

Norman Lupa

**Einsatz wissensbasierter Features
für die automatische Konfiguration
von Produktkomponenten**



Cuvillier Verlag Göttingen
Internationaler wissenschaftlicher Fachverlag

**Einsatz wissensbasierter Features für die
automatische Konfiguration von Produktkomponenten**

Von der Fakultät für Ingenieurwissenschaften, Abteilung Maschinenbau der
Universität Duisburg-Essen
zur Erlangung des akademischen Grades

DOKTOR-INGENIEUR

genehmigte Dissertation

von

Norman Lupa

aus

Wesel

Referent: Prof. Dr.-Ing. Peter Köhler
Korreferent: Prof. Dr.-Ing. Frank Rieg
Tag der mündlichen Prüfung: 9. Dezember 2009

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

1. Aufl. - Göttingen : Cuvillier, 2009

Zugl.: Duisburg-Essen, Univ., Diss., 2009

978-3-86955-192-0

© CUVILLIER VERLAG, Göttingen 2009

Nonnenstieg 8, 37075 Göttingen

Telefon: 0551-54724-0

Telefax: 0551-54724-21

www.cuvillier.de

Alle Rechte vorbehalten. Ohne ausdrückliche Genehmigung des Verlages ist es nicht gestattet, das Buch oder Teile daraus auf fotomechanischem Weg (Fotokopie, Mikrokopie) zu vervielfältigen.

1. Auflage, 2009

Gedruckt auf säurefreiem Papier

978-3-86955-192-0

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Produkt Engineering der Universität Duisburg-Essen in den Jahren von 2004 bis 2009.

Meinem Doktorvater, Herrn Prof. Dr.-Ing. Peter Köhler, danke ich für die Anregungen zu dieser Arbeit. Seine persönliche und wissenschaftliche Unterstützung haben zu einem Großteil zum Gelingen dieser Arbeit beigetragen.

Herrn Prof. Dr.-Ing. Frank Rieg von der Universität Bayreuth danke ich für die freundliche Übernahme des Korreferats und das entgegengebrachte Interesse an meiner Arbeit.

Darüber hinaus möchte ich mich bei meinen ehemaligen Kollegen am Institut für die immer kollegiale Zusammenarbeit bedanken. Mein besonderer Dank gilt hierbei Herrn Dr.-Ing. Jens Bechthold, dessen Motivation und freundschaftliche Unterstützung mich auf diesem Weg begleitet haben. Ebenso möchte ich Herrn Dipl.-Ing. Stéphane Danjou und Herrn Dr.-Ing. Sascha Dungs für die immer interessanten und anregenden persönlichen und fachlichen Diskussionen danken. Diese haben sowohl meine Zeit am Institut bereichert als auch zum Gelingen dieser Arbeit beigetragen. Ferner bedanke ich mich bei allen studentischen Hilfskräften für ihre tatkräftige Mitarbeit. Insbesondere bedanke ich mich bei Herrn cand. Ing. Marcin Humpa wegen seiner herausragenden Engagements.

Nicht zuletzt bedanke ich mich bei meiner Familie, die mich in jeder Situation selbstlos unterstützte und mir den notwendigen Rückhalt für die erfolgreiche Umsetzung dieser Arbeit gab. Mein besonderer Dank gilt hierbei meiner Frau Angela für ihre immer währende Unterstützung, ihr Verständnis und ihre Geduld.

Willich-Anrath, im Dezember 2009

Norman Lupa

Inhaltverzeichnis

1	EINLEITUNG	1
1.1	PROBLEMSCHILDERUNG	2
1.2	ZIELSETZUNG UND ABGRENZUNG DER ARBEIT	3
2	GRUNDLAGEN.....	5
2.1	KONSTRUKTIONSMETHODIK UND KONSTRUKTIONSARTEN	5
2.1.1	<i>Konstruktionsmethodik.....</i>	6
2.1.2	<i>Konstruktionsarten.....</i>	7
2.2	RECHNERGESTÜTZTES KONSTRUIEREN	8
2.2.1	<i>Datenmodelle</i>	10
2.2.1.1	Constructive Solid Geometry Model (CSG Volumenmodell).....	11
2.2.1.2	Boundary-Representation Model (Flächenbegrenzungsmodell)	12
2.2.1.3	Hybride Modelle	14
2.2.2	<i>Parametrisch-assoziative Modellierung</i>	14
2.2.3	<i>Featurebasierte Modellierung</i>	16
2.2.3.1	Begriffsdefinitionen.....	16
2.2.3.2	Die Feature-Technologie in CAD/CAM-Systemen.....	17
2.2.3.3	Die Feature-Technologie im Produktlebenszyklus.....	18
2.3	DATENBANKSYSTEME	19
2.3.1	<i>Anforderungen an Datenverwaltungssysteme.....</i>	19
2.3.2	<i>Aufbau und Struktur.....</i>	20
2.3.3	<i>Datenmodelle</i>	21
2.3.4	<i>Datenzugriff und Schnittstellen.....</i>	23
2.3.4.1	SQL	23
2.3.4.2	ODBC.....	24
2.4	PRODUKTDATENMANAGEMENT.....	25
2.4.1	<i>Grundlagen</i>	25
2.4.2	<i>Datenaustausch.....</i>	25
2.5	PRODUKTKONFIGURATION	27
2.6	WISSENSINTEGRATION IN DAS VIRTUELLE PRODUKT	28
2.6.1	<i>Künstliche Intelligenz.....</i>	29
2.6.2	<i>Wissen</i>	29
2.6.3	<i>Wissensbasierte Systeme</i>	32
2.6.4	<i>Architektur von wissensbasierten Systemen.....</i>	32
2.6.5	<i>Formen der Wissensrepräsentation</i>	34
2.6.6	<i>Inferenzbildung und Ablaufsteuerung.....</i>	38
2.6.7	<i>Wissensbasierte Konstruktionssysteme</i>	40

3	ANSÄTZE ZUR WISSENSINTEGRATION IM CAD-UMFELD	43
3.1	COMMONKADS.....	43
3.2	MOKA	44
3.3	VIVACE.....	46
3.4	RAD	47
3.5	KCM.....	48
3.6	ANSATZ VON STROHMEIER	49
3.7	ANSATZ VON DUNGS.....	50
3.8	ANSATZ VON JANITZA.....	50
3.9	ANSATZ VON LIESE	51
3.10	ANSATZ VON HAASIS	51
3.11	VDI-RICHTLINIE 2209	52
4	AUTOMATISIERTE ERZEUGUNG VON PRODUKTKOMPONENTEN.....	53
4.1	PRODUKTAUSWAHL: TURBOVERDICHTER	53
4.1.1	<i>Bauweisen</i>	54
4.1.2	<i>Betrachtete Komponenten</i>	55
4.1.2.1	Verdichterkonfiguration	55
4.1.2.2	Welle	55
4.1.2.3	Laufrad	58
4.2	PROBLEMANALYSE	60
4.2.1	<i>Umsetzung in einem parametrischen CAD-System</i>	60
4.2.2	<i>Gestaltung von Bauteilen und Baugruppen</i>	61
4.2.3	<i>Datenredundanz</i>	62
4.2.4	<i>Anreicherung mit semantischen Informationen</i>	63
4.3	ANFORDERUNGEN AN DAS GESAMTKONZEPT.....	63
4.3.1	<i>Anforderungen an die verwendeten Methoden</i>	64
4.3.1.1	Anforderungen an die Softwareentwicklung	64
4.3.1.2	Anforderungen an die Modellierungsmethodik.....	65
4.3.2	<i>Technische Anforderungen</i>	65
4.3.3	<i>Nichttechnische Anforderungen</i>	66
4.4	KONZEPTENTWICKLUNG	67
4.4.1	<i>Grundlegende Konzeption</i>	67
4.4.2	<i>Softwarekonzept</i>	69
4.4.2.1	Analyse und Auswahl möglicher Programmiersprachen.....	69
4.4.2.2	Schnittstellenarchitektur	71
4.4.2.3	Benutzeroberfläche.....	72
4.4.3	<i>Featurekonzept</i>	74
4.4.3.1	Datenbankanbindung und Informationsbereitstellung.....	75
4.4.3.2	Geometrisches Konzept.....	78

4.4.3.3	Diskussion	83
4.4.4	<i>Erzeugung der Komponententopologie</i>	85
4.4.4.1	Positionierung.....	86
4.4.4.2	Boolesche Verknüpfung	87
4.5	UMSETZUNG.....	88
4.5.1	<i>Aufbau des Konfigurationssystems</i>	89
4.5.2	<i>Aufbau der Anwenderschnittstelle</i>	91
4.5.3	<i>Aufbau der UDFs</i>	92
4.5.4	<i>Interaktion mit dem System</i>	94
4.6	WIRTSCHAFTLICHKEITSBETRACHTUNG.....	96
4.7	ZUSAMMENFASSUNG DER METHODIK	98
4.7.1	<i>Grundkonzept</i>	99
4.7.2	<i>Anforderungen an die Produkt- und Prozessstruktur</i>	100
4.7.3	<i>Anforderungen an das IT-Umfeld</i>	101
4.7.4	<i>Bemerkungen zur Übertragbarkeit von Teilaspekten</i>	102
4.7.4.1	Aufgabenstellung.....	103
4.7.4.2	Problemstellung und Übertragbarkeit von Teilaspekten des Gesamtkonzepts	104
5	ZUSAMMENFASSUNG UND AUSBLICK	109
6	LITERATURVERZEICHNIS	113
7	ABBILDUNGSVERZEICHNIS	123
8	TABELLENVERZEICHNIS	125

Abkürzungen

Abkürzung	Bedeutung
API	Application Programming Interface
B-REP	Boundary Representation
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CAM	Computer-Aided Manufacturing
CATIA	Computer-Aided Three-Dimensional Interactive Application
CFD	Computational Fluid Dynamics
CNC	Computerized Numerical Control
CORBA	Common Object Request Broker Architecture
CSG	Constructive Solid Geometry
DBMS	Datenbankmanagementsystem
DCL	Data Control Language
DDL	Data Definition Language
DFG	Deutsche Forschungsgemeinschaft
DML	Data Manipulation Language
DSDM	Dynamic Systems Development Method
ECAD	Electrical Computer-Aided Design
FEA	Finite Elemente Analyse
GRIP	Graphics Interactive Programming
IGES	Initial Graphics Exchange Specification
IT	Informationstechnologie
KBE	Knowledge-based Engineering
KCM	Knowledge Capture Methodology
KF	Knowledge Fusion

Abkürzung	Bedeutung
KI	Künstliche Intelligenz
LISP	List Programming
MCAD	Mechanical Computer-Aided Design
MIT	Massachusetts Institute of Technology
MML	MOKA Modelling Language
MOKA	Methodology and Software Tools oriented to Knowledge-based Engineering Applications
NURBS	Non-Uniform Rational B-Splines
OMG	Object Management Group
PDM	Produktdatenmanagement
PLM	Product Lifecycle Management
RAD	Rapid Application Development
SDK	Software Development Kit
SECI	Socialisation, Externalization, Combination, Internalization
SQL	Structured Query Language
STEP	Standard for the Exchange of Product Data
STEP	Standard for the Exchange of Product Data
UDF	User-Defined Feature
UML	Unified Modeling Language
VDA-FS	Verband der Automobilindustrie - Flächenschnittstelle
VIVACE	Value Improvement through a Virtual Aeronautical Collaborative Enterprise
VPE	Virtuelle Produktentwicklung
XML	Extensible Markup Language

1 Einleitung

Der zielgerichtete Einsatz der integrierten virtuellen Produktentwicklung (VPE) stellt einen entscheidenden Faktor für die Sicherung und den Ausbau der Wettbewerbsfähigkeit von Unternehmen dar. Die VPE umfasst dabei sowohl die Erzeugung als auch die Verwaltung sämtlicher produktbezogener Daten. Mit Hilfe der VPE stellen sich die Unternehmen der Vorgabe der Märkte, in immer kürzer werdenden Zykluszeiten neue Produkte mit steigender Qualität hervorbringen zu müssen, um am Markt bestehen zu können. Durch die konsequente Nutzung der durch die VPE bereitgestellten Werkzeuge auf Basis eines digitalen Master Modells können bereits in frühen Phasen der Produktentwicklung Aussagen über die Eigenschaften des zukünftigen realen Produktes getätigt werden [DuLu06].

Die Anzahl der Softwaretools, die im Rahmen der VPE genutzt werden können, ist bereits jetzt fast unüberschaubar. Neben der großen Anzahl an CAD-Systemen diverser Hersteller für unterschiedliche Branchen und Anwendungsbereiche (MCAD, ECAD, etc.) existieren diverse PDM-Systeme, die je nach Hersteller unterschiedliche Integrationstiefen für die angeschlossenen Softwarewerkzeuge bereitstellen. Die im CAD-System erzeugten Modelle dienen als Basis für weiterführende Prozessschritte. In frühen Phasen der Produktentwicklung sind dies vor allen Dingen Untersuchungen der zu erwartenden Produkteigenschaften z. B. in Hinblick auf Festigkeit, Geräuschentwicklung oder Strömungseigenschaften. Hierbei ist der Trend zu beobachten, dass die für diese Untersuchungen notwendigen Softwaresysteme immer stärker in das eigentlich nur für die Geometriemodellierung gedachte CAD-System integriert werden. Durch die immer weiter steigende Rechnerleistung sind am Arbeitsplatz des Konstrukteurs Berechnungen möglich geworden, für die noch vor kurzem ein oder mehrere Hochleistungsrechner von Nöten waren, um sie in akzeptabler Zeit durchführen zu können. Die CAD-Systeme werden damit immer mehr zu kompletten Produktentwicklungssystemen, die im Rahmen des gesamten Produktentwicklungsprozesses auf integrale Weise eingesetzt werden können.

Jedoch stellt die Anpassung und Integration aller Systeme in eine bestehende IT-Landschaft Unternehmen sowohl vor große monetäre als auch personelle Herausforderungen. Bei produzierenden Unternehmen verursachen der Betrieb, die Wartung und die Betreuung der Hard- und Software über 70 Prozent aller IT-Kosten, hinzu kommen alle Folgekosten der Systeme [AbSc04]. Trotz der immensen anfallenden Kosten werden von vielen Unternehmen die bestehenden Möglichkeiten der vorhandenen Softwarelösungen häufig nicht vollständig ausgenutzt und so wertvolles Potential für eine weitere Beschleunigung des Produktentwicklungsprozesses bei steigender Qualität der digitalen Modelle verschenkt.

1.1 Problemschilderung

Die zunehmende Globalisierung der Märkte und die damit einher gehende Intensivierung des Wettbewerbs zwischen den Unternehmen erfordert die Einführung neuer und innovativer Methoden, um die Produktkosten zu senken. Betrachtet man die Anteile der verschiedenen Unternehmensbereiche an den Kosten des endgültigen Produkts (Abbildung 1-1) so fällt auf, dass der Bereich der Konstruktion und Entwicklung den größten Teil der Kostenverantwortung im Unternehmen trägt, selbst aber die geringsten Kosten verursacht.

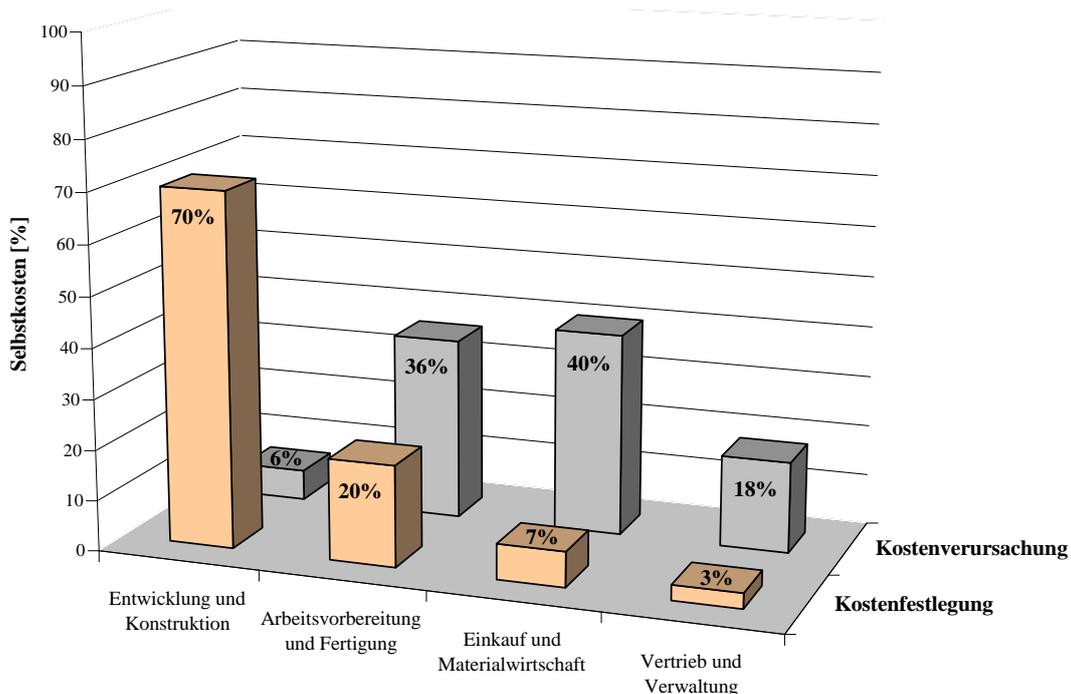


Abbildung 1-1: Kostenfestlegung und Kostenverursachung [VDI87]

Somit wird deutlich, dass die Qualität der Produktentwicklung und damit auch die Qualität der digitalen Modelle einen bedeutenden Einfluss auf die Kostenentwicklung in nachfolgenden Prozessschritten und damit auch auf die Gesamtkosten eines Produkts hat [Ste07]. Erschwerend kommt hinzu, dass die Unternehmen verstärkt auf die Kundenwünsche eingehen müssen, um am dynamischen Weltmarkt bestehen zu können. Das führt zu einer immer größer werdenden Produktvarianz im Unternehmen und damit zu einer Komplexitätserhöhung im Produktentwicklungsprozess. Viele Unternehmen begegnen diesen Problemen mit der Einführung von Baukastensystemen, um auf bewährte und geprüfte Elemente zugreifen zu können und die vielfältigen Abhängigkeiten beherrschbar zu machen. Für die Erzeugung der endgültigen Produkte auf der Basis dieser Baukastenelemente sind mittlerweile viele kommerzielle Softwaresysteme erhältlich, mit denen die Automatisierung des Konfigurationsprozesses ermöglicht wird. Die Baukastenelemente müssen jedoch nicht immer zwingend aus parametrisierten Bauteilen oder Baugruppen bestehen. Je nach Produktstruktur kann es sein, dass bestimmte Gestaltungsmodule zu einem neuen Bauteil zusammengesetzt werden müssen. Diese Aufgabe muss daher direkt im CAD-System gelöst werden, wozu meist keine vorge-

fertigte Softwarelösung existiert. Ein parametrisch aufgebautes Bauteil allein reicht hier u. U. nicht mehr aus, um die z. T. vielfältigen Gestaltänderungen umsetzen zu können. Dies trifft insbesondere auf den Fall der komplexen topologischen Änderung innerhalb eines Gestaltungsmoduls zu. Diese hohe Variabilität lässt sich mit herkömmlichen parametrischen Modellierungsmethoden nicht zufriedenstellend erreichen. Ein Trend der letzten Jahre ist die Erweiterung der parametrischen und featurebasierten CAD-Modelle um wissensbasierte Anteile, um so das CAD-Modell zum Träger des Produkt- und Prozesswissens im Produktentwicklungsprozess zu machen. Die konsequente Nutzung dieser wissensbasierten Features soll zu einer qualitativ hochwertigeren Produktentwicklung beitragen indem es das Wissen dort zugänglich und nutzbar macht wo es generiert, wiederverwendet und erweitert wird.

1.2 Zielsetzung und Abgrenzung der Arbeit

In dieser Arbeit werden die Möglichkeiten der automatisierten und wissensbasierten Erstellung von Komponenten und Baugruppen innerhalb von featurebasierten und parametrischen CAD-Systemen erläutert. Dazu wird am Beispiel der Läuferkomponenten eines Einwellenverdichters ein Konzept entwickelt, durch das der automatisierte Aufbau der CAD-Modelle unter Berücksichtigung der Anforderungen nachfolgender Prozessschritte in der Produktentwicklung ermöglicht wird. Grundlage des Konzepts ist der Aufbau eines Design Repositories auf der Basis wissensbasierter Features (Gestaltungsmodule). Eine wesentliche Eigenschaft dieser Gestaltungsmodule ist die Trennung der zugehörigen Wissensanteile. Wissen, das nicht ausschließlich für die Umsetzung der geometrischen Gestalt im CAD-System genutzt wird soll unternehmensweit und redundanzfrei zugänglich sein. Da eine erfolgreiche Konfiguration der Läuferkomponenten wesentlich von der geometrischen und informationstechnischen Stabilität der Gestaltungsmodule abhängt werden des Weiteren Methoden entwickelt, mit denen eine Erweiterung der Möglichkeiten des Geometriemodells durch die Integration wissensbasierter Anteile ermöglicht wird.

Für die eigentliche Konfiguration der Gestaltungsmodule wird ein in das CAD-System integriertes Anwendungssystem vorgestellt, dessen Aufgabe der Aufbau aller notwendigen CAD-Komponenten auf der Basis einer formalen Komponentenbeschreibung ist. Das Anwendungssystem wird derart gestaltet, dass eine Erweiterung um neue Produktkomponenten ohne Kenntnis des eigentlichen Quellcodes möglich ist. Damit soll sichergestellt werden, dass eine zukünftige Erweiterung des Systems um neue Gestaltungsmodule oder Produktkomponenten auf möglichst einfache Weise durchgeführt werden können.

Ziel dieser Arbeit ist es nicht, neue Methoden für die Produktkonfiguration zu entwickeln. Vielmehr sollen geeignete Wege aufgezeigt werden, wie das Ergebnis des Produktkonfigurationsprozesses in Form der Grobgestalt oder Konfiguration einer Komponente automatisiert in ein CAD-System übertragen werden kann, um so die Feingestalt der Komponente zu erhalten. Der Schwerpunkt der Arbeit wird hierbei auf die Konfiguration von Bauteilen gelegt, wobei jedoch auch die Konfiguration von Baugruppen mit diesen Bauteilen betrachtet wird.

2 Grundlagen

Die Bedeutung der Informationstechnologie im Produktentwicklungsprozess hat in den letzten Jahren stark zugenommen. Die Auswirkungen in Bezug auf Prozesse und Methoden zeigen sich in einer Neuorientierung in fast allen Unternehmensbereichen. Ziel der VPE ist die Erstellung eines digitalen Produktmodells. Dessen Kern bildet das 3D-CAD-Modell, aus welchem die Modelle für alle weiteren Phasen der Produktentwicklung abgeleitet werden [KrWö01]. Virtuell bezeichnet hierbei die über alle Phasen durchgängig digitale Realisierung dieses Prozesses über alle Unternehmensgrenzen und –Standorte hinweg [SpKr97].

Ziel der VPE ist die möglichst vollständige Beschreibung eines realen Produkts mit all seinen Eigenschaften als Modell im Rechner und das Management der in allen Teilprozessen erzeugten Daten sowie der Prozesse und verwendeten IT-Werkzeuge, um eine lückenlose Dokumentation des Produkts und seiner Entstehungsgeschichte zu ermöglichen. Innerhalb der VPE nehmen Produktdatenmanagementsysteme eine herausragende Stellung ein. Sie sammeln, speichern und verwalten alle produktbezogenen Dokumente, bilden mit entsprechenden Workflow-Funktionalitäten Unternehmensprozesse wie Freigabe- und Änderungswesen ab und unterstützen das abteilungsübergreifende kooperative Arbeiten auch an verteilten Standorten.

Die 3D-Produktmodellierung eröffnet eine Vielzahl neuer Möglichkeiten. Bauteile können rechnerunterstützt analysiert und deren Verhalten simuliert werden. So sind z. B. kinematische Untersuchungen im Rahmen von Kollisionsprüfungen für Ein- und Ausbauuntersuchungen durchführbar. Somit ist es möglich, bereits in frühen Phasen der Produktentwicklung potentielle Fehlerquellen zu ermitteln und zu eliminieren. Auch kann die mechanische Beanspruchbarkeit sowie das thermische und strömungsmechanische Verhalten der digitalen Modelle mit Hilfe spezieller CAD-Anwendungsmodule auf der Basis von Finite-Elemente-Methoden simuliert werden [Bec07]. Der Einsatz dieser Simulationsmethoden erlaubt es dem Produktdesigner in einer frühen Entwicklungsphase Aussagen über das spätere Verhalten des Produkts treffen zu können, womit aufwändige Iterationen vermieden werden können. Damit ergeben sich Potentiale für die frühzeitige Verifikation einer Konstruktion, auf deren Basis dann Optimierungen durchgeführt werden können.

Für die erfolgreiche Umsetzung des zu entwickelnden Konzepts ist die detaillierte Kenntnis der wichtigsten Techniken im Anwendungsgebiet der VPE notwendig. Daher werden in den folgenden Abschnitten die entsprechenden Grundlagen der virtuellen Produktentwicklung herausgearbeitet.

2.1 Konstruktionsmethodik und Konstruktionsarten

Immer komplexer werdende Produkte erfordern die Anwendung strukturierter Methoden, um den Produktentwicklungsprozess überschaubar und somit auch steuerbar zu machen. In der Vergangenheit wurden daher in der Forschung diverse Konstruktionsmethoden entwickelt, die ein Vorgehensmodell für die jeweils identifizierten Phasen des Gesamtprozesses bilden. Im folgenden Kapitel

sollen jedoch nur die Konstruktionsmethoden vorgestellt werden, die heutzutage sowohl in der Forschung als auch in der industriellen Praxis – nicht zuletzt durch die Einarbeitung in eine entsprechende Richtlinie – ihre Anwendung finden. Im darauf folgenden Kapitel werden dann die Konstruktionsarten erläutert und die entsprechenden Phasen der dargestellten Konstruktionsmethoden zugeordnet.

2.1.1 Konstruktionsmethodik

In der VDI-Richtlinie 2221 wird ein generelles und branchenübergreifendes Vorgehen für die Entwicklung und Konstruktion technischer Produkte vorgeschlagen [VDI93]. Die dokumentierten Ergebnisse eines Arbeitsschritts dienen hierbei als Eingangsgröße für den nächsten Arbeitsschritt. Ausgehend von einer bestimmten Aufgabenstellung wird eine konkrete Anforderungsliste erstellt (Pflichtenheft), welche die wesentlichen Produktmerkmale dokumentiert. Daraufhin wird das Gesamtsystem in technisch sinnvolle Teilfunktionen untergliedert und so eine Funktionsstruktur aufgestellt. Für die Teilfunktionen werden Lösungsprinzipien gesucht, um anschließend ein System aus realisierbaren Modulen zu erstellen, die eine überschaubare Komplexität aufweisen und definierte Schnittstellen zu anderen Modulen aufweisen. Dadurch kann eine Parallelisierung des darauf folgenden Konstruktionsprozesses erreicht werden, der sowohl die Gestaltung der Module als auch die Ausarbeitung des Gesamtprodukts beinhaltet. Der Dynamik des Konstruktionsprozesses wird durch die Möglichkeit, iterativ zu vorangegangenen Arbeitsschritten zurückzukehren, Rechnung getragen.

Ein im allgemeinen Sprachgebrauch gebräuchlicheres Modell ist das von PAHL & BEITZ entwickelte 4-Phasen Modell für den Maschinenbau. Die vier Phasen der Konstruktion werden danach wie folgt definiert:

- Planen und Aufgabe klären (informative Festlegung)
- Konzipieren (prinzipielle Festlegung)
- Entwerfen (gestalterische Festlegung)
- Ausarbeiten (herstellungstechnische Festlegung).

Diese Konstruktionsphasen können den Arbeitsschritten der Konstruktionsmethodik nach [VDI93] direkt zugeordnet werden (Abbildung 2-1).

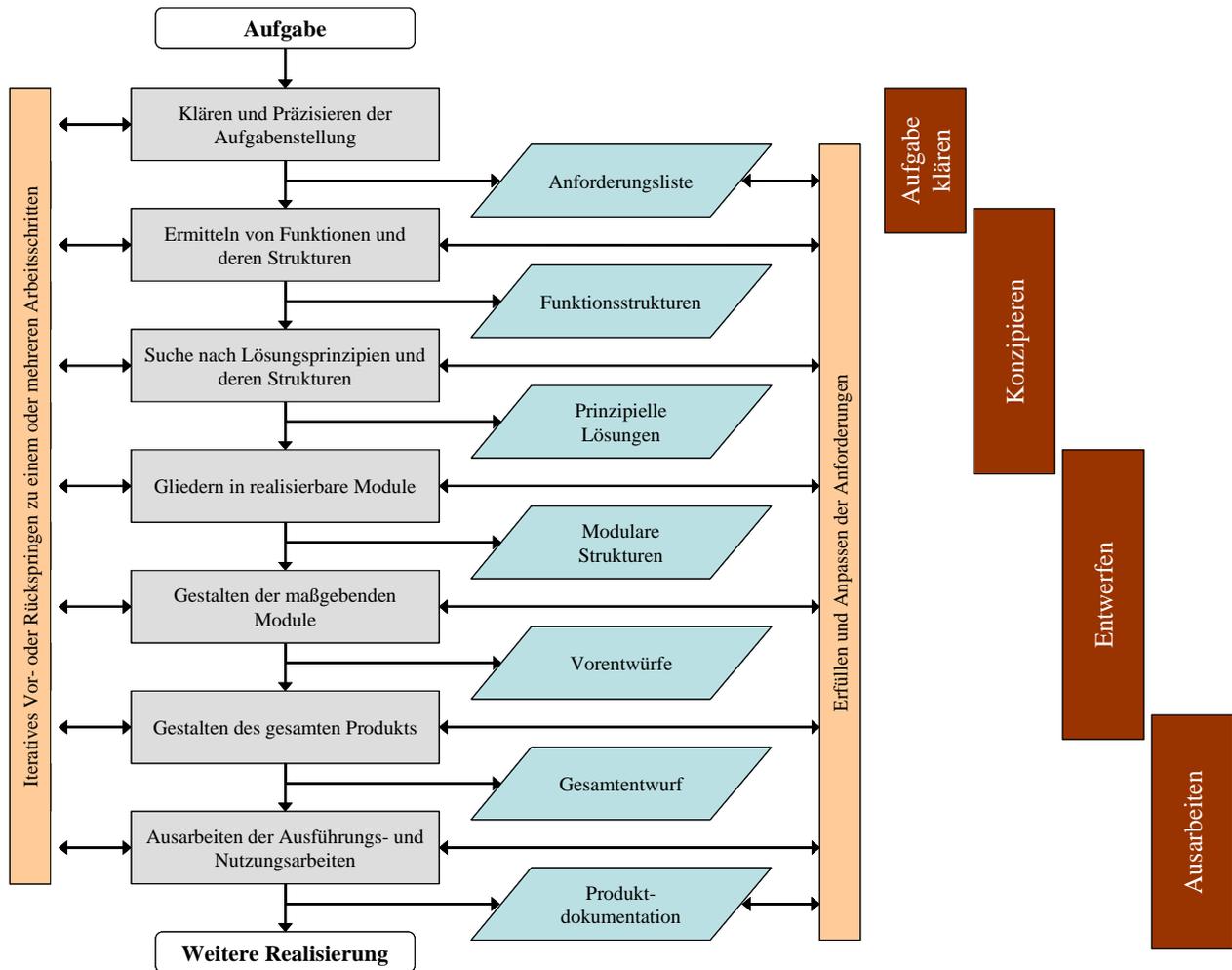


Abbildung 2-1: Vorgehensmodell für das Entwickeln und Konstruieren ([VDI93] und [PaBe07])

2.1.2 Konstruktionsarten

Je nach Innovationsgrad eines Produkts kann zwischen Neukonstruktionen, Anpassungskonstruktionen und Variantenkonstruktionen unterschieden werden. Nur im seltenen Fall der Neukonstruktion werden alle vier Phasen des Konstruktionsprozesses durchlaufen. Bei einer Anpassungskonstruktion hingegen werden bekannte und bewährte Lösungsprinzipien wiederverwendet. Lediglich die Gestaltung wird an die veränderten Randbedingungen angepasst. Bei dieser Aufgabenart stehen geometrische, festigkeitsrelevante und fertigungs- und werkstofftechnische Fragestellungen im Vordergrund [PaBe07]. Müssen aufgrund geänderter Randbedingungen die Produktabmaße geändert werden, ohne z. B. die Gestalt und den Werkstoff zu modifizieren, so spricht man von einer Variantenkonstruktion.

In der Praxis ist selten eine scharfe Abgrenzung zwischen den verschiedenen Konstruktionsarten vorzufinden. Dennoch lassen sich die zu durchlaufenden Phasen des Entwicklungsprozesses den einzelnen Konstruktionsarten grob zuordnen (Abbildung 2-2).

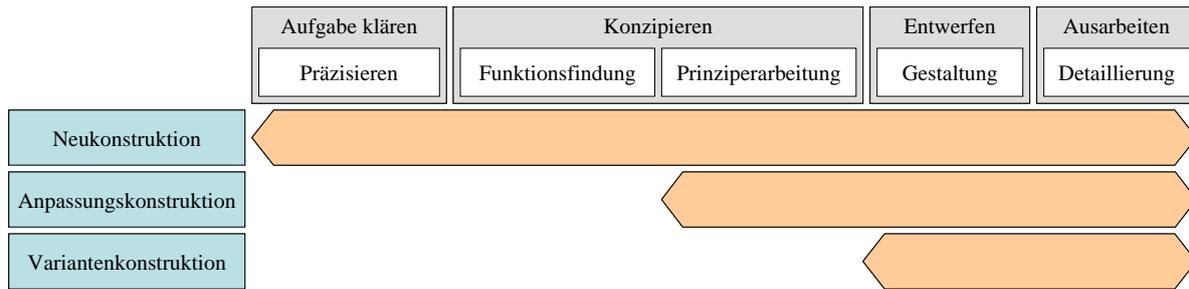


Abbildung 2-2: Zuordnung von Konstruktionsphasen zu Konstruktionsarten

2.2 Rechnergestütztes Konstruieren

Die Entwicklung von CAD-Systemen begann in den 1960er Jahren am Massachusetts Institute of Technology (MIT) in Boston mit der Entwicklung des Sketchpad durch SUTHERLAND [Sut03], mit dem es möglich war mit Hilfe eines Lichtstiftes und einer Tastatur interaktiv einfache technische Zeichnungen zu erstellen und zu manipulieren. Darauf aufbauend wurden durch ROSS erste Konzepte für 3D-Modelle entwickelt. Basis dieser 3D-Modelle bilden primitive Grundkörper, die zur Darstellung komplexerer Zusammenstellung verknüpft werden sollten [Ros60].

Im Jahr 1965 wurde dann erste Entwicklungsarbeiten für ein CAD-System für die Erstellung technischer Zeichnungen beim Flugzeughersteller LOCKHEED, USA begonnen [GaEb01]. Das System mit dem Namen CADAM (Computer Augmented Design And Manufacturing) basierte auf IBM-Großrechnern und spezieller Hardware (Monitore) und wurde durch IBM vermarktet. Ende der 1960er Jahre begann dann auch der französische Flugzeughersteller AVIONS MARCEL DASSAULT (heute DASSAULT AVIATION) mit der Entwicklung eines Programms für die Erstellung technischer Zeichnungen, das im Jahr 1981 unter dem Namen CATIA V1 (Computer-Aided Three-Dimensional Interactive Application) als eines der ersten kommerziellen Systeme auf den Markt gebracht wurde.

Die ersten 2D-CAD-Systeme waren ein digitaler Ersatz für sonst manuell am Zeichenbrett zu erstellende Zeichnungen. Durch die vielfältigen Möglichkeiten der Erstellung und Manipulation zweidimensionaler geometrischer Objekte war es möglich, innerhalb kürzester Zeit neue Zeichnungen zu erstellen oder bestehende Zeichnungen anzupassen.

Die Grundlage für die virtuelle Produktentwicklung wurde erst durch die Entwicklung dreidimensionaler CAD-Systeme geschaffen. Es existieren unterschiedliche rechnerbasierte Verfahren, um ein Produkt dreidimensional zu beschreiben [Abr05] (siehe Abbildung 2-3):

- Drahtmodelle

Drahtmodelle beschreiben die Produktgeometrie durch 3D-Punkte, die mit 3D-Kurven verbunden sind. Diese Form der Geometrirepräsentation ist nicht eindeutig, da es keine explizite Flächen- und Volumenzuweisung gibt. Daher lassen sich keine Bauteilschnitte aus einem Drahtmodell herleiten, ebenso fehlt die Erkennung von Körperdurchdringungen.

- Flächenmodelle

Flächenmodelle beschreiben die geometrische Gestalt des Produkts auf der Basis der Berandungsflächen. Flächenbasierte Modellersysteme bieten zumeist umfangreiche Funktionalitäten, um sowohl analytisch beschreibbare als auch approximierende oder interpolierende Flächen zu erzeugen und zu manipulieren. Mit Hilfe sog. NURBS-Flächen (**N**on-**U**niform **R**ational **B**-**S**plines) kann prinzipiell jede technisch herstellbare Form beliebig genau dargestellt werden. Wie im Fall der Drahtmodelle kann auch mit Flächenmodellen keine Volumeninformation gespeichert werden. Somit sind keine Volumenoperationen oder Schnittflächenermittlungen möglich, auch erfolgt keine Konsistenzprüfung des Modells bezogen auf die Realisierbarkeit. Aufgrund der hohen Leistungsfähigkeit einiger kommerzieller Flächenmodellierer werden vor allem in ästhetisch sensiblen Bereichen (z. B. Karosseriedesign der Automobilbranche) immer noch Flächenmodelle eingesetzt.

- Volumenmodelle

Volumenmodelle bieten im Gegensatz zu Draht- und Flächenmodellen eine eindeutige, widerspruchsfreie und genaue Körperbeschreibung. Auf der Basis eines Volumenmodells können automatisch Schnittkanten abgeleitet werden, ebenso lassen sich Baugruppenkonfigurationen auf Kollisionen überprüfen. Aus graphischer Sicht ergibt sich der Vorteil, dass sowohl verdeckte Kanten einfach ausgeblendet werden können als auch Schattierungen möglich sind. Weiterhin bilden Volumenmodelle eine detaillierte Informationsbasis für nachfolgende Prozesse wie z. B. Berechnungen und Simulationen. Aufgrund der hohen Komplexität von Volumenmodelliersystemen werden jedoch hohe Rechneranforderungen gestellt, ebenso sind für die Erstellung eines Volumenmodells andere Methoden als bei Draht- oder Flächenmodellen zu verwenden.

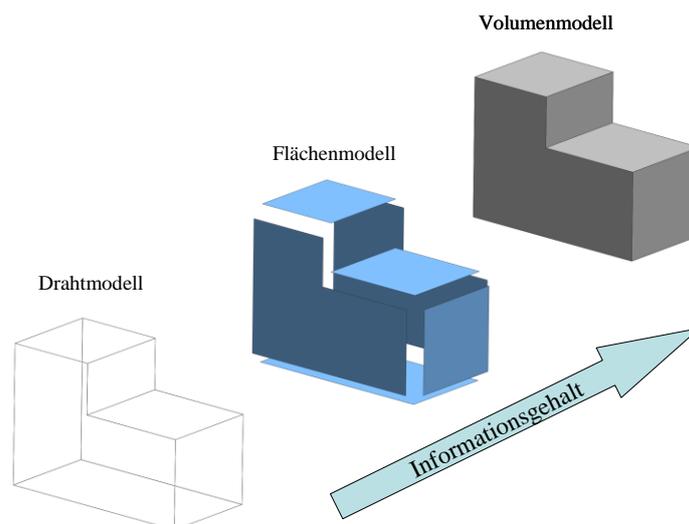


Abbildung 2-3: Geometriebeschreibung im CAD-System

2.2.1 Datenmodelle

Im Bereich der Konstruktion mechanischer Bauteile haben 3D-CAD-Systeme einen herausragenden Stellenwert eingenommen und sind vor allem in größeren Unternehmen zu einem De-facto-Standard für die Erzeugung digitaler Modelle in der VPE geworden.

Ein CAD-System basiert auf einem zumeist kommerziellen geometrischen Kernel, der eine Reihe von Basisfunktionen zur Erzeugung, Manipulation, Analyse und Visualisierung von 3D-Geometrie liefert, die auf einem rechnerinternen Datenmodell basiert. Im Bereich der Open-Source-Software hat sich dazu das SDK (Software **D**evelopment **K**it) Open CASCADE etabliert, um Anwendungsprogramme im Bereich CAD, CAM und CAE zu entwickeln [NN08a]. Die unterschiedlichen am Markt erhältlichen CAD-Systeme unterscheiden sich aufgrund der ähnlichen Grundfunktionalitäten daher zum größten Teil durch die Erweiterungen des bestehenden Kernel durch zusätzliche Funktionen. Die diversen geometrischen Kernel und ihre Datenmodelle sind zumeist nicht kompatibel, wobei es jedoch aufgrund der geschichtlichen Entwicklung grundlegende Gemeinsamkeiten in der Beschreibung geometrischer Modelle gibt.

Die rechnerinterne Beschreibung von Volumenmodellen kann auf der Basis multipler Ansätze erfolgen [SpKr97]. Grundsätzlich kann zwischen folgenden Beschreibungsformen unterschieden werden [Den04], die in Abbildung 2-4 nochmals unterteilt werden:

- Volumetrische Modelle (Punktmengenmodelle)
- Verknüpfungsmodelle
- Grenzflächenmodelle

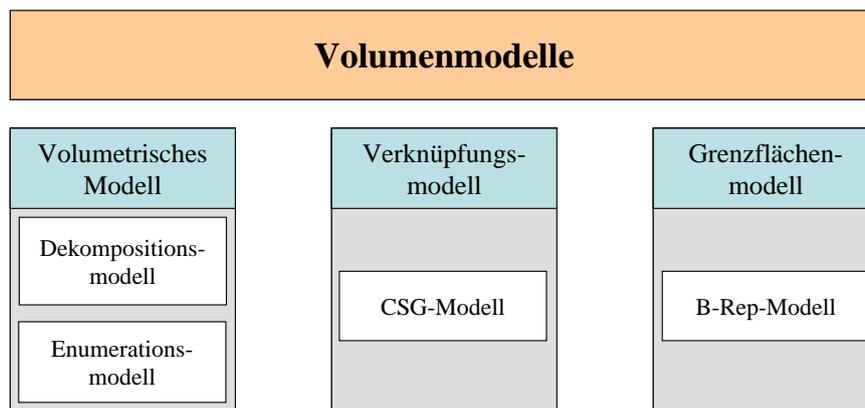


Abbildung 2-4: Arten von Volumenmodellen [Den04]

Da Volumetrische Modelle vor allem für diskrete Geometriemodelle im Bereich der Berechnung und Simulation verwendet werden und heutzutage im Bereich der Konstruktion nur eine unwesentliche Bedeutung haben wird in den folgenden Abschnitten detailliert nur auf die Verknüpfungs- und Grenzflächenmodelle nebst ihren Erweiterungen eingegangen.

2.2.1.1 Constructive Solid Geometry Model (CSG Volumenmodell)

In CSG-Modellen wird das zu modellierende Volumen durch mengentheoretische Verknüpfungen von Volumenprimitiva erzeugt. Die Volumenprimitiva können je nach Leistungsfähigkeit des CSG-Modellierers unterschiedlich generiert werden. Zu den Basiselementen zählen mathematisch einfach beschreibbare Körper wie z. B. Quader, Zylinder, Kugeln, Kegel oder Tori. Erweiterte Modelliersysteme bieten auch die Möglichkeit, Halbräume und nicht-primitive Volumenelemente zu verwenden. Halbräume werden durch eine Fläche, die ein Oberflächenelement eines Volumens darstellt, und die Angabe der Richtung in das Volumeninnere definiert. Mit Hilfe eines solchen Halbraumes können über die Definition der Materialseite ebene Schnitte durch andere Volumenelemente definiert werden. Die nicht-primitiven Volumenelemente sind zumeist systemspezifisch. Beispielhaft werden hier die Solid-Replica und Swept Area Solids aufgeführt. Solid-Replica-Elemente sind bereits verknüpfte Teilgeometrien eines Modells, die durch Kopieren an anderer Stelle des Modells eingesetzt werden können. Die Swept Area Solids beschreiben ein Volumen durch die Trajektion einer Fläche entlang einer räumlichen Kurve oder deren Rotation um eine Achse [And06].

Die Erzeugung des Volumenmodells basiert auf der geometrischen Transformation (Translation, Rotation, Skalierung) und mengentheoretischen Verknüpfung (Addition, Subtraktion, Durchschnitt) der verwendeten Volumenprimitiven. Die Informationen über diese Operationen werden in einem binären Baum¹ gespeichert. Jedes Blatt entspricht dabei einem geometrischen Primitiv, jeder Knoten repräsentiert eine mengentheoretische Verknüpfung. Das endgültige Modell wird durch die Wurzel des Baumes repräsentiert.

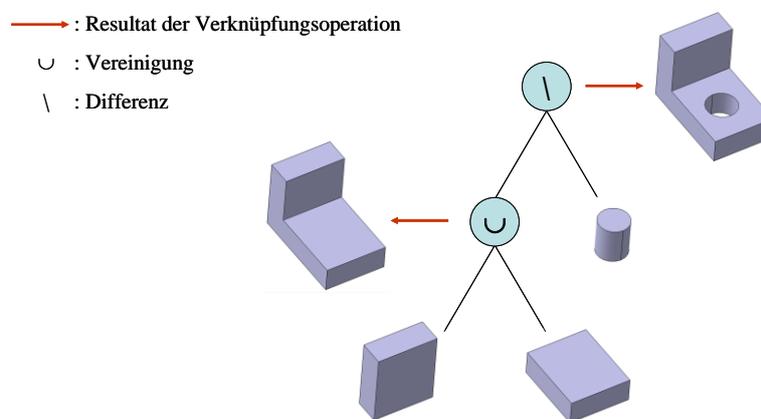


Abbildung 2-5: CSG-Struktur

Das CSG-Modell enthält keine expliziten Informationen über Eckpunkte, Kanten und Flächen der resultierenden Geometrie, da die Bauteilgeometrie über die Entstehungsgeschichte dokumentiert wird. Der daraus resultierende Vorteil dieses Datenmodells ist die Einsparung von Speicherplatz. Des Weiteren ist es auf diese Weise möglich, durch die Variation von Elementen und Operationen

¹ Als Binärbaum bezeichnet man in der Graphentheorie eine spezielle Form eines Graphen. Hierbei handelt es sich um einen gewurzelten Baum, bei dem jeder Knoten höchstens zwei Kindknoten besitzt.

in der Bauteilhistorie Änderungen der Bauteilgeometrie zu erreichen. Von Nachteil ist jedoch, dass bei jeder Modifikation die CSG-Struktur neu berechnet werden muss, was bei größeren Modellen in Abhängigkeit von der zur Verfügung stehenden Rechnerkapazität zeitaufwändig sein kann. Ebenso können zwei Modelle mit identischer Geometrie meist nicht direkt miteinander verglichen werden, da bei der Modellentstehung unterschiedliche Vorgehensweisen angewendet werden können. In Bezug auf die graphische Darstellung erfordert die implizite Darstellung der Geometrie zusätzliche Rechenoperation. Aufgrund der überwiegenden Nachteile des reinen CSG-Modells wird diese Form der rechnerinternen Geometrirepräsentation in modernden Modelliersystemen nicht mehr angewendet [Den04].

2.2.1.2 Boundary-Representation Model (Flächenbegrenzungsmodell)

Im Gegensatz zu der Modellerzeugung mit CSG-Methoden wird im Fall des Boundary-Representation-Modells (B-Rep Modell) das Bauteil durch die Volumenberandung beschrieben. Diese setzt sich aus Einzelflächen zusammen, wobei die Richtung des Flächennormalenvektors in Richtung des Objektinneren zeigt. In einem B-Rep-basierten Modell wird zwischen den geometrischen (Punkte, Kurven, Seiten) und den topologischen Elementen (Eckpunkt, Kante, Fläche) unterschieden (siehe Abbildung 2-6).

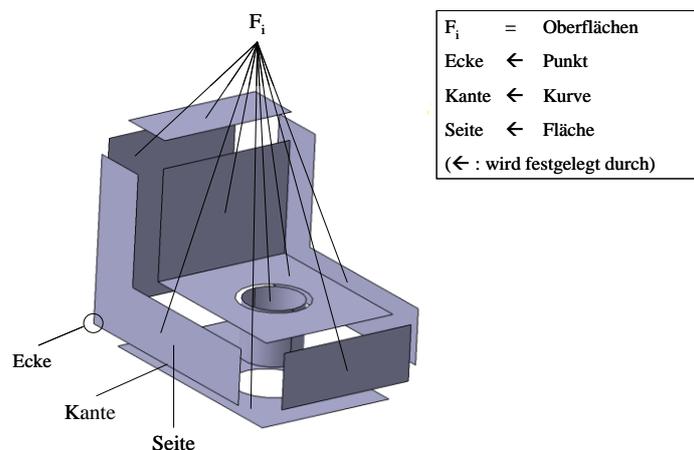


Abbildung 2-6: B-Rep-Modell

Das Datenmodell einer B-Rep-Struktur speichert sowohl die geometrische Beschreibung der einzelnen Elemente als auch ihre topologischen Beziehungen in einem gerichteten Graph. Die Knoten des Graphen repräsentieren hierbei die geometrischen Elemente, die Verbindungszeiger der Hierarchie speichern die topologischen Beziehungen. Im einfachsten Fall hat die Datenstruktur vier Hierarchieebenen. Eine alternative und häufig verwendete Beschreibungsform ist die *Baumgart's winged edge* Datenstruktur, die eine zusätzliche horizontale Verzeigerung der Objektkanten aufweist. Damit wird ein schnelles Durchsuchen des Datenmodells ermöglicht, wobei diese Art der Datenhaltung den Nachteil der Redundanz und des erhöhten Speicherplatzbedarfs mit sich bringt (Abbildung 2-6).

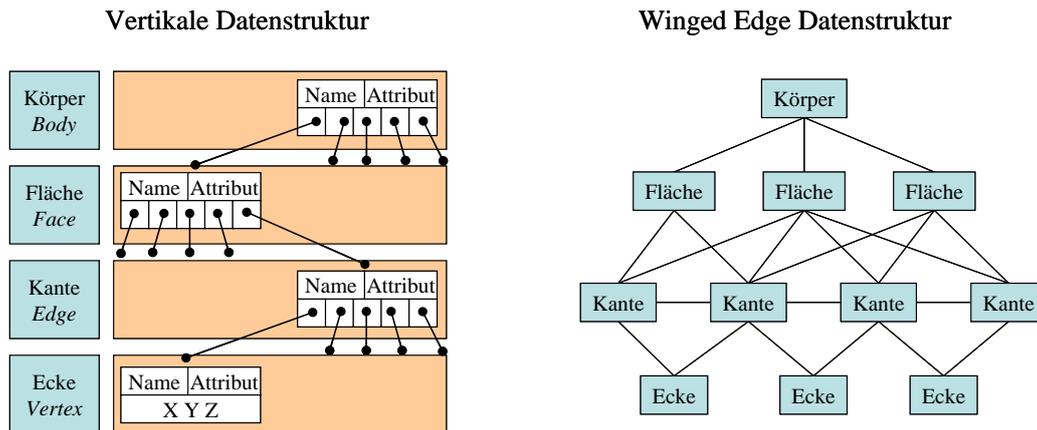


Abbildung 2-7: Datenstrukturen des B-Rep-Modells

Mit Hilfe von Euler-Operatoren können die Graphen des B-Rep-Modells verändert werden. Dabei gilt für alle Volumenmodelle die notwendige Bedingung, dass Sie die Euler-Poincaré-Gleichung erfüllen müssen, welche eine Aussage über die Anzahl der topologischen Elemente eines Modells trifft:

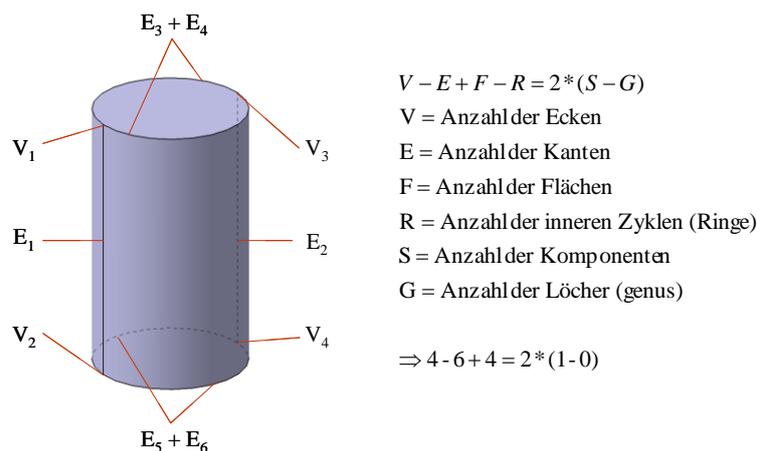


Abbildung 2-8: Euler-Poincaré-Gleichung des Zylinders

Euler-Operatoren enthalten eine Kombination aus topologischen Elementen, die selbst die Euler-Poincaré-Gleichung erfüllen. Neben der Euler-Poincaré-Gleichung als notwendige Bedingung für korrekte Volumenmodelle (sog. Manifold-Objekte) müssen noch folgende hinreichende Bedingungen erfüllt sein:

- Jede Kante trennt genau zwei Flächen
- Um jede Ecke existiert ein einzelner geschlossener Ring von Flächen
- Flächen können sich nur an einer gemeinsamen Ecke oder Kante schneiden

Da es in Modellersystemen nicht immer von Vorteil ist, ausschließlich Manifold-Objekte erzeugen zu können wurden die B-Rep-Datenstrukturen erweitert. Damit ist es möglich, auch Draht-, Flächen- und Zellmodelle zu beschreiben (sog. Non-Manifold-Objekte).

Im Gegensatz zu der CSG-Methode bietet das B-Rep-Modell keine Modellhistorie. Jedoch ist es mit dieser Beschreibungsform möglich, einzelne Körperelemente (z. B. eine Kante für das Anbringen einer Fase) zu referenzieren. Die Datenstruktur bietet des Weiteren die Möglichkeit, diese Elemente zu Attributieren, um so das Modell an bestimmten Stellen mit semantischen Informationen anzureichern. Aufgrund der expliziten Speicherung von Topologie und Geometrie gestaltet sich die grafische Ausgabe einfacher und schneller als bei CSG-Modellen.

