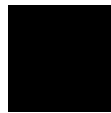


kassel
university



press

**Der fertigungsorientierte Modellierer FERMOD als Erweiterung
des Konstruktionssystems WISKON**

Axel Berthold

Die vorliegende Arbeit wurde vom Fachbereich Maschinenbau - der Universität Kassel als Inaugural-Dissertation zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften (Dr.-Ing.) angenommen.

Erster Gutachter: Prof. Dr.-Ing. B. Klein

Zweiter Gutachter: Prof. Dr.-Ing. F. Tikal

Tag der mündlichen Prüfung

13. Februar 2002

Bibliografische Information Der Deutschen Bibliothek

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

Zugl.: Kassel, Univ., Diss. 2002

ISBN 3-933146-73-9

© 2002, kassel university press GmbH, Kassel

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsschutzgesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Umschlaggestaltung: 5 Büro für Gestaltung, Kassel

Druck und Verarbeitung: Zentraldruckerei der Universität Kassel

Printed in Germany

Vorwort

Die vorliegende Arbeit entstand im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter im Fachgebiet Leichtbau-Konstruktion des Fachbereichs Maschinenbau der Universität Gesamthochschule Kassel.

Den Grundstein für diese Arbeit hat Herr Prof. Dr.-Ing. B. Klein gelegt indem er die Arbeitsgruppe „WISKON - Wissensbasiertes System für die mechanische Konstruktion“ initiierte. Dafür sowie für die langfristige Unterstützung dieser Arbeitsgruppe und die Gewährung der erforderlichen Freiräume zur Erstellung dieser Arbeit danke ich Ihm.

Herrn Prof. Dr.-Ing. F. Tikal, dem Leiter des Fachgebiets Produktionstechnik und Werkzeugmaschinen der Universität Kassel, danke ich für die wissenschaftliche Diskussion und die Übernahme des Koreferates.

Herrn Prof. Dr.-Ing. G. Johannsen sowie Herrn Prof. Dr.-Ing. H. Martin danke ich für ihre Bereitschaft als weitere Mitglieder der Promotionskommission mitzuwirken.

Herrn Dr.-Ing. D. Eulenbach sowie Herrn Dr.-Ing. W. Scherm danke ich für die zahlreichen wissenschaftlichen Diskussionen.

Nicht zuletzt gilt mein Dank den Mitarbeiterinnen und Mitarbeitern des Fachgebietes Leichtbau-Konstruktion.

Edertal-Böhne, im Januar 2001

Axel Berthold

1	Einleitung	1
1.1	Problemstellung	1
1.2	Stand der Technik	3
1.3	Zielsetzung und Gliederung der Arbeit	15
2	Die Konstruktionsmethodiken	17
2.1	Anforderungen an die Konstruktion	18
2.1.1	Die VDI-Richtlinien 2221, 2222 und 2223.....	19
2.1.2	Der Konstruktionsprozeß nach Pahl und Beitz.....	26
2.1.3	Der Konstruktionsprozeß nach Koller	28
2.1.4	Der Konstruktionsprozeß nach Roth.....	30
2.2	Zusammenfassung der Konstruktionsmethoden.....	33
2.3	Ableitbare Modelle	35
2.4	Anforderungen an EDV-Systeme	38
3	Fertigungstechnische Grundlagen	43
3.1	Gliederung der Fertigungsverfahren.....	43
3.2	Die Gliederung der Werkzeugmaschinen.....	48
3.2.1	Die Steuerung von Werkzeugmaschinen	49
4	CAD Systemaufbau	53
4.1	Der Systemaufbau von Standardsystemen	54
4.1.1	Modellierkerne	54
4.1.2	Der ACIS Modellierkern.....	56
4.1.3	Die CV-DORS-Klassenhierarchie	62
4.1.3.1	Die Klasse Geometrie	63
4.1.3.2	Die Klasse Topologie	64
4.1.3.3	Klassen zur Listenverwaltung	65
4.1.3.4	Die Datenbankklasse	66
4.1.3.5	Mathematische Funktionen	66
4.1.4	Grafische Programmierschnittstellen.....	67
4.1.5	Benutzeroberflächen	76
4.1.6	Erweiterungen	78
4.1.7	Software-Schnittstellen	79
4.2	Möglichkeiten der anwenderspezifischen Erweiterbarkeit.....	85
4.2.1	Die Generierung von MAKROS.....	86
4.2.2	Die Variantenprogrammierung	87
4.2.3	Die Anwendungsprogrammierung.....	88
4.2.4	Parametermodelle	90
4.2.5	Die Erzeugung von Features.....	92
5	Systemarchitekturen	95
5.1	Das System WISKON	97
5.1.1	Geomod.....	97
5.1.2	Fermod	100
5.1.3	Konzept & Topkon	106
5.1.4	WISMOD.....	106

