprozess mit einer CAD-basierten Vorauslegung.

fizienter zu gestalten. So liefert diese Vorgehensweise in einem Entwicklungsumfeld, das durch eine Fülle an Unsicherheiten und raschen Änderungen geprägt ist, schnelle Ergebnisse und dadurch eine Grundlage für Entscheidungen. Demzufolge steigt im Sinne des *Frontloadings* das Produktwissen in einem frühen Entwicklungsstadium und es können hochwertige Produktdaten erzeugt werden. Der wirksamen Anwendung einer CAD-basierten Vorauslegung stehen allerdings eine Vielzahl an Randbedingungen und Einschränkungen gegenüber, deren Erfüllung als Voraussetzung für eine erfolgreiche Umsetzung gelten.

4.3.4 Randbedingungen und Einschränkungen einer CAD-basierten Vorauslegung

Da es sich bei der CAD-basierten Vorauslegung um eine Anwendung von wissensbasierten Konstruktionsmethoden handelt, gelten ebenfalls die gleichen Einschränkungen und Voraussetzungen, wie sie auch in Kap. 3.2 zusammengefasst werden. Im Fall der CAD-basierten Vorauslegung liegt eine spezielle Einschränkung im begrenzten Einsatzzeitraum. So ist dieser Ansatz abgestimmt auf die Anforderungen einer frühen Entwicklungsphase und eignet sich daher besonders gut während der Konzeptphase. In späteren Entwicklungsabschnitten nehmen jedoch die Vorteile ab, während Schwächen zu Tage treten. Demnach sinkt im weiteren Entwicklungsprozess die Änderungsrate und es kommt vermehrt zu Detailanpassungen anstelle von grundlegenden Änderungen. Dadurch verschiebt sich der Bedarf an frühzeitig verfügbaren Ergebnissen, die eine geringere Genauigkeit aufweisen, hin zu detaillierten Resultaten, die allerdings mit einer langwierigen Erzeugung einhergehen. Darüber hinaus liegen in späteren Phasen des Entwicklungsprozesses hoch detaillierte Geometriemodelle vor, die nur mehr geringfügig geändert werden. Daher besteht auch keine Notwendigkeit für eine flexible Geometrieerzeugung mit *Templates*. Dieser Effekt wird noch verstärkt, da in einer Serienentwicklung bewusst auf wissensbasierte Elemente in CAD-Modellen sowie auf bauteilübergreifende Abhängigkeiten verzichtet

wird, um Geometrieadjustierungen nicht automatisiert über die Grenzen der Verantwortlichkeiten hinweg umzusetzen sondern in einem entsprechenden Freigabe- und Änderungsprozess zu dokumentieren [GNR⁺08]. Nichtsdestoweniger kann eine CAD-basierte Vorauslegung auch in einer seriennahen Entwicklung eingesetzt werden, selbst wenn sie in diesem Abschnitt einen weitaus geringeren Stellenwert besitzt, als in der frühen Entwicklungsphase.

Eine weitere Einschränkung für die Erstellung und den Einsatz einer CAD-basierten Vorauslegung besteht im Aufwand für die Entwicklung von geeigneten Modellen und Berechnungsroutinen zur Funktionsauslegung. Das Ziel des Einsatzes dieser Modelle besteht vordergründig darin, aussagekräftige Ergebnisse in einer kurzen Berechnungsdauer zu generieren. Die Resultate sollen in weiterer Folge bei Variantenentscheidungen sowie bei Grundsatzuntersuchungen unterstützen und hierbei ein qualitatives Systemverhalten darstellen. Darüber hinaus müssen die Berechnungsmethoden mit einer geringen Anzahl an Parametern auskommen, da in einer Konzeptphase in der Regel nur unvollständige und ungenaue Eingabedaten vorliegen. Abhängig von der Komplexität der Auslegung lassen sich unter diesen Voraussetzungen geeignete Minimalmodelle erstellen, mit denen der Zielkonflikt aus einer effizienten, robusten Berechnungsmethode und einer hohen Ergebnisgüte gelöst werden kann. Bei den verwendeten Modellen kann es sich sowohl um physikalische als auch um empirische Modelle sowie um Mischformen dieser beiden handeln. Das Wesen der Minimalmodelle besteht darin, dass diese für die Konzeptphase hinreichend genaue Ergebnisse liefern, sodass zusätzliche Freiheitsgrade oder Parameter nur zu einer geringfügigen Verbesserung der Ergebnisqualität beitragen.

Diese Tendenz veranschaulicht Abb. 4.6, indem qualitativ der Anstieg der Ergebnisgüte in Abhängigkeit der Komplexität der verwendeten Berechnungsmodelle gezeigt wird. Als Beispiele dienen Modelle, die für Untersuchungen der Fahrzeugquerdynamik herangezogen werden. Demnach können bereits mit einem vergleichsweise einfachen Einspurmodell, das über zwei Freiheitsgrade (DOF²⁷) mit sieben Parametern verfügt, grundlegende Aussagen über das querdynamische Verhalten eines Fahrzeuges getätigt werden [San07, Hir11]. Durch Detaillierungen des Modells lassen sich weitere Effekte abbilden. So liefert ein verfeinertes Mehrkörper-Simulationsmodell (MKS²⁸-Modell) zusätzliche Resultate, die über jene des Minimalmodells hinausgehen. Demnach können mit dem MKS-Modell aus Abb. 4.6 ebenfalls Aussagen über den Einfluss von Fahrwerkskomponenten wie Lenkung oder Stabilisatoren getroffen werden. Dennoch steigt die Ergebnisqualität bei Effekten, die auch schon das Einspurmodell abbildet, nicht proportional mit den zusätzlichen Modellverfeinerungen, sondern nähert sich asymptotisch dem Verhalten eines realen Fahrzeuges an.

Das hier beschriebene Beispiel des Einspurmodells eignet sich wegen der geringen Anzahl an Eingangsdaten bei gleichzeitig hoher Aussagekraft der Ergebnisse gut für eine CAD-basierte Funktionsauslegung. So kann es aufgrund des relativ simplen Modellaufbaus leicht in ein *Funktions-template* implementiert werden. Jedoch findet sich nicht für jede Auslegung ein entsprechend einfaches Minimalmodell, das alle notwendigen Effekte in hinreichender Genauigkeit abbildet und mit wenigen Parametern auskommt. Oftmals gestalten sich die Modellbildung und die

²⁷Degrees Of Freedom

²⁸Mehrkörper-Simulations

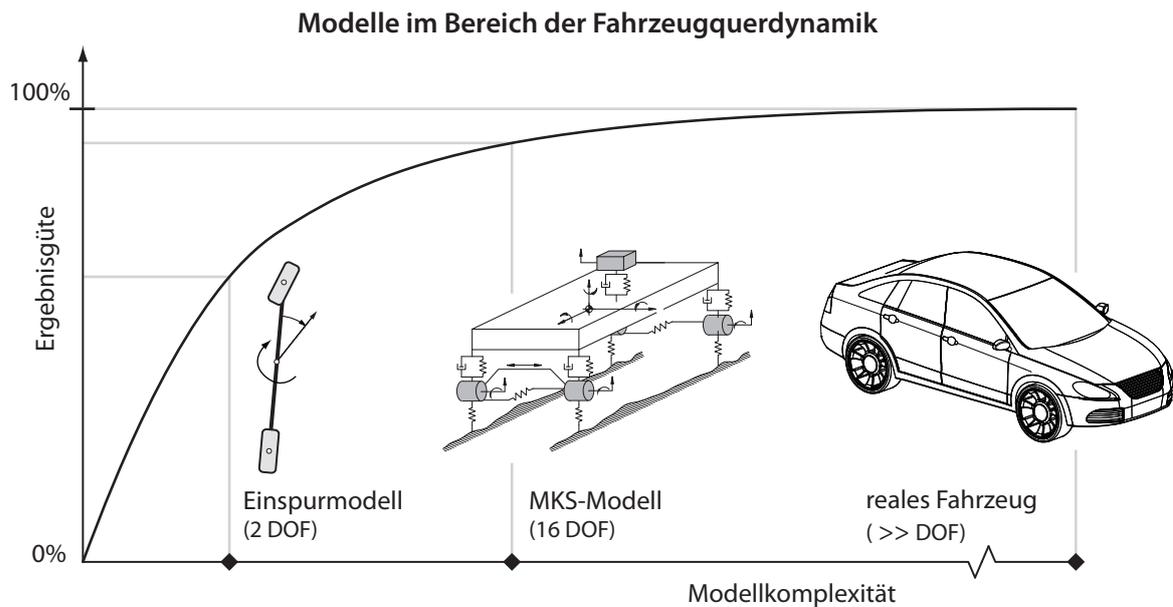


Abbildung 4.6: Abhängigkeit der Ergebnisgüte von der Modellkomplexität am Beispiel von Modellen für die Fahrzeugquerdynamik, zum Teil aus [Nie12].

Implementierung in ein *Funktionstemplate* als so komplex, dass sie mit einem unverhältnismäßig hohen Aufwand verbunden sind. Gerade bei Berechnungsdomänen, die derzeit noch unzureichend von CAD-Systemen abgedeckt werden, wie etwa die passive Sicherheit oder die Fluidodynamik, erweist sich die Erstellung von *Funktionstemplates* zumeist als schwierig. In jenen Fällen, bei denen sich die Erzeugung eines brauchbaren *Funktionstemplates* als extrem umfangreich gestaltet, stehen den Einsparungen und Verbesserungen einer CAD-basierten Vorauslegung daher hohen Aufwänden gegenüber, die den Einsatz unrentabel machen. Demnach ist gerade bei komplexen Aufgabenstellungen zu prüfen, ob sich die Erstellung und der Einsatz von *Templates* für eine CAD-basierte Vorauslegung im Vergleich zu einer konventionellen Entwicklung als vorteilhaft erweist.

Neben der Entwicklung der *Templates* stellt – wie generell bei wissensbasierten Konstruktionsmethoden – die Archivierung, die Bereitstellung und Wartung eine Voraussetzung für eine nachhaltige Verwendung dieser Werkzeuge dar. Darüber hinaus gilt es, die Anwender bei Problemen und Fragestellungen zu unterstützen. Dadurch steigen die Anforderungen an die Administration der EDV-Systeme, die entsprechende Kapazitäten vorsehen muss. Demnach können die Fähigkeiten und die entsprechende Infrastruktur der EDV-Systemadministration ebenfalls als eine Randbedingung für den erfolgreichen Einsatz einer CAD-basierten Vorauslegung angesehen werden.

Trotz ihrer Einschränkungen stellt die CAD-basierte Vorauslegung eine vielversprechende Methode dar, welche durch Ausschöpfen der Fähigkeiten von modernen, wissensbasierten CAD-Systemen eine Verbesserung des Entwicklungsprozesses speziell in der frühen Phase herbeiführen kann. Um jedoch die resultierenden Vorteile, wie etwa einen frühzeitigen Wissenszuwachs, nutzen zu können, gilt es die jeweiligen Anwendungsfälle auf ihre Eignung für eine CAD-basierte

Vorauslegung zu prüfen. Demnach hängt der Erfolg dieser Methode von den hier beschriebenen Randbedingungen ab und deren Berücksichtigung ist vor dem Hintergrund innerbetrieblicher Ressourcen für jeden Einzelfall sicherzustellen. Für geeignete Einsatzszenarien beschreibt das Vorgehensmodell der CAD-basierten Vorauslegung ein übergeordnetes Schema, welches für die eigentliche Umsetzung eine Vielzahl an Freiheiten zulässt. Dadurch werden potentielle Anwendungen bei ihrer Realisierung nicht eingeschränkt, sondern es wird vielmehr ein Leitfaden für die Umsetzung gegeben. Die Wirksamkeit der CAD-basierten Vorauslegung hängt stark vom jeweiligen Anwendungsfall ab. Um die Vorteile wie auch die Effektivität dieses Ansatzes zu demonstrieren, wird daher an dieser Stelle die prototypische Umsetzung anhand konkreter Anwendungsbeispiele präsentiert.

4.4 Prototypische Umsetzung einer CAD-basierten Vorauslegung

Das Ziel der folgenden zwei Kapitel (Kap. 5 und Kap. 6) besteht darin, das Konzept der CAD-basierten Vorauslegung anhand von geeigneten Beispielen aus der Industrie anzuwenden und damit die Wirksamkeit dieses Ansatzes zu verdeutlichen. Als Anwendungsgebiet wurde die Karosseriekonstruktion ausgewählt, da diese einerseits einen hohen Kostenanteil bei der Entwicklung von Fahrzeugen aufweist und daher Effizienzsteigerungen eine entsprechende hohe Auswirkung haben [GN06]. Andererseits besteht in diesem Bereich auch ein hoher Bedarf an einer konstruktionsbegleitenden Auslegung in der frühen Phase, da Karosseriebauteile einen relativ hohen Anteil an fahrzeugspezifischen Neuentwicklungen aufweisen. So stellen gerade sichtbare Anbauteile ein Alleinstellungsmerkmal dar und können daher im Gegensatz zu nicht sichtbaren Komponenten, wie Antriebseinheiten, Fahrwerken aber auch Karosserie-Strukturelementen, nicht von bereits vorhandenen Baureihen übernommen werden.

Um zu zeigen, dass der Ansatz einer CAD-basierten Vorauslegung ein breites Spektrum an verschiedenen Einsatzszenarien abdeckt, werden bewusst zwei fachlich unterschiedliche Aufgabenstellungen ausgewählt. Demnach beschäftigt sich das erste Anwendungsbeispiel mit einer kinematischen Fragestellung, während sich das zweite thematisch mit Strukturmechanik auseinandersetzt. Der Umsetzungsgrad der hier angeführten Anwendungsbeispiele variiert von prototypischen Prinzipstudien bis hin zu industriell eingesetzten Applikationen, die bereits in mehreren Entwicklungsprojekten erfolgreich zum Einsatz gekommen sind. Unabhängig davon wird Wert auf die Nachvollziehbarkeit der ausgearbeiteten Methoden gelegt, sodass eine gute Dokumentation und eine leichte Übertragbarkeit auf vergleichbare Anwendungen gegeben ist.

4.5 Zusammenfassung

In diesem Kapitel wird eingangs auf die Durchdringung von *Templates* in der frühen Phase des Entwicklungsprozesses eingegangen. So kann in einer Definitionsphase vor allem auf Entwurfssysteme für Gesamtfahrzeuge zurückgegriffen werden, die entweder eigenständige Software-Applikationen oder *Template*-basierte Systeme einschließen. In der folgenden Konzeptphase haben sich zumeist *Geometrietemplates* etabliert, die bei der Erstellung von flexibler und

gleichzeitig detaillierter Konzeptgeometrie herangezogen werden. In einem weitaus geringeren Ausmaß kommen bisher *Funktionstemplates* zur Anwendung, obwohl sich diese prinzipiell für die Entwicklung in der frühen Phase eignen. Mit Hilfe des Ansatzes einer CAD-basierten Vorauslegung wird gezeigt, wie durch den vermehrten Einsatz von *Funktionstemplates* und deren Kombination mit *Geometrietemplates* die Entwicklung in der Konzeptphase wesentlich unterstützt werden kann.

Hierbei berücksichtigt die CAD-basierte Vorauslegung das heterogene Entwicklungsumfeld dieser Phase, welches genauso durch sich schnell ändernde und unvollständige Daten wie auch durch eine Vielzahl an Varianten und einen rasant steigenden Detaillierungsgrad geprägt ist. Der Grundgedanke der CAD-basierten Vorauslegung besteht darin, dass die Konstruktionstätigkeiten neben der Erstellung von Geometriedaten ebenfalls konstruktionsbegleitende, vorläufige Funktionsauslegungen und -absicherungen umfassen. Die Ergebnisse dieser Auslegungstätigkeiten liefern die Grundlage für die Geometriemodellierung, wodurch die funktionale Qualität der Geometriedaten erhöht wird. Sowohl für die Funktionsauslegung als auch für die Geometriemodellierung liefern *Templates* die entscheidenden Werkzeuge. Der Ansatz einer CAD-basierten Vorauslegung wird in diesem Kapitel anhand eines Vorgehensmodells beschrieben. Darüber hinaus wird auf die Vorteile aber auch auf die Einschränkungen und Randbedingungen dieses Konzeptes eingegangen. Um die Wirksamkeit dieses Ansatzes zu verdeutlichen, ist in den folgenden Kapiteln die Umsetzung von zwei konkreten Anwendungsbeispielen aus der Industrie dargestellt.

5 Kinematikauslegung einer Fahrzeugtüre

5.1 Einleitung

Die Auslegung von Fahrzeugtüren ist in der frühen Phase der Gesamtfahrzeugentwicklung eine herausfordernde Aufgabenstellung. So hat die Form der Einstiegsöffnung, die im Wesentlichen dem Dichtungsflansch der Türe entspricht, einen direkten Einfluss auf den Einstiegscomfort der Passagiere und somit auf die Ergonomie des Fahrzeuges. Darüber hinaus liefern die Fahrzeugtüren und -klappen, als großflächige und mit einem charakteristischen Fugenverlauf abgegrenzte Elemente der Fahrzeugaußenhaut, ein wichtiges Designmerkmal. In Bezug auf die Funktion der Fahrzeugtüre muss das Design der Außenhaut allerdings einige Anforderungen erfüllen. So liefert insbesondere die Kinematik der Fahrzeugtüre einerseits beim Öffnen der Türe und andererseits beim Absenken der Fahrzeugscheibe Randbedingungen für das Fahrzeugdesign. Dadurch wird in der frühen Phase eine intensive Abstimmung zwischen den einzelnen Entwicklungsbereichen notwendig. Die funktionale Auslegung der Türkinematik vor dem Hintergrund der *Design-Technik-Konvergenz* eignet sich gut als Anwendungsbeispiel für die CAD-basierte Vorauslegung. In diesem Kapitel wird daher eingangs kurz auf die Anforderungen der Türentwicklung in der frühen Phase eingegangen und in weiterer Folge der Aufbau des umgesetzten Konzepts für die Anwendung der CAD-basierten Vorauslegung beschrieben. Die implementierten Lösungsansätze für die funktionale Vorauslegung werden ebenso dargelegt wie die Methoden für die Geometrierzeugung mit Hilfe von *Templates*.

5.2 Türentwicklung in der Konzeptphase

Die Türen und Klappen bilden sowohl in der Produktstruktur als auch in der Projektorganisation einer Gesamtfahrzeugentwicklung üblicherweise eine Teilmenge der Karosserie. Im Gegensatz zur übergeordneten Karosserieentwicklung, deren Strukturauslegungen bereits in der Definitionsphase beginnen, spielt die Türentwicklung vor der Konzeptphase nur eine untergeordnete Rolle und wird oftmals durch Prinzipschnitte von Vorgängermodellen berücksichtigt [Has05]. Beim Wechsel von der Definitions- in die Konzeptphase steigt allerdings der Detaillierungsgrad der einzelnen Karosseriemodule rasant an, sodass an dieser Stelle ebenfalls die konzeptionelle Entwicklung der Türen und Klappen zunehmend an Bedeutung gewinnt. Als wichtige Eingangsdaten liegen im Fall der Fahrzeugtüren zu diesem Zeitpunkt zumeist die Einstiegsöffnung, mit der schon Ergonomieuntersuchungen durchgeführt werden, das Fahrzeugdesign sowie Anforderungen und Ziele aus dem Lastenheft vor. Ebenfalls aus Ergonomieuntersuchungen sowie aus gesetzlichen Anforderungen kommen Begrenzungskurven und -flächen,

die beispielsweise das Sichtfeld oder die Ellbogenfreiheit abgrenzen und als Randbedingungen dienen.

Mit Hilfe der vorhandenen Eingaben gilt es, in der Konzeptphase alle wesentlichen funktionalen und bauraumspezifischen Auslegungen abzuschließen und die Resultate in eine Konzeptgeometrie zu überführen. *Schadenbauer* gibt hierzu eine Aufstellung von charakteristischen Anforderungen an die Entwicklung von Fahrzeugtüren und skizziert den typischen Ablauf einer konzeptionellen Türenentwicklung [Sch10a]. Demnach besteht die Hauptfunktion der Fahrzeugtüre darin, Passagieren ein bequemes Ein- und Aussteigen zu ermöglichen und sie im geschlossenen Zustand vor Umwelteinflüssen zu schützen. Anhand dieser Hauptfunktionen wird die Wichtigkeit sowohl der Ergonomie als auch der Türkinematik unterstrichen. Letztere gliedert sich, wie eingangs angeführt, im Wesentlichen in zwei Mechanismen, nämlich einen zum Öffnen der Türe sowie einen zum Absenken der Scheibe. Zusätzlich wird mit dem Schutz vor Umwelteinflüssen vor allem auf die Dichtheit der Türe eingegangen. So ist insbesondere bei der Entwicklung von rahmenlosen Türen, wie sie in Cabrios zum Einsatz kommen, auch bei extremen Witterungsverhältnissen eine vollständige Dichtheit sicherzustellen.

Eine weitere wichtige Anforderung an die Türenentwicklung liegt im Crashverhalten. So beeinflusst die Türstruktur besonders bei einem Seitenaufprall die passive Sicherheit des Fahrzeugs. Darüber hinaus wird versucht, auch nach einem Unfall das Öffnen der Türe zu gewährleisten. Neben den Lastfällen der passiven Sicherheit kommen bei der Entwicklung der Türstruktur ebenfalls Missbrauchsfälle, wie etwa das Überdrücken der Türe beim Öffnen, zum Tragen. Zusätzlich zu den Anforderungen an die Türstruktur liegen ebenso Vorgaben hinsichtlich der Aerodynamik vor. Gerade bei Vordertüren, die oftmals den Seitenspiegel als strömungsdynamisches Störobjekt enthalten, müssen diese Anforderungen berücksichtigt werden. Allerdings sind Detailanpassungen, wie die Optimierung des Seitenspiegels, vielfach Inhalt der Vorentwicklungsphase. Gleiches gilt beispielsweise für Verbesserungen des Öffnungs- und Schließgeräusches, welches als Qualitätsmerkmal gilt.

Demgegenüber treten in der Konzeptphase vielmehr Kosten- und Gewichtsaspekte in den Vordergrund. Beide Kriterien sind während des Entwicklungsprozesses ständig von großer Bedeutung, sie werden jedoch gerade in der Konzeptphase durch die Auswahl von Materialien und Herstellungsverfahren maßgeblich bestimmt. Gerade der Türrohbau nimmt hierbei eine wichtige Rolle ein, da er sowohl das Gewicht als auch die Kosten wesentlich beeinflusst. Daher werden bei der Entwicklung des Türrohbaus oftmals mehrere konkurrierende Varianten ausgearbeitet, um aus einer möglichst breiten Palette an verschiedenen Optionen die am besten geeignete auswählen zu können. Darüber hinaus nimmt bei der konzeptionellen Entwicklung des Türrohbaus gleichfalls die Bauraumauslegung einen großen Stellenwert ein. Demnach dient der Rohbau als Aufnahme für Module wie Fensterheber, Türschloss samt Türschnalle, Crashverstärkungen, Innenverkleidung oder diverse Multimedia- und Bedieneinrichtungen, die allesamt im Rahmen einer geometrischen Integration einzubeziehen sind.

Das Ziel der Konzeptphase besteht nun darin, ein ausgereiftes Türenkonzept zu erstellen, welches die angeführten Anforderungen berücksichtigt. Dies erfordert umfangreiche Konstruktionstätigkeiten samt geometrischer Integration sowie funktionale Auslegungen mit Berechnungs-

schleifen. Demnach bieten die unterschiedlichen Entwicklungsaufgaben einige potentielle Anwendungen für den Einsatz der CAD-basierten Vorauslegung. Im Rahmen dieser Arbeit wird speziell auf die Türkinematik eingegangen, da deren Auslegung wichtige Eingaben für die Konstruktion des Türrohbaus liefert und daher in der Konzeptphase von hoher Bedeutung ist. Außerdem ist die Türkinematik in der Regel unabhängig von der Bauweise der Türe und damit sind die entwickelten Methoden universell einsetzbar. Neben der funktionalen Auslegung der Türkinematik wird ebenfalls auf die Geometriemodellierung eines Türinnenblechs, als Hauptbestandteil des Türrohbaus eingegangen.

5.3 Anwendung des Vorgehensmodells

In Abb. 5.1 wird schematisch die Anwendung des Vorgehensmodells einer CAD-basierten Vorauslegung für die konzeptionelle Türentwicklung dargestellt. Demzufolge wird das vorhandene Wissen in Form von *Funktions-* und *Geometrietemplates* bereitgestellt, sodass während der Vorauslegung auf diese Vorlagen zugegriffen werden kann. Die *Funktionstemplates* liefern, in der hier beschriebenen prototypischen Umsetzung, die Werkzeuge für die Auslegung der Türöffnung und der Scheibenabsenkung. Bei der ersten Anwendung wird eine kollisionsfreie Öffnungs- und Schließbewegung der Türe sichergestellt. Die zweite Anwendung liefert den Fahrweg der Fahrzeugscheibe unter dem Gesichtspunkt einer minimalen Auslenkung der Scheibendichtungen. Dadurch werden die wesentlichen kinematischen Fragestellungen bei der Konzeption einer Türe abgedeckt. Die *Geometrietemplates* umfassen eine Bibliothek an vielfältigen Geometriemodellen, deren Umfang von vollständigen Türgeometrien auf Baugruppenebene bis hin zu charakteristischen Geometriesequenzen reicht. In der hier vorgestellten Umsetzung wird beispielhaft auf ein Bauteilmodell eines Türinnenblechs, als eines der komplexesten Bauteile im Türrohbau, eingegangen. Beide Kategorien – *Funktionstemplates* und *Geometrietemplates* – lassen sich relativ einfach erweitern und bilden die Grundlage für eine CAD-basierte Vorauslegung.

Abb. 5.1 zeigt außerdem eine typische Produktstruktur einer Fahrzeugtüre, die während der Konzeptphase sukzessive befüllt und detailliert wird. Stehen am Beginn der Konzeptphase zumeist nur Daten von der Fahrzeugaußenhaut, von einigen Begrenzungsflächen und -kurven sowie von der Einstiegsöffnung zur Verfügung, so enthält die Produktstruktur am Ende der Konzeptphase eine Produktbeschreibung in Form von detaillierten Geometriemodellen. Dieser Abschnitt behandelt beispielhaft die Erstellung eines Türinnenblechs nach einer vorangegangenen Kinematikauslegung. Der Ablauf vom Erhalt der Eingangsdaten bis zur Erstellung der Konzeptgeometrie des Innenblechs entspricht dem Vorgehensmodells der CAD-basierten Vorauslegung. Die dazu in Kap. 4.3.2 beschriebenen Phasen werden ebenfalls in Abb. 5.1 dargestellt.

Demzufolge werden in der ersten Phase alle Daten, die innerhalb der CAD-Umgebung vorliegen und für die funktionale Vorauslegung sowie die Geometriemodellierung notwendig sind, gesammelt in einem übergeordneten Skelett zusammengefasst. Innerhalb des Skeletts kommt es zu einer Aufbereitung der Daten. So müssen beispielsweise Designflächen oftmals aufgrund von

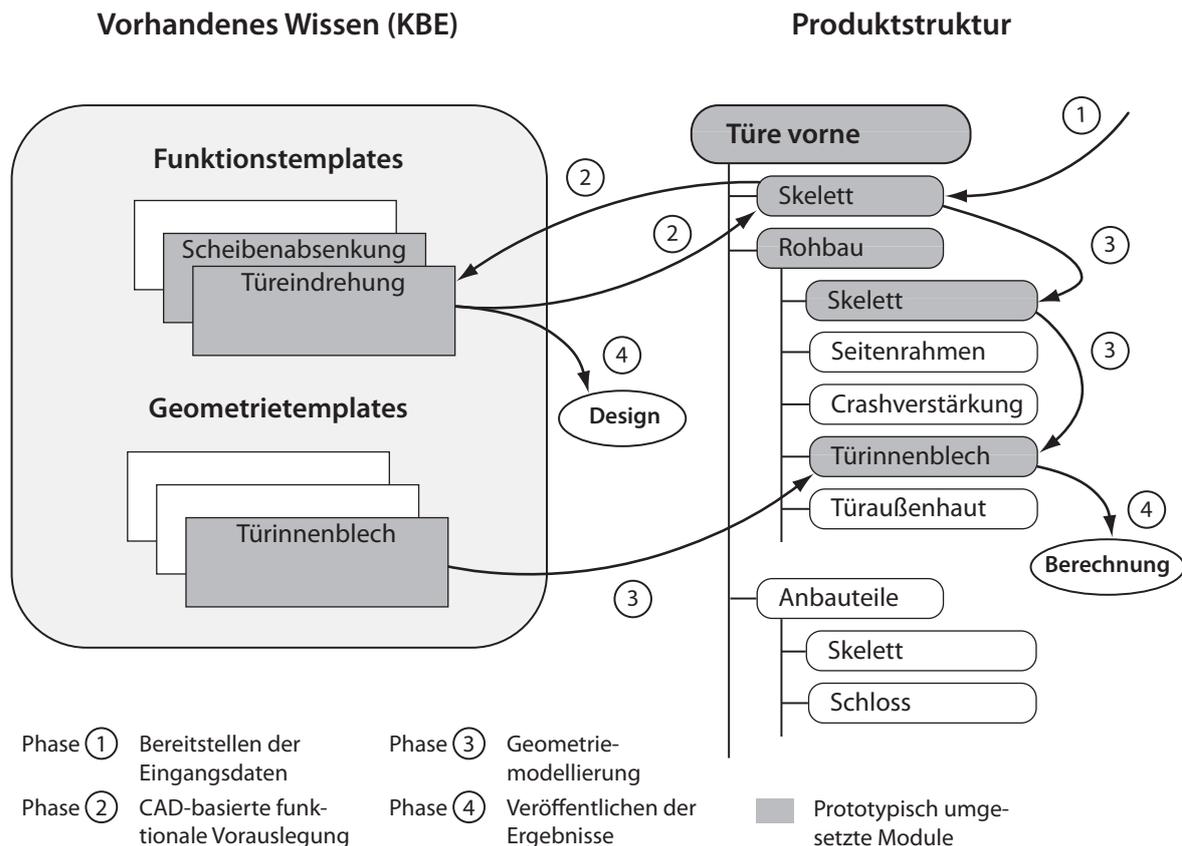


Abbildung 5.1: Prototypische Anwendung der CAD-basierten Vorauslegung mit den einzelnen Phasen; links: das in Templates bereitgestellte Wissen, rechts: die Produktstruktur einer Fahrzeugtüre.

fehlerhaften Geometriedaten, die infolge von Konvertierungsfehlern zwischen CAS- und CAD-Systemen auftreten, bereinigt werden. Nachdem die benötigten Daten in einer ausreichenden Qualität vorhanden sind, kommt es zur funktionalen Vorauslegung mit Hilfe der *Templates* „Scheibenabsenkung“ und „Türeindrehung“. In dieser zweiten Phase werden die Ergebnisse der kinematischen Untersuchungen, nämlich die Drehachse der Türe beim Öffnen und die Bewegung der Scheibe beim Absenken, ermittelt. Diese Ergebnisse dienen einerseits als Vorgabe für die Geometrieerzeugung in Phase drei und andererseits können sie schon vorzeitig als Teil der Phase vier anderen Fachbereichen zur Verfügung gestellt werden. So kann beispielsweise im Rahmen der Untersuchung der Türeindrehung ein möglicher Designvorschlag abgelehnt werden, da eine kollisionsfreie Türöffnung mit dem vorgeschlagenen Fugenverlauf nicht möglich ist. Gleichzeitig können potentielle Änderungen der Randbedingungen, etwa eine Vergrößerung des Fugenabstands oder eine Änderung des Fugenverlaufs, mit denen eine kollisionsfreie Türöffnung sichergestellt werden kann, aufgezeigt werden.

Sollte die Vorauslegung der Kinematik die gewünschten Ergebnisse enthalten, kann mit diesen Resultaten unter Berücksichtigung verschiedener weiterer Randbedingungen die Geometrie des Türinnenblechs erstellt werden. Hierzu werden die Daten der Vorauslegung in das Skelett der Gliederungsebene Rohbau transferiert. Dort dienen die Daten entsprechend der EVA- bzw.

IPO-Gliederung als Eingabe für das *Matingtemplate* des Türinnenblechs, das aus der Bibliothek der *Geometrietemplates* eingefügt werden kann, Abb. 5.1. Dieses *Matingtemplate* lässt sich flexibel an die Vorgaben anpassen und enthält einige charakteristische Varianten in Form von austauschbaren Geometriesequenzen. Demnach lassen sich unterschiedliche Bauformen des Türinnenblechs umsetzen. Die resultierende Geometrie kann dann in der vierten Phase des Vorgehensmodells dem Produktentstehungsprozess zur Verfügung gestellt werden und dient etwa als Eingabe für die Strukturberechnung des Türrohbaus.

Durch die Anwendung des Vorgehensmodells lässt sich effizient eine Konzeptgeometrie erstellen. Außerdem liegen in einem vergleichsweise frühen Stadium Ergebnisse von kinematischen Untersuchungen vor, die gegebenenfalls Unzulänglichkeiten von Eingabedaten aufzeigen und so zu frühzeitigen Anpassungen führen. Ein wesentlicher Bestandteil der erfolgreichen Umsetzung der CAD-basierten Vorauslegung bei diesem Anwendungsbeispiel liegt in den Lösungsansätzen der kinematischen Syntheseprobleme, die in den *Funktionstemplates* integriert sind. Auf diese wird in den folgenden Abschnitten detailliert eingegangen. Gleiches gilt für die Strukturierung und Umsetzung des *Geometrietemplates* für das Türinnenblech, die ebenfalls beschrieben werden.

5.4 Absicherung einer kollisionsfreien Türöffnung

5.4.1 Anforderungen

Um die Öffnungs- und Schließbewegung einer Türe umzusetzen, stehen unterschiedliche Mechanismen zur Verfügung. Prinzipiell können die Trajektorien, welche die Bewegung der Türe darstellen, und der zu Grunde liegende Mechanismus beliebig gewählt werden. Dennoch haben sich aus diversen Gründen, wie etwa einer wirtschaftlichen Umsetzbarkeit, Ergonomieanforderungen oder Gewichtsvorgaben, vor allem Schiebemechanismen und die Rotation um eine fahrzeugfeste Achse durchgesetzt. Für Schiebemechanismen spricht, dass der Ein- und Ausstieg aus ergonomischen Gesichtspunkten vorteilhaft ist. Darüber hinaus wird bei der Türbewegung wenig Raum überstrichen, was sich besonders bei engen Parklücken als positiv erweist. Dennoch stehen diesen Merkmalen im Vergleich zu einem konventionellen rotatorischen Öffnungsmechanismus wesentliche Nachteile gegenüber. Darunter fällt beispielsweise die komplexere Bauweise, die zu höheren Kosten, einer Mehrzahl an Bauteilen, höherem Gewicht sowie einem Mehraufwand bei der geometrischen und funktionalen Integration führt. Deshalb werden Schiebetüren fast ausschließlich bei großvolumigen Fahrzeugen mit mehreren Sitzreihen verwendet, da durch die großen rückwärtigen Einstiegsöffnungen die Nachteile von konventionellen rotatorisch öffnenden Türen überwiegen.

Aufgrund der dominanten Ausprägung von sich drehend öffnenden Fahrzeugtüren beschränkt sich der vorgestellte Ansatz auf diese Anordnung. Außerdem wird in weiterer Folge lediglich die vordere Türe betrachtet. Die ausgearbeitete Vorgehensweise lässt sich jedoch mit geringem Aufwand an die Gegebenheiten einer rückwärtigen Türe übertragen. Die Rotation der Fahrzeugtüre um eine fahrzeugfeste Drehachse ist üblicherweise durch zwei Scharniere definiert. Die Türscharniere befinden sich infolge von Design- und Aerodynamik-Anforderungen

zumeist innerhalb der Fahrzeugaußenhaut, wodurch der vordere Teil der Türe beim Öffnen in das Fahrzeug eintaucht.

Dieser Sachverhalt wird in Abb. 5.2 durch einen Schnitt im Bereich des oberen Türscharniers veranschaulicht. Beim Schnitt handelt es sich um einen Normalschnitt bezüglich der Scharnierachse, bei dem das Türaußenblech im geschlossenen und im geöffneten Zustand dargestellt ist. Durch das Eintauchen der Türe in diesem Bereich besteht die Gefahr, dass es hier zu einer Kollision der Türe mit den umliegenden Bauteilen kommt. Hierzu zählen der Kotflügel, die Seitenwand und das Türscharnier. Bei der Auslegung der Scharnierachse ist daher ein kollisionsfreies Öffnen eine wesentliche Anforderung, die es abzusichern gilt. Neben dieser Vorgabe müssen weitere Anforderungen berücksichtigt werden, wie etwa der Kraftverlauf der Türöffnungskräfte oder Produktions-, Design- und Ergonomievorgaben.

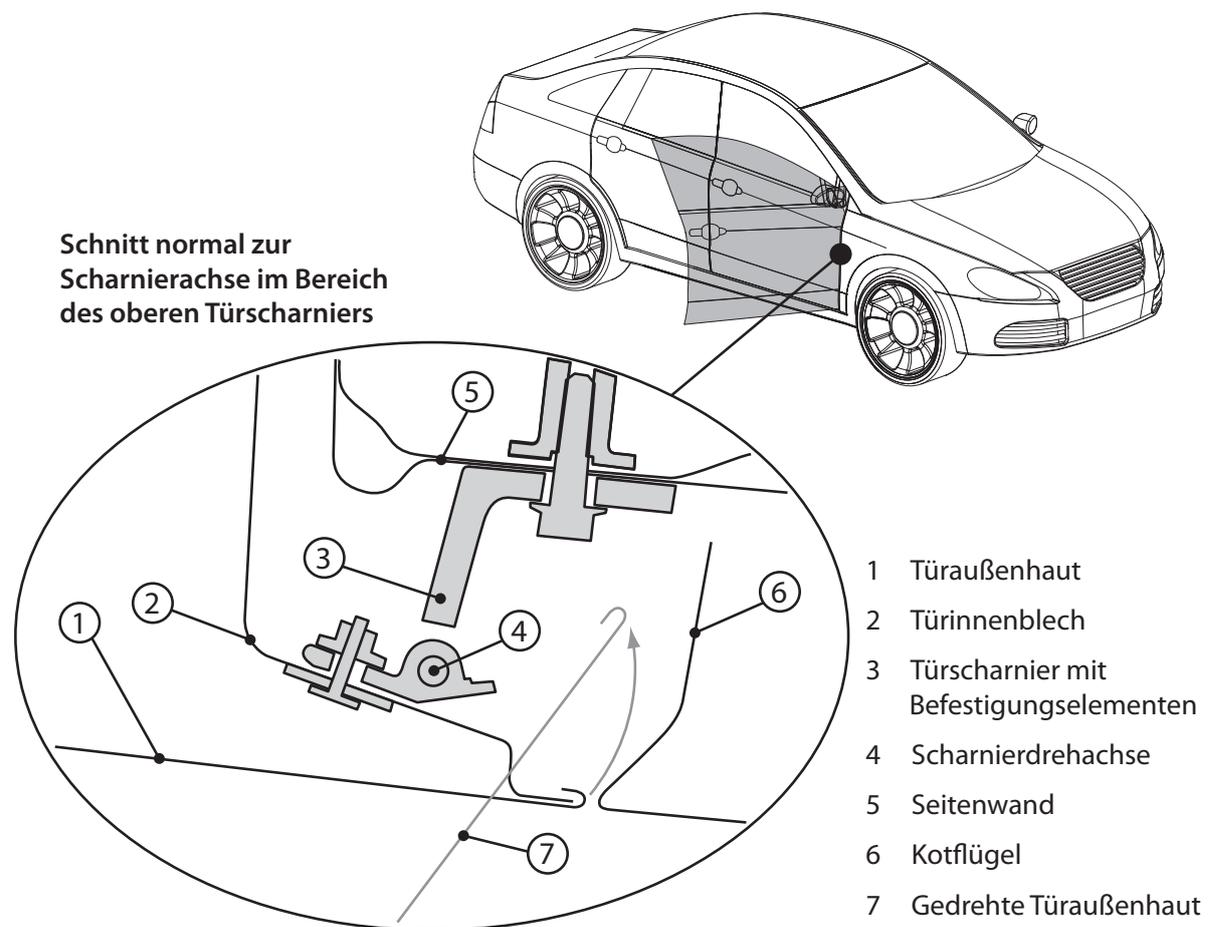


Abbildung 5.2: Darstellung der Türöffnung mittels Normalschnitt im Bereich des oberen Türscharniers.