2.2.1.3 Hybride Modelle

Sowohl CSG-basierte Modelle als auch B-Rep Modelle weisen methodenbedingte Vor- und Nachteile auf. Um eine flexiblere Datenstruktur mit den Vorteilen der beiden Modelliermethoden unter weitestgehendem Ausschluss der Nachteile zu erhalten basieren die modernen CAD-Systeme auf einem hybriden Datenmodell. Hybridmodelle bestehen immer aus zwei Datenstrukturanteilen, wobei es sich meist um eine primäre CSG- und eine sekundäre B-Rep-Datenstruktur handelt. Grundsätzlich kann zwischen zwei Arten von Hybriden Modellen unterschieden werden [And06].

Im ersten Fall enthält die Primärdatenstruktur (CSG) alle Modellinformation vollständig, die Sekundärstruktur (B-Rep) dient der schnellen Visualisierung des Modells. In jedem Knoten des CSG-Binärbaums wird als redundante Information die ausgewertete Darstellung des CSG-Modells gespeichert. Die Primärdatenstruktur dominiert das rechnerinterne Datenmodell, alle Operationen werden auf diese Datenstruktur angewendet. Folglich können im Gegensatz zu B-Rep-Modellen weiterhin keine Operationen an geometrischen Elementen des Modells angewendet werden.

Im zweiten Fall ergänzen sich CSG- und B-Rep-Modell, wobei die CSG-Struktur als Primärstruktur erhalten bleibt. Die Entstehungsgeschichte wird im Binärbaum festgehalten, gleichzeitig können alle geometrischen Elemente identifiziert und für weitere Operationen verwendet werden. Die aus diesen Operationen entstehenden Teilvolumina werden in der Primärstruktur gespeichert und die Sekundärstruktur neu berechnet. Entscheidendes Merkmal dieser Art von Datenmodell ist die bidirektionale Kopplung der beiden Datenstrukturen.

2.2.2 Parametrisch-assoziative Modellierung

Konventionelle Modellersysteme bieten mannigfaltige Möglichkeiten, um die Produktgestalt dreidimensional zu beschreiben. Jedoch können Änderungen sehr aufwändig sein, wenn umfangreichere Abhängigkeiten zwischen den geometrischen Elementen zu beachten sind. Der in der Konstruktion häufig auftauchende Fall der Variantenkonstruktion wird daher von diesen Systemen nur marginal unterstützt. Mit zunehmendem Einsatz der dreidimensionalen Produktmodellierung kam daher die Forderung nach möglichst variablen Modellbeschreibungen auf. Als Folge wurden parametrische CAD-Systeme entwickelt.

Parametrische 3D-CAD-Systeme können als Weiterentwicklung der volumenbasierten 3D-CAD-Systeme betrachtet werden. Diese Systeme implementieren sowohl eine bidirektionale Assoziativität zwischen Maßzahl und Geometrie als auch Constraints. Die Parametrisierung eines Modells

beruht auf der Verbindung der einzelnen geometrischen Modellelemente durch diese Parameter und Constraints, welche die Grundlage der Arbeitsweise dieser Methode darstellen [And06].

Die heute am Markt bestehenden Systeme basieren auf der Kombination unterschiedlicher Systemtechniken, jedoch wird der Terminus „Parametrik“ mittlerweile als Synonym sowohl für einzelne als auch für kombinierte Systemtechniken verwendet [Men99] [For03]. Eine grundsätzliche Unterscheidung kann anhand der Art der Gleichungsauflösung erfolgen. Beim parametrischen Ansatz wird das durch die Parameter und Constraints bestimmte Gleichungssystem sequentiell aufgelöst. Hierzu sind sehr stabile Lösungsverfahren entwickelt worden. Beim Variationsansatz hingegen wird der Satz von Gleichungen simultan gelöst [Dun08]. Mit Hilfe dieses Ansatzes können auch zirkuläre Beziehungen aufgelöst werden, jedoch weisen die Lösungen der Gleichungssysteme u. U. eine geringere numerische Stabilität auf, weswegen häufig der parametrische Ansatz in den Systemen implementiert ist.

Eine Erweiterung der rein parametrischen Systeme sind parametrisch-assoziative Systeme. Zwischen den einzelnen Geometrieelementen und Modellobjekten können unidirektionale Assoziationen aufgebaut werden. Abbildung 2-9 verdeutlicht dies an einem Sweep-Körper, dessen Definition mindestens aus einem skizzierten Profil und einer Leitkurve besteht.

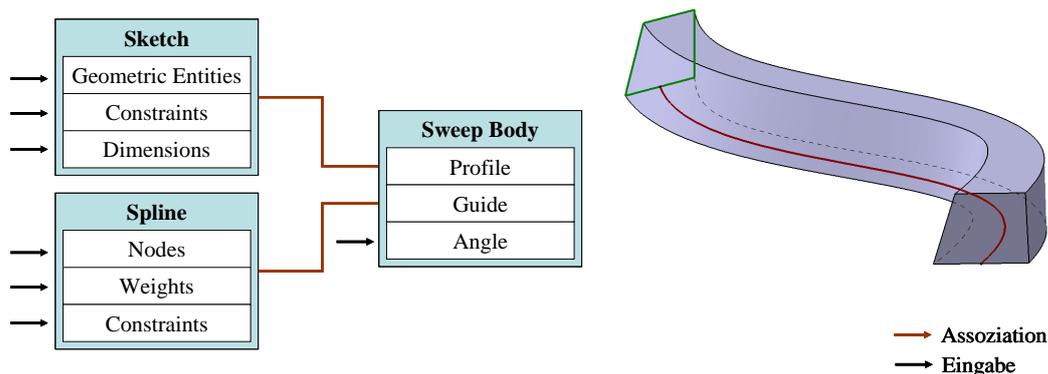


Abbildung 2-9: Assoziative Modellierung eines Sweep-Körpers

Der Aufbau von Assoziationen in einem Modell erzeugt gerichtete Verbindungen zwischen den Modellierungsobjekten, die auch als Eltern-Kind-Beziehungen bezeichnet werden. Neben der Erzeugungsvorschrift, den Constraints und Parameterwerten wird nun auch die Referenzkette im Datenmodell abgelegt. Diese Methode sichert die Modellkonsistenz bei Änderungen, erhöht jedoch zeitgleich die Modellkomplexität. Das Relationsgeflecht kann vor allem bei großen Modellen sehr schnell unüberschaubar werden, was wiederum den Aufwand bei Änderungen sehr erhöht. In zahlreichen Arbeiten wurden daher Methoden entwickelt, mit denen ein zielgerichteter Aufbau parametrisch-assoziativer Modelle ermöglicht werden soll [ShMä95] [SpKr97] [Men99] [Mei00] [For03] [Ger04] [Stra06] [Köh08].

2.2.3 Featurebasierte Modellierung

Die Einführung der parametrisch-assoziativen Modellierung ermöglichte eine zunehmende Effizienzsteigerung in der Phase der Konstruktion durch die Möglichkeit, innerhalb kurzer Zeit sowohl Anpassungs- als auch Variantenkonstruktionen herzustellen. Unberücksichtigt bleiben hier mögliche Einsparpotentiale durch die Integration von zusätzlichem Wissen aus nachgeschalteten Prozessen (Arbeitsplanung, Fertigung, Montage, Qualitätssicherung) in das auf der dreidimensionalen Geometriebeschreibung aufbauende Produktmodell. Seit dem Beginn der 90er Jahre unterliegt der gesamte Produktentwicklungsprozess jedoch einem starken Wandel. Die Forderung nach immer kürzeren Entwicklungszeiten, höherer Produktqualität und der Zwang zu Standort- und Unternehmensübergreifender Produktentwicklung zwingen die Unternehmen zu einer Effizienzsteigerung in allen Phasen der Produktentwicklung. Ein vielversprechendes Werkzeug zur Erreichung dieser Ziele ist die Feature-Technologie [JaDa97].

Erste Ansätze der Feature-Technologie wurden im Bereich der Arbeitsplanung entwickelt. Bestimmten Elementen eines Geometriemodells wurden Fertigungsanweisungen zugeordnet, um einen (teil-)automatisierten Fertigungsprozess zu ermöglichen [VDI03a]. Daraus entstand mit zunehmender Intensivierung der Arbeiten in diesem Bereich ein verallgemeinerter Feature-Begriff. In modernen Entwicklungen wird die Feature-Technologie als Möglichkeit angesehen, über alle Phasen des Produktentwicklungsprozesses hinweg Informationen zu erfassen und verknüpfen zu können. Auf der Basis eines featurebasierten Produktmodells sollen dann Informationen für nachgeschaltete Phasen der Produktentwicklung möglichst automatisiert verarbeitet werden, um so Zeit, Kosten und Qualität über den gesamten Produktlebenszyklus hinweg positiv beeinflussen zu können.

2.2.3.1 Begriffsdefinitionen

Die VDI-Richtlinie 2218 [VDI03a] stellt den aktuellen Stand der Technik im Bereich der Feature-Technologie dar. In diese Richtlinie flossen auch die Ergebnisse der FEMEX (**F**eature **M**odeling **E**Xperts) ein, eine internationale und interdisziplinäre Gruppe von Forschern, Entwicklern und Anwendern aus Universitäten, Forschungsinstituten und Industrie. Ein Ergebnis der Tätigkeiten einer Arbeitsgruppe ist eine allgemeine Definition des Feature-Begriffs [BäWe96]:

Features sind informationstechnische Elemente, die Bereiche von besonderem (technischen) Interesse von einzelnen oder mehreren Produkten darstellen

Diese Definition hebt hervor, dass ein Feature keine physikalische Entsprechung haben muss. Als informationstechnisches Element repräsentiert es ein Modell, das digital verarbeitet werden kann. Gemäß dieser Definition muss ein Feature keine Geometrie enthalten.

Ein Feature wird durch das Zusammenführen von Produkteigenschaften beschrieben. Ein Feature repräsentiert eine spezifische Sichtweise auf die Produktbeschreibung, die mit bestimmten Eigenschaftsklassen und bestimmten Phasen des Produktlebenszyklus im Zusammenhang steht [VDI03a].

Es filtert also die in einem gewissen Kontext relevanten Eigenschaften eines Produkts (vgl. Abbildung 2-10).

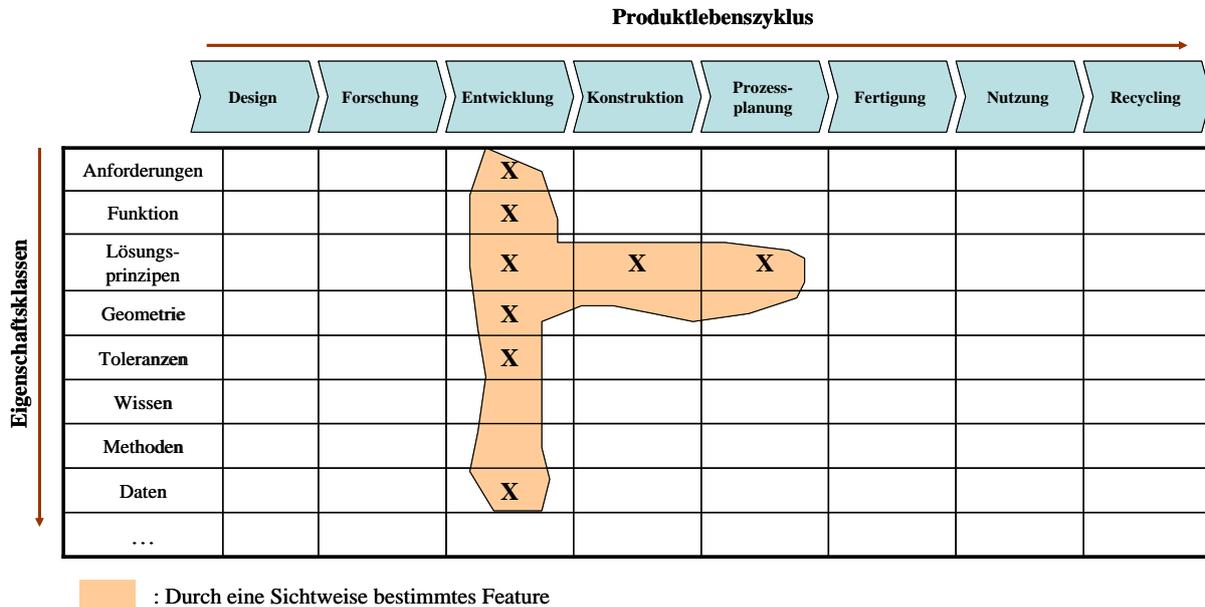


Abbildung 2-10: Feature im Produktlebenszyklus (nach [JaDa97])

Die Verknüpfung der verschiedenen Phasen des Produktlebenszyklus erfolgt über Features, die mehrere Eigenschaftsklassen integrieren. Ein Feature, das nur die Eigenschaftsklasse „Geometrie“ implementiert, dient z. B. primär nur der Vereinfachung oder Aggregation von geometrischen Elementen. Sobald es aber die Eigenschaftsklasse „Toleranzen“ implementiert wird es zu einem verknüpfenden Feature, das die Phasen Konstruktion, Prozessplanung und Fertigung miteinander in Beziehung setzt. HOFFMANN bezeichnet diesen Feature-Typ auch als „Höherwertiges Feature“ [Hof01].

Aufgrund der Tatsache, dass der Einsatz von Features in vielen CAx-Systemen vor allem mit der Eigenschaftsklasse „Geometrie“ verknüpft ist und viele Produkteigenschaften von dieser Eigenschaftsklasse abhängen, ist auch eine etwas spezifischere Definition gebräuchlich [Rie95]:

$$\text{Feature} = \text{Aggregation von Geometrie und /oder Semantik}$$

Die Semantik eines Features ist hier die in den verschiedenen Phasen des Produktlebenszyklus interpretierbare Bedeutung des Features [Rie95].

2.2.3.2 Die Feature-Technologie in CAD/CAM-Systemen

Nahezu alle modernen parametrisch-assoziativen CAD-Systeme unterstützen durch ihre integrierte Datenstruktur die Grundzüge der Feature-Technologie. In erster Linie wird der Konstrukteur durch Form-Features unterstützt, die im Bereich der Geometrieerstellung ihre Anwendung finden. Diese Features besitzen zumeist einen geringen Informationsgehalt für weitere Prozesse, da sie häufig systemspezifische Geometrieerzeugungsfunktionen abbilden. Formfeatures reichen von Bezugselementen (Punkte, Achsen, Koordinatensysteme, Bezugskurven) und Gestalt bestimmenden Geo-

metrien (Kurvenzüge) über querschnittorientierte Körpererstellung (Trajektionen) bis hin zu Freiformmodellierung (B-Splines, etc.). Die Detaillierung eines Modells wird durch die Anwendung fertigungsorientierter Features unterstützt (Bohrungen, Fasen, Nuten, etc.). Der Informationsgehalt dieser Features wird durch die Einbindung von Normen und Wissen aus den relevanten Fertigungstechnologien bestimmt. Neben diesen nativen Features finden sich in integrierten Anwendungssystemen auch Sammlungen branchenspezifischer Features, z. B. aus den Bereichen Schiffbau, Schweißkonstruktion, Formenbau oder auch Blechbearbeitung.

Der Begriff „Feature“ kann in vielen CAD/CAM-Systemen mit dem Begriff des „Objekts“ aus dem Bereich der Informationstechnik verglichen werden. Der Benutzer instanziiert graphisch und interaktiv ein Exemplar einer bestimmten Featureklasse. Das Featureobjekt stellt entsprechende Methoden, Eigenschaften und Bezüge zu anderen Objekten zur Verfügung. Eine Methode eines Nutenfeatures wäre z. B. das interaktive Platzieren der Nut im Modell, eine Eigenschaft die Nuttiefe und ein Bezug die Verknüpfung zu einem B-Rep Objekt oder einem weiteren Feature, um die Platzierungsfläche zu ermitteln. Im Sinne der objektorientierten Modellierung kann ein Feature weitere Features enthalten, auf die im Sinne einer durchgängigen Modellierung zugegriffen werden kann.

Entsprechend dieser Sichtweise kann die interaktive Modellierung eines Produktmodells auch als Programmieraufgabe verstanden werden, die mit einem graphischen Editor ausgeführt wird und ein direktes Feedback gibt. Diese Ansicht wird durch die Bestrebung einiger Systemhersteller gestützt, die neben den bekannten nativen Features weitere Objekte im Modellbaum des CAD-Modells erfassen und strukturieren. So werden z. B. auch geometrische Operationen (Spiegeln, Mustern, Transformationen) als Feature im Modellbaum erfasst. Neben den geometrischen Objekten eines Modells finden sich im Modellbaum auch Objekte der Wissensverarbeitung und weiterer Teilgebiete, die über die reine Geometriemodellierung hinausgehen können. Das Geometriemodell wird damit zu einem Partialmodell eines übergeordneten Produktmodells.

2.2.3.3 Die Feature-Technologie im Produktlebenszyklus

Zunehmender Wettbewerbsdruck und globalisierte Märkte üben einen verstärkten Zwang auf die Unternehmen aus, in immer kürzerer Zeit innovative und qualitativ hochwertige Produkte zu einem konkurrenzfähigen Preis auf den Markt zu bringen. Um diese Ziele zu erreichen ist eine Parallelisierung der traditionell sequentiell ablaufenden Phasen des Produktentwicklungsprozesses unumgänglich. Die Feature-Technologie unterstützt die dafür notwendigen organisatorischen Maßnahmen und Änderungen durch die Möglichkeit, Informationen aus den verschiedenen Prozessen und Phasen des Produktlebenszyklus zu verknüpfen [BäHa01] [Bos07]. Die Feature-Verwendung im Sinne von einheitlichen und durchgängigen Informations- und Integrationsobjekten bildet damit die Grundlage für die Beschreibung aller Vorgänge entlang der Prozesskette mit dem Ziel der vollständigen Vernetzung aller am Produktentstehungsprozess beteiligten Bereiche [Haa99]. Des Weiteren gilt eine durchgängige Produktbeschreibung durch Features als Voraussetzung für die rechnergestützte Wissensverarbeitung [VDI03a]. Jedoch durchdringen Features immer noch nicht alle Bereiche der VPE.

Insbesondere in den frühen Phasen der Produktentwicklung ist die Unterstützung durch die Feature-Technologie noch gering (Abbildung 2-11).

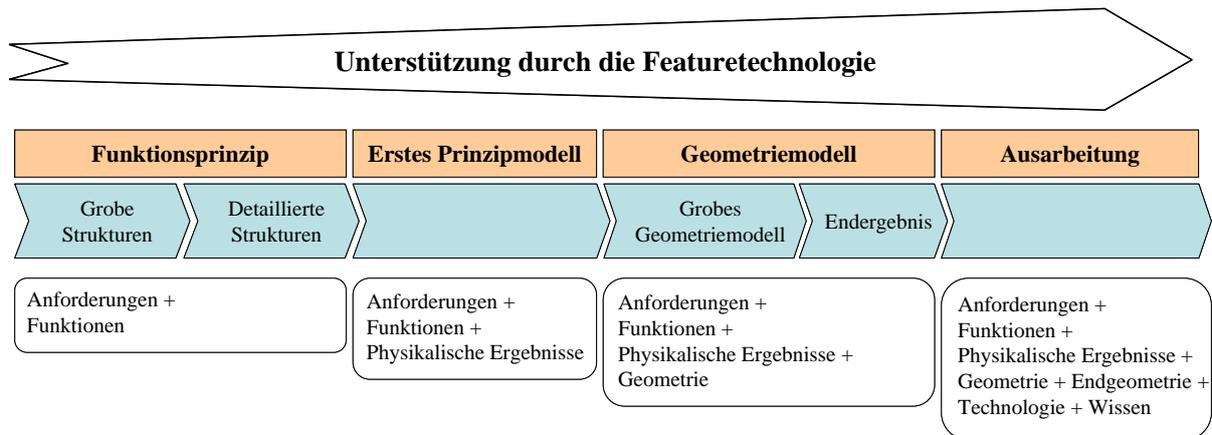


Abbildung 2-11: Durchdringung der Feature-Technologie in den Phasen des Produktentwicklungsprozesses (in Anlehnung an [VaPo98])

2.3 Datenbanksysteme

2.3.1 Anforderungen an Datenverwaltungssysteme

Die zunehmende Informationsflut in allen Bereichen des geschäftlichen und privaten Lebens erfordert geeignete Möglichkeiten, um alle anfallenden Daten effizient zu speichern und zu verwalten. Eine mögliche Variante ist der direkte Zugriff einer Anwendung auf eine Datei. Hierdurch ergeben sich jedoch folgende Nachteile:

- Sämtliche Routinen für die Dateimanipulation und Dateiverwaltung müssen in allen Anwendungen implementiert werden
- Die Daten- und Dateiformate sind nicht zwangsläufig standardisiert und portabel
- In einer Mehrbenutzerumgebung muss die Datenkonsistenz innerhalb jeder Anwendung gesichert werden
- Häufig redundante Datenhaltung durch Kopien

Aufgrund dieser schwerwiegenden Nachteile wurde in den 60er Jahren ein Konzept für ein Datenbanksystem entwickelt, um zwischen der datenverarbeitenden und datenerzeugenden Anwendungsebene und der Ebene der Datenspeicherung eine weitere vermittelnde Ebene zu platzieren. Datenbanksysteme müssen in der Lage sein, folgende Problemstellungen zu lösen [KIRa05]:

- Daten müssen manipulierbar sein
- weitgehende Redundanzfreiheit; kontrollierbare Redundanz ist jedoch manchmal notwendig und unvermeidbar
- Universelle Verwendbarkeit der Daten im jeweiligen Anwendungskontext

- Daten müssen anwendungsübergreifend les- und manipulierbar sein
- Konfigurationsunabhängigkeit der Datenbankverwaltung von Hardware- und Softwareumgebung sowie Netztopologie
- Funktionale Integration: semantische Datenzusammenhänge sollen so dargestellt werden, dass sie transparent und nutzbar sind
- Strukturflexibilität: Die Struktur der Daten sollte veränderbar sein
- Mehrbenutzerbetrieb
- Benutzer- und Berechtigungsverwaltung
- Gewährleistung der Datenintegrität
- Datensicherheit: Sicherheitsspeicherung (Backups) und Rekonstruktionsverfahren (Rollforward) sollten vom DBMS unterstützt werden

In nahezu allen Unternehmen finden sich heute Datenbanksysteme, die diese Aufgaben erfüllen. Der Aufbau und sichere Betrieb solcher Datenbanksysteme fällt in den Arbeitsbereich der IT-Administration und trägt entscheidend zum Unternehmenserfolg bei. Neben der gesetzlich vorgeschriebenen langjährigen Speicherung aller produktbezogenen Dokumente ist die zeitnahe Bereitstellung von Daten in verschiedenen Anwendungen in vielen Branchen ein kritischer Erfolgsfaktor (z. B. E-Commerce Kataloge oder Internet-basierte Konfiguratoren).

2.3.2 Aufbau und Struktur

Ein Datenbanksystem setzt sich aus einer Menge an Daten (**Datenbank**) und einer Verwaltungssoftware (**Datenbankmanagementsystem – DBMS**) zusammen. Im Gegensatz zu einer dateibasierten Datenhaltung erfolgt durch ein Datenbanksystem eine strikte Trennung zwischen den eigentlichen Daten und den Zugriffsalgorithmen. Des Weiteren werden zusätzliche Informationen über die Struktur der Daten (Metadaten) gespeichert (Abbildung 2-12).

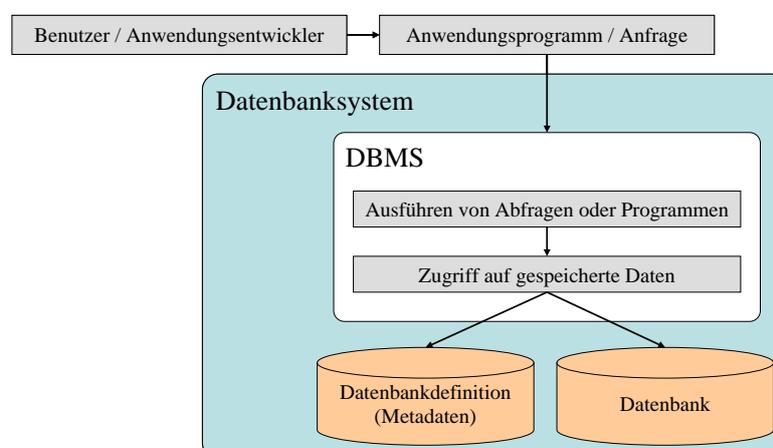


Abbildung 2-12: Aufbau eines Datenbanksystems

Die Trennung zwischen physischen Daten und Datenbankapplikationen wird durch ein dreischichtiges Architekturmodell ermöglicht. Dieses gliedert sich in eine interne, eine konzeptionelle und eine externe Ebene. Die interne Ebene beschreibt den physikalischen Zusammenhang der Datenbank und wird allein durch das DMBS verwaltet. In der konzeptionellen Ebene wird die logische Struktur der Datenbank erfasst (z. B. Tabellen oder Datenmodelle). Auf diese Ebene kann durch eine Abfragesprache wie z. B. SQL (Structured Query Language – Strukturierte Abfragesprache) zugegriffen werden. Eine individuellere Sicht auf die Daten bietet die externe Ebene, in der eine benutzerspezifische Sicht auf die Daten in Form von graphischen Benutzeroberflächen generiert werden kann.

2.3.3 Datenmodelle

Ein wesentliches Merkmal eines Datenbanksystems ist das bereits erwähnte Datenmodell. Folgende Datenmodelle stehen derzeit zur Verfügung [KeEi06]:

- Hierarchisches Datenbankmodell
- Netzwerkdatenbankmodell
- Relationales Datenbankmodell
- Objektrelationales Datenbankmodell
- Objektorientiertes Datenbankmodell

Das hierarchische Datenbankmodell und das Netzwerkdatenbankmodell wurden vor allem in den Anfängen der Entwicklung von Datenbanksystemen eingesetzt und finden heutzutage kaum noch Anwendung. Das objektorientierte Datenbankmodell gehört zu den modernsten Konzepten der Datenspeicherung. Es basiert auf der Idee, die Objekte aus den objektorientierten Programmiersprachen direkt in einer Datenbank zu halten. Aufgrund der schlechten Performanz bei komplexen Abfragen hat sich dieses Datenbankmodell nicht durchgesetzt. Das relationale Datenbankmodell ist heute vorherrschend und wird daher im Folgenden kurz beschrieben. Zu der Klasse der relationalen Datenbanken gehören auch die objektrelationalen Datenbanken, bei denen das Konzept der Vererbung für die Definition und Abfrage ähnlicher Tabellen übernommen wurde. Für weitergehende Informationen zu relationalen Datenbankmodellen sei auf die weiterführende Literatur verwiesen [KIRa05] [KeEi06].

Alle relationalen Datenbanksysteme basieren auf der von CODD vorgeschlagenen Relationalen Algebra [Cod70], mit der sich Anfragen über einem relationalen Schema durchführen lassen. Der Begriff „Relation“ ist hierbei wie folgt definiert:

Seien M_1, \dots, M_n beliebige Mengen. Die Menge $M_1 \times \dots \times M_n$ bezeichnet das kartesische Produkt von M_1, \dots, M_n . Es besteht aus den geordneten (n -)Tupeln (m_1, \dots, m_n) , wobei $m_i \in M_i, 1 \leq i \leq n$. Eine Relation R ist eine Teilmenge $R \subseteq M_1 \times \dots \times M_n$. Die Indizes $1, \dots, n$ heißen Attribute der Relation; die Relation hat dann die Attributmengende $A = \{1, \dots, n\}$. Ist $(m_1, \dots, m_n) \in R$ so heißt m_i Attributwert zum Attribut i .

Die Attribute werden im Allgemeinen nicht durch Indizes sondern durch Namen gekennzeichnet. Eine Relation wird in Datenbanksystemen in Form einer Tabelle dargestellt (vgl. Abbildung 2-13).

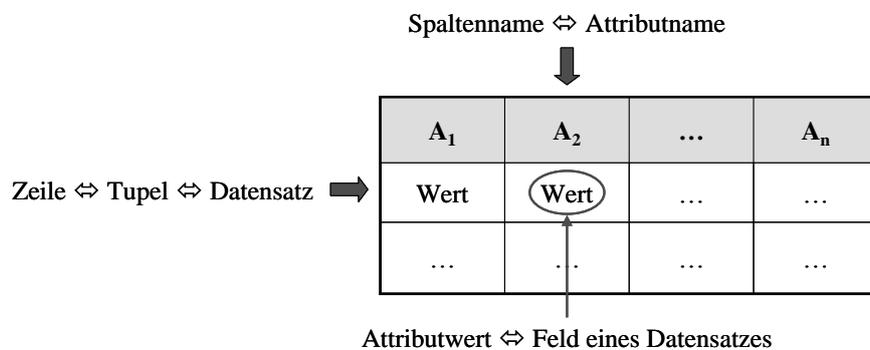


Abbildung 2-13: Relation in Tabellendarstellung

Jede Zeile einer Tabelle (Tupel) ist ein Datensatz (Record). Jedes Tupel besteht aus einer Anzahl von Attributwerten, deren Anzahl und Typ durch das Relationenschema festgelegt werden. Das Relationenschema beschreibt also den Aufbau der Tabelle. Für die eindeutige Identifikation eines Datensatzes werden sogenannte Schlüssel verwendet. Schlüssel können Attribute oder minimale Mengen von Attributen sein, die einen Datensatz eindeutig identifizieren (Primärschlüssel). Anhand der Schlüssel können Beziehungen zwischen den Relationen hergestellt werden. Über den Primärschlüssel ist eine Verknüpfung von Relationen möglich. Dazu wird der Primärschlüssel der einen Relation als Attribut in die andere Relation aufgenommen. In diesem Fall spricht man von einem Fremdschlüssel.

Der Entwurf einer relationalen Datenbank wird auch *Datenmodellierung* genannt. In einem Schema werden die Daten und die Beziehungen der Daten untereinander modelliert. Ein wichtiger Schritt nach der Erstellung des Schemas ist die Normalisierung, um Redundanzen und Anomalien innerhalb der Datenbank zu vermeiden. Es existieren fünf Normalformen, wobei den ersten drei Normalformen in der praktischen Umsetzung die größte Bedeutung zukommt [KlRa05].

- Erste Normalform
Ein Relationenschema ist in der ersten Normalform, wenn alle Attribute atomar sind (Keine Tupel oder Mengen)

- **Zweite Normalform**

Ein Relationenschema ist in der zweiten Normalform, wenn die erste Normalform vorliegt und jedes Nichtschlüsselattribut von jedem Schlüsselkandidaten voll funktional abhängig ist.

- **Dritte Normalform**

Ein Relationenschema ist in der dritten Normalform, wenn die zweite Normalform vorliegt und jedes Nichtschlüsselattribut von keinem Schlüsselkandidaten transitiv abhängt.

Die Datenmodellierung und die daran anschließende Normalisierung der Daten hat Auswirkungen auf die Art und Weise der Datenabfrage. Die hier zu berücksichtigenden Randbedingungen werden daher in den folgenden Abschnitten näher erläutert.

2.3.4 Datenzugriff und Schnittstellen

2.3.4.1 SQL

SQL ist eine standardisierte Datenbanksprache, mit der Daten aus relationalen Datenbanksystemen definiert, manipuliert und abgefragt werden können. 1987 wurde der erste SQL-Standard von der ISO ratifiziert (SQL:1987). Die letzte Fassung des Standards stammt aus dem Jahr 2006 und wird daher SQL:2006 genannt. SQL:2006 ist sehr umfangreich und in mehrere Teile gegliedert, die jeweils in einem eigenen Standardisierungsdokument beschrieben werden [ISO06]. Die meisten Datenbanken implementieren nur Teile des aktuellen SQL-Standards und erweitern ihren eigenen SQL-Dialekt um proprietäre Befehle [Sch07].

Im Gegensatz zu prozeduralen Programmiersprachen ist SQL eine mengenorientierte und deklarative Sprache, deren Semantik sich an umgangssprachliche englische Ausdrücke anlehnt. Die Sprachbefehle lassen sich in drei Kategorien einteilen:

- **Data Definition Language (DDL)**
Befehle zur Definition des Datenbankschemas.
- **Data Manipulation Language (DML)**
Befehle zum Erzeugen, Manipulieren und Ändern von Datensätzen. Schließt Befehle zur Erzeugung von Abfragen mit ein.
- **Data Control Language (DCL)**
Dient der Rechteverwaltung innerhalb der Datenbank.

Das Datenbankschema bestimmt den Aufbau der Abfrage-Klauseln (SQL-Statements). Die Komplexität der SQL-Statements steigt mit zunehmender Granulierung der Datenstruktur und ist für Anwender ohne genaue Kenntnis des Datenbankschemas nicht durchführbar. Aus diesem Grund können Sichten (Views) generiert werden, die eigentlich vordefinierte SQL-Statements darstellen. Alle weiteren SQL-Statements können auf dieser vereinfachten Sicht ausgeführt werden. Ein weite-

rer Vorteil von Sichten ist die bereits vorher durchgeführte Optimierung der Abfrage, da die SQL-Statements der Sicht bereits durch einen Abfrageoptimierer aufbereitet wurden.

2.3.4.2 ODBC

Die Einführung eines Datenbanksystems basiert auf der Idee, die Abfrage und Manipulation von Daten unabhängig von der jeweiligen Anwendung aus durchführen zu können. Eine wesentliche Komponente dieser Anwendungsneutralität ist die Bereitstellung einer standardisierten Schnittstelle, die es ermöglicht, Datenbankzugriffe unabhängig vom verwendeten DBMS zu gestalten. Im Bereich der relationalen Datenbanken wird dies durch die ODBC-Schnittstelle ermöglicht (**O**pen **D**atabase **C**onnectivity – Offene Datenbank-Verbindungs-fähigkeit). ODBC ist eine Spezifikation von MICROSOFT, die auf der Call Level Interface Spezifikation der OPEN GROUP basiert [NN08c].

ODBC stellt eine API bereit, mit der Anfragen und Befehle an Datenbanksysteme mit SQL-Statements formuliert werden können. Von Seiten des DBMS-Herstellers muss ein betriebssystem-spezifischer ODBC-Treiber bereitgestellt werden, der nun zwischen dem DBMS und der Anwendung vermittelt. Die Anwendung baut auf der Basis dieses Treibers eine Verbindung zum Datenbanksystem auf, übermittelt ein SQL-Statement und erhält gegebenenfalls das Ergebnistupel (Abbildung 2-14). Soll derselbe Zugriff auf ein anderes DBMS mit gleichem Datenbankschema erfolgen, so müssen keine neuen Zugriffsroutinen erzeugt werden, eine Änderung der ODBC-Schnittstelle ist ausreichend.

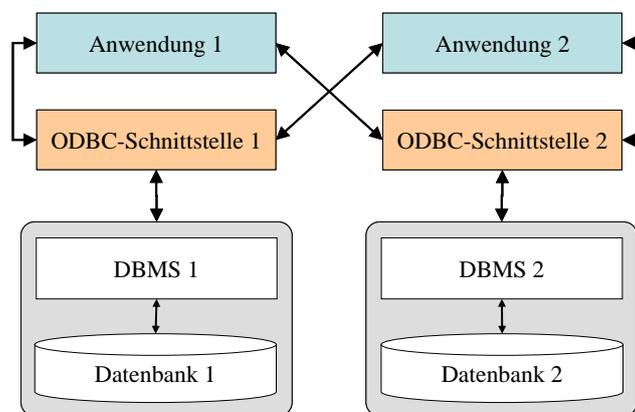


Abbildung 2-14: ODBC-Schnittstelle

Die ODBC-Schnittstelle ist ein Quasi-Standard für den Zugriff auf relationale Datenbanken. In vielen höheren Programmiersprachen (C++, Java) existieren Klassenbibliotheken, die bereits Methoden für die Erzeugung von SQL-Statements und den Zugriff auf Ergebnistupel implementieren. Obwohl die ODBC-Spezifikation von MICROSOFT erarbeitet wurde existiert die Schnittstelle auch für Unix Betriebssysteme.

2.4 Produktdatenmanagement

Aufgrund der zunehmenden Rechnerunterstützung in allen Phasen des Produktentwicklungsprozesses steigt auch die Zahl der zu speichernden und zu verwaltenden Daten. Des Weiteren wachsen die Anforderungen an integrierte Geschäftsprozesse und Informationsverfügbarkeit, sowohl unternehmensintern als auch in der Zusammenarbeit mit Partnern, Lieferanten und Kunden [ScBo05]. Daraus ergibt sich die Forderung, sowohl standortübergreifend als auch im Sinne des Concurrent und Simultaneous Engineering phasenübergreifend auf alle Daten, die im Laufe der VPE erzeugt werden, zugreifen zu können. Mit Hilfe der Konzepte des Produktdatenmanagements (PDM) sollen diese Aufgaben auf integrale Weise gelöst werden.

2.4.1 Grundlagen

Mit Hilfe von PDM sollen produktdefinierende, produktrepräsentierende und produktpräsentierende Daten und Dokumente als Ergebnis der Produktentwicklung gespeichert, verwaltet und in allen Phasen des Produktlebenszyklus zur Verfügung gestellt werden. Die Grundlage des PDM bildet ein integriertes Produktmodell. Das PDM-System implementiert softwaretechnisch die Methoden und Regeln des PDM und stellt Programmschnittstellen zu CAx-Software, ERP-Software und anderen Anwendungssystemen bereit.

Das PDM-System ermöglicht es dem Anwender, gezielt über bestimmte Kriterien nach archivierten Daten zu suchen. Die Grundlagen für diese Anforderungen bilden Methoden wie Klassifizierungen und Sachmerkmalelisten. Effiziente Suchalgorithmen ermöglichen kurze Antwortzeiten des Systems auf Anfragen. Um die Datenkonsistenz zu gewährleisten stellen PDM-Systeme – in Analogie zur PC-Netzwerkverwaltung – ein Berechtigungssystem zur Verfügung, mit dem die Rechte eines Benutzers in verschiedenen Rollen und Gruppen definiert sind.

Eine immer wichtiger gewordene Eigenschaft von PDM-Systemen ist die Bereitstellung von Workflows, um so die Prozesse im Unternehmen gezielt zu steuern und alle daran Beteiligten mit den richtigen Daten zu versorgen. Die im Voraus definierten Workflows werden von den Mitarbeitern des Unternehmens oder vom PDM-System bei bestimmten Ereignissen gestartet (z. B. im Fall der Freigabe eines Bauteils). Über ein systeminternes Nachrichtensystem werden die beteiligten Personen über ihre Aufgaben innerhalb dieses Workflows informiert. Die für die Erledigung dieser Aufgabe notwendigen Daten werden durch das PDM-System bereitgestellt und so verwaltet, dass keine Datenredundanz auftritt. Diese Funktionalität ist umso wichtiger, wenn es um die Integration externer Partner in einen unternehmensinternen Workflow geht. Da diese zumeist nicht an PDM-System des Unternehmens angebunden sind müssen geeignete Mechanismen vorhanden sein, um die Datenkonsistenz auch bei Datenweitergabe sicherzustellen.

2.4.2 Datenaustausch

Sowohl innerhalb eines Unternehmens als auch über Unternehmensgrenzen hinweg werden im gesamten Produktlebenszyklus Daten zwischen verschiedenen Anwendungssystemen ausgetauscht.

Das können sowohl geometriebeschreibende CAD-Daten (Bauteile, Baugruppen, Zeichnungen, etc.) aber auch administrative Produktdaten (Stücklisten, Produktstrukturen, Materialdaten, etc.) sein.

Der Austausch von Daten zwischen verschiedenen CAD-Systemen erfolgt zumeist über standardisierte, neutrale Schnittstellenformate (dateibasiert). Die verschiedenen, meist national entwickelten Austauschformate beschränken sich hierbei zumeist auf geometrische Daten (z. B. IGES, VDA-FS). Daher wurde im Rahmen der internationalen Normung eine Reihe von Standards entwickelt, mit denen ein allgemeines Beschreibungsmodell für sämtliche im Produktlebenszyklus anfallenden Produktdaten bereitgestellt wird (STEP, [ISO94]). Für die Datenbeschreibung wird dazu die Sprache EXPRESS (graphisch: EXPRESS-G) bereitgestellt, mit der beliebige Schemata (Applikationsprotokolle) erstellt werden können. Neuere Trends weisen jedoch darauf hin, dass für die Datenübertragung auf der Basis von STEP die Datenbeschreibungssprache XML vermehrt genutzt wird, da diese einen allgemein bekannten Standard darstellt und für die web-basierte Datenübertragung geeignet ist [ISO03] [LuPe04]. Somit eignet sich STEP für die Modellierung beliebiger Produktdaten und bietet somit die Basis für ein vollständig integriertes Produktmodell. Trotz der umfassenden Möglichkeiten von STEP existiert derzeit keine Möglichkeit, um parametrische und featurebasierte CAD-Modelle zwischen verschiedenen CAD-Systemen auszutauschen. Ergebnis eines Datenaustauschprozesses ist daher ein reines Geometriemodell, dessen Entstehungsgeschichte sich nicht mehr nachvollziehen lässt. Zwar lassen sich die Geometriemodelle durch nachträgliche Bearbeitungen anpassen und erweitern, dennoch werden durch den Austauschprozess die Vorteile, die sich durch die Modellierung mit parametrischen und featurebasierten CAD-Systemen ergeben, zunichte gemacht. Durch die unterschiedlichen Möglichkeiten der CAD-Systeme in Hinblick auf die zur Verfügung gestellten Funktionalitäten der Geometrieerstellung bleibt die Übertragung vollständiger CAD-Modelle im Blick aktueller Forschungen [ChMu02] [Bau04].

Im Rahmen des Datenaustausches zwischen verschiedenen PDM-Systemen müssen, im Gegensatz zum Datenaustausch zwischen CAD-Systemen, unterschiedliche Mechanismen berücksichtigt werden. Die Kommunikation der PDM-Systeme kann sowohl synchron (online) als auch asynchron (offline) erfolgen. Im Falle der synchronen Kommunikation wird direkt auf die Produktdaten zugegriffen, es erfolgt keine Datenreplikation. Dies erfolgt über standardisierte Schnittstellen wie XML oder die von der OMG (**O**bject **M**anagement **G**roup) auf CORBA-Basis (**C**ommon **O**bject **R**equest **B**roker **A**rchitecture) definierten PDM-Enablers. Beim asynchronen Datenaustausch wird das STEP-Produktmodell für den dateibasierten Datenaustausch in bestimmten Intervallen oder auf Anfrage verwendet. Probleme beim Datenaustausch zwischen PDM-Systemen ergeben sich vor allem durch den hohen Anpassungsgrad, verursacht durch die unternehmensspezifischen Prozesse, weniger durch die verwendeten Prozessoren für den Datenaustausch. Die Semantik der zu übertragenden Attribute in einem System A entspricht nicht zwangsläufig der Semantik im System B, wodurch es nicht möglich ist, eine einheitliche Schnittstelle zu schaffen. Die Lösung dieses Problems kann durch geeignete Vereinbarungen der Datenaustauschpartner erfolgen, indem eine entsprechen-

de Abbildung zwischen den beschreibenden Metadaten der jeweiligen Systeme vereinbart wird [NN09].

2.5 Produktkonfiguration

Ein zunehmender Trend in der Produktentwicklung ist die Zunahme der Variantenvielfalt bei gleichzeitiger Verringerung der Losgröße und Stückzahl [Ehr02]. Die Vermehrung der Teilevielfalt sowohl auf der Produkt- als auch auf der Teileebene führt zu immer komplexer werdenden Ablaufsteuerungsprozessen innerhalb der gesamten VPE. Seitens der Unternehmen wird daher versucht, die wirtschaftliche Bewältigung der Variantenvielfalt durch verschiedene Methoden zu bewältigen [Gra00]:

- **Komplexitätsreduzierung**
Basierend auf Analysen der Ist-Situation werden bestimmte Varianten, Komponenten oder auch Halbzeuge und Rohstoffe aus dem Produktspektrum eliminiert. Bewertungskriterium für diese Analysen ist die möglichst erfolgreiche Erfüllung der Kundenwünsche unter Berücksichtigung betriebswirtschaftlicher Kriterien.
- **Komplexitätsvermeidung**
Diese muss frühzeitig im Produktentwicklungsprozess erfolgen. Im Mittelpunkt steht die Erfüllung der Kundenanforderungen durch technische Merkmale unter Berücksichtigung einer möglichst hohen Wiederverwendbarkeit. Die Einführung der Plattformstrategie in der Automobilindustrie ist ein Beispiel für die erfolgreiche Umsetzung dieses Konzepts.
- **Komplexitätsbeherrschung**
Ein wichtiger Ansatz für die Beherrschung komplexer Produkte mit ausgeprägter Variantenvielfalt ist die gezielte Produktstrukturierung, um so eine unternehmerisch sinnvolle Produktvarianz zu erzielen. Im Bereich der Serienfertigung werden dazu häufig Baukastensysteme genutzt. Elemente eines Baukastensystems können Bauteile, Baugruppen, Software oder andere komplexe technische Gebilde sein, die zu Systemen unterschiedlicher Gestalt und Funktion zusammengesetzt werden. Die Elemente des Baukastensystems werden auch als „Lösungselemente“ bezeichnet.

Die Lösungselemente eines Baukastensystems können untereinander vielfältige Abhängigkeiten aufweisen. So kann das Vorhandensein eines Lösungselements im System dazu führen, dass andere Lösungselemente ausgeschlossen oder zwingend eingeschlossen werden müssen. Um diese z. T. sehr komplexen Abhängigkeiten sinnvoll beherrschbar zu machen werden Produktkonfiguratoren eingesetzt. Die Konfigurationsaufgabe kann dabei als Designaufgabe angesehen werden, wobei folgende Eigenschaften als typisch gelten [FeFr99]:

- Das konfigurierte Produkt besteht aus einer Menge von Bauteilen von vordefinierten Komponententypen und ihren Parametern
- Komponenten können mit anderen in vordefinierter Weise in Beziehung stehen

Das Ergebnis eines Konfigurationsvorgangs ist eine Auswahl von Komponenten mit ihren Parametern unter Berücksichtigung sowohl technischer als auch kundenspezifischer Vorgaben. Neben der Komponentenauswahl sollte auch die Komponententopologie (die Verbindungen der Komponenten) ein Ergebnis des Konfigurationsprozesses sein. Produktkonfiguratoren werden heutzutage in vielen Bereichen eingesetzt. Ein populäres Beispiel sind die von den Automobilherstellern im Internet angebotenen Konfigurationssysteme. Mit Hilfe dieser Systeme kann der Kunde ohne technische Vorkenntnisse ein interaktives Angebot für sein Wunschfahrzeug erstellen. Aber auch im Bereich der Entwicklung und Konstruktion werden zunehmend Konfigurationssysteme eingesetzt. Diese werden jedoch meist nicht durch den Kunden oder einen Vertriebsbeauftragten genutzt sondern dienen vielmehr der Erstellung und Prüfung technisch sinnvoller Lösungen für eine bestimmte Aufgabe. Durch diese Form der Konfiguration wird der Entwickler bzw. Konstrukteur von Routineaufgaben entlastet und eine möglichst hohe Problemlösungsqualität durch die Wiederverwendung bereits bewährter Teillösungen angestrebt [KöDu03]. Aufgrund der vielfältigen Verflechtungen mit unterschiedlichsten Unternehmensbereichen steht ein Konfigurationssystem niemals für sich allein. Das System muss vielmehr in die unternehmensinterne Informationsverarbeitung eingebettet sein. Abbildung 2-15 verdeutlicht diesen Umstand graphisch:

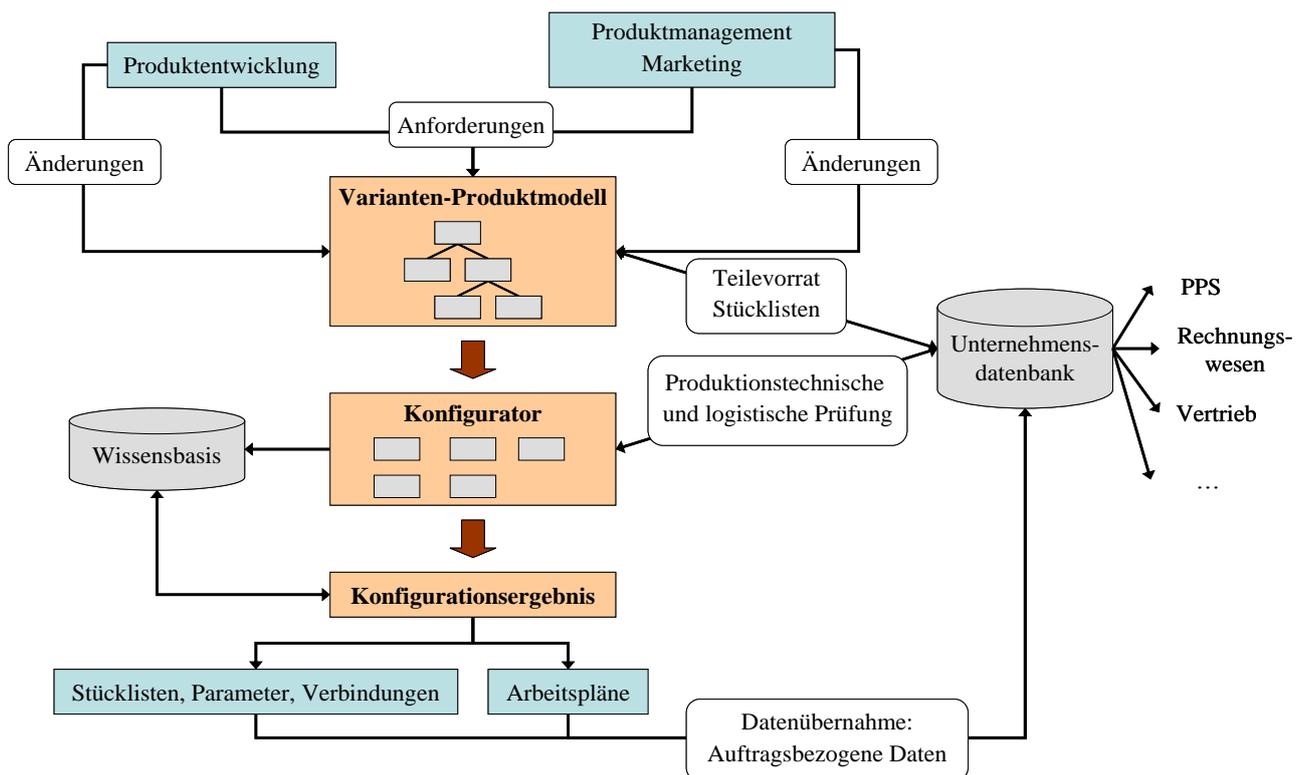


Abbildung 2-15: Integration der Produktkonfiguration in das betriebliche Informationsmanagement (in Anlehnung an [FeFr99])

2.6 Wissensintegration in das virtuelle Produkt

Die Bedeutung der Ressource „Wissen“ als entscheidender Produktionsfaktor rückt immer mehr in den Vordergrund wissenschaftlicher Betrachtungen. Für die erfolgreiche Entwicklung hochwertiger

Produkte benötigt der Konstrukteur ein umfangreiches und interdisziplinäres Wissen. Innerhalb der Produktentwicklung nimmt die Informationsbeschaffung einen hohen zeitlichen Anteil ein. Nach [San01] werden bis zu 50% der Zeit für die Wissensakquisition benötigt. Ein bedeutender Einflussfaktor ist vor allem die Vielfältigkeit dieses Wissens.

Die wissensbasierte Produktentwicklung (engl.: **Knowledge-based Engineering** – KBE) gilt daher als zukunftsweisende Methodik für eine innovative und effiziente Produktentwicklung. Die ersten Ansätze des KBE entstammen dem Forschungsfeld der Künstlichen Intelligenz (KI) und wurden primär entwickelt um Konstruktions- und Entwicklungsaufgaben automatisiert zu lösen, die mit herkömmlichen rechnergestützten Verfahren nicht zu erfassen waren [BlKn95]. In Analogie zu der Featuretechnik wird durch die Anwendung wissensbasierter Systeme die konstruktive Aufgabe auf einer semantisch höherwertigen Ebene durchgeführt. Das strategische Ziel von KBE ist die Überführung einer unstrukturierten, zufallsbedingten Wissensnutzung in eine strukturierte und zielführende Nutzung [RoSc01] [DaKö06]. Ein positiver Nebeneffekt von KBE ist die Entlastung des Konstrukteurs von sich wiederholenden Routinetätigkeiten. Damit wird ihm mehr Freiraum für kreative Prozesse gegeben, womit die Entwicklung innovativer Produkte gefördert wird.

2.6.1 Künstliche Intelligenz

Als Teilgebiet der Informatik beschäftigt sich die KI mit den Aktivitäten Planen, Verstehen, Erstellen, Sehen und Erkennen. Diese vorwiegend menschlichen Tätigkeiten sollen mit Hilfe der Datenverarbeitung simuliert werden, um so auf Intelligenzleistungen basierende Lösungen für Problemstellungen zu ermitteln. Die KI-Forschung untersucht dazu Techniken und Methoden der Wahrnehmung, Repräsentation und Verarbeitung von Wissen sowie Berechnungsverfahren des Schlussfolgerns und Handelns [Haa95] [GöSc00].

2.6.2 Wissen

Eine grundlegende Komponente für die Umsetzung Künstlicher Intelligenz ist das Verständnis von Wissen. Es existieren mehrere Definitionen des Begriffs. Der DUDEN definiert den Begriff Wissen wie folgt [Dud06]:

Wissen, das; -s: Gesamtheit der Kenntnisse, die jmd. [auf einem bestimmten Gebiet] hat

Diese allgemeine Definition lässt erkennen, dass Wissen an Personen gebunden ist. Damit ist es einer ständigen Variation unterworfen. Wissen beruht auf Erfahrungen, es wird erweitert, abgewandelt oder vermindert. Eine diesbezüglich deutlichere Definition von Wissen geben PROBST ET AL. [PrRa06]:

[Wissen ist] die Gesamtheit der Kenntnisse und Fähigkeiten, die Individuen zur Lösung von Problemen einsetzen. Wissen stützt sich auf Daten und Informationen, ist im Gegensatz zu diesen jedoch immer an Personen gebunden

Eine genauere Spezifikation der verschiedenen Wissensarten kann durch die Unterteilung in prozedurales und deklaratives Wissen erfolgen (Abbildung 2-16). Deklaratives Wissen beinhaltet das Wissen über Objekte und basiert auf Faktenwissen. Es kann sprachlich in der Form von Aussagen beschrieben werden. Prozedurales Wissen ist das Wissen über bestimmte Handlungsabläufe. Es basiert immer auf deklarativem Wissen.