Inhaltsverzeichnis

5.2	Die Implementierung durch Standardsoftware	106
5.2.1	Der Systemaufbau mit Einzelkomponenten	107
5.2.1.1	Die Schnittstelle zwischen CV-DORS und dem Benutzerinterface	108
5.2.1.2	Die Schnittstelle zwischen CV-DORS und HOOPS	109
5.2.1.3	Das Menüsystem	111
5.2.1.4	Das grafische Fenster	115
5.2.2	Die Einbindung in ein CAD-System	117
5.2.2.1	Das CAD-Programmierinterface	118
5.2.2.2	Die Verwendung des Benutzerinterfaces	118
6	Die Fermodklassen	123
6.1	Halbzeuge.....	124
6.1.1	Die Klasse „t_halfzeug“	125
6.1.2	Die Klasse „t_profile“	126
6.1.3	Die Klasse „t_bleche“	129
6.1.4	Die Klasse „t_rohre“	130
6.2	Fertigungsverfahren	132
6.2.1	Das Fertigungsverfahren Drehen.....	134
6.2.1.1	Das Unterverfahren Querplandrehen	135
6.2.1.2	Das Verfahren des Querabstechdrehens	137
6.2.1.3	Das Längsplandrehen	139
6.2.1.4	Profildrehen	140
6.2.1.5	Das Längsrunddrehen	141
6.2.1.6	Die Formelemente	143
6.2.2	Das Fertigungsverfahren Fräsen.....	146
6.2.3	Das Fertigungsverfahren Bohren.....	149
6.2.4	Die Erzeugung von Spankörpern	150
6.2.5	Der Aufbau der Programme zur Erzeugung von Spankörpern	152
6.2.6	Werkzeugmaschinen.....	152
6.3	Die Benutzeroberfläche.....	155
6.3.1	Das Fermod-Hauptfenster	155
6.3.2	Die Menüleiste	159
6.3.3	Die Untermenüstruktur.....	160
6.4	Der Arbeitsablauf.....	161
6.5	Die Erweiterbarkeit	162
6.6	Datenschnittstellen	163
7	Zusammenfassung	165
8	Literaturverzeichnis	169

1 Einleitung

1.1 Problemstellung

In der mechanischen Konstruktion stellen CAD-Systeme mittlerweile ein gebräuchliches Hilfsmittel dar. Die ersten Systeme wurden zunächst zur reinen Zeichnungsunterstützung konzipiert. Sie imitierten lediglich die vorhandenen Zeichenmaschinen und unterstützten nur den Vorgang des Detaillierens. Die CAD-Systeme besaßen jedoch gegenüber dem Zeichnen am „Brett“ die Vorteile, daß die Zeichnungen beliebig oft reproduzierbar waren, ästhetisch perfekte Zeichnungen erstellt werden konnten, Änderungen leichter und schneller durchführbar waren und die Zeichnungsarchivierung wesentlich effektiver durchgeführt werden konnte. Das Arbeiten fand anfangs nur im 2-dimensionalen Raum statt. Durch die rasante Entwicklung der Rechnertechnologie und die damit verbundene Leistungssteigerung wurden auch die CAD-Systeme zunehmend komplexer.