So liefert unter anderem der Kraftverlauf, der für das Öffnen und Schließen einer Türe notwendig ist, eine wesentliche Einflussgröße für die subjektive Bewertung der Türöffnung durch den Benutzer. Neben den Türbremsen, die für ein Einrasten der Türe sorgen, wird dieser Kraftverlauf hauptsächlich von der Ausrichtung der Scharnierachse bestimmt. Daher ist die Scharnierachse normalerweise nicht parallel zur Fahrzeughochachse ausgerichtet, sondern geringfügig mit

den Winkeln α_x und α_y um die Fahrzeuglängs- und -seitenachse geneigt. Aufgrund der geneigten Rotationsachse muss beim Öffnen und Schließen ein Teil der Gewichtskraft der Türe vom Benutzer abgestützt werden. Eine nach hinten und innen geneigte Scharnierachse erschwert beispielsweise die Türöffnung und erleichtert das Schließen der Türe. Der Kraftverlauf und somit die Ausrichtung der Achse wird zumeist vom jeweiligen Fahrzeughersteller vorgegeben und folgt keinen allgemeingültigen Regeln. Allerdings überschreiten α_x und α_y in der industriellen Praxis üblicherweise nicht den Wert von 3° . Am Beginn der Auslegung werden α_x und α_y in der Regel als Zielwerte vorgegeben, um dadurch die subjektive Bewertung des Türöffnens, als einen Teil des Ein- und Ausstiegsvorgangs, zu beeinflussen.

5.4.2 Lösungsweg in drei Stufen

Die Aufgabenstellung bei der Kinematikauslegung der Türöffnung besteht somit in der Positionierung der Scharniere, die dadurch die Drehachse der Türe definieren. Als geometrische Eingabe ist zumindest die Fahrzeugaußenhaut sowie zumindest eine vorläufige Türfuge notwendig. Weitere erforderliche Geometriedaten, wie etwa die Scharniergeometrie, lassen sich, solange sie in der Konzeptphase nicht vorliegen, durch Standardgeometrien ersetzen. Das gleiche gilt für die weiteren Randbedingungen, wie etwa die Vorgabewerte für die Neigungswinkel α_x und α_y , die ebenfalls durch Standardwerte und Annahmen ersetzt werden können. Der vorgestellte Lösungsweg ist unabhängig vom Reifegrad der Eingabedaten und bleibt daher unverändert. Dadurch kann die Auslegung der Türeindrehung auch über die Konzeptphase hinaus unterstützt werden. Abb. 5.3 skizziert den Lösungsweg und zeigt schematisch den Ablauf der Auslegung, der sich in drei Stufen gliedert.

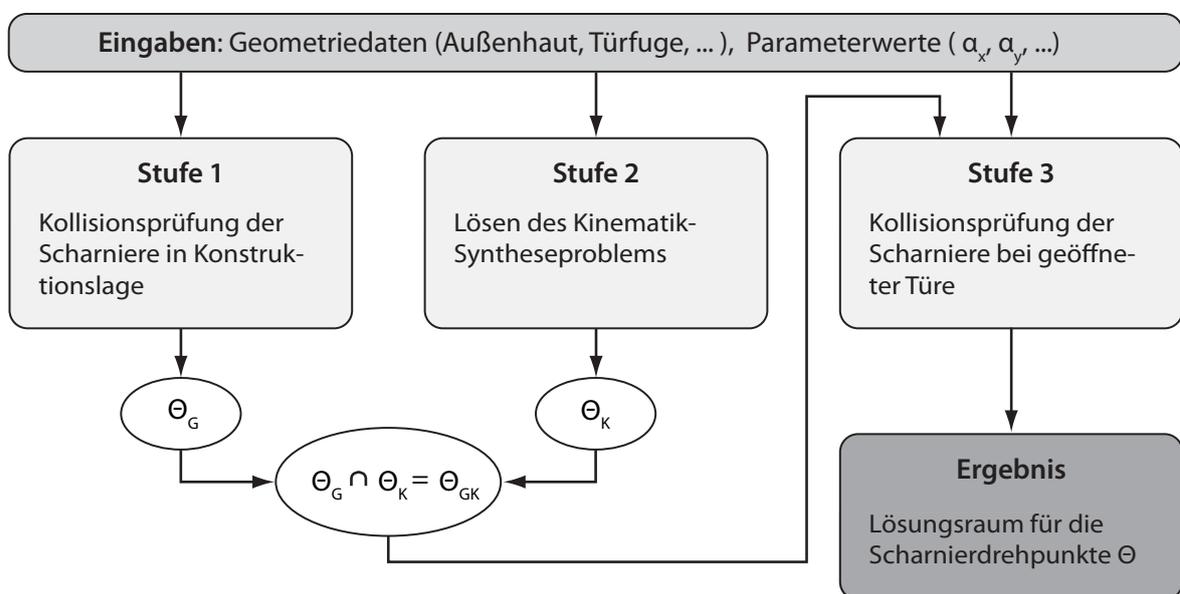


Abbildung 5.3: Schematischer Lösungsweg um den Lösungsraum Θ zu erhalten, innerhalb dessen die Scharnierdrehpunkte positioniert werden können.

In der ersten Stufe wird in der Konstruktionslage – also bei geschlossener Türe – der Bauraum Θ_G ermittelt. Θ_G liefert den Lösungsraum für die Positionierung der Scharnierdrehpunkte, sodass es in der Konstruktionslage zu keiner Überschneidung des Scharniermodells mit den umliegenden Komponentenmodellen kommt. Außerdem werden Freigänge und ein Mindestabstand zwischen den Scharnieren berücksichtigt. In der zweiten Stufe wird durch Lösung des Kinematik-Syntheseproblems der Raum Θ_K definiert. Θ_K liefert jene Positionen der Scharnierachsen, bei denen es zu keiner Kollision zwischen der sich öffnenden Türe und der Karosserie kommt. Es werden jedoch keine Kollisionen zwischen Türe und Scharnier überprüft. Außerdem werden die Einschränkungen aus der ersten Stufe nicht miteinbezogen. Daher wird vor der dritten Stufe mit Θ_{GK} die Schnittmenge von Θ_G und Θ_K ermittelt, sodass alle bisher getroffenen Randbedingungen berücksichtigt werden. Abschließend wird in der dritten Stufe die Kollision der offenen Türe mit dem Scharnier überprüft. Der verbleibende Lösungsraum Θ ergibt schlussendlich den gesuchten Bereich für die Scharnierdrehpunkte. Die einzelnen Stufen sind unabhängig voneinander, sodass die Abfolge prinzipiell vertauscht werden kann. Der hier gewählte Weg liefert unter Umständen eine geringere Anzahl an Operationen, da auf den letzten Schritt verzichtet werden kann, sobald $\Theta_G \cap \Theta_K$ einen leeren Lösungsraum liefert. In den hier folgenden Abschnitten wird auf die einzelnen Auslegungsstufen und die zu Grunde liegenden Methoden eingegangen.

Die dargestellte Vorgehensweise unterscheidet sich von einer konventionellen Auslegung. Diese erfolgt in der Regel manuell mit Hilfe von Schnittgeometrien und wird unter anderem von *Haslauer* beschrieben [Has05]. Demnach wird iterativ versucht, eine Lösung zu finden, was zum einen zeitaufwändig ist und zum anderen die Gefahr birgt, dass Lösungen übersehen werden. Ein automatisierbarer Ansatz, der die Nachteile einer manuellen Auslegung vermeidet, wird durch *Tecklenburg* geliefert [Tec11]. Allerdings werden etliche Vereinfachungen, insbesondere bei der Kinematik-Synthese getroffen, sodass sich der Ansatz lediglich als Abschätzung für eine frühe Konzeptphase eignet. Der hier vorgestellte Lösungsweg vermeidet diese Vereinfachungen und liefert somit eine allgemeingültige Methode für die Auslegung der Türöffnung, welche sich innerhalb eines *Funktionstemplates* leicht integrieren lässt.

5.4.3 Stufe eins: Kollisionsprüfung in der Konstruktionslage

In dieser ersten Stufe werden potentielle Positionen für die Scharniere gesucht, sodass es in der Konstruktionslage zu keiner Kollision zwischen dem Scharniermodell und der Karosserie kommt. Die Scharnierposition ist durch die Position des Drehpunktes, die Orientierung der Scharnierachse sowie durch die Ausrichtung von Scharnierober- und -unterteil relativ zur Drehachse gegeben. Die drei letzten Informationen müssen als Eingaben vorliegen und zumindest mit Standardwerten befüllt werden, während die Position der Drehpunkte gesucht ist. Die gesuchten Lösungen bilden den Lösungsraum Θ_G . Um Θ_G zu erhalten, werden Geometriedaten von der Fahrzeugaußenhaut und des Scharniers sowie Werte für Freigänge und Mindestabstände als Eingaben benötigt. Das Resultat sowie die notwendigen Eingaben dieses Auslegungsschrittes werden in Abb. 5.4 in einer Seiten- und einer Schnittansicht dargestellt.

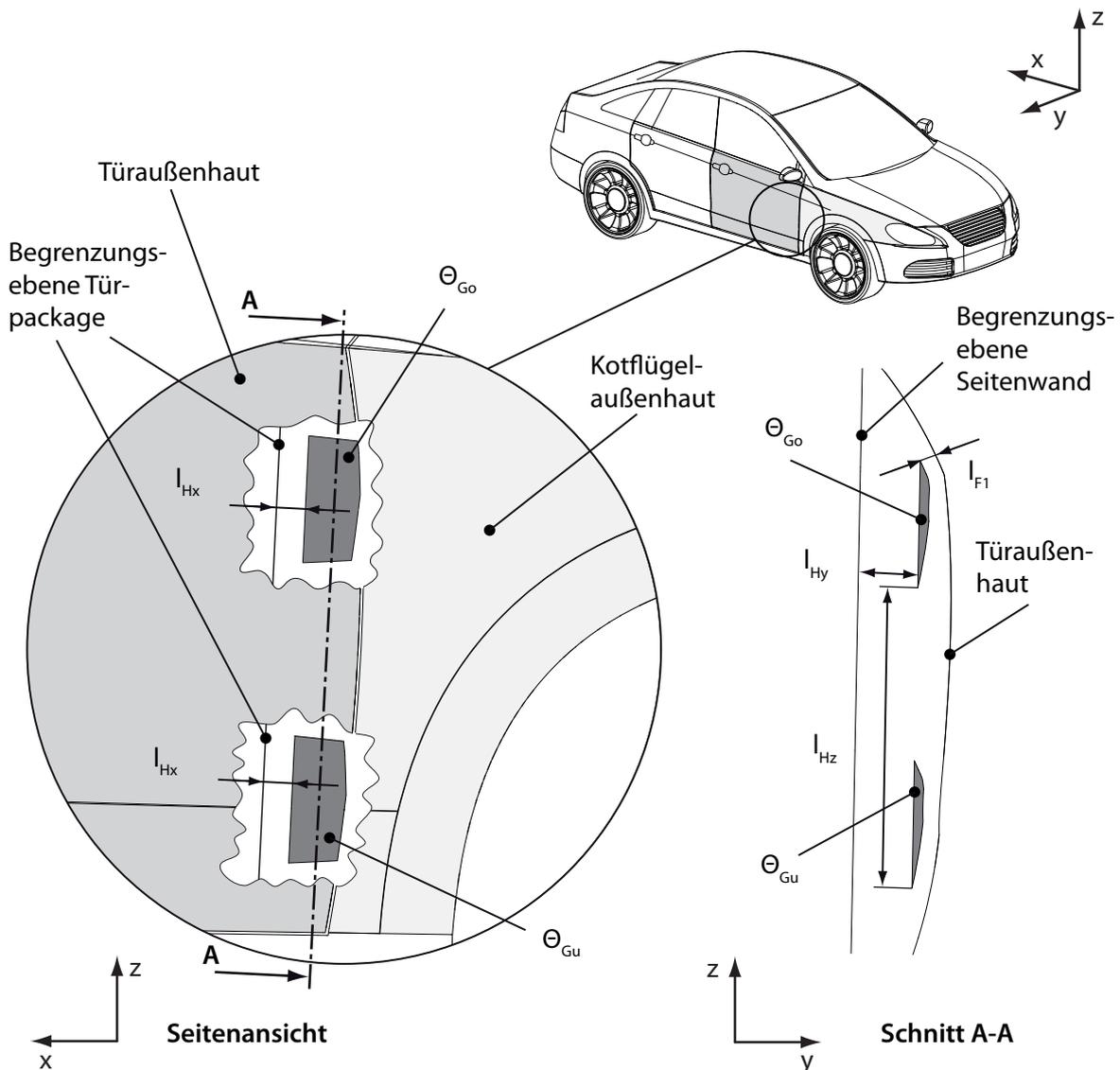


Abbildung 5.4: Lösungsraum für die Positionierung der Scharnierdrehpunkte Θ_{Go} und Θ_{Gu} .

Demnach müssen die Scharnierdrehpunkte mit einem Minimalabstand zur Türäußenhaut l_{F1} positioniert werden. Dieser Abstand schließt sowohl die Blechdicken des Türinnen- und Türäußenblechs als auch Lackdicken, Toleranzen und den Abstand der jeweiligen Scharniersilhouette bis zum Drehpunkt mit ein. Dies erfordert unter anderem die Kenntnis der Scharniergeometrie. Da es sich bei diesen Bauteilen oftmals um Übernahme- oder Standardbauteile handelt, sind diese Daten mitunter schon in einer frühen Phase vorhanden. Weitere Begrenzungsflächen für Θ_G liefern die Fahrzeugseitenwand sowie das Türpackage. Beide Grenzflächen sind in Abb. 5.4 als Ebenen eingezeichnet, sie können allerdings im Laufe der Entwicklung durch detailliertere Grenzflächen ersetzt werden. Während die Seitenwand als Schnittstelle zur Karosseriestruktur dient, muss beim Türpackage vorrangig Bauraum für die versenkbare Scheibe vorgehalten werden. An beiden Begrenzungen sind Türscharniere jeweils mit einem Arm befestigt. Die Längen vom Drehpunkt zum Ende des Scharnierarms sind in Abb. 5.4 als l_{Hx} und l_{Hy} bezeichnet.

Zusätzlich zu diesen Begrenzungen schließt der Bauraum für die Scharnierdrehpunkte in x-Richtung mit der Türfuge ab, da Drehpunkte vor der Fuge zu keiner eindrehenden Anordnung des Türmechanismus führen.

Eine zusätzliche Vorgabe besteht im Mindestabstand zwischen den Türscharnieren l_{Hz} . Die Scharniere stützen im geöffneten Zustand die Gewichtskraft der Türe ab. Im Missbrauchsfall, beispielsweise durch das Abstützen einer Person, können zusätzliche vertikale Kräfte auf den Türkörper wirken. Um sowohl die Scharniere als auch deren Anbindung an die Karosserie nicht unnötig stark dimensionieren zu müssen, wird ein Mindestabstand zwischen den beiden Scharnieren festgelegt. Dieser kann variieren, es wird allerdings in der industriellen Praxis ein Unterschreiten von 300 mm vermieden. Die angeführten Einschränkungen teilen Θ_G in einen oberen und einen unteren Lösungsraum Θ_{Go} und Θ_{Gu} auf. Darüber hinaus ist sicherzustellen, dass jeder Punkt aus Θ_{Go} , der als Drehpunkt für das obere Scharnier dienen kann, zumindest einen möglichen gegenüberliegenden Punkt in Scharnierachsenrichtung besitzt, der Teil von Θ_{Gu} ist. Diese Einschränkung der Lösungsräume führt dazu, dass Θ_{Go} und Θ_{Gu} bei einer Projektion in Scharnierachsenrichtung den gleichen Umriss besitzen.

Mit den angeführten Randbedingungen wird durch Θ_{Go} und Θ_{Gu} der Bereich für mögliche Scharnierdrehpunkte eingegrenzt und es wird sichergestellt, dass es in der Konstruktionslage zu keiner Kollision von Scharnier- und Karosseriemodellen kommt. Die erste Stufe des Lösungswegs aus Abb. 5.3 ist demnach abgeschlossen und in der nächsten Stufe gilt es, die Türbewegung mit einer Kinematik-Synthese zu untersuchen.

5.4.4 Stufe zwei: Lösung des Kinematik-Syntheseproblems

Die Synthesaufgabe besteht in der Darstellung jenes Lösungsraumes Θ_K , innerhalb dessen Scharnierdrehpunkte liegen können, ohne dass es bei der Öffnungsbewegung der Türe zu einer Kollision zwischen der Türe und dem restlichen Fahrzeug kommt. Obwohl die Topologie des Mechanismus mit einer reinen Rotation vergleichsweise einfach ausfällt, gestaltet sich die Lösung dieser räumlichen Problemstellung als nicht trivial. Daher wird das räumliche Problem in eine Reihe von ebenen Problemen übergeführt. Dies ist zulässig, da es sich bei der Rotation um eine ebene Bewegung handelt. Bei der Überführung werden $i = 1, \dots, i_e$ Schnitte erstellt und die Anforderung einer kollisionsfreien Öffnungsbewegung auf die Schnittgeometrien angewandt. Die Lösung der ebenen Problemstellungen gestaltet sich zum einen als weniger komplex und zum anderen liefert die Schnittmenge der ebenen Lösungen $\Theta_{K,i}$ schließlich den Lösungsraum für das räumliche Problem Θ_K und damit das gesuchte Resultat.

Die Schnittgeometrien werden im Bereich der Türfuge erstellt und die dazugehörigen Schnittebenen liegen normal zur Rotationsachse, deren Ausrichtung durch die Neigungswinkel α_x und α_y definiert ist. In Abb. 5.5 werden beispielhaft die Schnittebenen sowie die Geometrie eines Schnittes dargestellt. Die Geometrie beinhaltet den Schnitt der Türaußenhaut, des Kotflügels sowie der Seitenwand. Abhängig vom Einsatzzeitpunkt der Funktionsauslegung kann es sich bei der Geometrie um grobe Begrenzungsebenen und -flächen oder aber um detaillierte Konzeptgeometrie handeln. So kann insbesondere der Fugenbereich der Türe mit einer Bördelung des

Außenblechs oder einer Überlappung der Türe mit dem Kotflügel bereits definiert sein. Letzteres erfolgt vor allem für eine Verbesserung der passiven Sicherheit. Um die Entwicklung auch bei einer detaillierten Konzeptgeometrie zu unterstützen, erhebt die hier vorgestellte Methode den Anspruch, unabhängig von der Form der Schnittgeometrie und damit auch unabhängig vom Reifegrad der Geometrie gültig zu sein.

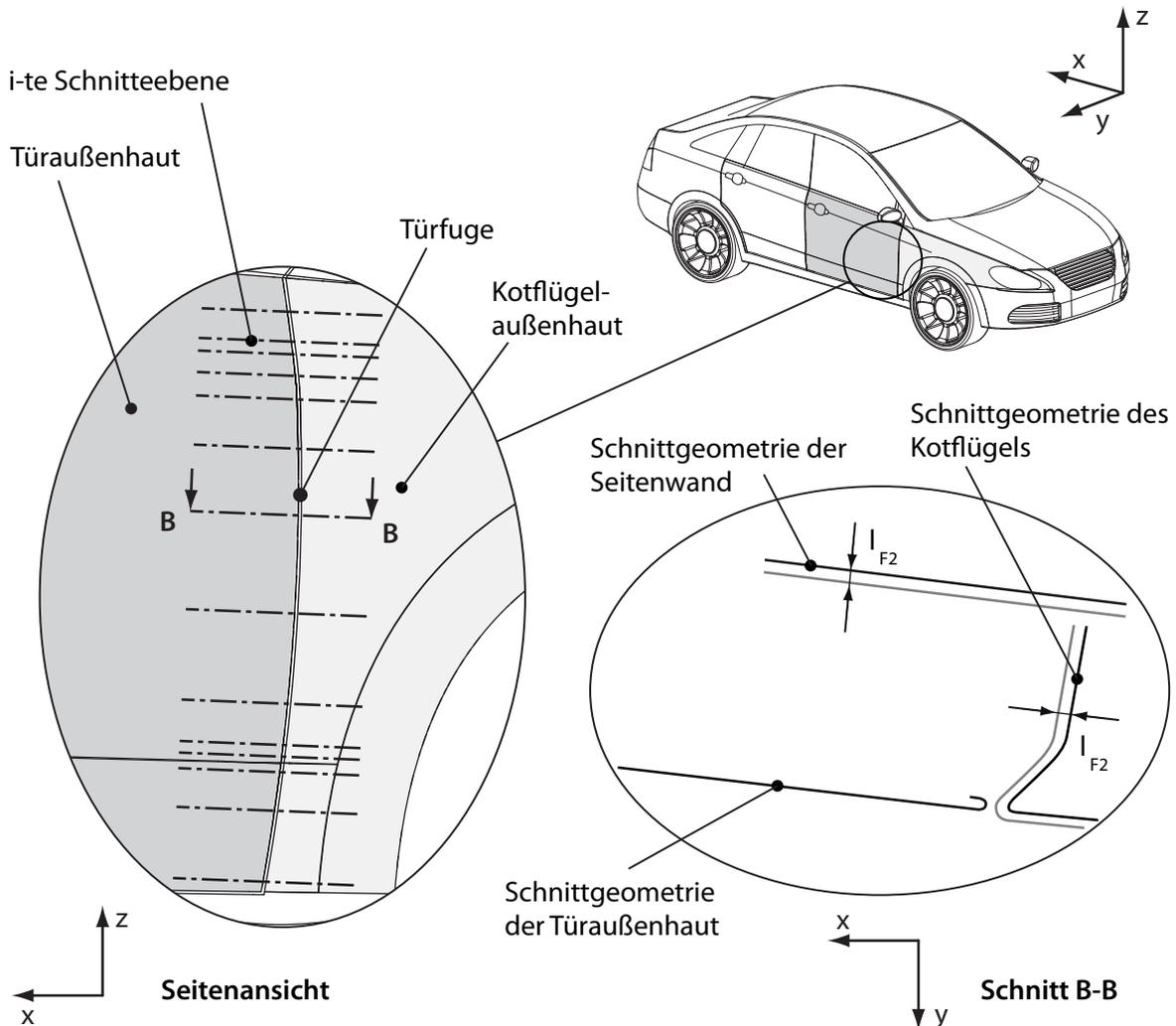


Abbildung 5.5: Lösungsraum für die Positionierung der Scharnierdrehpunkte.

Ebenfalls in Abb. 5.5 eingezeichnet sind Offsetgeometrien der Seitenwand und des Kotflügels. Diese resultieren aus der Forderung nach einem Freigang l_{F2} zwischen der sich öffnenden Türe und den umliegenden Komponenten. Der Freigang berücksichtigt fertigungsbedingte Toleranzen, innerhalb der sich die realen Bauteile im Vergleich zu den idealen Geometriemodellen befinden. Für das Syntheseproblem bedeutet dies lediglich, dass die Türgeometrie nicht die um l_{F2} abgesetzte Störgeometrie der umliegenden Bauteile berühren darf.

Ein wesentlicher Aspekt bei der Diskretisierung des Fugenbereichs durch Schnitte liegt in der Anzahl an benötigten Schnitten i_e . Zum einen soll eine hohe Auflösung und damit eine hohe Genauigkeit erzielt werden, zum anderen steigert eine große Anzahl an Schnitten die Berech-

nungszeit. Um eine ausgewogene Anzahl an Schnitten zu erzeugen, erfolgt die Diskretisierung in Abhängigkeit von der Türfuge. Mit dem Verlauf der Türfuge verändert sich aufgrund der Bombierung der Außenhaut gleichfalls die Position der Schnittgeometrie von Türe und Kotflügel. Im Gegensatz dazu ändert sich die Geometrie der Seitenwand sowohl in ihrer Gestalt als auch in ihrer Position zumeist nur geringfügig und hat daher wenig Einfluss auf die Anzahl und Anordnung der Schnittebenen. Daher haben die Schnitte, wie auch in Abb. 5.4 dargestellt, in jenen Bereichen, in denen die Fuge eine höhere Krümmung aufweist, einen geringeren Abstand zueinander als in jenen Bereichen mit einer kleinen Krümmung. Das zu Grunde liegende Verfahren zur Diskretisierung wird ebenfalls bei der CAD-basierten Darstellung von PVC²⁹-Nähten zur Fugenabdichtung erfolgreich eingesetzt [FTHH12]. Außerdem lässt dieses Verfahren eine Automatisierung zu, sodass die Diskretisierung leicht angepasst werden kann.

Sobald die Schnittgeometrien zur Verfügung stehen, kann in jedem Schnitt das ebene Syntheseproblem gelöst werden. Gegeben ist dazu die Schnittkurve der Türaußenhaut $c_{c,i}$, die Schnitte der Störgeometrie bestehend aus Kotflügel $c_{e,i}$ und Seitenwand $c_{f,i}$ sowie der Türöffnungswinkel β . Dieser gibt an, wie weit sich die Türe um die Scharnierachse dreht. Übliche Werte von β liegen bei etwa 75° , die zusätzlich zum beabsichtigten Öffnungswinkel ebenfalls Toleranzen und ein missbräuchliches Überdrücken der Türe berücksichtigen. Gesucht sind alle Drehpunkte, die bei einer Drehung von $c_{c,i}$ um den Winkel β zu keiner Kollision zwischen $c_{c,i}$ und der Störgeometrie $c_{e,i}$ bzw. $c_{f,i}$ führen. Die Drehpunkte werden in der Lösungsmenge $\Theta_{K,i}$ zusammengefasst, während Drehpunkte, die dieser Forderung nicht genügen, die komplementäre Lösungsmenge $\Theta_{K,i}^*$ bilden.

Da es sich sowohl bei $c_{c,i}$ als auch bei den Schnitten der Störgeometrie um keine Regelgeometrie handelt, gestaltet sich das Finden einer exakten Lösung als aufwändig. Daher wird wiederum eine Diskretisierung durchgeführt, um so zu einer hinreichend genauen Näherungslösung zu gelangen. Demnach werden die Schnitte der Störgeometrie in j_e Punkte $P_{0,ij}$, $j = 1, \dots, j_e$ unterteilt. Für jeden Punkt $P_{0,ij}$ wird dann der komplementäre Lösungsbereich $\Theta_{K,ij}^*$ ermittelt, der alle Drehpunkte zusammenfasst, die zu einer Kollision zwischen der sich um β drehenden Kurve $c_{c,i}$ und $P_{0,ij}$ führen. Die Vereinigung von allen $\Theta_{K,ij}^*$ liefert $\Theta_{K,i}^*$ und in weiterer Folge die eigentlich gesuchte Komplementärmenge $\Theta_{K,i}$. Abb. 5.6 zeigt mit $c_{c,i}$ und einem Punkt $P_{0,ij}$ die Ausgangssituation für die Konstruktion von $\Theta_{K,ij}^*$.

Die Kurve $c_{c,i}$ ist in parametrischer Darstellung durch den Ortsvektor $\mathbf{r}(\mu) = \mathbf{r}_{ij}(\mu)$ gegeben, wobei der Parameterraum $\mu \in [\mu_0, \mu_e]$ durch die Anfangs- und Endpunkte von $c_{c,i}$ begrenzt wird. Der Ausgangspunkt von $\mathbf{r}(\mu)$ liegt im Koordinatenursprung \mathcal{O} des inertialen, kartesischen Koordinatensystems Σ_0 , das durch die Einheitsvektoren \mathbf{x}_0 und \mathbf{y}_0 aufgespannt wird. \mathcal{O} liegt gleichzeitig auf einem Punkt der Störgeometrie $P_{0,ij}$ und darf daher bei einer Drehung von $c_{c,i}$ nicht überstrichen werden. $\mathbf{r}(\mu)$ definiert ebenfalls das bewegte, lokale Koordinatensystem Σ_τ mit seinen Einheitsvektoren \mathbf{x}_τ und \mathbf{y}_τ . \mathbf{x}_τ zeigt in Richtung $\mathbf{r}(\mu)$ und ist daher um $\varphi(\mu)$ zu \mathbf{x}_0 gedreht. Eine Transformation von Σ_τ in Σ_0 erfolgt mit Hilfe der Drehmatrix

²⁹Polyvinylchlorid

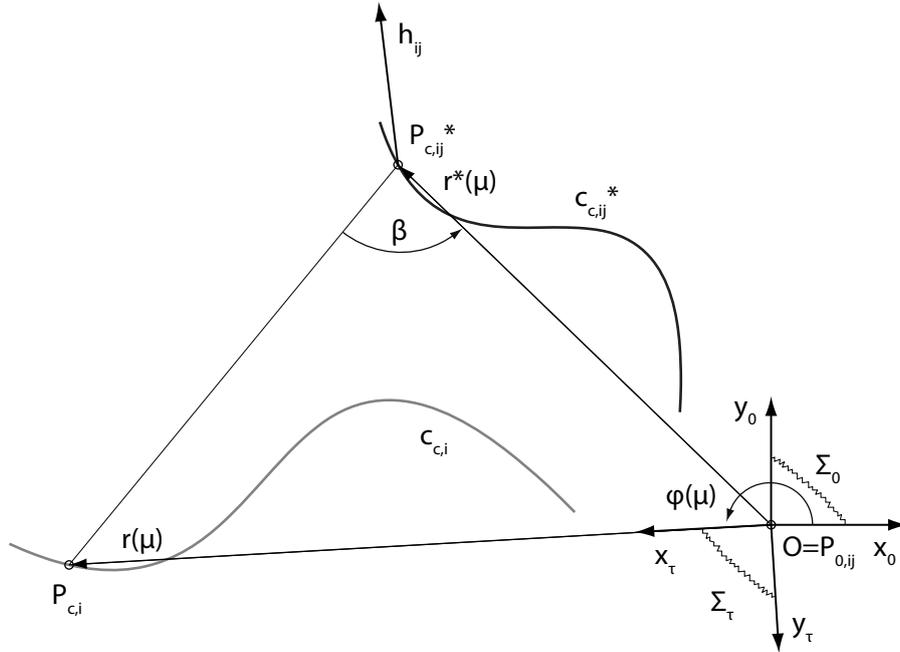


Abbildung 5.6: Konstruktion der Halbgerade h_{ij} , auf der alle Drehpunkte liegen, die bei einer Drehung von $c_{c,i}$ um β zu einer Kollision führen.

$$\mathbf{T}_{\mathbf{O}\tau} = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix}. \quad (5.1)$$

So kann beispielsweise $\mathbf{r} = \mathbf{r}(\mu)$ sowohl im System Σ_τ als auch im Σ_0 dargestellt werden:

$$\mathbf{r}_0 = \mathbf{T}_{\mathbf{O}\tau} \mathbf{r}_\tau. \quad (5.2)$$

Betrachtet man einen Punkt $P_{c,i}$, der auf $c_{c,i}$ liegt, so lässt sich jener Drehpunkt $P_{c,ij}^*$ bestimmen, bei dem $P_{c,i}$ und \mathcal{O} zur Deckung kommen, sobald $c_{c,i}$ mit β um $P_{c,ij}^*$ gedreht wird. Dabei bestimmen $P_{c,i}$, $P_{c,ij}^*$ und \mathcal{O} ein gleichschenkliges Dreieck mit Scheitel $P_{c,ij}^*$ und Scheitelwinkel β . Die Position von $P_{c,ij}^*$ ist durch den Vektor \mathbf{r}^* gegeben und lässt sich in Abhängigkeit von μ und β im System Σ_τ beschreiben:

$$\mathbf{r}_\tau^* = \frac{r(\mu)}{2 \sin\left(\frac{\beta}{2}\right)} \begin{pmatrix} \cos\left(\frac{\pi}{2} - \frac{\beta}{2}\right) \\ -\sin\left(\frac{\pi}{2} - \frac{\beta}{2}\right) \end{pmatrix}. \quad (5.3)$$

Hierbei entspricht $r(\mu)$ der Länge von $\mathbf{r}(\mu)$:

$$r(\mu) = \|\mathbf{r}(\mu)\|. \quad (5.4)$$

Der Drehpunkt $P_{c,ij}^*$ liefert nun den Grenzfall, bei dem die Bedingung einer kollisionsfreien Rotation gerade nicht erfüllt wird. Er ist der Startpunkt der Halbgerade $\mathbf{h}_{ij}(\eta)$. Diese steht normal auf \mathbf{r} und beinhaltet alle Drehpunkte, bei denen es bei einer Drehung mit β zu einem Zusammenfallen von $P_{c,i}$ und \mathcal{O} kommt. Die Parameterdarstellung von $\mathbf{h}(\eta) = \mathbf{h}_{ij}(\eta)$ in Σ_τ ist definiert als:

$$\mathbf{h}_\tau(\eta) = \mathbf{r}_\tau^* + \eta \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \quad \eta \in [0, \infty). \quad (5.5)$$

Wandert nun $P_{c,i}(\mu)$ als Endpunkt von $\mathbf{r}(\mu)$ auf $c_{c,i}$, so überstreicht $\mathbf{h}(\mu, \eta)$ den gesuchten Lösungsbereich $\Theta_{K,ij}^*$, innerhalb dessen die Drehpunkte nicht liegen dürfen. Setzt man Gl. 5.3 in Gl. 5.5 ein und transformiert man das Ergebnis entsprechend Gl. 5.2, so erhält man die Parameterdarstellung von $\mathbf{h}(\mu, \eta)$ in Σ_0 :

$$\mathbf{h}_0(\mu, \eta) = \frac{r(\mu)}{2 \sin\left(\frac{\beta}{2}\right)} \mathbf{T}_{\mathbf{O}\tau}(\varphi) \begin{pmatrix} \cos\left(\frac{\pi}{2} - \frac{\beta}{2}\right) \\ -\sin\left(\frac{\pi}{2} - \frac{\beta}{2}\right) - \eta \end{pmatrix}. \quad (5.6)$$

Abb. 5.7 zeigt für einige Punkte $P_{c,i}(\mu)$ die dazugehörigen Halbgeraden $\mathbf{h}(\mu, \eta)$ und die Grenzkurve $c_{c,ij}^*$, auf der die Startpunkte von $\mathbf{h}(\mu, \eta)$ liegen. $c_{c,ij}^*$ wird durch $\mathbf{r}^*(\mu)$ definiert und ist in Gl. 5.3 beschrieben. Demnach geht $c_{c,ij}^*$ aus $c_{c,i}$ durch eine Drehstreckung mit Zentrum $P_{0,ij}$, Drehwinkel γ und Streckfaktor ξ hervor:

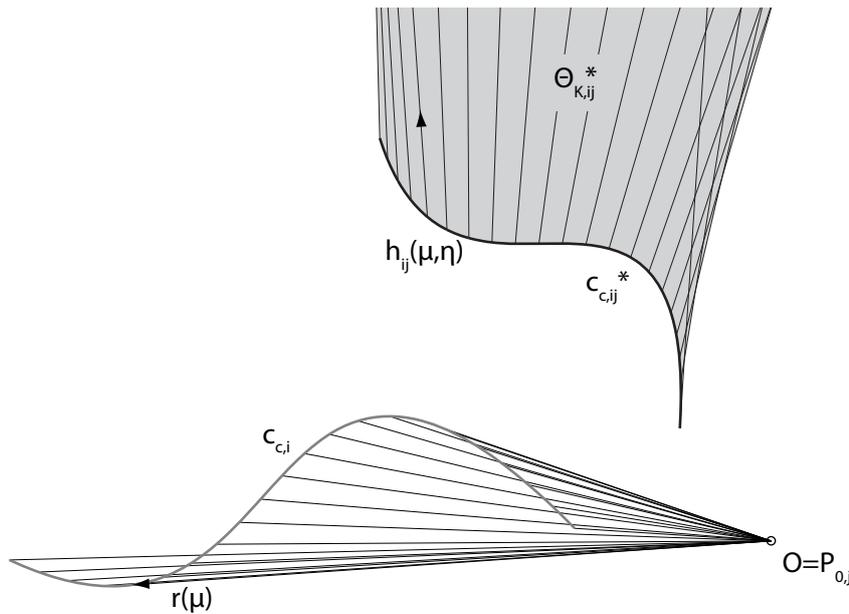
$$\gamma = \frac{\pi - \beta}{2}; \quad \xi = \frac{1}{2 \sin\left(\frac{\beta}{2}\right)}. \quad (5.7)$$

Dadurch kann $c_{c,ij}^*$ als eine Grenze von $\Theta_{K,ij}^*$ direkt ermittelt werden, während die Halbgeraden $\mathbf{h}(\mu, \eta)$ in

$$\mu_k = \mu_0 + \frac{\mu_e - \mu_0}{k_e - 1} k \quad (5.8)$$

mit $k = 0, \dots, k_e - 1$ diskreten Schritten von μ konstruiert werden. Die entstehenden k_e Halbgeraden für jeden $P_{0,ij}$ werden als \mathbf{h}_{ijk} zusammengefasst. In Abb. 5.7 ist eine Diskretisierung von $k_e = 21$ dargestellt, wobei es in diesem Fall zu keiner Überschneidung von $c_{c,ij}^*$ und den \mathbf{h}_{ijk} kommt.