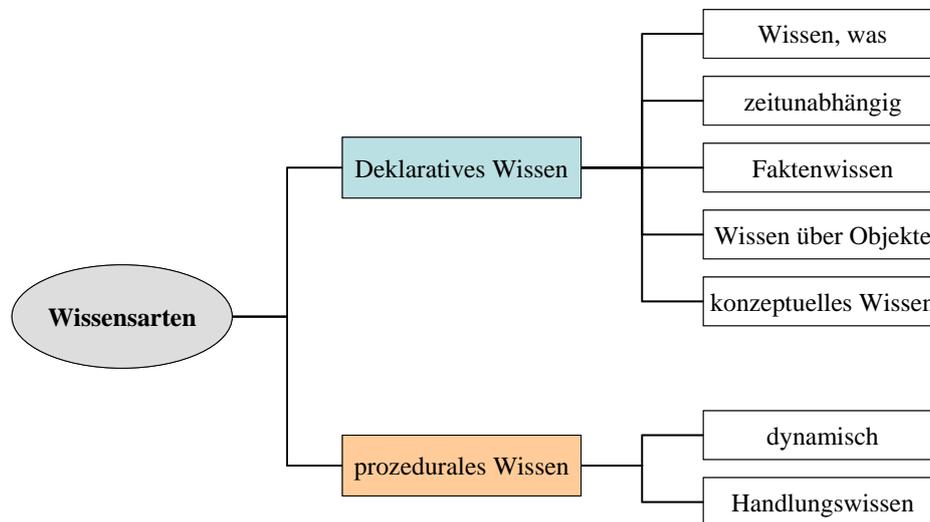


Abbildung 2-16: Wissensarten (in Anlehnung an [HaLu08])

Wissen lässt sich des Weiteren in explizites und implizites Wissen unterteilen. Explizites Wissen ist eine Form des bewussten Wissens, das bereits formalisiert ist. Da es eindeutig kodiert ist kann es durch Zeichen (z. B. Sprache oder Schrift) kommuniziert werden. Aus ingenieurwissenschaftlicher Sicht ist diese Form von Wissen u. a. in Richtlinien, Normen und Regelwerken zu finden. Im Gegensatz dazu existiert der Begriff des impliziten Wissens. Diese Form des Wissens ist nicht explizit formalisiert und daher auch nicht dokumentiert. Somit lässt sich implizites Wissen nur schwer oder gar nicht kommunizieren. Es basiert auf Erfahrungen und kann auch durch den Begriff der „Könnerschaft“ beschrieben werden. So kann z. B. ein erfahrener Ingenieur die für einen bestimmten Belastungsfall notwendige Dimension eines Bauteils beziffern, ohne jedoch bestimmt sagen zu können aus welchem Grund das Bauteil genau diese Beanspruchungen ertragen wird. Diese Schlussfolgerung basiert auf der Verknüpfung von Erfahrungswerten mit grundlegenden Kenntnissen der Festigkeitslehre und kann einer anderen Person nicht direkt vermittelt werden. Aus diesem Grund ist die rechnerinterne Verarbeitung dieser Wissensform nur eingeschränkt oder gar nicht möglich.

Aufbauend auf diesen beiden Begriffen wird in [NoTa97] das SECI-Modell (Socialisation, Externalization, Combination, Internalization) beschrieben. Im SECI-Modell wird neues Wissen durch eine kontinuierliche Transformation zwischen implizitem und explizitem Wissen generiert. Durch die aufeinander folgenden Prozesse „Externalisierung“ (implizit zu explizit), „Kombination“ (explizit zu explizit), „Internalisierung“ (explizit zu implizit) und „Sozialisation“ (implizit zu implizit)

wird Wissen innerhalb einer Organisation spiralförmig von individuellem Wissen auf höhere Organisationsstufen wie Personengruppen und ganze Firmen gehoben (Abbildung 2-17).

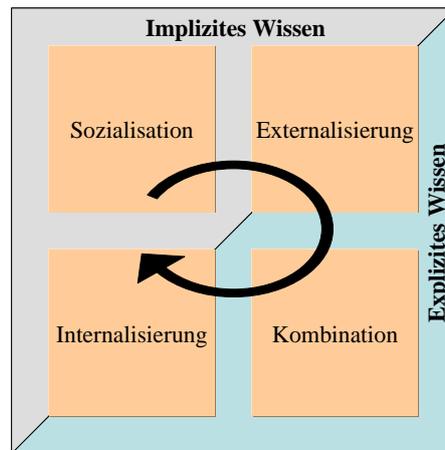


Abbildung 2-17: Wissenstransformation im SECI-Modell

Die gezielte Überführung von implizitem Wissen in explizites Wissen ist eine der Aufgaben des Wissensmanagement. Erst wenn Wissen in einer dokumentierten Form vorliegt ist es für weitere Personen nutzbar und kann damit auch rechnerintern verarbeitet werden. Am Anfang steht hier die Bereitstellung von Daten, die aus sinnvoll verknüpften Zeichen bestehen. Erst durch das Hinzufügen von Bedeutung werden aus diesen Daten dann Informationen. Das eigentliche Wissen entsteht, wenn diese Informationen in einem bestimmten Kontext verknüpft werden. NORTH stellt diesen Zusammenhang in seiner Wissenstreppe dar (Abbildung 2-18). Der zielgerichtete Umgang mit der Ressource Wissen wird hier als grundlegender Erfolgsfaktor für den strategischen Erhalt der Wettbewerbsfähigkeit eines Unternehmens gesehen. Dazu gehören sowohl ein geeignetes strategisches Wissensmanagement als auch ein operatives Daten-, Informations- und Wissensmanagement.

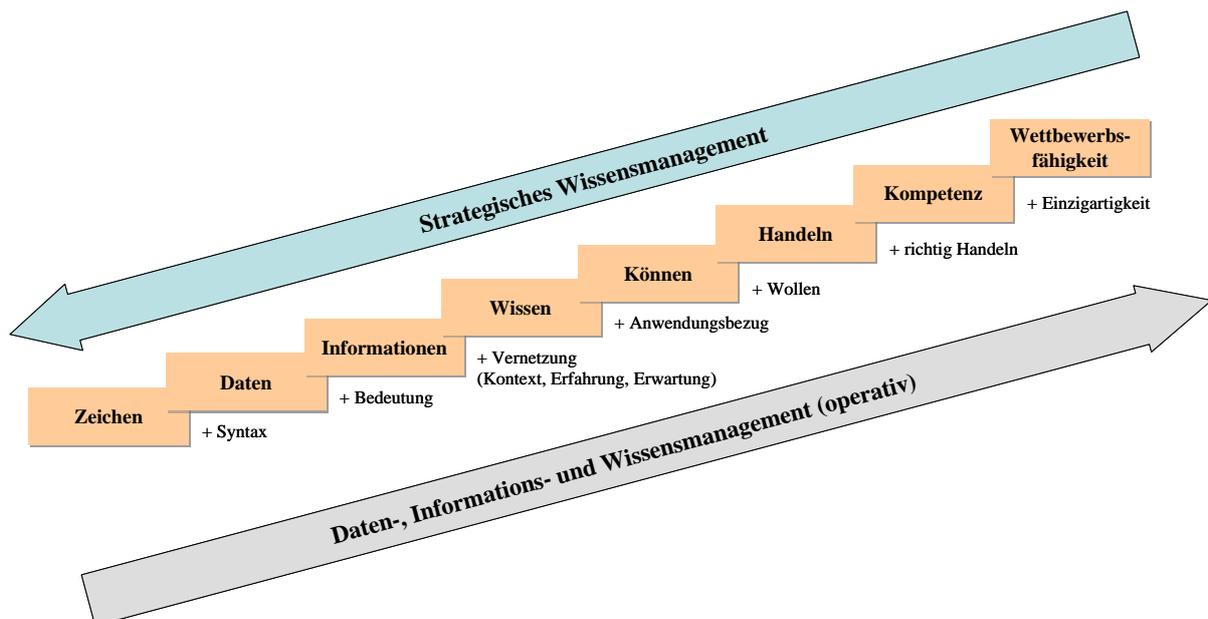


Abbildung 2-18: Wissenstreppe (nach [Nor05])

Die in Abbildung 2-18 suggerierte scharfe hierarchische Gliederung zwischen Daten, Informationen und Wissen ist jedoch nicht immer eindeutig durchführbar. Vielmehr handelt es sich um ein Kontinuum zwischen den beiden Polen *Daten* und *Wissen*, das durch folgende Schlagwörter gekennzeichnet ist (vgl. Abbildung 2-19) [Bod06].

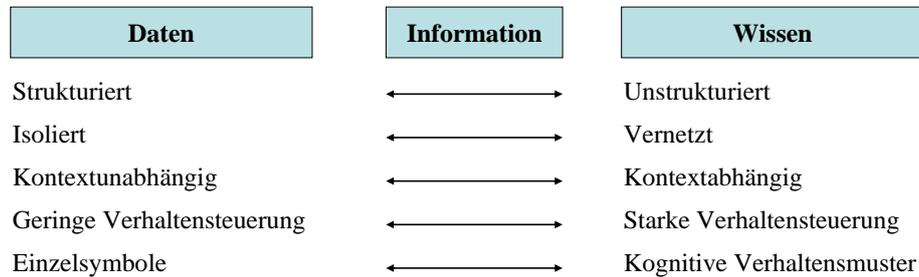


Abbildung 2-19: Differenzierung von Daten und Wissen

2.6.3 Wissensbasierte Systeme

Mit dem Begriff des wissensbasierten Systems werden Informationssysteme bezeichnet, in denen Wissen mit den Methoden der Wissensrepräsentation und der Wissensmodellierung nutzbar gemacht wird. Expertensysteme können als eine spezielle Form der wissensbasierten Systeme aufgefasst werden, deren Intention darin besteht, menschliches Problemlösungsverhalten auf der Basis von Expertenwissen nachzubilden. Die beiden Begriffe werden häufig synonym verwendet, wobei jedoch wissensbasierte Systeme auch für andere Zwecke eingesetzt werden können (maschinelles Lernen, Beweisen von mathematischen Sätzen, Spielen und automatisches Programmieren) [Era07]. Wissensbasierte Systeme bilden häufig neue Programmiersysteme mit alternativer Programmiermethodologie, da es mit konventionellen Programmieransätzen nur schwer oder gar nicht möglich ist, Algorithmen zu finden [Dil06].

2.6.4 Architektur von wissensbasierten Systemen

Der Mensch besitzt unterschiedliche Fähigkeiten, um zielgerichtet die Lösung zu einer Problemstellung zu ermitteln. Eine wichtige Rolle spielt hierbei der Erfahrungsschatz in Form von Fakten, Werten und Regeln. Dieser hier als Wissensbasis bezeichnete Erfahrungsschatz dient als Grundlage für Schlussfolgerungen, die dann zu einer Problemlösung führen. Diese Lösung ergänzt die Wissensbasis und trägt damit zur Lösung weiterer Problemstellungen bei. Damit ergibt sich ein wesentliches Merkmal wissensbasierter Systeme: die Trennung von Wissensbasis und Inferenzkomponente. Abbildung 2-20 verdeutlicht den strukturellen Aufbau eines wissensbasierten Systems.

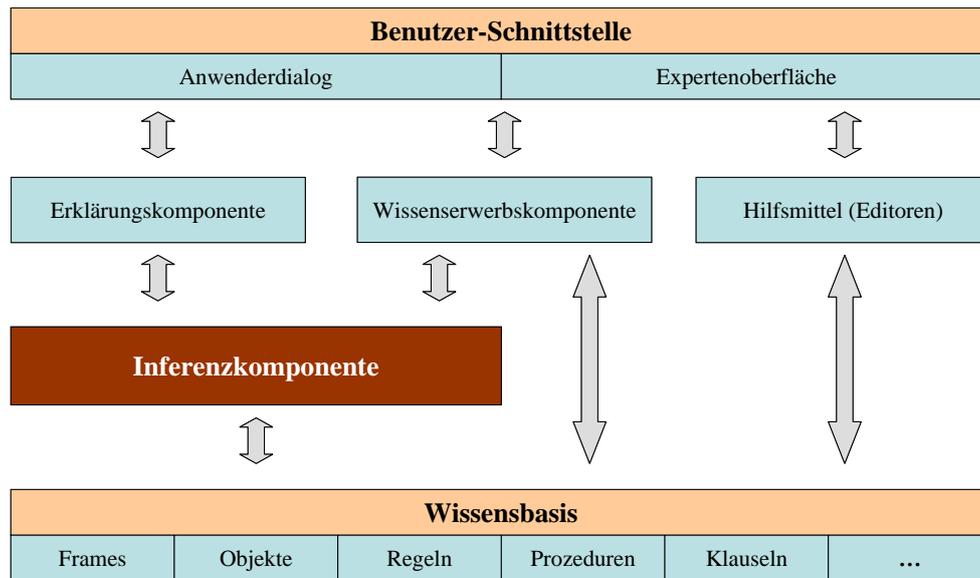


Abbildung 2-20: Struktureller Aufbau eines wissensbasierten Systems

Die *Wissensbasis* umfasst das für die Lösung erforderliche Expertenwissen einer bestimmten Domäne und bildet es in einer geeigneten Repräsentationsform ab. Sie ist die einzige systemspezifische Speicherkomponente und bietet damit den Vorteil der kompakten Pflege und Erweiterbarkeit der Wissensinhalte des Systems.

Die *Inferenzkomponente* wendet bestimmte Strategien zur Handhabung des in der Wissensbasis gespeicherten Wissens an, um zu einer Lösung zu kommen. Sie verfügt also über Wissen über die Verarbeitung von Wissen.

Die Aufgabe der *Erklärungskomponente* ist es, aktuelle Zustände, Entscheidungen und Regeln übersichtlich und transparent darzustellen. Die Funktionalität der Erklärungskomponente beinhaltet u. a. die Erklärung bestimmter Begriffe, die Bereitstellung von Informationen über aktuelle Werte und auch das Überprüfen und Begründen angewendeter Regeln.

Mit Hilfe der *Wissenserwerbskomponente* und anderer *Hilfsmittel (Editoren)* kann Wissen in der Wissensbasis gespeichert und zugänglich gemacht werden. Eine grundlegende Aufgabe des Bearbeiters (Experten) besteht darin, aus vorhandenen Quellen (Normen, Richtlinien, Kataloge, Expertenbefragungen) Wissen zu extrahieren, aufzubereiten und es dann widerspruchsfrei in die Wissensbasis einzupflegen. Idealerweise wird er dabei durch geeignete Komponenten unterstützt und geführt.

Eine wesentliche Komponente des wissensbasierten Systems ist die *Benutzerschnittstelle*, da sie einen großen Einfluss auf die Akzeptanz des Anwenderkreises (Experten und Benutzer) hat. Sie hat die Aufgabe, den Anwender durch geeignet gestaltete Dialoge bei der Anwendung des Wissens zu unterstützen und sollte ihn bei der Eingabe von benötigten Informationen leiten. Das vom Benutzer eingegebene fallspezifische Wissen formuliert dann die eigentliche Aufgabenstellung, die durch das wissensbasierte System gelöst werden soll. Ein wesentliches Element ist dabei die Darstellung von

Sachverhalten in Form graphischer und alphanumerischer Darstellungen (Prinzipiskizzen, graphische Erläuterungen, Erklärungstexte). Dem Experten sollte die Benutzerschnittstelle Möglichkeiten bieten, komfortabel die Wissensanwendung zu testen und gegebenenfalls zu modifizieren. Entsprechende Hinweise für die Gestaltung derartiger Dialoge finden sich in [VDI03b].

2.6.5 Formen der Wissensrepräsentation

Die Wissensrepräsentation dient der strukturierten Darstellung des notwendigen Wissens und ist eng mit der Art der Schlussfolgerung durch die Inferenzkomponente gekoppelt. Grundsätzlich können Daten innerhalb einer Wissensrepräsentationsform gekoppelt oder ungekoppelt vorliegen. Es kann zwischen strenger Hierarchie (ein Nachfolger hat höchstens einen Vorgänger), multipler Hierarchie (ein Nachfolger kann mehrere Vorgänger haben) und Netzwerken (Zuordnung durch Schleifen) unterschieden werden [Pup91]. Aufgrund der zentralen Bedeutung der Wissensrepräsentation sowohl für die Leistungsfähigkeit des wissensbasierten Systems als auch für die Pflegbarkeit und Erweiterbarkeit der Wissensbasis werden nach [GoFr90] folgende Anforderungen an die gewählte Repräsentationsform gestellt:

- **Verarbeitbarkeit**
Die Wissensrepräsentation sollte es ermöglichen, dass durch logisches Verknüpfen aus bestehendem Wissen auf neues Wissen geschlossen werden kann.
- **Flexibilität**
Der Repräsentationsansatz sollte dazu geeignet sein, Wissen aus möglichst vielen Anwendungsbereichen darzustellen. Andererseits sollen unabhängig vom Anwendungsgebiet unterschiedliche Aufgabenstellungen effizient verwirklicht werden können.
- **Modularität**
Die Wissensbasis eines intelligenten Systems soll möglichst einfach modifizierbar und erweiterbar sein. Eine geeignete Lösung stellt eine modulare Wissensbasis dar. Deklaratives Wissen ist unabhängig von prozeduralem Wissen und ist daher sehr gut geeignet für den Aufbau einer modularen Wissensbasis.
- **Verständlichkeit**
Der Inhalt der Wissensbasis muss gegenüber dem bearbeitenden Experten und dem Benutzer gleichermaßen leicht verständlich darstellbar sein. Die Art der Darstellung ist jedoch nicht für beide Gruppen gleich. Für den Benutzer muss nachvollziehbar sein, wie ein System zu einer bestimmten Entscheidung gekommen ist. Der Experte muss im Gegensatz dazu in der Lage sein, den Inhalt der Wissensbasis zu pflegen und zu überschauen.
- **Darstellbarkeit unsicheren Wissens**
Nicht immer kann die Wirklichkeit durch Fakten eindeutig beschrieben werden. Vielmehr treffen bestimmte Wahrscheinlichkeiten zu, dass eine bestimmte Lösung richtig ist. Daher

müssen intelligente Systeme in der Lage sein, mehrdeutige Aussagen zu verarbeiten. Diese Unsicherheiten können auf mehrere Arten entstehen:

- Inhärente Unsicherheit der Information
- Unvollständigkeit der Information
- Unsicherheit von Schlussfolgerungen
- Zusammenfassung von Informationen, die aus mehreren, eventuell einander widersprechenden Quellen stammen

Im Bereich der Konstruktion, Planung und Konfiguration werden im Allgemeinen die folgenden Wissensrepräsentationsformen angewendet:

- Constraintbasierte Repräsentation
- Regelbasierte Repräsentation
- Objektorientierte Repräsentation

Mit der Hilfe von Constraints in der *Constraintbasierten Wissensrepräsentation* können Relationen zwischen Variablen eines Objekts abgebildet werden. Constraints eignen sich zur Darstellung von Randbedingungen, die bei der Lösung des Problems unbedingt eingehalten werden müssen. Jeder neue Constraint schränkt den Lösungsraum zusätzlich ein [Pup91]. Durch die Bildung eines Constraint-Netzwerkes können, analog zur Berechnung eines Gleichungssystems, die fehlenden Variablen berechnet werden (siehe Abbildung 2-21). Hierzu wird, ausgehend von einer Anfangsbelegung der Eingangsvariablen, der Lösungsraum immer weiter eingeschränkt. Constraints eignen sich daher zur Konsistenzprüfung und Erzeugung neuer Variablenbelegungen. Constraints werden häufig nicht als alleinige Darstellungsform genutzt sondern in weitere Wissensrepräsentationsformen integriert [Haa95].

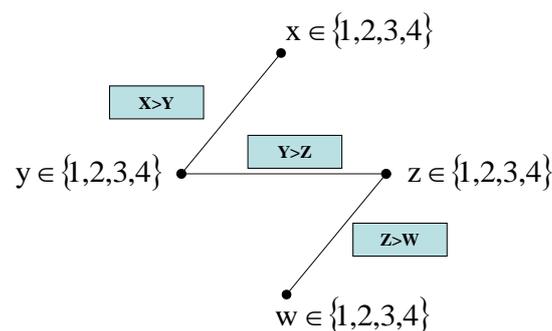


Abbildung 2-21: Constraint-Netz

Die *Regelbasierte Wissensrepräsentation* ist eine sehr weit verbreitete Repräsentationsform in vielen Expertensystemen, da sie einen guten Kompromiss zwischen Verständlichkeit der Wissensdarstellung und formalen Ansprüchen darstellt [BeKe08]. Regeln sind formalisierte Konditionalsätze der Form WENN (IF) A DANN (THEN) B. Der „WENN“-Teil einer Regel wird als Prämisse oder Antezedenz, der „DANN“-Teil als Konklusion oder Konsequenz bezeichnet. Entspricht die

Aussage B einer Aktion spricht man auch von Produktionsregeln. Die folgende Aufzählung zeigt einige mögliche Aktionen, die durch eine Regel gestartet werden können:

- Einfache Wertzuweisung an ein Attribut
- Wertetransfers zwischen Attributen
- Tabellenzugriffe
- Termauswertungen
- Zählerfunktionen
- Prozeduraufrufe

Wenn die Antezedenz einer Regel erfüllt ist so spricht man auch davon, dass die Regel „feuert“. Ein wesentliches Merkmal von regelbasierten Systemen ist, dass die Reihenfolge der auszuführenden Aktionen nicht unbedingt vorher bekannt sein muss, da sie von der Übereinstimmung vordefinierter Bedingungen abhängt. Somit weisen regelbasierte Systeme keinen unmittelbar prozeduralen Charakter auf, was sie von klassischen Programmen mit vordefinierten Algorithmen unterscheidet.

In einer *Objektorientierten Wissensrepräsentation* werden sämtliche Aussagen über ein Objekt in einer Datenstruktur zusammengefasst. Diese Form der Wissensrepräsentation folgt dem gleichnamigen Programmierparadigma, wurde jedoch unabhängig davon entwickelt [Kur92]. Zu den wesentlichen Formen gehören *semantische Netze*, *Objekt-Attribut-Wert-Tripel* und *Frames*.

In semantischen Netzen wird Wissen auf den Grundlagen von Netzwerkstrukturen abgebildet. Ein semantisches Netz ist ein gerichteter Graph, der aus einer Menge von Knoten und Kanten besteht (Abbildung 2-22). Die Knoten des Graphen stellen die abzubildenden Sachverhalte dar (Abstrakte Begriffe, Objekte, Konzepte, Ereignisse), die Kanten beschreiben die Beziehungen zwischen den Knoten. Häufig werden hierarchische Beziehungen („ist ein“, „hat“) genutzt. Mit Hilfe dieser hierarchischen Beziehungen lässt sich eine Vererbungsstruktur aufbauen. Eigenschaften eines in der Struktur höher liegenden Knotens können innerhalb der Hierarchie weitergegeben werden.

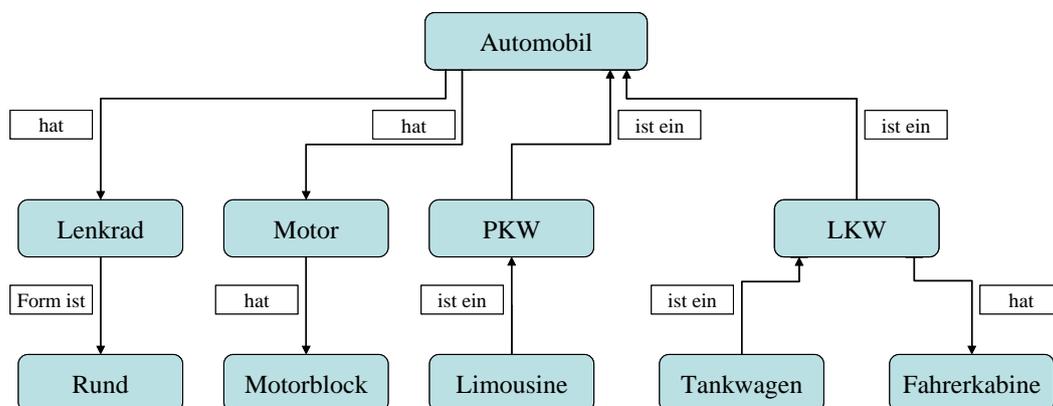


Abbildung 2-22: Semantisches Netz [Kur92]

Semantische Netze werden selten im Bereich der Expertensysteme eingesetzt, ein Hauptanwendungsgebiet ist die Sprachverarbeitung.

Objekt-Attribut-Wert-Tripel werden, ebenso wie semantische Netze, genutzt, um den Informationsgehalt von Fakten darzustellen. Objekte können Gegenstände oder Begriffe sein, die ein oder mehrere Attribute besitzen. Diese Attribute beschreiben die Eigenschaften des Objekts. Werte sind die spezifischen Ausprägungen eines Attributs, wobei ein Wert wiederum ein Objekt sein kann. In Abbildung 2-23 ist ein Objekt-Attribut-Wert-Tripel als semantisches Netz dargestellt.

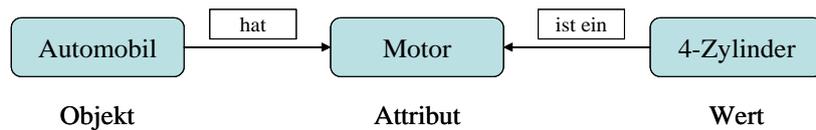


Abbildung 2-23: Objekt-Attribut-Wert-Tripel als semantisches Netz

Der Begriff *Frame* wurde erstmals von MINSKY eingeführt [Min75]. Diese Form der nicht-logikbasierten Wissensrepräsentation ähnelt sehr der objektorientierten Sichtweise bei der Programmierung. Ein Frame ermöglicht die Beschreibung eines Objekts und sämtlicher Informationen, die mit diesem Objekt verbunden sind. Ein Frame repräsentiert eine Klasse von Objekten, eine Unterklasse, eine Instanz oder ein selbstständiges Objekt. Jeder Frame enthält eine beliebige Anzahl von „Slots“, in denen die Eigenschaften und Prozeduren des Frames gespeichert werden (Abbildung 2-24). Bei der Instanzierung des Frames müssen die Slots mit Einträgen belegt werden. Dazu können Default-Werte angegeben werden, die bei der Instanzierung übernommen werden, falls kein anderer Eintrag vorgenommen wird. Slot-Einträge können auch durch Prozeduren, die bei einer beliebigen Gelegenheit gestartet werden, vorgenommen werden. Ebenso können Zeiger auf andere Frames (Sub-Frames oder übergeordnete Frames) als Eintrag dienen. Mit dieser Vorgehensweise wird ein Vererbungskonzept implementiert und eine Frame-Hierarchie ermöglicht, wodurch die Anzahl an Redundanzen in der Wissensbasis minimiert werden kann.

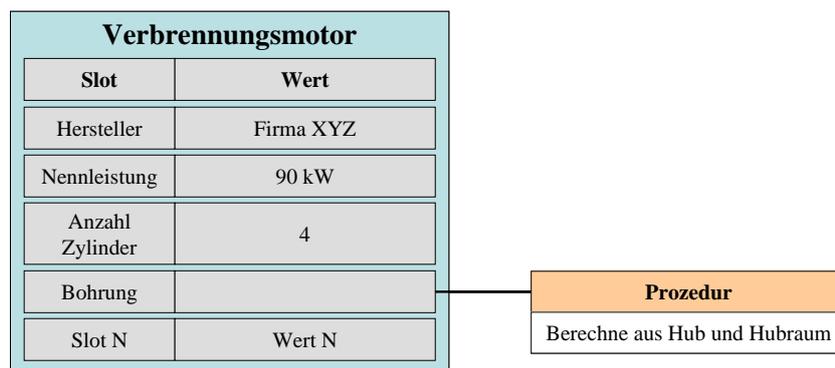


Abbildung 2-24: Frame

Eine klare Abgrenzung zwischen den verschiedenen Formen der Wissensrepräsentation ist nicht immer möglich oder gewollt. Vielmehr bedingen viele der Repräsentationsformen einander und

entstehen durch die Verknüpfung und Erweiterung elementarer Repräsentationsformen wie z. B. der Aussagenlogik (siehe Abbildung 2-25).

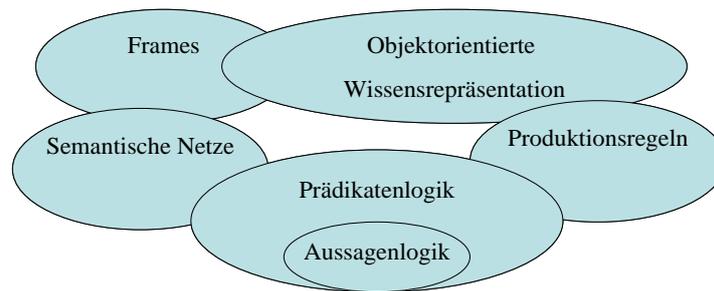


Abbildung 2-25: Verknüpfung unterschiedlicher Formen der Wissensrepräsentation

2.6.6 Inferenzbildung und Ablaufsteuerung

Die Repräsentation von Wissen in einer formalisierten Struktur ist die Ausgangsbasis für das Suchen von Problemlösungen. Diese können – auch in Abhängigkeit von der Art der Wissensrepräsentation – durch verschiedene Wissensverarbeitungsstrategien gefunden werden. Das wissensbasierte System erreicht dies durch Schlussfolgern. Ist der Prozess des Schlussfolgerns elementar und formalisierbar wird auch der Begriff „Inferenz“ genutzt.

Der Prozess der Inferenzbildung soll hier am Beispiel regelbasierter Systeme aufgezeigt werden, da diese im Rahmen dieser Arbeit eine wichtige Rolle spielen. Weitere Möglichkeiten der Inferenzbildung, vor allem aus dem Bereich der logikbasierten Wissensverarbeitung und des Case-Based Reasoning (engl.: fallbasiertes Schließen) können der Literatur entnommen werden (siehe [Pup91] [Kur92] [BeKe08]).

In regelbasierten Systemen wird für die Inferenzenbildung zumeist der Modus Ponens verwendet [Har03]. Ist eine Regel der Form „Wenn A dann B“ gegeben so folgt, dass B wahr ist, wenn A wahr ist. Eine weitere Möglichkeit ist die Anwendung des Modus Tollens (Wenn A falsch ist muss auch B falsch sein), der auch in Kombination mit dem Modus Ponens angewendet werden kann.

Neben der reinen Inferenzleistung ist für die Abarbeitung des Gesamtprozesses der Lösungssuche auch eine Steuerungsleistung zu erbringen. Die Aufgabengebiete der Steuerungskomponente lauten wie folgt:

- Es muss ein Startpunkt ermittelt werden (Beginn des Schlussfolgerungsprozesses)
- Es muss entschieden werden, zu welchem Zeitpunkt eine Regel feuert
- Es muss ein Abbruchkriterium ermittelt werden (Terminierung des Schlussfolgerungsprozesses)

Eine grundsätzliche Unterscheidung der Steuerungskomponenten ergibt sich durch den ermittelten Startpunkt für die Lösungssuche. Wird von einem bestimmten Ziel ausgegangen, dessen Gültigkeit zu beweisen ist, so spricht man von einer zielorientierten oder auch hypothesenorientierten Strategie (Rückwärtsverkettung). Wird die aktuelle Datenbasis als Ausgangspunkt gewählt, um

davon ausgehend eine Lösung zu finden, so handelt es sich um eine datengetriebene oder auch prämissenorientierte Strategie (Vorwärtsverkettung). Beide Strategien beinhalten Suchstrategien für das Auffinden von Regeln, die zu feuern sind, die sich jeweils in eine Breitensuche und eine Tiefensuche kategorisieren lassen.

Im Fall der *Vorwärtsverkettung* wird, ausgehend von der aktuellen Situation der Wissensbasis, eine Konfliktmenge bestehend aus Regeln mit gültigem Antezedenzteil gebildet. Das Lösungsverfahren ist beendet, wenn keine Regel mehr feuern kann oder ein vorgegebenes Abbruchkriterium erfüllt ist. Wie bereits erwähnt können für das Ermitteln der Konfliktmenge unterschiedliche Suchstrategien angewendet werden. Bei einer Breitensuche feuern alle Regeln einer Ebene nacheinander, die in der vorangegangenen Konfliktmenge existieren. Bei der Tiefensuche feuert jeweils die aktuell gefundene Regel der Konfliktmenge. Es kommen im Fall der Vorwärtsverkettung also auch Regeln zur Auswertung, die eigentlich nicht für das eigentliche Ziel relevant sind.

Durch eine Rückwärtsverkettung wird das Auswerten von Regeln, die keinen Einfluss auf den Nachweis der zu erbringenden Lösung haben, vermieden. Der Lösungsalgorithmus startet bei einem vorgegebenen Ziel und wertet dazu sukzessive den Konsequenzteil der Regeln aus, um diese dann in die Konfliktmenge aufzunehmen. Dann wird der Antezedenzteil dieser Regeln ausgewertet, um die darin enthaltenen Aussagen zu ermitteln. Dann wird die Auswertung der Regel unterbrochen, die unbekanntes Antezedenzen werden zu Zwischenzielen und es werden die Regeln gesucht, die diese Zwischenziele im Konsequenzteil enthalten. Auch hier kann eine Tiefensuche oder Breitensuche angewandt werden (vgl. Abbildung 2-26).

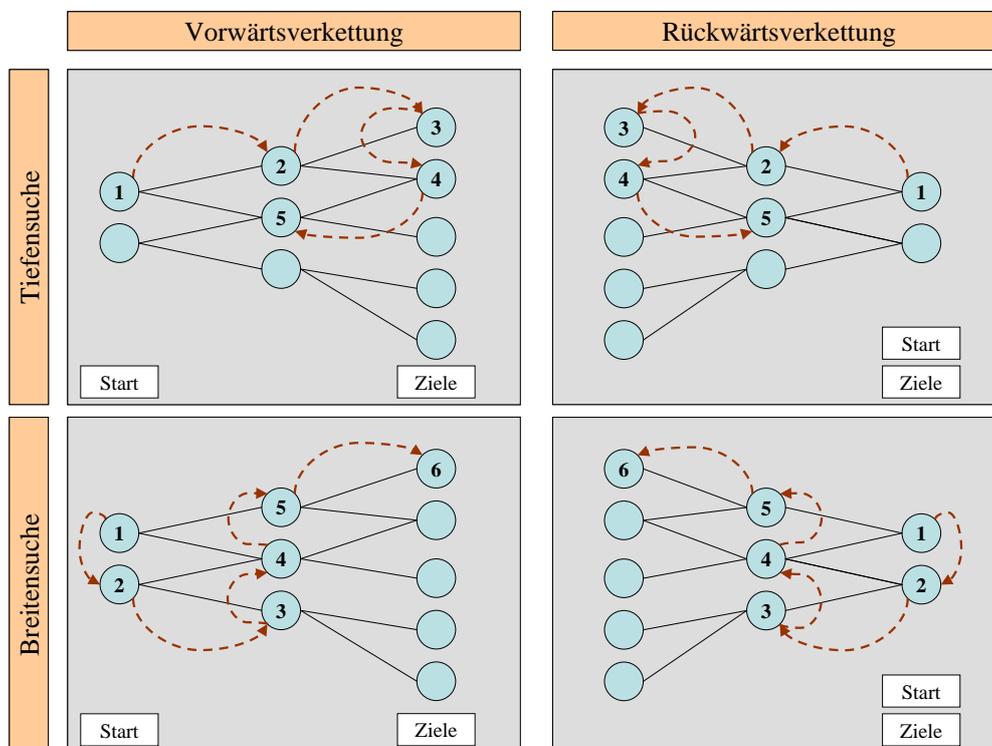


Abbildung 2-26: Ablaufsteuerung regelbasierter Lösungssuche

Die verwendete Suchstrategie hat Einfluss auf die Gültigkeitsdauer des Wissens während des Lösungsprozesses. Im Fall des monotonen Schließens bleiben die Fakten während des gesamten Lösungsprozesses dauerhaft gültig. Nicht-monotones Schließen tritt dann ein, wenn einmal bewiesene Fakten auch widerrufen werden können. Dazu wird bei Auswahl einer Regel ein Rücksetzpunkt gesetzt, der die zu diesem Zeitpunkt gültigen Fakten enthält. Im Laufe der Inferenz kann zu diesem Punkt dann zurückgekehrt werden, um mit einer anderen Regel weiterzuarbeiten. Sowohl die Vorwärtsverkettung als auch die Rückwärtsverkettung erlauben nicht-monotones Schließen.

2.6.7 Wissensbasierte Konstruktionssysteme

Wie bereits erwähnt soll mit Hilfe von KBE-Methoden eine zielführende und gerichtete Wissensnutzung innerhalb der VPE erreicht werden. Ein wesentliches Merkmal von KBE-Systemen ist hierbei die enge Verknüpfung zur geometrischen Ausprägung des Produkts, weshalb KBE-Systeme einen Sonderfall der wissensbasierten Systeme darstellen. Wie auch bei den wissensbasierten Systemen weisen KBE-Systeme eine Trennung zwischen der eigentlichen Wissensrepräsentation und dem Inferenzmechanismus auf. Streng genommen sind daher Programme, die mit der API eines CAD-Systems erstellt wurden keine KBE-Systeme, da hier der Inferenzmechanismus und die Wissensrepräsentation nicht voneinander getrennt sind. Dennoch werden auch diese Systeme in dieser Arbeit den KBE-Systemen zugeordnet.

Die folgende Auflistung gibt einen Überblick über die Eigenschaften, die ein KBE-System aufweisen sollte [ChPr07]:

- **Objektorientierte Architektur**
Probleme sollen auf der Basis einer objektorientierten Sprache beschreibbar sein. Dazu gehört die Implementierung von Eigenschaften objektorientierter Systeme (abstrakte Datentypen, Vererbung, Datenkapselung, Polymorphie)
- **Unterstützung von Inferenzmethoden wissensbasierter Systeme**
Dazu gehört im Fall der regelbasierten Systeme die Implementierung von Vorwärts- und Rückwärtsverkettung.
- **Dynamische Modellierung**
Das Modell muss auch nach der Instanzierung modifizierbar sein. Dazu gehört die Addition oder Subtraktion neuer Objekte und Eigenschaften.
- **Constraint-Mechanismen**
Zu den benötigten Constraint-Mechanismen gehört die Implementierung bidirektionaler Assoziativität und eine anforderungsgetriebene Attributauswertung (demand-driven). Durch die bidirektionale Assoziativität wird sichergestellt, dass Änderungen an einem Objekt oder einer Eigenschaft auf alle von diesem Objekt abhängigen Objekte übertragen werden. Die anforderungsgetriebene Attributauswertung stellt sicher, dass die Werte einer Eigenschaft

erst dann ausgewertet werden, wenn sie von einem Objekt aufgrund einer Änderung angefordert werden.

- Verknüpfung mit einem geometrischen Modellierkern
Das System muss mit einem leistungsfähigen Geometriekern verknüpft sein. Dies kann sowohl in Form einer Verknüpfung oder Einbettung innerhalb eines proprietären CAD-System als auch durch die direkte Implementierung eines der bekannten Geometriekerne (ACIS, PARASOLID, etc.) erfolgen. Das KBE-System sollte dazu in der Lage sein, geometrische Zusammenhänge auswerten zu können.
- Entwicklung graphischer Benutzeroberflächen
Das System sollte die Möglichkeit bieten, graphische Benutzeroberflächen für die Interaktion mit dem Anwender zu erzeugen.
- Verknüpfungen zu anderen Werkzeugen und Anwendungen
Durch definierte Schnittstellen soll eine systeminterne Verknüpfung zu anderen in der VPE genutzten Werkzeugen wie z. B. Datenbanken, Tabellenkalkulationen oder FE-Software ermöglicht werden.

Eine Einteilung von KBE-Systemen kann anhand der Integrationstiefe in CAD-Systemen erfolgen. Es kann hier grob zwischen Stand-Alone-Systemen, angepassten Systemen und integrierten Systemen unterschieden werden. Jedoch sind die Grenzen zwischen den verschiedenen KBE-Systemen teilweise fließend. Dies liegt vor allem daran, dass die CAD-System-Hersteller und die damit verbundenen Softwarepartner verschiedene Strategien bei der Implementierung wissensbasierter Komponenten verfolgen.

Stand-Alone-Systeme erzeugen die Produktgeometrie mit Hilfe des eingebauten Geometriekerns oder durch die formale Beschreibung der Geometrie auf der Basis von Parametersätzen oder Designvorschlägen. Dadurch ergibt sich der Vorteil, dass die Wissensrepräsentation unabhängig vom verwendeten CAD-System ist und somit bei einem Systemwechsel die Übertragbarkeit des im System gespeicherten Wissens abgesichert ist. Durch die indirekte Kopplung an das eigentlich geometrieverarbeitende CAD-System ergibt sich jedoch der Nachteil der Schnittstellenproblematik. Die Übertragung von im KBE-System erstellter Geometrie muss über neutrale Datenformate erfolgen, was u. U. zu Fehlern aufgrund fehlerhaft importierter Geometrien führen kann [RiGo04]. Des Weiteren kann kein Nutzen aus den Vorteilen einer parametrischen und featurebasierten Modellierung gezogen werden. In Hinblick auf die Folgeprozesse, die auf dem CAD-Modell aufbauen, ergeben sich weitere Nachteile durch die mangelnde Integration produktspezifischen Wissens. Die durch die Featuretechnologie gemachten Fortschritte im Bereich der Verknüpfung der unterschiedlichen Phasen des Produktentwicklungsprozesses werden so rückgängig gemacht.

Angepasste Systeme kommunizieren mit dem CAD-System über die API und nutzen so system-spezifische Funktionalitäten, um ein Produktmodell auf der Basis des im KBE-System gespeicherten Wissens zu erzeugen oder zu manipulieren. Die API bietet häufig auch die Möglichkeit der ge-

ometrischen Rückkopplung in das KBE-System (Messungen, Geometrieanalysen, Topologiebewertungen). Ebenso kann die Möglichkeit der Wissensintegration in das 3D-CAD-Modell genutzt werden, wenn die API und damit das CAD-System Knowledgeware-Features unterstützen. Ein grundsätzlicher Vorteil der angepassten Systeme besteht – wie auch bei den Stand-Alone Systemen – in der Möglichkeit, das Wissen in anderen CAD-Systemen weiterzuverwenden. Die verwendeten Geometriefunktionalitäten müssen dann an das neue CAD-System angepasst werden. Daraus ergibt sich jedoch auch der Nachteil der meist nur eingeschränkten Portabilität. Aufgrund der Tatsache, dass viele CAD-Systeme sowohl einen unterschiedlichen Funktionsumfang als auch recht unterschiedliche Möglichkeiten seitens der API aufweisen, muss bei der Umsetzung geometrischer Funktionalitäten meist auf Basisfunktionen zurückgegriffen werden [Bau04]. Somit wird die Leistungsfähigkeit des KBE-Systems insgesamt eingeschränkt.

Ein im CAD-System *integriertes KBE-System* weist aufgrund seiner engen Kopplung die größten Möglichkeiten in den Bereichen Geometriemanipulation und Wissensintegration auf. Im Gegensatz zu einem angepassten System wird hier das Produktwissen direkt in das CAD-Modell integriert, womit alle vom CAD-System bereitgestellten Wissensrepräsentationsformen genutzt werden können. Durch die zentrale Wissensspeicherung wird der Wartungsaufwand im Fall einer notwendigen Systemanpassung reduziert. Somit wird auch das Risiko einer Inkonsistenz von Wissensquelle zu Wissensrepräsentation reduziert. Die Vorteile dieser Systemart weisen jedoch auch direkt auf die Nachteile hin. Wird das CAD-System ausgetauscht, so muss auch das in den Modellen gespeicherte Wissen transferiert werden. Auch im Bereich der integrierten KBE-Systeme gibt es hier noch keine Lösung, eine verlustfreie Datenübertragung auf der Basis eines einheitlichen Produktmodells zu generieren.

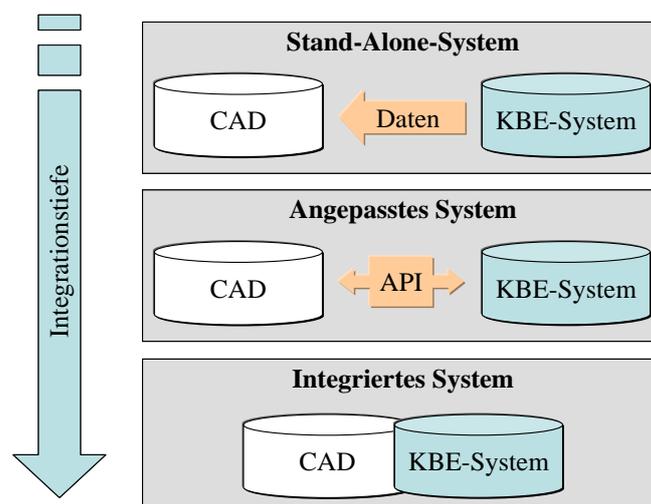


Abbildung 2-27: Integrationstiefe von KBE-Systemen

3 Ansätze zur Wissensintegration im CAD-Umfeld

In diesem Kapitel sollen die derzeit aktuellen Forschungsansätze und abgeschlossenen Arbeiten in den Bereichen KBE, CAD, Automatisierung im Umfeld der VPE und der Wissensrepräsentation in CAD-Modellen erläutert werden. Aufgrund der Vielzahl der unterschiedlichen Forschungs- und Anwendungsfelder können hierbei nicht alle Ansätze berücksichtigt werden. Daher werden hier nur die für diese Arbeit wesentlichen Konzepte und Modelle detaillierter betrachtet.

3.1 CommonKADS

CommonKADS (**Common Knowledge Acquisition and Documentation Structuring**) ist eine Methodik für die Entwicklung von wissensbasierten Systemen. Die Entwicklung der KADS-Methodologie begann 1983 innerhalb des europäischen ESPRIT-Forschungsprojekts P-1098 [WSB92] (KADS I) und wurde bis 1994 innerhalb des Nachfolgeprojekts P-5248 weiterentwickelt [ScAk99] (KADS II oder CommonKADS). CommonKADS wird als De-facto-Standard für die Entwicklung wissensbasierter Systeme angesehen [AbBe02] und deckt viele Teilaspekte eines Entwicklungsprojekts für ein wissensbasiertes System ab. Dazu gehören das strategische Management, das Projektmanagement, die Wissensakquisition und die Softwareentwicklung. Der grundlegende Ansatz von CommonKADS geht davon aus, dass ein wissensbasiertes System immer im Anwendungskontext zu sehen ist. Daher müssen auch organisatorische und individuelle Gegebenheiten auf die Wissensakquisition und Wissensverteilung in einer Organisation wirken und berücksichtigt werden.

Mit Hilfe von CommonKADS soll eine vollständige Entwicklungsmethodik für die Erstellung von wissensbasierten Systemen bereitgestellt werden, die den gesamten Lebenszyklus des Systems abdeckt. Dabei wird der Entwicklungsprozess eines wissensbasierten Systems als Modellierungsprozess aufgefasst, der sich in sechs aufeinander aufbauende Modelle gliedert, die wiederum auf drei Entwicklungsebenen verteilt sind (Abbildung 3-1).

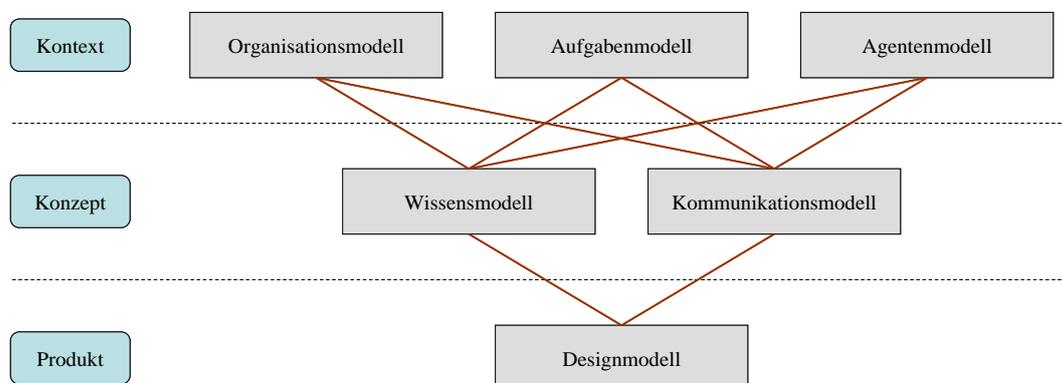


Abbildung 3-1: Die CommonKADS Modelle [ScAk99]

Durch die gesammelten Erkenntnisse der Kontext-Ebene werden Fragen bezüglich des Aufgabenlösungspotentials eines wissensbasierten Systems beantwortet. Ebenso werden Kosten, Nutzen und Folgen einer Systemeinführung beleuchtet. Aufbauend auf den Ergebnissen der Kontext-Ebene bildet das Wissensmodell die Struktur und Art des relevanten Wissens für die Aufgaben ab, die mit Hilfe von wissensbasierten Systemen sinnvoll gelöst werden können. Im Kommunikationsmodell werden dann die Informationsflüsse zwischen den beteiligten Agenten (Mensch, Informationssystem, Software) implementierungsunabhängig aufgezeigt. Im Designmodell der Produkt-Ebene wird dann schlussendlich die eigentliche Systemarchitektur erstellt. Die Erstellung aller Modelle wird durch vordefinierte Arbeitsblätter unterstützt, welche gleichzeitig als Dokumentation dienen.

CommonKADS ist die derzeit umfangreichste Methodologie in den Bereichen Knowledge Engineering und Entwicklung von wissensbasierten Systemen. Die Anwender von CommonKADS werden mittlerweile durch entsprechende rechnergestützte Werkzeuge unterstützt [NN08d]. Der Fokus von CommonKADS liegt jedoch auf der Akquisition, Dokumentation und Formalisierung von Wissen und hilft so dem Knowledge Engineer bei der Kommunikation mit den Experten. Die starke Generalisierung erschwert eine Anwendung von CommonKADS auf das dynamische Umfeld der KBE-Entwicklung im Konstruktionsumfeld, weswegen die Methodologie häufig nicht angewendet wird [Str06]. Aufgrund der Komplexität von CommonKADS und der konzeptbedingten Notwendigkeit, umfangreiche Modelle und Dokumentationen zu erstellen, ist die Methodologie nicht sehr geeignet für kleinere KBE-Anwendungen. Aus diesem Grund wurden reduzierte und angepasste Varianten entwickelt (Pragmatic KADS, siehe [Kin92]).

3.2 MOKA

MOKA (**M**ethodology and Software Tools **o**riented to **K**nowledge-based Engineering **A**pplications) wurde analog zu KADS im Rahmen eines ESPRIT Projekts (EP25418) entwickelt [OIKn98] [BrOI99]. Das dreißig Monate andauernde Projekt wurde 1994 begonnen, die beteiligten Projektpartner stammen aus der Automobil- und Flugzeugindustrie. Im Gegensatz zu KADS war das Ziel des Projekts die Bereitstellung einer Methodologie speziell für die Entwicklung von größeren KBE-Applikationen. Des Weiteren sollten Software-Werkzeuge entwickelt werden, mit denen eine Unterstützung bei der Anwendung der Methodologie erreicht wird.

Ein wesentliches Ergebnis des MOKA-Projekts war die Identifizierung eines KBE-Lebenszyklus, der sechs Phasen enthält (Abbildung 3-2).

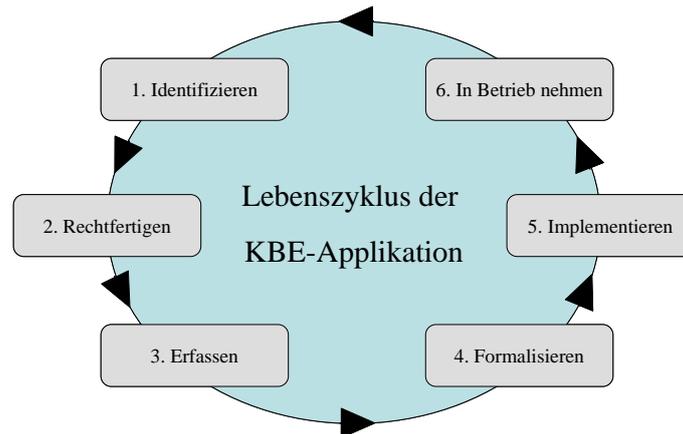


Abbildung 3-2: KBE-Lebenszyklus in MOKA

Besonderes Augenmerk wurde auf die Phasen „Erfassen“ und „Formalisieren“ gelegt. In Phase 3 (Erfassen) wird das relevante Wissen in informellen Modellen erfasst und strukturiert, um dann in Phase 4 (Formalisieren) in formale Modelle überführt zu werden.

In der Phase der informalen Modellierung werden sowohl die natürliche Sprache als auch graphische Darstellungsmöglichkeiten dazu genutzt, um das relevante Wissen in sogenannten Templates zu erfassen, zu dokumentieren und damit wieder verwertbar zu machen. Diese in MOKA als „ICARE-Forms“ bezeichneten Templates lassen sich wie folgt gliedern:

- **Illustration Forms (Illustrationsformblätter)**
Dienen der Erfassung aller Arten von generellem Wissen. Es können z. B. Übersichtsdiagramme oder auch Kommentare verwendet werden.
- **Constraints Forms (Randbedingungsformblätter)**
Hierin werden die Abhängigkeiten der einzelnen Instanzen untereinander modelliert.
- **Activity Forms (Aktivitätsformblätter)**
Aktivitätsformblätter dienen der Beschreibung verschiedener Teilschritte des Problemlösungsprozesses.
- **Rule Forms (Regelformblätter)**
Regelformblätter erlauben die Modellierung von Kontrollwissen.
- **Entity Forms (Instanzenformblätter)**
Instanzen sind die Beschreibungsform von Objekt-Klassen der betrachteten Produkte und Komponenten. Hiermit sind aber auch nicht-physikalische Objekte (z. B. Funktionen) gemeint.

Somit entstehen während der Phase des Erfassens verschiedene Repräsentationen des relevanten Wissens in Form von ausgefüllten ICARE-Forms. Diese müssen nun in der Phase des Formalisierens in ein formales Modell überführt werden. Dazu stellt die MOKA-Methodologie eine Erweiterung der UML, die MML (MOKA Modelling Language) und die in der UML bekannten Aktivitätsdiagramme zur Verfügung.

Die MOKA-Methodologie unterstützt den Knowledge Engineer bei der Erfassung und Formalisierung des expliziten und impliziten Wissens. Durch die Verwendung graphischer Notationen und textlicher Beschreibungsformen wird eine Abstraktionsstufe erreicht, die es dem Knowledge Engineer (der selber meist nicht der Wissensträger ist) ermöglicht, das wissensbasierte System zu erstellen (zu kodieren). Gleichzeitig erfolgt auch eine Wissensdokumentation. Nachteilig sind jedoch die mögliche Redundanz der Wissensspeicherung (die ICARE-Forms dienen zumeist nicht als allgemeiner Wissensspeicher in einer Unternehmung) und die damit einhergehende mögliche Inkonsistenz durch die Entkopplung des Wissens von seinem eigentlichen Träger.

3.3 VIVACE

An dem 2004 gestarteten, vier Jahre dauernden und durch die EU geförderten VIVACE-Projekt (Value Improvement through a Virtual Aeronautical Collaborative Enterprise) waren insgesamt 63 Projektpartner aus sämtlichen Bereichen der Luftfahrtindustrie beteiligt. Ziel des Projekts war die Entwicklung einer durchgängigen virtuellen und kollaborativen Produktentwicklung, um so Kosten und Zeit für die Entwicklung von Luftfahrzeugen und Triebwerken zu reduzieren. Das Gesamtprojekt war in drei Subprojekte aufgeteilt [NN07].