Im Bild 1.1 ist die Entwicklung der CAD-Modelle dargestellt :

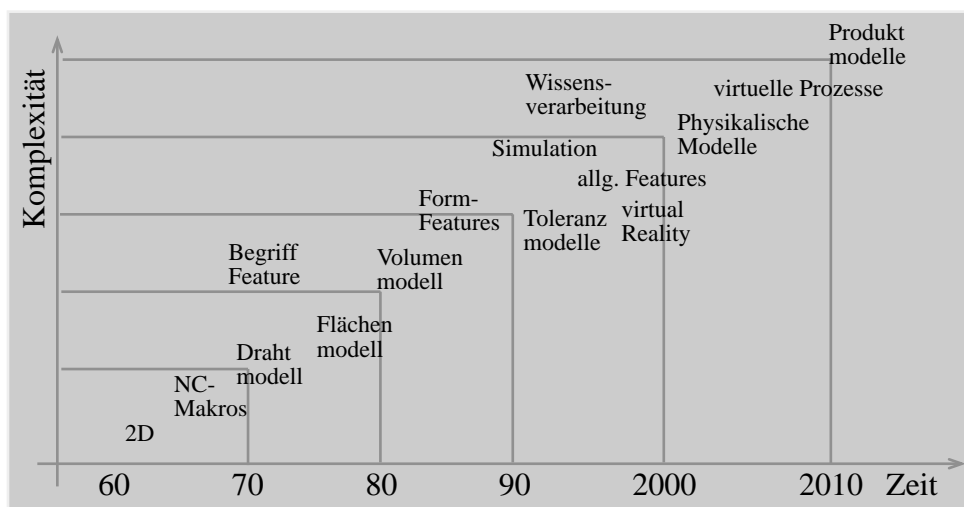


Bild 1.1 Modellkomplexität in Abhängigkeit der Zeit

Heutige CAD-Systeme besitzen einen sehr hohen Entwicklungsstand im Bereich der Geometriemodellierung /44, 45, 137 und 138/. Durch die vorhandene Rechnerleistung konnte so ein sehr großer Teil des geometrischen Modellierens exakt in einem rechnerinternen Modell abgebildet werden. Sehr komplexe rechnerinterne Modelle entstanden. Um diese zu erstellen muß ein Konstrukteur einen komplexen Modellaufbau durchführen. Dieser richtet sich nach exakten mathematischen Definitionen aus der Welt der Geometrie. Ferner muß er zur Erstellung der Modelle eine mächtige Benutzeroberfläche beherrschen. Diese besitzt wegen der komplexeren Modelle immer mehr Befehle, Optionen und Möglichkeiten des Modellaufbaus. Die komplizierte Handhabung sowie der schwierige Modellaufbau behindern den Konstrukteur bei seiner eigentlichen Tätigkeit jedoch erheblich. Die Akzeptanz des Arbeitens mit CAD-Systemen wird aufgrund der vielen Fehlermöglichkeiten geringer. Eine vereinfachte, problemorientierte Benutzeroberfläche würde dem ingenieurmäßigen Arbeiten wesentlich gerechter werden als die zur Zeit vorhandenen komplexen Systeme. Das Zitat: „Die Entwicklung von Anwendersoftware zur Unterstützung von Konstruktionsaufgaben als Ergänzung handelsüblicher Basissoftware ist wichtig, da die bekannten CAD-Systeme wegen ihrer großen Anwenderbandbreite nicht den gewünschten Benutzerkomfort und effiziente Arbeitsbedingungen liefern“ aus /8/ unterstützt diese These nachhaltig.

Ein weiteres Manko heutiger CAD-Systeme besteht darin, daß der Konstrukteur sein komplexes Gedankenmodell auf ein rein geometrisches Modell reduzieren muß /31/. Dabei gehen wichtige Informationen verloren, die in nachfolgenden Produktentwicklungsschritten wieder erarbeitet werden müssen. Es muß ein Ziel zukünftiger CAD-Systeme sein, dem Konstrukteur Möglichkeiten zu bieten, nicht nur sein geometrisches, sondern auch seine anderen Modelle in einem CAD-System abzulegen, um sie für den nachfolgenden Entwicklungsprozeß zu nutzen. Im Bild 1.2 ist dieser Sachverhalt verdeutlicht. Es zeigt den Einsatz von CAD-Systemen in den verschiedenen Bereichen des Produktlebenszyklus. Da der CAD-Einsatz im wesentlichen jedoch nur in der Phase der Detailkonstruktion stattfindet, können auch nur Daten dieser Phase dauerhaft gesichert werden. Die Definition des STEP-Standards zeigt, daß die derzeitigen Systementwickler die Einbeziehung nicht grafischer Daten in ein CAD-System als notwendig ansehen.