Durch die Konstruktion von $\Theta_{K,ij}^*$ für Punkte $P_{0,ij}$ der Störgeometrie lässt sich $\Theta_{K,i}^*$ als Vereinigungsmenge aller $\Theta_{K,ij}^*$ ermitteln. Bei einer großen Anzahl von j_e sowie von k_e kann sich dieser Vorgang allerdings als sehr rechen- und damit zeitintensiv gestalten. Bei den Karosserieschnitten, die für die Untersuchung der Türeindrehung notwendig sind, handelt es sich zwar um eine beliebige Geometrie, dennoch ähneln sich die betreffenden Bauteile und damit die Gestalt der Schnittgeometrie selbst bei unterschiedlichen Fahrzeugen. Dadurch lässt sich gezielt die Anordnung der betreffenden $P_{0,ij}$ und $P_{c,i}(\mu_k) = P_{c,ik}$ vorgeben und damit die Anzahl

Abbildung 5.7: Konstruktion der Grenzfläche $\Theta_{K,ij}^*$.

von j_e und k_e gering halten. Demnach werden lediglich jene Geometriebereiche, die von einer Kollision betroffen sind, diskretisiert. Abb. 5.8 zeigt anhand der Schnittgeometrie aus Abb. 5.5 eine entsprechende Aufteilung von $P_{0,ij}$ und $P_{c,ik}$ mit $j_e = 5$ und $k_e = 4$.

So kommt es im ersten Teil der Öffnungsbewegung vor allem zu Kollisionen zwischen der Bördelung des Türblechs und der Fugenkante des Kotflügels. Daher werden diese Bereiche fein diskretisiert. Hier gilt es, für jeden $P_{0,ij}$ und $P_{c,ik}$ sowohl die Grenzkurven $c_{c,ij}^*$ als auch die Halbgeraden h_{ijk} zu erzeugen. In der weiteren Öffnungsbewegung können zwei weitere Fälle unterschieden werden. Entweder die Fugenkante des Kotflügels trifft auf das Türaußenblech, oder die Bördelung des Türblechs auf die Störgeometrie von Seitenwand und Kotflügel.

Der erste Fall ist durch die Diskretisierung der Kotflügelkante, wie sie auch in Abb. 5.8 dargestellt ist, bereits ausreichend abgedeckt. So reichen die Grenzkurven $c_{c,ij}^*$, die bei der Drehstreckung der Türaußenhaut bezüglich jedes $P_{0,ij}$ entstehen, für eine hinreichend genaue Konstruktion von $\Theta_{K,i}$. Der Grund dafür liegt in der Form der Schnittgeometrie der Türaußenhaut, die abgesehen von der Bördelung in der Regel keine scharfkantigen Krümmungsänderungen aufweist. Daher kommt es auch zu keinen Überschneidungen der Halbgeraden h_{ijk} und $c_{c,ij}^*$, sodass in diesem Fall die Erzeugung der Halbgeraden h_{ijk} an den Endpunkten von c_{ij} ausreicht. Demnach ist eine feinere Diskretisierung durch zusätzliche $P_{c,ik}$ nicht notwendig.

Im zweiten Fall, also bei einer Kollision der Tür-Verbördelung mit der Störgeometrie abseits der Kotflügelkante, muss die Vorgehensweise angepasst werden. So kann jener Bereich der Störgeometrie, der von einer Kollision betroffen ist, kaum eingegrenzt werden. Folglich müsste die vollständige Störgeometrie fein diskretisiert werden, was wiederum eine große Zahl an Rechenoperationen und damit eine lange Berechnungszeit mit sich bringt. Im Gegensatz dazu lässt sich mit der Bördelung jener Bereich der Türgeometrie leicht eingrenzen, der im Falle einer Kollision die Störgeometrie berührt. Daher wird angenommen, dass die Türgeometrie fixiert ist,

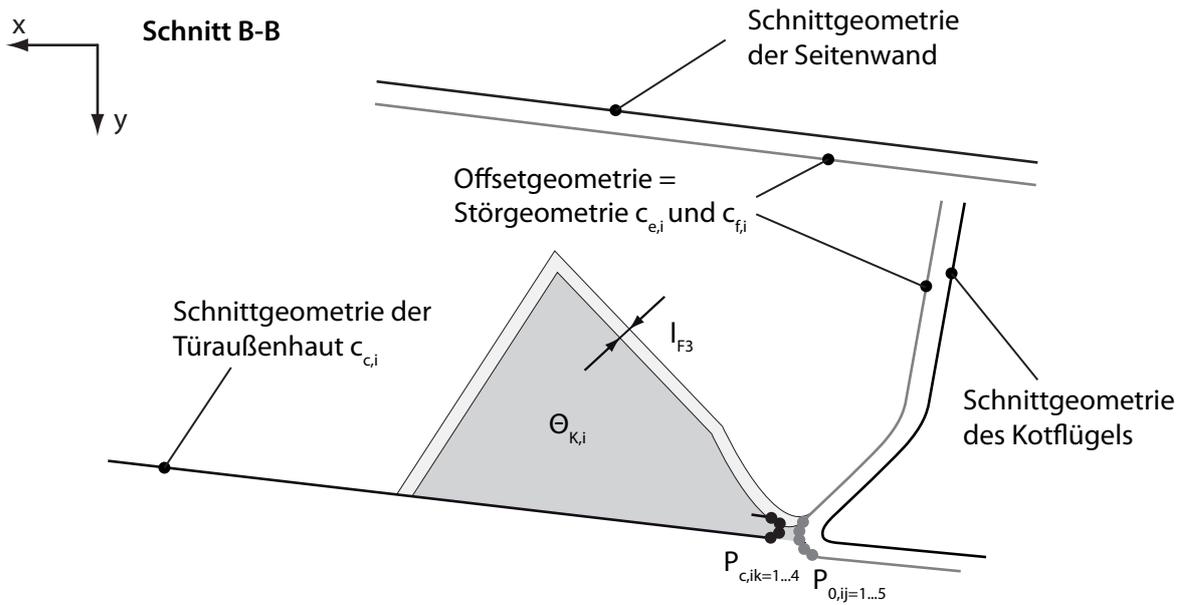


Abbildung 5.8: Diskretisierung der Schnittgeometrie aus Abb. 5.5 durch die Punkte $P_{0,ij}$ und $P_{c,ik}$ sowie die resultierende Grenzfläche $\Theta_{K,i}$.

während die Störgeometrie rotiert. Dieser Wechsel ist zulässig, da es für die Kollisionsprüfung unerheblich ist, ob die Tür- oder die Störgeometrie als beweglich angenommen wird. Dadurch werden aus den $P_{c,ik}$ in Abb. 5.8 Elemente einer Störgeometrie und mit der vormaligen Störgeometrie, bestehend aus den Schnitten von Seitenwand und Kotflügel $c_{e,i}$ bzw. $c_{f,i}$, wird eine Drehstreckung durchgeführt, die wiederum Grenzkurven $c_{e,ik}^*$ und $c_{f,ik}^*$ liefert. Auch hier kann auf eine zusätzliche Konstruktion von Halbgeraden h_{ijk} innerhalb der Endpunkte von $c_{e,ik}$ und $c_{e,ik}$ verzichtet werden, da diese für gewöhnlich zu keiner Verkleinerung des Lösungsbereiches $\Theta_{K,i}$ führt.

Zusammenfassend kann die Lösung des ebenen Syntheseproblems durch eine relativ geringe Anzahl an Stützpunkten $P_{0,ij}$ und $P_{c,ik}$ im Bereich der Tür-Verbördelung sowie der Kotflügelkante erfolgen. Diese bilden den Ausgangspunkt für die Konstruktion der erforderlichen $c_{c,ij}^*$, $c_{e,ik}^*$, $c_{f,ik}^*$ und h_{ijk} , mit denen die komplementäre Lösungsmenge $\Theta_{K,i}^*$ eingegrenzt wird. Mit der Kenntnis von $\Theta_{K,i}^*$ kann in weiterer Folge auf die eigentlich gesuchte Ergänzungsmenge $\Theta_{K,i}$ geschlossen werden, welche alle möglichen Drehpunkte beinhaltet. In Abb. 5.8 wird $\Theta_{K,i}$ dargestellt, wobei zusätzlich auch die Fahrzeugaußenhaut als Begrenzung dient, da für innen liegende Scharniere Lösungen außerhalb der Außenhaut keine Relevanz besitzen. Außerdem wird $\Theta_{K,i}$ durch l_{F3} weiter verkleinert. Mit diesem Offset sollen produktionsbedingte Ungenauigkeiten abgebildet werden, die für eine Verschiebung der Scharnierzpunkte sorgen.

Mit Hilfe des dargestellten Lösungswegs des ebenen Syntheseproblems lassen sich in allen i_e Schnitten entlang der Türfuge die Lösungsbereiche $\Theta_{K,i}$ erzeugen. Diese liefern in weiterer Folge den Lösungsraum Θ_K , der alle möglichen Scharnierachsen zusammenfasst, die zu keiner Kollision führen. So ist eine kollisionsfreie Türöffnung nur dann gegeben, wenn die Scharnierachse durch alle $\Theta_{K,i}$ verläuft. Infolgedessen kann durch eine Normalprojektion aller $\Theta_{K,i}$ in

eine der Schnittebenen und das Bilden einer Schnittmenge aus den projizierten $\Theta_{K,i}$ der Umriss von Θ_K erzeugt werden. Dieser Umriss kann in Richtung der Scharnierachse, welcher durch α_x und α_y gegeben ist, extrudiert werden und umschließt damit Θ_K .

Anhand des hier vorgestellten Ablaufs – also einer Diskretisierung der Geometrie in i_e Schnitten, das dortige Lösen des ebenen Syntheseproblems ebenfalls mit Hilfe von Diskretisierungen in j_e Punkten auf der Störgeometrie sowie k_e Punkten auf der Türgeometrie und das abschließende Bilden einer Schnittmenge aus den ebenen Lösungsmengen – kann schlussendlich das gesuchte Θ_K erzeugt werden. Die Vorgehensweise ist unabhängig vom Reifegrad der Geometrie und es lassen sich auch zusätzliche Anforderungen berücksichtigen. Demnach können ebenfalls Lagetoleranzen der einzelnen Bauteile abgebildet werden. Hierzu sind die betreffenden Bauteile entsprechend den Toleranzvorgaben zu positionieren und mit dem bereits beschriebenen Ablauf kann für jede Toleranzlage ein zusätzliches Θ_K erstellt werden. Die Durchschnittsmenge aller Θ_K deckt schlussendlich die Anforderungen hinsichtlich der Lagetoleranzen ab.

Eine weitere produktionsbedingte Anforderung liegt vor, sobald aushängbare Scharniere verwendet werden. Bei dieser Anordnung wird die Türe beim Zusammenbau mit dem Fahrzeug im geöffneten Zustand eingehängt. Es muss daher sichergestellt sein, dass die geöffnete Türe normal zur Scharnierachse um einen Montagehub verfahrbar ist, ohne dass es zu einer Kollision mit den umliegenden Bauteilen kommt. Auch in diesem Fall ist die Türgeometrie in Einbaulage, also um den vorgegebenen Montagehub angehoben, zu positionieren und wiederum die Kinematik-Synthese durchzuführen. Die entstehenden Einschränkungen fließen wiederum in den Lösungsraum Θ_K ein. Mit der Erstellung von Θ_K ist die zweite Stufe des Lösungswegs aus Abb. 5.3 abgeschlossen.

5.4.5 Stufe drei: Kollisionsprüfung der Scharniere

Als nächster Schritt wird mit Θ_{GK} die Durchschnittsmenge von Θ_G und Θ_K gebildet. Abb. 5.9 zeigt hierzu beide Lösungsbereiche sowie deren Schnittmenge. In diesem Beispiel liefert der Durchschnitt von Θ_G und Θ_K einen Bereich, innerhalb dessen die Scharnierzähnpunkte positioniert werden können, ohne dass die bisher getroffenen Einschränkungen verletzt werden. Demgegenüber kann jedoch durchaus der Fall eintreten, dass die Durchschnittsmenge leer ist und somit kein Scharnierzähnpunkt alle Randbedingungen erfüllt. In diesem Fall kann auf die dritte Stufe des Lösungswegs verzichtet werden und es gilt vielmehr, durch Anpassen von Vorgaben eine nicht-leere Lösungsmenge zu erhalten.

Sobald Θ_{GK} einen nicht-leeren Lösungsbereich liefert, kann in der dritten Stufe die Kollisionsprüfung von Scharniergeometrie und vollständig geöffneter Türe durchgeführt werden. Die Kollisionsprüfung bei geöffneter Türe gestaltet sich jedoch als aufwändiger, als bei der geschlossenen. So ist die Position der geschlossenen Türe bekannt, während die vollständig geöffnete aufgrund der noch nicht festgelegten Scharnierzähnpunkte nicht eindeutig definiert ist. Allerdings gibt es einige Einschränkungen, welche bereits in der ersten Stufe des Lösungswegs getätigt werden und welche die räumliche Lage der Scharniere festlegen. Demnach muss die Ausrichtung der Scharnierachse durch α_x und α_y genauso wie die radiale Ausrichtung der Scharniere

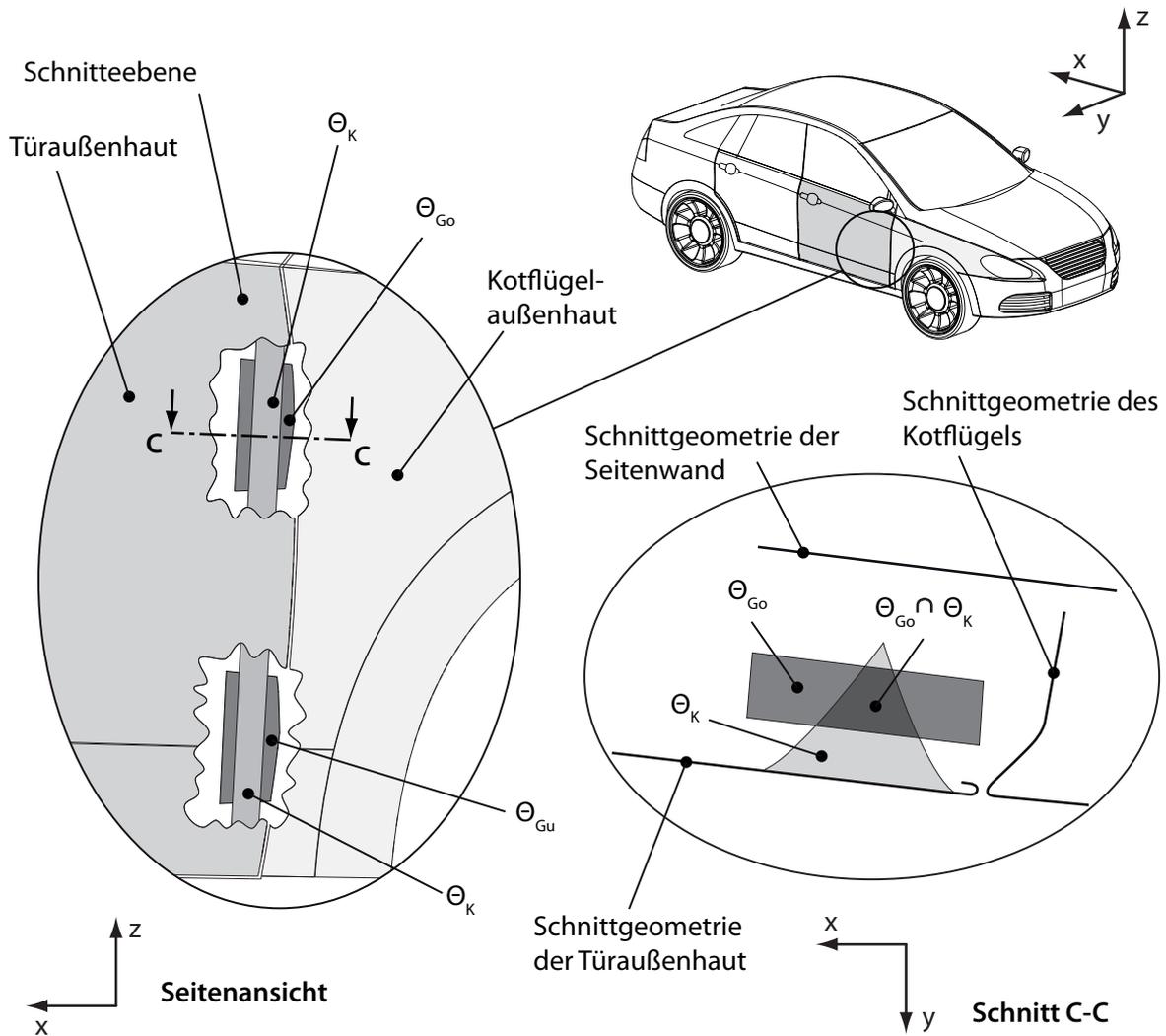


Abbildung 5.9: Diskretisierung von θ_G und θ_K sowie deren Durchschnittsmenge θ_{GK} .

zur Drehachse bekannt sein. Gerade die letzte Eingabe wird in einer frühen Phase oft durch Annahmen und Standardwerte festgelegt.

Anhand dieser Einschränkungen steht die räumliche Lage der Scharniere und insbesondere des kollisionsrelevanten, fahrzeugseitigen Scharnierarms genauso zur Verfügung wie der vorläufige Lösungsraum für Scharnierdrehpunkte θ_{GK} . Es gilt nun jene Scharnierdrehpunkte zu identifizieren, die zu einer Kollision zwischen offener Türe und dem fahrzeugseitigen Scharnierarm führen. Da die Position der Scharniere zwar eingeschränkt, jedoch nicht eindeutig festgelegt ist, muss auf eine punktweise Überprüfung einer möglichen Kollision zurückgegriffen werden. Abb. 5.10 veranschaulicht die punktweise Kollisionsprüfung anhand des oberen Lösungsraums θ_{GKo} . Demnach wird θ_{GKo} mit $m = 1, \dots, m_e$ vertikalen Ebenen, die normal zur Scharnierachsenrichtung stehen, und $n = 1, \dots, n_e$ horizontalen Ebenen, die parallel zur Befestigungsfläche des fahrzeugseitigen Scharnierarms sind, gerastert. Dadurch entstehen $m_e \times n_e$ Gittergeraden g_{mn} . Mit Hilfe der Offsetgeometrie des eingesetzten Scharniers, die wiederum produktionsbedingte

Toleranzen mit einem Freigang l_{F2} berücksichtigt, lässt sich nun entlang jeder g_{mn} überprüfen, ob es zu einer Kollision kommt.

Wie auch in Abb. 5.10 dargestellt, wird die Offsetgeometrie des Scharniers in Bezug auf die Türaußenhaut um β gedreht und so positioniert, dass der Scharnierdrehpunkt auf einer g_{mn} zum Liegen kommt. In weiterer Folge kann nun die Offsetgeometrie entlang von g_{mn} verfahren werden und es lassen sich jene Bereiche identifizieren, bei denen es zu einer Kollision zwischen der verfahrbaren Offsetgeometrie und der Türaußenhaut kommt. Der zulässige Bereich entlang einer g_{mn} lässt sich mit Grenzpunkten P_{mn} markieren. Werden die benachbarten P_{mn} miteinander verbunden, so entsteht eine Grenzfläche S_{lim} , welche jene Scharnierdrehpunkte abgrenzt, die zu einer Kollision mit der geöffneten Türe führen. Um nun zum resultierenden Lösungsbereich Θ_o zu gelangen, wird in Θ_{GK_o} mit S_{lim} beschnitten. Die gleiche Vorgehensweise gilt auch für den unteren Lösungsbereich und führt ausgehend von Θ_{GK_u} zu Θ_u . Anschließend wird sichergestellt, dass jeder Scharnierdrehpunkt aus Θ_o zumindest einen gegenüberliegenden Drehpunkt in Θ_u hat. Diese Einschränkung führt dazu, dass Θ_o und Θ_u in Scharnierachsenrichtung den gleichen Umriss besitzen.

Diese punktweise Konstruktion lässt sich leicht automatisieren, allerdings führt eine hohe Anzahl an g_{mn} zu einer entsprechend langen Berechnungszeit. Daher wird dieser Schritt am Ende des Lösungswegs von Abb. 5.3 durchgeführt, da die bereits zuvor getroffenen Einschränkungen Θ_{GK} verkleinern und folglich kleine Werte von m_e und n_e für eine ausreichend genaue Diskretisierung sorgen.

5.4.6 Bewertung des Lösungswegs

Die dritte Stufe des Lösungswegs aus Abb. 5.3 liefert schlussendlich den gesuchten Lösungsbereich Θ für alle Scharnierdrehpunkte, die zu keiner Kollision zwischen Türe und umliegenden Bauteilen führen. Der vorgestellte Lösungsweg trifft Vereinfachungen lediglich aufgrund von Diskretisierungen und liefert daher zuverlässige und genaue Ergebnisse. Gerade diese Zuverlässigkeit schafft im Vergleich zu rein manuellen, iterativen Verfahren einen wesentlichen Vorteil bei der Bewertung von geänderten Eingaben, wie etwa neue Designflächen der Außenhaut. So kann bei der hier beschriebenen Methode umgehend und sicher ermittelt werden, ob die Lösungsmenge der zulässigen Scharnierdrehpunkte leer ist. Gerade diese Sicherheit bei der Bewertung von Änderungen schafft eine zuverlässige Entscheidungsgrundlage und beschleunigt damit den Änderungsprozess. Im Gegensatz dazu kann bei iterativen Verfahren erst bei einer sehr hohen Anzahl an Parameterkombinationen und einer damit einhergehenden langen Bearbeitungszeit auf eine zu vermeidende, leere Lösungsmenge geschlossen werden.

Der hier beschriebene Ansatz kann – wie bereits erwähnt – die Auslegung der Türöffnung von einem relativ frühen Entwicklungsstand bis über eine Konzeptphase hinaus unterstützen. Infolgedessen wird auch die Eingabe sowohl von einfachen als auch von relativ detaillierten Geometriedaten unterstützt. Die Vielzahl an Einflussparametern bildet den Wissensstand am Ende der Konzeptphase ab. Daher müssen die Parameterwerte einer frühen Konzeptphase, sofern

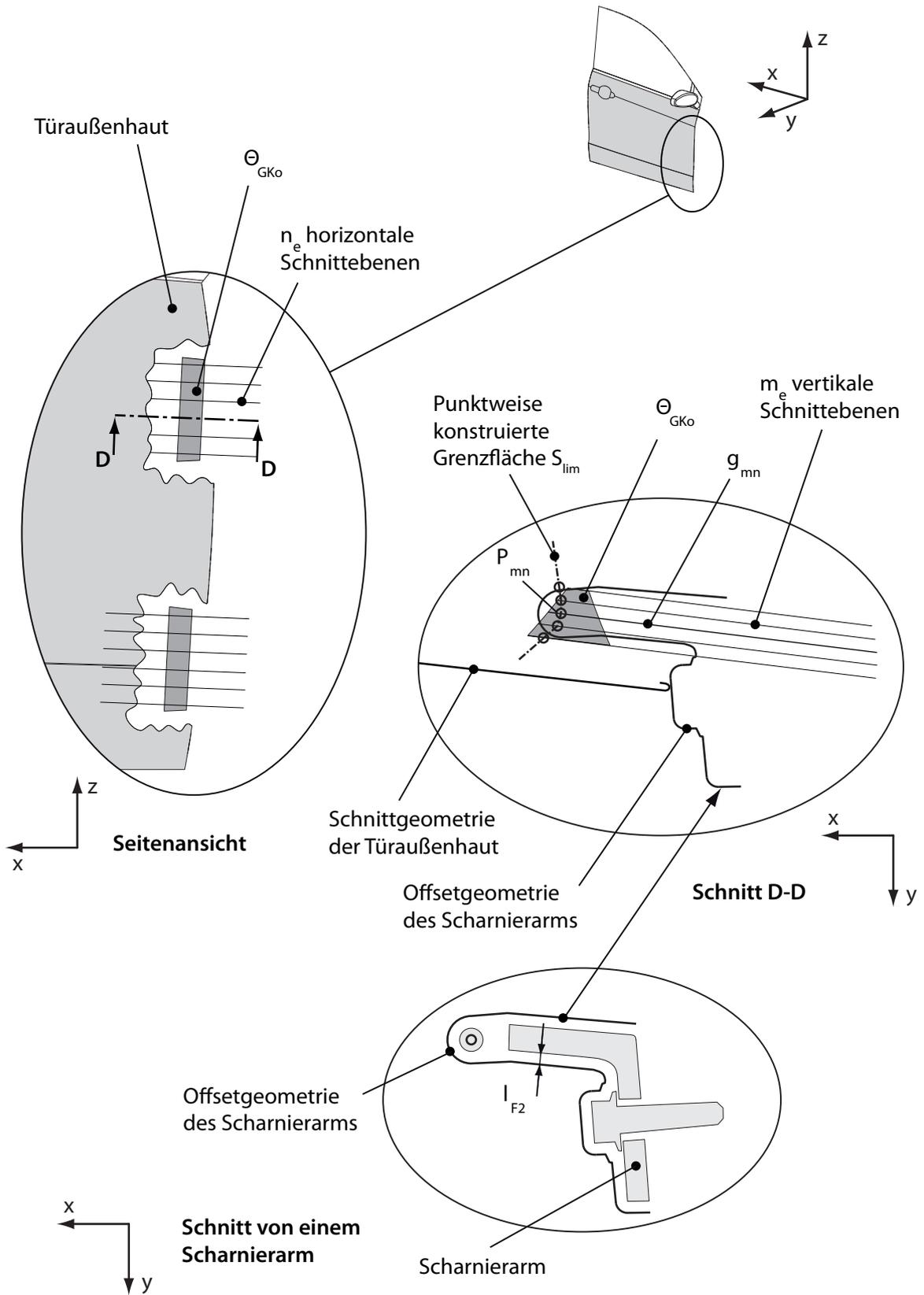


Abbildung 5.10: Punktweise Überprüfung einer Kollision von geöffneter Türe und Scharnier.

sie nicht bereits bekannt sind, mit Standardwerten befüllt werden, um einen vollständigen Parametersatz zu erhalten. Dieser Parametersatz bietet allerdings die Möglichkeit für gezielte Optimierungen von eingangs getätigten Standardwerten. Demnach lässt sich beispielsweise die Ausrichtung der Scharnierachse, die durch α_x und α_y gegeben ist, dahingehend anpassen, dass Θ einen möglichst großen Bereich an zulässigen Scharnierdrehpunkte liefert:

$$\Theta(\alpha_x, \alpha_y) \longrightarrow \max. \quad (5.9)$$

Gleiches gilt auch für andere Parameter, die ebenfalls optimiert werden können. So lässt beispielsweise die Breite der Türfuge, sofern das Geometriemodell der Türaußenhaut diesen Wert als Parameter enthält, minimieren. Dies ist von Interesse, da eine kleine Fugenbreite ein Merkmal für eine qualitativ hochwertige Karosserie darstellt.

Das frühzeitige Befüllen des Parametersatzes führt ebenfalls zu einem Wissenszuwachs im Sinne des *Frontloadings*. Infolgedessen können nach einer erfolgreich durchgeführten Absicherung die vormals auf Standardwerten beruhenden Parameterwerte als Zielvorgaben an andere Fachbereiche weitergegeben werden. Das gleiche gilt für die Auslegungsergebnisse der Türeindrehung, nämlich die beiden abgesicherten Positionen der Türscharniere sowie die sich daraus ergebenden Scharnierachse. Diese werden gleichfalls als wichtige Informationen für die Geometriemodellierung der Türe herangezogen. So dient etwa die Scharnierachse als Referenz für die Positionierung des Fahrzeugschlusses, dessen Öffnungsrichtung normal zur Drehrichtung der Türe steht. Außerdem können ausgehend von den Scharnieren die entsprechenden Anbindungsflächen an Türe und Karosserie erstellt werden.

5.5 Scheibenabsenkung

5.5.1 Anforderungen

Um eine Interaktion mit der unmittelbaren Umwelt ohne das Öffnen der Türe zu ermöglichen und die Erreichbarkeit von externen Bedienelementen wie Parkschraken oder Mautstellen sicherzustellen, werden die Seitenscheiben eines Fahrzeuges offenbar gestaltet. Darüber hinaus bietet eine sich öffnende Seitenscheibe die Möglichkeit, die Frischluftzufuhr und die Innenraumklimatisierung zu beeinflussen. Aus diesen Gründen werden Fahrzeugscheiben in der Regel dahingehend ausgelegt, dass diese aus dem Innenraum von den Insassen leicht geöffnet und wieder geschlossen werden können. Wie auch schon bei der Fahrzeugtüre lässt sich die Öffnungsbewegung der Seitenscheibe prinzipiell beliebig gestalten, allerdings haben sich versenkbare Fahrzeugscheiben durchgesetzt, die entlang eines Fensterrahmens durch einen Schacht in das Innere der Fahrzeugtüre geführt werden. Daher beschränkt sich der hier vorgestellte Ansatz auf diese Standardanordnung und spart komplexere Mechanismen, wie sie unter anderem bei rückwärtigen Cabrioscheiben anzutreffen sind, aus.

Die Hauptaufgabe bei der Auslegung der Scheibenabsenkung besteht darin, die Scheibenfläche versenkbar zu gestalten und die Trajektorie der Absenkbewegung zu ermitteln. Die Trajektorie

dient unter anderem als Eingabe für die Konstruktion des Fensterhebermechanismus [Gfr11]. Außerdem wird der Bewegungsraum der Fensterscheibe für die Bauraumauslegung der Türe benötigt. Als Eingaben für die Scheibenabsenkung steht am Beginn der Konzeptphase lediglich die Scheibenfläche, als Teil der Designaußenhaut, zur Verfügung. Abb. 5.11 zeigt eine typische Scheibenfläche S umrandet von ihren Begrenzungskurven. Diese bestehen aus der Dachkurve c_T , den Begrenzungen von A- und B-Säule c_A bzw. c_B sowie der Brüstungskurve c_K . Es handelt sich hier wiederum um eine vordere Türe. In weiterer Folge wird nur auf die Vordertüre eingegangen, jedoch können die Anforderungen genauso wie der Lösungsweg mit geringem Aufwand an die Gegebenheiten einer Standard-Hintertüre übertragen werden.

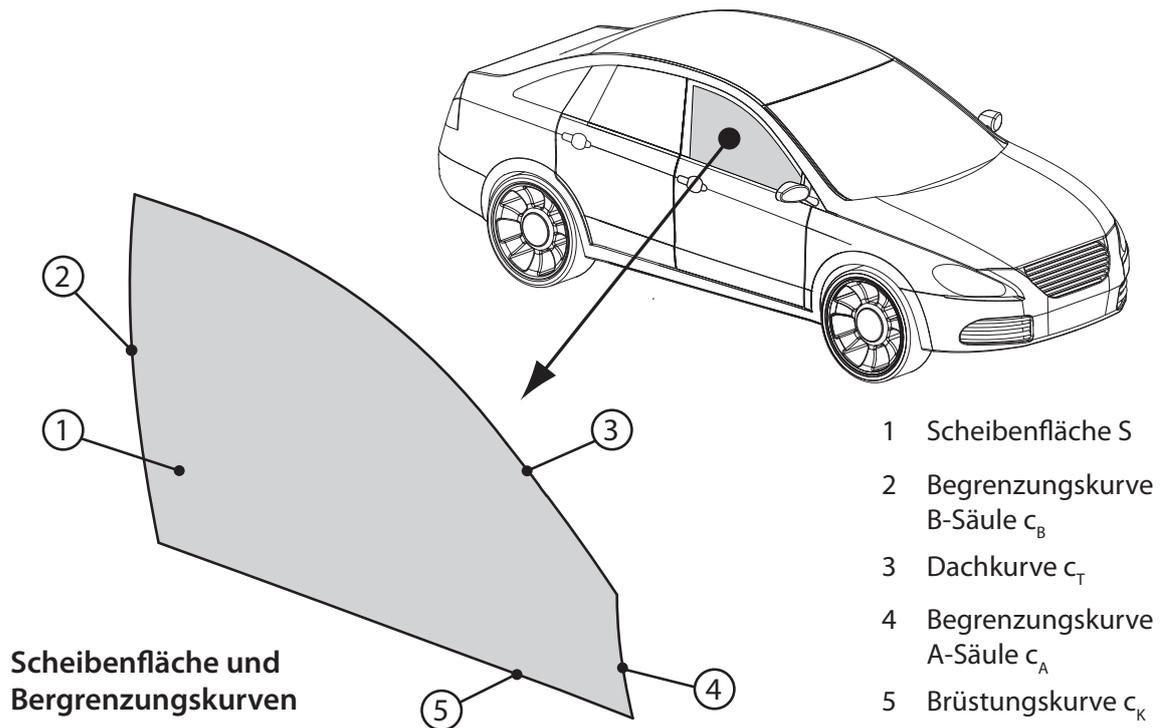


Abbildung 5.11: Darstellung der Scheibenfläche samt Begrenzungskurven.

Randbedingungen, welche die Absenkbewegung einschränken, liefern lediglich die am Türrohbau befestigten Scheibendichtungen. Demnach verfährt die Seitenscheibe innerhalb von Dichtungen, um die Insassen vor Umwelteinflüssen zu schützen. Die Dichtungen sind flexibel, jedoch können sie nur im begrenzten Maße Auslenkungen der Scheibe ausgleichen. So führen unzulässig hohe Auslenkungen während der Scheibenabsenkung zu einer Beschädigung der Dichtungen und in weiterer Folge zu Undichtheit. Übliche Werte für eine zulässige Auslenkung liegen in der Größenordnung von 2 mm. Für die Auslegung während der Absenkung der Seitenscheibe haben die Dichtungen entlang der Begrenzungskurven unterschiedliche Prioritäten. Demnach kann die Dichtung entlang von c_T für die Auslegung vernachlässigt werden, da die Scheibenfläche unmittelbar nach Beginn der Öffnungsbewegung die Dichtung verlässt und entlang von c_T keine Einschränkungen mehr wirken. Ebenfalls vernachlässigt wird die Dichtung entlang von c_A , da diese gewöhnlicherweise kurz ist und daher einen geringen Einfluss hat.

Für die Auslegung der Absenkung werden vor allem die Dichtungen entlang von c_B und c_K als relevant angesehen. Typische Dichtungsgeometrien für beide Anwendungen sind in Abb. 5.12 zusammen mit der Seitenscheibe und ausgewählten Türblechen in Schnitten dargestellt. Die Dichtungen schränken die Scheibenbewegung ein, wobei die Dichtungen entlang von c_K nur normal auf die Scheibenfläche wirken und daher einen weitaus geringeren Einfluss auf die Bewegung der Scheibe haben als die Dichtung entlang von c_B . So kommen hier Dichtungsgeometrien zum Einsatz, welche die Scheibe sowohl normal zur Scheibenfläche als auch in Längsrichtung umschließen. Diese Geometrien schränken die Scheibenbewegung massiv ein und sind auch ein Grund für den sog. Schrägabzug. Hierbei werden die Dichtungen nicht parallel zur Bewegungsrichtung der Scheibe sondern unter einem geringen Winkel angeordnet. Dies führt zu einem Herausfahren der Scheibe aus der Dichtung entlang von c_B , sodass die Scheibe in Längsrichtung nicht mehr an den Dichtungen scheuert. Dadurch kann der Verschleiß verringert werden. Der Winkel ist allerdings so klein, dass die Scheibe weiterhin durch die Dichtungslippen, die in Normalrichtung wirken, geführt wird.

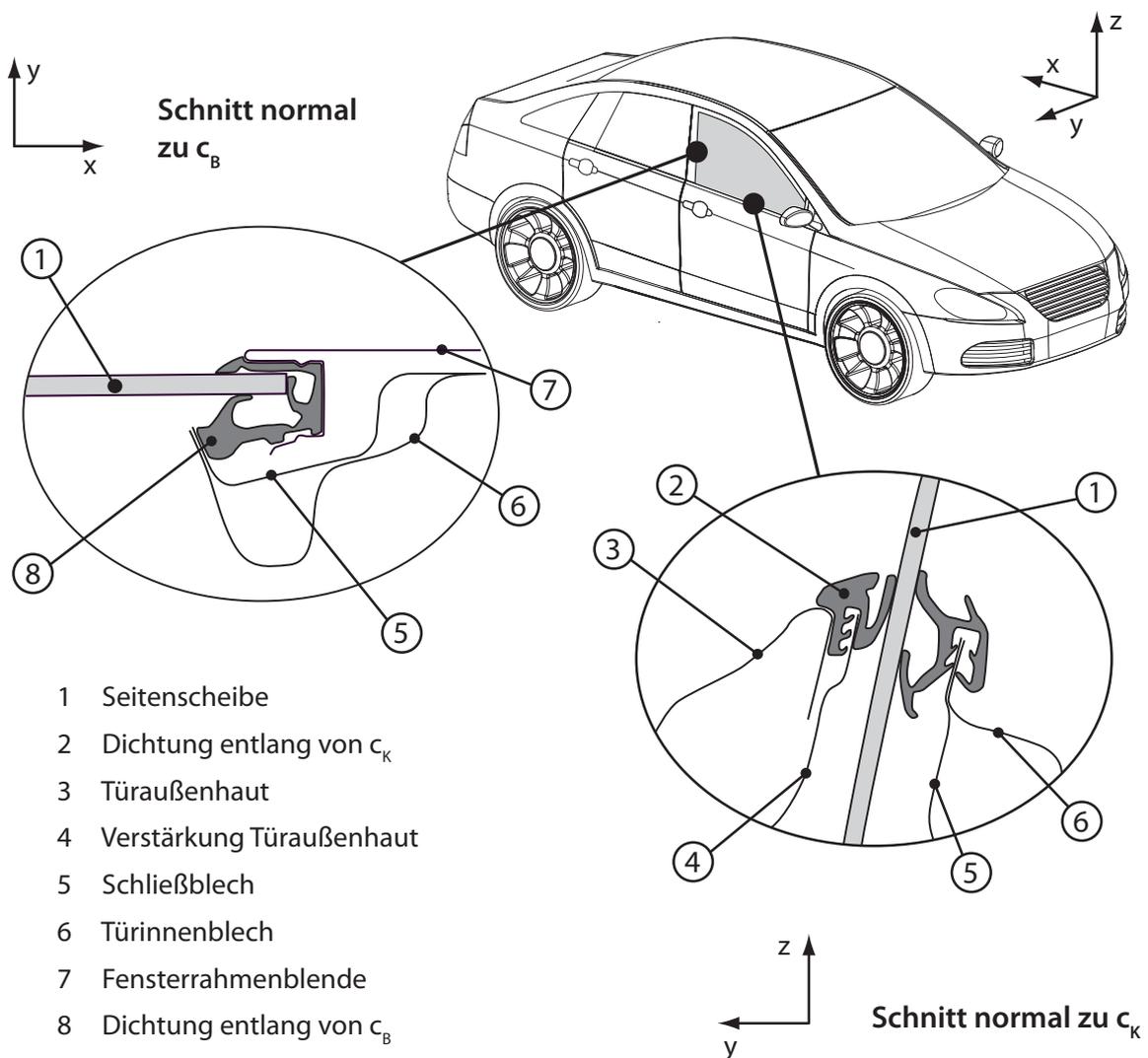


Abbildung 5.12: Schnittdarstellungen der Dichtungen entlang von c_B und c_K .