- Virtuelles Luftfahrzeug
Dieses Sub-Projekt war in sechs Arbeitspakete aufgeteilt (System-Simulation, Komponenten, Globales Luftfahrzeug, Flugphysik-Simulation, komplexe Subsysteme, Entwicklung in Bezug auf Wartbarkeit), um so die Entwicklung der wesentlichen Komponenten eines Luftfahrzeugs in einer kollaborativen Umgebung über den gesamten Produktlebenszyklus hinweg zu betrachten.
- Virtuelles Triebwerk
Ebenso wie das Sub-Projekt „Virtuelles Luftfahrzeug“ wurde dieses Sub-Projekt in fünf unterschiedliche Arbeitspakete aufgeteilt mit dem Ziel, einen ganzheitlichen Entwicklungsprozess im Bereich der europäischen Triebwerksentwicklung zu schaffen.
- Fortgeschrittene Ressourcen
Das Sub-Projekt „Fortgeschrittene Ressourcen“ stellt Werkzeuge und Methoden für die anderen beiden Sub-Projekte bereit und dient der Schaffung einer entsprechenden Umgebung für das kollaborative Entwickeln an verteilten Standorten über die Unternehmensgrenzen hinweg. Dazu gehört u. a. die Bereitstellung von verteilten Informationssystemen oder der Aufbau einer IT-Infrastruktur für große Unternehmen.

Innerhalb des für diese Arbeit interessanten Arbeitspaket „Knowledge Enabled Engineering“ des Sub-Projekts „Fortgeschrittene Ressourcen“ wurden Methoden und Werkzeuge entwickelt, mit denen das im Unternehmen explizit und implizit vorhandene Wissen nutzbar gemacht werden soll. Grundlage des Systems bildet eine Datenbank aufbauend auf verschiedenen K-Elementen (Knowledge-Elementen). Dies können z. B. optimale Verfahren basierend auf bereits abgewickelten Pro-

jekten oder auch Richtlinien, Normen oder Verknüpfungen zu Berechnungswerkzeuge sein. Auf der Basis einer kontextbasierten Suche können aus der Fülle der Informationen nun die geeigneten Elemente gefiltert werden. Das System ist lernfähig, da es die Resonanz des Benutzers auf die ausgeführte Suche aufnimmt.

3.4 RAD

Aufgrund der Erkenntnis, dass typische KBE-Systeme mit der Zeit wachsen, wurde von BEYNON DAVIS ET AL. der RAD-Ansatz (**R**apid **A**pplications **D**evelopment) vorgeschlagen [BeCa99]. Grundidee des RAD-Ansatzes und der darin verwendeten Methoden ist die Entwicklung einer KBE-Applikation in einer strukturierten Umgebung und die Betrachtung des Entwicklungsprozesses über die gesamte Lebenszeit der Applikation.

Eine bekannte Methode des RAD-Ansatzes ist DSDM (**D**ynamic **S**ystems **D**evelopment **M**ethod). DSDM wird durch ein gemeinnütziges Konsortium entwickelt, das aus Anbietern, Anwendern und freien Mitgliedern besteht und seinen Sitz in Großbritannien hat [NN08b]. DSDM basiert auf neun Grundsätzen:

1. Aktive Einbindung der Anwender ist notwendig
2. Das Entwicklungsteam muss entscheidungsfähig sein
3. Der Schwerpunkt liegt auf der Lieferung von Applikation in kurzen Zeitabständen
4. Das essentielle Kriterium für die Akzeptanz der Applikationen ist die Eignung für geschäftliche Zwecke
5. Iterative und inkrementelle Entwicklung um zu einer akkuraten Geschäftslösung zu kommen
6. Alle Änderungen während des Entwicklungsprozesses sind reversibel
7. Anforderungen erhalten eine hohe Priorität
8. Während des gesamten Lebenszyklus werden Testphasen integriert
9. Zusammenarbeit und Kooperation zwischen allen Beteiligten ist essentiell

Die Entwicklung von KBE-Applikationen in einer dynamischen Entwicklungsumgebung basierend auf dem RAD-Ansatz führt zu einem evolutionären Lebenszyklus. Die Applikation startet auf der Basis eines groben Skelettmodells. Im Laufe der Entwicklungszeit werden neue Module hinzugefügt, bis ein definierter Zwischenstand erreicht ist, an dem die verantwortlichen Beteiligten eine Begutachtung der bis dahin erreichten Ergebnisse durchführen. Ein wesentliches Kriterium eines erfolgreichen Entwicklungsprozesses ist die Einbindung der Anwender. Auf der Grundlage offener Diskussionen mit allen Beteiligten werden die Anforderungen sowohl zu Beginn des Projekts als auch während der Entwicklungszeit festgelegt und immer weiter konkretisiert (Abbildung 3-3).

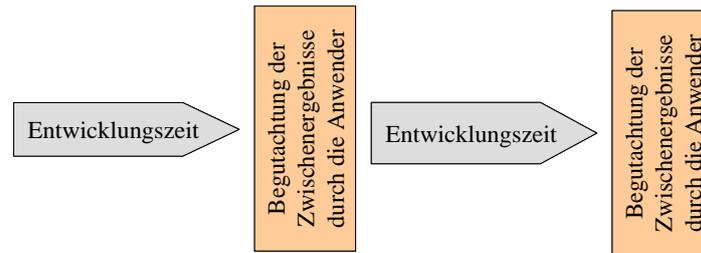


Abbildung 3-3: Entwicklungszeiten und Begutachtung im DSDM

Der RAD-Ansatz und die DSDM-Methode finden ihre Anwendung vor allem bei der Entwicklung von KBE-Applikation mittlerer Komplexität mit hohem Interaktionsgrad der Anwender. Aufgrund der direkten Beteiligung des Endnutzerkreises im Entwicklungsprozess kann eine hohe Akzeptanz der Applikation erreicht werden. Grundvoraussetzung für eine erfolgreiche Umsetzung ist jedoch eine präzise Wissensakquisition schon zu Beginn des Projekts, um so die Anzahl der Iterationsschleifen gering zu halten. DSDM gibt hierzu jedoch keine Vorgaben.

3.5 KCM

KCM (**K**nowledge **C**apture **M**ethodology) soll den Prozess der Wissensakquisition und Wissensspeicherung unterstützen und stellt dazu ein Vorgehensmodell bereit [TeSt00]. Ausgangspunkt der Methodologie ist die Analyse eines erfolgreichen Produkts. Mit Hilfe der in der Methodologie vorgeschlagenen elf Schritte soll dieses Produkt nun in wieder verwendbare Komponenten zerlegt werden. KCM basiert auf der ingenieurwissenschaftlichen Methodik der Problemlösung. Ausgehend von einem Gesamtproblem wird eine Zerlegung durchgeführt, bis eine Untergliederung in verständliche Teilprobleme vorliegt, die untereinander Beziehungen über Schnittstellen aufweisen. Daraufhin erfolgt die Synthese neuer Lösungen durch die Rekombination der Teillösungen. Folgende Schritte für eine strukturierte Analyse werden in KCM vorgeschlagen:

1. Wähle ein Produkt oder einen Prozess für die Modellierung aus
2. Zerlege das Produkt / den Prozess in atomare Komponenten
3. Vergebe Eigenschaften an die atomaren Komponenten
4. Erzeuge atomare Instanzen in Datenbanken oder tabellarischen Datenquellen
5. Erzeuge eine Komponentenklassifikation der atomaren Instanzen
6. Erzeuge Anwendungsfälle durch die Kombination atomarer Komponenten zu Baugruppen
7. Erzeuge einen Anwendungsfall für jede Komponentenklassifikation
8. Definiere einen Satz von Beziehungen und erzeuge Verbindungen zwischen den Teilaspekten
9. Nutze die Verbindungen zwischen Komponenten um parametrische Werte zu propagieren und wende Auswahlbeschränkungen an

10. Assoziiere parametrische Werte von Komponenten für eine Berichterstattung und Visualisierung

11. Wiederhole den Prozess

Ein Beispiel für die Komponentenklassifikation gibt Abbildung 3-4:

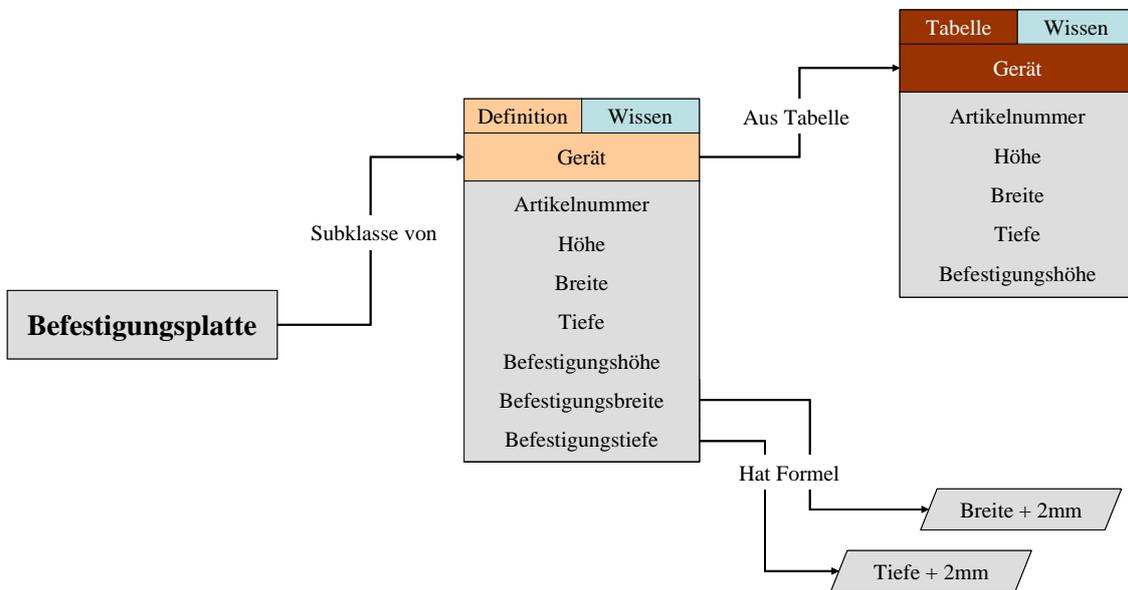


Abbildung 3-4: Komponentenklassifikation in KCM

3.6 Ansatz von Strohmeier

STROHMEIER schlägt in seiner Arbeit [Str06] einen modularen Ansatz für die Entwicklung von KBE-Applikationen vor. Er geht dabei von einem komponentenorientierten Softwareentwurf aus. Grundidee ist die Erstellung von KBE-Applikationen auf der Basis wieder verwendbarer Module, die sich beliebig kombinieren lassen. Dies wird durch die Vereinbarung standardisierter Schnittstellen ermöglicht. Die einzelnen Partialmodelle seines modellbasierten Ansatzes basieren auf einer quantitativen und einer qualitativen Zerlegung des anwendungsspezifischen Wissens. Zur qualitativen Zerlegung wurde ein Klassifikationssystem aufgebaut mit dem es möglich ist, Wissen durch bestimmte Merkmalkombinationen innerer Eigenschaftsklassen, wie Formalisierbarkeit, Beständigkeit und Sicherheit und äußerer Eigenschaftsklassen, wie Relevanz, Ausrichtung und Spezialisierung, einzuordnen. Die qualitative Zerlegung bildet die Basis für die darauf folgende quantitative Zerlegung, in der eine übergeordnete Struktur erstellt wird und Beziehungen zwischen den Eigenschaftsklassen erstellt werden.

Für die Erstellung von KBE-Applikationen in heterogenen und verteilten Systemen bedient er sich dem Konzept der Middleware auf der Basis der CORBA-Spezifikation der OMG. Damit die

verwendeten Wissensmodule unabhängig vom CAD-System bleiben kapselt er die geometrischen Funktionalitäten mit Hilfe von Wrapper²-Klassen.

Durch den Ansatz von STROHMEIER soll die Erstellungszeit für KBE-Applikationen deutlich verringert und zugleich die Qualität durch die Verwendung abgesicherter Wissensmodule verbessert werden. Nachteilig ist jedoch die vom CAD-System unabhängige Implementierung, da auf diese Weise nur Grundfunktionalitäten genutzt werden können, aufgrund der unterschiedlichen Funktionalitätsumfänge. Die Verwendung systemspezifischer – u. U. für die Lösung der Aufgabe notwendiger – Funktionen wird hierdurch stark eingeschränkt.

3.7 Ansatz von Dungs

DUNGS entwickelt in seiner Arbeit ein Konzept für den Aufbau stabiler Modelle in CAD-Systemen mit hybridem Datenmodell [Dun08]. Ausgehend von der Beobachtung, dass die Verwendung von B-Rep Elementen als Referenz für die Features in Modellerstellung zu Fehlern bei Topologieänderungen führen kann entwickelt er eine Referenzarchitektur für die Erstellung von Tragstrukturen. Grundidee seines Ansatzes ist die Kapselung von B-Rep Elementen in Features (Objekten). Damit stellt er ein strukturiertes Datenmodell bereit. Durch die Verwendung von vordefinierten, konfigurierbaren Modellelementen und Methoden zur Manipulation der Geometrie und Topologie auf der Basis der Datenstruktur wird ein hochvariables Modell ermöglicht, das für die Verwendung in nachfolgenden Prozessschritten (insbesondere die Berechnung und Simulation in der frühen Phase der Produktentwicklung) optimiert ist.

Dieses Modell dient als Basis für die Integration semantischer Informationen in das CAD-Modell im Hinblick auf die Integration produktstrukturbezogener und berechnungstechnischer Eigenschaften. Auf der Basis des strukturierten CAD-Modells erarbeitet er eine Methodik, um berechnungstechnische Informationen (Vernetzungsparameter, Modellbildungselemente) in das CAD-Datenmodell zu integrieren und so für weiterführende Prozessschritte nutzbar zu machen.

Die in der Arbeit von DUNGS erarbeiteten Ansätze und Erkenntnisse werden im Rahmen der hier betrachteten Thematik berücksichtigt und weiterentwickelt. Dazu gehört vor allem die Methodik zur Kapselung instabiler B-Rep Elemente, um so eine stabile Referenzkette aufzubauen.

3.8 Ansatz von Janitza

JANITZA entwickelt in seiner Arbeit das Konzept der Freiheitsgrade für die Erstellung variabler CAD-Modelle [Jan04]. Ausgangspunkt seiner Arbeit ist die Idee, ein Produktmodell zu erstellen, das direkt vom Endkunden konfigurierbar ist. Die Gestaltung des Modells erfolgt durch die Integration aller für die Konfigurationsaufgabe notwendigen Wissensanteile mit Hilfe fortschrittlicher

² Als Wrapper bezeichnet man in der EDV allgemein ein Programm, das als Schnittstelle zwischen dem aufrufenden und dem umschlossenen (engl. wrapped) Programmcode agiert.

KBE-Methoden (Parametrik, Regeln, Analysen, etc.). Das so aufgebaute Produktmodell bildet einen allgemeinen Lösungsraum, die Schnittstelle zum Endkunden wird durch die vom Erzeuger des CAD-Modells definierten Freiheitsgrade gewährleistet. Diese Freiheitsgrade dienen auch als Kopplungspunkt für eine in der Arbeit entwickelte Applikation die es ermöglicht, das Produktmodell von verteilten Standorten aus über entsprechende graphische Benutzerdialoge Internet-basiert zu konfigurieren.

Janitza diskutiert in seiner Arbeit keinen Lösungsansatz für die Erstellung variabler Modelle auf Bauteilebene. Sämtliche Überlegungen werden auf der Basis einer konfigurierbaren Baugruppe angestellt. Jedoch werden der Grundgedanke der Flexibilitätserhöhung und die dazu beschriebenen Methoden in dieser Arbeit verwendet.

3.9 Ansatz von Liese

LIESE entwickelt in seiner Arbeit grundlegende Methoden für die Wissensrepräsentation auf Basis eines 3D-CAD-Modells [Lie04]. Aufbauend auf einer Analyse vorhandener Möglichkeiten der Wissensintegration und von KBE-Systemen und ihrer Integration in 3D-CAD-Systeme entwickelt er ein Konzept zur Hierarchisierung der Wissensintegrationstiefe. Er schlägt vor, die Objektorientierung als Repräsentationsformat für die 3D-CAD Wissensrepräsentation zu wählen und zeigt Möglichkeiten auf, um sowohl Funktion, Gestalt und Verhalten in das Repräsentationsformat zu integrieren. Die in der Arbeit durchgeführten Analysen und daraus abgeleiteten Erkenntnisse fließen zu einem Teil in diese Arbeit mit ein.

3.10 Ansatz von Haasis

HAASIS beschreibt in seiner Arbeit den featurebasierten Aufbau von Stirnradtriebekomponenten [Haa95]. Dabei legt er besonderen Wert auf die gussgerechte Gestaltung des Getriebegehäuses. Dies realisiert er durch die Kopplung eines Frame-basierten Expertensystems auf Common LISP³-Basis mit einem proprietären, parametrischen und featurebasierten CAD-System. Durch den Frame-basierten Ansatz wurde die Forderung nach einer Erweiterbarkeit der Wissensbasis (hier: Kombination aus Feature und Frame) erfüllt. Jeder Frame entspricht einem Feature im CAD-System und der bereitgestellten Getriebedatenbank. Die Erzeugung neuer Feature oder die Modifikation bereits bestehender Feature wird durch ein Steuermodul verwaltet, um so die beiden unabhängigen Datenbanken konsistent zu halten. Des Weiteren verknüpft er das CAD-System mit einem Kostenkalkulationssystem und einem Arbeitsplanprozessor, die jeweils an eine Datenbank angekoppelt sind. Der Benutzer interagiert ausschließlich mit dem CAD-System.

Die Arbeit von HAASIS zeigt die Leistungsfähigkeit von in den Konstruktionsprozess eingebundenen Expertensystemen unter Nutzung der Feature-Technologie auf. Durch die durchgehende Nut-

³ Common LISP (**L**ist **P**rocessing) ist eine 1958 am MIT spezifizierte Programmiersprache. LISP ist eine in der KI häufig verwendete Programmiersprache.

zung der Feature-Technologie im Konstruktionsprozess der Getriebekomponenten können auch die komplexen Zusammenhänge in der gussgerechten Konstruktion überprüft und bewertet werden. Ebenso zeigt er den Nutzen der Feature-Technologie für nachfolgende Prozesse wie z. B. die Arbeitsplanung oder Kostenkalkulation auf. Ein Nachteil des Gesamtkonzepts ist jedoch die fehlende Implementierung der Wissensbasis in das CAD-Produktmodell. Aufgrund der dualen Datenhaltung sowohl im CAD-System (Geometriedaten) als auch im Expertensystem (Repräsentation der Feature als Frame) können die Vorteile der wissensbasierten Modellierung nicht voll ausgeschöpft werden. Das Expertensystem dient hier ausschließlich der konstruktionsbegleitenden Plausibilitätskontrolle und stellt Verbesserungsvorschläge zur Verfügung, ohne einen direkten Einfluss auf die Geometrie nehmen zu können.

3.11 VDI-Richtlinie 2209

Der Entwurf der VDI-Richtlinie 2209 befasst sich intensiv mit der dreidimensionalen Produktmodellierung [VDI06] und ihren Auswirkungen auf die Prozesskette der Produktentwicklung. Ein in der VDI-Richtlinie behandelter Teilaspekt ist die Unterstützung des Entwicklungs- bzw. Konstruktionsprozesses durch Wissensverarbeitung. Es wird der Begriff der Wissensbasierten Parametrik eingeführt. Im Gegensatz zur „einfachen“ Parametrik, bei der Maßzahlen die treibende Kraft für die Gestaltbestimmung eines Bauteils sind, basiert die Wissensbasierte Parametrik auf der Integration von Konstruktions- und Konfigurationsregeln in das CAD-Modell. Ziel der Anwendung von Wissensbasierter Parametrik ist der Aufbau eines „intelligenten“ Modells, das auch die Konstruktionsabsichten widerspiegelt.

Die VDI-Richtlinie 2209 zeigt auf, dass die Integration von weiterführendem Wissen in das 3D-CAD Modell derzeit noch nicht zum Stand der Technik gehört. Jedoch werden keine ausführlichen Erläuterungen zum Umgang mit den Möglichkeiten der Wissensbasierten Parametrik im Konstruktionsprozess getroffen.

4 Automatisierte Erzeugung von Produktkomponenten

Modulare, konfigurierbare Produkte sind vor allem im Bereich der Investitionsgüter eine wichtige Möglichkeit, um die Risiken bei der Auftragsabwicklung sowohl für den Kunden als auch für das entwickelnde Unternehmen zu minimieren. Dies gilt vor allem für komplexe Anlagen oder Maschinen. Trotz allem muss die Kundenforderung nach Anpassung an bereits bestehende Prozesse oder Anlagen erfüllt werden. Dazu ist ein entsprechendes Produktkonzept nötig, das es ermöglicht, variable Produkte auf der Basis getesteter und standardisierter Komponenten zu erzeugen. Der Aufbau eines modularen Produktkonzepts auf Plattform- oder Baukastenbasis ermöglicht es, eine hohe äußere Varianz (aus Sicht des Kunden) zu erreichen und dennoch eine möglichst geringe innere Varianz (aus Sicht des Unternehmens) zu sichern. Die Beherrschung der Variantenvielfalt und der Produktkomplexität ist ein entscheidendes Kriterium für die Wettbewerbsfähigkeit des Unternehmens und spielt eine wichtige Rolle für die Sicherung der Marktposition, nicht zuletzt durch die Sicherung hoher Qualitätsstandards auf der Basis intern standardisierter Teile.

Die Verfügbarkeit eines modularen Produktkonzepts allein reicht jedoch nicht aus, um die Produktivität eines Unternehmens konkurrenzfähig zu halten, es bildet jedoch den Grundstein. Die wesentlichen Phasen der Produktentwicklung müssen das Produktkonzept unterstützen, um die Vorteile der Modularisierung richtig nutzen zu können. Dabei spielt die Rechnerunterstützung eine wesentliche Rolle. Erst durch die Nutzung leistungsfähiger Produktkonfiguratoren im Kontext der VPE können die Vorteile eines modularen Produktkonzepts effektiv genutzt werden.

Das in den folgenden Abschnitten dargestellte CAD-basierte Produktkonfigurationssystem wurde für einen namhaften Hersteller von Turbomaschinen im Rahmen eines gemeinschaftlichen Forschungsprojekts entwickelt und umgesetzt. Ziel des Projekts war die automatische Erzeugung von Verdichterkomponenten und Baugruppen in einem parametrischen, featurebasierten CAD-System unter Berücksichtigung der Anforderungen nachfolgender Prozessschritte in der virtuellen Produktentwicklung.

4.1 Produktauswahl: Turboverdichter

Turboverdichter gehören in die Gruppe der Strömungsmaschinen. Im Gegensatz zu den Verdrängermaschinen arbeiten sie kontinuierlich und zumeist in einem stationären Betriebspunkt. Beim Turboverdichter wird durch einen rotierenden Läufer dem strömenden Fluid kinetische Energie zugeführt. In dem nachgeschalteten Leitrad wird kinetische Energie durch Strömungsführung in statischen Druck umgewandelt. Diese Bauart zeichnet sich im Gegensatz zu den Kolbenverdichtern durch eine vergleichsweise geringe Druckerhöhung pro Stufe und hohen Volumendurchsatz aus. Radial- und Axialverdichter sind die beiden Hauptbauarten für Turboverdichter. Beim Axialverdichter strömt das zu komprimierende Fluid in paralleler Richtung zur Drehachse durch den Verdichter. Nach dem rotierenden Laufrad wird das Fluid durch ein stationäres Leitrad geführt, das die Zuströmung zum nächsten Laufrad wieder axial ausrichtet und zur Druckerhöhung beiträgt. Beim

Radialverdichter strömt das Gas axial in das Laufrad der Verdichterstufe und wird dann nach außen (radial) abgelenkt. Das Leitrad hat auch hier die Aufgabe, das nun radial strömende Fluid wieder in axialer Richtung in die nächste Stufe einzuleiten.

4.1.1 Bauweisen

Prozessgasverdichter werden in der Produktion unterschiedlicher Branchen eingesetzt, z. B. in der Petrochemie oder der Düngemittelproduktion. Es existieren unterschiedliche Bauformen für diverse Anwendungsbereiche, z. B. mehrstufige Radialverdichter mit integriertem Getriebe oder auch mehrstufige Axial-Radial-Verdichter. Die Arbeitsgrundlage dieser Arbeit sind die mehrstufigen Einwellen-Radialverdichter (Abbildung 4-1). Bei dieser Verdichterart kann zwischen zwei verschiedenen Bauweisen unterschieden werden, die jeweils von der Lage der Gehäusetrennfuge abhängen. Der Verdichter kann sowohl eine horizontale Gehäusetrennfuge als auch eine vertikale Gehäusetrennfuge (Topfbauweise) aufweisen. Ein Vorteil der Topfbauweise ist die Umsetzung höherer Auslassdrücke (Faktor 10 im Vergleich zu Verdichtern mit horizontaler Trennfuge). Allerdings wird durch diese Bauweise die Zugänglichkeit der innenliegenden Bauteile erschwert, da für Wartungsarbeiten immer der gesamte Verdichterkern gezogen werden muss. Bei einem Verdichter mit horizontaler Gehäusetrennfuge reicht hier das Abheben des Gehäuseoberteils.

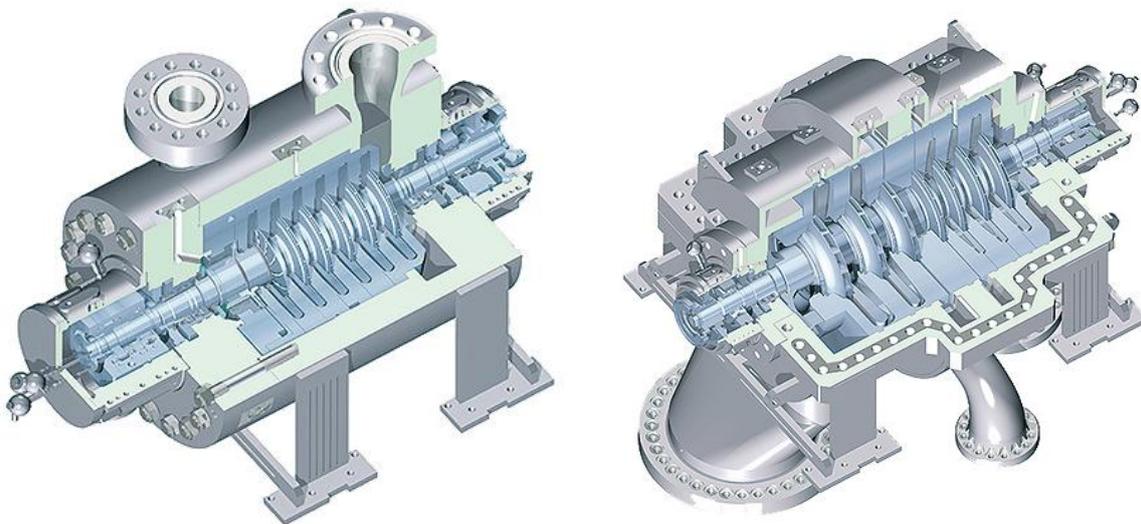


Abbildung 4-1: Verdichter mit horizontaler (links) und vertikaler Gehäusetrennfuge (rechts) (Quelle: Siemens-Pressbild)

Des Weiteren erfolgt eine Unterteilung der jeweiligen Verdichtertypen anhand der Stufenzahl. Bei einem mehrstufigen Verdichter kann das Fluid an bestimmten Stellen des Verdichters ausgeleitet, gekühlt und dem Verdichter an anderen Stellen wieder zugeführt werden. Prozessgasverdichter können je nach Anwendungsfall mehrere Prozessstufen, mehrere Zwischenkühlungsschritte und je Welle mehrere Laufräder aufweisen.

Zum Ausgleich des Axialschubs wird ein Teil der Stufen gegensinnig zu den anderen angeordnet oder ein Ausgleichkolben vorgesehen. Die gegensinnige Anordnung vermeidet die Spaltverluste des Ausgleichkolbens, der in Form einer Labyrinthdichtung ausgeführt wird, führt aber zu komplexeren Gehäuseformen. Auch ist der Ausgleich des Axialschubes nicht für jeden Betriebszustand gewährleistet, so dass verhältnismäßig kräftig dimensionierte Axiallager erforderlich werden [SuSt07]. Der rotierende Teil des Ausgleichskolbens ist je nach Verdichterkonfiguration Bestandteil der Welle oder eines Laufrades.

4.1.2 Betrachtete Komponenten

4.1.2.1 Verdichterkonfiguration

Die Hauptfunktionen eines Verdichters – das Ansaugen und Verdichten des Prozessgases – werden durch den Rotor realisiert. Er besteht aus der Welle und bis zu zehn Laufrädern und weiteren Anbauteilen wie z. B. Hülsen. Die Konfiguration des Rotors beeinflusst in erheblichem Maße die dynamischen Eigenschaften der Gesamtmaschine. Eine wichtige Einflussgröße für die erforderliche Betriebssicherheit ist dabei die kritische Drehzahl. Die kritische Drehzahl und die Resonanz sind Phänomene, die in vielen Maschinen und Anlagen eine wesentliche Rolle spielen. Gerade bei umlaufenden Wellen und Rotoren treten bei bestimmten Drehzahlen unzulässig hohe Vibrationen (hohe Schwingungsamplituden) auf. Diese Vibrationen bezeichnet man als Resonanz von Schwingungen. Bei einer bestimmten Winkelgeschwindigkeit überlagern sich die Schwingungen und werden als Resonanzschwingungen sichtbar. Durch diese Vibrationen werden die Lager der Antriebe stark belastet und es kann unter Umständen sogar zur Zerstörung der Maschine führen. Der Betriebspunkt des Verdichters muss daher in ausreichender Entfernung zu einer kritischen Drehzahl liegen. Auch müssen beim Anfahren der Maschine die kritischen Drehzahlbereiche möglichst schnell durchlaufen werden, um eine Beschädigung der Maschine zu vermeiden. Eine der wesentlichen Einflussgrößen auf das dynamische Verhalten des Rotors ist u. a. der Lagerabstand, der implizit die Gesamtorlänge bestimmt.

Die Konfiguration der Maschine und damit auch des Rotors erfolgt auf der Basis eines Lastenhefts. Durch die meist notwendige Einbindung der Prozessgasverdichteranlage in einen Gesamtprozess beim Kunden sind die prozess- und anwendungsspezifischen Randbedingungen wie z. B. Gas, Einlassdruck, Auslassdruck, Massenstrom und Antriebsmaschine vorgegeben. Anhand dieser Größen erfolgt mit Hilfe unternehmensinterner Berechnungsprogramme eine Auslegung der Maschinenkomponenten. Auf der Basis der Auslegung wird dann ein Verdichterkonzept erstellt, das im Zuge des Forward Engineering dann weiter detailliert und an die kunden- und anwendungsspezifischen Gegebenheiten angepasst und erweitert wird.

4.1.2.2 Welle

Die Welle des Rotors erfüllt mehrere funktionale Aspekte. Sie überträgt das an der Kupplung eingeleitete Drehmoment über eine reibschlüssige Verbindung an die Laufräder. Werden mehrere Ver-

dichter in einem Strang betrieben leitet sie auch das Drehmoment an die folgenden Verdichterwellen weiter. Im Bereich der Dichtungselemente verhindert sie den Austritt des Prozessgases in die Umgebung.

Die Verdichterwelle wird auf der Basis vordefinierter Wellenabschnitte konfiguriert. Diese Abschnitte erfüllen funktionale Aspekte. So existieren z. B. ein Kupplungselement oder mehrere Varianten für Dichtungsabschnitte. Das modulare Konzept der Welle ist damit erweiterbar und kann durch neue funktionale Elemente mit abgeänderter Funktionsweise ergänzt werden. Abbildung 4-2 zeigt eine schematische Darstellung der Wellenmodule und ihrer möglichen Anordnung im Verbund.

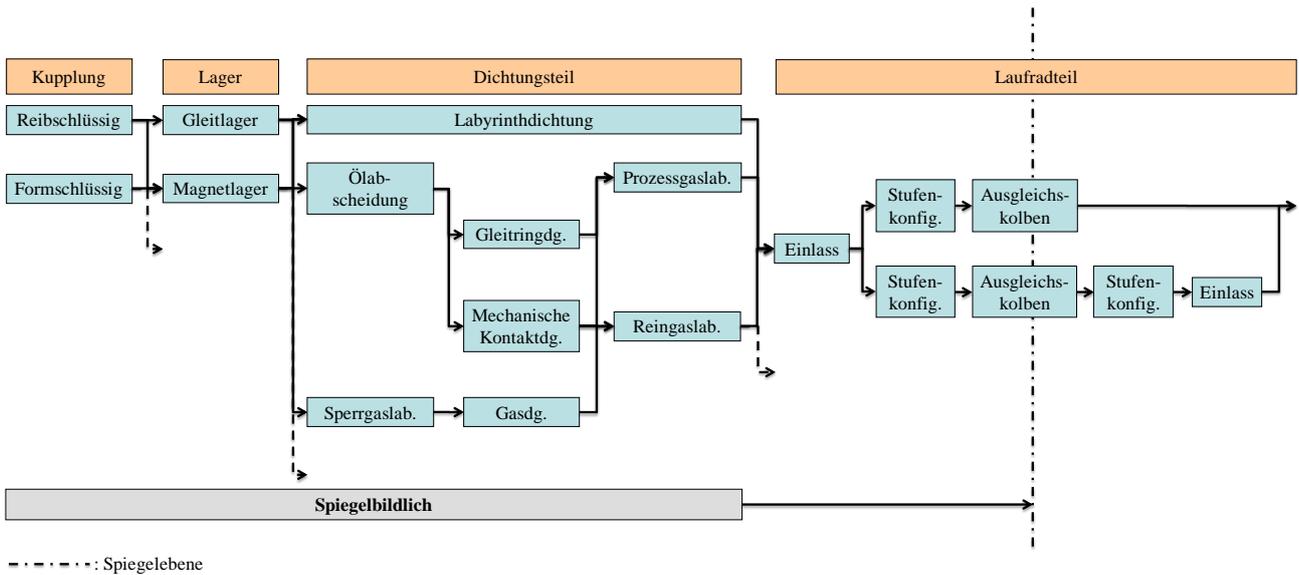


Abbildung 4-2: Modularer Wellenaufbau

Jedes Wellenelement wird durch einen unternehmensinternen Konstruktionsstandard beschrieben, der eine Beschreibung der geometrischen Ausprägung und eine multiple Anzahl an Tabellen enthält. Im Sinne einer Konstruktionstabelle erfolgt der Zugriff auf die entsprechenden Werte durch die Angabe einer oder mehrerer Bezugsgrößen (z. B. Baugröße oder Wellendurchmesser). Im einfachsten Fall kann ein Wertebereich durch die Angabe einer bestimmten Bezugsgröße erfolgen (Tabelle 4-1).

Tabelle 4-1: Einfache Bezugsgröße

Bezugsgröße	Größe 1	Größe 2	Größe 3
Baugröße 1	20	10	30
Baugröße 2	30	15	40

Häufig ergibt sich ein Wertebereich erst durch die Angabe einer Kombination aus mehreren Bezugsgrößen (siehe Tabelle 4-2).

Tabelle 4-2: Multiple Bezugsgrößen

Bezugsgröße 1	Bezugsgröße 2	Größe 1	Größe 2	Größe 3
Baugröße 1_1	Baugröße 2_1	20	10	30
Baugröße 1_1	Baugröße 2_2	30	15	40
Baugröße 1_2	Baugröße 2_1	40	20	10
Baugröße 1_2	Baugröße 2_2	60	20	15

Eine weitere Komplexitätserhöhung ergibt sich, wenn Wertebereiche durch die Einschränkung eines Bereichs der Bezugsgröße (z. B. Gewichtsbereiche) ermittelt werden müssen.

Die Tabellen eines Konstruktionsstandards sind gemeinhin nicht ausreichend, um ein Wellenelement ausreichend zu beschreiben. Da die Elemente im Kontext der gesamten Welle betrachtet werden müssen erfolgt häufig ein Verweis auf Ergebniswerte aus Tabellenabfragen aus weiteren Konstruktionsstandards. In Abbildung 4-3 wird dieser Umstand schematisch dargestellt. Für die Ermittlung eines Wertebereichs aus Tabelle 2 eines Konstruktionsstandards B muss zuerst die notwendige zweite Bezugsgröße aus dem Wertebereich des für das eigentliche Wellenelement bestimmten Konstruktionsstandards A bestimmt werden. Eine derartige Verknüpfung und Kombination von Abfrageergebnissen und Eingabegrößen kann eine beliebige Komplexität erreichen. Die Verknüpfung ist notwendig, um eine nicht-redundante Datenhaltung zu gewährleisten, da sich die Tabellen explizit auf ein Wellenelement beziehen.

Neben den unternehmensintern standardisierten Elementen sind auch tabellarische geordnete Daten für genormte Teile notwendig. Dazu gehören z. B. genormte Konstruktionselemente wie Freistiche oder Zentrierbohrungen.

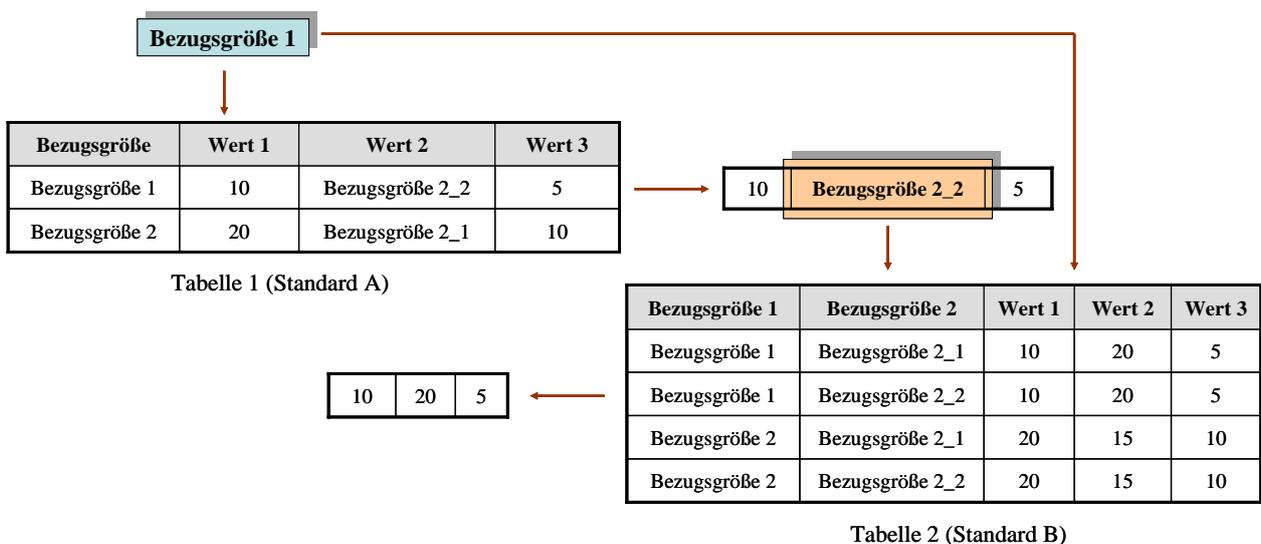


Abbildung 4-3: Konstruktionsstandardübergreifende Abfragen

In den jeweiligen Konstruktionsstandards wird die geometrische Gestalt des Wellenelements beschrieben, die für die Ausprägung notwendigen Maße werden in Form von Parametern dargestellt. In diesem Zusammenhang ist hervorzuheben, dass die Topologie eines Elements nicht konstant ist und sich in Abhängigkeit der Bezugsgrößen ändern kann (z. B. zusätzliche Wellenabsätze oder Formelemente). Die Geometrie eines Wellenelements hat dabei grundsätzlich rotationssymmetrischen Charakter. Dieser wird jedoch beeinträchtigt, wenn z. B. prismatische Formen (Passfedernuten) oder rotationssymmetrische Formen außerhalb der Hauptdrehachse hinzukommen.

4.1.2.3 Laufrad

Das Laufrad ist ein wesentliches Bauteil des Rotors und wird mit der Welle meist reibschlüssig über einen Schrumpfsitz verbunden. Das Fluid strömt axial in den Saugmund ein und wird entlang der Schaufelkontur geführt. Durch die Fliehkraftwirkung wird die Geschwindigkeit des Fluids kontinuierlich erhöht. Am Laufradaustritt strömt das Fluid dann in radialer Richtung aus und wird in die Rückführstufe geleitet. Die Rückführstufe entspricht prinzipiell einem statischen Laufrad und leitet das radial strömende Fluid axial in das nächste Laufrad ein.

Es werden Laufräder mit und ohne Deckscheibe verwendet (Abbildung 4-4). Die Deckscheibe kann in Sonderfällen (Zugänglichkeit für die mechanische Bearbeitung) integraler Bestandteil des Laufrades sein (Laufrad aus dem Vollen gefräst). Üblicherweise wird die Deckscheibe jedoch als Einzelteil gefertigt und durch die Verfahren Schweißen, Schlitzschweißen oder Lötten mit der Radscheibe gefügt.

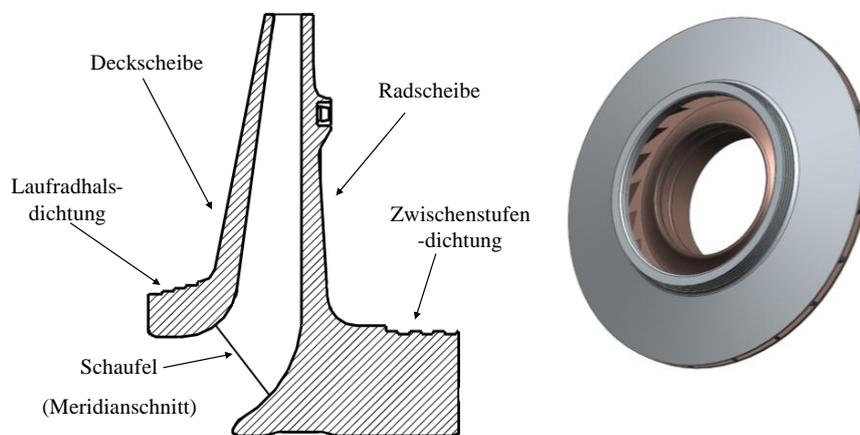


Abbildung 4-4: Laufrad (Halbschnitt und dreidimensional)

Die deckscheibenseitige Laufradhalsdichtung dient dazu, den Leckagemassenstrom vom Laufradaustritt (Bereich hohen Drucks) zurück zum Laufradeintritt (Bereich niedrigen Drucks) zu minimieren. Die radscheibenseitige Zwischenstufendichtung soll einen zu großen Leckagestrom von den einzelnen Verdichterstufen zur jeweils vorherigen Stufe verhindern [GaNo06].

Sowohl die Rad- als auch die Deckscheibe müssen die im Betrieb auftretenden Fliehkräfte aufnehmen und dabei eine möglichst geringe Verformung aufweisen, so dass der konstruktiv erforder-

liche Spalt zwischen dem rotierenden und dem statischen Teil des Verdichters so gering wie möglich gehalten werden kann. Dies minimiert die Spaltverluste und trägt damit zur Erhöhung des Wirkungsgrads des Verdichters bei.

Die Beschaukelung des Laufrades orientiert sich an der erforderlichen Druckerhöhung pro Stufe und den vorliegenden Strömungsverhältnissen. Die Auslegung der Schaufelgeometrie erfolgt zu meist durch numerische Verfahren der Strömungsmechanik (Computational Fluid Dynamics – CFD). Auf der Basis dieser Berechnungen und durch experimentelle Verifizierungen können Kennfelder für standardisierte Laufraddurchmesser erstellt werden. Auf der Basis dieser Kennfelder erfolgt dann im Konfigurationsprozess die Auswahl eines geeigneten Schaufeltyps. Es kann dabei zwischen verschiedenen Grundgeometrietypen unterschieden werden. Die einfachste Schaufelgeometrie basiert auf einem Profildesign aus geometrischen Grundelementen (Kreisbögen und Linien) oder einem Spline durch vordefinierte Punkte. Die Schaufel weist keine Verwindung in Längsrichtung auf, daher wird dieser Schaufeltyp hier als zweidimensionale Schaufel bezeichnet. Der als dreidimensionale Schaufel bezeichnete Schaufeltyp basiert auf einer Schaufelmittenfläche mit variabler Dickenverteilung. Die Mittenfläche ist eine Regelfläche, die durch das Ziehen einer Verbindungsgeraden zwischen einem deck- und einem radscheibenseitigen Kurvenzug entsteht. Die Dickenverteilung, die normal zur Mittenfläche und entlang der beiden Konturzüge verläuft, beschreibt rad- und deckscheibenseitige Profilkurven, deren Verbindungsfläche die endgültige Schaufelaußenfläche ist.

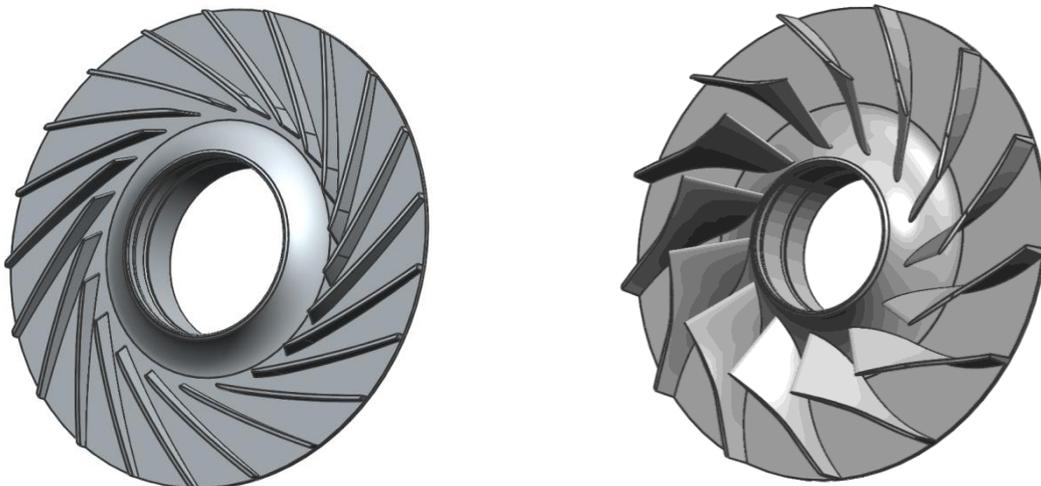


Abbildung 4-5: Radscheiben mit unterschiedlichen Schaufeltypen (links: 2D, rechts: 3D)

Die jeweiligen Schaufeltypen sind grundsätzlich parametrischer Natur und können im Rahmen der standardisierten Laufraddurchmesser skaliert werden. Damit ergibt sich jedoch nicht zwangsweise eine lineare Skalierung der gesamten Schaufelgeometrie, da weitere Randbedingungen im Aufbau der Schaufel beachtet werden müssen.

Die Beschreibung der dreidimensionalen Schaufelgeometrie auf der Basis einer Mittenfläche mit variabler Dickenverteilung ist in ihrer vorliegenden Form nicht unmittelbar in einem 3D-CAD-

Modell umsetzbar, vielmehr orientiert sich die Beschreibungsform an den Bedürfnissen der Arbeitsvorbereitung und der strömungstechnischen Auslegung (CNC-Programmerstellung und CFD). Zwar kann die Schaufelgeometrie für einen spezifischen Durchmesser nach der vektoriellen Berechnung der nötigen Profilkurven im CAD-System erfolgen, jedoch ist diese Geometrie dann nicht linear skalierbar und somit nicht für eine parametrische Vorlage zu gebrauchen.

Die Laufräder eines Verdichters beeinflussen den Wirkungsgrad der Anlage in sehr hohem Maße und sind somit eine wichtige Kenngröße des Auslegungsprozesses der Maschine. Daher ist die aus der Konfiguration stammende Beschreibung des Laufrades sehr detailliert. Ebenso wie die Wellenabschnitte werden auch die Laufräder in genau beschriebene Abschnitte unterteilt. Dazu gehören die Deckscheibe, die Radscheibe und die Schaufeln. Jeder dieser Abschnitte wird durch Konstruktions- und Auslegungsstandards genau beschrieben. Wie in Abschnitt 4.1.2.2 bereits erläutert beinhalten die Laufrad-bezogenen Konstruktionsstandards Tabellenwerte für bestimmte geometrische Größen, die auf der Basis einer oder mehrerer Bezugsgrößen eindeutig bestimmbar sind. Auch hier gibt es viele Verknüpfungen zu weiteren Konstruktionsstandards. Im Gegensatz zu den Konstruktionsstandards beinhalten Auslegungsstandards zumeist Berechnungsvorschriften oder Gestaltungshinweise, z. B. für die Radscheibengestaltung in Bezug auf die Laufradfestigkeit. Das in den Auslegungsstandards dokumentierte Wissen liegt also zumeist in Form von mathematischen und / oder verbalen Beschreibungen vor.

Im Unterschied zur Wellenkonfiguration, bei der die Gestalt der Welle bereits in hohem Maße festgelegt ist, bestehen für den Konstrukteur u. a. Freiheiten bei der Gestaltung der Nabe, des Radscheibenauslaufes und im Bereich der Laufradhalsdichtung. Während des Prozesses des Forward-Engineering werden hier teilweise umfangreiche Anpassungen vorgenommen. Dabei ist jedoch sicherzustellen, dass die vorliegenden Auslegungsregeln nicht verletzt werden.

4.2 Problemanalyse

Vor der eigentlichen Konzeptentwicklung für die automatisierte Erstellung der Verdichterkomponenten soll in diesem Kapitel detailliert auf die einzelnen Problemstellungen eingegangen werden, die hierbei berücksichtigt werden müssen. Aufgrund der Komplexität der jeweiligen Gestaltungsmodule einer Verdichterkomponente werden diese hier nicht einzeln betrachtet. Vielmehr werden auf der Basis einer Analyse aller Gestaltungsmodule Gemeinsamkeiten ermittelt, die aufgrund ihrer Eigenschaften eine Auswirkung auf den Prozess der Konzeptentwicklung haben.

4.2.1 Umsetzung in einem parametrischen CAD-System

Das Konzept für die automatisierte Verdichterkonfiguration muss sich in die bestehende IT-Landschaft des Unternehmens einfügen. Dazu gehört vor allen Dingen die Umsetzung im vorlie-

genden CAD-System NX der Firma SIEMENS PLM SOFTWARE. NX ist ein teil-parametrisches⁴ und featurebasiertes CAD-System auf der Basis des Parasolid Geometriekernels. Es bietet die Möglichkeit der Flächen- und Volumenbasierten Modellierung im Sinne eines hybriden Modellersystems. Es existieren jedoch auch noch Geometrieelemente, die ausschließlich in der B-Rep Struktur vorkommen und somit nicht im CSG-basierten Modellbaum auftauchen. Diese Elemente können nicht assoziativ mit anderen Modellelementen verknüpft werden.

Das System kann durch integrierte Module erweitert werden, die auf dem gleichen Datenmodell aufbauen. Auf diese Weise können z. B. Anwendungen zu Ableitung von CNC-Programmen oder zur konstruktionsbegleitenden Berechnung mittel FE-Methoden modellbasiert genutzt werden.

4.2.2 Gestaltung von Bauteilen und Baugruppen

Betrachtet man die in Kapitel 4.1.2 erläuterten Verdichterkomponenten, so wird deutlich, dass die automatisierte Erstellung der CAD-Modelle sowohl auf Baugruppenebene als auch auf Teileebene stattfinden muss. Die Komplexität der erforderlichen Teilekonfiguration wird hier am Beispiel der Welle erläutert. Die Ausführungen gelten jedoch in ähnlicher Weise auch für die Laufräder des Rotors.

Die Welle setzt sich aus vielen Gestaltungsmodulen zusammen, die einem bestimmten funktionalen Bereich zugeordnet sind. Dazu gehört der Bereich der Antriebs- und Abtriebskupplung einschließlich der Lager, die Bereiche der Dichtungselemente und der sogenannte Radialteil, in dem die Laufräder und eventuell der Ausgleichskolben angeordnet sind. In jedem funktionalen Bereich kann eine unterschiedliche Zahl von Gestaltungsmodulen vorkommen. Es sind z. B. unterschiedliche Anordnungen von Dichtungselementen oder Radialteilelementen mit variierender Elementanzahl möglich. Im Bereich des Radialteils ist des Weiteren zu beachten, dass der Maschinentyp Einfluss auf die Anordnung der Gestaltungsmodule hat. Bei einer Maschine vom Typ A sind die einzelnen Radialteilelemente in Reihe angeordnet, die darauf angeordneten Laufräder weisen also alle in dieselbe Richtung. Im Gegensatz dazu werden bei einer Maschine des Typs B die Radialteilelemente an einer bestimmten Position gedreht, so dass die Laufräder gegensätzlich zueinander angeordnet sind. Entscheidend für die Position der Laufräder ist die Laufradreferenzebene eines Radialteilelements. Neben diesen beiden Maschinentypen müssen für eine Übertragbarkeit des Konzepts jedoch auch weitere Typen berücksichtigt werden, bei denen die Radialteilelemente beliebig angeordnet sein können (Abbildung 4-6).

⁴ In teil-parametrischen CAD-Systemen erfolgt keine automatische Parametrisierung aller Modellelemente bei der Erzeugung von Features. Der Anwender kann das Modell gemäß den eigenen Vorstellungen mit Parametern versehen. Bei einer undurchdachten Arbeitsweise ergibt sich jedoch u. U. der Nachteil eines unkontrollierbaren Modellverhaltens bei Anpassungen.

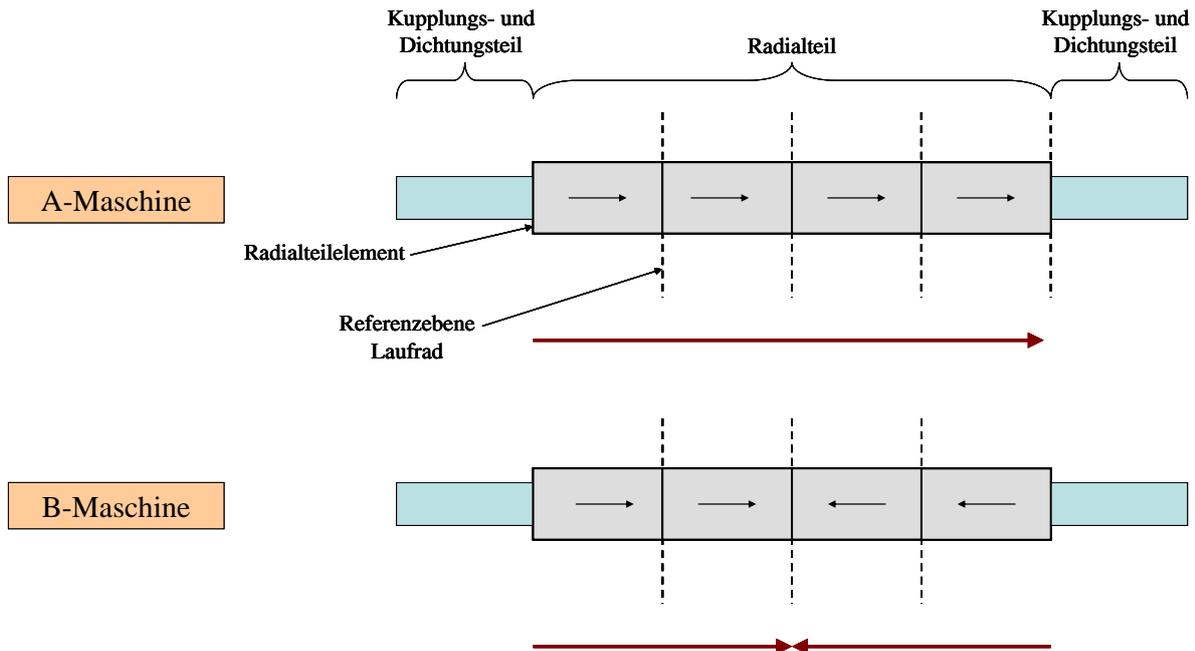


Abbildung 4-6: Positionierung der Radialteilelemente

Die Konfiguration auf Teileebene ist jedoch nicht für alle Komponenten ausreichend. Dies wird anhand des Laufrades aufgezeigt. Das Laufrad setzt sich aus den grundlegenden Gestaltungsmodulen Radscheibe, Schaufeln und Deckscheibe zusammen, wobei die Deckscheibe nicht zwingend vorhanden sein muss. Handelt es sich um ein offenes Laufrad, so wird die Deckscheibe nur in Form eines Materialschnittes ausgeführt, da die eigentliche Schaufeldefinition immer ein Übermaß vorsieht, das durch eine nicht physisch vorhandene Deckscheibe begrenzt wird. Auch wenn die Radscheibe mit Schaufeln und die Deckscheibe am Ende ein integrales Bauteil bilden hängt es von der Fertigungsart ab, ob eine Baugruppe oder ein Einzelteil vorliegt. Werden die Deckscheibe und die Radscheibe aus dem Vollen gefräst, so handelt es sich um ein Einzelteil, ansonsten werden die Radscheibe und Deckscheibe durch die Fügeverfahren Löten, Schweißen oder Schlitzschweißen verbunden, womit der Fall der Schweißbaugruppe vorliegt. Ein selten auftretender Fall ist das Fügen von Schaufeln mit der Radscheibe und das daran anschließende Fügen der Deckscheibe. Hier handelt es sich dann um eine Schweißbaugruppe mit Unterbaugruppe. Im Fall des offenen Laufrades ohne Deckscheibe ist lediglich ein Einzelteil vorzusehen.

4.2.3 Datenredundanz

Aufgrund der Vielzahl an zu berücksichtigenden Konstruktions- und Auslegungsstandards mit jeweils mehreren Tabellen mit einer großen Anzahl von Einträgen muss eine große Datenmenge gespeichert und verwaltet werden, wobei die Daten, wie bereits in Kapitel 4.1.2 erläutert, auch miteinander verknüpft sind. Hinzu kommt, dass die Daten nicht statisch sind, sondern aufgrund von Revisionen oder Ergänzungen ständig aktualisiert oder erweitert werden.

Die Tabellen der Konstruktionsstandards enthalten neben den Baugrößeninformationen (parametrische Werte) auch fertigungstechnisch relevante Informationen (z. B. Passungsangaben und

Toleranzinformationen), die keinen direkten Einfluss auf die geometrische Ausprägung eines Gestaltungsmoduls haben, jedoch in weiterführenden Prozessschritten genutzt werden müssen, da auch diese das modulare Konzept der Teile berücksichtigen (Arbeitsplanung, CAM-Bearbeitung). Die fertigungstechnischen Informationen müssen jedoch Bestandteil des Modells sein, damit sie für die Ableitung fertigungstechnischer Zeichnungen zur Verfügung stehen.

Des Weiteren existiert eine Vielzahl von Prozessen und Beteiligten, die auf die in den Konstruktionsstandards hinterlegten Informationen zugreifen müssen, ohne einen direkten Zugriff zum CAD-Modell zu haben. So basieren z. B. die Ergebnisse der Verdichterauslegung auf diesen Daten. Dieser Prozessschritt ist jedoch zeitlich vor der Erstellung des CAD-Modells angesiedelt und wird mit Hilfe eines unternehmensinternen Auslegungsprogramms durchgeführt, welches auch standortübergreifend genutzt wird.

Das Konzept muss daher ein geeignetes Datenmanagement für die zielgerichtete Bereitstellung aller für die CAD-Modell Erstellung notwendigen Daten zur Verfügung stellen.

4.2.4 Anreicherung mit semantischen Informationen

3D-CAD Modelle werden zunehmend als Speicher aller produktrelevanten Information genutzt (integriertes Produktmodell) [Lie04]. Eine wesentliche Komponente zur Erreichung dieses Ziels ist die Anreicherung des Modells mit semantischen Informationen, die in weiterführenden Prozessschritten genutzt werden können. Die Geometrie des CAD-Modells ist damit nur eine Teilkomponente des eigentlichen virtuellen Produkts. Von wesentlicher Bedeutung für die Wieder- und Weiterverwendbarkeit des CAD-Modells in nachfolgenden Prozessschritten ist daher eine geeignete Semantik, welche auf die Anforderungen der entsprechenden Bereiche abgestimmt ist. Dazu gehört die Wahl einer Informationsrepräsentationsform in Kombination mit einem Informationsträger im Modell. Bei der Entwicklung des Gesamtkonzepts müssen daher die Nutzer des CAD-Modells identifiziert werden, um so Anforderungen an die Semantik des Modells zielgerichtet zu ermitteln.

4.3 Anforderungen an das Gesamtkonzept

In diesem Kapitel sollen die Anforderungen an das Konzept für den automatisierten Aufbau von Verdichterkomponenten erläutert und diskutiert werden. Die hier gestellten Anforderungen ergeben sich zum großen Teil aus dem in Kapitel 4.1.2 erläuterten Produktaufbau und den in Kapitel 4.2 dargestellten Problemstellungen und lassen sich in einem Anforderungskatalog zusammenfassen. Ein neues Softwaresystem, das auch im produktiven Betrieb eines Unternehmens durch eine Vielzahl an Mitarbeitern mit unterschiedlichem Ausbildungsstand eingesetzt werden soll, muss eine Vielzahl sowohl technischer als auch nicht-technischer Anforderungen erfüllen. Dazu kommt die Berücksichtigung der bereits bestehenden IT-Landschaft, die zumeist eine nicht-variable Größe bei der Konzeptentwicklung ist. Des Weiteren müssen grundlegende Anforderungen an die für den Konfigurationsprozess verwendeten Methoden gestellt werden. Die Unterteilung des gesamten Anforderungskatalogs wird in Abbildung 4-7 strukturiert dargestellt.

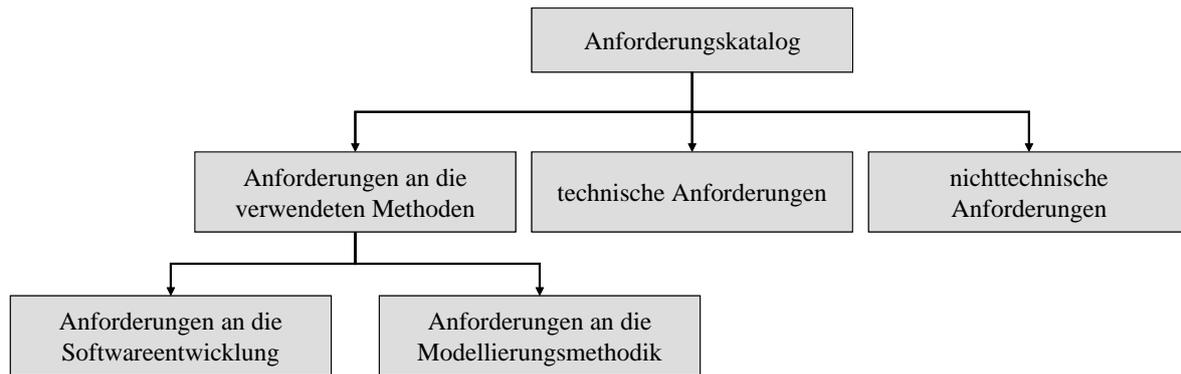


Abbildung 4-7: Aufbau des Anforderungskatalog

Die in den folgenden Kapiteln detaillierter dargestellten Anforderungen sind zunächst allgemein formuliert. Diese Anforderungen werden später für die Entwicklung des Konfigurationssystems genutzt und dienen als Bewertungskriterium für die Eignung.

4.3.1 Anforderungen an die verwendeten Methoden

4.3.1.1 Anforderungen an die Softwareentwicklung

Die Konfiguration der Verdichterkomponenten soll auf der Basis einer im CAD-System integrierten Softwarelösung erfolgen. Folgende Anforderungen sind hierbei zu berücksichtigen:

- **Modularer Aufbau**
Das System muss sich um zusätzliche Funktionalitäten erweitern lassen, ohne die grundlegenden Funktionalitäten zu stören. Dies betrifft vor allem die Zeit nach der ersten Inbetriebnahme. Im Laufe der Systemlebenszeit können im Rahmen von Produkt- und Prozessänderungen funktionale Modifikationen am System nötig sein. Diese müssen auch ohne die Kenntnis des Gesamtsystems implementierbar sein.
- **Komponentenspezifische Erweiterung durch Plug-Ins**
Neben den hier zu betrachtenden Komponenten der verschiedenen Verdichterbauweisen müssen auch weitere Komponenten automatisiert im CAD-System erzeugbar sein, ohne die Grundfunktionalität des Gesamtsystems detailliert zu kennen. Daher soll eine definierte Schnittstelle für die Erstellung komponentenspezifischer Plug-Ins vorhanden sein.
- **Unabhängigkeit vom Betriebssystem**
Aufgrund der Tatsache, dass sowohl das CAD-System als auch das PDM-System und viele weitere Softwarewerkzeuge auf verschiedenen Betriebssystemen eingesetzt werden müssen ist das System betriebssystemunabhängig zu implementieren. Dazu gehört die ausschließliche Nutzung betriebssystemunabhängiger Schnittstellen.

4.3.1.2 Anforderungen an die Modellierungsmethodik

Die Ausgabe des Systems soll ein CAD-Modell sein, das durch die Anwender in allen weiteren Schritten der VPE genutzt werden kann. Aufgrund dessen müssen einige Anforderungen an die Modellierung des Produktmodells gestellt werden. Diese ergeben sich sowohl aus den vorliegenden Rahmenbedingungen (Integration in eine bestehende CAx-Landschaft) als auch aus allgemein modellierungsmethodischer Sicht.

- Durchgängig parametrischer, assoziativer und featurebasierter Aufbau der Gestaltungsmodule
Parametrische und featurebasierte CAD-Systeme stellen den aktuellen Stand der Technik dar. Die durchgängige Nutzung der zur Verfügung stehenden Techniken soll die Verwendung der Modelle in weiterführenden Prozessschritten unterstützen.
- Sicherung eines stabilen Geometriemodells
Ein jederzeit stabiles Geometriemodell ist eine Grundvoraussetzung für die Automatisierbarkeit eines CAD-Modells. Während des Erstellungsprozesses muss durch geeignete Methoden die Konsistenz des Modells sichergestellt werden, ohne dass der Anwender interaktiv eingreifen muss. Dies gilt auch für nachträgliche Änderungen am Modell.
- Transparenter und strukturierter Modellaufbau
Ein wesentliches Kriterium für die weitergehende Nutzung der erstellten Modelle ist ein transparenter und strukturierter Modellaufbau. Für den Fall der Weiterverwendung und Modifikation des Modells muss sichergestellt werden, dass auch Anwender ohne Kenntnis der Gestaltungsmodule die Entstehungsgeschichte nachvollziehen und verstehen können.
- Berücksichtigung der Anforderungen aus weiterführenden Prozessen
Eine durchgängige VPE ist nur möglich, wenn die weiterführenden Prozesse das Produktmodell ohne Anpassungen oder Modifikationen direkt nutzen können. Das betrifft sowohl die methodische Modellierung als auch die Semantik der Gestaltungsmodule. Diese Anforderungen müssen durch das Konzept berücksichtigt werden. Ebenso muss die Dynamik der Anforderungen durch eine geeignete Strategie zur Modellanpassung beachtet werden.

4.3.2 Technische Anforderungen

Das Konfigurationssystem muss sich in eine bestehende PDM-Landschaft integrieren, die sich zudem in einem ständigen Wandel befindet. Die hier dargestellten Anforderungen berücksichtigen die für eine lauffähige Umsetzung notwendigen Randbedingungen. Dabei wird auch der Fall eines Ausfalls des Konfigurationssystems in Betracht gezogen.

- Kommunikation mit dem CAD-System
Das Gesamtsystem muss bidirektional mit dem CAD-System gekoppelt sein. Änderungen am Modell müssen durch das System erkannt werden. Das System muss dazu in der Lage

sein, geometrische Elemente und topologische Elemente eindeutig zu identifizieren und zu analysieren.

- **Kommunikation mit dem PDM-System**

Das System muss dazu in der Lage sein, mit dem PDM-System des Unternehmens zu kommunizieren. Sämtliche erzeugten Teile und Baugruppen müssen konsistent in den Datenbestand des PDM-Systems eingepflegt werden. Dazu gehören sowohl die Erstellung von Stücklisten als auch die Pflege von Revisionsständen von Bauteilen und Baugruppen.

- **Graphische Benutzerschnittstelle**

Die graphische Benutzerschnittstelle dient zur Kommunikation des Systems mit dem Anwender. Es muss unklare Sachverhalte in geeigneter Form (Bilder, Grafiken, Text) vermitteln und den Prozessfortschritt vermitteln. Die Gestaltung der Benutzerschnittstelle hat entscheidenden Auswirkungen auf die Akzeptanz des Gesamtsystems durch die Endanwender.

- **Einsatz der Modellierungselemente auch ohne Konfigurationssystem**

Die für die Gestaltung von Welle und Laufrad zu verwendenden Elemente müssen auch ohne das Konfigurationssystem durch den Endanwender erreichbar sein, um so die Modellierung von Verdichterkomponenten auch interaktiv im System zu ermöglichen. Zum Einen soll dadurch ermöglicht werden, dass Komponenten auch ohne das Vorhandensein einer Formalen Beschreibung erzeugt werden können (z. B. im Fall der experimentellen Konfiguration). Zum Anderen wird dadurch ermöglicht, dass auch beim Ausfall des Konfigurationssystems die Erzeugung von Verdichterkomponenten möglich ist.

Die Elemente müssen daher derart gestaltet sein, dass bei einem interaktiven Einsatz geeignete graphische Schnittstellen zur Verfügung stehen, die dem Anwender die Wertbelegung ermöglichen und unklare Sachverhalte erläutern.

- **Trennung von Datenbasis und Gestaltungsmodul**

Die in den Konstruktions- und Auslegungsstandards definierten Daten dürfen nicht innerhalb der Gestaltungsmodule gespeichert werden. Dies verringert den Aufwand für die Datenverwaltung, da im Fall einer Erweiterung des Datenbestands (z. B. durch das Hinzufügen einer zusätzlichen Baugröße) die Gestaltungsmodule nicht angepasst werden müssen.

4.3.3 Nichttechnische Anforderungen

Bei der Entwicklung und Einführung eines neuen Softwaresystems spielen die Faktoren „Mensch“ und „Betriebswirtschaft“ eine wesentliche Rolle für den Erfolg. Die folgenden Anforderungen sollen dies berücksichtigen.

- **Erhöhung der Mitarbeitermotivation**

Die Nutzung des Softwaresystems und der Gestaltungsmodule muss einen spürbaren Vorteil für die involvierten Mitarbeiter bringen. Dies kann sowohl eine Entlastung von Routinetätigkeiten sein als auch die Schaffung neuer kreativer Freiräume.

- Reduzierung der Produktentwicklungskosten
Aus betriebswirtschaftlicher Sicht muss sich der Einsatz des Systems durch eine mittelfristige Reduzierung der gesamten Produktentwicklungskosten rechnen. In diese Betrachtung muss die Wartung des Systems einbezogen werden.
- Systemwartung und Erweiterung durch Mitarbeiter
Änderungen an den Gestaltungsmodulen und - in begrenztem Maße – an den Grundfunktionalitäten des Konfigurationssystems müssen durch Mitarbeiter des Unternehmens möglich sein, um kurze Reaktionszeiten im Fall von Änderungswünschen zu ermöglichen.

4.4 Konzeptentwicklung

Aufgrund der in Kapitel 4.2 analysierten Problemstellungen und den daraus in Kapitel 4.3 abgeleiteten Anforderungen wird nun das für die automatisierte Konfiguration der Produktkomponenten zugrundeliegende Konzept entwickelt.

4.4.1 Grundlegende Konzeption

Für die Konfiguration der Verdichterkomponenten muss sowohl ein Konzept für die konfigurierende Software als auch für die zu konfigurierenden Gestaltungsmodule entwickelt werden. Beide Konzepte sind voneinander abhängig und müssen daher im Zusammenhang betrachtet werden. Aus diesem Grund erfolgt in den weiteren Kapiteln jeweils eine detaillierte Beschreibung der Teilkonzepte.

Die Verdichterkomponenten sind auf Teileebene modular aufgebaut und setzen sich aus verschiedenen Gestaltungsmodulen zusammen. Der automatische Aufbau solch variabler Bauteile kann prinzipiell auf zwei Arten erfolgen:

- Dynamische Geometrieerzeugung durch vordefinierte Makros
- Einsatz vordefinierter Komponenten mit wissensbasierten Anteilen

Bei der dynamischen Erstellung der Gestaltungsmodule durch entsprechende Makros werden die Gestaltungselemente jedes Gestaltungsmoduls automatisiert eingebaut, mit Werten belegt und miteinander verknüpft. Für die Erstellung der gesamten Komponente müssen die Makros dann durch das Konfigurationssystem aufgerufen werden. Ein wesentlicher Nachteil dieser Vorgehensweise ist die Verlagerung der eigentlich interaktiven Geometriemodellierung hin zu einer programmiertechnischen Aufgabenstellung, die nur von einem Spezialisten durchgeführt werden kann. Die interaktive Modellierung im CAD-System ist jedoch – vor allem im Hinblick auf die Berücksichtigung von Abhängigkeiten im Geometriemodell – wesentlich effektiver und benötigt weniger Zeit als die Programmierung von Modellierungsabläufen. Ebenso kann mit Hilfe dieser Vorgehensweise die Forderung nach einer Datenbankanbindung nur teilweise erfüllt werden. Zwar kann innerhalb der Makros auf die Datenbank zugegriffen werden, jedoch ist dieser Zugriff nur beim Aufruf der Makros aktiv. Sollen die Eingabewerte eines Gestaltungsmoduls geändert werden, so hat dies u. U. einen erneuten

Datenbankzugriff zur Folge. Bei der Erstellung der Geometrie mit Hilfe von Makros müssen daher die Gestaltungselemente des zu bearbeitenden Gestaltungsmoduls gelöscht werden, damit das Makro mit den neuen Datenbankwerten ausgeführt werden kann. Dadurch ergibt sich eine dauerhaft notwendige Kopplung zwischen dem die Makros steuernden Konfigurationssystem und dem Verdichterkomponentenmodell. Durch diese erzwungene Kopplung kann die Forderung, Gestaltungsmodule auch ohne Konfigurationssystem einsetzen zu können, nur eingeschränkt genutzt werden.

Eine weitere Möglichkeit, die variablen Teilemodelle der Verdichterkomponenten zu erstellen, ist die Nutzung des Konzepts der benutzerdefinierten Features (**User-Defined Feature – UDF**). Jedes UDF fasst alle notwendigen Gestaltungselemente zusammen und konfiguriert sich auf der Basis von vordefinierten Eingabewerten. Jedes UDF muss daher durch eine geeignete Modellierungsmethodik in der Lage sein, alle Konfigurationen eines Konstruktionsstandards abbilden zu können. Ein wesentliches Merkmal der hier verwendeten UDFs ist die Trennung von prozeduralem und deklarativen Wissensanteil. Der prozedurale Wissensanteil bezeichnet in diesem Fall das Wissen, wie ein UDF zu agieren hat. Dieser Wissensanteil weist eine enge Kopplung zum CAD-Modell auf, womit eine Speicherung und Verarbeitung im jeweiligen UDF sinnvoll ist. Der deklarative Wissensanteil beinhaltet dagegen alle Daten, die vom prozeduralen Wissensanteil benötigt werden. Die Verknüpfung der beiden Wissensanteile erfolgt über eine Datenbankschnittstelle. Sämtliche Daten aus den Konstruktionsstandards werden in einem unternehmensweit nutzbaren DBMS gespeichert. Um auf die Daten dynamisch zugreifen zu können, müssen die UDFs über eine ODBC-Schnittstelle verfügen. Mit Hilfe der Abfragesprache SQL können die aktuell benötigten Daten aus dem DBMS ermittelt werden (Abbildung 4-8).

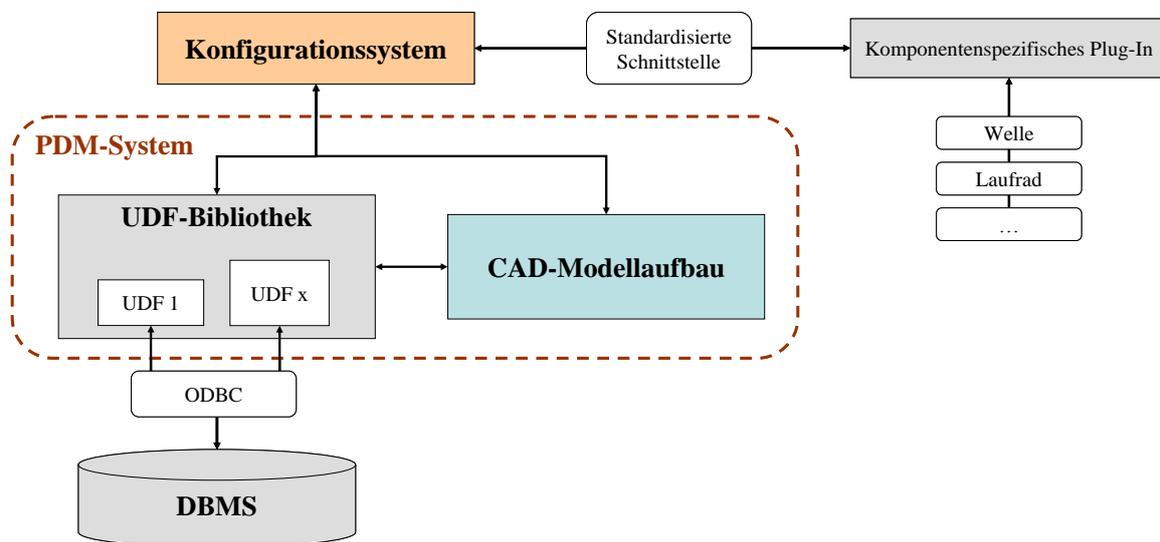


Abbildung 4-8: Grundlegender Aufbau des Konfigurationssystems

Das eigentliche wissensbasierte Konfigurationssystem ist mit dem CAD-System verknüpft und hat direkten Zugriff auf die UDF-Bibliothek. Aus Sicht des Konfigurationssystems ist ein UDF im Sinne der objektorientierten Sichtweise eine Klasse. Daher müssen die UDFs Attribute über ihren derzeitigen Zustand aufweisen und Methoden bereitstellen, um ihr Verhalten zu beeinflussen. Aus-

gehend von einer formalen Beschreibung der Bauteile und der Baugruppen erzeugt das Konfigurationssystem im PDM-System die notwendigen Items⁵ und Dateien und verknüpft diese anhand der vordefinierten Baugruppenstruktur. Anschließend werden in den Bauteilen die formal beschriebenen UDFs instanziiert und miteinander verknüpft.

4.4.2 Softwarekonzept

4.4.2.1 Analyse und Auswahl möglicher Programmiersprachen

Das vorliegende CAD-System NX weist eine Vielzahl von verfügbaren Programmiersprachen auf die dazu genutzt werden können, das System um spezifische Anwendungen zu erweitern [Sie08]. Dazu gehören die bekannten Hochsprachen C, C++ und Java. Eine Sonderstellung nimmt die Sprache GRIP (**G**raphics **I**nteractive **P**rogramming) ein. Diese Sprache weist Ähnlichkeiten zu Basic und Fortran auf und ist eine Eigenentwicklung des CAD-System Herstellers.

Für die Implementierung eines wissensbasierten Systems bieten sich grundsätzlich alle der oben genannten Programmiersprachen an. Das Konzept der Objektorientierung hat jedoch im Bereich der Anwendungsentwicklung in den letzten Jahren zunehmend an Bedeutung gewonnen. Ein wesentlicher Grund dafür ist die Erhöhung der Flexibilität und Wiederverwendbarkeit von Programmen. Im Gegensatz zu prozeduralen Programmiersprachen, in denen eine strikte Trennung zwischen Daten und Funktionen herrscht, werden bei einer objektorientierten Programmiersprache Objekte durch die Aggregation von Daten und Funktionen, die auf diesen Daten angewendet werden können (Methoden) gebildet. Objekte können weitere Objekte beinhalten und mit anderen Objekten über Nachrichten kommunizieren. Durch die Strukturierung der die Eigenschaften der Objekte definierenden Klassen in einer Klassenhierarchie lassen sich umfangreiche Bibliotheken erstellen. Diese können verbreitet und so wiederverwendet werden. Ein Beispiel für eine objektorientierte Programmiersprache mit einer sehr umfangreichen Klassenbibliothek ist Java.

Ein weiteres Kriterium für die Wahl einer objektorientierten Sprache für die Entwicklung eines wissensbasierten Systems ist, dass das Konzept der Objektorientierung sehr dem menschlichem Problemverständnis entspricht und sich auch in Methoden der formalen Problembeschreibung wiederfindet. Ein gutes Beispiel hierfür ist die Modellierung von Aufgabenstellungen mit Hilfe der graphischen Notation der UML. Die auf UML basierende MML wird in der MOKA-Methodologie (siehe Kapitel 3.2) für die Entwicklung wissensbasierter Anwendungen zur formalen Beschreibung der Objekte und ihrer Beziehungen untereinander genutzt. Anhand dieser Problembeschreibungsform kann bei der Nutzung einer objektorientierten Programmiersprache eine direkte Ableitung der Objektimplementierung erfolgen.

Eine besondere Stellung im Bereich der systemnahen Programmiersprachen nimmt das explizit als KBE-Sprache ausgewiesene Knowledge Fusion (KF) ein. KF basiert auf der KBE-Sprache

⁵ Der Begriff „Item“ wird hier synonym zum nicht gebräuchlichen Begriff „Artikel“ gewählt.

INTENT! der Firma HEIDE CORPORATION und ist vollständig im CAD-System NX integriert [NuLo05]. Im Gegensatz zu den anderen hier erwähnten Sprachen ist KF eine deklarative und objektorientierte Sprache. In Analogie zu den in Kapitel 2.6.5 erläuterten Wissensrepräsentationsformen kann KF also als frame- und regelbasiertes System aufgefasst werden, womit die Auswertung des Programmcodes (Produktionsregeln) durch eine Inferenzkomponente gesteuert wird. Die Programmierung erfolgt interaktiv im CAD-System oder auch in einem Editor. Jedes geometrische Objekt oder auch Feature des CAD-Systems entspricht einer Klasse (Frame) in KF. Analog zum Modellbaum existiert ein Strukturbaum für die Wissensrepräsentation im CAD-System, in dem die Instanzen der Klassen aufgelistet werden und bearbeitet werden können. Der Strukturbaum enthält des Weiteren Attribute (Slots mit Werten eines bestimmten Datentyps), die ihren Wert aus bestimmten Regeln erhalten. Es ist hierbei zu beachten, dass es sich hierbei nicht um Attribute handelt, die gemeinhin an geometrische Objekte des CAD-Modells vergeben werden können. In Abbildung 4-9 wird der Zusammenhang zwischen wissensbasiertem Teil (KF-Wissensrepräsentation) und CAD-Modell an einem einfachen Beispiel erläutert. Das dargestellte CAD-Modell eines Quaders mit Durchgangsbohrung kann durch die Subtraktion eines Zylinders von einem Quader erzeugt werden. Im CAD-System werden die dazu benötigten Features im Modellbaum hinterlegt. In der Wissensrepräsentation findet sich eine ähnliche Struktur wie im Modellbaum, jedoch werden hier die entsprechenden Objekte (z. B. ist der Quader eine Instanz der Klasse Block) hinterlegt. Die programmiertechnischen Objekte der Wissensrepräsentation und die Features im Modellbaum sind jeweils eine Abbildung aufeinander. Durch die Erzeugung von Verknüpfungen und die Integration beliebig komplexer Regeln ist es nun möglich, erweitertes Produktwissen in das Modell zu integrieren.

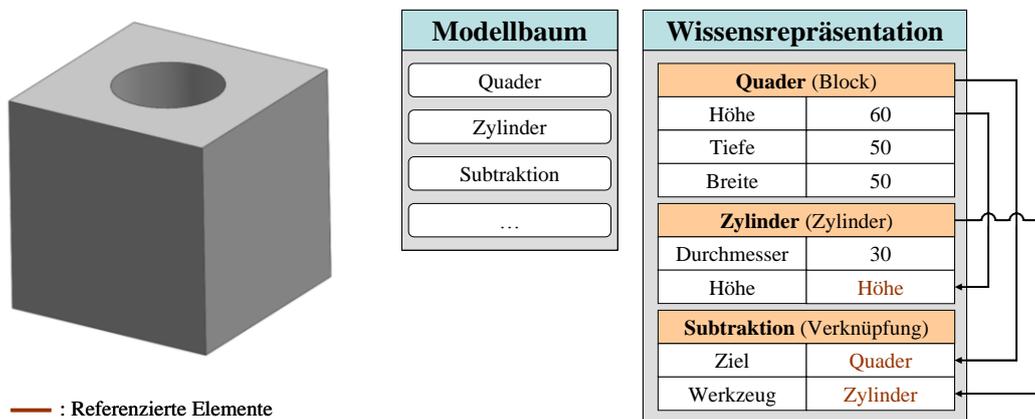


Abbildung 4-9: Koexistenz von CAD-Features und Wissensrepräsentation

Die Regeln und Algorithmen werden persistent in der Teiledatenbank gespeichert, können jedoch bei Bedarf gelöscht werden, ohne das eigentliche CAD-Modell zu zerstören. Es ist des Weiteren möglich, selbstdefinierte Klassen zu erzeugen und zu speichern und so unternehmensweit zugänglich zu machen. Für die Erzeugung von wissensbasierten Anwendungen steht zusätzlich ein Editor für die Erstellung graphischer Benutzeroberflächen zur Verfügung.

Die Möglichkeit, mit Hilfe einer einheitlichen Sprache sowohl Bauteile als auch Baugruppen mit umfangreichen Regelwerken ausrüsten als auch wissensbasierte Systeme erstellen zu können, prädestiniert KF als Sprache für dieses Konzept.

4.4.2.2 Schnittstellenarchitektur

Das Konfigurationssystem muss dazu in der Lage sein, sowohl mit dem CAD-System als auch mit den komponentenspezifischen Plug-Ins über definierte Schnittstellen zu kommunizieren. Beide Schnittstellen können nicht losgelöst voneinander betrachtet werden, da sie in integraler Weise voneinander abhängen. Die Plug-Ins haben in erster Linie die Aufgabe, die Daten der formalen Komponentenbeschreibung und der Nutzereingaben auszuwerten und zu verknüpfen, um daraus eine geeignete Strukturierung abzuleiten. Diese Struktur bildet dann die Grundlage für die Anordnung der Modellelemente und der Baugruppenelemente im CAD-System.

Die formale Beschreibung der Verdichterkomponenten liegt in rechnerauswertbarer Form vor. Die Komponente wird in protokollarischer Form durch globale, auftragsbezogene Daten und die Anordnung ihrer Gestaltungsmodule beschrieben, wobei der Aufbau der Beschreibung nicht unbedingt den Anforderungen der Modellierungsmethodik in einem CAD-System folgt. Daher wird die formale Beschreibung in eine hierarchisch strukturierte Listenform überführt (Abbildung 4-10), welche dann die Basis für alle weiteren Operationen mit den Modellelementen und den Baugruppenelementen bildet. Die in der Abbildung 4-10 dargestellte XML-Notation der Struktur wurde hier bewusst gewählt. Die Speicherung und Strukturierung von Daten in Form von XML-basierten Dokumenten hat den Vorteil, dass die Daten im Klartext lesbar und interpretierbar sind. Bei XML handelt es sich um eine standardisierte Datenbeschreibung, so dass die Übertragbarkeit der Strukturen und Daten in weitere Anwendungen gewährleistet ist. Da sich mit XML beliebige hierarchisch gegliederte Daten beschreiben lassen, findet sich hier eine Entsprechung zur systeminternen Datenspeicherung und Datenstrukturierung.

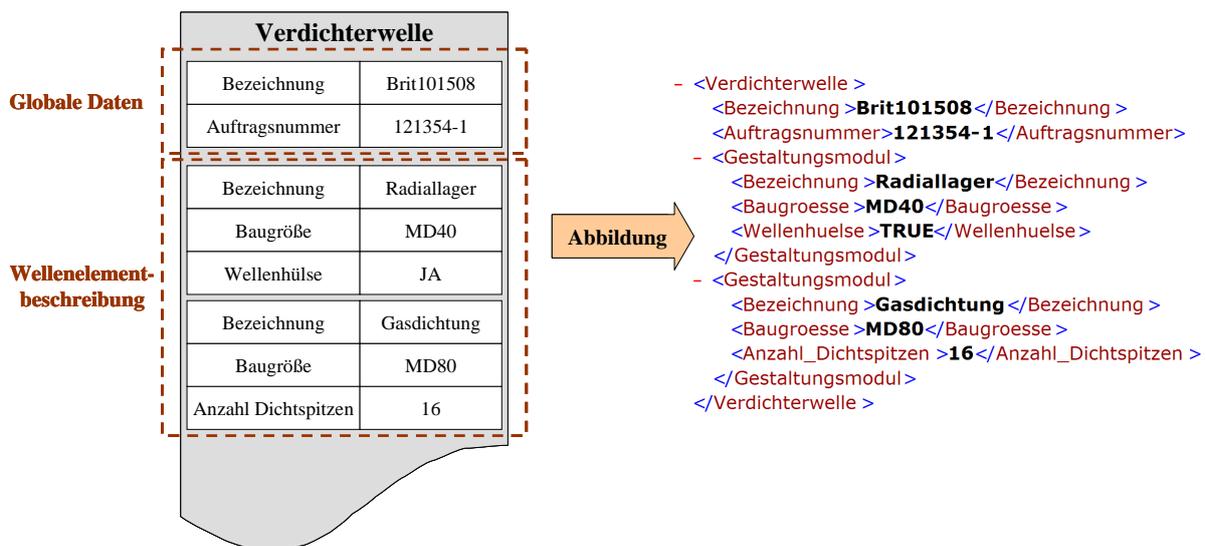


Abbildung 4-10: Abbildung der Komponentenbeschreibung auf eine Listenstruktur in XML-Notation

Durch die Vereinbarung der hier vorgestellten Struktur ist die Ankopplung von weiteren Plug-Ins ohne Manipulation des Hauptprogramms möglich. Ein positiver Effekt dieser Vorgehensweise ist die einfache Erweiterbarkeit der Systemfunktionalität im Hinblick auf ähnliche Komponenten. Die wesentlichen Funktionalitäten des Programms sollten jedoch, auch im Sinne der Bewahrung der Übersichtlichkeit des Programmcodes, nicht unbedingt angepasst werden müssen.

Aufgrund der vollständigen Integration der gewählten Programmiersprache KF in das CAD-System sind die informationstechnischen Grundlagen, welche für die Kommunikation mit dem CAD-System nötig sind, bereits vorhanden. Jedoch müssen geeignete Konzepte vorhanden sein, die eine systematische Suche von Elementen im CAD-Datenmodell erlauben. Die Strukturierung der Modellelemente auf Bauteilebene erfolgt im CAD-Modell sequentiell, da es sich hier um ein System mit hybridem Modellierkern handelt, in dem die Features in einem historisch geordneten Modellbaum angeordnet sind. Im Gegensatz dazu sind auf Baugruppenebene beliebige hierarchische Strukturen möglich. Die Gliederung der Modellelemente im CAD-System erfolgt daher auf der Basis einer Namenskonvention, die so das Auffinden der UDFs und damit auch ihrer internen Gestaltungselemente ermöglicht. Der Aufbau des Namens richtet sich nach der Position des Modellelements in der systeminternen Listenstruktur, somit erfolgt eine Abbildung der Datenstruktur auf das CAD-Modell. Dadurch wird erreicht, dass das CAD-Modell auch nach Beenden und Neustart des Konfigurationssystems durchsucht und manipuliert werden kann, wenn die gleiche Beschreibungsdatei geladen wird.

4.4.2.3 Benutzeroberfläche

Die Gestaltung der Benutzeroberfläche eines wissensbasierten Systems hat eine besondere Bedeutung für die spätere Akzeptanz des gesamten Konzepts. Eine für den Anwender undurchsichtige oder komplizierte Darstellung von ihm unbekanntem Sachverhalten oder eine nicht zielgerichtete Dialogführung kann dazu führen, dass die durch den Einsatz des Systems gewünschte Effizienzsteigerung im Produktentwicklungsprozess signifikant reduziert wird. Von NIELSEN wurde ein Modell erstellt, mit dem die Akzeptanz eines Systems nach verschiedenen Kriterien beurteilt werden kann (Abbildung 4-11).

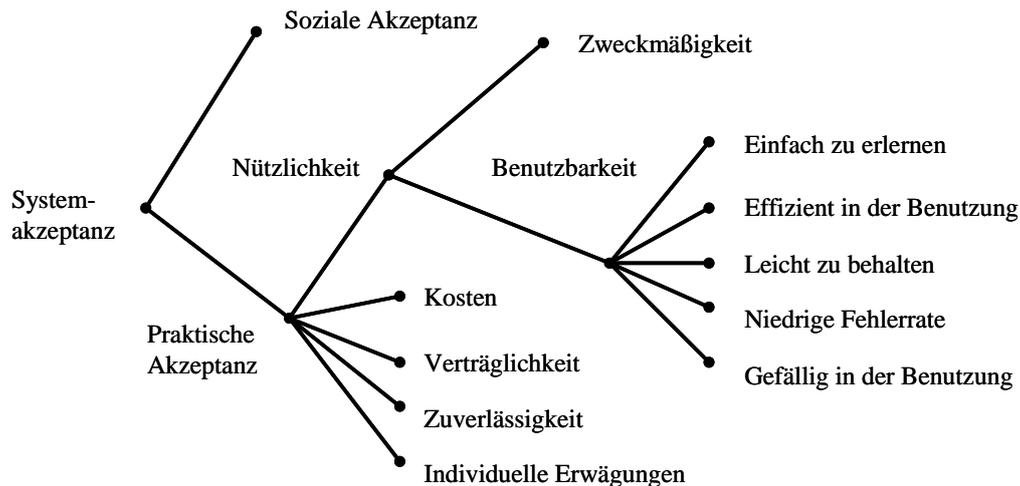


Abbildung 4-11: Akzeptanzmodell nach NIELSEN [Nie96]

Daraus leitete er zehn sogenannte „Heuristiken“ für die Gestaltung benutzerfreundlicher Dialoge ab. Ähnliche Ausführungen über die Gestaltung von Benutzerdialogen werden auch in [DIN84] gemacht.

- **Sichtbarkeit des Systemstatus**
Das System sollte den Anwender immer über die aktuellen Vorgänge informieren. Dies sollte durch eine angemessene Rückmeldung in angemessener Zeit erfolgen.
- **Übereinstimmung zwischen System und „Realer Welt“**
Das System sollte die Sprache des Anwenders sprechen. Dazu gehören Begriffe und Konzepte die dem Anwender gebräuchlich sind. System-spezifische Termini sollten keine Verwendung finden. Informationen sollten in einer natürlichen und logischen Ordnung erscheinen.
- **Anwenderkontrolle und Freiheit**
Wenn der Anwender eine im Kontext falsche Funktionalität aufruft, muss er durch einen klar hervorgehobenen „Sicherheitsschalter“ wieder in den Ausgangszustand zurückkehren können.
- **Konsistenz und Standardisierung**
Benutzer sollten nicht sich wundern müssen, ob verschiedene Wörter, Situationen oder Tätigkeiten die gleiche Sache bedeuten.
- **Fehlerprävention**
Mögliche Probleme sollten nicht nur durch Fehlermeldungen mitgeteilt werden. Vielmehr sollte das Konzept der Benutzeroberfläche dazu beitragen, dass mögliche Fehler gar nicht auftreten können.
- **Wiedererkennung statt Erinnerung**
Der Anwender soll sich durch eine geeignete Dialoggestaltung an bestimmte Situation und Zustände aus ähnlichen Dialogelementen erinnert fühlen.

- **Effektivität und Benutzerfreundlichkeit**
Erfahrenen Benutzern soll die Möglichkeit geboten werden, wiederkehrende Aktionen zu beschleunigen.
- **Ästhetisches und minimalistisches Design**
Es sollten nur die wirklich wichtigen und häufig benötigten Informationen präsentiert werden.
- **Fehlermanagement**
Der Anwender soll bei Auftreten eines Fehlers durch geeignet gestaltete Hinweise in die Lage versetzt werden, aus dem Fehler zu lernen. Dazu sollte eine Erklärungskomponente konstruktive Hinweise zur Fehlerbehebung geben.
- **Dokumentation und Hilfestellung**
Auch wenn der Anwender bei gut gestalteten Dialogen eigentlich keine Hilfestellung benötigen sollte, ist ein entsprechendes System vorzusehen. Dieses muss leicht zugänglich sein und alle wesentlichen Informationen kurz und verständlich präsentieren.

Diese Heuristiken haben eher den Charakter einer Daumenregel und lassen sich nicht direkt in Anweisungen für die Dialoggestaltung umwandeln. Jedoch sind sie ein gutes Kontrollinstrument für die Entwicklungsphase und helfen bei der Auswertung und Beurteilung der Rückmeldungen seitens der Anwender.

Ein weiteres Kriterium für das Erzielen einer hohen Benutzerakzeptanz ist die graphische Gestaltung der Dialoge in Anlehnung an die im CAD-System verwendeten Dialogkomponenten. Dadurch wird erreicht, dass beim Anwender ein Wiedererkennungseffekt erzielt wird, der ihm die Interaktion mit dem System erleichtert und das Gefühl vermittelt, eine integrierte Zusatzfunktion des CAD-Systems zu nutzen.

4.4.3 Featurekonzept

Die unterschiedlichen Gestaltungsmodule, die für den Aufbau der Verdichterkomponenten benötigt werden, werden mit Hilfe des UDF-Konzepts erstellt. Jedes UDF entspricht in diesem Sinne einer Klasse des Gestaltungsmoduls, wobei jedem Gestaltungsmodul ein entsprechender Konstruktionsstandard zugeordnet ist. Eine wesentliche Komponente des Gesamtkonzepts ist das Ziel, jedes UDF autark agieren lassen zu können. Zwar muss es möglich sein, Beziehungen zwischen verschiedenen UDFs aufzubauen, jedoch muss dabei gewährleistet sein, dass ein Aufbrechen der Beziehung nicht zu einer fehlerhaften Regeneration des UDFs führt. Um dieses Verhalten zu ermöglichen ist es nicht ausreichend, die UDFs mit den üblichen Methoden der parametrischen und featurebasierten Konstruktion zu erstellen. Dies ergibt sich durch folgende Anforderungen:

- Ankopplung einer externen Datenbank via ODBC-Schnittstelle
- Darauf aufbauende Verwaltung und Speicherung semantischer Informationen
- Stabile topologische Änderungen innerhalb eines UDFs

Diese Anforderungen postulieren die Erweiterung der UDFs um einen wissensbasierten Anteil.

Die geometrische Umsetzung eines UDFs, die Datenbankanbindung und die Integration wissensbasierter Anteile können nicht voneinander entkoppelt betrachtet werden, da sie einander bedingen. Das Geometriemodell ist die wesentliche Grundlage des virtuellen Produkts und Träger der grundlegenden Informationen. Die zur Gestaltbestimmung notwendigen Informationen werden durch die Abfrage einer Datenbank ermittelt. Die Anbindung an diese Datenbank kann nur durch die Erweiterung des Modells um einen wissensbasierten Anteil erreicht werden. Die Erweiterung ist auch notwendig, um ein erweitertes und anpassungsfähiges Geometriemodell zu ermöglichen.

Im folgenden Abschnitt wird daher zuerst das Konzept für die Datenbankanbindung der UDFs erläutert, da es die Grundvoraussetzung für die geometrische Gestaltdefinition ist. Daran anschließend wird dann das geometrische Konzept für die UDFs vorgestellt. Das Konzept hat zum Ziel, ein im Rahmen der Anforderungen stabiles Geometriemodell zur Verfügung zu stellen, um so ein qualitativ hochwertiges digitales Modell zu ermöglichen. Im Rahmen der darauf folgenden Diskussion werden die beiden Teilkonzepte im Verbund kritisch hinterfragt.

4.4.3.1 Datenbankanbindung und Informationsbereitstellung

Im Sinne einer unternehmensweit durchgängigen Datenhaltung werden alle in den Konstruktions- und Auslegungsstandards hinterlegten Daten in einer relationalen Datenbank gespeichert. Ein Zugriff auf die durch das DBMS verwalteten Daten erfolgt über die ODBC-Schnittstelle. Die Datenbank wird über das Betriebssystem deklariert. Dazu erfolgen die Angabe des für das DBMS notwendigen ODBC-Treibers und die Vergabe eines eindeutigen Schnittstellennamens. Das DBMS stellt eine Benutzerverwaltung zur Verfügung, so dass bei der Schnittstellendeklaration auch die Anmeldeinformationen eines geeigneten Benutzers eingetragen werden können. Für die Wahl der Anmeldeinformationen sollte die Funktionalität der Zugriffssprache SQL berücksichtigt werden. Verfügt der angemeldete Benutzer über die entsprechenden Rechte, so kann er Daten in der Datenbank manipulieren und auch löschen. Daher sollten für die Anbindung der UDFs nur Anmeldeinformationen eines leseberechtigten Nutzers eingetragen werden, um so die Datenintegrität zu schützen.

Die informationstechnische Anbindung der UDFs an die Datenbank erfolgt durch zwei Klassen, die sich in dieser Form auch in anderen objektorientierten Programmiersprachen finden. In einem ersten Schritt muss eine Verbindung zu einer Datenbank aufgebaut werden. Diese Verbindung erfolgt durch eine Instanz der Klasse „ODBC_Datenbank“. Die für den Verbindungsaufbau notwen-

digen Informationen sind der Name der ODBC-Schnittstelle und die Anmeldeinformationen des Benutzers, sofern diese nicht bereits im Treiber eingetragen worden sind.

Die Abfrage der Informationen erfolgt dann im Rahmen sogenannter „Recordsets“. Das dazu notwendige Objekt ist eine Instanz der Klasse „ODBC_Recordset“. Die Recordset-Klasse stellt Eigenschaften und Methoden zur Verfügung, mit denen die Abfrage, Manipulation und Verwaltung von Datensätzen ermöglicht wird. Das Recordset sendet das als String vorliegende SQL-Statement an das vorher angegebene Datenbankverbindungsobjekt und erhält das Ergebnis dieser Abfrage in Form einer einzeiligen Tabelle oder eines Wertes zurück. Eine wesentliche Funktionalität des Recordset-Objekts ist hierbei die Wandlung der in der Datenbank vorliegenden Datentypen in die entsprechenden Datentypen der verwendeten Programmiersprache. Durch die Abfrage der Datenbank mittels SQL-String können nun beliebig komplexe Abfragen dynamisch erstellt werden. Durch die Verknüpfung mehrerer String-Komponenten können die in den String-Typ konvertierten Eingabedaten des UDFs oder andere vorliegende Eingaben genutzt werden, um ein Abfrageergebnis aus der Datenbank zu erhalten. In der Abbildung 4-12 wird dies anhand von zwei Datensätzen dargestellt. *Datensatz1* ermittelt den Wert des Datenbankattributs *Bezugsgröße2* aus der *Tabelle 1* der Datenbank anhand des im Attribut *Bezugsgröße1* gespeicherten Wertes. Daraufhin nutzt das Objekt *Datensatz2* das Ergebnis dieser Abfrage, um alle verfügbaren Werte der *Tabelle 2* zu ermitteln.

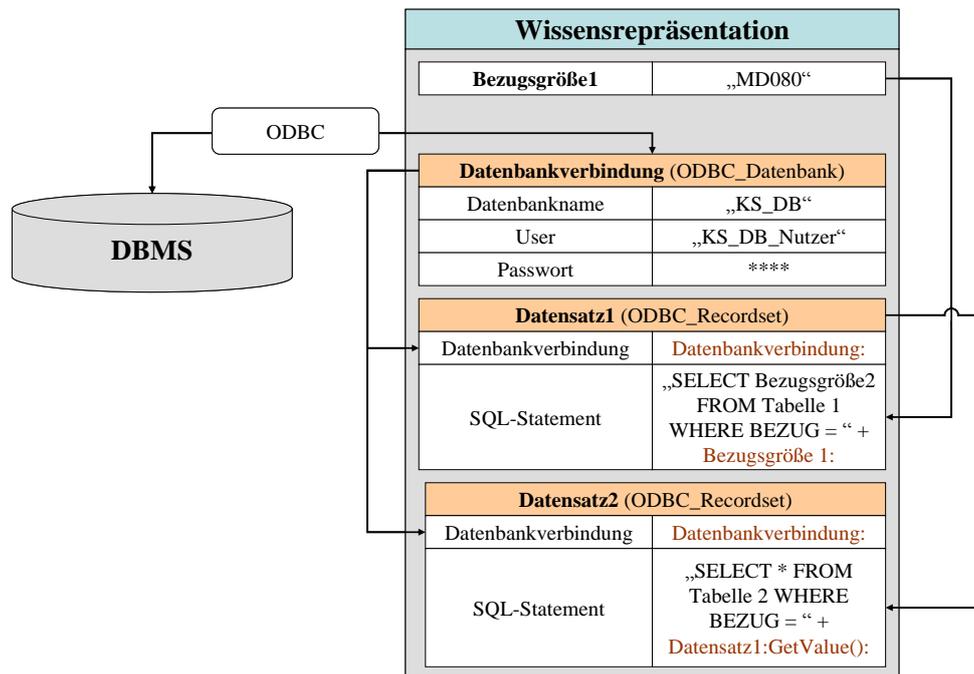


Abbildung 4-12: Datenbankanbindung und -abfrage

Eine für diesen Anwendungsfall wichtige Eigenschaft ist die Möglichkeit, Recordsets zu verknüpfen. Wie in Kapitel 4.1.2.2 erläutert, basieren die Ergebnisse bestimmter Abfragen auf den Ergebnisdaten eines vorhandenen Recordsets. Durch die Instanzierung mehrerer Recordset-Klassen können beliebig viele Datensätze zur Verfügung gestellt werden, deren Ergebnisse dann auch für den Aufbau eines SQL-Strings von weiteren Recordsets genutzt werden können.

Die in den Datenbankabfragen gesammelten Daten müssen nun dem geometrischen und dem wissensbasierten Teil des UDFs zugänglich gemacht werden. Dazu sind einige Vorüberlegungen notwendig. Eine wesentliche Anforderung an das Konzept besteht darin, die Manipulation der UDFs einem möglichst großen Anwenderkreis zu ermöglichen. Dazu gehören auch Anwender, die keine Kenntnisse im Bereich der Programmierung der wissensbasierten Erweiterung haben. Daher müssen die gesammelten Daten nach Möglichkeit in Form von Features oder Modellelementen bereitgestellt werden, die einem Anwender mit CAD-Kenntnissen bekannt sind. In diesem Fall wird die dynamische Transformation der im wissensbasierten Teil gespeicherten Daten in Modellparameter verfolgt. Jeder Parameter ist hier – in Analogie zu den Recordsets und der Datenbankverbindung – eine Instanz der Klasse „Parameter“. Die jeweilige Parameterinstanz erhält ihren Wert aus dem zugehörigen Recordset-Ergebnis, wobei eine Zwischenspeicherung der Ergebnisse in Form einer Liste erfolgt, um so eine im Verlauf der Programmierung einfachere Zugänglichkeit zu gewährleisten. Die Semantik der Parameter wird dem Benutzer sowohl durch eine geeignete Namensvergabe als auch durch die Vergabe eines erklärenden Kommentars vermittelt. Abbildung 4-13 verdeutlicht diesen Sachverhalt graphisch. Das Objekt *Datensatz1* ermittelt hier auf der Basis des Attributs *Bezugsgröße1* alle Werte der *Tabelle 1* der Datenbank. Die Werte werden in der Liste *Datensatz1_Werte* in Form von Parameter-Wert-Kombinationen gespeichert. Der eigentliche Parameter *Bauteillaenge* wird durch eine Instanz der Klasse „Parameter“ repräsentiert. Der Parameterwert wird mit einer Suchfunktion aus dem Attribut *Datensatz1_Werte* ermittelt. Eine zusätzliche Eigenschaft des Parameter-Objekts ist ein erklärender Kommentar. Der Kommentar wird in der globalen Parameterliste des CAD-Modells aufgeführt. Ändert sich die Eingabegröße *Bezugsgröße1*, so werden alle verknüpften Regeln erneut ausgewertet. Als Folge der Schlussfolgerung wird der Parameterwert entsprechend angepasst.