Die Einschränkungen aufgrund der Scheibendichtungen und deren limitierte Auslenkung resultieren in der Forderung, dass Fahrzeugscheiben bei der Öffnung in sich selbst verfahren müssen. Diese Anforderung stellt hohe Ansprüche sowohl an die Scheibenform als auch an die Sollbewegung der Scheibe. Ausschließlich Schraubflächen sowie Schraubbewegungen können diesen Anforderungen genügen. Dies schließt Sonderformen von Schraubflächen, wie Rotations- und Zylinderflächen gleichermaßen mit ein [GLH⁺11]. Handelt es sich bei der Scheibenfläche also um eine Schraubfläche S und wird diese gemäß der zugehörigen Schraubbewegung M bewegt, so verfährt die Scheibe in sich selbst und es kommt zu keinen Auslenkungen von Dichtungen, die normal zur Scheibenfläche stehen. Handelt es sich bei c_B gleichfalls um die Bahnkurve eines Punktes $P_B \in S$ bei der Schraubbewegung M und um keine beliebige Randkurve, so kommt es auch zu keinen Auslenkungen an den Dichtungslippen entlang von c_B , die längs zur Scheibenfläche stehen.

Demnach lassen sich Scheibenflächen und Trajektorien konstruieren, die zu keinen Auslenkungen der Dichtungen führen. Dennoch stehen gerade in der frühen Entwicklungsphase oftmals keine entsprechenden Flächen zur Verfügung. So kann nicht sichergestellt werden, dass die Designflächen der Außenhaut auch Schraubflächen sind, da es sich mitunter um gescannte Flächen von physischen Modellen oder um Freiformflächen handelt, die auf Basis von rationalen Kurven erstellt werden [Ges01, Klu07]. Diese Flächen stehen einer auslenkungsfreien Absenkung entgegen, da sie in der Regel nicht in sich bewegt werden können. Darüber hinaus wird aus ästhetischen Gründen die Begrenzungskurve c_B – selbst wenn es sich bei der Scheibenfläche um eine Schraubfläche handelt – nicht als eine Bahnkurve der Schraubfläche gewählt. Dadurch wird bewusst eine Verformung der Dichtungen in Kauf genommen.

Die Aufgabe bei der Auslegung der Scheibenabsenkung besteht nun darin, aus den gegebenen Designflächen eine Schraubfläche zu ermitteln, welche die Designfläche und damit die künstlerische Vorgabe des Designers bestmöglich approximiert. Darüber hinaus soll die optimale Absenkbewegung ermittelt werden, wobei neben der Scheibenfläche auch der Verlauf von c_B berücksichtigt wird.

5.5.2 Lösungsweg in zwei Stufen

Abb. 5.13 zeigt schematisch den Lösungsweg für die Auslegung der Scheibenabsenkung samt den erforderlichen Eingaben. Demnach wird in der ersten Stufe auf Basis einer vorläufigen Design-Scheibenfläche S_D eine Schraubfläche S_S approximiert, welche die Vorgaben des Designers bestmöglich widerspiegelt. S_D kann – wie bereits erwähnt – als gescannte Fläche oder als Freiformfläche vorliegen. Neben S_S liegt als Ergebnis ebenfalls die dazugehörige Schraubbewegung M_S vor. Anhand von M_S kann auf eine Begrenzungskurve c_B geschlossen werden, mit der es zu keinen Auslenkungen der B-Säulen-Dichtung kommt. Diese Ergebnisse dienen unter anderem als Eingaben für spätere Designentwürfe und CAS-Modelle.

Sollte sich die in der ersten Stufe des Lösungswegs ermittelte Begrenzungskurve c_B aufgrund von ästhetischen Gesichtspunkten nicht für einen Designentwurf eignen, kann in der zweiten

Stufe des Lösungswegs eine Schraubbewegung M_{SB} berechnet werden, die neben einer Scheibenfläche S auch die Vorgabe einer Begrenzungskurve c_B berücksichtigt. Hierbei wird mit Hilfe von Optimierungsroutinen eine Schraubbewegung M_{SB} ermittelt, welche auf die Anforderung einer minimalen Dichtungsauslenkung abzielt. Anhand der Schraubbewegung M_{SB} kann auf eine Schraubfläche S_{SB} geschlossen werden, die wiederum die Designvorgaben bestmöglich widerspiegelt und gleichzeitig zu geringen Auslenkungen der Dichtungen führt. Da die erste Stufe des Lösungswegs nicht zwingend vor der zweiten erfolgen muss, ist die Verbindungslinie zwischen diesen Stufen in Abb. 5.13 nur gestrichelt dargestellt. Darüber hinaus kommt in beiden Stufen der gleiche Lösungsalgorithmus zum Einsatz, so dass die Stufen auch unabhängig voneinander betrachtet werden können.

Die Ergebnisse M_{SB} sowie gegebenenfalls S_{SB} dienen im Anschluss an deren Auslegung für die Beurteilung, ob die Absenkbewegung zu einer unzulässigen Auslenkung der Dichtungen führt. Hierzu sind ebenfalls die maximalen Auslenkungen der Dichtungen als Eingaben notwendig. Da diese Werte jedoch nur für eine Beurteilung verwendet werden und die Ergebnisse M_{SB} und S_{SB} nicht beeinflussen, sind sie in Abb. 5.13 nicht separat angeführt. Die Bewertung dient unter anderem als Diskussionsgrundlage für allfällige Änderungen des Designs oder der Dichtungen. M_{SB} liefert in Kombination mit der Scheibenfläche ebenfalls den Bewegungsraum der Scheibe, der wiederum als Eingabe für die Bauraumauslegung der Türe dient. Außerdem wird M_{SB} für die Auslegung des Fensterhebermechanismus verwendet.

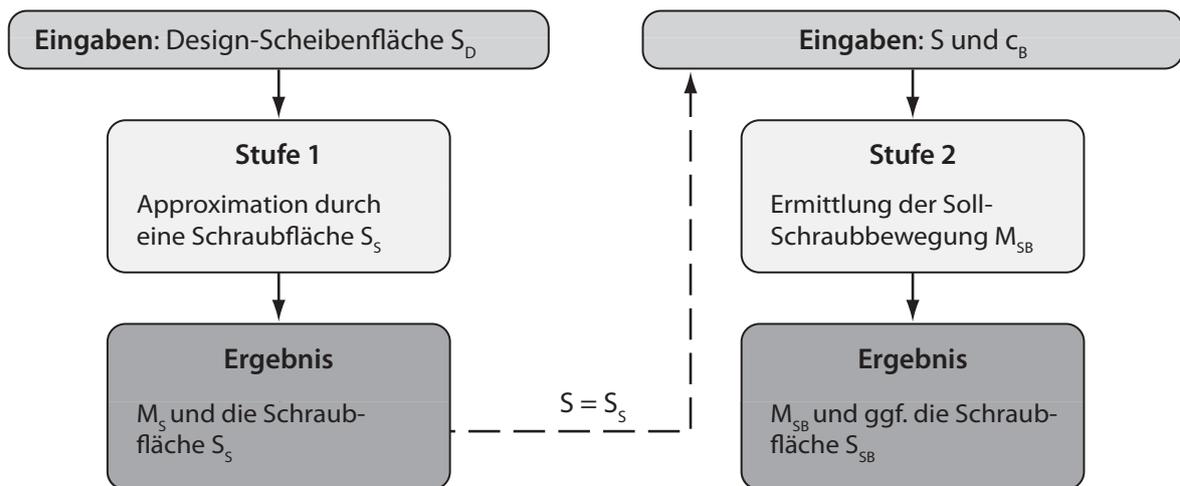


Abbildung 5.13: Schematische Vorgehensweise zur Ermittlung einer approximierten Scheibenfläche S_S und der Sollbewegung M_{SB} .

Der hier beschriebene Ansatz wird wiederum bewusst allgemein gehalten. Dadurch lässt sich die Auslegung auch mit detaillierten Geometriemodellen durchführen und kann folglich die Produktentwicklung auch über die Konzeptphase hinaus unterstützen. Sowohl der Lösungsweg als auch die zu Grunde liegenden Berechnungsalgorithmen wurden mehrfach veröffentlicht [HMH⁺10, GLH⁺11]. Im Besonderen ist die Patentierung der Berechnungsmethode hervorzuheben, die hierdurch den Neuigkeitsgrad dieses Ansatzes unterstreicht [GLH⁺10]. Vergleichbare Ansätze sind in der Literatur zwar vorhanden, haben allerdings Schwachpunkte. So werden die jeweiligen Lösungswege nur unzureichend dargestellt, wie etwa durch *Klemmstein* [Kle10],

oder es werden manuelle, iterative Herangehensweisen vorgeschlagen, wie etwa durch *Franzen* [Fra08]. Demgegenüber liefert der hier vorgestellte Lösungsansatz einen detailliert beschriebenen und automatisierbaren Lösungsweg.

5.5.3 Approximation durch Schraubflächen

In der ersten Stufe der Vorgehensweise aus Abb. 5.13 wird ein Verfahren zur Approximation von abgetasteten Flächen durch Rotations- oder Schraubflächen verwendet [PR98]. Dieses Verfahren wird vielfach im Bereich des *Reverse Engineering*, also der Rekonstruktion von bestehenden Systemen – in diesem Fall Geometriedaten – angewendet [OS04]. Außerdem kommt es zu Anpassungen und Weiterentwicklungen wie etwa durch *Pottmann* [PHOW04]. In der zweiten Stufe der Vorgehensweise wird ebenfalls auf diesen Ansatz zurückgegriffen, jedoch wird dieser erweitert, um die Anforderungen der Begrenzungskurve c_B zu berücksichtigen. Der eingesetzte Berechnungsalgorithmus greift auf Methoden der Liniengeometrie zurück und verwendet insbesondere die Abbildung von Geraden in Punkte eines sechs-dimensionalen Raums. Diese sogenannten *Plückerkoordinaten* werden in Anh. A.1 beschrieben. An dieser Stelle wird auf die Eigenschaften von Schraubbewegungen eingegangen, um dadurch ein besseres Verständnis für den verwendeten Ansatz zu liefern.

Bei einer Schraubbewegung handelt es sich um eine Überlagerung einer reinen Rotation und einer Translation in Richtung der Rotationsachse, wobei die Schiebstrecke eine lineare Funktion des Drehwinkels ist. Wird eine Schraubbewegung M im kartesischen Koordinatensystem Σ_0 mit den Koordinatenrichtungen \mathbf{x} , \mathbf{y} und \mathbf{z} betrachtet, so lässt sich für jeden Punkt P_g , der durch den Ortsvektor $\mathbf{p} = [p_1, p_2, p_3]^T$ gegeben ist, eine Bahnkurve c_H in Form einer Helix bzw. Schraubenlinie bestimmen, Abb. 5.14. Das Geschwindigkeitsvektorfeld einer solchen Schraubbewegung lässt sich durch

$$\dot{\mathbf{p}} = \mathbf{w} \times \mathbf{p} + \mathbf{v} \quad (5.10)$$

beschreiben. Hierbei ist \mathbf{p} der Ortsvektor eines beliebigen Punktes, $\dot{\mathbf{p}}$ sein Geschwindigkeitsvektor und $\mathbf{v} = [v_1, v_2, v_3]^T$ sowie $\mathbf{w} = [w_1, w_2, w_3]^T$ sind konstante Vektoren. \mathbf{w} kann als Vektor der Winkelgeschwindigkeit interpretiert werden und ist parallel zur Schraubachse d . Anhand der Eigenschaften von \mathbf{v} und \mathbf{w} lässt sich M kategorisieren:

- Unter den Bedingungen, dass $\mathbf{w} \neq [0, 0, 0]^T$ und $\langle \mathbf{w}, \mathbf{v} \rangle \neq 0$, ist M tatsächlich eine Schraubung. Hierbei kennzeichnet $\langle \cdot, \cdot \rangle$ das Skalarprodukt zweier Vektoren. In diesem Fall liefert

$$b = 2\pi \frac{\langle \mathbf{w}, \mathbf{v} \rangle}{\langle \mathbf{w}, \mathbf{w} \rangle} \quad (5.11)$$

die sog. Gewindesteigung oder Ganghöhe,

$$\mathbf{d} = \mathbf{w} \quad (5.12)$$

beschreibt den Richtungsvektor der Schraubachse d und

$$\mathbf{a} = \frac{\mathbf{w} \times \mathbf{v}}{\langle \mathbf{w}, \mathbf{w} \rangle} \quad (5.13)$$

ist der Ortsvektor zu einem Punkt $P_a \in d$.

- M beschreibt eine Rotation wenn $\mathbf{w} \neq [0, 0, 0]^T$ aber $\langle \mathbf{w}, \mathbf{v} \rangle = 0$. Die Rotationsachse berechnet sich in gleicher Art und Weise wie bei einer Schraubachse.
- Bei M handelt es sich um eine Translation parallel zu \mathbf{v} nur wenn $\mathbf{w} = [0, 0, 0]^T$.

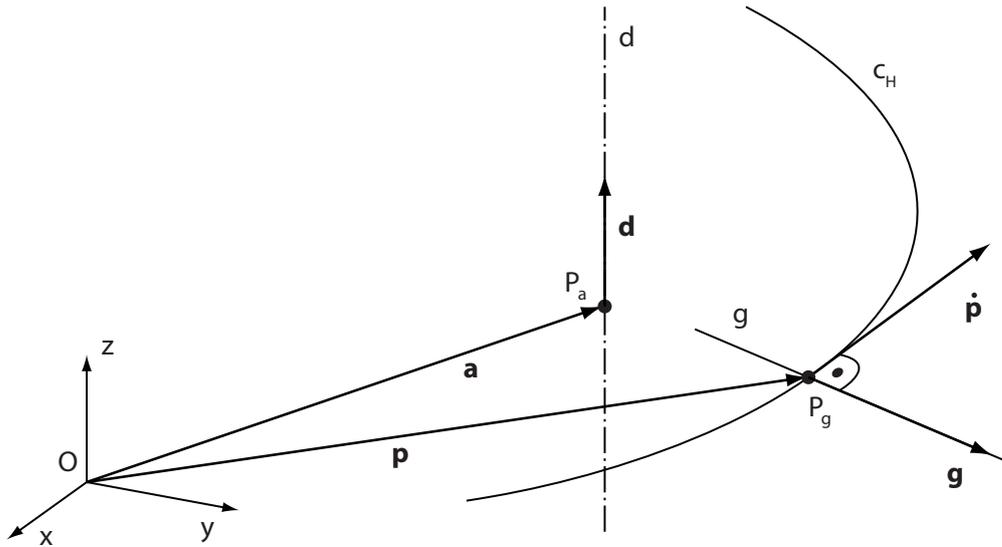


Abbildung 5.14: Darstellung einer Helix c_H samt Schraubachse d und einer Normalen g zu c_H durch den Punkt P_g .

Wird nun, wie auch in Abb. 5.14 dargestellt, eine Normale g angenommen, die durch P_g geht und normal auf c_H steht, so verschwindet das Skalarprodukt des Tangentenvektors $\dot{\mathbf{p}}$ und des Richtungsvektors \mathbf{g} der Geraden g :

$$0 = \langle \dot{\mathbf{p}}, \mathbf{g} \rangle. \quad (5.14)$$

Umformen liefert

$$\langle \dot{\mathbf{p}}, \mathbf{g} \rangle = \langle \mathbf{w} \times \mathbf{p} + \mathbf{v}, \mathbf{g} \rangle = \det(\mathbf{w}, \mathbf{p}, \mathbf{g}) + \langle \mathbf{v}, \mathbf{g} \rangle = \langle \mathbf{w}, \mathbf{p} \times \mathbf{g} \rangle + \langle \mathbf{v}, \mathbf{g} \rangle. \quad (5.15)$$

Da \mathbf{g} dem Richtungsvektor der Geraden g und \mathbf{p} dem Ortsvektor zu einem Punkt $P_g \in g$ entspricht, sind $\mathbf{g} = [g_1, g_2, g_3]^T$ und $\bar{\mathbf{g}} = \mathbf{p} \times \mathbf{g} = [\bar{g}_1, \bar{g}_2, \bar{g}_3]^T$ die *Plückervektoren* von g (siehe auch Anh. A.1). Demnach führen Gl. 5.14 und Gl. 5.15 zu

$$\langle \mathbf{v}, \mathbf{g} \rangle + \langle \mathbf{w}, \bar{\mathbf{g}} \rangle = 0. \quad (5.16)$$

Eine wesentliche Eigenschaft von *Plückervektoren* besteht im Erfüllen der homogenen, quadratischen Gleichung

$$\mathcal{Q} : \quad \langle \mathbf{g}, \bar{\mathbf{g}} \rangle = g_1 \bar{g}_1 + g_2 \bar{g}_2 + g_3 \bar{g}_3 = 0. \quad (5.17)$$

Werden, wie durch *Pottmann* vorgeschlagen [PHOW04], nur *normalisierte Plückervektoren* verwendet, so hat \mathbf{g} außerdem die Länge 1,

$$\mathcal{Q}^* : \quad \langle \mathbf{g}, \mathbf{g} \rangle = g_1^2 + g_2^2 + g_3^2 = 1. \quad (5.18)$$

Mit der Interpretation des Sechstupel der *Plückerkoordinaten* $[\mathbf{g}, \bar{\mathbf{g}}]$ als Koordinaten eines Punktes G in einem sechs-dimensionalen Raum \mathbb{R}^6 lässt sich die Abbildung

$$\mathcal{K} : \quad g \longrightarrow G = \mathcal{K}(g). \quad (5.19)$$

zwischen der Menge aller orientierten Geraden eines drei-dimensionalen Raumes \mathbb{R}^3 und einer vier-dimensionalen Mannigfaltigkeit von Punkten im \mathbb{R}^6 erstellen. Bei dieser Abbildung haben die Koordinaten $g_1, g_2, g_3, \bar{g}_1, \bar{g}_2, \bar{g}_3$ weiterhin die Bedingungen aus Gl. 5.17 und 5.18 zu erfüllen und liegen daher auf der Verschneidung \mathcal{G} der beiden quadratischen Flächen \mathcal{Q} und \mathcal{Q}^* im \mathbb{R}^6 . Diese Verschneidung $\mathcal{G} = \mathcal{Q} \cap \mathcal{Q}^*$ liefert die notwendige vier-dimensionale Mannigfaltigkeit. \mathcal{Q} kann als quadratischer Kegel des \mathbb{R}^6 aufgefasst werden, dessen Scheitel im Koordinatenursprung liegt, während \mathcal{Q}^* eine quadratische Zylinderfläche im \mathbb{R}^6 ist. Eine wesentliche Eigenschaft der Abbildung \mathcal{K} liegt in der Bijektivität. Demnach kann jedem Punkt $G \in \mathcal{G}$ eindeutig eine orientierte Gerade des \mathbb{R}^3 zugeordnet werden und umgekehrt.

Wird nun Gleichung 5.16 betrachtet, die für jede Bahnnormale g der durch \mathbf{v} und \mathbf{w} bestimmten Schraubbewegung erfüllt sein muss, so ergibt sich eine lineare und homogene Bedingung für die *Plückerkoordinaten* $g_1, g_2, g_3, \bar{g}_1, \bar{g}_2, \bar{g}_3$ von g . Diese Bedingung kann im sechs-dimensionalen Raum \mathbb{R}^6 als Hyperebene \mathcal{H}_0 durch den Koordinatenursprung \mathcal{O} interpretiert werden, auf der der Punkt $G = \mathcal{K}(g)$ liegen muss. Für den umgekehrten Fall, also das Vorhandensein einer ausreichenden Zahl an Kurven- bzw. Flächennormalen g einer Schraubenlinie oder -fläche, bedeutet dies, dass die dazugehörige Hyperebene \mathcal{H}_0 konstruiert werden kann. Hierzu sind mindestens 5 Normalen g_1, \dots, g_5 notwendig, deren Abbildungen G_1, \dots, G_5 eine allgemeine Lage besitzen. Nach der Berechnung von \mathcal{H}_0 lassen sich folglich die Vektoren \mathbf{v} und \mathbf{w} sowie mit Gl. 5.11, 5.12 und 5.13 die Gewindesteigung b , der Richtungsvektor der Schraubachse \mathbf{d} und der Ortsvektor zu einem Punkt auf der Schraubachse \mathbf{a} ermitteln.

5.5.4 Stufe eins: Approximation einer Schraubfläche als Scheibenfläche

Anhand des zuvor beschriebenen Berechnungsansatzes zur Ermittlung von Schraubbewegungen kann in der ersten Stufe der Vorgehensweise aus Abb. 5.13 ausgehend eine gegebene Designfläche S_D durch eine Schraubfläche S_S approximiert werden. Hierzu wird mit M_S jene

Schraubbewegung gesucht, mit der S_D optimal angenähert werden kann. Für die Erzeugung von S_S ist in weiterer Folge zusätzlich eine erzeugende Kurve c_E notwendig, wobei sich hierfür insbesondere die Randkurven c_K oder c_T eignen, da diese auf der Außenhaut liegen und damit den Vorgaben des Designs entsprechen.

Wird davon ausgegangen, dass S_D bereits eine Schraubfläche ist, so sind die Flächennormalen $g_i = 1, \dots, i_e$ gleichzeitig auch Normalen der Schraubbewegung M_S . Die Flächennormalen g_i lassen sich mit der Abbildung \mathcal{K} als Punkte $G_i = \mathcal{K}(g_i), i = 1, \dots, i_e$ in \mathbb{R}^6 überführen. Wird eine ausreichende Anzahl $i_e > 5$ an Punkten G_i erstellt, so lässt sich die zu Grunde liegende Schraubbewegung M_S ermitteln. Daher wird S_D , wie in Abb. 5.15 (links) dargestellt, gerastert und zu jedem Rasterpunkt die entsprechende Flächennormale g_i und in weiterer Folge die dazugehörigen G_i erstellt. Sofern es sich bei S_D tatsächlich um eine Schraubfläche handelt, liegen alle G_i exakt auf der Hyperebene \mathcal{H}_0 . Sollte es sich jedoch um eine Fläche handeln, die nicht als Schraubfläche erstellt wurde oder um eine Fläche, die auf Basis eines abgetasteten physischen Designmodells angenähert wurde, so bilden die G_i eine Punktwolke und sie liegen nicht in einer gemeinsamen durch \mathcal{O} gehenden Hyperebene \mathcal{H}_0 des \mathbb{R}^6 .

Um nun ausgehend von dieser Punktwolke auf eine Schraubbewegung zu schließen, gilt es eine \mathcal{H}_0 zu berechnen, welche die Punktwolke am besten annähert. \mathcal{H}_0 repräsentiert gleichzeitig auch die gesuchte Schraubbewegung M_S , mit der S_D approximiert werden kann. Wie bereits in Gl. 5.16 dargestellt, wird \mathcal{H}_0 definiert als

$$\mathcal{H}_0 : \quad \langle \mathbf{v}, \mathbf{g} \rangle + \langle \mathbf{w}, \bar{\mathbf{g}} \rangle = 0, \quad (5.20)$$

wobei $[\mathbf{v}^T, \mathbf{w}^T]^T = [v_1, v_2, v_3, w_1, w_2, w_3]^T$ dem normalisierten Normalvektor von \mathcal{H}_0 entspricht und $[\mathbf{g}^T, \bar{\mathbf{g}}^T]^T = G$ einen Punkt auf \mathcal{H}_0 beschreibt. Im vorliegenden Fall – also einer Punktwolke mit i_e Punkten G_i in ihren *Plückerkoordinaten* $[g_{1i}, g_{2i}, g_{3i}, \bar{g}_{1i}, \bar{g}_{2i}, \bar{g}_{3i}]^T$ – gilt es die Koeffizientenvektoren \mathbf{v} und \mathbf{w} zu ermitteln. Hierzu werden die Quadratabstände aller G_i zu einer \mathcal{H}_0 minimiert. Der Abstand h_i eines G_i zu einer \mathcal{H}_0 ergibt sich aus:

$$h_i = \sqrt{(\langle \mathbf{v}, \mathbf{g}_i \rangle)^2 + (\langle \mathbf{w}, \bar{\mathbf{g}}_i \rangle)^2}. \quad (5.21)$$

Demnach ist die Summe

$$e(\mathbf{v}, \mathbf{w}) = \sum_{i=1}^{i_e} h_i^2 \quad (5.22)$$

der quadratischen Abstände unter der Randbedingung

$$f(\mathbf{v}, \mathbf{w}) = v_1^2 + v_2^2 + v_3^2 + w_1^2 + w_2^2 + w_3^2 - 1 = 0 \quad (5.23)$$

zu minimieren. Es handelt sich hierbei also um eine Näherung mit einer *Methode der kleinsten Quadrate* oder *Least Squares Fit*. Die Nebenbedingung 5.23 garantiert im übrigen, dass die Normalvektoren von \mathcal{H}_0 die Länge 1 besitzen. $e(\mathbf{v}, \mathbf{w})$ lässt sich auch in Matrizenform darstellen

$$e(\mathbf{v}, \mathbf{w}) = \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix}^T \mathbf{A} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix}, \quad (5.24)$$

wobei die 6×6 Matrix \mathbf{A} die *Plückerkoordinaten* der Punktwolke G_i enthält:

$$\mathbf{A} = \begin{bmatrix} \sum_{i=1}^{i_e} g_{1i}^2 & \sum_{i=1}^{i_e} g_{1i}g_{2i} & \cdots & \sum_{i=1}^{i_e} g_{1i}\overline{g_{3i}} \\ \sum_{i=1}^{i_e} g_{2i}g_{1i} & \sum_{i=1}^{i_e} g_{2i}^2 & \cdots & \sum_{i=1}^{i_e} g_{2i}\overline{g_{3i}} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{i_e} \overline{g_{3i}}g_{1i} & \sum_{i=1}^{i_e} \overline{g_{3i}}g_{2i} & \cdots & \sum_{i=1}^{i_e} \overline{g_{3i}}^2 \end{bmatrix}. \quad (5.25)$$

Mit der Einführung des Lagrange-Multiplikators λ lässt sich dieses Optimierungsproblem auf ein Eigenwertproblem zurückführen. Der Eigenvektor von \mathbf{A} mit dem kleinsten Eigenwert λ^* liefert die gesuchte Lösung. Wird dieser Eigenvektor normiert, genügt er ebenfalls der Bedingung aus Gl. 5.23.

Mit dem dadurch gefundenen Lösungsvektor $[\mathbf{v}^T, \mathbf{w}^T]^T$ lässt sich nun die Schraubbewegung M_S charakterisieren. Demnach liefern die Gl. 5.11, 5.12 und 5.13 die für eine Darstellung im \mathbb{R}^3 und damit im CAD-System relevanten Werte für die Gewindesteigung b , den Richtungsvektor der Schraubachse \mathbf{d} und den Ortsvektor zu einem Punkt auf der Schraubachse \mathbf{a} . Mit der Information von M_S kann nun jene Schraubfläche S_S erstellt werden, die den Anforderungen des Designs am besten folgt. Hierzu wird eine erzeugende Kurve c_E entsprechend der Schraubung M_S bewegt und überstreicht dabei eine S_D approximierende Schraubfläche S_S . Als mögliche Kurven für c_E eignen sich –wie bereits erwähnt– die Randkurven c_K oder c_T .

5.5.5 Stufe zwei: Berechnung einer Solltrajektorie

Selbst wenn es sich bei der vorhandenen Design-Scheibenfläche um eine Schraubfläche handelt und eigentlich eine Bewegung entlang einer Trajektorie zu keiner Verformung der Dichtungen führt, wird durch die Wahl von c_B als Nicht-Trajektorie eine Verformung der Dichtungen bewusst in Kauf genommen. Die Vorgabe der Begrenzungskurven und Scheibenfläche unterliegen ästhetischen Gesichtspunkten und vor diesem Hintergrund kann es mitunter zu Designanpassungen kommen, die einer verformungsfreien Absenkung entgegenstehen. Für die Auslegung der Absenkbewegung bedeutet dies, dass die Dichtungsverformungen zumindest gering sein sollten. Es wird daher jene Absenkbewegung M_{SB} gesucht, die aus der Vorgabe einer minimalen Dichtungsverformung resultiert. Diese Aufgabenstellung entspricht der zweiten Stufe der Vorgehensweise aus Abb. 5.6.

Der hier vorgeschlagene Lösungsweg basiert auf der zuvor beschriebenen Methode zur Approximation von Schraubflächen. Es wird wiederum davon ausgegangen, dass es sich bei der optimalen Absenkbewegung M_{SB} um eine Schraubbewegung handelt. Allerdings steht nun neben der Scheibenfläche S zusätzlich auch die Information der Begrenzungskurve c_B zur Verfügung.

Damit es entlang von c_B zu keinen Auslenkungen der Dichtungen kommt, müsste c_B idealerweise in sich selbst verfahren, also die Bahnkurve eines Punktes $P \in S$ bei der gesuchten Schraubbewegung M_B sein. Diese Schraubbewegung M_B lässt sich anhand der Normalen $g_{B,jk}$ entlang von c_B ermitteln, Abb. 5.15 (rechts). Demnach ist jeder Punkt $P_{B,j}$, $j = 1, \dots, j_e$ auf c_B Ausgangspunkt für die Geraden $g_{B,jk}$, $k = 1, \dots, k_e$ eines Strahlenbüschels \mathcal{L}_j . Die Geraden $g_{B,jk}$ dieses Strahlenbüschels stehen normal auf c_B , sodass mit einer Überführung als Punkte $P_{B,j}$ in einen \mathbb{R}^6 und einer Optimierung wiederum eine angenäherte Schraubung M_B ermittelt werden kann. Hierzu kann der Algorithmus aus Abschnitt 5.5.4 unverändert übernommen werden.

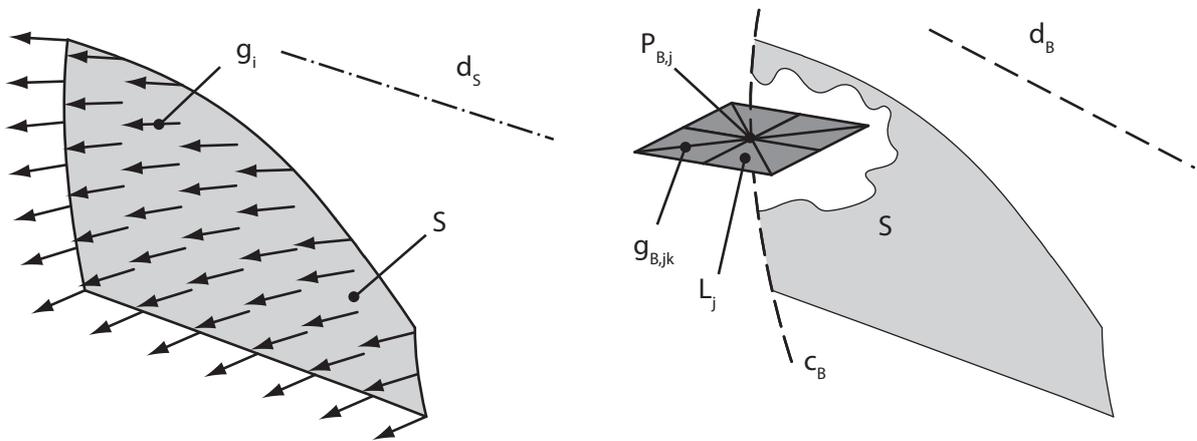


Abbildung 5.15: Links: Raster von Flächennormalen g_i und optimierte Schraubachse d_S für die Scheibenfläche S ; rechts: Normalen $g_{B,jk}$ auf die Kurve c_B und die optimierte Schraubachse d_B für c_B .

Wird M_B für die Absenkung von S verwendet, führt diese Bewegung zu einer minimalen Auslenkung der Dichtungen entlang von c_B . Gleichzeitig kommt es in der Regel jedoch zu unzulässigen Auslenkungen entlang von c_K . Umgekehrt würde die in Abschnitt 5.5.4 ermittelte und rein auf den Informationen der Scheibenfläche S beruhende Absenkbewegung M_S zu geringen Auslenkungen entlang von c_K und zu großen Auslenkungen entlang von c_B führen. In Abb. 5.16 werden beispielhaft für eine Scheibe S die Achsen d_B und d_S der Schraubbewegungen M_B und M_S gezeigt. Mit c_S wird ebenfalls eine Trajektorie von M_S dargestellt.

Um nun sowohl entlang von c_B als auch von c_K keine unzulässige Auslenkung zu erhalten, wird eine Schraubbewegung M_{SB} gesucht, welche die Geometrie von S und von c_B berücksichtigt. Als Lösung dieses Problems wird wiederum auf den Algorithmus aus Abschnitt 5.5.4 zurückgegriffen, allerdings werden für die Optimierung sowohl die Flächennormalen g_i von S als auch die Normalen $g_{B,jk}$ von c_B verwendet. Dadurch werden beide Anforderungen berücksichtigt und es kommt in Summe zu einer geringeren Auslenkung der Dichtungen.

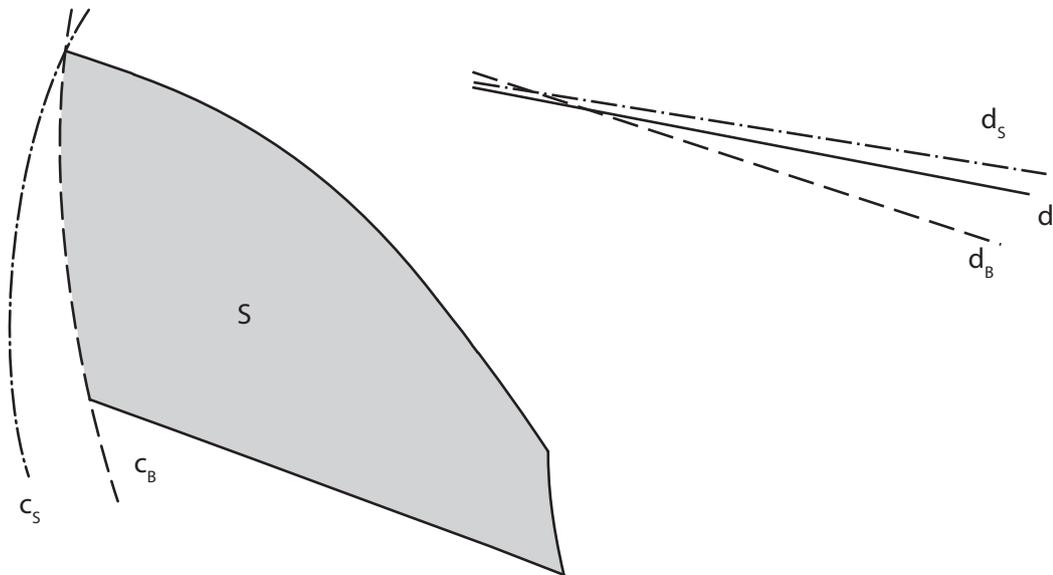


Abbildung 5.16: Scheibenfläche S und die dazugehörigen Schraubachsen für die Absenkbewegung d_S , d und d_B .

Die Gewichtung der beiden Vorgaben kann entweder durch Gewichtungsfaktoren oder durch die Anzahl an g_i und $g_{B,jk}$ erfolgen. Letzteres wird durch *Gfrerrer* in einem Anwendungsfall beschrieben und liefert für $i = 1, \dots, 100$, $j = 1, \dots, 50$ und $k = 1, 2$ – also bei einer gleich großen Gewichtung für die Scheibenfläche S und die Begrenzungskurve c_B – zufriedenstellende Ergebnisse [GLH⁺11]. Des Weiteren besteht die Möglichkeit, mit M_{SB} und einer erzeugenden Kurve, wie etwa c_K oder c_T , eine neue Scheibenfläche S zu generieren. Diese neue Fläche liefert zugleich einen Designvorschlag, sollten die Auslenkungen in den Dichtungen zu groß sein.

5.5.6 Bewertung des Lösungswegs

Der vorgestellte Lösungsweg beschreibt ein allgemeingültiges Verfahren zur Ermittlung helixförmiger Solltrajektorien von Seitenscheiben. Wesentliche Merkmale dieses Ansatzes bestehen in der geringen Anzahl an Eingaben – im geringsten Fall reichen Normalvektoren an fünf beliebigen Punkten der Seitenscheibe – und in der Möglichkeit, sowohl abgetastete Daten von Designmodellen als auch CAS-Modelle zu verarbeiten. Außerdem kann die Begrenzungskurve entlang der B-Säule c_B als zusätzliche Eingabe berücksichtigt werden. Darüber hinaus zeichnet sich der zu Grunde liegende Berechnungsalgorithmus durch eine kurze Berechnungsdauer aus, die üblicherweise in der Größenordnung von wenigen Sekunden liegt. Dadurch ist dieser Ansatz iterativen, manuellen Suchverfahren überlegen, die in der Regel eine längere Bearbeitungszeit benötigen und mitunter eine geringere Ergebnisgüte aufweisen.

Da der Ansatz allgemein gewählt ist und keine Vereinfachungen getroffen werden, kann in diesem Fall auf eine Überprüfung der Ergebnisse durch weitere Berechnungswerkzeuge verzichtet werden. Dadurch kann die Funktionsauslegung vollständig mit Hilfe eines *Funktionstemplates* durchgeführt werden. Dies sorgt für einen frühzeitigen Wissenszuwachs, der aufgrund der hohen

Ergebnisgüte als fundierte Grundlage für die Abstimmung zwischen den ästhetischen und technischen Gesichtspunkten im Rahmen einer *Design-Technik-Konvergenz* dient. Darüber hinaus kann der Lösungsweg die Bewertung von Scheibenflächen und die Ermittlung einer optimalen Absenkbewegung auch über eine Konzeptphase hinaus unterstützen.

5.6 Geometriemodellierung eines Türinnenblechs

Entsprechend dem Vorgehensmodell der CAD-basierten Vorauslegung fließen die Ergebnisse der funktionalen Vorauslegung in die Geometriemodellierung ein. In dem hier beschriebenen Anwendungsbeispiel der Fahrzeugtüre liefern die Resultate der Kinematik-Vorauslegung die Eingaben für die Geometriemodellierung des Tür-Rohbaus. Die Auslegungsergebnisse bestehen zum einen aus der Scharnierachse und zum anderen aus der Trajektorie der Seitenscheibe beim Absenken. Erstere beeinflusst unter anderem die Anbindungen für die Scharniere und für das Türschloss sowie die Breite der Türfugen. Die Absenkung der Seitenscheibe bestimmt die Form der Scheibe und damit auch des einrahmenden Rohbaus. Außerdem muss für die abgesenkte Scheibe entsprechender Bauraum innerhalb der Türe vorgesehen werden.

Ein Bauteil innerhalb eines typischen Tür-Rohbaus, welches von beiden Auslegungsergebnissen betroffen ist, stellt das Türinnenblech dar. Es gibt zwar eine Vielzahl an unterschiedlichen Bauweisen und Varianten für den Tür-Rohbau, jedoch kommt häufig eine zweischalige Blechbauweise mit einem Innen- und einem Außenblech zur Anwendung [HMF07]. Daher fokussiert die hier vorliegende Arbeit auch auf diese Rohbau-Architektur und geht detailliert auf die Geometriemodellierung des Türinnenblechs ein. Dieses Bauteil dient, im Gegensatz zum Außenblech, welches als Teil der Außenhaut hauptsächlich den mechanischen Schutz vor äußeren Einflüssen sicherstellt, als zentrale Aufnahme für eine Reihe von Türkomponenten. Hierunter fallen unter anderem Fensterheber, Crashverstärkungen, Lautsprecher, Türschloss oder Türscharniere. Da für das Türinnenblech aufgrund der vielen Schnittstellen zu anderen Bauteilen eine Vielzahl an Anforderungen gelten, hat es in der Regel eine komplexe Form. Daher wird neben dem Tür-Rohbau auch insbesondere das Türinnenblech oftmals als Anwendungsbeispiel für die Umsetzung von neuartigen Methoden in der Karosseriekonstruktion und in der Geometriemodellierung herangezogen.

Demnach verwenden auch *Forsen*, *Haslauer* und *Leinweber* die Fahrzeugtüre, um die Wirksamkeit ihrer Ansätze zu demonstrieren [For03, Has05, Lei09]. Gleichfalls dient es bei *Braß* als beispielhaftes Bauteil, um gängige Methoden in der Geometriemodellierung darzustellen [Bra09]. Aufgrund der Vielzahl an Arbeiten, die auf die Modellierung der Fahrzeugtüre – unter anderem mit Hilfe von *Geometriemplates* – eingehen, ist dieser Anwendungsfall gut erschlossen und der Stand der Technik auf dem Gebiet der Konstruktionsmethodik wird vielfach aufgezeigt. Etliche Ansätze und Methoden, welche in das hier vorgestellte *Geometriemplate* einfließen, werden auch in den Diplomarbeiten von *Mayr* und *Brauchart* zusammengefasst [May09, Bra10].

Das prototypisch umgesetzte *Geometriemplate* des Türinnenblechs kann, wie bereits in Abb. 5.1 dargestellt, in einer zentralen *Template*-Bibliothek archiviert und im Rahmen der CAD-basierten Vorauslegung in eine Baugruppenstruktur einer Fahrzeugtüre eingefügt werden. Die

hier vorgeschlagene Struktur besteht aus einer übergeordneten Baugruppe der Fahrzeugtüre sowie einer Unterbaugruppe des Tür-Rohbaus, wobei jede Gliederungsebene ein Skelettbauteil für die Geometriesteuerung enthält. Sowohl die Gesamtbaugruppe als auch die Skelettbauteile beschränken sich in ihrer Ausführung darauf, dem *Geometrietemplate* des Türinnenblechs entsprechend dem IPO- bzw. EVA-Schemas eine übergeordnete Struktur bereitzustellen. Demnach enthalten die Skelettbauteile ausschließlich Geometrielemente und Parameter, die für die Geometriemodellierung des Türinnenblechs benötigt werden. Jedoch lassen sich sowohl die Baugruppenstruktur als auch die Skelettbauteile erweitern, sodass neben dem Türinnenblech auch andere *Templates* einbezogen werden können. Dadurch lässt sich in weiterer Folge eine komplette Türgeometrie aufbauen.

Das Türinnenblech-*Geometrietemplate* ist ebenfalls stringent dem IPO- bzw. EVA-Schemas folgend aufgebaut und enthält etliche Subsysteme in Form von Geometriesequenzen. Demnach gleicht der Aufbau des *Templates* jenem in Abb. 3.12. Um eine durchgängige, bauteilübergreifende Geometriesteuerung darzustellen, ist das Innenblech bei der Integration in die Baugruppenstruktur des Tür-Rohbaus mit dem übergeordneten Skelettbauteil zu verknüpfen. Dies kann abhängig vom verwendeten CAD-System auch automatisiert erfolgen. In weiterer Folge liefert das Skelettbauteil alle Eingaben auf Baugruppenebene, mit dem Ziel bauteilübergreifende Informationen zentral bereitzustellen. Hierunter fallen unter anderem die Eingaben des Fahrzeugdesigns, wie die Außenhaut samt Scheibenfläche, oder die Ergebnisse der Kinematik-Vorauslegung. Diese Daten sind mit der Eingabeebene des Türinnenblechs verbunden und bilden dort auf Bauteilebene zusammen mit den Referenz- und Steuerelementen für die bauteilinterne Geometriesteuerung ein *Matingmodell*.

Dieses *Matingmodell* mit seinen Geometrieelementen und Parametern fasst auf Bauteilebene alle Eingaben zusammen, welche für die Steuerung des Geometriemodells erforderlich sind. Da alle Eingaben aus einer übergeordneten Gliederungsebene einfließen, entspricht es einem *Skelettmodell*. Der Modellaufbau der Innenblech-Geometrie erfolgt – wie auch in Abb. 3.12 dargestellt – modular mit Hilfe von Geometriesequenzen, die miteinander verknüpft sind. Abb. 5.17 zeigt die Abfolge der Geometriemodellierung beginnend mit dem Skelett. Demnach werden zuerst die Abstellflächen mit dem Skelett verbunden und darauf aufbauend die Innenflächen festgelegt. Abschließend geben Verprägungen die resultierende Form des Türinnenblechs. Für jeden dieser Schritte steht eine Reihe unterschiedlicher Geometriesequenzen zur Verfügung, sodass sich dadurch eine Vielzahl an unterschiedlichen Bauteilvarianten erstellen lässt. Dies wird auch in Abb. 5.17 aufgezeigt, indem zu jeder Detaillierungsstufe der hier vorgestellten Umsetzung die Anzahl an möglichen Varianten angeführt wird. Demnach lassen sich durch unterschiedliche Abstellflächen bereits 16 Geometrievarianten erzeugen. So können etwa ein- oder zweistufige Schwellerflächen oder unterschiedliche Schlossflächen modelliert werden. Durch weitere Geometriesequenzen für Innenflächen und Verprägungen wächst die Gesamtzahl an möglichen Geometriemodellen exponentiell auf 320 unterschiedliche Varianten an. Hervorzuheben ist, dass alle verwendeten Geometriesequenzen Merkmale enthalten, die auf Daten von realen Fahrzeugtüren basieren.

Der modulare Aufbau des Türinnenblechs, bei dem unterschiedliche Geometriesequenzen miteinander verknüpft werden können, unterstreicht die Vorteile einer Geometriemodellierung mit

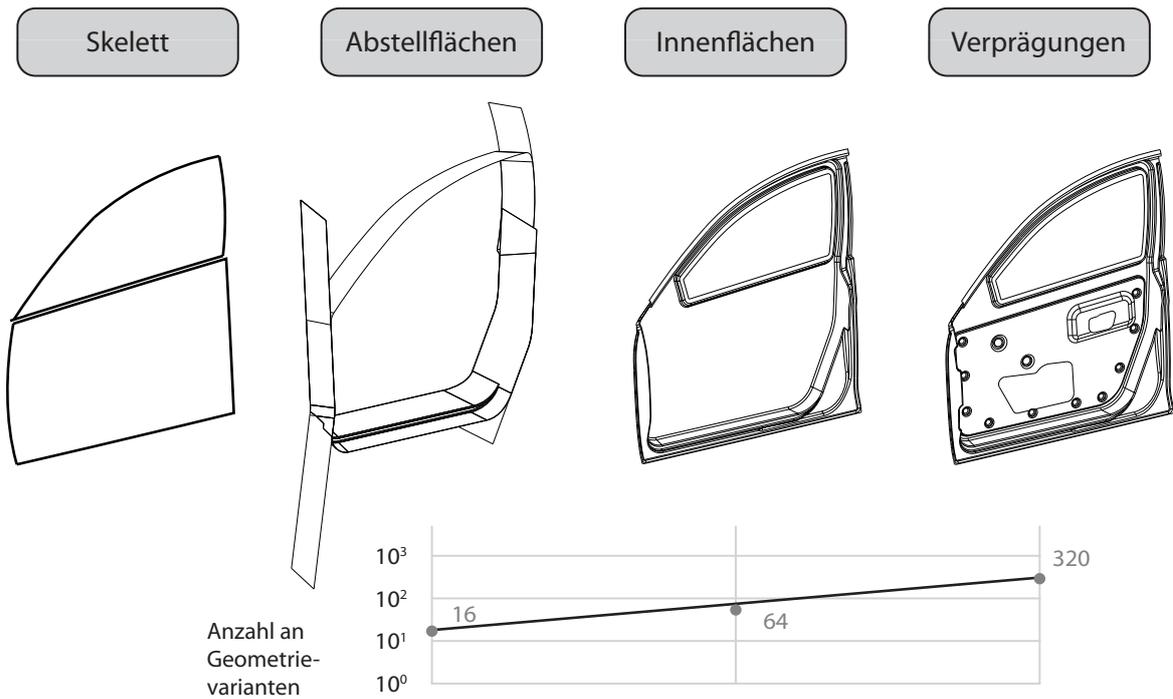


Abbildung 5.17: Abfolge der Geometriemodellierung innerhalb des Türinnenblech-Geometriemplates.

Hilfe von *Matingtemplates*. Infolgedessen können durch die Änderung einer Verknüpfung die Merkmale der Innenblechgeometrie verändert werden. Dadurch lassen sich effizient und benutzerfreundlich unterschiedliche Bauteilgeometrien für Variantenstudien erzeugen. Dies wird gleichfalls durch Abb. 5.18 verdeutlicht, in der beispielhaft zwei unterschiedliche Geometrievarianten dargestellt sind. Die Unterschiede in der Geometrie resultieren aus der Änderung von vier Verknüpfungen innerhalb des Bauteilmodells.

Neben der Möglichkeit der effizienten Erstellung von Varianten zeichnet sich das Türinnenblech-Geometriemplate – wie generell *Matingtemplates* – durch seine durchgängige Geometriesteuerung aus. Diese kann mit Hilfe von Skelettbauteilen auch über die Grenzen von Bauteilen und Baugruppen hinweg erfolgen. Dadurch passt sich die Bauteilgeometrie automatisiert an neue Eingaben, wie etwa einem überarbeiteten Design oder aktuellen Ergebnissen der Kinematik-Vorauslegung, an. So ändert sich beispielsweise der Flächenverbund, welcher die Türschlossanbindung repräsentiert, sobald eine neue Scharnierachse zur Verfügung steht und gewährleistet damit, dass die Anbindungsflächen stets parallel zur Scharnierachse stehen. Das Gleiche gilt auch für ein neues Design, wie dies durch Abb. 5.19 veranschaulicht wird. In diesem Beispiel kommt es nicht nur zu einer geringfügigen Designanpassung, sondern die Designdaten werden komplett ausgetauscht und durch Eingaben von einer unterschiedlichen Fahrzeugklasse ersetzt. Dadurch soll gezeigt werden, dass das umgesetzte Türinnenblech-Geometriemplate robust hinsichtlich neuer Eingaben ist und dadurch flexibel eingesetzt werden kann.

Diese automatisierten Anpassungen der Geometrie verzichten auf langwierige, manuelle Änderungen und erhöhen gleichzeitig die Qualität, da menschliche Einflussfaktoren vermieden wer-

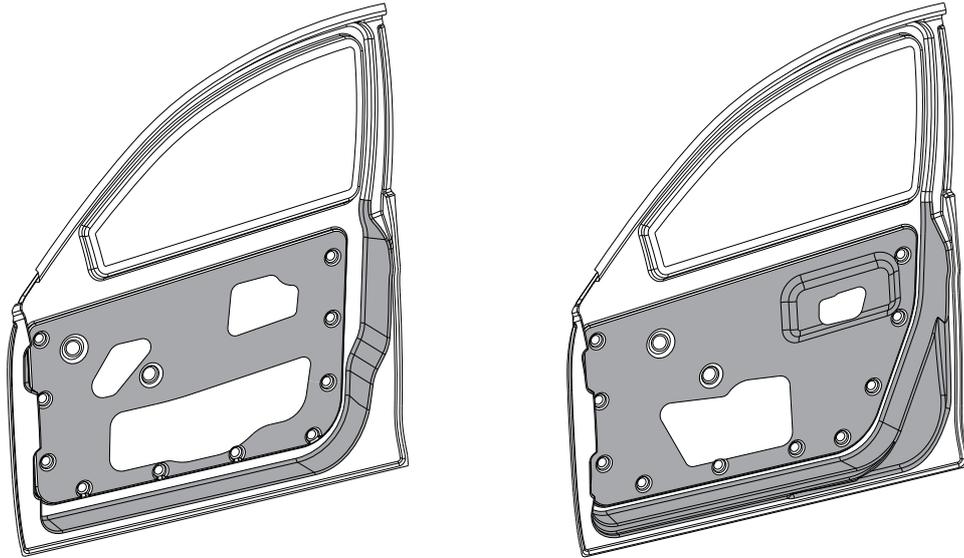


Abbildung 5.18: Zwei mögliche Varianten der Türinnenblechgeometrie.

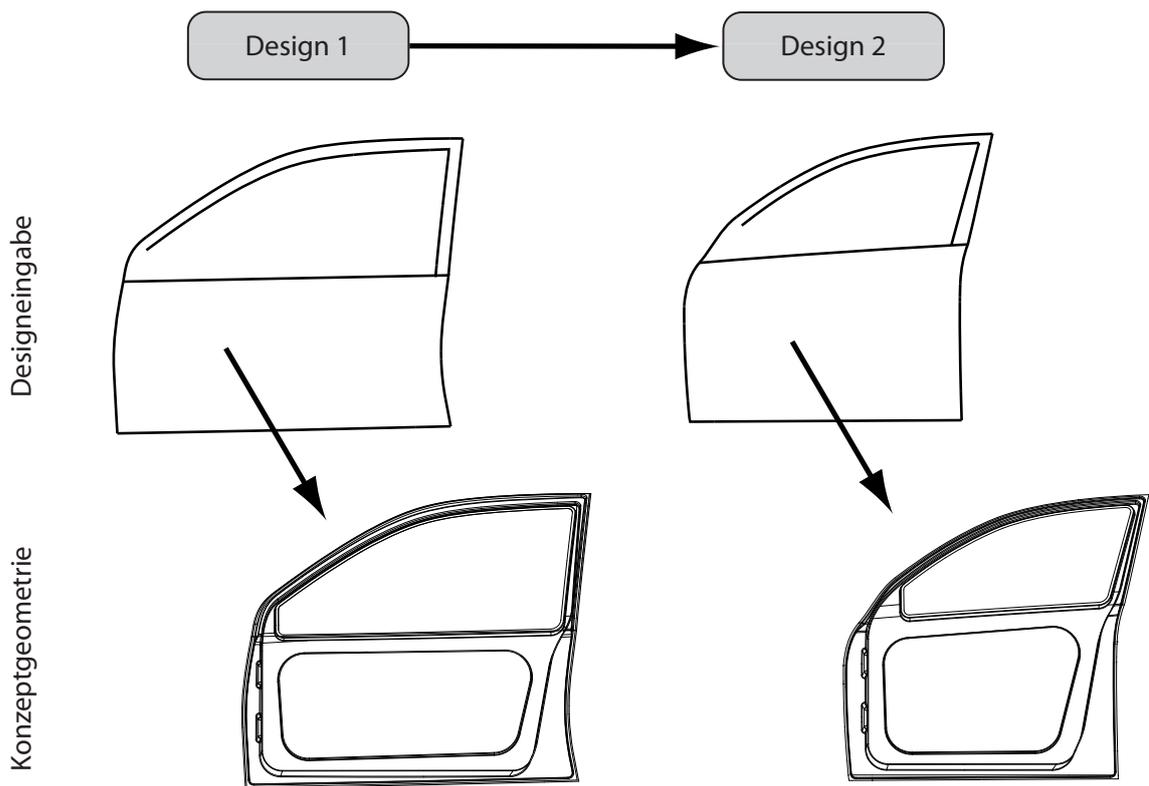


Abbildung 5.19: Anpassung der Geometrie des Türinnenblech-Geometriemplates an ein geändertes Fahrzeugdesign.

den. Da die Eingaben für die Geometriesteuerung einen weiten Bereich an möglichen Werten umfassen, muss sichergestellt werden, dass die *Geometrietemplates* innerhalb dieses Wertebereichs gültige Ergebnisse liefern. Dies gilt insbesondere auch für die Designflächen, welche in der Regel originär und daher einzigartig sind. Daher ist eine wesentliche Anforderung an die Entwicklung von *Geometrietemplates*, dass diese robust im Bezug auf sich ändernde Eingaben sind. Dies erfordert, wie auch schon in Kap. 3.2 beschrieben, eine intensive Testphase vor dem Einsatz und eine ständige Betreuung, bei der mögliche Fehler behoben und Verbesserungen umgesetzt werden.

Zusammenfassend lassen sich mit Hilfe der umgesetzten *Templates* die Vorteile einer CAD-basierten Vorauslegung zeigen. Demnach liefern die *Matingtemplates* auf Ebene von Bauteil- und Geometriesequenzen eine durchgängige Bauteilmodellierung, die sich einerseits auf ein *Skelettmodell* stützt und andererseits die Verknüpfung von einzelnen Geometriesequenzen verwendet. Dadurch lassen sich beide Anforderungen erfüllen: eine rasche Anpassung an aktuelle Eingaben und die Möglichkeit, benutzerfreundlich eine Vielzahl an Bauteilvarianten zu erstellen. Folglich liefert die hier umgesetzte Bauteilmodellierung eine konsequente Fortführung der Auslegungen mit Hilfe der *Funktionstemplates* und führt zu qualitativ hochwertigen Bauteilmodellen. Diese stehen den anschließenden Prozessen, wie etwa der Berechnung, rasch zur Verfügung und liefern die Grundlage für einen optimierten Entwicklungsprozess.

5.7 Zusammenfassung

Die kinematischen Aufgabenstellungen bei der Auslegung einer Standard-Fahrzeugtüre und der daraus resultierende Einfluss auf die Türgeometrie liefern das erste Anwendungsbeispiel für die CAD-basierte Vorauslegung. Demnach sind die Ergebnisse der funktionalen Vorauslegung in Form der Scharnierachse und der Absenkbewegung der Fahrzeugscheibe gleichzeitig auch wichtige Eingaben für die Geometriemodellierung der Türe und beeinflussen beispielsweise Türfugen, Anbindungsflächen für Schloss und Scharniere oder die Form der Seitenscheibe und damit des einrahmenden Rohbaus.

Die Aufgabenstellung unterstreicht aufgrund der hier auftretenden engen Verflechtungen zwischen Fahrzeugdesign, Konstruktion und Berechnung die Vorteile einer CAD-basierten Vorauslegung. So können mit den Auslegungsergebnissen innerhalb der CAD-Umgebung Designvorschläge und Entwicklungsvorgaben bewertet werden, genauso wie die gewonnenen Erkenntnisse als Vorgaben für Geometriemodelle dienen. Dadurch lassen sich mit Hilfe von *Templates* effizient Konzeptgeometrien für Folgeprozesse, wie beispielsweise eine Berechnungsschleife, erstellen.

6 Bauraumauslegung einer Stoßfängerstruktur

6.1 Einleitung

Die Bauraumauslegung im Bereich des Fahrzeugvorderwagens wird von einer Reihe divergierender Anforderungen bestimmt. Zum einen gibt das Fahrzeugdesign die Außenflächen vor, zum anderen müssen funktionale und gesetzliche Vorgaben erfüllt werden. So gilt es beispielsweise Sichtfelder, den Fußgängerschutz oder die Motorraumdurchströmung zu beachten. Des Weiteren wird versucht, den vorhandenen Platz im Motorraum optimal auszunutzen, um diesen möglichst kompakt ausführen zu können. Dadurch kann bei gleichen Fahrzeugabmessungen der Motorraum verkleinert und mehr kundenrelevanter Bauraum für die Insassen genutzt werden. Eine Herausforderung liegt in der frühzeitigen Festlegung von Bauraumgrenzen, welche oftmals auf Basis von unvollständigen und zumeist unsicheren Informationen erfolgt. Dennoch lassen sich einmal festgelegte Bauraumgrenzen und Bauteilpositionen in weiterer Folge nur schwer anpassen. Gerade unter diesen sich schnell ändernden und unsicheren Bedingungen kann die CAD-basierte Vorauslegung ihre Vorteile ausspielen. Daher wird der Ansatz der CAD-basierten Vorauslegung für die Dimensionierung und Bauraumauslegung des Crash Management Systems (CMS³⁰), als ein wesentlicher Teilbereich des Fahrzeugvorderwagens, angewandt und die Umsetzung in diesem Kapitel beschrieben.

6.2 Entwicklung von Crash Management Systemen in der Konzeptphase

Das CMS, wie es beispielhaft in Abb. 6.2 dargestellt ist, besteht üblicherweise aus dem Stoßfänger und den dazugehörigen Deformationselementen, den sog. Crashboxen, und wird hauptsächlich durch Anforderungen an die passive Sicherheit bestimmt. Trotz der hohen Bedeutung für die Fahrzeugsicherheit wird diese Baugruppe in der Produktstruktur, wie auch in der Projektorganisation einer Gesamtfahrzeugentwicklung, oftmals den Anbauteilen und nicht der Karosseriestruktur zugerechnet. Die Anbau- oder Exterieurbauteile bestehen zumeist aus Außenhautteilen, wie beispielsweise Kotflügel oder Schweller, und bilden eine Untergruppe der Karosserie. Ein Grund für die Eingliederung des CMS in die Exterieurumfänge liegt in der späteren Vergabe der Fertigung in Form von modularen Vorderwageneinheiten. Diese sog. Frontend-Module können neben dem CMS eine durchaus umfangreiche Anzahl an Komponenten wie Verkleidungen, Scheinwerfer, diverse Kühler oder Luftführungen enthalten. In der Regel

³⁰Crash Management System

werden Frontend-Module in einer Einheit von Lieferanten bezogen und im Zuge der Montage am Fahrzeug befestigt. Dadurch ergibt sich eine fertigungsbedingte Schnittstelle, die sich auch in der CAD-Produktstruktur widerspiegelt.

Erste Auslegungen der passiven Sicherheit eines Fahrzeuges werden genauso wie Packagingstudien bereits in der Definitionsphase durchgeführt. Allerdings liegt der Fokus in dieser Phase vor allem auf Crashtests mit hohen Aufprallgeschwindigkeiten, die eine Grenze von 15 km/h deutlich übersteigen. Versuchsanordnungen mit geringeren Geschwindigkeiten, die insbesondere für die Auslegung der Deformationselemente des CMS und der vertikalen Position des Stoßfängers relevant sind, werden zumeist nur in Form von geometrischen Randbedingungen berücksichtigt. Gerade diese Randbedingungen werden durch Entwurfswerkzeuge für die Gesamtfahrzeugentwicklung abgedeckt, sodass beim Übergang in die Konzeptphase bereits ein vorläufiges CMS vorliegt [Pot08].

In der Konzeptphase gewinnen die bisher vernachlässigte Anforderungen zunehmend an Bedeutung. Im Fall des CMS zählen zu diesen Anforderungen vor allem die Vorgaben aus dem Fußgängerschutz und aus Crashversuchen mit geringer Aufprallgeschwindigkeit sog. *Low-Speed Crash Tests*. So soll gerade bei letzteren die Aufprallenergie hauptsächlich im CMS abgebaut werden, um die angrenzende Karosserie sowie umliegende Anbauteile, wie etwa Kühleinheiten, nicht zu beschädigen. Dadurch können allfällige Reparaturkosten minimiert werden. Für die Bauraumauslegung des Motorraums ist demnach die Kenntnis des CMS-Deformationsverhaltens von großer Bedeutung. Erschwerend bei der Auslegung des CMS kommt hinzu, dass der Bauraum durch die umliegende Außenhaut und damit durch das Fahrzeugdesign beeinflusst wird. Darüber hinaus bietet die Konzeptphase noch relativ viele Freiheiten, sodass oftmalige Anpassungen von Design und Bauraumgrenzen den Konstruktionsprozess bestimmen. Diese Änderungen erfordern in weiterer Folge eine Vielzahl an Berechnungsschleifen, da die Funktionsauslegung und -absicherung des CMS mit Hilfe von Crashberechnungen erfolgen. Daher bietet sich ein Einsatz der CAD-basierten Vorauslegung an, mit der geänderte Anforderungen frühzeitig bewertet und mitunter Berechnungsschleifen eingespart werden können.

6.3 Anwendung des Vorgehensmodells

Die Anwendung der CAD-basierten Vorauslegung ermöglicht innerhalb der CAD-Umgebung qualitative Aussagen über den Einfluss von Geometrie- und Materialänderungen auf das Crashverhalten bei *Low-Speed Crash Tests*. Ein wichtiger Bestandteil umfasst dabei die Darstellung von Bauteilverformungen, welche insbesondere bei der Festlegung von Bauraumgrenzen für umliegende Komponenten hilfreich sind. Die Ergebnisse dieser funktionalen Auslegungen werden –dem Ansatz der CAD-basierten Vorauslegung folgend– mit Hilfe von *Geometriemplates* in eine resultierende Konzeptgeometrie überführt. Dadurch stehen Folgeprozessen, wie etwa einer Absicherung der Vorauslegung mittels Simulationswerkzeugen oder weiteren Optimierungen der Crashstruktur, hochwertige Eingangsdaten zur Verfügung.

Abb. 6.1 zeigt schematisch die Anwendung des Vorgehensmodells anhand einer vorgegebenen Produktstruktur eines Frontend-Moduls. Die Produktstruktur enthält ein übergeordnetes Ske-

lett und diverse Unterbaugruppen, wobei eine der Unterbaugruppen die CMS-Umfänge beinhaltet. Im Laufe der Konzeptentwicklung kommt es zur Generierung von Auslegungsergebnissen und einem sukzessiven Befüllen der Produktstruktur. Diese Abläufe sind ebenfalls in Abb. 6.1 eingezeichnet und orientieren sich an den Phasen des Vorgehensmodells der CAD-basierten Vorauslegung, die ausführlich in Kap. 4.3.2 beschrieben werden. Des Weiteren sind als Werkzeuge einer wissensbasierten Konstruktion in Abb. 6.1 auch jene *Templates* dargestellt, auf die im Rahmen der CAD-basierten Vorauslegung zurückgegriffen wird. Hierunter fallen ein *Funktionstemplate*, mit dem die Dimensionierung der CMS-Komponenten aufgrund von *Low-Speed Crash*-Anforderungen erfolgt, und diverse *Geometrietemplates*, mit denen eine Konzeptgeometrie des CMS effizient erzeugt werden kann.

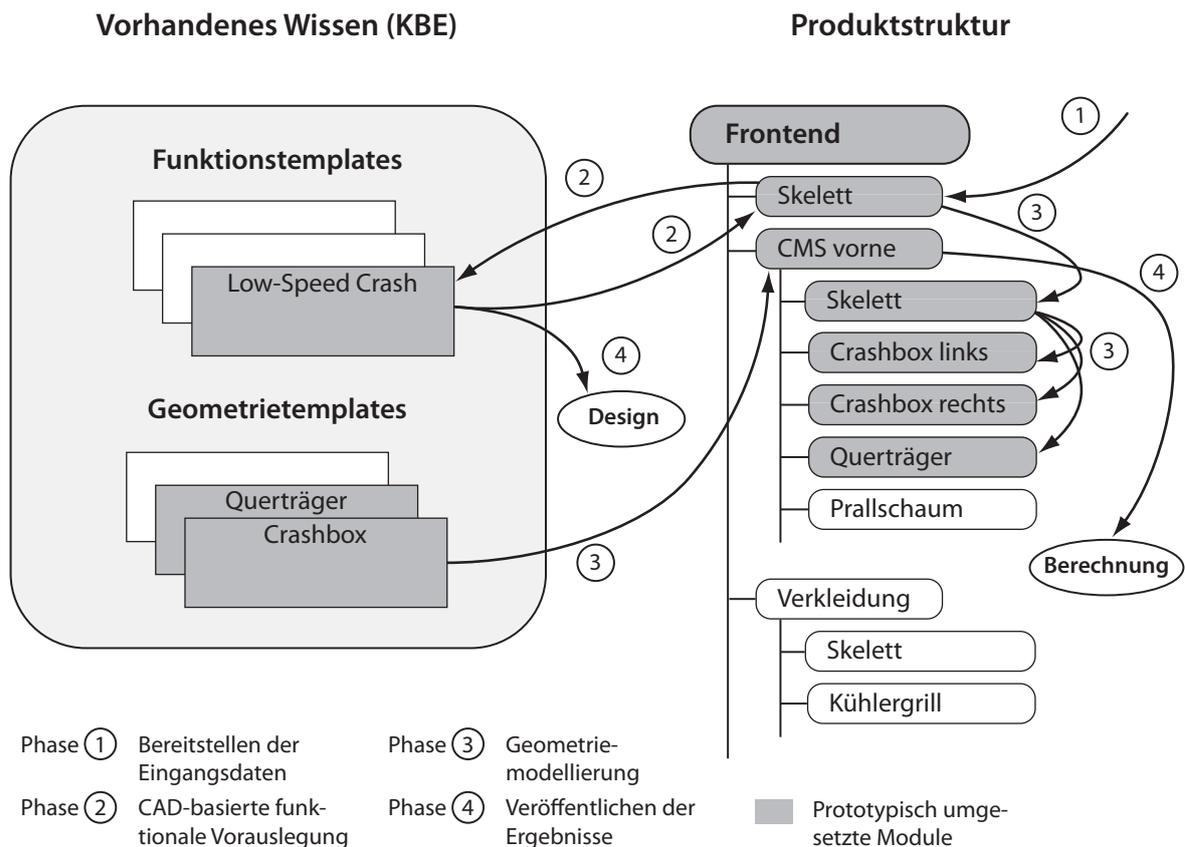


Abbildung 6.1: Prototypische Anwendung der CAD-basierten Vorauslegung mit den einzelnen Phasen: links das in Templates bereitgestellte Wissen, rechts die Produktstruktur eines Frontend-Moduls.

In der hier vorgestellten prototypischen Umsetzung der CAD-basierten Vorauslegung können die Vorteile dieses Ansatzes ausgenutzt werden. Demnach wird die Konstruktion unmittelbar in der Auslegung unterstützt, ohne dass auf externe Berechnungsergebnisse gewartet werden muss. Gleichzeitig liefern die *Geometrietemplates* ausgereifte Geometriedaten, die sowohl eine hohe Detailtiefe als auch einen hohen Grad an Flexibilität aufweisen. Außerdem können die Ergebnisse der Vorauslegung rasch dem parallel zur Konstruktion ablaufenden Entwicklungsprozess zugeführt werden. Ein wesentlicher Bestandteil für die erfolgreiche Umsetzung dieses Anwen-

dungsbeispiels liegt in den verwendeten Lösungsansätzen für die funktionale Auslegung. Auf diese wird im folgenden Abschnitt detailliert eingegangen. Darüber hinaus wird abschließend ein Überblick über die Geometriemodellierung gegeben.

6.4 Dimensionierung eines CMS

6.4.1 Anforderungen

Um das breite Spektrum an möglichen Kollisionsszenarien abdecken zu können, müssen die passiven Sicherheitssysteme eines Fahrzeuges aufeinander abgestimmt sein. Potentielle Kollisionsgegner sind die verschiedenen Verkehrsteilnehmer, wie Fahrzeuge oder Fußgänger, genauso wie stehende Hindernisse. Daraus ergeben sich zum einen unterschiedliche Aufprallenergien und zum anderen stellen insbesondere weiche Unfallgegner, wie Radfahrer oder Fußgänger, erhöhte Anforderungen an den Partnerschutz. Aus diesen Gründen bedarf es im Fall von frontalen Aufprallunfällen eines stufenweisen passiven Sicherheitskonzeptes. Demnach wird die Aufprallenergie hauptsächlich in plastische Verformung der passiven Sicherheitskomponenten umgewandelt, wobei das Kraftniveau abhängig von der Eindringtiefe des Unfallgegners stufenweise ansteigt. Das maximale Kraftniveau wird üblicherweise bei der Verformung des Motorlängsträgers erreicht, während die geringeren Stufen im Kraftverlauf aus den Verformung der CMS-Komponenten resultieren [Krö02].

In Abb. 6.2 wird der stufenweise Anstieg des Kraftniveaus bei der Verformung des CMS und des Motorlängsträgers skizziert. Ebenfalls in einem Schnitt dargestellt sind einerseits die Bauteile eines typischen CMS, bestehend aus dem Prallschaum, dem Stoßfänger und den Crashboxen, und andererseits der Motorlängsträger. Kommt es zu einer frontalen Kollision, so komprimiert der ins Fahrzeug eindringende Körper neben den Designabdeckungen zuerst den Prallschaum. Dieser besitzt ein geringes Kraftniveau und soll in Kombination mit der Motorhaube für den Schutz bei Unfällen mit Fußgängern sorgen. Sobald der Prallschaum vollständig komprimiert ist und seine Blocklänge erreicht hat, kommt es zu einem sprunghaften Anstieg der Kontaktkraft. Diese steigt an, bis sich die zweite Stufe des passiven Sicherheitssystems, nämlich die Crashboxen, deformieren. Durch die Crashboxen soll die Aufprallenergie bei Unfällen mit geringen Geschwindigkeiten abgebaut werden, ohne dass der dahinter liegende Motorlängsträger beschädigt wird. Daher liegt auch das Kraftniveau niedriger und erst wenn die Crashbox vollständig gestaucht ist, erfolgt der weitere Energieabbau durch den Motorlängsträger.

Da die Hauptfunktion der CMS-Komponenten in der Energieaufnahme bei Unfällen mit geringen Geschwindigkeiten und im Partnerschutz bei weichen Unfallgegnern liegt, sind ebenjene Unfallszenarien für die Dimensionierung ausschlaggebend. Diese Unfallszenarien werden in einer Vielzahl an standardisierten Crashversuchen, die im Gegensatz zu realen Unfällen unter Laborbedingungen reproduzierbar und dadurch vergleichbar sind, abgebildet. Neben gesetzlichen Vorgaben, in Form von Pendeltests [UNECE80, USDOT90] oder Überprüfungen des Fußgängerschutzes [Amt03], sind Versuchsanordnungen von Versicherungsgesellschaften für die Auslegung von Bedeutung [RCAR11, RCAR10]. Diese Versicherungs-Crashtests wurden entwickelt,

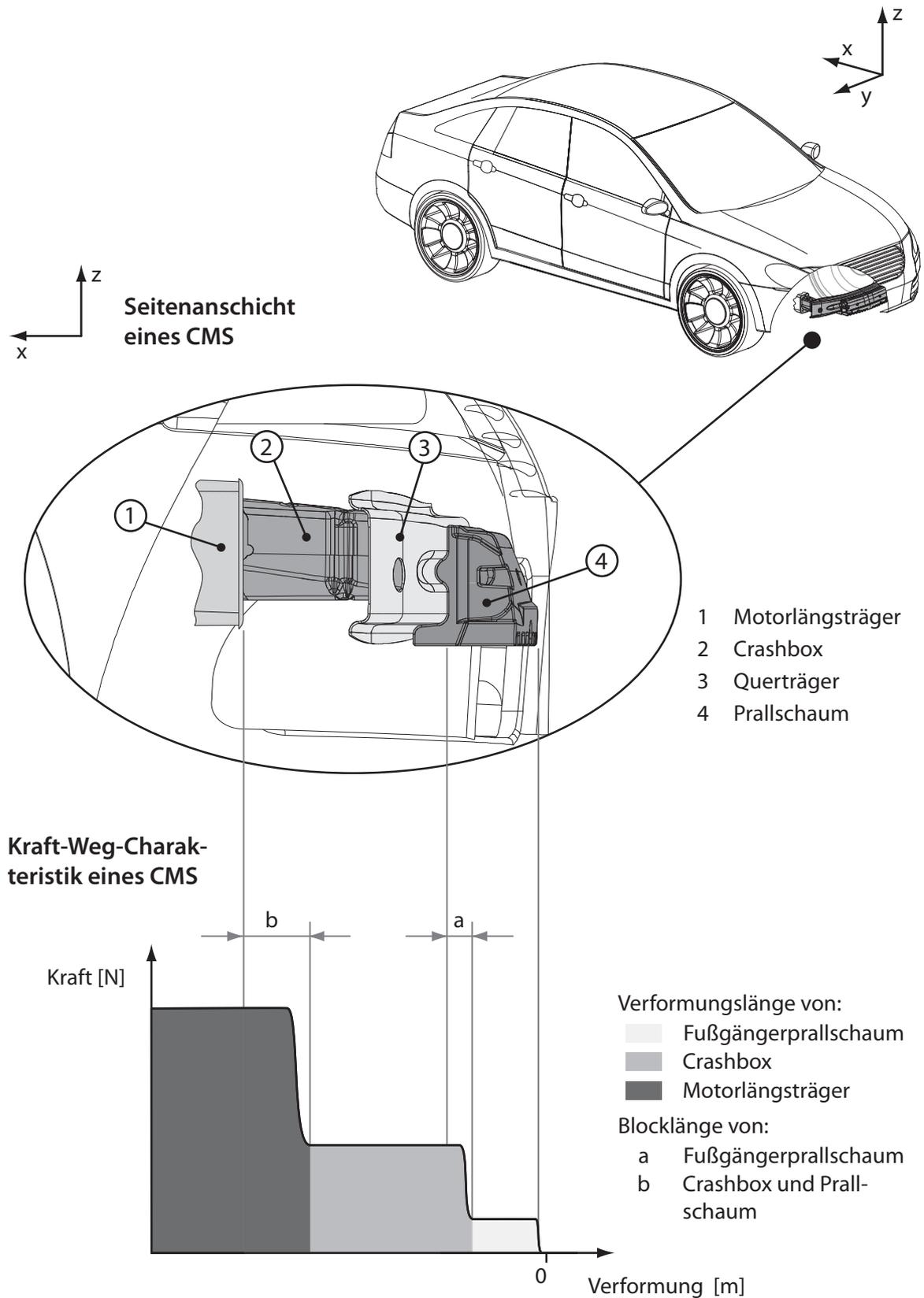


Abbildung 6.2: Komponenten eines CMS (oben) sowie die dazugehörige Kraft-Weg-Charakteristik bei einer Deformation (unten).

um Reparaturkosten abzuschätzen und stellen daher ein Kriterium für die Versicherungseinstufung von Fahrzeugen dar. Insbesondere die Abmessungen der Crashboxen sowie die Position des Stoßfängers werden durch diese Crashtests beeinflusst und sind für die Bauraumauslegung in der Konzeptphase entscheidend. Daher werden auch vorrangig die Versicherungs-Crashtests für die Vorauslegung der CMS-Komponenten herangezogen [HSHL10].