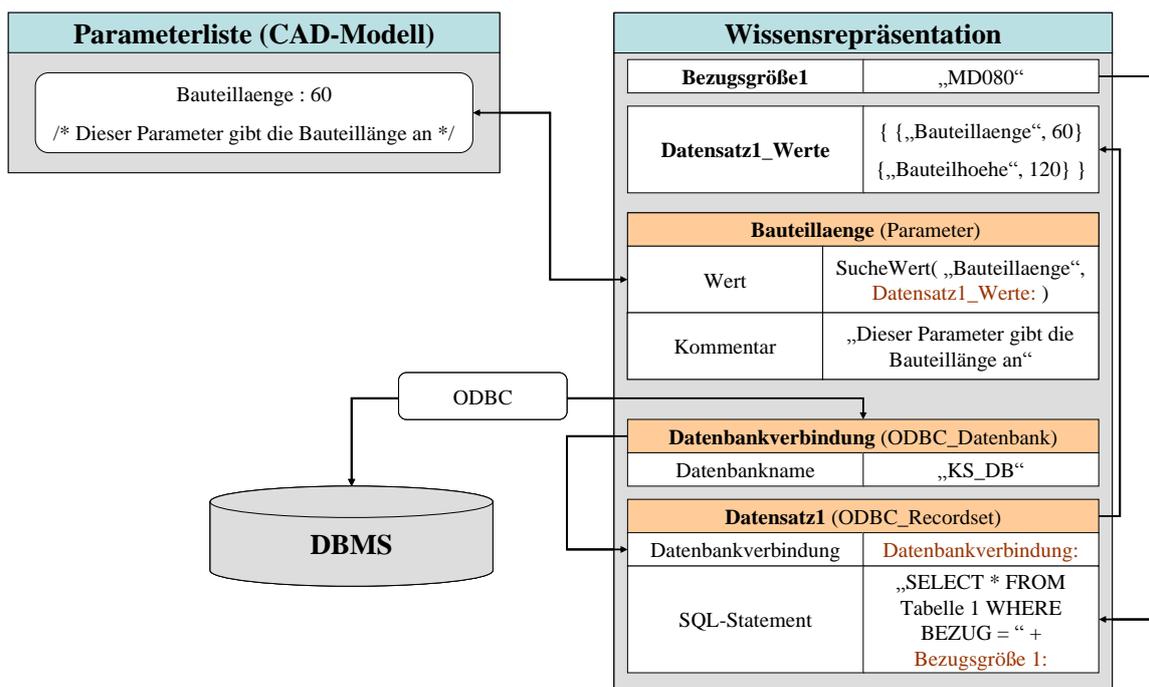


Abbildung 4-13: Datenbankgesteuerte Modellparameter

Die Verwendung von Parametern für die Gestaltbestimmung von CAD-Modellen ist ein allgemein bekanntes Konzept. Dem Anwender stehen durch die Bereitstellung der Parameter alle Freiheiten zur Verfügung, die Geometrie auf geeignete Weise zu beschreiben. Durch die hier erläuterte Vorgehensweise wird eine Trennung von interaktiver Modellierung und regelbasierter Datenakquisition erreicht. Der regelbasierte und damit programmierte Teil des Modells wird durch einen entsprechend geschulten Anwender erstellt. Das Ergebnis dieser Arbeiten ist eine Liste von Modellparametern. Durch eine geeignete Namensvergabe und die Bereitstellung der entsprechenden Kommentare kann ein direkter Bezug zu den Parametern des Konstruktionsstandards geschaffen werden. Damit können auch Anwender ohne Kenntnis der regelbasierten Komponente das Modell interaktiv auf der Basis der ihm bekannten Parameter in gewissen Bereichen verändern. Dabei muss jedoch die Modellstabilität beachtet werden. Enthält ein Modell eine Vielzahl verknüpfter Datenbankabfragen, so kann u. U. der Fall eintreten, dass ein Recordset keine gültigen Daten enthält. Dieses Ereignis kann z. B. durch eine Verletzung des Wertebereichs einer oder mehrerer Relationen in der Datenbank auftreten. Dadurch würde innerhalb des Modells in direkter Folge ein Parameterwert gesetzt werden, der zu einer nicht regenerierbaren Geometrie führen kann (z. B. die Länge Null als Bemaßung eines Linienelements in einer Skizze). Daher muss bei der Erstellung der Datenbankabfragen und der Verknüpfung mit den Modellparametern ein Fehlermanagement implementiert werden, das eine zumindest regenerierbare Geometrie erzeugt und den Anwender oder auch das Konfigurationssystem durch Nachrichten über die Wertebereichsverletzung informiert. Bei Bedarf können dann durch entsprechende im Modell implementierte Funktionalitäten die fehlerhaften Informationen manuell ergänzt werden.

Ein weiterer wichtiger Aspekt ist die Bereitstellung technisch relevanter Informationen für nachfolgende Prozessschritte (z. B. Zeichnungserstellung oder CAM-Bearbeitung), die keinen direkten Einfluss auf die Geometrie der Komponente haben. Dies können z. B. Toleranzinformationen oder auch technologische Informationen (z. B. Angaben über Bearbeitungsverfahren oder Produktstrukturinformationen) sein. Diese Informationen müssen in geeigneter Weise im Modell hinterlegt werden, wobei der Speicherort und die Verknüpfung zu Modellelementen für die Interpretation entscheidend ist. Eine Toleranzinformation bezieht sich immer auf ein oder mehrere geometrische Objekte und muss mit diesen verknüpft sein, wohingegen Informationen über das Gestaltungsmodul selbst einen globalen Charakter besitzen und damit von der Geometrie unabhängig sind. Das beschriebene Konzept der datenbankgetriebenen Modellparameter kann nun ebenso dazu genutzt werden, um Informationen sowohl an Geometrieelemente zu binden, als auch global bereitzustellen. Eine weitere Möglichkeit besteht in der Nutzung der wissensbasierten Komponenten als Informationsspeicher.

4.4.3.2 Geometrisches Konzept

Die geometrische Gestalt des UDFs ist die für den Benutzer wichtigste Eigenschaft, da sie die wesentlichen Merkmale des Produkts repräsentiert. In Bezug auf den automatisierten Aufbau des Ver-

dichterkomponentenmodells ist die Stabilität des geometrischen Modells von besonderer Bedeutung. Der Begriff „Stabilität“ bezieht sich hier auf die problemlose Regenerierung für den Fall der Parameteränderung. Der Anwender der Konfigurationssoftware hat meistens keinerlei Kenntnisse über die Entstehungsgeschichte und den Aufbau der einzelnen UDFs. Da ein UDF grundsätzlich auch dazu genutzt wird, die Modellentstehung vor dem Anwender zu verstecken, kann es auch vorkommen, dass die Modellpolitik des Unternehmens oder auch das CAD-System es nicht zulässt, das UDF aufzulösen. Fehler während des Regenerierungsprozesses im CAD-System können daher nur begrenzt durch den Anwender analysiert und behoben werden. Durch die fehlenden Eingriffsmöglichkeiten kann die Akzeptanz des Gesamtsystems sinken, was die Einführung in einen produktiven Betrieb gefährdet und somit auch das Investitionsrisiko der Systementwicklung erhöht.

Die geometrische Stabilität eines CAD-Modells wird in erheblichem Maße durch den Modellbau beeinflusst. Grundsätzlich kann für die Beurteilung der Modellstabilität zwischen zwei verschiedenen Ursachen unterschieden werden. Zum Einen müssen die systeminternen Vorgaben und Restriktionen, die prinzipbedingt aus der Wahl des CAD-Systems und damit des Geometrikerns ergeben, berücksichtigt werden. In diesem Zusammenhang ist vor allem die Verwendung von B-Rep Referenzen ein Problem, das im Folgenden besondere Berücksichtigung finden soll. Zum Anderen erfordert ein stabiles Geometriemodell eine belastbare Schnittstellendefinition. Mit dem Begriff „Schnittstelle“ ist hier zum Einen die geometrische Schnittstelle zu anderen UDFs und zum Anderen auch die Schnittstelle zu in den UDFs hinterlegten Informationen gemeint.

Auch wenn die geometrische Komplexität der hier zu erstellenden Gestaltungsmodule – ausgenommen die Beschaukelung der Laufräder - auf den ersten Blick gering ist (es handelt sich fast ausschließlich um Rotationskörper), so ergibt sich doch aufgrund der Vielzahl der darzustellenden Geometrien innerhalb eines UDFs eine hohe Komplexität im Modellaufbau. Dies ist auch dadurch bedingt, dass ein instanziiertes UDF zwar mehrere getrennte Volumina (je nach Leistungsfähigkeit des CAD-Systems) darstellen kann, im Modellbaum jedoch immer nur als ein Feature repräsentiert wird. Somit müssen Wege gefunden werden, um auch komplexe Topologieänderungen stabil durchführen zu können. Dieser Umstand soll an einem einfachen Beispiel hier erläutert werden. In Abbildung 4-14 sind zwei topologisch unterschiedliche Skizzen (Sketch 1 und Sketch 2) dargestellt. Diese Skizzen sollen die Basis für den Volumenkörper des UDFs bilden. Geht man davon aus, dass hier für die Modellierung keinerlei wissensbasierte Möglichkeiten zur Verfügung stehen, so ist es in einem 3D-CAD-System nicht möglich, auf die referenzierten Objekte einer Feature-Definition zuzugreifen und diese zu steuern. Aus diesem Grund kann hier das Extrusionsfeature nur auf ein interaktiv zugewiesenes Skizzenfeature zugreifen, die Eltern-Kind-Beziehung ist starr. Selbiges gilt für alle weiteren Operationen auf den durch die Extrusion entstehenden Volumenkörper. Das führt in diesem Fall dazu, dass hier eigentlich zwei voneinander unabhängige Modellbaumzweige erstellt werden. Die Auswahl des Volumenkörpers des UDFs erfolgt durch die parametergesteuerte Steuerung des Unterdrückungsstatus der jeweiligen Features einer Modellbaumverzweigung.

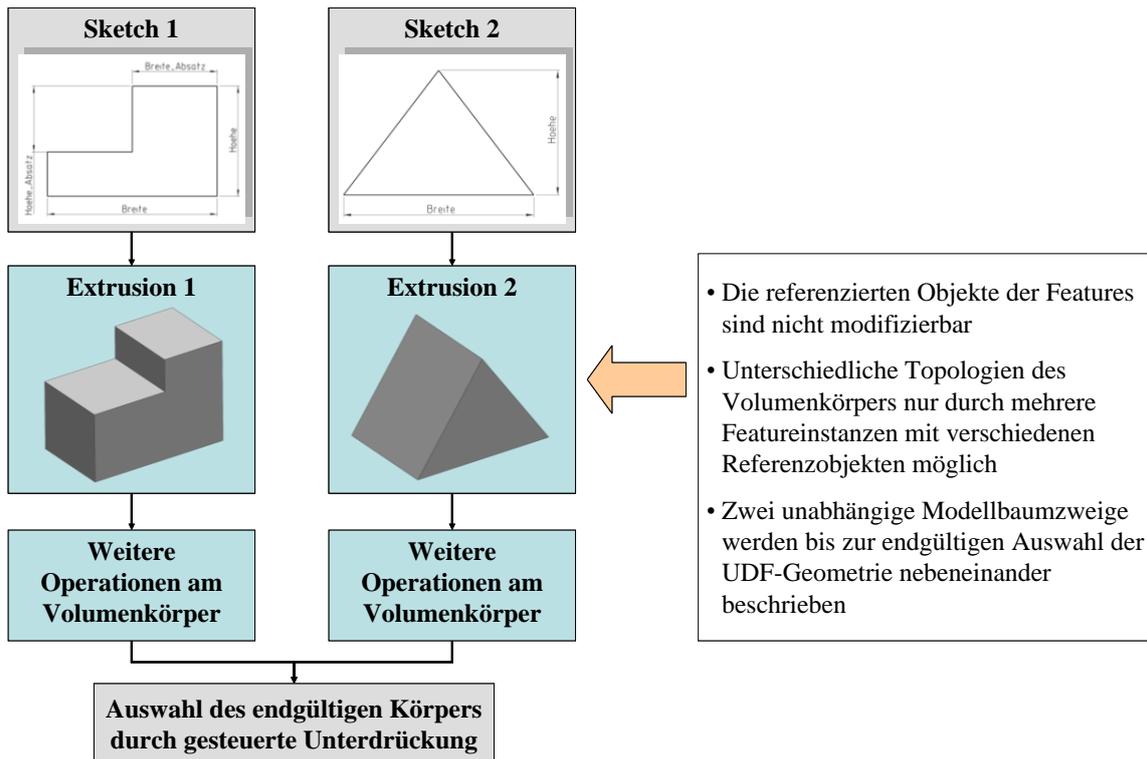


Abbildung 4-14: Einfache Erzeugung variabler Topologien

Die Erzeugung topologisch variabler UDFs mit der hier beschriebenen Methode ist sinnvoll, wenn die Nutzung von B-Rep Referenzen unumgänglich ist. Soll z. B. eine Fase an einer Volumenkante erzeugt werden, so kann hier interaktiv ein B-Rep Element gewählt werden, das immer dem entsprechenden Volumenkörper zugeordnet werden kann. Jedoch bleibt auch bei dieser Modellierungsmethode die Problematik der instabilen B-Rep Referenzen in seinen Grundzügen bestehen, da je nach Modellaufbau auch durch einfache Größenänderungen eine Modifikation des B-Rep Modells möglich ist. Ein wesentlicher Nachteil dieser Modellierungsmethodik ergibt sich durch die redundante Datenhaltung im Modell. Features, die eigentlich auf demselben Objekt (hier z. B. das Extrusionsfeature) operieren müssen nun mehrfach im Modellbaum aufgeführt werden und erhöhen so auch das für das UDF notwendige Datenvolumen.

Das gleiche Ergebnis kann auch durch die Verwendung wissensbasierter Komponenten erreicht werden (siehe Abbildung 4-15). Im Unterschied zur oben erläuterten Vorgehensweise wird das Extrusionsfeature nun durch ein entsprechendes Objekt in der Wissensrepräsentation abgebildet. Damit wird der Zugriff auf die Eigenschaften der Featuredefinition gestattet. Es ist nun möglich, die Referenz des der Extrusion zugrundeliegenden Features regel- oder algorithmenbasiert auszutauschen.

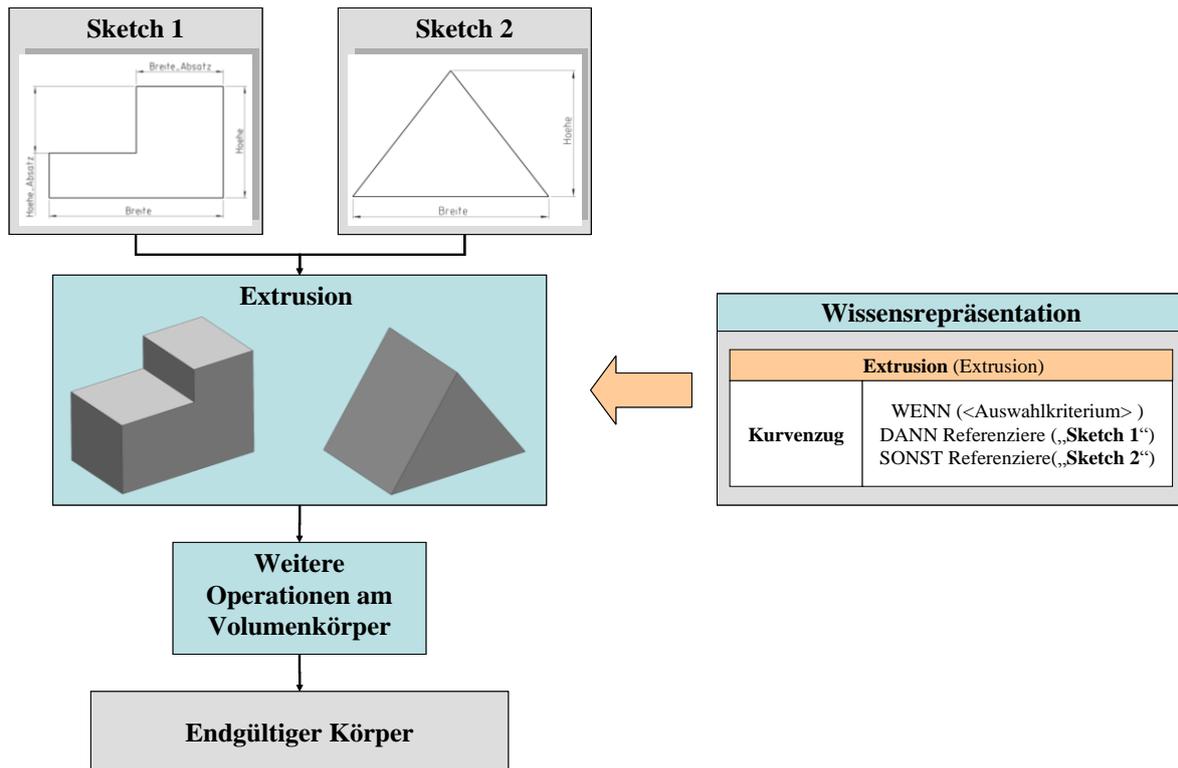


Abbildung 4-15: Wissensbasierte Erzeugung variabler Topologien

Diese Methode ist auch auf andersartige Features anwendbar und eröffnet somit vielfältige Möglichkeiten, um ein erweitertes Geometriemodell aufzubauen. Die Grenzen werden hier nur durch die vom CAD-System bereitgestellten Klassen und ihre zur Verfügung stehenden Eigenschaften und Methoden gesetzt. Bei der Umsetzung dieser Methodik in ein CAD-Modell ist jedoch eine strukturierte Modellierung unter Berücksichtigung der wissensbasierten Komponenten nötig. Dies betrifft insbesondere die Berücksichtigung der B-Rep Problematik. Anhand des letzten Beispiels ist zu erkennen, dass ein auf die B-Rep Elemente des Volumenkörpers referenziertes Element (z. B. Rundung an einer Körperkante) nur für jeweils eine Ausprägung des Körpers gelten darf. Dies bedeutet, dass z. B. in diesem Fall zwei Rundungen im Modellbaum instanziiert werden müssen, wobei der Unterdrückungsstatus der jeweils nicht benötigten Rundung durch eine mit der Ausprägung des Körpers verbundenen Regel gesteuert wird. Dadurch ergibt sich eine Kombination aus herkömmlicher Modellierungsstrategie und wissensbasierter Erweiterung, da die auf das B-Rep Modell referenzierten Elemente als Duplikate im Modellbaum auftauchen.

Die hier grundlegend erläuterte Modellierungsmethodik soll nun anhand eines komplexen Anwendungsfalls exemplarisch erweitert werden. Ein geeignetes Beispiel ist die Beschau felung der Laufräder mit gewundenen Schaufeln. Zum Einen ist die geometrische Gestalt dieser Schaufeln recht komplex, zum Anderen bildet dieses Beispiel eine gute Basis für die in Abschnitt 4.4.3.3 folgende Diskussion der Vor- und Nachteile der wissensbasierten Modellierung.

Die geometrische Gestalt der Beschau felung ist das Ergebnis thermodynamischer und strömungsmechanischer Berechnungen. Aufgrund der vorliegenden Berechnungsprogramme und der

damit verbundenen Modellbeschreibung können die Daten nicht direkt in ein CAD-Modell importiert werden (z. B. in Form von geometriebeschreibenden Punktefeldern, Kurvenzügen oder Berandungsflächen), sondern müssen aufbereitet und verarbeitet werden, um ein featurebasiertes Modell im CAD-System erzeugen zu können. Dies betrifft insbesondere die Umsetzung der parametrischen Größenvariation der Schaufeln. Um die dynamische Größenänderung der Beschauflung zu ermöglichen sind grundsätzlich zwei Anwendungsfälle zu berücksichtigen, die sich aus der Beschreibung der Schaufelgeometrie ergeben. Die Beschauflung der Laufräder lässt sich durch verschiedene Schaufeltypen klassifizieren. Jeder Schaufeltyp hat hierbei einen grundlegend ähnlichen geometrischen und topologischen Aufbau. Die Größenänderung (in Abhängigkeit des Durchmessers) kann im ersten Fall durch die Skalierung der Ausgangsdatenbasis mit einem spezifischen Faktor erfolgen. In diesem Fall bleibt die Ausgangsdatenbasis gleich, die Größenvariation erfolgt innerhalb der Algorithmen des Gestaltungsmoduls. Im zweiten Fall müssen bei einer Durchmesservariation neue Ausgangsdaten herangezogen werden.

Die geometrische Modellierung der Schaufel basiert auf der wissensbasierten Akquisition und Aufbereitung der Daten. Auf der Basis steuernder Attribute (z. B. Laufraddurchmesser und Laufradtyp) werden die benötigten Ausgangsdaten aus der Datenbank abgefragt und bereitgestellt (Abbildung 4-16).

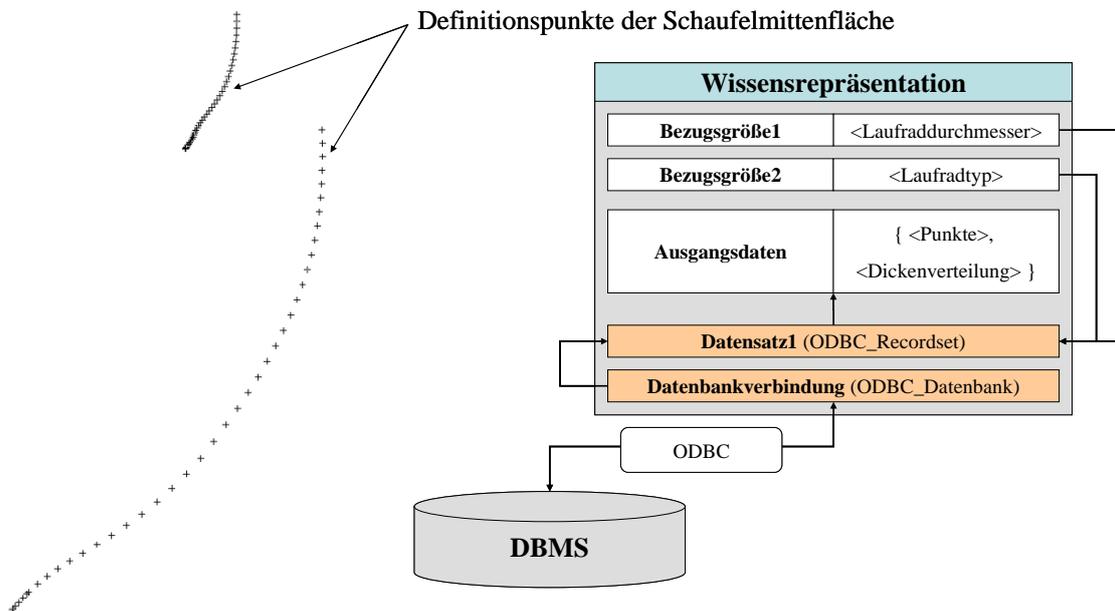


Abbildung 4-16: Ausgangsdaten der Beschauflung

Die gesammelten Ausgangsdaten werden nun in weiteren Attributen anhand vorgegebener Algorithmen weiter aufbereitet, wobei die detaillierte Vorgehensweise (Berechnungsvorschrift) hier nicht erläutert werden soll. Auf der Basis von umfangreichen Berechnungen in diesen Attributen erfolgt nun schrittweise die Überführung in eine für die interaktive Modellierung geeignete Form. Ergebnis dieser Berechnungen sind Definitionspunkte für die Begrenzungskurven der Schaufelflächen (Abbildung 4-17).

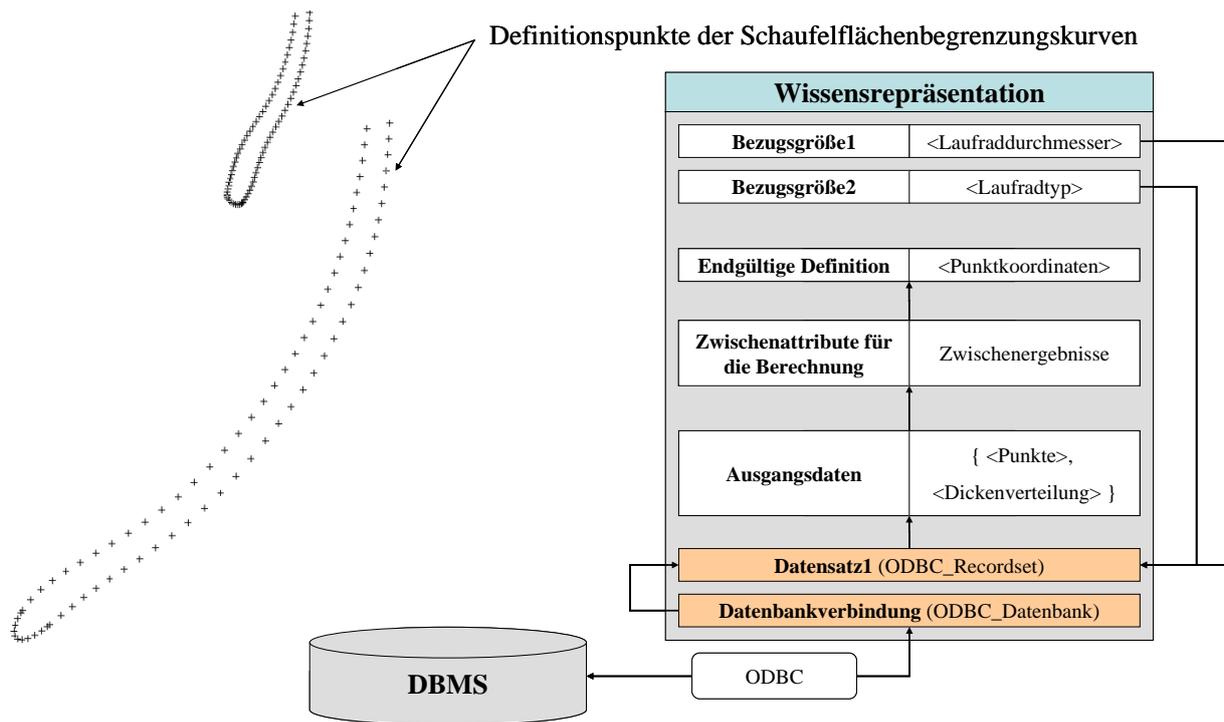


Abbildung 4-17: Verarbeitete Ausgangsdaten der Beschauflung

Auf der Basis der so bereitgestellten Definitionenpunkte kann nun die rein interaktive Modellierung der Schauffelgeometrie beginnen. Die interaktive Modellierung bietet ab hier den Vorteil der einfacheren Handhabbarkeit von Abhängigkeiten im Modell. In Analogie zu den skizzenbasierten Features erleichtert die Systemunterstützung in diesem Fall die Erzeugung von Eltern-Kind-Beziehungen und Referenzen, so dass ein strukturierter Modellaufbau möglich ist. Dazu werden die Definitionenpunkte als Stützpunkte für die Berandungskurven der Schauffel­flächen gewählt. Anhand dieser Kurven können nun alle weiteren Flächen der Schauffelgeometrie erzeugt und in einem Gestaltungsmodul bereitgestellt werden. Wird das Gestaltungsmodul in einem Modell instanziiert, so können die Bezugsgrößen an die neuen Anforderungen angepasst werden. Aufgrund der Änderungen der Attribute werden die verknüpften Objekte (Datenbank, abhängige Attribute) erneut berechnet und somit die Koordinaten der Definitionenpunkte angepasst. Die auf diese Definitionenpunkte bezogene Geometrie wird aufgrund der assoziativen Verknüpfung zu den Definitionenpunkten vom System automatisch aktualisiert.

4.4.3.3 Diskussion

Die Modellierungsmethodik für die Gestaltungsmodulare der Bauteilkomponenten hat eine entscheidende Bedeutung für die Stabilität des Gesamtsystems. Die hierzu entwickelte Methode kombiniert dazu die interaktive Modellierung mit wissensbasierten Komponenten. Diese werden je nach Anwendungsfall im Modell implementiert. Das primäre Ziel ist hierbei die Erweiterung des Modells um fehlende Funktionalitäten wie z. B. die Anbindung an eine externe Datenbank, die mit den üblichen Funktionen eines parametrischen CAD-Systems nicht umsetzbar sind. Des Weiteren wird ein strukturierter und stabiler Modellaufbau unterstützt, da systembedingte Nachteile (B-Rep Problema-

tik) in bestimmtem Maße umgangen werden können. Das Beispiel der Schaufel zeigt die Leistungsfähigkeit der Methodik auf. Die Komplexität der für die Berechnung der Schaufelkonturen notwendigen Algorithmen kann nicht durch einfache parametrische Funktionalitäten abgedeckt werden. Hier ist die Umsetzung in einer zu komplexen Berechnungen fähigen Programmiersprache unumgänglich. Jedoch ist es ab einem bestimmten Zeitpunkt der Modellerstellung nicht mehr sinnvoll, das gesamte Modell durch wissensbasierte Komponenten zu beschreiben, da die programmiertechnische Umsetzung von assoziativen Verknüpfungen und anderen Abhängigkeiten im Modell für die Übersichtlichkeit und Strukturierung des Modells, auch in Hinblick auf nachträgliche Anpassungen der Geometrie, nicht mehr förderlich ist. Die interaktive Modellierung der assoziativen featurebasierten Geometrie wird durch die CAD-Systeme sehr gut unterstützt und ist somit wesentlich effizienter.

Die Ergänzung des Geometriemodells um programmiertechnisch umgesetzte wissensbasierte Anteile muss jedoch auf strukturierte Weise erfolgen. Aufgrund der engen Verknüpfung der Partiamodelle (Featurebasierte Geometrieerzeugung und wissensbasierte Komponenten), die einander sogar teilweise bedingen, kommt dem systematischen Modellaufbau eine besondere Bedeutung zu. Jedes CAD-Modell sollte daher in geeigneter Weise dokumentiert werden. Diese Dokumentation kann auf zwei Arten erfolgen. Zum Einen muss das CAD-Modell so strukturiert werden, dass sowohl die Entstehungsgeschichte als auch die Abhängigkeiten der Features in annehmbarer Zeit durch den Anwender verstanden werden können. Dazu können alle aktuellen Konzepte für einen strukturierten Modellaufbau verwendet werden [Men99] [Köh02] [AbMe07]. Eine wichtige Komponente sind hierbei auch die durch die CAD-Systeme bereitgestellten Kommentarmöglichkeiten für die Features des Modellbaums. Zum Anderen ist es erforderlich, die wissensbasierten Komponenten des Modells in geeigneter Weise zu dokumentieren. Jede Komponente muss durch möglichst standardisierte Kommentare und Hinweise innerhalb des Codes erläutert werden. Hierbei ist insbesondere die Verknüpfung zu anderen Komponenten des wissensbasierten Modells zu berücksichtigen, um die sich ergebenden Abhängigkeiten nachvollziehbar zu machen. Eine standardisierte Namenskonvention kann sich dabei als hilfreich erweisen, um so bereits anhand der Komponentenbezeichnung etwaige Abhängigkeiten erkennen zu können. Auch in Hinblick auf die Verknüpfung von Geometriemodell und wissensbasierten Elementen ist die Nutzung einer Namenskonvention nützlich. Features, die durch eine wissensbasierte Komponente gesteuert werden können durch eine Bezeichnung im Modellbaum gekennzeichnet werden. Somit wird es auch dem geometriebezogenen Anwender ermöglicht eine Entscheidung darüber zu treffen, ob er fehlerfrei Modifikationen an diesem Feature vornehmen kann oder ob eine Modifikation der dazugehörigen wissensbasierten Komponente nötig ist.

Ein weiterer Aspekt, der sowohl bei der Modellierung als auch bei der Fehlersuche im Modell zu beachten, ist resultiert aus der Ablaufsteuerung der hier auf Regeln und Algorithmen aufbauenden wissensbasierten Komponenten. Wie bereits in Kapitel 2.6.6 erläutert, wird im Gegensatz zu prozeduralen Sprachen bei regelbasierten Systemen die Ablaufsteuerung nicht durch den Anwender fest-

gelegt. Das System bestimmt die Reihenfolge der Auswertung anhand der implementierten (und häufig auch nicht dokumentierten) Ablaufsteuerungsmethode. Hinzu kommt die Verknüpfung der regelbasierten Komponenten mit dem Geometrikern des CAD-Systems, der einen eigenen Aktualisierungsmechanismus besitzt (Abbildung 4-18). Steigt die Anzahl der im System implementierten Objekte und der dazugehörigen Regeln an, ist es nur schwer möglich, die Reihenfolge der Regelinterpretation und die Aktualisierung der Geometrie in Kombination nachzuvollziehen, wodurch die Fehlersuche und Fehlerbehebung teilweise erschwert wird.

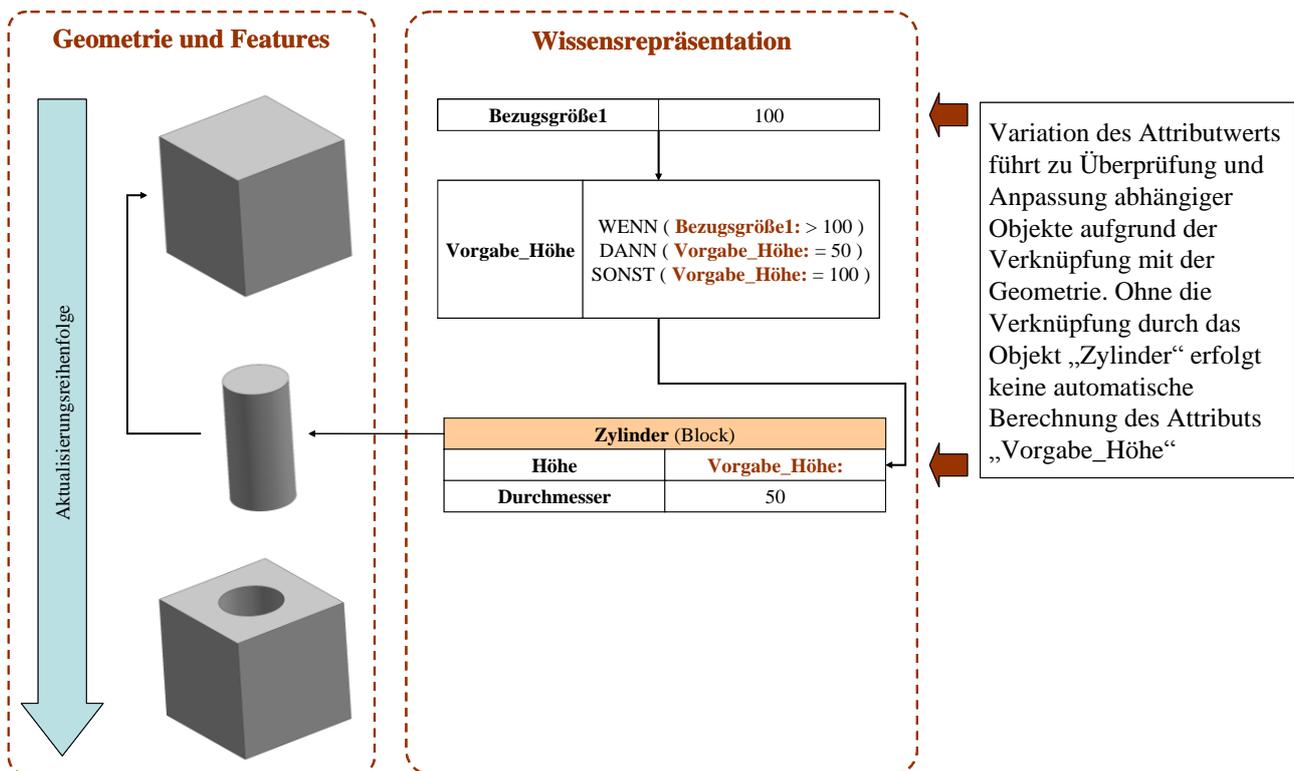


Abbildung 4-18: Kombinierte Aktualisierungssteuerung

Abschließend lässt sich feststellen, dass die Integration wissensbasierter Komponenten ein mächtiges Werkzeug für die Erweiterung der Funktionalität und Stabilität von CAD-Modellen ist, insbesondere in Hinblick auf die Integration und zielgerichtete Nutzung zusätzlichen Produktwissens. Die sich hieraus ergebenden Möglichkeiten können genutzt werden, um aus der sonst geometriegetriebenen hin zu einer funktionsgetriebenen Produktentwicklung zu wechseln. Der steigenden Komplexität der CAD-Modelle muss jedoch mit den hier vorgestellten Möglichkeiten der Modellstrukturierung und Dokumentation durchgehend begegnet werden, um die Vorteile der erweiterten Funktionalität vollständig nutzen zu können. Dies betrifft insbesondere die Möglichkeit der Wartung und Anpassung der CAD-Modelle.

4.4.4 Erzeugung der Komponententopologie

Um die gewünschte Komponentengestalt zu erstellen, müssen die einzelnen Gestaltungsmodule in geeigneter Weise miteinander verknüpft werden. Dazu müssen die Gestaltungsmodule im Raum

und auch zueinander positioniert werden. Darauf aufbauend erfolgt die Körperverknüpfung in Form Boolescher Feature.

4.4.4.1 Positionierung

Interaktiv erfolgt die Platzierung von Features zumeist durch die Wahl von Referenzelementen (geometrische Objekte). Teilweise wird aber auch die Platzierung durch die Vorgabe absoluter Koordinaten in Bezug auf ein Referenzkoordinatensystem durchgeführt. Die Automatisierung der interaktiven Platzierung mit der Hilfe geometrischer Referenzelemente lässt sich nur begrenzt softwaretechnisch umsetzen. Das größte Problem ist hierbei nicht die Auswahl geeigneter Referenzelemente. Diese lässt sich durch die Vergabe entsprechend standardisierter Namen zumeist stabil durchführen. Wesentlich schwieriger ist die stabile Positionierung aufgrund der geometrischen Mehrdeutigkeit einiger geometrischer Elemente. So hat z. B. eine Kante eines Körpers eine bestimmte Richtung und damit auch einen Startpunkt und einen Endpunkt. Das darauf zu referenzierende Element kann nun in seiner Richtung entsprechend angepasst werden. Diese Mehrdeutigkeit wird bei der interaktiven Elementauswahl durch den Anwender kompensiert. Dieser kann anhand der von den CAD-Systemen zur Verfügung gestellten Vorschau entscheiden, welche der möglichen Konfigurationen für seinen Anwendungsfall korrekt ist. Im Rahmen der softwaretechnischen Platzierung muss die Entscheidung des Anwenders simuliert werden, was jedoch nur schwer gelingt. Dennoch muss eine Platzierung der einzelnen Gestaltungsmodule erfolgen. Eine assoziative Verknüpfung der Gestaltungsmodule erfolgt daher hier durch die Nutzung vordefinierter geometrischer Bezugselemente. Diese hier als Konnektoren bezeichneten Elemente dienen als standardisierte Schnittstelle für weitere Gestaltungsmodule. Der Konnektor sollte möglichst eindeutige Richtungsvorgabe ermöglichen. Aus diesem Grund bieten sich hier Koordinatensysteme als bevorzugte Referenzelemente an. Sind die Richtungen der Konnektoren der Gestaltungsmodule vorgegeben, so kann eine stabile Platzierung erfolgen. Aus algorithmischer Sicht erfolgt dann die Verknüpfung der Konnektoren zweier Gestaltungsmodule auf der Basis der Namenskonventionen für diese Elemente. Die Position eines Konnektors innerhalb eines Gestaltungsmoduls muss hierbei nicht fixiert sein, sondern kann – in Analogie zu den Ausführungen in Kapitel 4.4.3.2 – durch wissensbasierte Komponenten entsprechend den Anforderungen modifiziert werden. Dies wird in Abbildung 4-19 am Beispiel von Gestaltungsmodulen für die Wellenkomponente exemplarisch dargestellt. Bei einer gleichsinnigen Ausrichtung der Gestaltungsmodule kann der Endkonnektor des ersten Wellenelements mit dem Anfangskonnektor des darauf folgenden Gestaltungsmoduls verbunden werden. Wird die Ausrichtung eines Gestaltungsmoduls geändert, so müssen die Referenzelemente der Konnektoren vertauscht und mit einem Winkel von 180° um die X-Achse gedreht werden. Aus programmtechnischer Sicht wird jedoch derselbe Konnektor verwendet. Die für die Drehung erforderlichen Aktionen werden durch einen entsprechenden Eingabewert des Gestaltungsmoduls gesteuert.

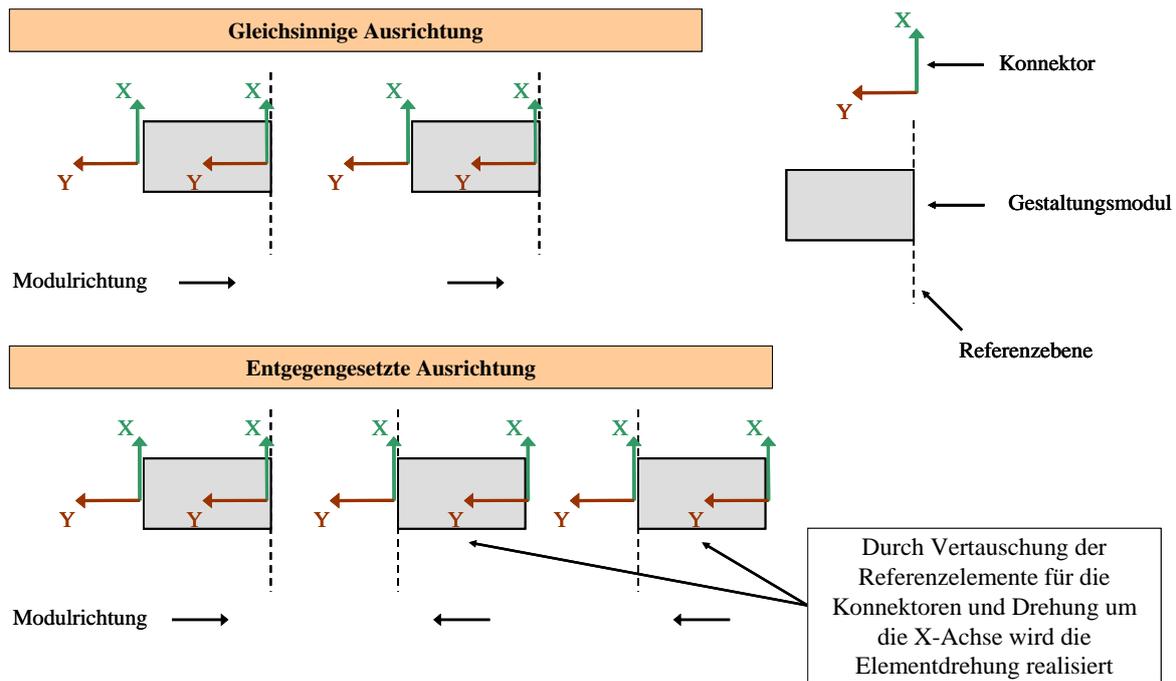


Abbildung 4-19: Einsatz von Konnektoren für die Ausrichtung der Gestaltungsmodule

Aufgrund des sequentiellen Modellaufbaus und der eindeutigen Identifizierbarkeit eines vorangegangenen Gestaltungsmoduls auf der Basis der Namenskonvention ist es des Weiteren möglich, Informationen für die Ausrichtung der Konnektoren innerhalb des CAD-Modells zu ermitteln. Dazu muss die für die Konnektoren verantwortliche wissensbasierte Komponente überprüfen, ob das vorangegangene Gestaltungsmodul alle notwendigen Informationen enthält und diese gegebenenfalls auswerten. Können keine Informationen ermittelt werden, so wird entweder die Standardausrichtung gewählt oder die Information aus der entsprechenden Eingabe berücksichtigt.

4.4.4.2 Boolesche Verknüpfung

Der Aufbau der gesamten Komponente basiert auf der in der formalen Beschreibung dokumentierten Produktstruktur. Die einzelnen Gestaltungsmodule werden sequentiell in das Modell eingeladen und zueinander positioniert. Ein Problem ergibt sich durch Gestaltungselemente, die formal zu einem bestimmten Gestaltungsmodul gehören, jedoch eine direkte Verknüpfung zu einem darauf folgenden Gestaltungsmodul aufweisen. Dies können z. B. Rundungen oder auch Freistiche am Ende eines Wellenelements sein. Diese übergreifenden Gestaltungselemente können nicht in die Gestaltungsmodule integriert werden und müssen daher als separate Gestaltungsmodule aufgefasst werden. Dieser Umstand muss in den komponentenspezifischen Plug-Ins bei der Erzeugung der Bauteilstruktur berücksichtigt werden. Aus diesem Grund erfolgt eine Unterteilung des Modellaufbaus in mehrere „Läufe“. Der erste Lauf dient dazu, die Hauptgestaltungsmodule in das Bauteil einzusetzen. Daraufhin erfolgt die mehrfache Instanzierung einer Klasse, die ein Feature vom Typ „Boolesche Verknüpfung“ repräsentiert. Mit diesen Features werden jeweils zwei Gestaltungsmodule zu einem Körper zusammengefasst. Die Art der Booleschen Verknüpfung (Vereinigung, Subtraktion,

Durchschnitt) wird als zusätzliche Information in die formale Bauteilbeschreibung integriert. Nach dem ersten Lauf werden die übergreifenden Elemente in das Bauteil instanziiert. Die Positionierung erfolgt auf der Basis der Konnektoren der entsprechenden Hauptgestaltungselemente. Im Gegensatz zum ersten Lauf werden nun die Gestaltungsmodul mit dem Basiskörper, bestehend aus den Booleschen Verknüpfungen des ersten Laufs, verbunden. Diese hierarchische Gliederung des Modellbaus kann in beliebiger Tiefe erfolgen, muss jedoch beim Aufbau der komponentenspezifischen Plug-Ins berücksichtigt werden, um eine geeignete Beschreibung der Modellstruktur zu ermöglichen.

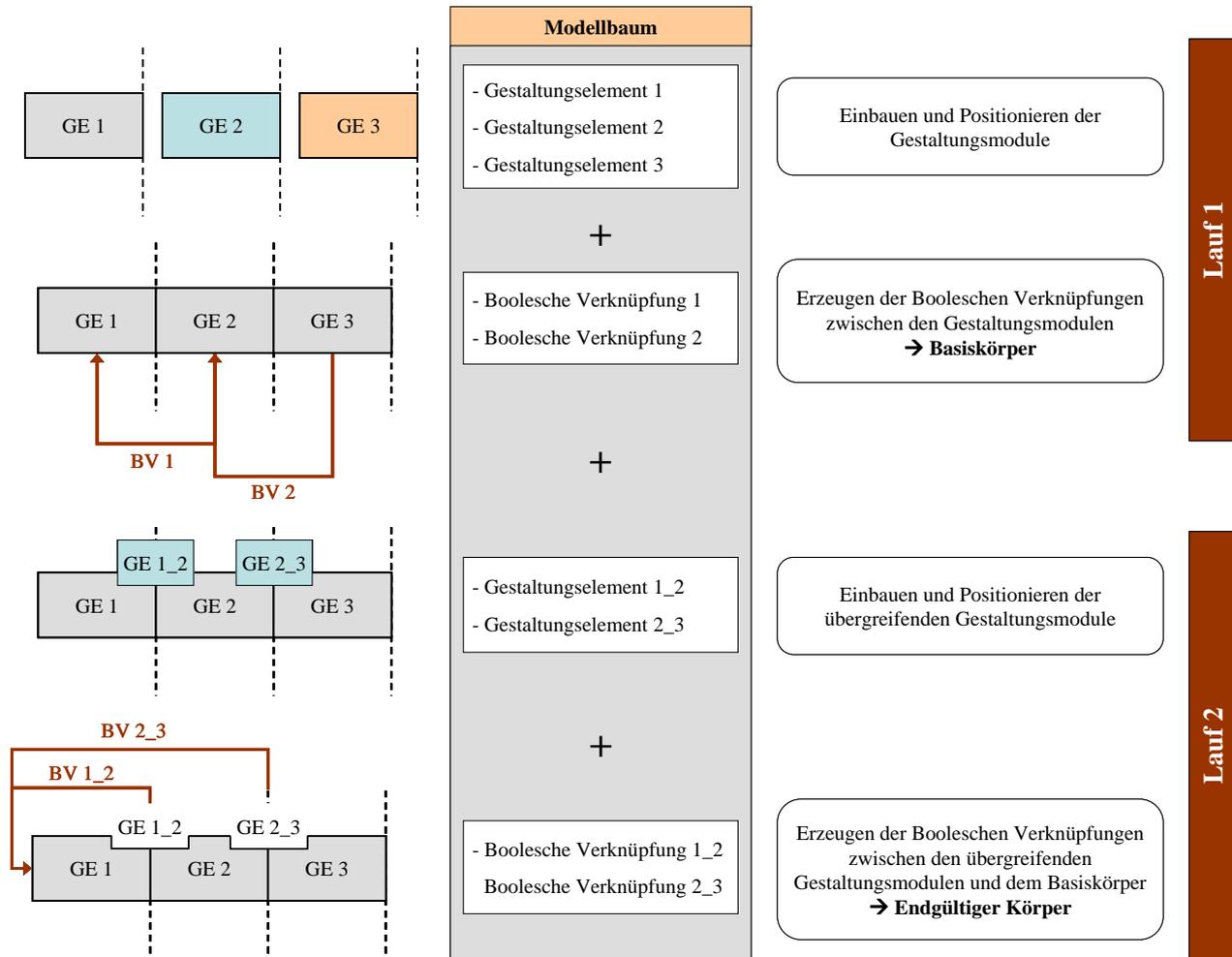


Abbildung 4-20: Erzeugung der booleschen Elementverknüpfungen

4.5 Umsetzung

In den folgenden Abschnitten wird die Umsetzung der erläuterten Konzepte detailliert dargestellt. Im ersten Abschnitt wird dazu der systemtechnische Aufbau des Konfigurationssystems dargestellt. Inhalt des darauf folgenden Abschnitts 4.5.3 ist die Darstellung des UDF-Aufbaus. Dazu wird exemplarisch die Modellierungsmethodik auf ein Gestaltungsmodul angewandt und die Schnittstelle zum Konfigurationssystem erläutert. In Abschnitt 4.5.2 wird dann der Aufbau der graphischen

Benutzerschnittstelle dargestellt, wobei die Interaktion des Anwenders mit dem Gesamtsystem in Abschnitt 4.5.4 anhand eines beispielhaften Anwendungsfalls näher erläutert wird.

4.5.1 Aufbau des Konfigurationssystems

Die Umsetzung des Konfigurationssystems erfolgt mit der für die UDFs verwendeten KBE-Programmiersprache KF, die einen objektorientierten Softwareaufbau erlaubt. Basis des Systems ist eine Hauptklasse, in der alle weiteren benötigten Sub-Klassen instanziiert werden (Abbildung 4-21). Jede Sub-Klasse repräsentiert spezifische Funktionalitäten, die über Methodenaufrufe und Attributabfragen des jeweiligen Objekts erfolgen.

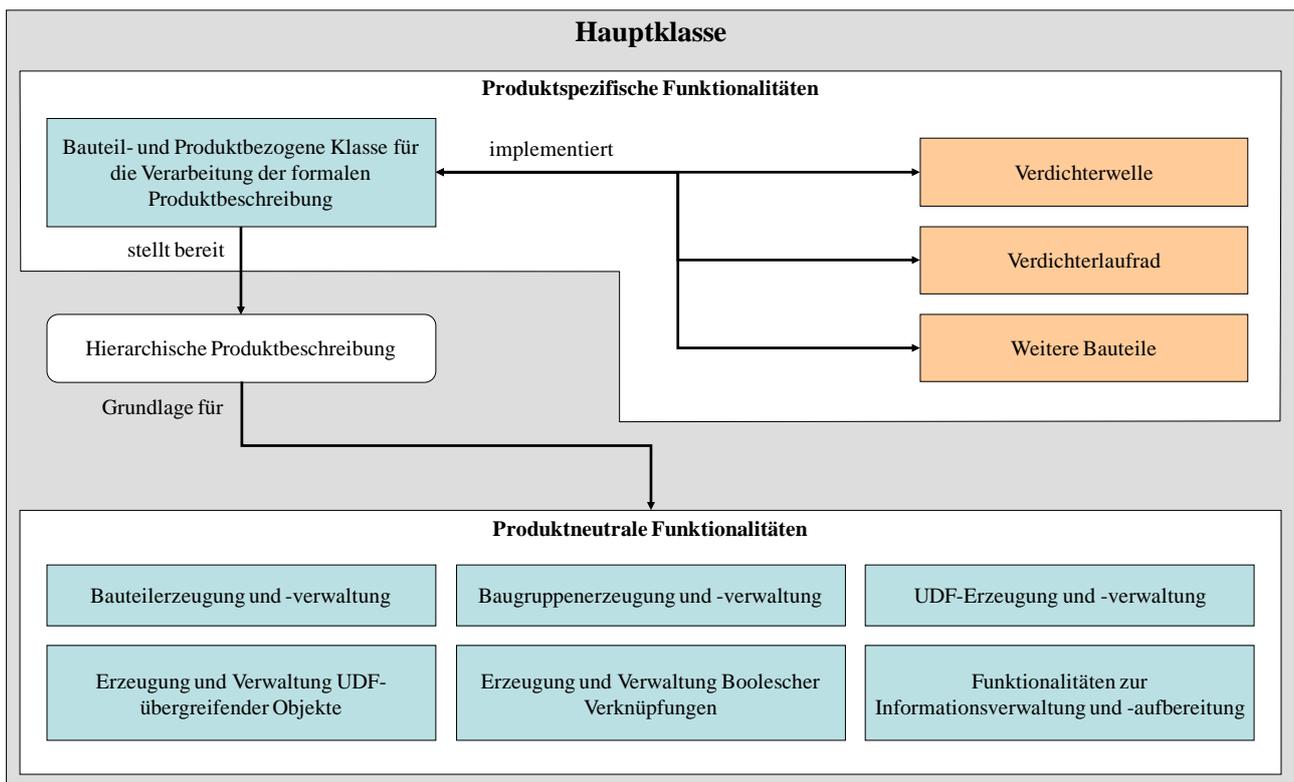


Abbildung 4-21: Applikationsstruktur

Die Übergabe von produktspezifischen Informationen innerhalb des Systems erfolgt grundsätzlich über die in Kapitel 4.4.2.2 definierte Listenstruktur in Form eines Listenattributs. Dessen hierarchischer Aufbau erlaubt die Abbildung der Produktstruktur und damit auch den Aufbau eines Benennungssystems, mit dem die UDFs, Bauteile und Baugruppen eindeutig identifiziert werden können. Der Objektname beginnt mit einem Präfix, das den Objekttyp spezifiziert (z. B. UDF). Das Suffix setzt sich aus Positionsangaben zusammen, die einen Rückschluss auf die Position des Objekts innerhalb der Produktstruktur geben.

Durch die Vereinbarung einer produktunabhängigen Schnittstelle ist es nun möglich, den größten Teil der Systemfunktionalität sehr allgemein zu halten. Eine Erweiterung des Produktspektrums erfolgt, indem eine zusätzliche Klasse implementiert wird, welche für die Datenakquisition und Datenaufbereitung im Kontext des neuen Produkts zuständig ist (Abbildung 4-22). Der erste Teil

der Klasse implementiert Objekte und Attribute, mit denen die Daten aus der formalen Produktbeschreibung ausgelesen und in eine auswertbare Form gebracht werden. Zusätzlich werden die vom Benutzer über den entsprechenden Dialog eingegebenen Werte und Anweisungen verarbeitet. Die hiermit auf globaler Ebene bereitstehenden Daten sind bis jetzt unabhängig von den im Produkt verwendeten Gestaltungsmodulen. Die Spezifizierung erfolgt dann im Bereich der Datenbereitstellung. Jedes Gestaltungsmodul wird hier durch eine entsprechende Klasse repräsentiert. Bei der Instanzierung der Klasse werden die notwendigen globalen Daten zur Verfügung gestellt. Anhand der für jedes Gestaltungsmodul spezifischen Logik werden diese Eingangsdaten nun ausgewertet und in eine Form gebracht, die direkt als Eingabeparameterblock für das im CAD-System einzubauende UDF genutzt werden können. Dieser Eingabeblock wird systemintern durch ein Listenattribut repräsentiert, das wiederum weitere Listenelemente enthält. Diese untergeordneten Elemente fassen je eine Parameter / Wert Kombination.