Derzeit haben vor allem zwei dieser Versuchsanordnungen eine hohe Verbreitung und werden daher für die Vorauslegung verwendet. Beide Versuche werden durch die internationale Arbeitsgruppe RCAR³¹ festgelegt und ständig aktualisiert. Bei der ersten Versuchsanordnung handelt es sich um den *Structure Test* [RCAR11]. Bei diesem Test fährt das Versuchsfahrzeug mit 15 km/h gegen ein starres Hindernis, Abb. 6.3. Das Hindernis überdeckt das Fahrzeug zu 40 %, ist 10° normal zur Fahrtrichtung angestellt und reicht vom Boden bis über die Motorhaube. Als Bewertungskriterium werden die Reparaturkosten herangezogen. Dieser Crashtest ist insbesondere für die Dimensionierung der Crashboxen von Interesse, da der Hauptanteil der Kollisionsenergie bei diesem Versuch ebendort abgebaut wird. Daraus ergibt sich unter anderem die Länge der Crashbox und die Eindringtiefe des Hindernisses bestimmt die Anordnung von kostenintensiven Bauteilen, wie Kühlern.

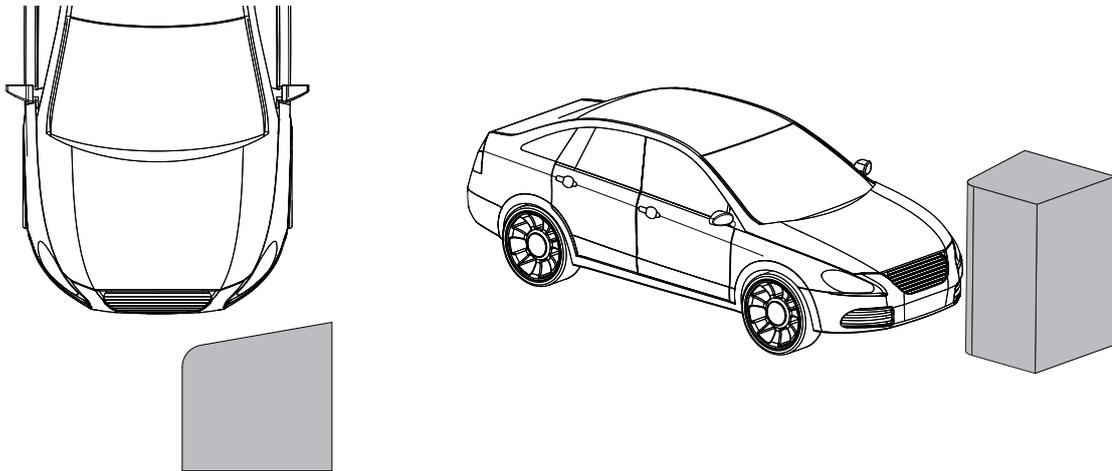


Abbildung 6.3: Darstellung eines RCAR-*Structure Tests*

Die zweite Versuchsanordnung legt ihren Fokus auf die Auslegung des Stoßfängers. Beim *Bumper Test* fährt das Versuchsfahrzeug gegen eine Barriere, die dem Stoßfänger eines Kollisionsgegners nachempfunden ist [RCAR10]. So kollidiert das Versuchsfahrzeug mit 10 km/h mit einer Barriere, die zum Fahrzeug zwar eine horizontale Überdeckung von 100 % aufweist, allerdings nur eine Höhe von 100 mm besitzt und in einem definierten Abstand zum Boden befestigt ist, Abb. 6.4. Außerdem besteht die Barriere aus einer deformierbaren Kunststoffschicht, die den Prallschaum eines Fahrzeuges nachbildet, und einer starren Aufnahme, welche die Fahrzeugstruktur repräsentiert. Neben den Reparaturkosten wird beim *Bumper Test* ebenfalls die vertikale Überdeckung zwischen dem Fahrzeugstoßfänger und der Barriere überprüft. Dadurch

³¹Research Council for Automobile Repairs

wird sichergestellt, dass die Stoßfänger unterschiedlicher Fahrzeuge in der gleichen Höhe angebracht sind und sich bei Kollisionen mit einer Mindestüberdeckung berühren.

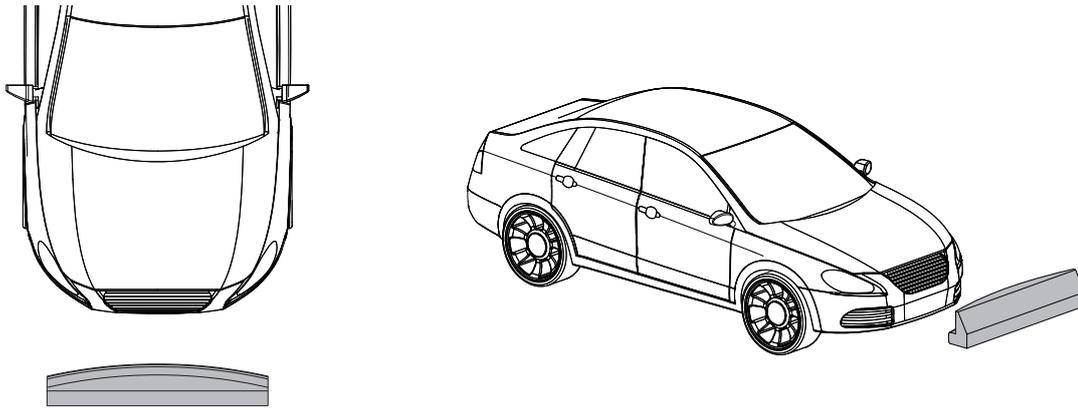


Abbildung 6.4: Darstellung eines RCAR-Bumper Tests

6.4.2 Umgesetzter Lösungsansatz

Mit dem *Structure Test* sowie dem *Bumper Test* liegen ausreichend Lastfälle vor, um in der Konzeptphase die CMS-Komponenten auszulegen und ihre Bauraumgrenzen festzulegen. Beide Crasheszenarien werden daher, wie auch in Abb. 6.1 dargestellt, in einem *Funktionstemplate* integriert. Mit Hilfe dieses *Templates* können qualitative Aussagen zur Form des Crashbox- und des Stoßfängerquerschnitts genauso wie Abschätzungen zur Eindringtiefe der jeweiligen Barriere getroffen werden.

Um diese Resultate zu erhalten, ist im *Funktionstemplate* ein Berechnungsmodell implementiert, das sowohl auf analytische als auch auf *Finite-Elemente*-basierte Methoden zurückgreift. Darüber hinaus stützt sich der umgesetzte Ansatz auf Parameterwerte, die zum Teil in der Literatur angegeben sind oder auf empirische Daten, wie etwa Erfahrungswerte aus der industriellen Anwendung. Die Kombination dieser Methoden liefert ein Berechnungsmodell, das einerseits speziell auf die Anforderungen der *Low-Speed Crash Tests* bei der CMS-Auslegung abzielt und andererseits die Randbedingungen der Konzeptentwicklung berücksichtigt. Demnach können, im Gegensatz zu konventionellen Crashesimulationen, mit wenigen Eingabeparametern und nach einer kurzen Berechnungsdauer bereits aussagekräftige, konzeptionelle Ergebnisse erzeugt werden.

Der eingeschlagene Lösungsweg orientiert sich an *Kröger*, der sich mit der methodischen Auslegung von CMS-Komponenten beschäftigt [Krö02]. Neben der Untersuchung von unterschiedlichen Mechanismen für die Energieaufnahme von Crashboxen stellt er außerdem ein Modell vor, mit dem die Energieaufnahme und die Verformung eines CMS berechnet werden kann. Dieses Modell verzichtet auf eine FEM-Berechnung und konzentriert sich auf analytische Methoden. In der hier vorliegenden Arbeit wird der Ansatz von *Kröger* erweitert und der Stoßfänger mit Balkenelementen diskretisiert. Diese feinere Diskretisierung wird notwendig, um die Auswirkungen des *Bumper Tests*, den *Kröger* nicht betrachtet, darstellen zu können. Dennoch

unterscheiden sich beide Ansätze im Vergleich zu Arbeiten, die sich der Berechnung und Optimierung von Crashstrukturen in einer frühen Entwicklungsphase widmen, da sie den Fokus klar auf *Low-Speed Crash Tests* legen und andererseits ohne FEM-Programmpakete auskommen. So verwenden sowohl *Lasek* als auch *Hilmann* kommerziell erhältliche Simulationsumgebungen, mit denen Untersuchungen des Crashverhaltens für eine Konzeptentwicklung durchgeführt werden können [Hil09, Las06]. *Lasek* konzentriert sich in seiner Arbeit auf Versuche mit hohen Aufprallgeschwindigkeiten und verbindet die konzeptionelle Crashauslegung mit einem Entwurfswerkzeug für die Auslegung von Gesamtfahrzeugen. *Hilmann* beschäftigt sich vor allem mit der Optimierung der Crashstruktur und nimmt unter anderem den *Structure Test* als Anwendungsfall.

Das in der hier vorliegenden Arbeit eingesetzte Berechnungsmodell und die zu Grunde liegenden Annahmen werden im folgenden Abschnitt detailliert beschrieben. Die Plausibilisierung der Berechnungsergebnisse und die Überprüfung der Ergebnisgüte erfolgten sowohl durch einen Vergleich zu FEM-Simulationen als auch durch die Rückmeldungen im Rahmen des industriellen Einsatzes dieses Werkzeuges. In beiden Fällen konnte eine gute Übereinstimmung erzielt werden. Der Vergleich zwischen Berechnungsmodell und FEM-Simulation wird in der Diplomarbeit von *Frener* zusammengefasst und ein Teil der Ergebnisse in Abschnitt 6.4.3 behandelt [Fre12]. Abschließend ist die Mitarbeit an der Erstellung des Modells im Rahmen von studentischen Arbeiten hervorzuheben [FHH11].

6.4.3 Beschreibung des Berechnungsmodells

Da bei den vorliegenden Lastfällen die Verformung des CMS vornehmlich parallel zur Fahrbahnebene auftritt, wird ein zweidimensionales Modell verwendet. Dieses Berechnungsmodell besteht aus miteinander verbundenen Komponentenmodellen der Crashboxen, des Querträgers, des Prallschaums und der Barriere. Abb. 6.5 zeigt schematisch den Aufbau des Gesamtsystems. Demnach werden die Crashboxen durch Stauchrohrelemente abgebildet, die zum einen starr eingespannt und zum anderen mit dem Querträger verbunden sind. Der Querträger selbst ist als FEM-Struktur, bestehend aus $i_e - 1$ Balkenelementen, aufgebaut. Die Balkenelemente sind mit Fließgelenken, die ein ideal-plastisches Werkstoffverhalten besitzen, miteinander verbunden. An jedem der $i = 1, \dots, i_e$ Knotenpunkte $P_{P,i}$ greift ein nicht-lineares Federelement an, das den Prallschaum repräsentiert. Die Barriere ist als starrer Körper modelliert, der durch $j = 1, \dots, j_e$ Punkte $P_{W,j}$ diskretisiert wird.

Sollte, wie beim *Bumper Test*, die Barriere einen verformbaren Teil besitzen, wird eine kombinierte Federkennlinie aus Prallschaum und Barriere ermittelt. Diese Kennlinie wird dann für die Federelemente des Prallschaums herangezogen. Außerdem wird bei der Berechnung des *Bumper Tests* von einer vollständigen Überdeckung von Barriere und Querträger in Richtung der z-Achse ausgegangen, da bei dem gewählten zweidimensionalen Ansatz eine teilweise Überdeckung genauso wenig abgebildet wird wie die Nickbewegung des Fahrzeugs.

Das hier beschriebene Berechnungsmodell lässt Rückschlüsse auf das Verhalten eines CMS-Lastpfades zu. Aufgrund der Anforderungen hinsichtlich des Fußgängerschutzes kommen jedoch

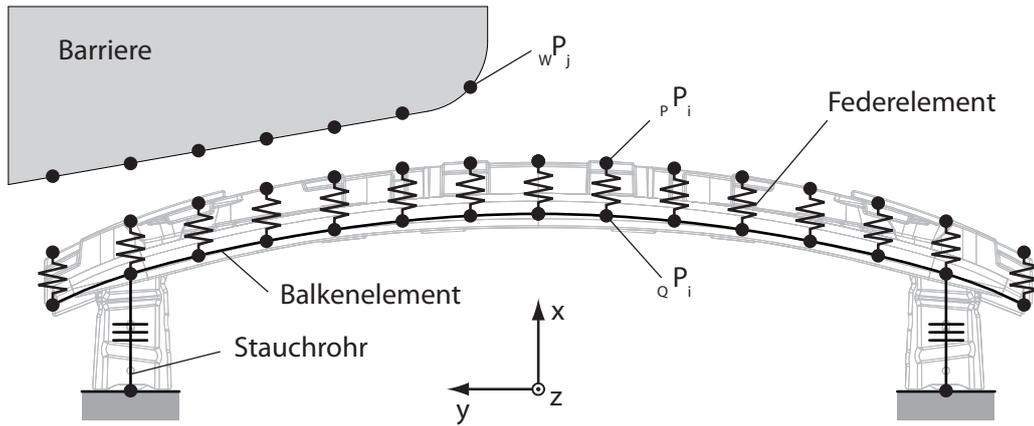


Abbildung 6.5: Schematischer Aufbau des umgesetzten Berechnungsmodells bestehend aus Stauchrohr-, Balken- und Federelementen sowie der Barriere des *Structure Tests*.

zunehmend CMS-Systeme zum Einsatz, die eine zweite, geringer dimensionierte Anordnung von Stoßfänger und Crashboxen zwischen dem Haupt-Stoßfänger und der Fahrbahn aufweisen. Dadurch sollen die Unfallfolgen für Fußgänger verringert werden. Da sich die Lastpfade zu meist nur durch ihre Dimensionierung voneinander unterscheiden, werden in diesem Fall zwei CMS-Berechnungsmodelle, wie sie in Abb. 6.5 dargestellt sind, miteinander kombiniert. Die Koppelung der beiden Modelle erfolgt durch die Intrusion der Barriere ins Fahrzeug. So ist die Position der Barriere relativ zum Fahrzeug bei beiden Lastpfaden gleich. Das kombinierte Modell mit zwei Lastpfaden wird ausschließlich für den *Structure Tests* eingesetzt, da die verwendete Barriere vom Boden bis über die Motorhaube reicht und es daher zu einer Verformung von beiden Lastpfaden kommt. Im Gegensatz dazu trifft die Barriere des *Bumper Tests* ausschließlich den Hauptlastpfad, sodass auf die Modellierung eines zweiten Lastpfades verzichtet werden kann.

Mit den eingesetzten Komponentenmodellen können Auswirkungen von Geometrie- oder Materialänderungen untersucht und gezielt beeinflusst werden. Im Fall des Prallschaums wird anhand dieser Eingangsdaten eine Kraft-Weg-Charakteristik $F_P(x_P)$ der einzelnen Federelemente berechnet. Als Geometrieangaben stehen zum einen die Auflagefläche am Querträger A_{Px} , die sich aus der Gesamtlänge und -höhe des Querträgers l_{Qy} bzw. l_{Qz} ergibt, und zum anderen die Tiefe der Schaumgeometrie in x -Richtung l_{Px} zur Verfügung. Während der erste Wert durch den Querträger bestimmt wird, muss l_{Px} im Laufe des Entwicklungsprozesses in Kombination mit dem Schaummaterial optimiert werden, um Anforderungen an den Fußgängerschutz zu erfüllen. Die Materialcharakteristik des Schaums liegt üblicherweise in Form einer Spannungs-Dehnungskurve $\sigma(\varepsilon)$ vor. Um nun die Kraftcharakteristik $F_P(x_P)$ eines Federelements in Abhängigkeit von seiner Verformung in x -Richtung zwischen $P_{P,i}$ und dem Querträgermodell zu berechnen, sind zusätzlich die Anzahl der Knoten i_e am Querträger notwendig:

$$F_P(x_P) = \sigma(\varepsilon) l_{Qz} \frac{l_{Qy}}{i_e} \frac{x_P}{l_{Px}}. \quad (6.1)$$

Anhand dieser Federkennlinie $F_P(x_P)$ und der Eindrückung $x_{P,i}$ an einem Knoten $P_{P,i}$ lässt sich die Energieaufnahme eines Federelements $E_{P,i}$ ermitteln:

$$E_{P,i} = \int F_P(x_P) dx_{P,i}. \quad (6.2)$$

Die Summe über alle i Knotenpunkte ergibt schließlich die Gesamtenergieaufnahme des Prallschaums:

$$E_P = \sum_{i=1}^{i_e} E_{P,i}. \quad (6.3)$$

Das Komponentenmodell des Prallschaums ist, wie auch in Abb. 6.5 dargestellt, gekoppelt mit dem Querträgermodell. So lassen sich zu jedem Zeitpunkt t aus den Relativverschiebungen des Schaums $x_{P,i}$, die bei einem Eindringen des Barrieremodells auftreten, und der hinterlegten Federcharakteristik $F_P(x_P)$ die Schaumkräfte $F_{P,i}$ berechnen. Diese Kräfte wirken gleichzeitig als äußere Kräfte auf das Querträgermodell und es kommt zu einer Verformung des Querträgers. Diese Verformungen haben wiederum Auswirkung auf die $x_{P,i}$, die sich aus den Positionen der $P_{Q,i}$ und $P_{P,i}$ ergeben. Daher ist es notwendig, zu jedem Zeitpunkt iterativ einen Gleichgewichtszustand zwischen der Schaum- und der Querträgerverformung zu ermitteln. Im CMS-Gesamtmodell erfolgt die Berechnung dieses Gleichgewichtszustands mit Hilfe einer Optimierungsroutine. Darüber hinaus enthält die Koppelung von Schaum- und Querträgermodell ein vereinfachtes Kontaktmodell [Fre12].

Das Querträgermodell ist als FEM-Struktur, wie sie auch vielfach in der Literatur aufbereitet ist, aufgebaut [Zie75, BZ01]. In dem hier vorgestellten Modell kommen kombinierte Stab- und Balkenelemente zum Einsatz, mit denen die Knotenpunkte $P_{Q,i}$ verbunden werden. Der elastische Elementtyp berücksichtigt sowohl Normal- als auch Querkkräfte und die Elemente des Querträgermodells lassen sich entsprechend des FEM-Formalismus in einer Steifigkeitsmatrix \mathbf{K} zusammenfassen. Diese Steifigkeitsmatrix \mathbf{K} verbindet die von außen einwirkenden Kräfte \mathbf{F} und die Verformungen \mathbf{u} , die an den Knotenpunkten $P_{Q,i}$ auftreten:

$$\mathbf{F} = \mathbf{K}\mathbf{u}. \quad (6.4)$$

\mathbf{F} enthält die Schaumkräfte $F_{P,i}$, welche die äußeren Kräfte in x-Richtung darstellen sowie Null-Einträge für die vernachlässigten äußeren Kräfte in y-Richtung und für die äußeren Momente. \mathbf{u} umfasst die Verformungen der $P_{Q,i}$ in x- und y-Richtung sowie die Verdrehwinkel der Balkenelemente. In die Steifigkeitsmatrix \mathbf{K} gehen sowohl die Geometrie des Querträgers als auch das Verhalten des verwendeten Werkstoffs ein. Letzteres wird im linearen Fall durch den Elastizitätsmodul E berücksichtigt. Die geometrischen Einflussfaktoren des Querträgers umfassen das Flächenträgheitsmoment um die z-Achse I_{Qz} , die Querschnittsfläche A_{Py} und die Gesamtlänge des Querträgers l_{Qy} . Während sich l_{Qy} aus der Fahrzeugbreite ergibt und somit vorgegeben ist, können die übrigen Parameter variiert werden. Dadurch kann die Steifigkeit des

Querträgermodells gezielt verändert werden. Auf das verwendete Balkenelement wird in Anh. A.2 eingegangen.

Die für die Berechnung des Gesamtsystems benötigte Energieaufnahme dieses elastischen Fachwerks berechnet sich aus,

$$E_{Qel} = \frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u}. \quad (6.5)$$

Jedoch bildet diese ideal-elastische Struktur die plastischen Verformungen, welche selbst bei *Low-Speed Crash Tests* auftreten, nur unzureichend ab. Daher kommt es zu einer Erweiterung des Querträgermodells durch das Einfügen von plastischen Gelenken. Ein Umschalten von einer elastischen Verbindung zu einem Fließgelenk erfolgt an jenen Knotenpunkten $P_{Q,i}$, bei denen während eines Berechnungsdurchlaufs ein maximales Biegemoment M_{Qpl} überschritten wird. Es wird davon ausgegangen, dass M_{Qpl} erreicht wird, sobald die Randfasern des Querträgers über der Streckgrenze des Materials σ_y belastet werden. Mit diesem Ansatz lässt sich zwar kein komplexes Bauteilversagen, wie etwa durch *Kecman* ausgearbeitet [Kec83], berücksichtigen, es ermöglicht jedoch qualitative Aussagen in einem frühen Entwicklungsstadium.

Die Energie, die in einem plastischen Gelenk abgebaut wird, berechnet sich aus M_{Qpl} und dem zusätzlichen Biegewinkel ζ_{pl} :

$$E_{Qpl,i} = \int 2M_{Qpl} d\zeta_{pl}. \quad (6.6)$$

Der Faktor zwei berücksichtigt, dass die plastische Verformung beidseitig des Gelenkes auftritt. Sollten sich $k = 1, \dots, k_e$ plastische Gelenke ausbilden, so ergibt wiederum die Summe aller $E_{Qpl,i}$ den plastischen Energieanteil des Querträgers

$$E_{Qpl} = \sum_{k=1}^{k_e} E_{Qpl,i}. \quad (6.7)$$

Wie auch in Abb. 6.5 dargestellt, besitzt das Querträgermodell neben der Koppelung zum Prallschaummodell ebenfalls eine Koppelung zum Crashboxmodell und übernimmt daher eine zentrale Rolle. Während jedoch die Schaumkräfte $F_{P,i}$ als äußere Kräfte auf den Querträger wirken, sind die Crashboxen gleich wie der Querträger als lineare Stab- bzw. Balkenelemente modelliert und in der FE-Struktur des Querträgers integriert. Allerdings hat das Crashboxelement aufgrund seiner Geometrie sowohl in Quer- als auch in Normalrichtung eine wesentlich höhere Steifigkeit als ein typisches Querträgerelement.

Um mit dem Crashboxmodell ein Falten der Crashbox abzubilden und somit die Energieaufnahme dieses Deformationselements darzustellen, ist zusätzlich das Modell eines Stauchrohrs implementiert. Mit diesem Modell lässt sich auf ein durchschnittliches Kraftniveau F_{Rpl} schließen, welches beim Stauchen in Axialrichtung – also normal zum Rohrquerschnitt – auftritt. In weiterer Folge kann damit die umgewandelte Energie berechnet werden. Das verwendete Modell basiert auf dem Ansatz der *Superfolding-Elemente*, welcher in der Literatur weitreichend

beschrieben wird. So gibt beispielsweise *Abramowicz*, der selbst an der Entwicklung dieser Methode beteiligt war, einen Überblick und liefert weiterführende Literaturstellen [Abr03].

Dieser Ansatz macht sich geometrische Ähnlichkeiten beim Faltenbeulen von dünnwandigen Profilen zu Nutze, um modellhaft grundlegende Faltmechanismen zu beschreiben. Da beim Stauchen dieser Profile regelmäßige Faltmuster entstehen, wird nur eine Schicht dieses Musters betrachtet und die Ergebnisse auf die übrigen Schichten übertragen. Anhand der Faltmechanismen lassen sich die Verformungen, welche beim Stauchen einer Schicht auftreten, vereinfacht darstellen. Aus diesem Modell der Faltenbildung ergeben sich fünf plastische Verformungsmechanismen, mit denen die Energieaufnahme des Systems berechnet werden kann. In weiterer Folge können unter anderem Aussagen über die Schichthöhe der Faltmuster und die durchschnittliche Deformationskraft F_{Rpl} , die für die Stauchung des Profils notwendig ist, gemacht werden.

Die Methode steht offen für eine Reihe an möglichen Querschnittsgeometrien, wie etwa Hutprofile, Sechse- oder Achtecke, jedoch wird im Fall des Crashboxmodells nur auf einen rechteckigen Querschnitt zurückgegriffen, da diese Querschnittsform sowohl bei CMS-Deformationselementen als auch generell in der Fahrzeugstruktur vorherrscht. Infolge der großen Bedeutung für die Craschauslegung von Fahrzeugen wird dieser Spezialfall ausführlich durch *Ambrosio* beschrieben [Amb01]. Hierbei wird nicht nur eine Zusammenfassung der übergeordneten *Superfolding-Element-Methode* gegeben, sondern auch detailliert auf das Verhalten von Stauchrohren mit rechteckigem Querschnitt eingegangen und es werden die Resultate für ein freistehendes Deformationselement hergeleitet. Des Weiteren werden Modellvereinfachungen hinsichtlich der Materialkennwerte angeführt. Im einfachsten Fall eines ideal-plastischen Materials kann demnach F_{Rpl} folgendermaßen berechnet werden:

$$F_{Rpl} = 52,2 \frac{\sigma_y l_{R3}^2}{4} \sqrt[3]{\frac{l_{R1} + l_{R2}}{2l_{R3}}}. \quad (6.8)$$

Die Einflussparameter sind zum einen die Streckgrenze des Materials σ_y und zum anderen Geometriedaten des Crashbox-Querschnitts, wie die Seitenlängen des Rechtecks l_{R1} bzw. l_{R2} und die Dicke des Stauchrohrmaterials l_{R3} . Der Faktor 52,2 umfasst sowohl Faktoren, die sich aus der Berechnung entsprechend dem *Superfolding-Element-Ansatz* ergeben, als auch Annahmen, wie die Blocklänge des Stauchrohrs. Diese wird mit 27% der Ausgangslänge angegeben und ist für einen weiten Bereich an Querschnitten gültig [Amb01]. In Gl. 6.8 wird zwar nur ein Faltmechanismus – nämlich der symmetrische – betrachtet, die Ergebnisse bei den restlichen Mechanismen unterscheiden sich jedoch nur unwesentlich. So gibt *Abramowicz* abhängig vom Profilquerschnitt und von der Stauchrohrhöhe eine Abweichung von 4-8 % an [AW89].

Gl. 6.8 gibt die durchschnittliche Deformationskraft in Abhängigkeit vom Profilquerschnitt und dem gewählten Material an. Sofern also ein Kraftniveau bekannt ist, bei dem das Falten der Crashbox eintreten soll, lässt sich gleichfalls auf möglich Werte von l_{R1} , l_{R2} , l_{R3} und σ_y schließen. Dies ist insbesondere bei der Abstimmung der einzelnen Crashkomponenten der Fall, Kap. 6.4.1.

Eine mögliche Erweiterung von Gl. 6.8 liefert die Modellierung eines Materialverhaltens, welches die dehnungsabhängige Verhärtung berücksichtigt. Dies kann durch den Potenzansatz

$$\sigma(\varepsilon) = \sigma_0 \left(\frac{\varepsilon}{\varepsilon_0} \right)^q \quad (6.9)$$

erfolgen. Bei diesem Ansatz sind σ_0 und ε_0 zwei bekannte Werte der Spannungs-Dehnungskurve, wie beispielsweise die Streckgrenze $\sigma_0 = \sigma_y$ bei $\varepsilon_0 = 0,2\%$. Anhand des Exponenten q kann nun das tatsächliche Materialverhalten bei einer dehnungsabhängigen Verhärtung approximiert werden [Amb01]. Eine weitere Detaillierung von Gl. 6.8 lässt sich durch die Berücksichtigung einer geschwindigkeitsabhängigen Verhärtung mit Hilfe des *Cowper-Symonds*-Ansatzes erzielen [CS57]. Die beiden angeführten Erweiterungen sind ausführlich durch *Abramowicz* und *Ambrosio* beschrieben und werden hier nicht weiter ausgeführt [AJ84, Amb01]. Das Stauchrohrmodell, welches im CMS-Gesamtsystem implementiert ist, beinhaltet die beiden vorgestellten Erweiterungen. Allerdings hat sich im industriellen Einsatz gezeigt, dass aus Mangel an entsprechenden Materialkennwerten in der frühen Entwicklungsphase zumeist auf ein ideal-plastisches Materialverhalten, wie es Gl. 6.8 enthält, zurückgegriffen wird.

Unabhängig davon, mit welchem Detaillierungsgrad F_{Rpl} berechnet wird, dient dieses Kraftniveau im Stauchrohrmodell als Triggerwert. So wird die Crashbox bis zum Erreichen dieses Wertes als lineares Stab-Balkenelement behandelt und ist in der FE-Struktur mit dem Querträgermodell kombiniert. Nach dem Überschreiten dieses Wertes wird die Normalsteifigkeit des Stab-Balkenelements reduziert, um das Stauchen der Crashbox abzubilden. Gleichzeitig wird ab diesem Zeitpunkt die Relativverschiebung zwischen dem Fußpunkt der Crashbox und der Verbindungsstelle zum Querträger x_{Rpl} für die Berechnung der plastischen Energieaufnahme des Stauchrohrmodells E_{Rpl} herangezogen:

$$E_{Rpl} = F_{Rpl} x_{Rpl}. \quad (6.10)$$

Im hier realisierten Ansatz wird unter anderem davon ausgegangen, dass F_{Rpl} während des Faltens konstant bleibt. Dies trifft zwar nicht für ein Stauchrohr mit konstantem Querschnitt zu, allerdings können mit Zusatzmaßnahmen wie Sicken oder Löchern die Schwankungen im Kraftverlauf homogenisiert werden. Da diese Maßnahmen im Fahrzeugbau üblich sind und dadurch im realen Fahrzeug ein weitgehend gleichmäßiger Kraftverlauf erzielt wird, ist diese Annahme zulässig [HSHL10].

Das Crashboxmodell liefert in Kombination mit dem Barriere-, Schaum- und Querträgermodell das Gesamt-Berechnungssystem eines CMS, wie es in Abb. 6.5 dargestellt ist. Anhand dieses Gesamtsystems lassen sich nun die Verformungen, wie sie bei einem *Structure* oder einem *Bumper Test* auftreten, ermitteln. Der Berechnungsdurchlauf erfolgt schrittweise, wobei das entsprechende Barrieremodell bei jedem Berechnungsschritt um einen vorgegebenen Wert in x-Richtung ins CMS-Modell eindringt. Hierbei handelt es sich um keinen konstanten Wert, sondern die Schrittweite wird während des Berechnungsdurchlaufs ständig angepasst, um einerseits die Stabilität der Berechnung zu erhöhen und andererseits die Berechnungsdauer zu

reduzieren. In jedem Berechnungsschritt wird iterativ die Verformung des CMS-Gesamtmodells berechnet und darauf aufbauend die abgebaute Energie E_{CMS} ermittelt. Diese setzt sich aus den Energiewerten der einzelnen Submodelle von Barriere bzw. Schaum E_P , Querträger E_Q und Crashbox E_R zusammen:

$$E_{CMS} = E_P + E_Q + E_R. \quad (6.11)$$

Die abgebaute Energie E_{CMS} steht der kinetischen Energie E_0 , die das Fahrzeug zu Beginn des Crashes aufweist, gegenüber. E_0 wird bestimmt von der Anfangsgeschwindigkeit des Fahrzeuges v_0 , die in den entsprechenden Crashversuchen vorgegeben ist, und der Fahrzeugmasse m_0 :

$$E_0 = \frac{1}{2} m_0 v_0^2. \quad (6.12)$$

Sobald also die Anfangsenergie E_0 vollständig in die elastische und plastische Verformung des Fahrzeugs und der Barriere umgewandelt ist, hat die Barriere die höchste Intrusion in das Fahrzeug und es tritt die höchste Verformung der Fahrzeugkomponenten auf. Zu diesem Zeitpunkt kann der Berechnungsdurchlauf beendet werden und es lassen sich die auftretenden Verformungen anzeigen.

Da jedoch im CMS-Modell nicht alle Komponenten modelliert sind, die bei einem *Low-Speed Crash Test* beschädigt werden, führt das gewählte Abbruchkriterium, nämlich das Überschreiten der berechneten Verformungsenergie E_{CMS} vom gegebenen Wert E_0 , zu einer konservativen Auslegung. Demnach wird die Beschädigung von Kleinteilen, Haltern, Scheinwerfern oder der Motorhaube nicht berücksichtigt. Folglich ist das Abbruchkriterium des Berechnungsdurchlaufs um einen entsprechenden Faktor f_{CMS} anzupassen, um ein vergleichbares Ergebnis zu erhalten und eine zu starke Verformung der CMS-Komponenten zu verhindern. Betrachtet man den Anteil an der Gesamtenergie, die bei einem *Low-Speed Crash Tests* tatsächlich durch die CMS-Komponenten abgebaut wird, liegt dieser bei ca. 70 %. Abb. 6.6 zeigt beispielhaft die Energieanteile von Crashboxen und Stoßfängern aus zwei *Structure* und aus zwei *Bumper Tests*. Bei den untersuchten Fahrzeugen liegt der Energieanteil der CMS-Komponenten zwischen 63 und 83 %, was verdeutlicht, dass dieser Energieanteil stark vom jeweiligen Fahrzeug abhängt und in einem weiten Bereich schwankt. Darüber hinaus ist ersichtlich, dass ein Großteil der Energie beim *Structure Test* durch die Crashboxen abgebaut wird, während beim *Bumper Tests* der Stoßfänger die größte Verformungsenergie aufnimmt.

Die hohe Schwankungsbreite der Energieaufnahme von CMS-Komponenten erschwert das Festlegen des Korrekturfaktors f_{CMS} , mit dem die Anfangsenergie angepasst wird. Dadurch kann bei diesem Anwendungsfall auf eine im Entwicklungsprozess folgende Verifikation der Vorauslegung mit Hilfe von detaillierteren Simulationen oder Versuchen nicht verzichtet werden. Darüber hinaus kann mit einer großen Datenbasis an verifizierten Ergebnissen der Wertebereich von f_{CMS} eingegrenzt werden, genauso wie die Ergebnisgüte der Vorauslegung überprüft werden kann. Dies unterstreicht die Wichtigkeit eines Informationsrückflusses, um die Wirksamkeit

von *Funktionstemplates* – wie auch generell von KBE-Anwendungen, Kap. 3.2 – zu überprüfen und Möglichkeiten für Verbesserungen aufzuzeigen.

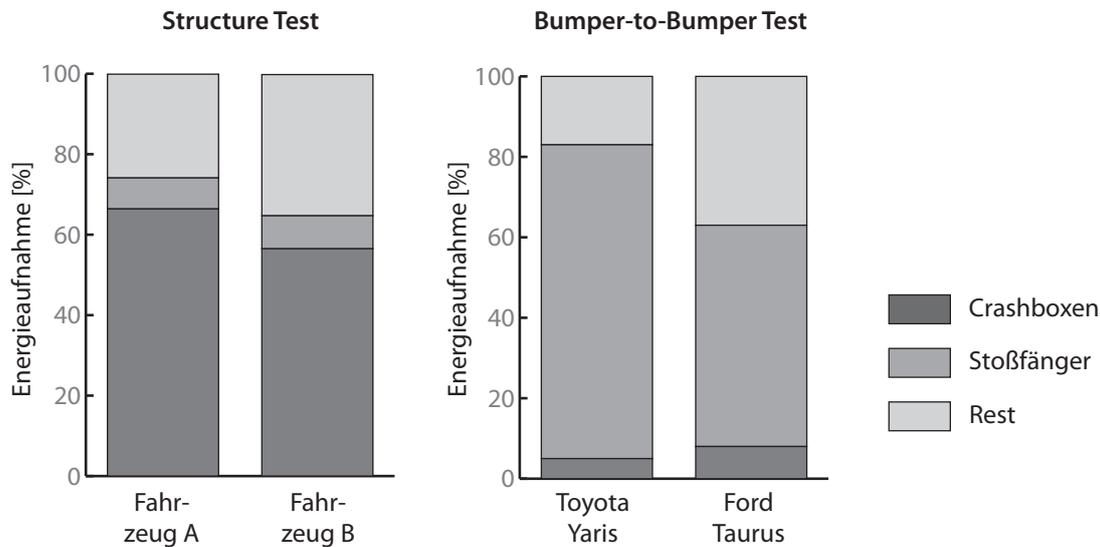


Abbildung 6.6: Mit Hilfe von Crashberechnungen ermittelte Energieanteile von Crashboxen und Querträgern bei unterschiedlichen *Low-Speed Crash Tests*^{32,33}.

6.4.4 Bewertung des Lösungswegs

Anhand des vorgestellten CMS-Gesamtmodells lassen sich zum einen die Verformungen aufgrund der *Low-Speed Crash Tests* berechnen und zum anderen die Geometrie- und Werkstoffkennwerte der jeweiligen Komponenten gezielt anpassen. So können insbesondere bei der Dimensionierung der Bauteile Rückschlüsse auf den Einfluss der einzelnen Parameter gezogen werden. Die Ergebnisqualität ist für eine frühe Phase ausreichend und lässt qualitative Aussagen zu. Dies unterstreicht eine Gegenüberstellung der Ergebnisse des CMS-Gesamtmodells mit einer FEM-basierten Crashberechnung [Fre12].

Für diesen Vergleich wurde ein FEM-Modell erstellt, welches das CMS-Gesamtmodell abbildet und aus zwei Stauchrohren, einem Querträger samt Prallschaum und den entsprechenden Barrieren besteht. Durch den Vergleich dieser Modelle sollten die Ergebnisse der Vorauslegung plausibilisiert und dadurch die Aussagequalität bestätigt werden. Abb. 6.7 zeigt einige Ergebnisse dieser Gegenüberstellung sowohl für den *Structure* als auch für den *Bumper Test*. Demnach zeigen die Verformungen bei unterschiedlichen Parametervariationen großteils eine gute Übereinstimmung. Stärkere Abweichungen treten vor allem durch die Vereinfachung des Stauchrohrmodells auf, welches im CMS-Gesamtmodell lediglich mit einer durchschnittlichen

³²Die Daten des *Structure Tests* stammen vom Industriepartner Magna Steyr Fahrzeugtechnik, Graz. Die Daten des *Bumper Tests* sind aus der Literatur entnommen [SSS12]

³³Das Fahrzeug Ford Taurus besitzt keine separaten Crashboxen, sondern stellt diese durch Versickungen am Längsträger dar. Daher wird der Energieanteil, der durch der Verformung des Längsträgers entsteht in Abb. 6.6 den Crashboxen zugerechnet.

Deformationskraft der Crashbox berücksichtigt ist, während die FEM-Simulation den Deformationskraftverlauf berechnet. Diese Abweichungen lassen dennoch qualitative Aussagen zu. Darüber hinaus wird im Laufe des Entwicklungsprozess versucht, den Deformationskraftverlauf der Crashboxen durch Zusatzmaßnahmen, wie Sicken oder Löcher, zu glätten und damit der durchschnittlichen Deformationskraft anzunähern. Daher ist davon auszugehen, dass die Abweichungen bei einem Vergleich der Crashbox-Deformationskraft zwischen dem CMS-Gesamtmodell und einer Crashberechnung der finalen Geometrie geringer ausfallen.

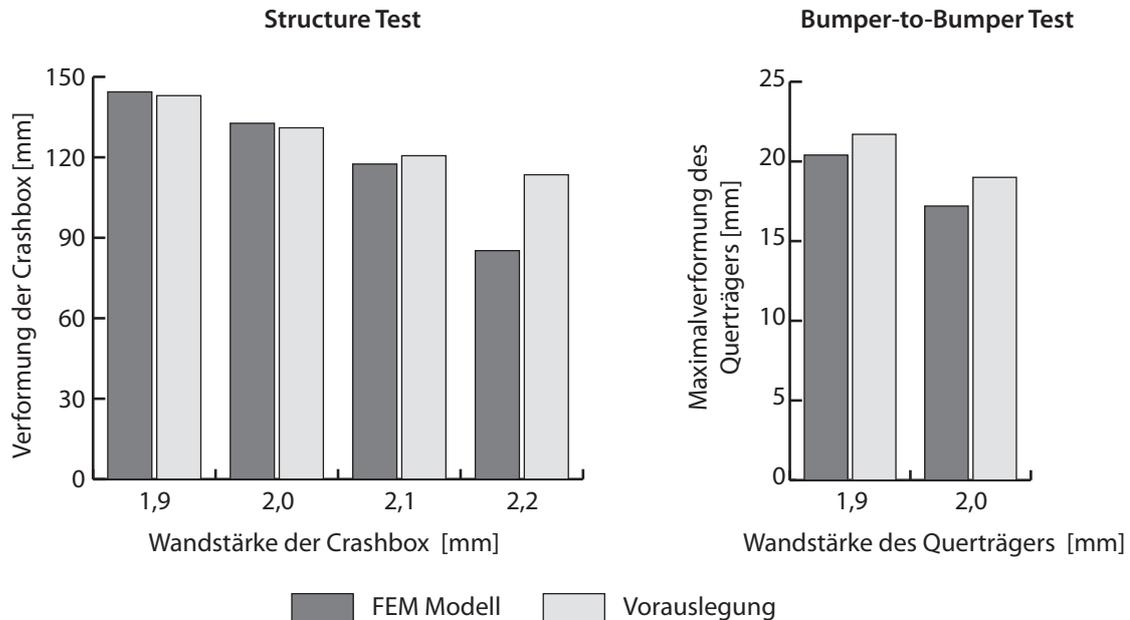


Abbildung 6.7: Vergleich zwischen dem vorgestellten CMS-Modell und einer FEM-basierten Crashberechnung der modellierten Komponenten.

Ein wesentlicher Vorteil des Lösungswegs im Vergleich zu einer FEM-Berechnung liegt in der vergleichsweise geringen Anzahl an Einflussparametern, die sich zumeist aus der Geometrie der einzelnen Komponenten ergeben. Darüber hinaus kommt der Ansatz mit relativ wenigen Werkstoffkenngrößen aus und verzichtet gänzlich auf die Eingabe von Berechnungsparametern, wie Netzgrößen oder Schrittweiten, sondern gibt hierfür Standardwerte vor. Ein weiterer Vorteil liegt im schnellen Berechnungsdurchlauf, der in der Regel in wenigen Minuten abgeschlossen ist. Dadurch können die Auswirkungen von geänderten Geometrie- oder Werkstoffdaten rasch bewertet werden.

Durch die angeführten Eigenschaften – schneller Berechnungsdurchlauf, relativ wenige Parameter, die gegebenenfalls auch mit Standardwerten befüllt sind, sowie qualitativ aussagekräftige Ergebnisse – eignet sich das Berechnungsmodell für den Einsatz in der Konzeptphase und sorgt für einen frühzeitigen Wissenszuwachs. Dadurch haben die Produktmodelle der einzelnen Komponenten einen höheren Reifegrad und Bauraumgrenzen können mit einer höheren Sicherheit festgelegt werden. Dies konnte auch im Rahmen des industriellen Einsatzes dieser Methode gezeigt werden. Dennoch bleibt eine Bestätigung der Ergebnisse durch detaillierte Berechnungsmethoden weiterhin notwendig. So sind einerseits die Ergebnisse der Vorauslegung zu überprüfen

und andererseits empirische Annahmen, wie etwa der Energieanteil der CMS-Komponenten, zu verifizieren und gegebenenfalls anzupassen. Gerade dieser Informationsrückfluss stellt einen wichtigen Faktor für die Weiterentwicklung von *Funktionstemplates* dar.

6.5 Geometriemodellierung eines CMS

Anhand der Ergebnisse der funktionalen Vorauslegung, nämlich der Dimensionierung der CMS-Komponenten, lässt sich mit Hilfe von *Geometrietemplates* eine detaillierte Konzeptgeometrie erzeugen. So stehen mit den Hauptabmessungen von Crashbox, Querträger und Prallschaum schon frühzeitig wesentliche Eingaben für die Geometriemodellierung zur Verfügung. Die Methoden, die bei der Geometriemodellierung in der hier beschriebenen prototypischen Umsetzung zur Anwendung kommen, gleichen jenen, welche auch schon bei der Erstellung des Türinnenblechs in Abschnitt 5.6 eingesetzt werden. Demnach werden *Templates* auf Bauteilebene in einer zentralen Bibliothek bereitgestellt und können in weiterer Folge in eine vorgegebene Baugruppenstruktur eingefügt werden.

In der hier beschriebenen prototypischen Umsetzung wird die CMS-Geometrie mit Hilfe von *Matingtemplates* für Crashboxen und Querträger dargestellt. Die meisten CMS-Anordnungen enthalten beide Bauteile, sodass ein weiter Anwendungsbereich abgedeckt ist. Allerdings gibt es eine Vielzahl an unterschiedlichen Varianten dieser Bauteile, welche sich zumeist aufgrund der Materialwahl oder aufgrund des Herstellungsverfahrens ergeben. So kommen Querträger aus Aluminium-Strangpressprofilen genauso zur Anwendung wie aus tiefgezogenen Stahlblechen. Das gleiche gilt für die Crashboxen, die ebenfalls aus geschweißten Blechen bestehen können oder aus Extrusionsprofilen mit Verprägungen. Daher werden in der hier vorgestellten Umsetzung beispielhaft jeweils zwei unterschiedliche *Matingtemplates* von Crashbox- und Querträgermodellen dargestellt, die sich miteinander kombinieren lassen. Abb. 6.8 zeigt exemplarisch die Anordnungen eines CMS mit jeweils verschiedenen Arten von Crashbox und Stoßfängerquerträger. Gerade die Möglichkeit, unterschiedliche Bauteil-*Templates* austauschbar zu gestalten, stellt eine Herausforderung für die Entwicklung von *Template*-Bibliotheken dar. So muss sichergestellt werden, dass alle *Matingtemplates* mit den vorhandenen Eingaben aus dem übergeordneten Skelettbauteil gültige Ergebnisse liefern. Dies erfordert in der Regel eine intensive Testphase und mitunter Anpassungen bei der Einführung von neuen *Template*-Varianten.

Die umgesetzte Baugruppenstruktur, in der die *Matingtemplates* von Crashbox und Querträger integriert werden können, wird in Abb. 6.1 dargestellt. Sie umfasst in der obersten Ebene das komplette Frontend und als Teil davon die Unterbaugruppe des CMS. In jeder Gliederungsebene sorgt ein Skelettbauteil für eine durchgängige Geometriesteuerung. Sowohl die Baugruppenstruktur als auch die Skelettbauteile dienen ausschließlich dazu, den Bauteilmodellen von Crashbox und Querträger eine übergeordnete Struktur bereitzustellen. Daher enthalten die Skelettbauteile ausschließlich Geometrielemente und Parameter, die für die Steuerung dieser beiden Bauteile benötigt werden. Das gleiche gilt für die Unterbaugruppen des Frontends, in

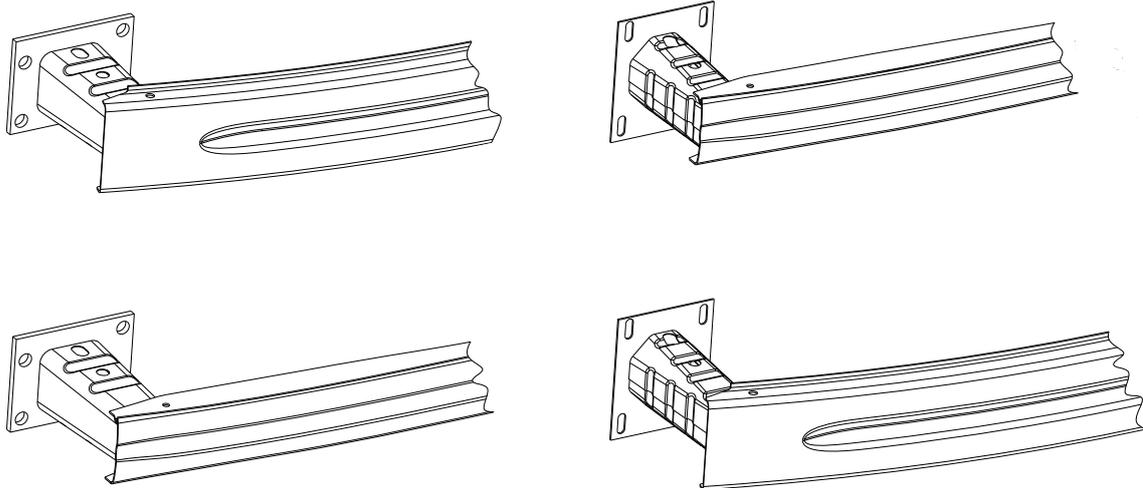


Abbildung 6.8: Varianten der CMS-Geometrie, die durch die Kombination aus jeweils zwei unterschiedlichen *Geometriemplates* von Crashbox und Querträger erstellt wurden.

denen weitere typische Bauteile, wie Kühlergrill oder Abschleppösen, nicht enthalten sind. Jedoch lassen sich sowohl die Skelettbauteile als auch die Baugruppenstruktur erweitern, sodass sich neben den Crashboxen und dem Querträger auch andere *Templates* integrieren lassen.

Da die Eingabeebene der Crashbox- und Querträger-*Matingtemplates* beim Einfügen in die Frontend-Baugruppe mit den Skelettbauteilen verknüpft werden, entsteht entsprechend des IPO- bzw. EVA-Schemas die Möglichkeit einer durchgängigen Geometriesteuerung. Dadurch wird sichergestellt, dass Eingaben, die etwa im Zuge der funktionalen Vorauslegung entstehen, unmittelbar zu einer Anpassung der Geometriemodelle führen. Infolgedessen kann die Länge der Crashbox genauso angepasst werden wie die Wandstärke des Querträgers. Darüber hinaus lassen sich Details, wie Versickungen und Bohrungen, steuern. Diese *Features* werden in der funktionalen Vorauslegung nicht berücksichtigt, allerdings wird die Möglichkeit einer externen Geometriesteuerung geschaffen, wie sie beispielsweise bei CAE-gesteuerten Optimierungsroutinen zum Einsatz kommt.

Ein weiteres Ergebnis der funktionalen Vorauslegung, welches in der Geometriemodellierung berücksichtigt wird, besteht in der berechneten Verformungen des CMS aufgrund der definierten Lastfälle. Diese Eingaben führen zwar zu keiner unmittelbaren Anpassung der CMS-Geometrie, jedoch lassen sich die Verformungen des Querträgers mit Hilfe von Referenzelementen darstellen. Diese befinden sich in der hier vorgeschlagenen Umsetzung im Skelettbauteil der CMS-Baugruppe und liefern eine wichtige Randbedingung für die Bauraumauslegung von umliegenden Bauteilen, wie etwa Motorkühlern.

Anhand des hier vorgestellten Beispiels werden die Vorteile einer *Template*-unterstützten Geometrieerzeugung im Rahmen der CAD-basierten Vorauslegung aufgezeigt. Demnach fließen die Ergebnisse der funktionalen Vorauslegung direkt in die resultierende Konzeptgeometrie ein. So

lassen sich beispielsweise der Crashboxquerschnitt oder die Wandstärken der einzelnen Komponenten anpassen. Dadurch finden notwendige Anpassungen rasch ihre Umsetzung in der Geometrie, die gleichzeitig eine erforderliche Eingabe für weiterführende Berechnungsschleifen darstellt. Darüber hinaus kann eine Vielzahl an Bauteilvarianten abgebildet werden und die unterschiedlichen Geometriemodelle lassen sich miteinander kombinieren. Diese Bauteilvarianten haben bereits einen hohen Detaillierungsgrad und berücksichtigen gängige Herstellungsverfahren. Demnach liegen genauso Modelle von Stoßfängern und Crashboxen aus stranggepressten Profilen wie aus tiefgezogenen Blechen vor. Aufgrund dieses hohen Detaillierungsgrades kann bereits in einer Konzeptphase auf hochwertige Geometriedaten zurückgegriffen werden. Des Weiteren sorgt der stringente Aufbau entsprechend des IPO- bzw. EVA-Schemas für eine leichte Adaptierbarkeit der enthaltenen *Features*, wie Versickungen oder Bohrungen.

6.6 Zusammenfassung

Die konzeptionelle Auslegung eines Crash Management Systems liefert das zweite Anwendungsbeispiel für die CAD-basierte Vorauslegung. Anhand von zwei charakteristischen Crash-Versuchsanordnungen können sowohl die einzelnen CMS-Komponenten ausgelegt als auch die Verformung des Gesamtsystems ermittelt werden. Letzteres unterstützt die Bauraumauslegung des Fahrzeugvorderwagens. Die Ergebnisse der funktionalen Vorauslegung dienen neben einem frühzeitigen Wissenszuwachs gleichfalls als Eingaben für die Geometriemodellierung mit Hilfe von *Templates*.

Innerhalb dieses Kapitels wird neben der Problembeschreibung sowohl auf die übergeordnete Anwendung der CAD-basierten Vorauslegung als auch auf deren prototypische Umsetzung eingegangen. So wird insbesondere das Berechnungsmodell beschrieben, welches in Form eines *Funktionstemplates* die Verformungen des CMS bei *Low Speed Crash Tests* ermittelt. Des Weiteren wird die Anwendung einer effizienten Geometriemodellierung mit Hilfe von *Geometrietemplates* dargelegt.

7 Zusammenfassung

Die frühe Phase in einer Fahrzeugentwicklung ist von großer Bedeutung für den Erfolg der Produktentstehung, da in diesem Abschnitt bereits die wesentlichen Produkthanforderungen definiert und darauf aufbauend technische Lösungskonzepte ausgearbeitet werden. Während ein ausgereiftes Fahrzeugkonzept, als Resultat einer erfolgreichen frühen Entwicklungsphase, für geringe Folgekosten, eine kurze Entwicklungszeit sowie eine hohe Produktqualität sorgt, können falsche Annahmen und Fehler in dieser Phase zu einem späteren Zeitpunkt nur mehr mit hohen Aufwänden korrigiert werden. Trotz des großen Einflusses auf den Entwicklungsprozess müssen in diesem frühen Entwicklungsabschnitt Entscheidungen zumeist anhand von unvollständigen und sich schnell ändernden Daten sowie basierend auf einer Vielzahl von Annahmen getroffen werden. Um dieser Herausforderung erfolgreich zu begegnen, zielt eine Reihe von Ansätzen auf eine frühzeitige, qualitativ hochwertige Informationsbereitstellung ab, welche die Entscheidungsgrundlagen verbessert und infolgedessen Fehlentscheidungen vermeidet.

Während in der sehr frühen Entwicklungsphase, der sog. Definitionsphase, etablierte Methoden und Werkzeuge für eine frühzeitige Gewinnung von Produktinformationen zur Verfügung stehen, wird der Übergang von der Definitions- in die darauf folgende Konzeptphase noch unzureichend unterstützt. Dieser Übergang ist durch eine massive Ausweitung des involvierten Personenkreises charakterisiert, die mit einem stark wachsenden Datenumfang, einem steigenden Detaillierungsgrad und einer damit verbundenen zunehmenden Produktkomplexität einhergehen. Allerdings können die Qualität und der Reifegrad der Daten oft nicht mit dem großen Zuwachs an Datenmaterial schritthalten. Dies erfordert einen vermehrten Abstimmungsaufwand und führt zu oftmaligen Änderungsschleifen. Um die Produktentwicklung in diesem sich schnell ändernden Umfeld zu unterstützen, sind daher flexible und gleichzeitig robuste Methoden und Werkzeuge notwendig.

Da Prototypen und Versuchsträger in diesem frühen Entwicklungsstadium nur in einem unzureichenden Maße zur Verfügung stehen, kommen vor allem virtuelle Werkzeuge zum Einsatz. Hierbei spielen moderne CAD-Programme, deren Fähigkeiten über eine reine Geometriemodellierung hinausgehen, eine entscheidende Rolle. So lassen sich unter anderem wissensbasierte Konstruktionsansätze umsetzen und beispielsweise funktionale Auslegungs- und Absicherungsschritte innerhalb der CAD-Umgebung durchführen. Ein wesentliches Element von modernen, wissensbasierten CAD-Programmen besteht in der Möglichkeit, vorhandenes Wissen in Form von Vorlagen sog. *Templates* abzulegen und dem Entwicklungsprozess leicht zugänglich zu machen. Dadurch lassen sich insbesondere in einer frühen Phase Entwicklungsabläufe effizienter gestalten und es kann die Qualität der Produktdaten gesteigert werden.

Der Einsatz von *Templates* zur Unterstützung der Produktentwicklung in einer frühen Phase wird im Rahmen dieser Arbeit aufgegriffen. Es wird auf die Beschreibung und Kategorisierung von *Templates* eingegangen und ihre Bedeutung für die Produktentwicklung beleuchtet. Darüber hinaus wird ein Vorgehensmodell erstellt, das eine Koppelung von funktionaler Absicherung und Auslegung mit einer teilautomatisierten Geometriemodellierung auf Baugruppen- und Bauteilebene mit Hilfe von *Templates* vorsieht. Dadurch wird einerseits eine konstruktionsnahe Auslegung von Produkteigenschaften unterstützt und andererseits lassen sich detaillierte Geometriemodelle erstellen, die unter anderem als hochwertige Eingangsdaten für Folgeprozesse dienen. Die umfassende Beschreibung der CAD-basierten Vorauslegung stellt einen Schwerpunkt der vorliegenden Arbeit dar. Es wird gleichermaßen auf Auswirkungen und Vorteile des entwickelten Ansatzes als auch auf Grenzen und Schwächen eingegangen.

Des Weiteren wird die beispielhafte Umsetzung der CAD-basierten Vorauslegung im Rahmen von zwei Anwendungsbeispielen behandelt. Dadurch kann die Wirksamkeit des aufgestellten Vorgehensmodells unterstrichen werden. Die Anwendungsbeispiele umfassen typische Entwicklungsaufgaben, die in der Konzeptphase einer Karosserieentwicklung bearbeitet werden. Es wird einerseits die Modellierung eines Türinnenblechs unter Berücksichtigung der Türkinematik behandelt und andererseits die Bauraumauslegung einer Stoßfängerstruktur dargestellt. Ein Schwerpunkt bei der Beschreibung der beiden Anwendungsbeispiele liegt auf den funktionalen Vorauslegungen, da diese zum Teil auf neuartigen Lösungsansätzen beruhen.

Die erste Aufgabenstellung –die Kinematikauslegung einer Standard-Fahrzeughür– ist stark geprägt von den Eingaben des Fahrzeugdesigns, die im Rahmen der Vorauslegung überprüft werden. So beeinflusst etwa die Anforderung eines kollisionsfreien Türöffnens die Fahrzeugaußenhaut und den Verlauf sowie die Breite der Türfuge. Ebenfalls gilt es, die Absenkbarkeit der Seitenscheibe abzusichern, die einerseits durch das Design und andererseits durch Vorgaben der Scheibendichtungen bestimmt wird. Die Ergebnisse der Kinematikauslegung werden dann als Eingangsdaten für die Erstellung eines Türinnenblechs herangezogen. Das zweite Anwendungsbeispiel –die Vorauslegung einer Stoßfängerstruktur– wird wesentlich bestimmt von Packaging-Anforderungen. So muss frühzeitig sichergestellt werden, dass genügend Bauraum für Deformationselemente, wie etwa Crashboxen oder Prallschäume, vorgehalten wird. Zusätzlich hat die Verformung der Deformationselemente aufgrund von standardisierten Crashtests Einfluss auf die geometrische Integration von angrenzenden Bauteilen, wie beispielsweise Kühlerpakete. Durch den Einsatz einer funktionalen Vorauslegung wird der notwendige Bauraum für eine Stoßfängerstruktur ermittelt und die einzelnen Bauteile vordimensioniert. Im Anschluss kann auf Basis dieser Ergebnisse mit Hilfe einer teilautomatisierten Geometriemodellierung eine detaillierte Konzeptgeometrie der Stoßfängerkomponenten erstellt werden. Beide Beispiele liefern erfolgreiche Anwendungen der CAD-basierten Vorauslegung und zeigen die Vorteile dieses Ansatzes auf. Darüber hinaus konnten für beide Aufgabenstellungen prototypisch umgesetzte, virtuelle Werkzeuge erstellt werden, die sich auch im industriellen Einsatz bewährt haben.

Neben der erfolgreichen Umsetzung der beiden Anwendungsbeispiele erhebt die vorliegende Arbeit einen wissenschaftlichen Neuheitsanspruch, der im Wesentlichen auf zwei Säulen beruht. Zum einen liefert das Vorgehensmodell der CAD-basierten Vorauslegung, als Anwendung von

wissensbasierten Konstruktionsmethoden, einen neuartigen Ansatz zur Verbesserung der Produktentwicklung in der Konzeptphase. Dadurch lässt sich in dieser frühen Entwicklungsphase die Qualität der Produktdaten unter Anwendung von CAD-Systemen steigern und in Form von detaillierten, adaptiven Geometriemodellen bereitstellen. Zum anderen kommen bei der funktionalen Vorauslegung im Rahmen der Anwendungsbeispiele neuartige Lösungsansätze zum Einsatz, die sich für eine Implementierung in wissensbasierten CAD-Systemen eignen. Somit stellt die vorliegende Arbeit einen wichtigen Beitrag für die Weiterentwicklung von computerunterstützten Methoden zur Produktentwicklung in der frühen Entwicklungsphase dar.

A Anhang

A.1 Plückerkoordinaten

Anhand von *Plückerkoordinaten* lassen sich Geraden eines drei-dimensionalen projektiven Raums \mathbb{P}^3 mit homogenen Koordinaten in einem fünf-dimensionalen Raum \mathbb{P}^5 darstellen. Diese Koordinaten sind nach dem deutschen Mathematiker Julius Plücker (1801–1868) benannt und ihre Verwendung zählt zu den Standard-Verfahren in der Liniengeometrie. Aus diesem Grund werden sie auch umfassend in der Fachliteratur behandelt, wie beispielsweise durch *Beutelsbacher* oder *Pottmann* [BR98a, PW01]. Auf eine Beweisführung wird hier daher verzichtet und an dieser Stelle werden lediglich in Anlehnung an *Gferrer* die *Plückerkoordinaten* eingeführt und definiert [Gfr04]. Außerdem wird überblicksartig auf die wesentlichen Eigenschaften eingegangen.

Gegeben sei eine Gerade g des drei-dimensionalen Raums \mathbb{P}^3 . P_x und P_y seien zwei Punkte auf g mit den homogenen Koordinaten x_1, x_2, x_3, x_4 und y_1, y_2, y_3, y_4 bezüglich eines fest gewählten Koordinatensystems. Es sei weiters

$$D_{ij} := \det \begin{bmatrix} x_i & y_i \\ x_j & y_j \end{bmatrix}. \quad (\text{A.1})$$

Dann heißen

$$g_1 := D_{01}, \quad g_2 := D_{02}, \quad g_3 := D_{03}, \quad \bar{g}_1 := D_{23}, \quad \bar{g}_2 := D_{31}, \quad \bar{g}_3 := D_{12} \quad (\text{A.2})$$

die *Plückerkoordinaten* und die Vektoren

$$\mathbf{g} := \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix}, \quad \bar{\mathbf{g}} := \begin{bmatrix} \bar{g}_1 \\ \bar{g}_2 \\ \bar{g}_3 \end{bmatrix} \quad (\text{A.3})$$

die *Plückervektoren* von g .

Für die *Plückerkoordinaten* gilt:

1. Ist $P_x \neq P_y$ dann ist $[\mathbf{g}^T, \bar{\mathbf{g}}^T] \neq [0, \dots, 0]$

2. Die *Plückerkoordinaten* bzw. *Plückervektoren* sind bis auf einen gemeinsamen Faktor $\rho \neq 0$ eindeutig durch die Gerade g festgelegt.
3. Für die *Plückerkoordinaten* bzw. *Plückervektoren* gilt die sog. *Plückerbedingung*:

$$g_1\bar{g}_1 + g_2\bar{g}_2 + g_3\bar{g}_3 = 0 \quad (\text{A.4})$$

bzw.

$$\langle \mathbf{g}, \bar{\mathbf{g}} \rangle = 0. \quad (\text{A.5})$$

Hierbei beschreibt $\langle \cdot, \cdot \rangle$ das Skalarprodukt zweier Vektoren.

Da die *Plückerkoordinaten* $g_1, g_2, g_3, \bar{g}_1, \bar{g}_2, \bar{g}_3$ einer Geraden nur bis auf einen gemeinsamen Faktor $\rho \neq 0$ bestimmt sind, liefert die Forderung

$$g_1^2 + g_2^2 + g_3^2 = 1 \quad (\text{A.6})$$

eine Normierung, sofern $(g_1, g_2, g_3) \neq (0, 0, 0)$. Daher spricht man von *normierten Plückerkoordinaten* sobald der erste Plückervektor \mathbf{g} der Forderung A.6 entspricht. Man beachte, dass $\bar{\mathbf{g}}$ nicht normiert sein muss.

Interpretiert man die *Plückerkoordinaten* g_1, \dots, \bar{g}_3 als homogenen Koordinaten von Punkten eines fünf-dimensionalen projektiven Raums \mathbb{P}^5 , werden durch die Bedingung A.4 die Punkte einer quadratischen Hyperfläche (Quadrik) \mathcal{M}^4 dieses Raums erfasst. Diese Quadrik wird nach dem deutschen Mathematiker Felix Klein (1849-1925) auch als *Kleinsche Quadrik* bezeichnet. Des Weiteren wird die Abbildung \mathcal{K} , welche einer Geraden g des \mathbb{P}^3 jenen Punkt G des \mathbb{P}^5 zuordnet, als *Kleinsche Geradenabbildung* bezeichnet, sofern die homogenen Koordinaten von G mit den *Plückerkoordinaten* von g übereinstimmen. Ist Γ die Menge aller Geraden des dreidimensionalen \mathbb{P}^3 , kann die *Kleinsche Geradenabbildung* als

$$\mathcal{K} : \left\{ \begin{array}{l} \Gamma \rightarrow \mathcal{M}^4 \\ g \rightarrow G \end{array} \right\} \quad (\text{A.7})$$

dargestellt werden. Im Übrigen kann gezeigt werden, dass diese Abbildung bijektiv ist.

An dieser Stelle wird noch eine euklidische Deutung der *Plückerkoordinaten* bzw. *Plückervektoren* gegeben. Sei also g eine eigentliche Gerade (Nicht-Ferngerade) mit dem Richtungsvektor

$$\mathbf{b} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}, \quad (\text{A.8})$$

dann besitzt der Fernpunkt P'_g von g die homogenen Koordinaten $0, b_x, b_y, b_z$. Des Weiteren sei P_g ein eigentlicher Punkt von g mit der affinen Koordinatenspalte

$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}. \quad (\text{A.9})$$

Dann sind $1, p_x, p_y, p_z$ die homogenen Koordinaten von P_g .

Werden nun gemäß der Definition A.2 die beiden Punkte P'_g und P_g als Punkte P_x und P_y herangezogen, ergibt die Berechnung der *Plückerkoordinaten*:

$$g_1 = \det \begin{bmatrix} 0 & 1 \\ b_x & p_x \end{bmatrix} = -b_x, \quad (\text{A.10})$$

$$g_2 = \det \begin{bmatrix} 0 & 1 \\ b_y & p_y \end{bmatrix} = -b_y, \quad (\text{A.11})$$

$$g_3 = \det \begin{bmatrix} 0 & 1 \\ b_z & p_z \end{bmatrix} = -b_z, \quad (\text{A.12})$$

$$\bar{g}_1 = \det \begin{bmatrix} b_y & p_y \\ b_z & p_z \end{bmatrix} = b_y p_z - b_z p_y, \quad (\text{A.13})$$

$$\bar{g}_2 = \det \begin{bmatrix} b_z & p_z \\ b_x & p_x \end{bmatrix} = b_z p_x - b_x p_z, \quad (\text{A.14})$$

$$\bar{g}_3 = \det \begin{bmatrix} b_x & p_x \\ b_y & p_y \end{bmatrix} = b_x p_y - b_y p_x. \quad (\text{A.15})$$

Da die Multiplikation mit einem beliebigen Faktor $\rho \neq 0$ erlaubt ist, liefert die Multiplikation mit $\rho = -1$: $\mathbf{g} = \mathbf{b}$, $\bar{\mathbf{g}} = \mathbf{p} \times \mathbf{b}$. Zusammenfassend gilt also: Ist g eine eigentliche Gerade, \mathbf{b} ein Richtungsvektor von g und \mathbf{p} der Ortsvektor eines Punktes $P_g \in g$, dann sind

$$\mathbf{g} = \mathbf{b} \quad , \quad \bar{\mathbf{g}} = \mathbf{p} \times \mathbf{b} \quad (\text{A.16})$$

Plückervektoren von g .

Der Vektor $\bar{\mathbf{g}}$ wird in diesem Zusammenhang auch als Momentenvektor der Geraden g bezeichnet. Wie auch in Abb. A.1 dargestellt, steht er normal zur Verbindungsebene des Koordinatenursprungs \mathcal{O} mit der Geraden g und sein Betrag entspricht der Fläche des von \mathbf{p} und \mathbf{b} aufgespannten Parallelogramms. Geraden durch \mathcal{O} sind daher auch durch einen verschwindenden Momentenvektor gekennzeichnet.

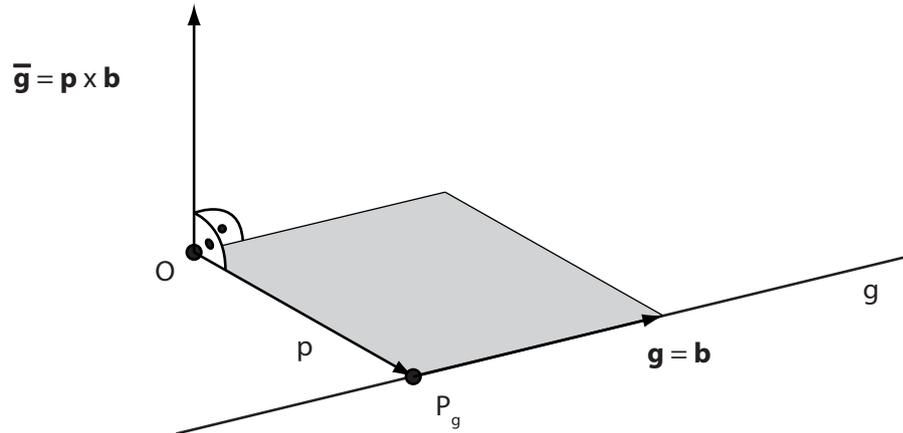


Abbildung A.1: Euklidische Deutung der beiden *Plückervektoren* \mathbf{g} und $\bar{\mathbf{g}}$ einer Geraden g .

A.2 Lineares Balkenelement

Bei dem Modell des CMS, welches in Abschnitt 6.4 beschrieben wird, kommt eine Kombination aus einem Stab- und einem Balkenelement zum Einsatz. Durch den hier verwendeten ebenen, elastischen Elementtyp lassen sich nicht nur die Verformungen normal zum Balken sondern auch in Balkenlängsrichtung berechnen. Auf die Abhängigkeiten zwischen den äußeren Belastungen und den resultierenden Verformungen wird hier überblicksartig eingegangen. Weiterführende Informationen und Herleitungen zu diesem Elementtyp, welcher vorwiegend für Fachwerke verwendet wird, finden sich in diversen Standardwerken zur FEM, wie beispielsweise jenen von *Zienkiewicz* und *Bathe* [Zie75, BZ01]

In Abb. A.2 wird das verformte Balkenelement dargestellt und neben den äußeren Belastungen sind ebenfalls die Verschiebungen an den Knotenpunkten P_1 und P_2 eingezeichnet. Die äußeren Belastungen bestehen aus den Kräften F_{1x} , F_{1y} , F_{2x} , F_{2y} und den Momenten M_1 und M_2 . Die Verformungen umfassen die Relativverschiebungen in x_0 -Richtung, u_1 und u_2 , die Relativverschiebungen in y_0 -Richtung, v_1 und v_2 , sowie den Tangentialwinkel an den Knoten, ζ_1 und ζ_2 . Sowohl die äußeren Belastungen als auch die Verformungen lassen sich in Vektoren zusammenfassen und bilden den Belastungsvektor \mathbf{F} und den Verformungsvektor \mathbf{u} . Sowohl \mathbf{F} als auch \mathbf{u} sind im Koordinatensystem Σ_0 , welches in Abb. A.2 durch x_0 und y_0 aufgespannt wird, gegeben.

$$\mathbf{F} = \begin{pmatrix} F_{1x} \\ F_{1y} \\ M_1 \\ F_{2x} \\ F_{2y} \\ M_2 \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} u_1 \\ v_1 \\ \zeta_1 \\ u_2 \\ v_2 \\ \zeta_2 \end{pmatrix}. \quad (\text{A.17})$$

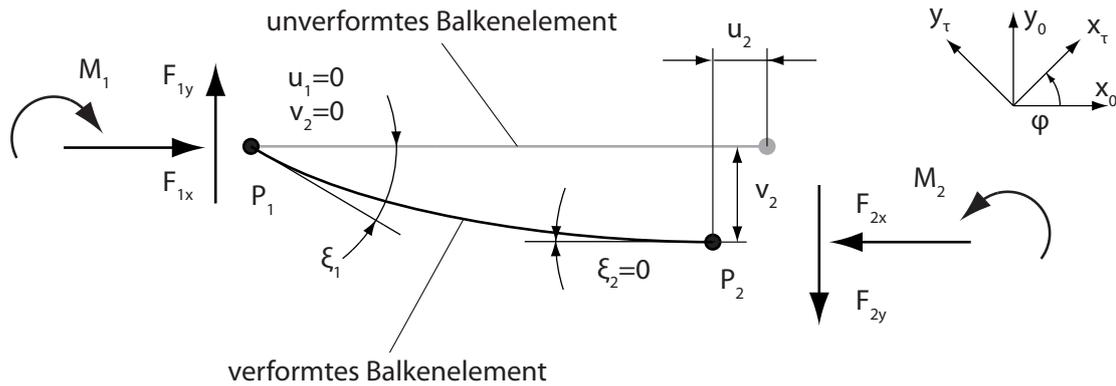


Abbildung A.2: Belastungen und Verformungen des elastischen Stab-Balkenelements.

Wie auch schon in Gl. 6.12 angeführt, werden die Belastungen \mathbf{F} und die Verformungen \mathbf{u} im ideal-elastischen Fall durch die Steifigkeitsmatrix \mathbf{K} miteinander verknüpft:

$$\mathbf{F} = \mathbf{K}\mathbf{u}. \quad (\text{A.18})$$

In \mathbf{K} werden Einflüsse sowohl aus der Geometrie als auch des Werkstoffverhaltens berücksichtigt. Unter die Parameter, die aus der Geometrie abgeleitet werden, fallen die Querschnittsfläche A , die Länge des Balkenelements l und das Flächenträgheitsmoment um die neutrale Balkenfaser I . Das Werkstoffverhalten wird durch den Elastizitätsmodul E bestimmt.

$$\mathbf{K} = \frac{EI}{l^3} \begin{bmatrix} \frac{Al^2}{I} & 0 & 0 & -\frac{Al^2}{I} & 0 & 0 \\ 0 & 12 & 6l & 0 & -12 & 6l \\ 0 & 6l & 4l^2 & 0 & -6l & 2l^2 \\ -\frac{Al^2}{I} & 0 & 0 & \frac{Al^2}{I} & 0 & 0 \\ 0 & -12 & -6l & 0 & 12 & -6l \\ 0 & 6l & 2l^2 & 0 & -6l & 4l^2 \end{bmatrix}. \quad (\text{A.19})$$

Die 6x6-Steifigkeitsmatrix \mathbf{K} lässt sich in vier 3x3-Untermatrizen \mathbf{K}_{11} , \mathbf{K}_{12} , \mathbf{K}_{21} und \mathbf{K}_{22} aufteilen:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{21} \\ \mathbf{K}_{12} & \mathbf{K}_{22} \end{bmatrix}. \quad (\text{A.20})$$

Sollte das Stab-Balkenelement eine allgemeine Lage besitzen, also nicht genau in Richtung der x_0 -Achse zeigen, ist eine Koordinatentransformation erforderlich. Dies ist insbesondere dann notwendig, wenn mehrere Balkenelemente zu einem Fachwerk verknüpft werden und nur ein globales Koordinatensystem für die Berechnung des Gesamtsystems maßgeblich ist. In Abb.

A.2 wird dieser Fall durch das Koordinatensystem Σ_τ , welches um φ zum Koordinatensystem Σ_0 gedreht ist, dargestellt. Die entsprechende Transformationsmatrix von Σ_0 nach Σ_τ lautet:

$$\mathbf{T}_{\tau 0} = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (\text{A.21})$$

Wendet man die Koordinatentransformation bei Gl. A.18 an, erhält man nach Umformen:

$$\mathbf{F}_\tau = \begin{bmatrix} \mathbf{T}_{\tau 0}^T \mathbf{K}_{11} \mathbf{T}_{\tau 0} & \mathbf{T}_{\tau 0}^T \mathbf{K}_{21} \mathbf{T}_{\tau 0} \\ \mathbf{T}_{\tau 0}^T \mathbf{K}_{12} \mathbf{T}_{\tau 0} & \mathbf{T}_{\tau 0}^T \mathbf{K}_{22} \mathbf{T}_{\tau 0} \end{bmatrix} \mathbf{u}_\tau. \quad (\text{A.22})$$

Durch die Kombination mehrerer Balkenelemente, wie sie durch Gl. A.22 beschrieben werden, lassen sich entsprechend des *Finite-Elemente*-Ansatzes umfangreiche Modelle erstellen. Durch Aufbringen von Randbedingungen und Berechnungseingaben kann in weiterer Folge das Festigkeitsverhalten dieser Modelle simuliert werden.