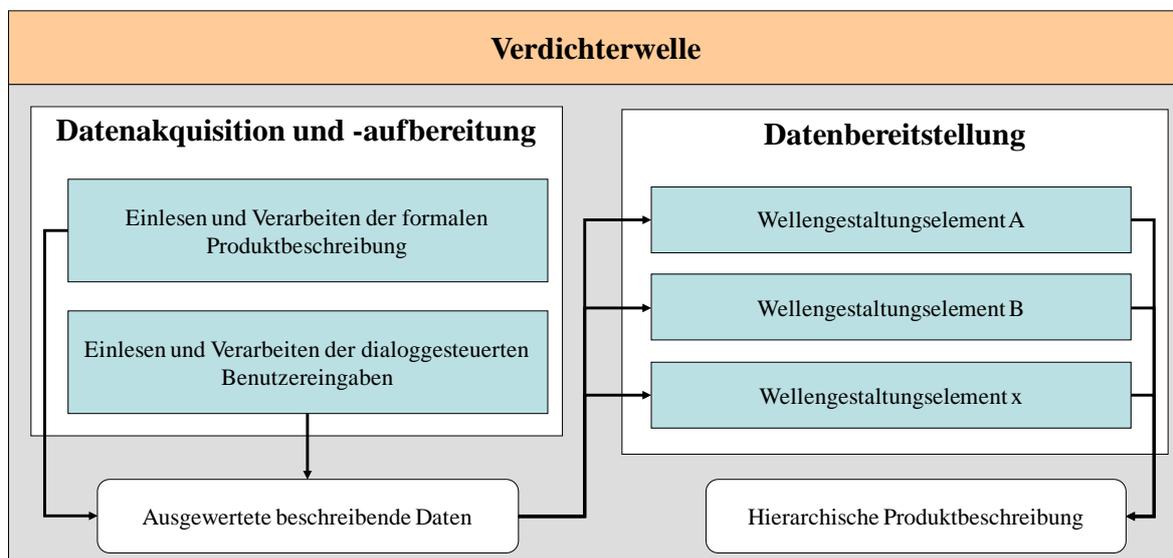


Abbildung 4-22: Produktspezifische Klassen

Die von außerhalb der Klasse ansprechbare Methode sammelt nun die durch die Gestaltungsmodulklassen bereitgestellten Informationen und strukturiert diese. Das Ergebnis dieser Prozeduren ist die hierarchische Produktbeschreibung.

Wie bereits erläutert muss für jedes Gestaltungsmodul neben dem UDF im CAD-System auch eine entsprechende Klasse existieren, welche dann für die Aufbereitung der produktspezifischen Informationen bereit steht. Die Klassen der einzelnen Gestaltungsmodule unterscheiden sich hierbei in der Art und Weise der Informationsverarbeitung und der Anzahl und Art der Attribute. Da sich die Klassen im strukturellen Aufbau jedoch gleichen ist es möglich, anhand des UDFs mit Hilfe einer Funktion einen entsprechenden Klassenrumpf aus den Eingabeparametern zu generieren. Dieser Klassenrumpf muss dann mit den gewünschten Funktionalitäten aufgefüllt werden. Dadurch wird zum Einen ein Beitrag zur Qualitätssicherung bei der Erweiterung der Anwendung geleistet und zum Anderen auch der Anwender bzw. der Entwickler in seiner Arbeit unterstützt.

4.5.2 Aufbau der Anwenderschnittstelle

Die Struktur und das Aussehen der Dialogkomponenten des Konfigurationssystems orientieren sich in erster Linie an den notwendigen Benutzereingaben. Es ist für die Akzeptanz der Gesamtanwendung von besonderer Bedeutung, den Aufbau der Dialogstruktur in Rücksprache mit den späteren Anwendern des Systems und entsprechenden Experten der Wissensdomäne zu gestalten. Dies betrifft insbesondere die Art der Informationsvermittlung durch den Benutzerdialog und die Führung des Anwenders durch die verschiedenen Dialoge. Zum Einen muss der Anwender sich bei der Abarbeitung „seiner“ Aufgabenstellung durch die Dialogführung unterstützt fühlen, zum Anderen sollte aber auch die Logik der Dialogführung aus dem CAD-System weitgehend übernommen werden, um so die für die Benutzung des Systems notwendige Einarbeitungsphase zu verkürzen. Nicht zuletzt spielen auch unternehmensinterne Vorgaben eine wichtige Rolle, insbesondere für die Ausarbeitung der graphischen Gestaltung.

Die graphische Benutzerschnittstelle des Konfigurationssystems kann interaktiv mit einem im CAD-System integrierten graphischen Editor erstellt werden (Abbildung 4-23). Das Basisdialogfenster kann hier in Analogie zum Formular-Editor der Programmiersprache Visual Basic um beliebig viele Kontrollelemente erweitert werden. Eine Besonderheit des hier verwendeten Editors (User Interface Styler) ist die systemdurchgängige Nutzung auch für Dialogkomponenten des Systemherstellers.

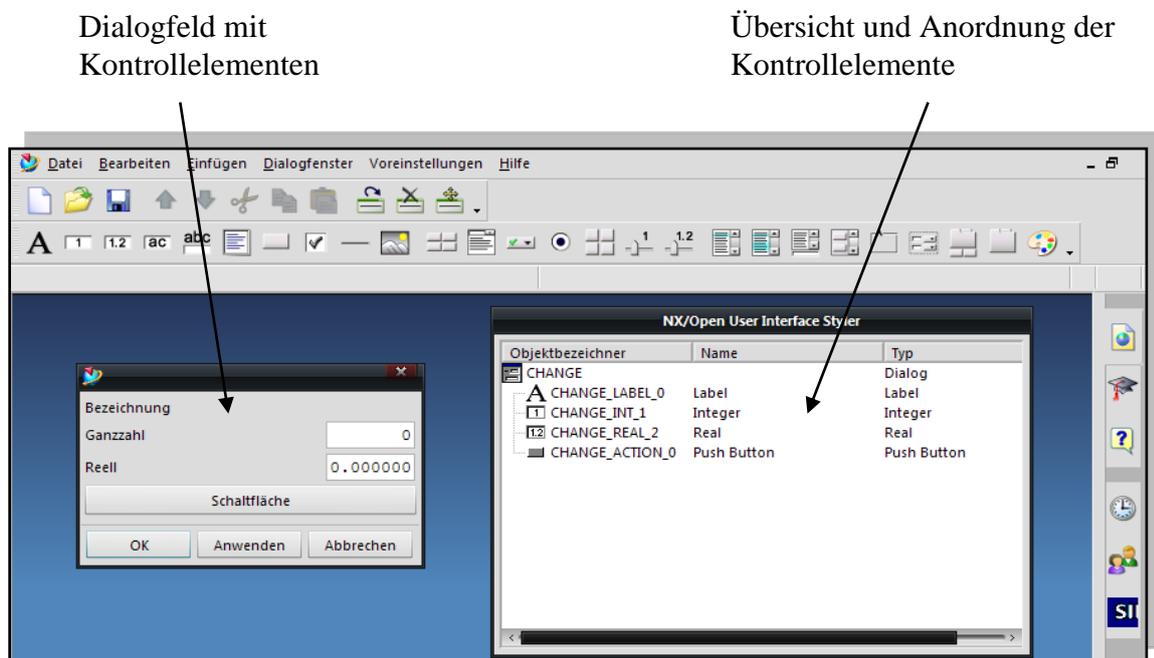


Abbildung 4-23: Aufbau von Dialogkomponenten im User Interface Styler von NX

Als Kontrollelemente stehen u. A. Schaltflächen, Eingabefenster für bestimmte Datentypen (z. B. Real, Integer, String), Auswahlfenster und Auswahlhaken zur Verfügung. Zusätzlich werden Elemente für die strukturierte Anordnung der Kontrollelemente bereitgestellt um den Aufbau be-

nutzerfreundlicher Dialoge zu ermöglichen. Jedes Kontrollelement erhält eine eindeutige Kennzeichnung. Die Verknüpfung der Dialogfeldkomponenten mit den Attributen und Objekten der Anwendung erfolgt durch Namensbindung. Somit kann mehreren unterschiedlichen Klassen dasselbe Dialogfeld zugeordnet werden. Voraussetzung hierfür ist, dass eine Abbildung der im Dialog vorhandenen Kontrollelemente auf die Attribute und Objekte der Klasse möglich ist.

4.5.3 Aufbau der UDFs

Der geometrische Aufbau, die Erzeugung der Datenbankbindung und die Verknüpfung mit den wissensbasierten Komponenten erfolgt mit den in Kapitel 4.4.3.1 und Kapitel 4.4.3.2 vorgestellten Methoden. Die praktische Umsetzung soll in diesem Abschnitt an einem einfachen Beispiel im CAD-System NX erläutert werden.

Grundlage jeden UDFs ist ein zumeist unternehmensspezifisches Startteil, in dem alle wichtigen Systemeinstellungen (z. B. Layervorgaben oder Farbgebung) gespeichert sind. Bevor mit der Körpererzeugung begonnen wird, werden in KF die notwendigen Eingabedaten als Attribute erzeugt (Abbildung 4-24).

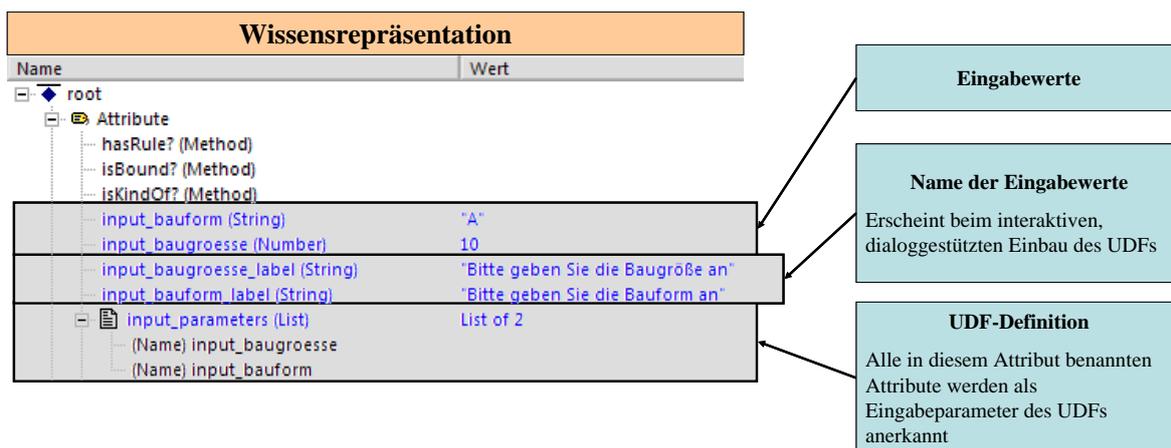


Abbildung 4-24: Erzeugung von UDF-Eingabedaten

Damit die Attribute in der UDF-Definition zur Verfügung stehen, muss im standardisierten Listenattribut „input_parameters“ auf ihre Bezeichnung verwiesen werden.

Mit diesen Attributen als Variablen kann dann die Datenbankabfrage erfolgen (Abbildung 4-25, einige Attribute wurden der Übersichtlichkeit halber entfernt). Dazu wird zuerst eine Instanz der Klasse „ug_odbc_database“ erzeugt. Diese baut die Verbindung zu der im Betriebssystem deklarierten Datenbank auf. Der Zugriff auf die Daten erfolgt über eine beliebige Anzahl von Recordset-Objekten. Diese knüpfen an die Datenbankverbindung an und senden einen mit den verarbeiteten Eingabedaten versehenen SQL-Befehl an die Datenbank. Für eine bessere Übersichtlichkeit werden die Ergebnisse der Abfragen dann in separaten Listenattributen in Form von Parametername / Wert Kombinationen zwischengespeichert. Bei einer Änderung der durch den SQL-Befehl referenzierten Eingabedaten wird die Datenbankabfrage aktualisiert und der Inhalt der Listenattribute angepasst.

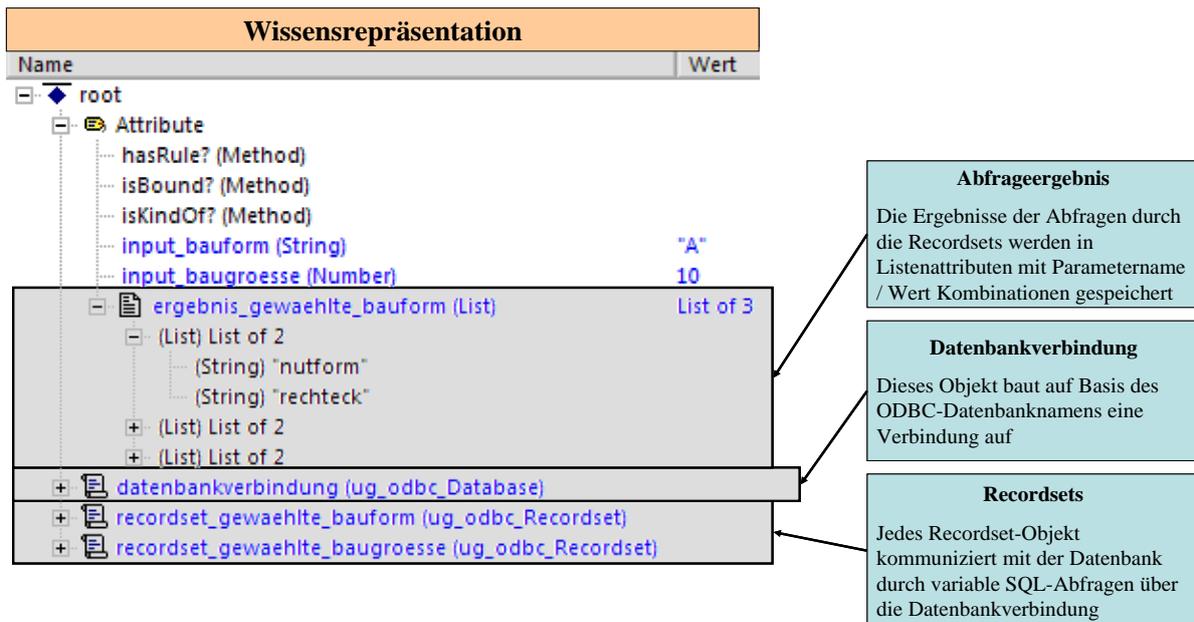


Abbildung 4-25: Erzeugung der UDF-Datenbankverbindung

Die so bereitgestellten Daten können nun innerhalb von KF auf beliebige Weise verwendet werden. Die Kopplung zum geometrischen Modell erfolgt auf zwei Arten. Zum Einen werden durch Instanzen der Klasse „ug_expression“ Modellparameter erzeugt, deren Wert dann durch die Daten gesteuert wird. Diese Modellparameter können z. B. für die Bemessung der Features genutzt werden. Zum Anderen können die zu manipulierenden Features direkt über ein entsprechendes KF-Objekt angesprochen werden. In Abbildung 4-26 wird dies anhand eines Beispiels verdeutlicht. Der Aufbau des geometrischen Modells beginnt mit der Erzeugung dreier Skizzenfeatures. Die erste Skizze repräsentiert den Grundkörper, die zwei anderen die Querschnittformen als Merkmale der jeweiligen Bauform. Die Skizzengestalt wird durch die datenbankgesteuerten Parameter eindeutig bestimmt. Daraufhin wird der Grundkörper als Rotationskörper erzeugt, ebenso wie der Körper mit halbkreisförmigem Querschnitt. An dieser Stelle wird im KF-Regelbaum der Querschnitt des zweiten Rotationskörpers durch eine Regel gesteuert. Ändert sich das Attribut „Bauform“, so wird für diesen Körper die Skizze mit dem rechteckigen Profil übernommen. Die beiden Körper werden nun mit einem Feature miteinander verknüpft. Auch hier greift eine entsprechende Regel, die den Typ der Verknüpfung anhand des Attributs „Bauform“ zwischen „Unite“ und „Subtract“ variiert. Die möglichen geometrischen Ergebnisse sind am Ende des Modellbaums dargestellt.

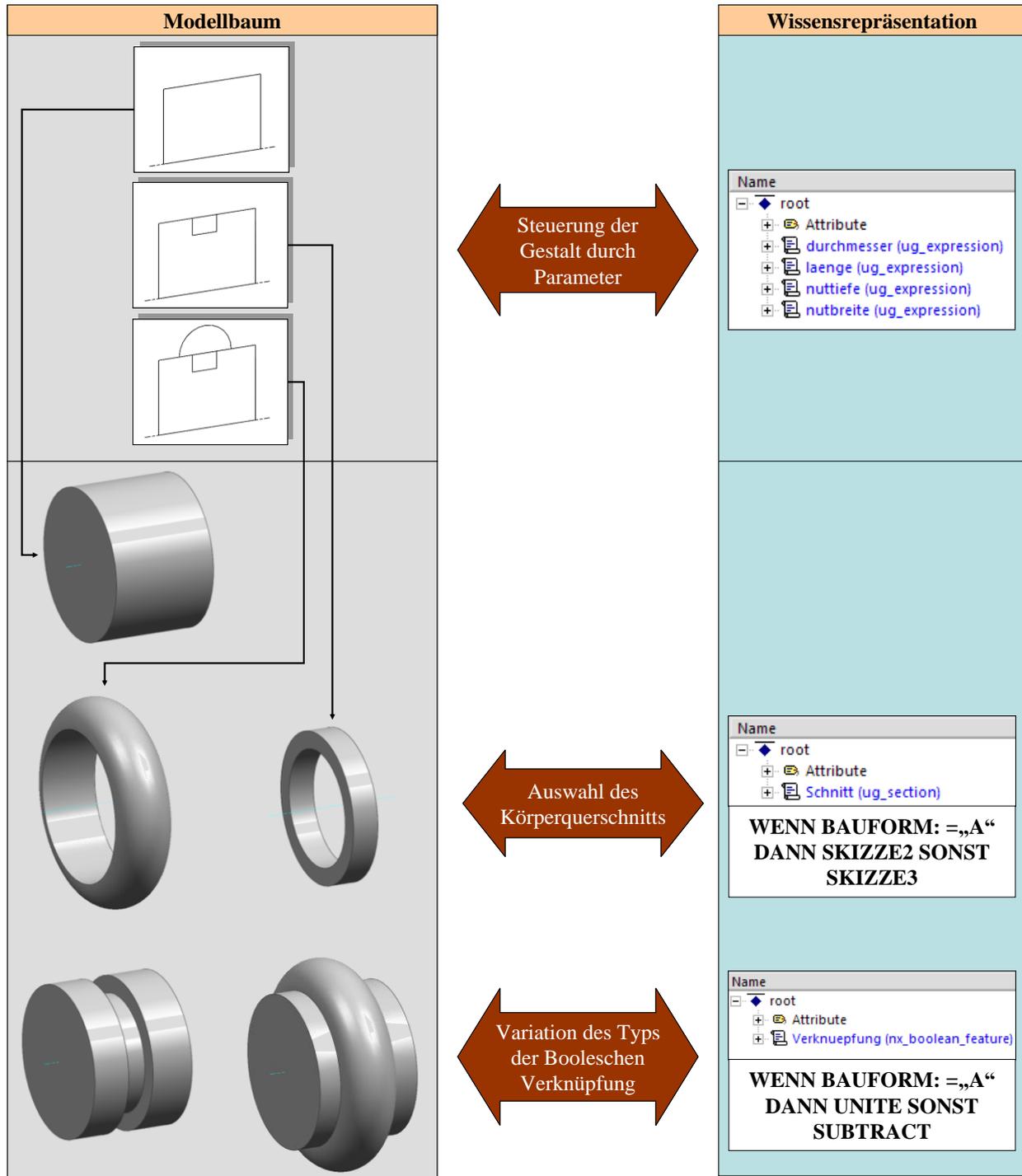


Abbildung 4-26: Exemplarischer Aufbau eines UDFs

4.5.4 Interaktion mit dem System

Der Prozess der Erzeugung von Verdichterkomponenten beginnt außerhalb des CAD-Systems. Ein unternehmensinternes Auslegungssystem erzeugt anhand der gestellten Anforderungen an das Produkt eine Konfigurationsbeschreibung der benötigten Verdichterkomponenten. Diese Beschreibung dient nun als Eingangsgröße für die Erstellung der Komponenten und Baugruppen im CAD-System.

Das Konfigurationssystem wird über einen in das Menü des CAD-Systems integrierten Button gestartet. Daraufhin öffnet sich der Hauptdialog (Abbildung 4-27).

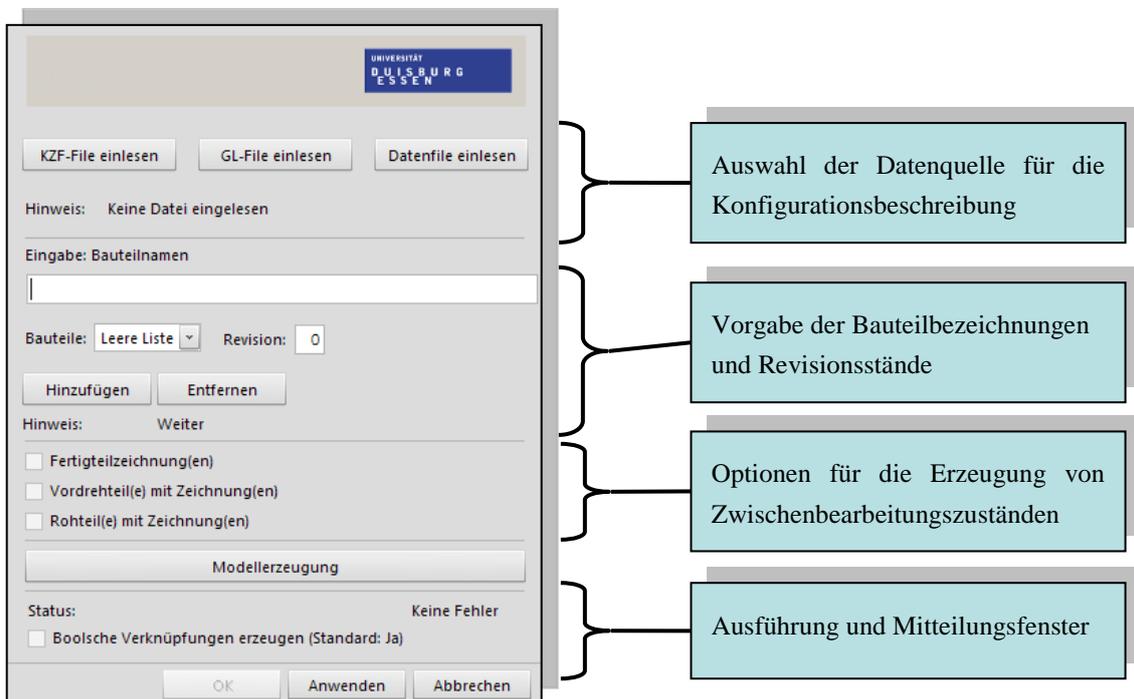


Abbildung 4-27: Hauptdialogfenster

Die im Hauptdialog getätigten Eingaben dienen in erster Linie dazu, Metadaten der zu erzeugenden Bauteile und Baugruppen zu sammeln, da diese in der PDM-Umgebung des Unternehmens erzeugt werden. Zudem kann der Anwender bestimmen, ob weitere Bearbeitungszustände der Bauteile erzeugt werden sollen. Das System teilt dem Anwender den aktuellen Status seiner Eingaben über ein Nachrichtenfenster mit. Zusätzlich wird er über eine dynamische Dialogsteuerung geführt, um so mögliche Fehleingaben zu vermeiden und eine Hilfestellung zu gewährleisten. Die Namensgebung der einzelnen Dialogkomponenten orientiert sich an den unternehmensintern gebräuchlichen Wendungen. Dadurch werden dem Anwender der Einstieg und der Umgang mit dem System erleichtert.

Durch die Wahl einer Datenquelle gelangt der Anwender in einen komponentenspezifischen Dialog, der ihn bei der Konfiguration unterstützt. Abbildung 4-28 zeigt beispielhaft Auszüge aus dem Dialogfenster für die Laufräder der Verdichterwelle. Die verschiedenen Bereiche des Laufrads werden durch einzelne Reiter repräsentiert, in denen der Anwender, unterstützt durch erläuternde Grafiken, Eingaben tätigen kann. Die Grafiken werden den Eingaben entsprechend dynamisch angepasst so dass der Anwender eine Rückmeldung auf seine Eingaben erhält.

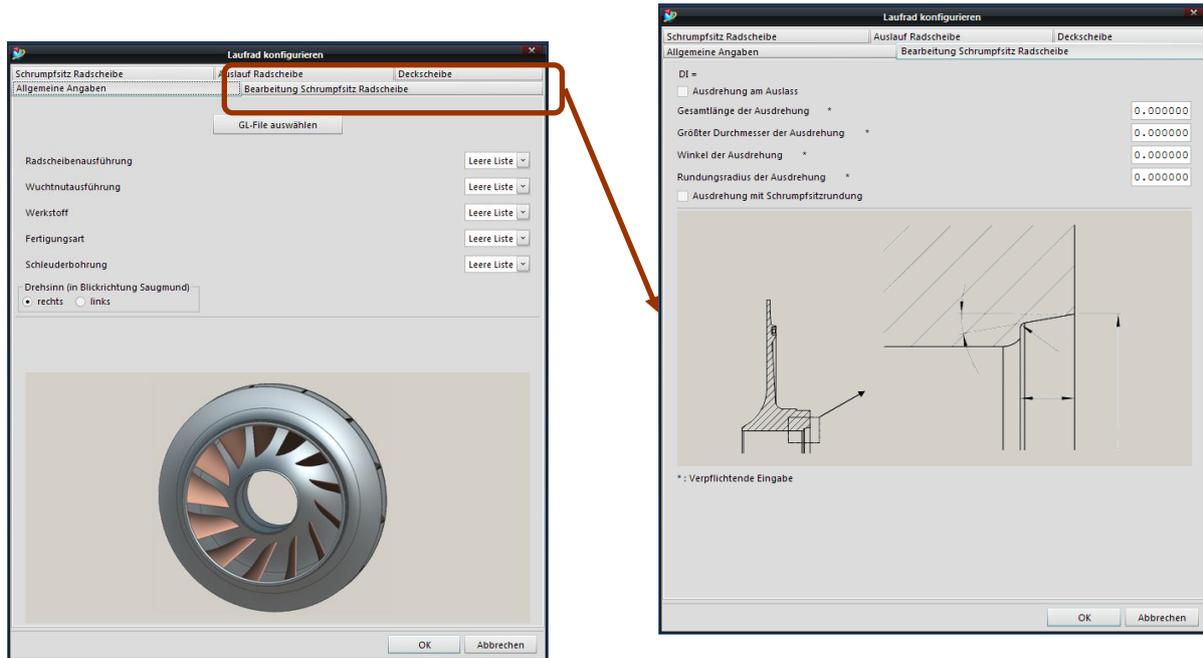


Abbildung 4-28: Komponentenspezifische Dialogkomponenten

4.6 Wirtschaftlichkeitsbetrachtung

Die Neuentwicklung eines Anwendungssystems birgt ein nicht unerhebliches finanzielles Risiko für ein Unternehmen. Die durch die Entwicklung, Anwendung und Wartung eines solchen Systems entstehenden Kosten müssen daher mittelfristig geringer sein als die möglichen Einsparungen, um so einen Gewinn erzielen zu können. Um den Einsatz eines neuen Softwaresystems betriebswirtschaftlich beurteilen zu können stehen verschiedene Hilfsmittel zur Verfügung. Eine häufig angewendete Methode ist die Kosten-Nutzen-Analyse. Mit dieser werden die durch die Investition anfallenden Kosten über den gesamten Lebenszyklus mit den Nutzeffekten verglichen. Sind die Nutzeffekte höher als die Kosten so ist der Einsatz des Systems aus unternehmerischer Sicht sinnvoll [WiMe09]. Bei der Betrachtung der Nutzeffekte wird jedoch deutlich, dass sich diese sehr oft nicht monetär quantifizieren lassen. Damit wird die Aussagekraft der quantitativen Ergebnisse der Kosten-Nutzen-Analyse abgeschwächt.

Eine umfassende Kosten-Nutzen-Analyse anhand des hier vorgestellten Systems würde den Rahmen dieser Arbeit sprengen. Jedoch sollen hier die wesentlichen Aspekte näher beleuchtet werden, die es zu berücksichtigen gilt. Dies betrifft insbesondere die Beurteilung der Nutzeffekte, die durch den Einsatz der hier vorgestellten Methoden und Systeme erreicht werden können.

Für die Planung, Entwicklung und Einführung des hier vorgestellten Softwaresystems fallen zunächst einmal Kosten an. Diese einmaligen Kosten werden beim Betrieb des Systems durch die laufenden Kosten abgelöst. Eine angepasste Gesamtübersicht über die projektbezogenen Kosten zeigt die folgende Tabelle (Tabelle 4-3).

Tabelle 4-3: Kosten des Softwaresystems

Einmalige Kosten	Laufende Kosten
<ul style="list-style-type: none"> ▪ Akquisition und Analyse impliziten Wissens aus unternehmensinternen Quellen (Bindung der Mitarbeiter und anderer Ressourcen) ▪ u. U. Konzeption und Aufbau der Datenbanken (nicht unmittelbar projektbezogen) ▪ Konzeption, Aufbau, Dokumentation und Test der Gestaltungsmodule ▪ Planung, Entwicklung und Einführung des Konfigurationssystems ▪ u. U. neue Hardware ▪ u. U. zusätzliche Softwarekosten ▪ u. U. erweiterte Infrastruktur 	<ul style="list-style-type: none"> ▪ Schulung der Mitarbeiter ▪ Wartung und Pflege des Konfigurationssystems ▪ Wartung und Pflege der Gestaltungsmodule ▪ Lizenzkosten ▪ Pflege und Wartung der Datenbanken (nicht unmittelbar projektbezogen) ▪

Den hier dargestellten Kosten stehen Nutzeneffekte gegenüber, welche durch die Anwendung des Systems zu erwarten sind. Im Gegensatz zu vielen materiellen Investitionsgütern können hier jedoch bereits in der Planungs- und Entwicklungsphase Nutzeneffekte erzielt werden. Für eine erfolgreiche Implementierung des Gesamtsystems ist es unabdingbar, das im Unternehmen vorhandene Wissen in Bezug auf das Produkt und seine Gestaltungsmodule zu erfassen und zu dokumentieren. Im Rahmen dieser Tätigkeiten werden so implizit vorhandene Wissensanteile in eine explizite Form überführt und so auch für weitere Anwendungen nutzbringend zur Verfügung gestellt. Des Weiteren verbessert – wenn nicht bereits geschehen – der Aufbau eines DBMS die Zugänglichkeit und Verfügbarkeit unternehmensweit benötigter Daten.

Das primäre Ziel beim Einsatz eines Konfigurationssystems ist die Automatisierung von Routinetätigkeiten des Konstrukteurs. Für die Bewertung der damit verbundenen Nutzeneffekte ist daher die Gegenüberstellung der zum Aufbau des Modells notwendigen Zeit sinnvoll. Aufgrund der vielfältigen Ausprägungen der Bauteile und Baugruppen (unterschiedliche Anzahl von Gestaltungsmodulen) kann hier jedoch keine allgemeingültige Aussage getroffen werden. Ein weiterer Einflussfaktor ist die Erfahrung des Konstrukteurs. Ist dieser bereits seit langem mit dem Anwendungsgebiet vertraut und beherrscht er die Modellierung der Geometrie mit dem CAD-System so ist er sicherlich in der Lage, die benötigten Daten schnell zu sammeln, zu verknüpfen und das Modell aufzubauen.

Tabelle 4-4 gibt einen Überblick über die notwendigen Tätigkeiten für die Erstellung einer typischen Komponente.

Tabelle 4-4: Vergleich zwischen manueller und automatisierter Modellerstellung

Manuelle Erstellung	Automatisierte Erstellung
<ul style="list-style-type: none"> ▪ Analyse der Komponentenkonfiguration ▪ Beschaffung aller notwendigen Datenquellen ▪ Auswertung der Datenquellen hinsichtlich der aktuellen Konfiguration ▪ Erstellung des CAD-Modells ▪ Erstellung und Anpassung der Fertigungsunterlagen 	<ul style="list-style-type: none"> ▪ Einlesen der Produktkonfiguration ▪ Starten des Konfigurationssystems ▪ Anpassung der Fertigungsunterlagen

Der Aufwand für die Datenbeschaffung und Datenaufbereitung nimmt hier aufgrund der hohen Datenmenge mit einer Vielzahl an Verknüpfungen den zeitlich größten Anteil ein. Ist dieser Arbeitsabschnitt abgeschlossen, kann mit der Modellierung des CAD-Modells begonnen werden. Dieses dient dann als Basis für die Zeichnungsableitung. Bei der automatisierten Modellerstellung werden die Zeichnungsdateien mit geeigneten Template-Dateien erzeugt. Durch den strukturierten Modellaufbau der Gestaltungsmodule können die im Modell hinterlegten Bemaßungen zu einem großen Teil auch für die Fertigungsunterlagen genutzt werden. Die hier notwendige Anpassung der Fertigungsunterlagen dient vor allen Dingen der normgerechten Darstellung. Bei der manuellen Modellerstellung kann ebenso von einem strukturierten Modellaufbau profitiert werden. Persönliche Vorlieben oder angelernte Arbeitstechniken des Anwenders beeinflussen die Modellstruktur jedoch. Einen Überblick über die zu erwartenden Nutzeneffekte des Konzepts gibt Tabelle 4-5.

Tabelle 4-5: Nutzeneffekte der automatisierten Modellerstellung

Prinzip	Nutzeneffekt
<ul style="list-style-type: none"> ▪ Standardisierter und strukturierter Aufbau der Gestaltungsmodule ▪ Datenbankbindung der Gestaltungsmodule ▪ Integration aller benötigten Wissensanteile in die Gestaltungsmodule ▪ Zeitersparnis durch Automatisierung ▪ Integration sämtlicher Ausprägungen eines Funktionselements in ein Gestaltungsmodul 	<ul style="list-style-type: none"> ▪ Sicherung der Modellqualität und der Modellstabilität ▪ Sicherung der Konsistenz und Aktualität der Daten ▪ Einfachere Wartbarkeit durch Entkopplung des Wissens vom Softwaresystem ▪ Entlastung von Routinetätigkeiten und Schaffung kreativer Freiräume ▪ Interaktive Modellierung aus funktionaler Sicht

4.7 Zusammenfassung der Methodik

Die in dieser Arbeit dargestellten Methoden und Ansätze wurden anhand konkreter Produktkomponenten (Welle und Laufräder eines Radialverdichters) entwickelt und in einem funktionsfähigen Anwendungssystem implementiert [DaLu08][DaLu08a][DaLu08b][DaLu08c]. Eine Verallgemeinerung des Gesamtkonzepts auf andere Problemstellungen ist aufgrund der Vielzahl an anwendungs-

spezifischen Anforderungen und Gegebenheiten nicht ohne Weiteres möglich. Dennoch sollen in diesem Abschnitt die Übertragbarkeit der wesentlichen Aspekte des Gesamtkonzepts auf andere Problemstellungen und die sich daraus ergebenden Anforderungen erläutert und diskutiert werden.

4.7.1 Grundkonzept

Abbildung 4-29 zeigt die Einordnung des hier entwickelten Konfigurationssystems auf Basis eines Design Repositories, bestehend aus Gestaltungsmodulen mit DBMS-Anbindung, in den Gesamtprozess der Produktkonfiguration.

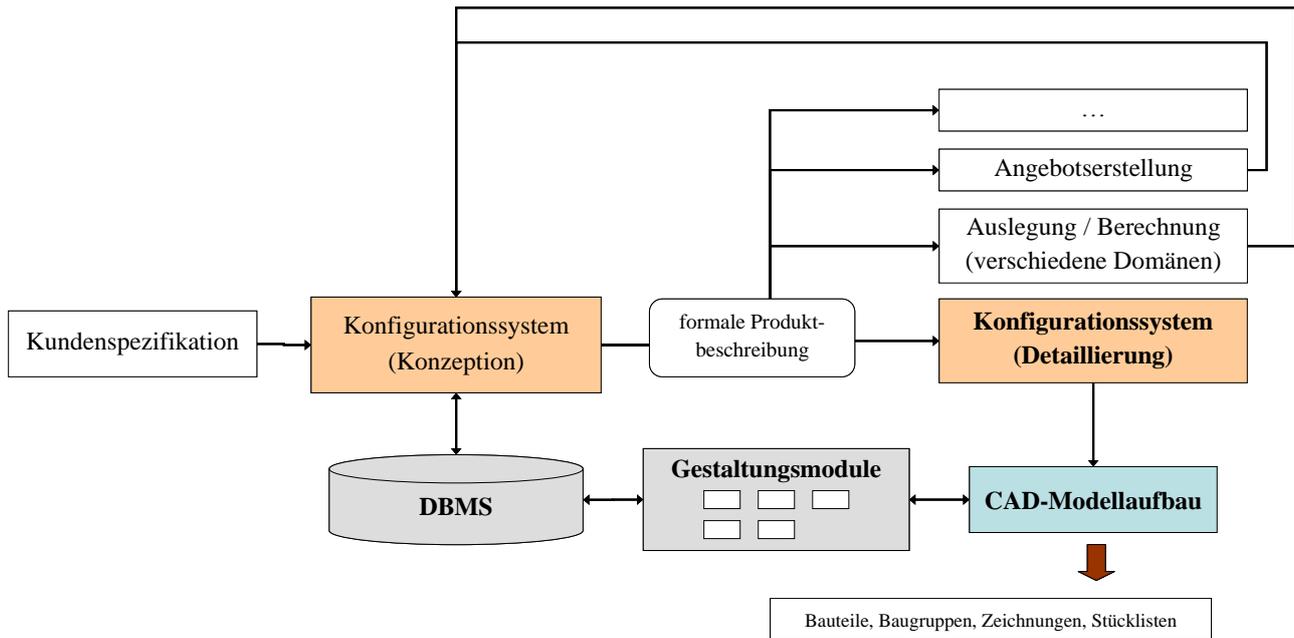


Abbildung 4-29: Einordnung des Konfigurationssystems in den Gesamtprozess

Für den sinnvollen Einsatz eines solchen Konfigurationssystems im Rahmen der Anpassungs- und Variantenkonstruktion sollten der Produktentwicklungsprozess bestimmte Merkmale und Rahmenbedingungen aufweisen:

- Variable Topologie des Produkts auf Bauteilebene
Die Notwendigkeit eines Konfigurationssystems ergibt sich aus A-priori unbekannter Komponententopologie. Diese kann mit den üblichen Methoden der parametrischen Konstruktion nicht automatisiert erzeugt werden.
- Aufbau des Produkts aus Gestaltungsmodulen
Die zu konfigurierenden Bauteile müssen im Wesentlichen durch die Rekombination vordefinierter Gestaltungsmodule entstehen. Diese Gestaltungsmodule müssen entsprechend standardisiert und dokumentiert sein. An dieser Stelle wird auf die in Kapitel 3.5 vorgestellte KCM-Methode verwiesen, mittels derer die hier geforderte Komponentenmodularisierung in Hinblick auf die Wiederverwendbarkeit und unter Berücksichtigung der Schnittstellen der einzelnen Teillösungselemente unterstützt wird.

- **Beschreibung der Gestaltungsmodule durch charakteristische Größen**
Die Anzahl der Größen, mit denen die Ausprägung eines Gestaltungsmoduls eindeutig bestimmt werden kann, sollte wesentlich kleiner sein als die Anzahl der Größen, die für die endgültige Gestaltbeschreibung notwendig sind. Die gestaltbeschreibenden Größen müssen durch Regeln oder Algorithmen aus den charakteristischen Größen ermittelbar sein.
- **Zentrale Speicherung der den Gestaltungsmodulen zugrundeliegenden Daten in einem DBMS**
Das DBMS speichert und verwaltet redundanzfrei alle Daten der Gestaltungsmodule. Dazu gehören sowohl Daten, die für den Aufbau des CAD-Modells benötigt werden als auch Daten für weitere Anwendungen. Der Zugriff muss über standardisierte Schnittstellen ermöglicht werden.
- **Unabhängigkeit der Gestaltungsmodule**
Jedes Gestaltungsmodul muss in der Lage sein, anhand der charakteristischen Größen selbstständig agieren zu können. Die Konnektivität zu weiteren Gestaltungsmodulen wird über fest vereinbarte Schnittstellen (Konnektoren) sichergestellt.
- **Formale Produktbeschreibung**
Als Eingangsgröße für die automatisierte Erzeugung der Feingestalt in Form eines CAD-Modells muss eine Beschreibung des Produkts vorliegen. Dazu gehört die Beschreibung der Komponentenstruktur durch die charakteristischen Größen der Gestaltungsmodule als auch die Beschreibung der Struktur der Baugruppen, die mit diesen Komponenten erzeugt werden soll. Diese Produktbeschreibung kann z. B. das Ergebnis eines Produktkonfigurationsprozesses einer eigenständigen Anwendung (im Rahmen der Angebotserstellung auf Kundenspezifikationsbasis) sein.

4.7.2 Anforderungen an die Produkt- und Prozessstruktur

Das grundlegende Ziel beim Einsatz der hier beschriebenen Ansätze ist die automatisierte Erzeugung der Produktfeingestalt anhand einer formalen Produktbeschreibung, die im Wesentlichen die Grobgestalt und die Struktur des Produkts anhand möglichst weniger Bezugsgrößen beschreibt. Ausgangsbasis hierfür sind eine entsprechende Produktmodularisierung und Produktstrukturierung, die auf der Basis einer umfangreichen Produktanalyse entwickelt werden müssen. Hierbei ist zwischen Bauteilen und Baugruppen zu unterscheiden. Baugruppen sind grundsätzlich hierarchisch organisiert. Durch die Verwendung des Top-Down Ansatzes ist die Produktstruktur im hier vorliegenden Fall der Anpassungs- und Variantenkonstruktion zumeist klar definiert. Ein wesentlich wichtigerer Aspekt ist die Analyse der Bauteilstrukturen und darauf aufbauend die Erzeugung eines Design Repositories mit einer bestimmten Anzahl von Gestaltungsmodulen. Die Gliederung eines Bauteils muss hierbei sowohl aus funktionaler Sicht als auch aus geometrischer Sicht erfolgen.

Die Dekomposition der Gesamtstruktur erfolgt vorrangig auf der Basis funktionaler Aspekte. Die Schnittstellen zwischen den einzelnen Gestaltungsmodulen sind jedoch geometrischer Natur und müssen entsprechende Beachtung finden, insbesondere in elementübergreifenden Bereichen. Hier ist eine eindeutige Zuordnung der elementübergreifenden Geometrie zu einem Hauptgestaltungsmodul anzustreben (siehe Abschnitt 4.4.4.2). Für die Verbindung der einzelnen Gestaltungsmodule untereinander müssen dann die Konnektoren ermittelt werden. Für die Position eines Konnektors innerhalb eines Gestaltungsmoduls sind entsprechende Regeln aufzustellen und zu dokumentieren. Die Zerlegung der Bauteilstruktur in für die Automatisierung geeignete Gestaltungselemente ist ein komplexer Prozess, der durch die in dieser Arbeit vorgestellten Anforderungen und Randbedingungen nur unterstützt werden kann. Jedoch finden sich ähnliche, vordefinierte Strukturen auch bei anderen Problemstellungen des Maschinenbaus. Hier ist insbesondere die Getriebeentwicklung hervorzuheben, die bereits in einigen Arbeiten Gegenstand ähnlicher Untersuchungen war [Haa95] [Jar07].

Das hier vorgestellte Konfigurationssystem hat nun die Aufgabe, aus der Grobgestalt mit Hilfe autarker Gestaltungsmodule die detaillierte Feingestalt der einzelnen Produktkomponenten und die Baugruppenstruktur zu erzeugen. Dazu muss das System die Produktbeschreibung hinsichtlich der komponentenspezifischen Vorgaben analysieren und zusätzliche Eingaben des Anwenders verarbeiten. Daraufhin werden die notwendigen Gestaltungsmodule der Komponenten aus dem Design Repository ermittelt, in die Bauteile eingebaut und mit den notwendigen Bezugsgrößen versehen. Abschließend werden die Gestaltungsmodule über Konnektoren miteinander verknüpft und zu einem Körper verbunden.

4.7.3 Anforderungen an das IT-Umfeld

Die Funktionsvielfalt des CAD-Systems spielt eine entscheidende Rolle für die Umsetzbarkeit der vielfältigen Anforderungen an die Möglichkeiten der Produktmodellierung. Der Trend bei aktuellen Entwicklungen der großen Systemanbieter weist darauf hin, dass die durch die Systeme gebotenen Möglichkeiten der erweiterten Produktmodellierung stetig ergänzt werden. Durch die Konkurrenzsituation der Anbieter im Markt ist die Dynamik der Systemerweiterungen sehr hoch, so dass im Rahmen dieser Arbeit keine Bewertung einzelner Systeme hinsichtlich des Erfüllungsgrades einzelner Anforderungen erfolgen soll.

Eine wesentliche Forderung an das CAD-System ist die Möglichkeit, Wissen direkt im Produktmodell integrieren und auf geeignete Weise wiederverwenden zu können. Das System muss daher verschiedene Wissensrepräsentationsformen implementieren, die in Gestaltungsmodulen (UDFs oder andere systemspezifische Formen der Wiederverwendung) gespeichert und zugänglich gemacht werden können. Neben den grundsätzlichen Formen der Wissensrepräsentation (Parameter, Formeln, Beziehungen, Tabellen) muss das System erweiterte Formen der Wissensrepräsentation in Form von Regeln und algorithmischen Aktivitäten auf der Basis unterschiedlicher Datentypen unterstützen, um auch komplexe Wissensinhalte abbilden zu können. Neben der reinen Speicherung

und Verwaltung des Wissens im CAD-Modell ist damit die Möglichkeit der Geometriemanipulation auf der Basis wissensbasierter Feature eine wichtige Voraussetzung, um eine funktionsgetriebene Verhaltensweise der Gestaltungsmodule zu ermöglichen. Die durch die Art der Geometrieprepräsentation (CSG, B-Rep oder Hybrid) entstehenden Nachteile in Hinblick auf Modellvariabilität müssen durch die Möglichkeit der direkten Manipulation der Featuredefinition ausgeglichen werden können.

Die Nutzung der erweiterten Möglichkeiten der Geometriemanipulation führt damit zu einer Unterteilung des Gestaltungsmoduls in einen Geometrieteil und einen wissensbasierten Teil, die miteinander in Wechselwirkung stehen (Abbildung 4-30).

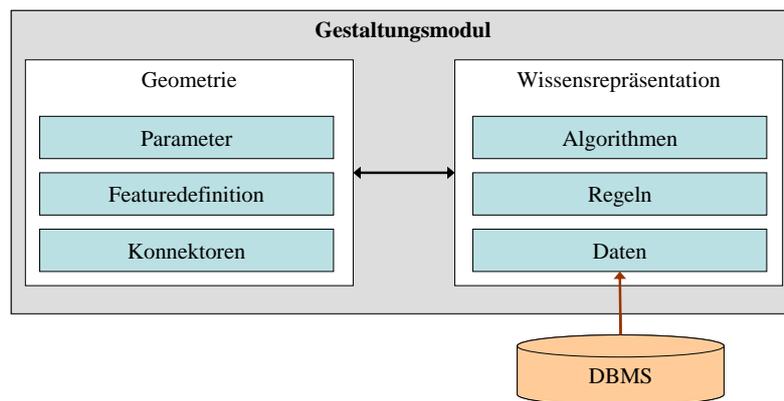


Abbildung 4-30: Trennung von Wissen, Geometrie und Daten in einem Gestaltungsmodul

Das Verhalten des Gestaltungsmoduls wird durch die Definition charakteristischer Größen (Bezugsgrößen) bestimmt. Das in Algorithmen und Regeln kodierte Wissen ermittelt daher zuerst alle notwendigen Daten, die für die Steuerung der Parameter und Features des Geometriemodells notwendig sind. Hierzu muss das CAD-System die Möglichkeit bieten, innerhalb eines Gestaltungsmoduls eine Datenbankverbindung zu implementieren. Ist diese Voraussetzung nicht erfüllt, so ist zu prüfen, ob eine Erweiterung der Systemfunktionalität auf der Basis einer möglichst systemnahen Schnittstelle erfolgen kann. Moderne und leistungsfähige CAD-Systeme bieten die Möglichkeit, zusätzliche Objekte und Features durch API-Programmierung zu erzeugen. Hierbei hängt es von der Lizenzpolitik des CAD-System Herstellers und der systembedingten Offenheit des CAD-System Kerns ab, welche Integrationstiefe durch eine nachträgliche Systemerweiterung möglich ist.

4.7.4 Bemerkungen zur Übertragbarkeit von Teilaspekten

Aufgrund der Komplexität und des Umfangs des Gesamtkonzepts war es im Rahmen dieser Arbeit nicht möglich, die Übertragbarkeit aller Ansätze auf weitere Problemstellungen zu betrachten. Wesentliche Teilaspekte des Konzepts, insbesondere die Modellierungsmethoden für die Gestaltung funktionaler Gestaltungsmodule, wurden jedoch auch für die erfolgreiche Lösung anders gearteter Problemstellungen eingesetzt.

Im folgenden Abschnitt soll zuerst die Aufgabenstellung, die es im Rahmen eines durch die DFG geförderten Forschungsprojekts⁶ zu lösen gilt, kurz erläutert werden. Hierbei wird auf eine detaillierte Darstellung aller Details des Forschungsprojekts zugunsten der Übersichtlichkeit verzichtet. Der Schwerpunkt der Ausführungen liegt daher auf den Aspekten, die für die Anwendbarkeit der in dieser Arbeit dargestellten Methoden von Bedeutung sind. Ausführliche Erläuterungen zur Gesamtsituation finden sich in [KoWo08] und [KoWo09]. Daran anschließend werden die sich ergebenden Probleme und die mit den hier dargestellten Ansätzen erarbeiteten Lösungen erläutert.

4.7.4.1 Aufgabenstellung

Bei der Extrusion thermoplastischer Halbzeuge und Fertigprodukte nehmen das Werkzeug sowie die Fließkanalgeometrie eine zentrale Stellung ein. Das Geschwindigkeitsprofil am Werkzeugaustritt definiert die Dicken- / Wandstärkenverteilung des Extrudates und die Spannungen, welche hier eingebracht werden. Ebenso spielen Verweilzeitverhalten, thermische Verhältnisse und Druckverlust beim Durchströmen der Fließkanäle einer Kunststoffmaschine eine wichtige Rolle in Hinblick auf Produktqualität und Umweltfreundlichkeit des Prozesses. Dies beeinträchtigt nachhaltig die Qualität des Endprodukts. Hersteller von Extrusionsanlagen sowie Anlagenbetreiber sind deshalb von je her daran interessiert, die Druck- und Geschwindigkeitsverteilung im Werkzeug und zum Teil auch nach dem Werkzeugaustritt zu quantifizieren und vorbestimmen zu können. Als Beispiel sei hier für die komplexen Fließvorgänge in T-Verteilern sowie in 90°-Umlenkungen genannt.

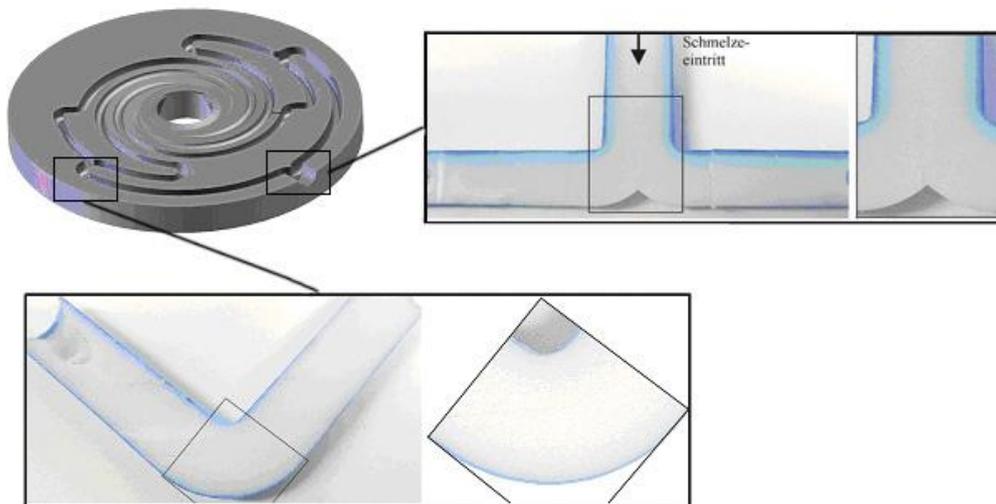


Abbildung 4-31: Spülverhalten in Umlenkungen [Nie06]

Hierbei wurde die Wandschergeschwindigkeit durch NIEMEIER als wesentliche Einflussgröße für die Beeinflussung der Spülbarkeit ermittelt [Nie06]. Er entwickelte ein automatisches Optimie-

⁶ Gemeinschaftsprojekt mit dem Lehrstuhl für Konstruktion und Kunststoffmaschinen der Universität Duisburg-Essen,, Titel: „Integrative Kopplung von Gestaltung, Berechnung und Simulation für die Auslegung von Fließkanälen in der Kunststoffverarbeitung“, Projekt-Nr.: WO-302-34-1 und KO1620-8-1

rungsverfahren für die Variation der Kanalgeometrie auf der Basis eines diskreten numerischen Modells. Die Verschiebung der Knotenpunkte basiert bei diesem Verfahren auf der Analyse der Schergeschwindigkeiten an einzelnen FE-Knotenpunkten entlang der Kanalwände. Alle Wandknoten, deren Schergeschwindigkeit mehr als einen vordefinierten Betrag von der Zielschergeschwindigkeit abweicht, werden in Abhängigkeit von der Schergeschwindigkeitsdifferenz um eine berechnete Strecke entlang eines vorbestimmten Vektors verschoben. Basierend auf diesen Berechnungen wird automatisch ein neues FEM-Netz erzeugt und die Simulation startet erneut. Dieser Zyklus wird so lange wiederholt, bis die Summe aller Veränderungen der Schergeschwindigkeit in den einzelnen Wandknoten in Relation zur letzten Iteration einen definierten Betrag unterschreitet. Ergebnis dieses Prozesses ist eine diskrete Geometriebeschreibung, die im Anschluss in ein fertigungsgerecht gestaltetes CAD-Modell überführt werden muss (Reverse Engineering).