Literaturverzeichnis

- [Abr03] ABRAMOWICZ, W.: Thin-walled structures as impact energy absorbers. In: *Thin-Walled Structures* 41 (2003), Nr. 2-3, S. 91–107. – ISSN 0263–8231
- [Abr07] ABRAMOVICI, M.: Future Trends in Product Lifecycle Management (PLM). In: KRAUSE, F. (Hrsg.): *The Future of Product Development – Proceedings of the 17th CIRP Design Conference*. S. 665-674. Berlin : Springer Verlag, 2007. – ISBN 978-3-540-69819-7
- [AH94] ALLEN, G. ; HASSELBECK, J.: Associative Modelling for Mechanical CAD/CAM. In: HOSCHEK, J. (Hrsg.) ; DANKWORT, W. (Hrsg.): *Parametric and Variational Design*. S. 3-15. Stuttgart : Teubner Verlag, 1994. – ISBN 3-519-02632-5
- [AJ84] ABRAMOWICZ, W. ; JONES, N.: Dynamic axial crushing of square tubes. In: *International Journal of Impact Engineering* 2 (1984), Nr. 2, S. 179–208. – ISSN 0734-743X
- [Amb01] AMBROSIO, J. (Hrsg.): *Crashworthiness – Energy Management and Occupant Protection*. Wien : Springer Verlag, 2001. – ISBN 978-3-211-83334-6
- [Amt03] AMTSBLATT DER EUROPÄISCHEN UNION: L 321/15: *Richtlinie 2003/102/EG des Europäischen Parlaments und des Rates zum Schutz von Fußgängern und anderen ungeschützten Verkehrsteilnehmern vor und bei Kollisionen mit Kraftfahrzeugen und zur Änderung der Richtlinie 70/156/EWG des Rates*. Brüssel, 2003
- [And93] ANDERL, R.: *CAD-Schnittstellen – Methoden und Werkzeuge zur CA-Integration*. München : Hanser Verlag, 1993. – ISBN 978-3-446-17103-9. – zgl. Habilitation an der Universität Karlsruhe
- [Arn08] ARNDT, H.: *Eine Ontologie-basierte Methode zur Entscheidungsunterstützung in der Produktentwicklung*. Stuttgart : IRB Verlag, 2008. – ISBN 978-3-8167-7641-3. – zgl. Dissertation an der TU Berlin
- [AW89] ABRAMOWICZ, W. ; WIERZBICKI, T.: Axial Crushing of Multicorner Sheet Metal Columns. In: *Journal of Applied Mechanics* 56 (1989), Nr. 1, S. 113–120. – ISSN 0021-8936
- [BABS10] BERKAN, D. ; ANDERSSON, H. ; BÖTTRICH, M. ; STRUCKMANN, C.: Advanced Chassis Engineering: Neue Wege bei der automatisierten Ableitung von FE-Modellen im Entwicklungsprozess. In: TECKLENBURG, G. (Hrsg.): *Die digitale*

- Produktentwicklung* Bd. 2, S. 177-188. Renningen : Expert Verlag, 2010. – ISBN 978-3-8169-2961-1
- [BGW11] BRACHT, U. ; GECKLER, D. ; WENZEL, S.: *Digitale Fabrik – Methoden und Praxisbeispiele*. Berlin : Springer Verlag, 2011. – ISBN 978-3-540-88973-1
- [Böh04] BÖHME, M.: *Ein methodischer Ansatz zur parametrischen Produktmodellierung in der Fahrzeugentwicklung*. Düsseldorf : VDI Verlag, 2004. – ISBN 978-3-18-356712-6. – zgl. Dissertation an der TU Berlin
- [Bos07] BOSSMANN, M.: *Feature-basierte Produkt- und Prozessmodelle in der integrierten Produktentstehung*. Saarbrücken : Universität des Saarlandes, 2007. – ISBN 978-3-930429-67-7. – zgl. Dissertation an der Universität des Saarlandes
- [BR97] BILGIC, T. ; ROCK, D.: Product Data Management Systems: State-of-the-Art and the Future. In: ASME (Hrsg.): *Proceedings of DETC'97*. S. 1-7. Sacramento, 14.-17.9.1997
- [BR98a] BEUTELSPACHER, A. ; ROSENBAUM, U.: *Projective Geometry: From Foundations to Applications*. Cambridge : Cambridge University Press, 1998. – ISBN 0-521-48277-1
- [BR98b] BRÜDERLIN, B. (Hrsg.) ; ROLLER, D. (Hrsg.): *Geometric Constraint Solving and Applications*. Berlin : Springer Verlag, 1998. – ISBN 3-540-64416-4
- [Bra09] BRASS, E.: *Konstruieren mit CATIA V5 – Methodik der parametrisch-assoziativen Flächenmodellierung*. 4., aktualisierte und erweiterte Auflage. München : Hanser Verlag, 2009. – ISBN 978-3-446-41378-8
- [Bra10] BRAUCHART, G.: *Entwicklung von Konstruktionsmethoden zur semiautomatischen Generierung von Karosseriebauteilen in der Konzeptphase*. 2010. – Diplomarbeit an der TU Graz
- [Bri06] BRILL, M.: *Parametrische Konstruktion mit CATIA V5 – Methoden und Strategien für den Fahrzeugbau*. München : Hanser Verlag, 2006. – ISBN 978-3-446-40705-3
- [Bro10] BROCKMEYER, H.: *Rechnergestützte Methoden zur frühzeitigen Produktbeeinflussung und Produktabsicherung im Karosseriebau*. Aachen : Shaker Verlag, 2010. – ISBN 978-3-8322-9404-5. – zgl. Dissertation an Helmut-Schmidt-Universität Hamburg
- [BS07] BERGHOLZ, W. ; SACHSE, P.: Ergonomische Gestaltung von Mustervorlagen in vorstrukturierten Startmodellen bei der CAD-Konstruktion. In: RÖTTING, M. (Hrsg.) ; WOZNY, G. (Hrsg.) ; KLOSTERMANN, A. (Hrsg.) ; HUSS, J. (Hrsg.): *Prospektive Gestaltung von Mensch-Technik-Interaktion*. S. 495-500. Düsseldorf : VDI Verlag, 2007 (VDI Fortschritt-Berichte Reihe 22: Mensch-Maschine-Systeme, Bd. 25). – ISBN 978-3-18-302522-0

- [Bur08] BURR, H.: *Informationsmanagement an der Schnittstelle zwischen Entwicklung und Produktionsplanung im Karosserierohbau*. Saarbrücken : Universität des Saarlandes, 2008. – ISBN 978-3-930429-73-8. – zgl. Dissertation an der Universität des Saarlandes
- [BZ01] BATHE, K. J. ; ZIMMERMANN, P.: *Finite-Elemente-Methoden*. 2., vollständig neu bearbeitete und erweiterte Auflage. Berlin : Springer Verlag, 2001. – ISBN 978-3-540-66806-0
- [CKMH12] CHEON, S.-U. ; KIM, B. C. ; MUN, D. ; HAN, S.: A procedural method to exchange editable 3D data from a free-hand 2D sketch modeling system into 3D mechanical CAD systems. In: *Computer-Aided Design* 44 (2012), Nr. 2, S. 123–131. – ISSN 0010-4485
- [CM07] ČATIĆ, A. ; MALMQVIST, J.: Towards Integration of KBE and PLM. In: BOCQUET, J.-C. (Hrsg.): *Proceedings of ICED 2007, the 16th International Conference on Engineering Design*. Paris, 28.-31.8. 2007. – ISBN 1-904670-02-4
- [CS57] COWPER, G. R. ; SYMONDS, P. S.: *Strain-hardening and strain-rate effects in the impact loading of cantilever beams*. Providence, 1957. – Technical Report No. 28 from Brown University to the Office of Naval Research under Contract Nonr-562
- [Deu06] DEUSCHL, M.: *Gestaltung eines Prüffelds für die Fahrwerksentwicklung unter Berücksichtigung der virtuellen Produktentwicklung*. 2006. – Dissertation an der TU München
- [DGHK04] DANNENBERG, J. ; GEHR, F. ; HELLINGRATH, B. ; KLEINHANS, C.: Future Automotive Industry Structure (FAST) 2015 – die neue Arbeitsteilung in der Automobilindustrie. In: MERCER MANAGEMENT CONSULTING (Hrsg.) ; FRAUNHOFER GESELLSCHAFT (Hrsg.): *Materialien zur Automobilindustrie* Bd. 32. Frankfurt am Main : VDA, 2004
- [DO95] DETER, T. ; OERTEL, C.: Einsatz eines Entwurfssystems beim Entwicklungsprozeß des Automobils. In: *Automobiltechnische Zeitschrift ATZ* 97 (1995), Nr. 10, S. 652–657. – ISSN 0001-2785
- [EKL07] EHRENSPIEL, K. ; KIEWERT, A. ; LINDEMANN, U.: *Kostengünstig Konstruieren – Kostenmanagement bei der integrierten Produktentwicklung*. 6., überarbeitete und korrigierte Auflage. Berlin : Springer Verlag, 2007. – ISBN 978-3-540-74222-7
- [ES09] EIGNER, M. ; STELZER, R.: *Product Lifecycle Management – Ein Leitfaden für Product Development und Life Cycle Management*. 2., neu bearbeitete Auflage. Berlin : Springer Verlag, 2009. – ISBN 978-3-540-44373-5
- [Far02] FARIN, G.: A History of Curves and Surfaces in CAGD. In: FARIN, G. (Hrsg.) ; HOSCHEK, J. (Hrsg.) ; KIM, M.-S. (Hrsg.): *Handbook of Computer Aided Geometric Design*. S. 1-21. Amsterdam : North-Holland Publishing, 2002. – ISBN 978-0-44-451104-1

- [FGSS90] FORKEL, M. ; GÖBLER, T. ; SPECHT, D. ; SPUR, G.: Wissensbasierte Unterstützung der Konzeptphase des Konstruierens. In: *VDI Berichte* 861 (1990), Nr. 3, S. 163–178. ISBN 3–18–093861–0
- [FHH11] FRENER, G. ; HIRZ, M. ; HARRICH, A.: *Development of calculation methods for conceptual layout of automotive front and rear end crash systems.* – (Poster) – ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Washington, 28.-31.8.2011
- [For03] FORSEN, J.: *Ein systemtechnischer Ansatz zur methodisch parametrisch-assoziativen Konstruktion am Beispiel von Karosseriebauteilen.* Aachen : Shaker Verlag, 2003. – ISBN 978–3–8322–1862–9. – zgl. Dissertation an der Universität Duisburg-Essen
- [Fra08] FRANZEN, O.: Designbetreuung und funktionale Auslegung einer Fahrzeugtür mittels parametrisch assoziativer Modelle in CATIA V5. In: TECKLENBURG, G. (Hrsg.): *Die digitale Produktentwicklung* Bd. 1, S.192-209. Renningen : Expert Verlag, 2008. – ISBN 978–3–8169–2776–1
- [Fre12] FRENER, G.: *Erstellung eines Berechnungsmodells für die Vorauslegung von Vorderwagen-Crashsystemen.* 2012. – Diplomarbeit an der TU Graz
- [Fri08] FRIEDRICH, R.: Einsatz von V5-Knowledgeware in der Produkt-Entwicklung. In: TECKLENBURG, G. (Hrsg.): *Die digitale Produktentwicklung* Bd. 1, S.45-53. Renningen : Expert Verlag, 2008. – ISBN 978–3–8169–2776–1
- [FTHH12] FRENER, G. ; THUM, K. ; HIRZ, M. ; HARRICH, A.: An application of enhanced 3D-CAD methods with integrated geometry creation algorithms for PVC-seams in automotive body in white design. In: *3D Research* 3 (2012), Nr. 2, S. 1–9. – ISSN 2092–6731
- [GA90] GRABOWSKI, H. ; ANDERL, R.: *Produktdatenaustausch und CAD-Normteile: für Konstruktionsleiter, Normstellenleiter, CAD-Projektleiter, DV-Leiter.* Ehningen bei Böblingen : Expert Verlag, 1990. – ISBN 3–8169–0606–0
- [Ges01] GESSNER, K.: *Package-Features für die Kommunikation in den frühen Phasen der Automobilentwicklung.* Stuttgart : IRB Verlag, 2001. – ISBN 978–3–8167–5636–1. – zgl. Dissertation an der TU Berlin
- [Gfr04] GFRERRER, A.: *Kinematik und Robotik.* 2004. – Vorlesungskriptum an der TU Graz
- [Gfr11] GFRERRER, A.: Computational line geometry as a tool for solving engineering problems. In: *Proceedings of the 1st International Workshop on Line Geometry and Kinematics (IW-LGK-11).* Paphos : Technische Universität Dresden, 26.-30.4.2011

- [GLH⁺10] GFRERRER, A. ; LANG, J. ; HIRZ, M. ; HARRICH, A. ; MAYR, J. ; HASELWANTER, A.: *Verfahren und Vorrichtung zum Ermitteln einer Sollbewegung, zum Bewegen und zum Modifizieren eines verfahrbaren Bauteils*. 2010. – Patent DE 10 2010 003 882.2
- [GLH⁺11] GFRERRER, A. ; LANG, J. ; HARRICH, A. ; HIRZ, M. ; MAYR, J.: Car side window kinematics. In: *CAD Computer Aided Design* 43 (2011), Nr. 4, S. 410–416. – ISSN 0010–4485
- [GN06] GRABNER, J. ; NOTHHAFT, R.: *Konstruieren von PKW - Karosserien*. 3., erweiterte Auflage. Berlin : Springer Verlag, 2006. – ISBN 978–3–540–23884–4
- [GNR⁺08] GÜLICH, S. ; NEZEL, A. ; RING, S. ; SCHOBERT, T. ; WEIGL, W.: Das Technikmodell in der Prozesskette einer modernen Karosserieauslegung. In: *VDI Berichte* 2031 (2008), S. 485–500. ISBN 978–3–1809–2031–3
- [HAR06] HAASIS, S. ; ARNDT, H. ; REHNER, H.-P.: CATIA V5 Templates zur Umsetzung von Standardkonzepten. In: VIEWEG TECHNOLOGY FORUM (Hrsg.): *Karosseriebautage Hamburg*. S. 159-167. Hamburg : Vieweg Verlag, 23.-25.5.2006
- [Has05] HASLAUER, R.: *CATIA V5 - Konstruktionsprozesse in der Praxis*. München : Hanser Verlag, 2005. – ISBN 978–3–446–22970–9
- [Has06] HASLAUER, R.: Der Entwicklungsprozess Karosserie in der Konzept- und Serienphase. In: *Automobiltechnische Zeitschrift* 108 (2006), Nr. 12, S. 1082–1085. – ISSN 0001–2785
- [Her07] HERFELD, U.: *Matrix-basierte Verknüpfung von Komponenten und Funktionen zur Integration von Konstruktion und numerischer Simulation*. München : Dr. Hut Verlag, 2007. – ISBN 978–3–89963–545–4. – zgl. Dissertation an der TU München
- [HGL08] HIRZ, M. ; GÖBER, T. ; LANG, M.: Integrated 3D-CAD Design Strategies in Vehicle- and Engine Development Processes. In: *8. Internationales Stuttgarter Symposium - Automobil- und Motorentechnik*. S. 453-470. Stuttgart : Vieweg Verlag, 2008. – ISBN 978–3–8348–0511–9
- [HHR11] HIRZ, M. ; HARRICH, A. ; ROSSBACHER, P.: Advanced Computer Aided Design Methods for Integrated Virtual Product Development Processes. In: *Computer-Aided Design and Applications* 8 (2011), Nr. 6, S. 901–913. – ISSN 1686–4360
- [Hil09] HILMANN, J.: *On the development of a process chain for structural optimization in vehicle passive safety*. 2009. – Dissertation an der TU Berlin
- [Hir11] HIRZ, M.: *Advanced Computer Aided Design in Conceptual Automotive Development*. 2011. – Habilitation an der TU Graz

- [HM08] HÄNSCHKE, A. ; MAREK, C.: Interaktionen und Konflikte der Fahrzeugattribute Design, Komfort und Ergonomie mit mechanischen Anforderungen an die Karosserie – speziell zur Dynamik und passiven Sicherheit. In: *VDI Wissensforum Systemintegration in der Karosserietechnik*. S. 133-144. Baden-Baden : VDI Verlag, 2008. – ISBN 978-3-98-12624-4-5
- [HMF07] HILFRICH, E. ; MAAS, J. ; FLÖTH, T.: Türen-Benchmark – Vergleich und Optimierung von Fahrzeugtüren. In: *Automobiltechnische Zeitschrift ATZ* 109 (2007), Nr. 06, S. 522-529. – ISSN 0001-2785
- [HMH⁺10] HARRICH, A. ; MAYR, J. ; HIRZ, M. ; ROSSBACHER, P. ; LANG, J. ; GFRERRER, A.: CAD-based synthesis of a window lifter mechanism. In: *SAE Technical Paper Series* (2010), Nr. 2010-01-0009. – ISSN 1048-7191
- [Hän90] HÄNSCHKE, A.: *Ein Strukturmodell zur rechnergestützten Fahrzeugentwicklung*. 1990. – Dissertation an der TU Berlin
- [Hof05] HOFFMANN, C.: Constraint-based CAD. In: *Journal of Computing and Information Science in Engineering* 5 (2005), Nr. 3, S. 182-197. – ISSN 1530-9827
- [HSHL10] HARRICH, A. ; SINZ, W. ; HIRZ, M. ; LORENZER, M.: CAD-basiertes Werkzeug zur Vorauslegung der Crashstruktur eines Frontendmoduls. In: *VDI Berichte* 2107 (2010), S. 585-600. ISBN 978-3-18-092107-5
- [HVA08] HAASIS, S. ; VOGEL, U. ; ARNDT, H.: Neuausrichtung der Fahrzeugentwicklung – Ganzheitliches Lastenheftmanagement und nächste Generation der CAx-basierten Produktentwicklung bei Mercedes-Benz Cars. In: *VDI Wissensforum Systemintegration in der Karosserietechnik*. S. 115-132. Baden-Baden : VDI Verlag, 2008. – ISBN 978-3-98-12624-4-5
- [II10] IZYDORCZYK, J. ; IZYDORCZYK, M.: Microprocessor scaling: What limits will hold? In: *Computer* 43 (2010), Nr. 8, S. 20-26. – ISSN 0018-9162
- [Int78] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO): *Road vehicles; Three-dimensional reference system and fiducial marks; Definitions*. 1978. – ISO 4130
- [Int94] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO): *Industrial automation systems and integration - Product data representation and exchange - Part 1: Overview and fundamental principles*. 1994. – ISO 10303-1
- [Int12] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO): *Industrial automation systems and integration – JT file format specification for 3D visualization*. 2012. – ISO 14306
- [Jun05] JUNGKUNZ, R. M.: *PDM-basierte Überwachung komplexer Entwicklungsprojekte*. München : Utz Verlag, 2005. – ISBN 978-3-8316-0515-6. – zgl. Dissertation an der TU München

- [KB10] KLEM, S. ; BRILL, M.: OEM-übergreifende Standardisierung von Konstruktionsmethoden am Beispiel des OEM-Lieferantenmodells für CATIA V5 R19. In: TECKLENBURG, G. (Hrsg.): *Die digitale Produktentwicklung* Bd. 2. S. 64-69. Renningen : Expert Verlag, 2010. – ISBN 978-3-8169-2961-1
- [Kec83] KECMAN, D.: Bending collapse of rectangular and square section tubes. In: *International Journal of Mechanical Sciences* 25 (1983), Nr. 9/10, S. 623-636. – ISSN 0020-7403
- [Kir08] KIRCHGÄSSNER, M.: CATIA V5 im Jahr 2007: Vision und Realität in Süddeutschland. In: TECKLENBURG, G. (Hrsg.): *Die digitale Produktentwicklung* Bd. 1. S. 15-29. Renningen : Expert Verlag, 2008. – ISBN 978-3-8169-2776-1
- [Kle10] KLEMMSTEIN, T.: Side Door Modeler: Effektive Gestaltung des Konstruktionsablaufs der Türentwicklung unter Anwendung parametrischer Bausteine. In: TECKLENBURG, G. (Hrsg.): *Die digitale Produktentwicklung* Bd. 2. S. 113-121. Renningen : Expert Verlag, 2010. – ISBN 978-3-8169-2961-1
- [Klu07] KLUG, L.: *Methodischer Einsatz von parametrischen Prototypen in der Produktentwicklung*. Aachen : Shaker Verlag, 2007. – ISBN 978-3-8322-5927-3. – Dissertation an der TU Darmstadt
- [Krö02] KRÖGER, M.: *Methodische Auslegung und Erprobung von Fahrzeug-Crashstrukturen*. Düsseldorf : VDI Verlag, 2002. – ISBN 978-3-18-349612-9. – zgl. Dissertation an der Universität Hannover
- [Kri11] KRISH, S.: A practical generative design method. In: *Computer-Aided Design* 43 (2011), 1, Nr. 1, S. 88-100. – ISSN 0010-4485
- [KS95] KRAUSE, F.-L. ; SCHLINGHEIDER, J.: Development and design with knowledge-based software tools - an overview. In: *Expert Systems with Applications* 8 (1995), Nr. 2, S. 233-248. – ISSN 0957-4174
- [KSV08] KATZENBACH, A. ; SCHULTZE, H. ; VETTERMANN, S.: Kooperationen in der Automobilindustrie - Status und Perspektiven. In: GESELLSCHAFT FÜR FERTIGUNGSTECHNIK (Hrsg.): *Fertigungstechnisches Kolloquium (FTK)*. Stuttgart, 10.-11.9.2008. – ISBN 978-3-00-025623-3
- [Las06] LASEK, L.: *Verfahren zur initialen Crashauslegung in der frühen Phase der Fahrzeugentwicklung*. Düsseldorf : VDI Verlag, 2006. – ISBN 3-18-361712-9. – zgl. Dissertation an der TU Berlin
- [Leh89] LEHMANN, C.: *Wissensbasierte Unterstützung von Konstruktionsprozessen*. München : Hanser Verlag, 1989. – ISBN 3-446-15847-2. – zgl. Dissertation an der TU Berlin
- [Lei09] LEINWEBER, S.: Effiziente Produktentwicklung durch Standardisierung der CA-Methodiken am Beispiel der Fahrzeugtüre. In: *VDI Berichte* 2064 (2009), S. 39-48. ISBN 978-3-18-092064-1

- [Lie04] LIESE, H.: *Wissensbasierte 3D-CAD Repräsentation*. Aachen : Shaker Verlag, 2004. – ISBN 3–8322–2363. – zgl. Dissertation an der Universität Darmstadt
- [Lin06] LINDEMANN, U.: *Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden*. 2., bearbeitete Auflage. Berlin : Springer Verlag, 2006. – ISBN 978–3–540–37435–0
- [Man08a] MANTWILL, F. (Hrsg.): *1. Fachkonferenz CAD-Produktdaten „top secret“!?* München : VDI, 2.-3.12.2008
- [Man08b] MANTWILL, F.: Automatisch Konstruieren: ja oder lieber doch nicht?! In: MANTWILL, F. (Hrsg.): *Tagungsband der 1. Fachkonferenz CAD-Produktdaten „top secret“!?* München : VDI, 2.-3.12.2008
- [Mar03] MARKWORTH, R.: *Entwicklungsbegleitendes Digital Mock-Up im Automobilbau*. Aachen : Shaker Verlag, 2003. – ISBN 3–8322–1171–3. – zgl. Dissertation an der TU Berlin
- [May09] MAYR, J.: *Datenbankgestützte parametrisierte Geometrieerzeugung und Methodenentwicklung am Beispiel eines PKW-Türinnenblechs*. 2009. – Diplomarbeit an der TU Graz
- [Mba08] MBANG SAMA, A.: *Holistic Integration of Product, Process and Resources Integration in the Automotive Industry using the Example of Car Body Design and Production*. Karlsruhe : Universitätsverlag Karlsruhe, 2008. – ISBN 978–3–86644–243–6. – zgl. Dissertation an der Universität Karlsruhe
- [MBW⁺11] MARCHENKO, M. ; BEHRENS, B. A. ; WROBEL, G. ; SCHEFFLER, R. ; PLESSOW, M.: A New Method of Visualization and Documentation of Parametric Information of 3D CAD Models. In: *Computer-Aided Design and Applications* 8 (2011), Nr. 3, S. 435–448. – ISSN 1686–4360
- [Men99] MENDGEN, R.: *Methodische Vorgehensweise zur Modellierung in parametrischen und featurebasierten 3D-CAD-Systemen*. Aachen : Shaker Verlag, 1999. – ISBN 3–8265–4886–8. – zgl. Dissertation an der TU Darmstadt
- [Mey07] MEYWERK, M.: *CAE-Methoden in der Fahrzeugtechnik*. Berlin : Springer Verlag, 2007. – ISBN 978–3–540–49866–7
- [Mol06] MOLLICK, E.: Establishing Moore’s law. In: *Annals of the History of Computing* 28 (2006), Nr. 3, S. 62–75. – ISSN 1058–6180
- [Nie88] NIEMIERSKI, S.: *Parametergesteuerte Karosserie-Generierung im PKW-Vorentwurf*. 1988. – Dissertation an der TU Berlin
- [Nie12] NIEDERKOFLEDER, H.: *Analyse radselektiv eingreifender Fahrdynamikregelsysteme für die Anwendung in elektromechanischen Corner-Modulen*. Graz : Verlag der TU Graz, 2012. – ISBN 978–3–85125–220–0. – zgl. Dissertation an der TU Graz

- [OS04] ODEHNAL, B. ; STACHEL, H.: The upper talocalcanean join. In: *Geometry preprint series*, TU Wien, 2004. – Technical Report 127
- [PEK08] PEKERMAN, D. ; ELBER, G. ; KIM, M.: Self-intersection detection and elimination in freeform curves and surfaces. In: *Computer-Aided Design* 40 (2008), 2, Nr. 2, S. 150–159. – ISSN 0010–4485
- [PHOW04] POTTMANN, H. ; HOFER, M. ; ODEHNAL, B. ; WALLNER, J.: Line geometry for 3D shape understanding and reconstruction. In: PAJDLA, T. (Hrsg.) ; MATAS, J. (Hrsg.): *Lecture Notes in Computer Science* Bd. 3021. S. 297–309. Berlin : Springer Verlag, 2004 (Computer Vision - ECCV 2004). – ISBN 978–3–540–21984–2
- [PLH⁺05] POTTMANN, H. ; LEOPOLDSEDER, S. ; HOFER, M. ; STEINER, T. ; WANG, W.: Industrial geometry: Recent advances and applications in CAD. In: *Computer-Aided Design* 37 (2005), Nr. 7, S. 751–766. – ISSN 0010–4485
- [Pot08] POTTHOFF, J.: CAVA: Fahrzeugauslegung mit CATIA V5 unter Berücksichtigung gesetzlicher Vorgaben und Richtlinien. In: TECKLENBURG, G. (Hrsg.): *Die digitale Produktentwicklung* Bd. 1. S. 30–41. Renningen : Expert Verlag, 2008. – ISBN 978–3–8169–2776–1
- [PR98] POTTMANN, H. ; RANDRUP, T.: Rotational and Helical Surface Approximation for Reverse Engineering. In: *Computing* 60 (1998), Nr. 4, S. 307–322. – ISSN 0010–485X
- [Pri06] PRIEUR, M.: *Functional Elements and Engineering Template-based Product Development Process*. Karlsruhe : Universitätsverlag Karlsruhe, 2006. – ISBN 978–3–86644–033–3. – zgl. Dissertation an der Universität Karlsruhe
- [PW01] POTTMANN, H. ; WALLNER, J.: *Computational Line Geometry*. Berlin : Springer Verlag, 2001. – ISBN 3–540–42058–4
- [QRP⁺10] QUINTANA, V. ; RIVEST, L. ; PELLERIN, R. ; VENNE, F. ; KHEDDOUCI, F.: Will Model-based Definition replace engineering drawings throughout the product life-cycle? A global perspective from aerospace industry. In: *Computers in Industry* 61 (2010), Nr. 5, S. 497 – 508. – ISSN 0166–3615
- [Ras98] RASENACK, W.: *Parametervariation als Hilfsmittel bei der Entwicklung des Fahrzeug-Package*. Berlin : Offset-Druckerei Gerhard Weinert, 1998. – ISBN 3–7983–1775–5. – zgl. Dissertation an der TU Berlin
- [Ray09] RAY, J.: Predicting the future of CAD. In: *Desktop Engineering* 14 (2009), Nr. 6. – ISSN 1085–0422
- [RCAR10] RESEARCH COUNCIL FOR AUTOMOBILE REPAIRS: *RCAR Bumper Test*. 2010. – Versuchsanweisung
- [RCAR11] RESEARCH COUNCIL FOR AUTOMOBILE REPAIRS: *RCAR Low-speed structural crash test protocol*. 2011. – Versuchsanweisung

- [RG08] REUTER, M. ; GESCHKE, T.: Bereichsmodelle in der Konzeptphase. In: TECKLENBURG, G. (Hrsg.): *Die digitale Produktentwicklung* Bd. 1. S. 71-80. Renningen : Expert Verlag, 2008. – ISBN 978-3-8169-2776-1
- [RHH⁺09] ROSSBACHER, P. ; HIRZ, M. ; HARRICH, A. ; DIETRICH, W. ; THEISS, N.: The Potential of 3D-CAD Based Process – Optimization in the Automotive Concept Phase. In: *SAE International Journal of Materials and Manufacturing* 2 (2009), Nr. 1, S. 250–257. – ISSN 1946–3979
- [San07] SANDER, A.: *Potentiale eines multidisziplinären Fahrzeugauslegungssystems*. Düsseldorf : VDI Verlag, 2007. – ISBN 978-3-18-366312-5. – zgl. Dissertation an der TU Berlin
- [Sch02] SCHICHEL, M.: *Produktdatenmodellierung in der Praxis*. München : Hanser Verlag, 2002. – ISBN 3-446-21857-2
- [Sch09] SCHEUFLER, A.: CAD noch lange nicht am Ende. In: *KEM – Konstruktion, Entwicklung, Management* (2009), Nr. 3, S. 52–53. – ISSN 1612–7226
- [Sch10a] SCHADENBAUER, P.: *Die Konzeptphase in der Türenentwicklung*. 2010. – Diplomarbeit an der TU Graz
- [Sch10b] SCHREIER, P.: Closing the CAD/CAE gap. In: *Scientific Computing World* 109 (2010), Dezember/Jänner, S. 26. – ISSN 1356–7853
- [Sch10c] SCHULZE, J.: Siemens entscheidet sich für NX von Siemens PLM. In: *Konstruktionspraxis* (2010), Nr. 12, S. 8. – ISSN 0937–4167
- [SFBP07] SCHULTE-FRANKENFELD, N. ; BRASS, M. ; PIECK, A.: Methoden und Prozesse zur Kostensenkung – Ein Status der Wandlungen im Fahrzeugentwicklungsprozess durch CAE-Methoden. In: LS-DYNA (Hrsg.): *6. LS-Dyna Anwenderforum*, 11.-12.10.2007. – Frankenthal
- [SFJ06] SIR, Z. ; FEICHTINGER, R. ; JÜTTLER, B.: Approximating Curves and their Offsets using Biarcs and Pythagorean Hodograph Quintics. In: *CAD Computer Aided Design* 38 (2006), Nr. 6, S. 608–618. – ISSN 0010–4485
- [SH14] STADLER, S. ; HIRZ, M.: An Application of Enhanced Knowledge-Based Design in Automotive Seat Development. In: *Computer-Aided Design and Applications* 11 (2014), Nr. 3, S. 335–345. – ISSN 1686–4360
- [SK97] SPUR, G. ; KRAUSE, F.: *Das virtuelle Produkt - Management der CAD-Technik*. München : Hanser Verlag, 1997. – ISBN 978-3-446-19176-1
- [Ska07] SKARKA, W.: Application of MOKA methodology in generative model creation using CATIA. In: *Engineering Applications of Artificial Intelligence* 20 (2007), 8, Nr. 5, S. 677–690. – ISSN 0952–1976

- [SM01] STOKES, M. (Hrsg.) ; MOKA CONSORTIUM (Hrsg.): *Managing Engineering Knowledge: MOKA Methodology for Knowledge Based Engineering Applications*. Chichester : Wiley-Blackwell, 2001. – ISBN 978–1–86058–295–0
- [Soh92] SOHLENIUS, G.: Concurrent Engineering. In: *CIRP Annals – Manufacturing Technology* 41 (1992), Nr. 2, S. 645–655. – ISSN 0007–8506
- [SON06] SAPUAN, S. M. ; OSMAN, M. R. ; NUKMAN, Y.: State of the art of the concurrent engineering technique in the automotive industry. In: *Journal of Engineering Design* 17 (2006), Nr. 2, S. 143–157. – ISSN 0954–4828
- [SSS12] STEIN, M. ; SCHWANITZ, P. ; SANKARASUBRAMANIAN, H.: Unified parametric car model – A simplified model for frontal crash safety. In: *11. LS-Dyna Forum*, 9.-10.10.2012. – Ulm
- [Ste07] STEKOLSCHIK, A.: *Ein Beitrag zum ganzheitlichen Qualitätsmanagement von CAD-Modellen in der Produktentstehung*. Aachen : Shaker Verlag, 2007. – ISBN 978–3–8322–6669–1. – zgl. Dissertation an der Ruhr-Universität Bochum
- [Str06] STROHMEIER, O.: *Integration von Wissensmodulen in den virtuellen Produktentwicklungsprozess*. Aachen : Shaker Verlag, 2006. – ISBN 978–3–8322–5128–4. – zgl. Dissertation an der Universität Duisburg-Essen
- [Sut63] SUTHERLAND, I. E.: Sketchpad: a man-machine graphical communication system. In: *Proceedings of the spring joint computer conference – AFIPS '63*. New York : ACM, 21.-23.5.1963, S. 329–346
- [SW11] SENDLER, U. ; WAWER, V.: *CAD und PDM - Prozessoptimierung durch Integration*. 3., überarbeitete und erweiterte Auflage. München : Hanser Verlag, 2011. – ISBN 978–3–446–42585–9
- [Tec11] TECKLENBURG, G.: *Design of Automotive Body Assemblies with Distributed Tasks under Support of Parametric Associative Design*. 2011. – Dissertation an der University of Hertfordshire
- [TM10] TORNINCASA, S. ; MONACO, S. D.: The future and the evolution of CAD. In: EKINOVIČ, S. (Hrsg.) ; UCTUG, Y. (Hrsg.) ; CALVET, J. V. (Hrsg.): *Proceedings of the 14th International Research/Expert Conference: Trends in the Development of Machinery and Associated Technology – TMT 2010*. Mediterranean Cruise, 11.-18.9.2010. – ISSN 1840–4944
- [UNECE80] UNITED NATIONS ECONOMIC COMMISSION FOR EUROPE (UNECE): *Uniform provisions concerning the approval of vehicles with regard to their front and rear protective devices (bumpers, ect.)*. 1980. – ECE Regulation: R42
- [USDOT90] U.S. DEPARTMENT OF TRANSPORTATION, NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION (NHTSA): *Laboratory Test Procedure for Regulation Part 581- Bumper Standard*. 1990. – Federal Motor Vehicle Safety Standards (FMVSS) Part 581

- [VBGvC12] VERHAGEN, W. J. C. ; BERMELL-GARCIA, P. ; VAN DIJK, R. E. C. ; CURRAN, R.: A critical review of Knowledge-Based Engineering: An identification of research challenges. In: *Advanced Engineering Informatics* 26 (2012), Nr. 1, S. 5–15. – ISSN 1474–0346
- [VDI93] VEREIN DEUTSCHER INGENIEURE (VDI): *Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte*. Düsseldorf, 1993. – VDI-Richtlinie: VDI 2221
- [VDI03] VEREIN DEUTSCHER INGENIEURE (VDI): *Informationsverarbeitung in der Produktentwicklung – Feature-Technologie*. Düsseldorf, 2003. – VDI-Richtlinie: VDI 2218
- [VDI06] VEREIN DEUTSCHER INGENIEURE (VDI): *Digitale Fabrik*. Düsseldorf, 2006. – VDI-Richtlinie: VDI 4499
- [VDI09] VEREIN DEUTSCHER INGENIEURE (VDI): *3-D-Produktmodellierung – Technische und organisatorische Voraussetzungen – Verfahren, Werkzeuge und Anwendungen – Wirtschaftlicher Einsatz in der Praxis*. Düsseldorf, 2009. – VDI-Richtlinie: VDI 2209
- [Vic06] VICKERS, S.: Effiziente Produktentwicklung durch intelligentes Wissensmanagement. In: *CAD-CAM Report* (2006), Nr. 3, S. 32–37. – ISSN 0930–7117
- [Vie05] VIELHABER, M.: *Assembly Oriented Design – Zusammenbauorientiertes Konstruieren im Produktentstehungsprozess der Automobilindustrie am besonderen Beispiel des Karosserierohbaus*. Saarbrücken : Universität des Saarlandes, 2005. – ISBN 3–930429–63–2. – zgl. Dissertation an der Universität des Saarlandes
- [VWBZ09] VAJNA, S. ; WEBER, C. ; BLEY, H. ; ZEMAN, K.: *CAx für Ingenieure – eine praxisbezogene Einführung*. 2., völlig neu bearbeitete Auflage. Berlin : Springer Verlag, 2009. – ISBN 978–3–540–36038–4
- [Wan02] WANG, G. G.: Definition and review of virtual prototyping. In: *Journal of Computing and Information Science in Engineering* 2 (2002), Nr. 3, S. 232–236. – ISSN 1530–9827
- [WdM98] WILLAERT, S. S. A. ; DE GRAAF, R. ; MINDERHOUD, S.: Collaborative engineering: A case study of Concurrent Engineering in a wider context. In: *Journal of Engineering and Technology Management* 15 (1998), 3, Nr. 1, S. 87–109. – ISSN 0923–4748
- [Wie05] WIERMEIER, B.: *Referenzmodelle für die simultane Entwicklung und Produktionsplanung eines Automobils*. 2005. – Dissertation an der TU Graz
- [Woz92] WOZNY, M. J.: Research Trends in the U.S. for the Next Generation CAD Systems. In: KRAUSE, F.-L. (Hrsg.) ; RULAND, D. (Hrsg.) ; JANSEN, H. (Hrsg.): *CAD '92: Neue Konzepte zur Realisierung anwendungsorientierter CAD-Systeme*. S. 30-65. Berlin : Springer, 1992. – ISBN 978–3–540–55494–3

- [WWD03] WEBER, C. ; WERNER, H. ; DEUBEL, T.: A different view on Product Data Management/Product Life-Cycle Management and its future potentials. In: *Journal of Engineering Design* 14 (2003), Nr. 4, S. 447–464. – ISSN 0954–4828
- [Zie75] ZIENKIEWICZ, O. C.: *Methode der finiten Elemente*. München : Hanser Verlag, 1975. – ISBN 3–446–12046–7