Diese Prozesslücke soll durch ein integriertes Optimierungsverfahren geschlossen werden, das auf einem CAD-Modell der Fließkanalgeometrie beruht. Dieses CAD-Modell soll sowohl die fertigungstechnischen Randbedingungen abbilden als auch als Master-Modell für nachgeschaltete Prozessschritte sein.

4.7.4.2 Problemstellung und Übertragbarkeit von Teilaspekten des Gesamtkonzepts

Für die Optimierungsaufgabe wurde ein Ablauf konzipiert, in dem das CAD-Modell im Mittelpunkt steht (Abbildung 4-32). Durch die softwareseitige Integration der CFD-Software Fluent der Firma ANSYS in das CAD-System CATIA V5 von DASSAULT SYSTEMES ist es möglich, ein zur CAD-Geometrie vollständig assoziatives Berechnungsmodell zu erstellen. Mit Hilfe der Software Optimus des Herstellers NOESIS wurde ein Optimierungsablauf erstellt, der die Reihenfolge der Prozessschritte steuert.

Die Optimierung beginnt mit der Definition der modifizierbaren Parameter des CAD-Modells in Optimus. Diese Parameter werden nun an ein selbstentwickeltes Steuerprogramm übergeben. Dieses Programm ruft nun das CAD-Modell des Fließkanals auf, modifiziert die Parameterwerte und veranlasst das CAD-System dazu, eine aktualisierte Eingabedatei für die anschließende CFD-Berechnung zu erstellen. Diese Datei wird nun in den Solver von Fluent eingeladen und daraufhin die Berechnung gestartet. Der Solver extrahiert nun die für die Optimierung interessanten Zielgrößen in eine Ausgabedatei. Diese Zielgrößen werden von Optimus ausgewertet, wobei hier verschiedene Optimierungsstrategien zur Verfügung stehen. Daraufhin wird ein neuer Parametersatz für das CAD-Modell generiert und der Ablauf wiederholt, bis ein Abbruchkriterium (Optimiertes Ergebnis oder maximale Anzahl Iterationen) erreicht wird

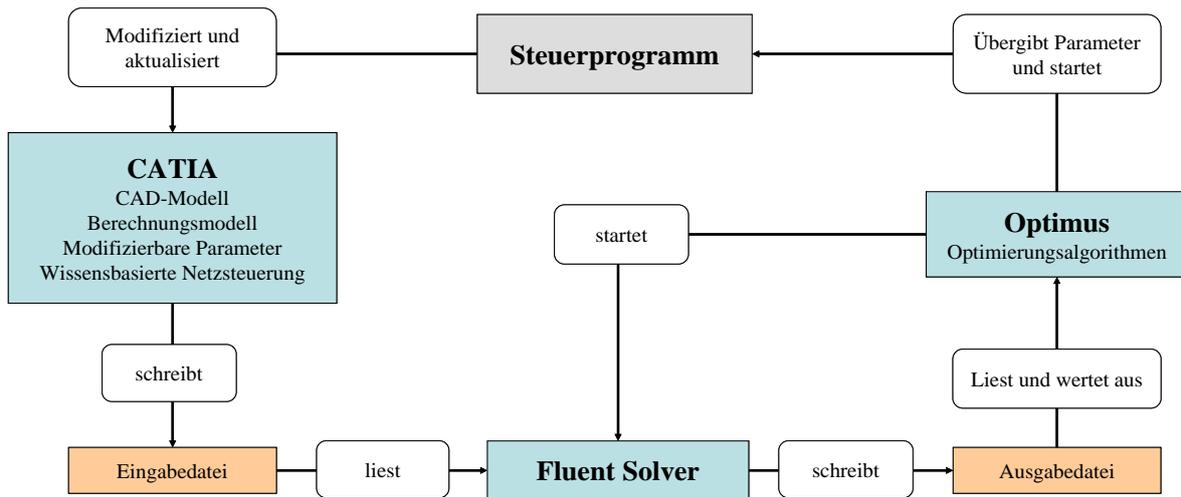


Abbildung 4-32: Optimierungsstrategie auf Basis des CAD-Modells

Das CAD-Modell hat für die Optimierungsaufgabe eine besondere Bedeutung, da es in besonderem Maße die Stabilität des gesamten Prozesses beeinflusst. Das Modell muss vor allem in der Lage sein, verschiedene Topologien abbilden zu können, um eine möglichst große geometrische Vielfalt zu erlauben. Auf der linken Seite der Abbildung 4-33 ist eine Hälfte eines Fließkanalmoduls dargestellt. Die Geometrie dieses Fließkanals entsteht durch die Verbindung mehrerer Querschnitte mit einer Fläche entlang ein oder mehrerer Leitkurven. Die Abmaße dieser Features werden durch Parameter beschrieben.

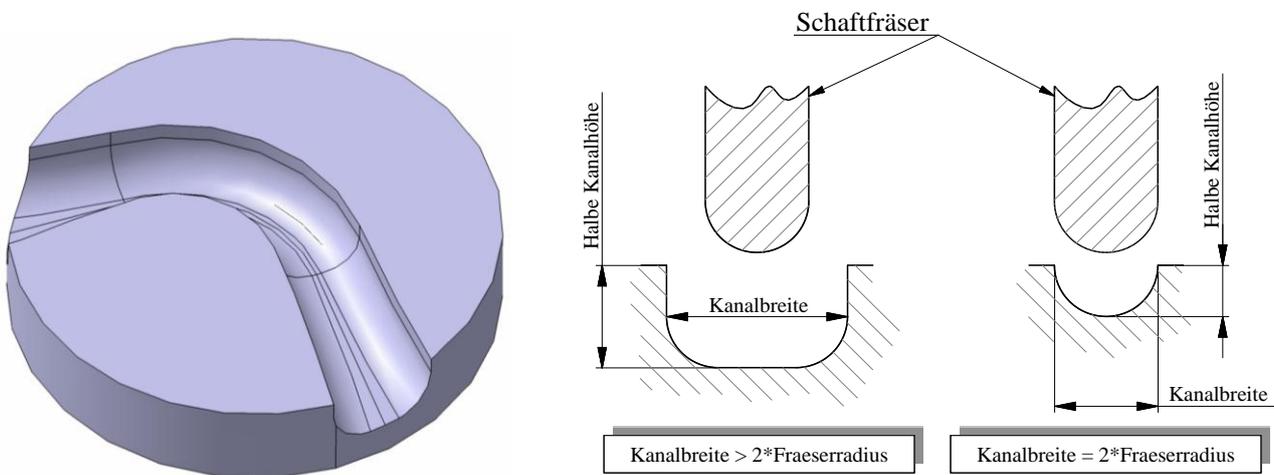


Abbildung 4-33: Einfluss der Fertigungsparameter auf die Kanaltopologie

Die Fertigung derartiger Strukturen kann auf vielfältige Art und Weise erfolgen, wobei hier das Fräsen der Kanalgeometrie betrachtet werden soll. Geht man davon aus, dass die Geometrie mit einem abgerundeten Schafffräser erstellt werden soll, so ist der Fräserradius eine geometriebestimmende Größe, ebenso die Kanalbreite im jeweiligen Querschnitt (rechte Seite der Abbildung 4-33). Schon dieses einfache Beispiel zeigt die vielfältige topologische Variabilität auf, die durch das geometrische Modell abgebildet werden muss. Wird die Kanalbreite so weit reduziert, dass diese dem zweifachen Fräserradius entspricht, so entfällt der gerade Steg am unteren Teil des Kanals. Um eine

automatische Optimierung erfolgreich durchführen zu können muss das Modell diese topologische Änderung erkennen, analysieren und geeignete Gegenmaßnahmen ergreifen können, um eine regenerierbare Geometrie bereitzustellen, die im weiteren Verlauf der Optimierung nutzbar ist. Dazu können die in dieser Arbeit entwickelten Methoden genutzt werden. Das hier genutzte CAD-System CATIA V5 unterstützt die Erweiterung des Geometriemodells um wissensbasierte Komponenten durch verschiedene Knowledgware-Module. Es besteht u. a. die Möglichkeit, Regeln und Algorithmen in das Produktmodell zu integrieren und auf bestimmte Ereignisse (z. B. die Modellregenerierung) reagieren zu lassen.

Das Geometriemodell des Fließkanals wird auch hier aus einzelnen Gestaltungsmodulen zusammengesetzt. Die Geometrie wird dabei hauptsächlich durch die Definition der Eintritts- und Austrittsquerschnitte und der dazwischen liegenden Verbindung (Leitkurve) bestimmt. Innerhalb dieser Begrenzung können verschiedene Kanalabschnitte mit vom Anwender gewählten Eigenschaften (z. B. unterschiedliche Querschnittsformen oder Stetigkeitsanforderungen der Flächenbegrenzung) eingefügt werden. Der Anwender gibt bei der Instanzierung der Gestaltungselemente nur die wesentlichen Größen vor (z. B. Endquerschnitt und geometrische Hauptabmessungen). Entscheidend für die endgültige geometrische Ausprägung sind die im Gestaltungsmodul eingebauten wissensbasierten Komponenten.

Der Aufbau eines Gestaltungsmoduls lässt sich auch hier in einen geometrischen und einen wissensbasierten Teil unterteilen, wobei diese beiden Anteile miteinander agieren. Das System CATIA V5 bietet die Möglichkeit, die Elemente des Modellbaums in Form einer Ordnerstruktur zu organisieren. Damit sind die Voraussetzungen für eine strukturierte Modellerstellung vorhanden. Vor der eigentlichen Modellerstellung müssen allerdings die Randbedingungen geklärt werden, innerhalb derer die Geometrie des Gestaltungsmoduls regenerierbar sein soll. Des Weiteren müssen die Einflüsse der zur Verfügung stehenden Fertigungsverfahren auf die Kanalgeometrie ermittelt werden. Ergebnis dieser Analyse, die jeweils problembezogen erfolgen muss, ist eine Ansammlung von Gestaltungsregeln, die in einem Katalog dokumentiert werden können. Auf der Basis dieser Regeln wird dann das geometrische Modell des Gestaltungsmoduls erstellt. Auch hier bilden die Querschnittsskizzen die Grundlage für die weitere Modellgeometrie. Dabei müssen die verschiedenen Modelltopologien durch die Skizzen erfasst werden. Im wissensbasierten Teil des Produktmodells werden daraufhin die Gestaltungsregeln implementiert. Diese entscheiden anhand der vorliegenden Parameterkombination oder der Ergebnisse von Modelluntersuchungen, welche finale Ausprägung der Geometrie gewählt wird. Ein Beispiel für eine unterstützende Modelluntersuchung ist die Suche nach den extremsten Querschnittsflächen des Gestaltungsmoduls. Dazu wurde ein Makroelement in das Produktmodell integriert, das die Modellgeometrie analysiert. Grundlage der Untersuchung ist ein parametrischer Punkt auf der Leitkurve des Fließkanals. Der Laufparameter des Punktes kann zwischen 0 und 1 variiert werden. Durch diesen Punkt und normal zur Leitkurve wird eine Schnittebene definiert, mit der eine Schnittkurve erzeugt wird. Diese Schnittkurve dient als Grundlage für verschiedene Messungen, die ihren aktuellen Wert als Parameter bereitstellen. Wird

das Modell mit neuen Parameterwerten belegt, so führt ein Reaktionselement das Makro aus. Dieses belegt den Laufparameter des Punktes mit einem Wert, der anhand einer vordefinierten Schrittweite ermittelt wird. Daraufhin werden die aktuellen Messwerte auf Basis der regenerierten Geometrie der Schnittkurve gespeichert. Diese Aktionen werden so lange ausgeführt, bis der Laufparameter den Wert 1 erreicht hat. Anhand des zwischengespeicherten Messwertfeldes werden die gewünschten Extremwerte auf vordefinierte Modellparameter geschrieben und stehen damit sowohl dem Geometriemodell als auch dem Berechnungsmodell zur Verfügung. Auch die im Modell enthaltenen Regeln können durch die Parameterwerte für die Entscheidungsfindung nutzen. Das so aufgebaute Produktmodell ist so in der Lage, ein weites Parameterspektrum und damit auch eine große topologische Vielfalt abzubilden. Des Weiteren kann durch die Regeln auch sichergestellt werden, dass im Fall einer nicht zu realisierenden Parameterkombination eine Grundkonfiguration der Geometrie eingestellt wird, so dass zumindest ein regenerierbares Modell erzeugt wird und damit der Optimierungsablauf nicht gefährdet ist. Diese Grundkonfiguration kann als Boolescher Parameterwert gespeichert werden, um so den aktuellen Modellzustand zu dokumentieren.

Neben den möglichst selbstständig agierenden Gestaltungsmodulen, welche die einzelnen Fließkanalelemente repräsentieren, existiert eine übergeordnete wissensbasierte Komponente im Fließkanalmodell. Diese Komponente überwacht und steuert die Verknüpfung zwischen den einzelnen Gestaltungsmodulen. Des Weiteren enthält diese Komponente Algorithmen, mit denen das endgültige Fließkanalmodell für die anschließende Berechnung vorbereitet wird. Ein wesentlicher Einflussfaktor für eine konvergente und numerisch genaue Berechnung des Kunststofffließverhaltens ist die Qualität der Bauteildiskretisierung. Hierbei spielen sowohl der gewählte Elementtyp wie auch die Auflösung (Elementgröße) eine wichtige Rolle. Ein global zu feines Netz erhöht die Berechnungszeit sehr stark und kann die vorhandenen Rechnerkapazitäten überschreiten. In Bereichen hoher Gradienten ist jedoch eine feine Auflösung unbedingt anzustreben, um belastbare Ergebnisse zu erhalten. Dies betrifft insbesondere die Randschichten der Kunststoffschmelze aufgrund der Wandhaftungsbedingung. Um eine sowohl stabile (in Hinblick auf die Automatisierung) als auch qualitativ hochwertige Vernetzung zu erhalten vergibt die übergeordnete wissensbasierte Komponente geometrische Zwangsbedingungen an das Modell, die vom Vernetzungsalgorithmus für die Vergabe von Knotenpunkten verwendet werden. Dazu analysiert die Komponente z. B. die in den jeweiligen Gestaltungsmodulen vorhandenen Parameterwerte für die Extrema der Querschnitte und errechnet so einen Versatzwert für die Höhe der Randschicht. Anhand größenunabhängiger Parameter ist der Anwender in der Lage, die Vernetzung qualitativ zu beeinflussen. So kann er z. B. die gewünschte Anzahl der Elemente im Querschnitt oder die Anzahl der Knoten auf der Querschnittkurve bestimmen.

Die automatisierte parameterbasierte Optimierung von Fließkanalstrukturen auf der Basis eines CAD-Modells wäre ohne die Integration von wissensbasierten Komponenten zur Unterstützung und Erweiterung des Geometriemodells in dieser Varianz nicht möglich gewesen. Erst durch wissensbasierte Komponenten ist es möglich, sowohl die fertigungstechnischen Restriktionen als auch die für

eine Berechnung notwendigen Automatismen in das Produktmodell zu integrieren, um so einen stabilen automatischen Optimierungsablauf zu erhalten. Auch in diesem Fall ist eine geordnete und strukturierte Modellierung des CAD-Modells eine Grundvoraussetzung für die anschließende Integration der wissensbasierten Komponenten und für die zukünftige Erweiterbarkeit des Produktmodells um weitere Varianten und Ausprägungen.

5 Zusammenfassung und Ausblick

Der durch die Globalisierung der Märkte zunehmende Kostendruck zwingt die Unternehmen, ihre Produktentwicklungsprozesse immer effizienter zu gestalten. Trotz des hohen Kostendrucks werden seitens der Kunden qualitativ immer hochwertigere Produkte erwartet. Die Grundvoraussetzungen für die Erfüllung dieser Erwartungen werden bereits in den frühen Phasen des Produktentwicklungsprozesses geschaffen. Das reine Vorhandensein softwaretechnischer Lösungen sowohl für die Datenerzeugung (CAD, CAE, FEA, etc.) als auch für die Datenverwaltung (PDM) ist jedoch nicht ausreichend, um den komplexen Prozess der VPE in geeignetem Maße zu unterstützen. Zum Einen muss der zielgerichtete Einsatz der vorhandenen Werkzeuge durch entsprechende Methoden im Unternehmen unterstützt werden. Das können z. B. Prozessmodelle (Concurrent und Simultaneous Engineering) sein, welche das gleichzeitige Arbeiten an verteilten Standorten und über Unternehmensgrenzen hinweg unterstützen. Zum Anderen müssen aber auch die bestehenden Potentiale der Werkzeuge vollständig ausgeschöpft werden, was jedoch häufig nicht der Fall ist.

Moderne CAD-Systeme bieten mittlerweile eine über die reine Geometriemodellierung hinausgehende Funktionsvielfalt und haben sich zu Produktentwicklungswerkzeugen entwickelt. Diese Ansicht wird durch Bestrebungen der Softwarehersteller bestärkt, Anwendungen aus allen Bereichen der VPE in ihre CAD-Systeme zu integrieren und so von den Vorteilen einer einheitlichen (jedoch proprietären) Datenbasis zu profitieren. Vor allem die wissensbasierte Konstruktion hat in den letzten Jahren zunehmend an Bedeutung gewonnen, insbesondere durch die zunehmende Integration wissensbasierter Komponenten in die CAD-Systeme. Dadurch wird es möglich, das für eine qualitativ hochwertige Produktentwicklung notwendige Wissen in dem Bereich zu verarbeiten und nutzbar zu machen, in dem es entsteht und auch benötigt wird.

Ziel dieser Arbeit war es daher, anhand einer konkreten Problemstellung die möglichen Potentiale dieser Technologien für die automatisierte Erstellung von Komponenten und Baugruppen im Rahmen der Anpassungs- und Variantenkonstruktion aufzuzeigen. Die hierzu ausgewählten Komponenten eines Verdichterläufers weisen einen auf Bauteilebene modularen Aufbau auf. Dieser folgt weniger geometrischen als funktionalen Gesichtspunkten. Das mit diesen Komponenten und ihren Gestaltungsmodulen verbundene Wissen ist in Form von umfangreichen Konstruktionsstandards dokumentiert und die damit verbundenen Daten sind durch die Speicherung und Verwaltung in einem DBMS unternehmensweit zugänglich. Damit stehen sie auch dem für die Produktentwicklung genutzten Konfigurationssystem zur Verfügung. Dieses spezialisierte System wird zu Beginn des Produktentwicklungsprozesses eingesetzt und dient der Erzeugung der Produktstruktur unter Berücksichtigung funktionaler und technischer Aspekte. Das Ergebnis des Konfigurationsprozesses ist die formale Beschreibung der Verdichterkomponenten (Grobgestalt), die als Eingangsgröße für die automatisierte Erzeugung des Geometriemodells im CAD-System dienen soll (Feingestalt).

Eine Analyse der Ist-Situation zeigte, dass ein automatisierter Modellaufbau innerhalb des CAD-Systems in diesem Umfang nur durch eine selbstentwickelte Anwendung realisierbar ist. Im Rah-

men des hier entwickelten Konzepts erfolgt daher der Aufbau der Verdichterkomponenten durch den Einbau und die Verknüpfung wissensbasierter Features. Jedes Feature repräsentiert ein spezifisches Gestaltungsmodul und integriert sämtliches Wissen, das für die Gestalterzeugung und die Anknüpfung weiterer Prozessschritte (CAM, Arbeitsvorbereitung, etc.) notwendig ist. Dazu implementiert jedes Feature ein oder mehrere Datenbankverbindungen und –abfragen über eine ODBC-Schnittstelle. Anhand der gewählten Eingangsgrößen werden die neuen Daten automatisch ermittelt und die Gestalt des Features angepasst. Dadurch können die Daten weiterhin redundanzfrei im DBMS verwaltet werden und müssen nicht in die CAD-Modelle integriert werden. Um eine möglichst große geometrische Variabilität und vor allem Stabilität zu ermöglichen wird eine Modellierungsmethodik auf der Basis wissensbasierter Erweiterungen eingesetzt. Durch die gezielte Verwendung dieser Komponenten ist es möglich, die üblichen Grenzen der parametrischen und featurebasierten Modellierung zu erweitern. Das Feature wird dadurch zu einem funktionalen Element, dessen geometrische Gestalt nur das Ergebnis funktionaler Vorgaben ist.

Das für den Einbau und die Verknüpfung der Gestaltungselemente zuständige Konfigurationssystem wurde grundsätzlich produktneutral konzipiert. Dadurch wird kein komponentenspezifisches Wissen im Quellcode des Konfigurationssystems gespeichert. Durch die Vereinbarung einer standardisierten hierarchischen Struktur ist die Erweiterung der Anwendung um komponentenspezifische Plug-Ins möglich. Diese Plug-Ins dienen als Schnittstelle zum Anwender und bilden die komponentenspezifischen Beschreibungsdaten auf die systeminterne Beschreibungsform ab. Durch dieses Vorgehen wird die einfache Erweiterbarkeit des Konfigurationssystems um neue Komponenten unterstützt und die Menge an komponentenspezifischem Wissen, das in der Anwendung gespeichert wird, minimiert.

Zum Abschluss wurde die Übertragbarkeit von Teilaspekten der Methodik für die automatisierte strömungstechnische Optimierung von Fließkanalgeometrien aus dem Bereich der Kunststofftechnologie dargestellt. Am Beispiel der Kopplung von CAD-Modell und Berechnungsmodell mit zwischengeschaltetem Optimierungsalgorithmus wurde die Notwendigkeit für die Integration wissensbasierter Komponenten in das CAD-Modell aufgezeigt. Neben der Stabilisierung der CAD-Geometrie bei Topologieänderungen aufgrund von Parametervariationen dienen die wissensbasierten Anteile hier vor allem der Vorbereitung und Analyse des CAD-Modells in Hinblick auf die nachfolgende Strömungsberechnung, um so eine sowohl stabile als auch qualitativ hochwertige Vernetzung für die automatisierte Berechnung zu erhalten.

Der zielgerichtete Einsatz wissensbasierter Komponenten in der CAD-Modellerstellung bietet weitreichende Möglichkeiten, um qualitativ hochwertigere und dynamischere Modelle zu erstellen und das mit diesen Modellen verknüpfte Wissen zu speichern und wieder verwertbar zu machen. Das große Potential integrierter KBE-Systeme für die VPE weist jedoch auch direkt auf die möglichen Nachteile hin. Wie schon bei der Einführung parametrischer und featurebasierter Systeme sollten aufgrund der vielfältigen Möglichkeiten der Modellierung Methoden entwickelt werden, mit denen die mögliche Komplexität der Modelle beherrscht werden kann um so die Nachvollziehbar-

keit des Modellaufbaus sicherzustellen und vor allem das in den Modellen gespeicherte Wissen extrahieren und dokumentieren zu können. Aufgrund der wachsenden Leistungsfähigkeit der CAD-Systeme sollte diesem Umstand besondere Bedeutung beigemessen werden, um so auch bei bevorstehenden Systemwechseln die Gefahr des Wissensverlusts zu minimieren. Wie auch bei parametrischen und featurebasierten Modellen ist jedoch derzeit keine Schnittstelle vorhanden, die eine neutrale Datenübertragung in ein anderes CAD-System sicherstellt.

6 Literaturverzeichnis

- [AbBe02] Abdullah, M. S.; Benest, I.; Evans, A.; Kimble, C.: *Knowledge Modelling - Techniques For Developing Knowledge Management Systems*, 3rd European Conference on Knowledge Management, Dublin, Irland, September 2002
- [AbMe07] Abramovici, M.; Meimann, V.: *Ein Ansatz zur Erhöhung der Stabilität der CAD-Modelle für die Verbesserung der Modell-Wiederverwendung in der Virtuellen Produktentwicklung*. Tagungsband zum 5. Kolloquium Konstruktionstechnik 09/2007, S. 111-118, Dresden
- [Abr05] Abramovici, M.: *Rechnerintegrierte Produktentwicklung*. Skript zur Vorlesung, Lehrstuhl für Maschinenbauinformatik, Ruhruniversität Bochum, 2005
- [AbSc04] Abramovici, M.; Schlingensiepen, J.: *E-Services als Wettbewerbsvorteil: Ingenieurdienstleistungen aus der Steckdose*. RUBIN, Sonderheft MaschinenbauRUBIN, Konstruktionstechnik - Beitrag 8, Ruhr-Universität Bochum, 2004
- [And06] Anderl, R.: *Produktdatentechnologie A – CAD-Systeme und CAx-Prozessketten*. Skript zur Vorlesung, Fachgebiet Datenverarbeitung in der Konstruktion, TU Darmstadt, 2006
- [BäHa01] Bär, Th.; Haasis, S.: *Verkürzung der Entwicklungszeiten durch den Einsatz von Skelettmodellen und der Feature-Technologie*. VDI-Berichte Nr. 1614, S. 143-155, VDI Verlag, Düsseldorf, 2001
- [Bau04] Baumann, Richard A.: *Structure-Oriented Exchange of Product Model Data*. Berichte aus dem Produktionstechnischen Zentrum Berlin, IRB Verlag, 2004
- [BäWe96] Bär T.; Weber C.: *Neues aus dem Bereich der Feature-Technologie – Ergebnisse der FEMEX Arbeitsgruppe I*. CAD-CAM Report, Nr. 9, 1996
- [Bec07] Bechthold, J.: *Experimentelle Absicherung virtueller Prototypen*. Dissertation, Cuvillier Verlag, Göttingen 2007
- [BeCa99] Beynon-Davies, P.; Carne, C.; Mackay, H.; Tudhope, D.: *Rapid application development (RAD): an empirical review*. European Journal of Information Systems, Band 8, S. 211–223, 1999
- [BeKe08] Beierle, C.; Kern-Isberner, G.: *Methoden wissensbasierter Systeme*. 4. Auflage, Vieweg Verlag, Wiesbaden, 2008
- [BlKn95] Blount, G. N.; Kneebone, S.: *Selection of knowledge-based engineering design applications*, Journal of Engineering Design, 1995, Band. 6, Heft 1, S. 31

- [Bod06] Bodendorf, F.: *Daten- und Wissensmanagement*. 2. Auflage, Springer Verlag, Berlin, Heidelberg, 2006
- [Bos07] Bossmann, M.: *Feature-basierte Produkt- und Prozessmodelle in der integrierten Produktentstehung*. Dissertation, Schriftenreihe Produktionstechnik, Band 38, Universität des Saarlandes, 2007
- [BrOl99] Brimble, R.; Oldham, K.; Callot, M.; Murton, A.: *MOKA - A Methodology for Developing KBE Applications*, Proceedings of the 8th European Conference on Product Data Technology, Stavanger, Norwegen, April 1999, S. 361-366
- [ChMu02] Choi, G.-H.; Mun, D.; Han, S.: *Exchange of CAD part models based on the macro-parametric approach*. International Journal of CAD/CAM Band 2, Heft 2, S. 13–21, 2002
- [ChPr07] Chapman, C.; Preston, S.; Pinfold, M.; Smith, G.: *Utilising enterprise knowledge with knowledge-based engineering*. International Journal of Computer Applications in Technology, Band. 28, Heft 2-3, S. 169-179, 2007
- [Cod70] E. F. Codd: *A relational model of data for large shared data banks*. Communications of the ACM, Association for Computing Machinery, S. 377–387, 13.06.1970
- [DaKö06] Danjou, S.; Köhler, P.: *Herausforderungen an das Konstruktionsmanagement*. Zeitschrift Konstruktion, Springer-VDI-Verlag 10/2006.
- [Den04] Denkena, B.: *CAX-Anwendungen in der Produktion*. Skript zur Vorlesung, Institut für Fertigungstechnik und Werkzeugmaschinen, Leibniz Universität Hannover, 2004
- [Dil06] Dilger, W.: *Wissensrepräsentation und Problemlösung – Expertensysteme*. Skript zur Vorlesung, Professur Künstliche Intelligenz, TU Chemnitz, 2006
- [DIN84] Deutsches Institut für Normung e.V.: *DIN-Norm 66234, Teil 8 zu Bildschirmarbeitsplätzen*. Berlin, 1984.
- [Dud06] Duden. *Deutsches Universalwörterbuch*, 6. Auflage, Bibliographisches Institut, Mannheim, 2006
- [Dun08] Dungs, S.: *Wissensbasierte Geometriemodelle zur Strukturanalyse*. Dissertation, Universität Duisburg-Essen, Shaker Verlag, Aachen, 2008
- [Ehr02] Ehrlenspiel, K.: *Integrierte Produktentwicklung*. 3. Auflage, München, Carl Hanser Verlag, 2002
- [Era07] Eraßme, R.: *Der Mensch und die „Künstliche Intelligenz“*. Vdm Verlag Dr. Müller, Saarbrücken, 2007

- [FeFr99] Felfernig, A.; Friedrich, G.; Jannach, D.: *Intelligente Produktkonfiguratoren als Voraussetzung für maßgeschneiderte Massenprodukte*. In: Elektrotechnik und Informationstechnik 116, Bd. 3, S. 201-207, 1999
- [For03] Forsen, J.: *Ein systemtechnischer Ansatz zur methodisch parametrisch-assoziativen Konstruktion am Beispiel von Karosseriebauteilen*. Dissertation, Universität Duisburg-Essen, Shaker Verlag, Aachen 2003.
- [GaEb01] Gausemeier, J.; Ebbesmeyer, P.; Kallmeyer, F.: *Produktinnovation: Strategische Planung und Entwicklung der Produkte von morgen*. Carl Hanser Verlag, München, Wien, 2001
- [GaN06] Gasch, R.; Nordmann, R.; Pfützner, H.: *Rotordynamik*. 2. Auflage, Springer Verlag, Berlin Heidelberg, 2006
- [Ger04] Gerkens, M.: *Standardisierung von 3D-CAD-Modellen und Modellierungsmethoden*. Dissertation, Aachen, Shaker Verlag, 2004
- [GoFr90] Gottlob, G.; Frühwirt, T.; Horn, W.: *Expertensysteme*. Springer Verlag Wien, 1990
- [GöSc00] Görz, G.; Rollinger, C.-R.; Schneeberger, J.: *Handbuch der Künstlichen Intelligenz*, Oldenbourg Wissenschaftsverlag GmbH, Auflage 3, 2000
- [Gra00] Grasmann, M.: *Produktkonfiguration auf Basis von Engineering Data Management-Systemen*. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 75, 2000
- [Haa95] Haasis, S.: *Wissens- und featurebasierte Unterstützung der Konstruktion von Stirnradgetrieben unter besonderer Berücksichtigung des Gußgehäuses*. Dissertation, VDI-Fortschrittsberichte Reihe 1, Nr. 254, VDI-Verlag, Düsseldorf, 1995
- [Haa99] Haasis, S.: *Featurebasierte CAx-Prozessketten*. Tagungsunterlagen, Vorsprung durch Wissen und Technik - Wirtschaftsfaktor Produktionstechnik. Dresdner Produktionstechnik Kolloquium (DPK '99), Dresden, 24. und 25.9.1999
- [HaLu08] Harms, I.; Luckhardt, H.-D.: *Virtuelles Handbuch Informationswissenschaft*. URL: <http://is.uni-sb.de/studium/handbuch/index.html>, (Abgerufen: 31. Juli 2008, 17:11 UTC+1)
- [Har03] Hartmann, D.: *Wissensbasierte Methoden*. Skript zur Vorlesung, Lehrstuhl für Ingenieurinformatik im Bauwesen, Ruhr-Universität Bochum, 2003
- [Hof01] Hoffmann, R.: *Integration höherwertiger Feature-Elemente in den rechnergestützten Konstruktionsprozess*. Dissertation, Universität Duisburg-Essen, VDI-Fortschritt-Bericht Nr. 344, 2001.

- [ISO03] ISO 10303-28: *Industrial automation systems and integration -- Product data representation and exchange - Part 28: Implementation methods: XML representations of EXPRESS schema and data*. Berlin, Beuth Verlag, 2003
- [ISO06] ISO/IEC 9075:2006: *Information Technology - Database Languages - SQL*, 2006
- [ISO94] ISO 10303-1: *Industrielle Automatisierungssysteme und Integration - Produktdatendarstellung und -austausch - Teil 1: Überblick und grundlegende Prinzipien*. Berlin, Beuth Verlag, 1994
- [JaDa97] Janocha, A.; Dankwort, C. W.; Podehl, G.: *Innovative Produktentwicklung - Mit oder trotz Features*. VDI-Berichte, Band 1322, S. 331 – 347, VDI Verlag, Düsseldorf, 1997
- [Jan04] Janitza, D.: *Der Kunde als Konstrukteur – Kundenintegration durch erweiterte CAD-Modelle*. Dissertation, VDI-Fortschritt-Bericht Nr. 392, Düsseldorf 2004.
- [Jar07] Jaroš, M.: *Integration des STEP-Produktmodells in den Getriebeentwicklungsprozess*. Dissertation, Fakultät für Maschinenwesen, TU München, 2007
- [KeEi06] Kemper, A.; Eickler, A.: *Datenbanksysteme - Eine Einführung*. 6. Auflage, Oldenbourg Verlag, München, 2006
- [Kin92] Kingston, J.: *Pragmatic KADS – A Methodological approach to a small knowledge-based system project*. Expert Systems, Band 4, Nr. 4, November 1992
- [KIRa05] Kleinschmidt, P.; Rank, C.: *Relationale Datenbanksysteme*. 3. Auflage, Springer Verlag, Berlin, Heidelberg, 2005
- [KöDu03] Köhler, P.; Dungs, S.: *Auslegung und Modellierung von Greifersystemen*. CAD/CAM-Report Heft 10, Heidelberg, Dressler Verlag, 2003.
- [Köh02] Köhler, P.: *Moderne Konstruktionsmethoden im Maschinenbau*, Würzburg, Vogel Verlag, 2002
- [Köh08] Köhler, P.: *Pro/ENGINEER Praktikum*. 4. Auflage, Vieweg Verlag, Wiesbaden, 2007
- [KrWö01] Krause, F.-L.; Wöhler, Th.: *Automatisierte Ableitung von anwendungsspezifischen Sichten aus Konstruktionsdaten*. VDI-Berichte Nr. 1614, S. 373-387, VDI Verlag, Düsseldorf, 2001
- [Kur92] Kurbel, K.: *Entwicklung und Einsatz von Expertensystemen: Eine anwendungsorientierte Einführung in wissensbasierte Systeme*. Springer, Berlin, Heidelberg, 1992
- [Lie04] Liese, H.: *Wissensbasierte 3D-CAD-Repräsentation*. Dissertation, Aachen, Shaker Verlag, 2004

- [LuPe04] Lubell, J.; Peak, R. S.; Srinivasan, V.; Waterbury, S. C.: *STEP, XML, and UML: Complementary Technologies*. Journal of Computing and Information Science in Engineering, Band 4, S. 379ff., 2004
- [Mei00] Meissner, M.: *Methoden zur qualitätsgerechten CAD-Modellerzeugung für die virtuelle Produktentwicklung am Beispiel der Automobilindustrie*. Dissertation, Aachen, Shaker Verlag, 2000
- [Men99] Mendgen, R.: *Methodische Vorgehensweisen zur Modellierung in parametrischen und featurebasierten 3D-CAD-Systemen*. Shaker Verlag, Aachen 1999.
- [Min75] Minsky, M.: *A Framework for Representing Knowledge*. The Psychology of Computer Vision, McGraw-Hill, New York, USA, 1975
- [Nie06] Niemeier, H.: *Analyse und Optimierung wandnaher Schmelzeströmungen in Extrusionswerkzeugen für die Kunststoffverarbeitung*. Dissertation, Universität Duisburg-Essen, Shaker Verlag, Aachen, 2006
- [Nie96] Nielsen, J.: *Multimedia, Hypertext und Internet: Grundlagen und Praxis des elektronischen Publizierens*. Braunschweig, Vieweg Verlag, 1996.
- [NN07] N. N.: *VIVACE Abschlussbericht*.
URL: http://www.vivaceproject.com/technical_leaflet_final.pdf (Abgerufen: 25. August 2008, 13:01 UTC+1)
- [NN08a] N.N.: *Open CASCADE Dokumentation*.
URL: <http://www.opencascade.org/org/doc/>, Open CASCADE S.A.S., (Abgerufen: 6. Mai 2008, 09:00 UTC+1)
- [NN08b] N. N.: <http://www.dsdm.org>
- [NN08c] N. N.: *Data Management – SQL Call Level Interface (CLI)*.
URL: <http://www.opengroup.org/products/publications/catalog/c451.htm> (Abgerufen: 19. August 2008, 09:11 UTC+1)
- [NN08d] N. N.: *Knowledge system development*.
URL: <http://www.commonkads.uva.nl/frameset-knowsystem.html> (Abgerufen: 20. August 2008, 16:32 UTC+1)
- [NN09] N. N.: *Allgemeine Informationen PDM*.
URL: <http://www.prostep.org/de/services/best-practice/allgemeine-informationen-pdm.html> (Abgerufen: 23. Februar 2009, 13:17 UTC+1)
- [Nor05] North, K.: *Wissensorientierte Unternehmensführung. Wertschöpfung durch Wissen*. 4. Auflage, Gabler, Wiesbaden, 2005

- [NoTa97] Nonaka, I.; Takeuchi, H.: *Die Organisation des Wissens: Wie japanische Unternehmen eine brachliegende Ressource nutzbar machen*. Campus Fachbuch, Frankfurt, New York (USA), 1997
- [NuLo05] Nuzzo, P.; Lockwood, F.: *Knowledge Enabled Solution Components - State of the Art – Version 2*.
URL: http://www.vivaceproject.com/content/advanced/keesoav2_full.pdf (Abgerufen: 18. September 2008, 14:27 UTC+1)
- [OIKn98] Oldham, K.; Kneebone, S.; Callot, M.; Murton, A.; Brimble, R.: *MOKA - A Methodology and tools Oriented to Knowledge-based engineering Applications*. Changing the Ways We Work, Advances in Design and Manufacturing, Ausgabe 8, Proceedings of the Conference on Integration in Manufacturing, Göteborg, Schweden, IOS Press, Amsterdam, October 1998, S. 198-207
- [PaBe07] Pahl, G.; Beitz, W.; Feldhusen, J.; Grote, K.-H.: *Pahl/Beitz Konstruktionslehre*. 7. Auflage, Springer Verlag, Berlin, Heidelberg, 2007
- [PrRa06] Probst, G.; Raub, S.; Romhardt, K.: *Wissen managen: Wie Unternehmen ihre wertvollste Ressource optimal nutzen*. 5. Auflage, Gabler, Wiesbaden, 2006
- [Pup91] Puppe, F.: *Einführung in Expertensysteme*. Springer Verlag, Berlin, Heidelberg, 1991
- [Rie95] Rieger, E.: *Semantikorientierte Features zur kontinuierlichen Unterstützung der Produktgestaltung*. Dissertation, Reihe Produktionstechnik, Bd. 158, TU Berlin, München, Hanser Verlag, 1995
- [RiGo04] Rieg, F.; Goering, J.-U.; Hackenschmidt, R.: *CAD-Datenaustausch in der Praxis*. CAD-CAM Report Heft 1, Heidelberg, Dressler Verlag, 2004
- [Ros60] Ross, D. T.: *Computer-Aided Design: A Statement of Objectives*. MIT Electronic Systems Laboratory Report N. 8346, 1960
- [RoSc01] Rother, K.; Schneider, T.: *Wissensmanagement für Ingenieure - Die Pole-Position behaupten*. CADPlus, Band 3, 2001
- [San01] Sander, S.: *Konzept einer digitalen Lösungsbibliothek für die integrierte Produktentwicklung*. Fortschritt-Berichte, VDI Reihe 1 Nr. 342, VDI-Verlag, Düsseldorf, 2001
- [ScAk99] Schreiber, G.; Akkermans, H.; Anjewierden, A.; De Hoog, R.; Shadbolt, N.; Van De Velde, W.; Wielinga, B.: *Knowledge Engineering and Management - The CommonKADS Methodology*. MIT Press, Cambridge, 1999

- [ScBo05] Scheer, A.-W.; Boczanski, M.; Muth, M.; Schmitz, W.-G.; Segelbacher, U.: *Prozessorientiertes Product Lifecycle Management*. Berlin, Heidelberg, Springer Verlag, 2005
- [Sch07] Schubert, M.: *Datenbanken - Theorie, Entwurf und Programmierung relationaler Datenbanken*, 2. Auflage, Teubner Verlag, Wiesbaden, 2007
- [ShMä95] Sha, J.; Mäntylä, M.: *Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications*. New York, USA, Wiley-Interscience Inc., 1995
- [Sie08] Siemens PLM Software: *Open by Design – Open Automation*.
URL: <http://www.ugsplm.de/produkte/nx/unigraphics/obd/> (Abgerufen: 18. September 2008, 10:51 UTC+1)
- [SpKr97] Spur, G.; Krause, F.-L.: *Das virtuelle Produkt*. Carl Hanser Verlag, München, Wien, 1997
- [Ste07] Stekolschik, A.: *Ein Beitrag zum ganzheitlichen Qualitätsmanagement von CAD-Modellen in der Produktentstehung*. Dissertation, Shaker Verlag, Aachen, 2007
- [Str06] Strohmeier, O.: *Integration von Wissensmodulen in den virtuellen Produktentwicklungsprozess*. Dissertation, Universität Duisburg-Essen, Berichte aus der Konstruktionstechnik, Shaker Verlag, Aachen 2006.
- [Stra06] Straßmann, T.: *3D-CAD-Einsatz erfordert Konstruktionsmethodik*, CAD-CAM Report, Band 25, Heft 1, 2006
- [SuSt07] Surek, D.; Stempin, S.: *Angewandte Strömungsmechanik für Praxis und Studium*. Teubner Verlag, Wiesbaden, 2007
- [Sut03] Sutherland, I. E.: *Sketchpad: A man-machine graphical communication system*. Cambridge, University of Cambridge Technical Report 574, 2003
- [TeSt00] Terpenney, J.; Strong, S.; Wang, J.: *A methodology for knowledge discovery and classification*. Proceedings of the 10th Flexible Automation and Intelligent Manufacturing Conference, University of Maryland, College Park, Maryland, 26.6 – 28.6 2000
- [VaPo98] Vajna, S.; Podehl, G.: *Durchgängige Produktmodellierung mit Features*. CAD-CAM Report, Heft 3, 1998
- [VDI03a] VDI-Richtlinie 2218: *Informationsverarbeitung in der Produktentwicklung - Feature-Technologie*. Beuth Verlag, Berlin, Köln, 2003
- [VDI03b] VDI-Richtlinie 2249: *Informationsverarbeitung in der Produktentwicklung – CAD-Benutzungsfunktionen*. Beuth Verlag, Berlin, Köln, 2003
- [VDI06] VDI-Richtlinie 2209: *3D-Produktmodellierung*. Beuth-Verlag, Berlin, Köln, 2006

- [VDI87] VDI-Richtlinie 2235: *Wirtschaftliche Entscheidungen beim Konstruieren – Methoden und Hilfen*. Beuth Verlag, Berlin, Köln, 1987
- [VDI93] VDI-Richtlinie 2221: *Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte*. Beuth Verlag, Berlin, Köln, 1993
- [WiMe09] Wieczorrek, H.-W.; Mertens, P.: *Wirtschaftlichkeitsanalysen für IT-Investitionen*. Berlin, Heidelberg, Springer Verlag, 2009
- [WSB92] Wielinga, B. J.; Schreiber, A. Th.; Breuker, J. A.: *KADS - A Modelling Approach to Knowledge Engineering*. Knowledge Acquisition, Band 4, Heft 1, 1992

Eigene Veröffentlichungen

- [DuLu06] Dungs, S.; Lupa, N.; Köhler, P.: *Das Mastermodell als Basis teilautomatisierter Simulation*. ProduktDaten Journal Nr. 1, ProSTEP 07/2006
- [DaLu08] Danjou, S.; Lupa, N.; Köhler, P.: *Approach for Automated Product Modeling Using Knowledge-Based Design Features*. Computer-Aided Design & Applications, Vol. 5, No. 5, 2008, S. 622-629
- [DaLu08a] Danjou, S.; Lupa, N.; Köhler, P.: *Feature-basierende Produktmodellierung zur Sicherung des Wissenstransfers*. CAD-CAM Report, Nr. 2, 2008, S. 50-54
- [DaLu08b] Danjou, S.; Lupa, N.; Köhler, P.: *Approach for a Modular Knowledge-Based CAD Modeling Process*. Proceedings of the 18th CIRP Design Conference, University of Twente, Enschede, NL, 2008
- [DaLu08c] Danjou, S.; Lupa, N.; Köhler, P.: *Einsatz KBE-basierter Konstruktionsfeatures*. VDI Konstruktion, November 2008, S. 56-60
- [KoWo08] Köhler, P.; Wortberg, J.; Lupa, N.; Saul, K.: *Integrative Interconnection of design, calculation and simulation for dimensioning a flow channel in polymer processing*. PPS 2008, Salerno, Italien
- [KoWo09] Köhler, P.; Wortberg, J.; Lupa, N.; Saul, K.: *Automatisierte Optimierung von Strömungskanälen zur Verbesserung des Spülverhaltens von Kunststoff-Extrusionswerkzeugen*. Zeitschrift Kunststofftechnik / Journal of Plastics Technology, Seite 130-153, Februar 2009

7 Abbildungsverzeichnis

ABBILDUNG 1-1:	KOSTENFESTLEGUNG UND KOSTENVERURSACHUNG [VDI87]	2
ABBILDUNG 2-1:	VORGEHENSMODELL FÜR DAS ENTWICKELN UND KONSTRUIEREN ([VDI93] UND [PABe07])	7
ABBILDUNG 2-2:	ZUORDNUNG VON KONSTRUKTIONSPHASEN ZU KONSTRUKTIONARTEN.....	8
ABBILDUNG 2-3:	GEOMETRIEBESCHREIBUNG IM CAD-SYSTEM	9
ABBILDUNG 2-4:	ARTEN VON VOLUMENMODELLEN [DEN04].....	10
ABBILDUNG 2-5:	CSG-STRUKTUR	11
ABBILDUNG 2-6:	B-REP-MODELL	12
ABBILDUNG 2-7:	DATENSTRUKTUREN DES B-REP-MODELLS	13
ABBILDUNG 2-8:	EULER-POINCARÉ-GLEICHUNG DES ZYLINDERS	13
ABBILDUNG 2-9:	ASSOZIATIVE MODELLIERUNG EINES SWEEP-KÖRPERS	15
ABBILDUNG 2-10:	FEATURE IM PRODUKTLEBENSZYKLUS (NACH [JADA97]).....	17
ABBILDUNG 2-11:	DURCHDRINGUNG DER FEATURE-TECHNOLOGIE IN DEN PHASEN DES PRODUKTENTWICKLUNGSPROZESSES (IN ANLEHNUNG AN [VAPO98])	19
ABBILDUNG 2-12:	AUFBAU EINES DATENBANKSYSTEMS	20
ABBILDUNG 2-13:	RELATION IN TABELLENDARSTELLUNG.....	22
ABBILDUNG 2-14:	ODBC-SCHNITTSTELLE	24
ABBILDUNG 2-15:	INTEGRATION DER PRODUKTKONFIGURATION IN DAS BETRIEBLICHE INFORMATIONSMANAGEMENT (IN ANLEHNUNG AN [FEFR99])	28
ABBILDUNG 2-16:	WISSENSARTEN (IN ANLEHNUNG AN [HALU08])	30
ABBILDUNG 2-17:	WISSENSTRANSFORMATION IM SECI-MODELL.....	31
ABBILDUNG 2-18:	WISSENSTREPPE (NACH [NOR05]).....	31
ABBILDUNG 2-19:	DIFFERENZIERUNG VON DATEN UND WISSEN	32
ABBILDUNG 2-20:	STRUKTURELLER AUFBAU EINES WISSENSBASIERTEN SYSTEMS	33
ABBILDUNG 2-21:	CONSTRAINT-NETZ	35
ABBILDUNG 2-22:	SEMANTISCHES NETZ [KUR92]	36
ABBILDUNG 2-23:	OBJEKT-ATTRIBUT-WERT-TRIPEL ALS SEMANTISCHES NETZ.....	37
ABBILDUNG 2-24:	FRAME	37
ABBILDUNG 2-25:	VERKNÜPFUNG UNTERSCHIEDLICHER FORMEN DER WISSENSREPRÄSENTATION	38
ABBILDUNG 2-26:	ABLAUFSTEUERUNG REGELBASIERTER LÖSUNGSSUCHE.....	39
ABBILDUNG 2-27:	INTEGRATIONSTIEFE VON KBE-SYSTEMEN	42
ABBILDUNG 3-1:	DIE COMMONKADS MODELLE [SCAK99]	43
ABBILDUNG 3-2:	KBE-LEBENSZYKLUS IN MOKA	45
ABBILDUNG 3-3:	ENTWICKLUNGSZEITEN UND BEGUTACHTUNG IM DSDM.....	48
ABBILDUNG 3-4:	KOMPONENTENKLASSIFIKATION IN KCM.....	49

ABBILDUNG 4-1:	VERDICHTER MIT HORIZONTALER (LINKS) UND VERTIKALER GEHÄUSETRENNFUGE (RECHTS) (QUELLE: SIEMENS-PRESSEBILD)	54
ABBILDUNG 4-2:	MODULARER WELLENAUFBAU	56
ABBILDUNG 4-3:	KONSTRUKTIONSSTANDARDÜBERGREIFENDE ABFRAGEN.....	57
ABBILDUNG 4-4:	LAUFRAD (HALBSCHNITT UND DREIDIMENSIONAL)	58
ABBILDUNG 4-5:	RADSCHIEBEN MIT UNTERSCHIEDLICHEN SCHAUFELTYPEN (LINKS: 2D, RECHTS: 3D)	59
ABBILDUNG 4-6:	POSITIONIERUNG DER RADIALTEILELEMENTE.....	62
ABBILDUNG 4-7:	AUFBAU DES ANFORDERUNGSKATALOG.....	64
ABBILDUNG 4-8:	GRUNDLEGENDER AUFBAU DES KONFIGURATIONSSYSTEMS	68
ABBILDUNG 4-9:	KOEXISTENZ VON CAD-FEATURES UND WISSENSREPRÄSENTATION.....	70
ABBILDUNG 4-10:	ABBILDUNG DER KOMPONENTENBESCHREIBUNG AUF EINE LISTENSTRUKTUR IN XML-NOTATION	71
ABBILDUNG 4-11:	AKZEPTANZMODELL NACH NIELSEN [Nie96]	73
ABBILDUNG 4-12:	DATENBANKANBINDUNG UND -ABFRAGE.....	76
ABBILDUNG 4-13:	DATENBANKGESTEUERTE MODELLPARAMETER	77
ABBILDUNG 4-14:	EINFACHE ERZEUGUNG VARIABLER TOPOLOGIEN	80
ABBILDUNG 4-15:	WISSENSBASIERTE ERZEUGUNG VARIABLER TOPOLOGIEN	81
ABBILDUNG 4-16:	AUSGANGSDATEN DER BESCHAUFELUNG	82
ABBILDUNG 4-17:	VERARBEITETE AUSGANGSDATEN DER BESCHAUFELUNG	83
ABBILDUNG 4-18:	KOMBINIERTER AKTUALISIERUNGSSTEUERUNG	85
ABBILDUNG 4-19:	EINSATZ VON KONNEKTOREN FÜR DIE AUSRICHTUNG DER GESTALTUNGSMODULE.....	87
ABBILDUNG 4-20:	ERZEUGUNG DER BOOLESCHEN ELEMENTVERKNÜPFUNGEN	88
ABBILDUNG 4-21:	APPLIKATIONSSTRUKTUR.....	89
ABBILDUNG 4-22:	PRODUKTSPEZIFISCHE KLASSEN	90
ABBILDUNG 4-23:	AUFBAU VON DIALOGKOMPONENTEN IM USER INTERFACE STYLER VON NX...	91
ABBILDUNG 4-24:	ERZEUGUNG VON UDF-EINGABEDATEN.....	92
ABBILDUNG 4-25:	ERZEUGUNG DER UDF-DATENBANKVERBINDUNG	93
ABBILDUNG 4-26:	EXEMPLARISCHER AUFBAU EINES UDFs	94
ABBILDUNG 4-27:	HAUPTDIALOGFENSTER.....	95
ABBILDUNG 4-28:	KOMPONENTENSPEZIFISCHE DIALOGKOMPONENTEN	96
ABBILDUNG 4-29:	EINORDNUNG DES KONFIGURATIONSSYSTEMS IN DEN GESAMTPROZESS	99
ABBILDUNG 4-30:	TRENNUNG VON WISSEN, GEOMETRIE UND DATEN IN EINEM GESTALTUNGSMODUL.....	102
ABBILDUNG 4-31:	SPÜLVORFÄHREN IN UMLLENKUNGEN [Nie06].....	103
ABBILDUNG 4-32:	OPTIMIERUNGSSTRATEGIE AUF BASIS DES CAD-MODELLS.....	105
ABBILDUNG 4-33:	EINFLUSS DER FERTIGUNGSPARAMETER AUF DIE KANALTOPOLOGIE.....	105

8 Tabellenverzeichnis

TABELLE 4-1:	EINFACHE BEZUGSGRÖßE.....	56
TABELLE 4-2:	MULTIPLE BEZUGSGRÖßEN	57
TABELLE 4-3:	KOSTEN DES SOFTWARESYSTEMS	97
TABELLE 4-4:	VERGLEICH ZWISCHEN MANUELLER UND AUTOMATISIERTER MODELLERSTELLUNG	98
TABELLE 4-5:	NUTZENEFFEKTE DER AUTOMATISIERTEN MODELLERSTELLUNG.....	98

Lebenslauf

10.11.1978	geboren in Wesel; Eltern: Gisela Lupa, geb. Remmy, Hans-Jürgen Lupa
1985 – 1989	Grundschule in Neukirchen-Vluyn
1989 – 1998	Gymnasium in Neukirchen-Vluyn; Abschluss: Allgemeine Hochschulreife
1998 – 1999	Zivildienst: Küsterhilfe der evangelischen Kirchengemeinde Vluyn
1999 – 2003	Studium des Maschinenbaus an der Gerhard-Mercator-Universität Duisburg; Abschluss: Bachelor of Engineering
1999 – 2003	Studium des Maschinenbaus an der Universität Duisburg-Essen, Studien- schwerpunkt Produktengineering; Abschluss: Diplom-Ingenieur
2002 – 2004	Studentische Hilfskraft im Fachgebiet Rechnereinsatz in der Konstruktion
2004 – 2009	Wissenschaftlicher Mitarbeiter am Institut für Produkt Engineering, Rechner- einsatz in der Konstruktion Ordinarius: Prof. Dr.-Ing. Dipl.-Math. P. Köhler
seit März 2009	Ingenieur bei der SMS Meer GmbH in Mönchengladbach
Familienstand	verheiratet