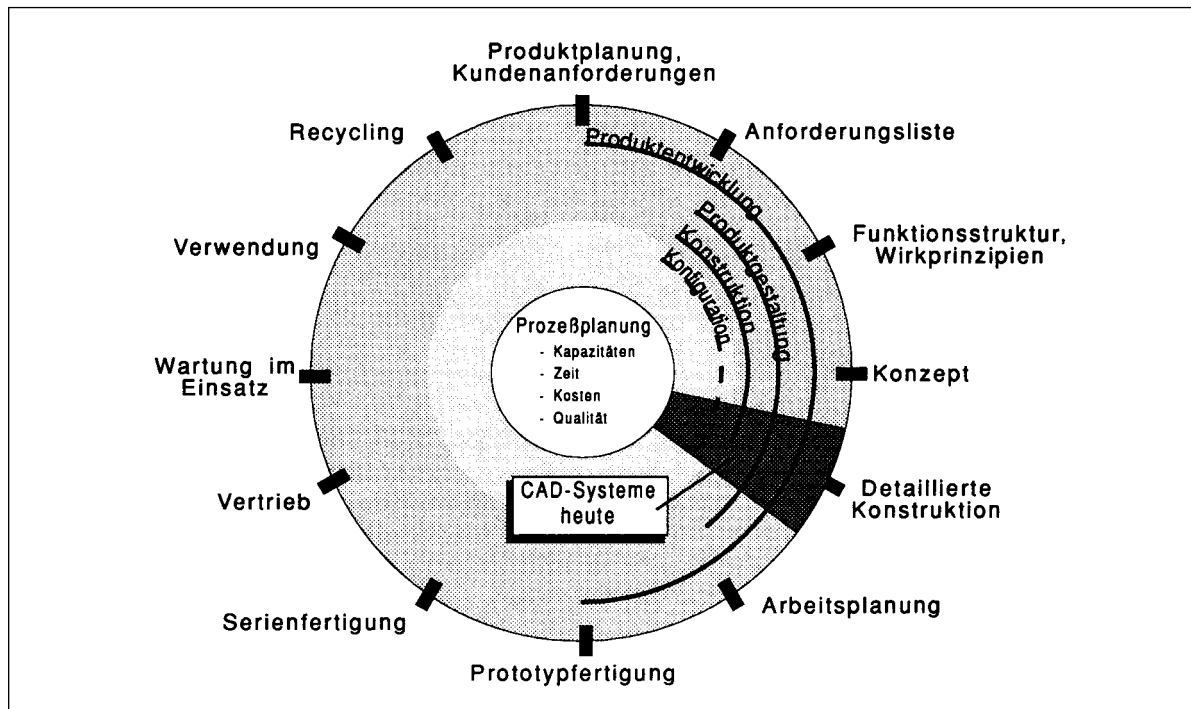


Bild 1.2 Produktlebenszyklen aus /61/

Neben dem Vorteil der Erhaltung von Daten zwingen die verschiedenen Modelle den Konstrukteur zu einer ganzheitlichen Betrachtung seines Handelns. In einem immer härter und internationaler werdenden Wettbewerb ist dies unter der Berücksichtigung der Tatsache, daß der Konstrukteur bis zu 80% /24 und 28/ der zukünftigen Produktionskosten festlegt, zwingend notwendig.

1.2 Stand der Technik

Die derzeit im industriellen Einsatz befindlichen CAD-Systeme sind fast ausnahmslos auf Workstations und High-End PC's lauffähig. Die Firma AutoDesk mit ihrem lowcost CAD-System AutoCad, das bereits vom Anfang seiner Entwicklung auf PC's der mittleren Preiskategorie lauffähig war, stellt hier eine Ausnahme dar. Großrechner spielen für den Einsatz von CAD-Systemen eine immer geringere Rolle.

Workstations und High End PC's dominieren diesen Bereich aufgrund des Erreichens einer hohen Rechenleistung, sowie der Steigerung ihrer Grafik- und Netzwerkfähigkeit. Bei einer gleich großen Anzahl von CAD-Arbeitsplatzinstallationen ist ein Netzwerk bestehend aus Workstations und PC's wesentlich leistungsfähiger, ausfallsicherer und kostengünstiger als ein

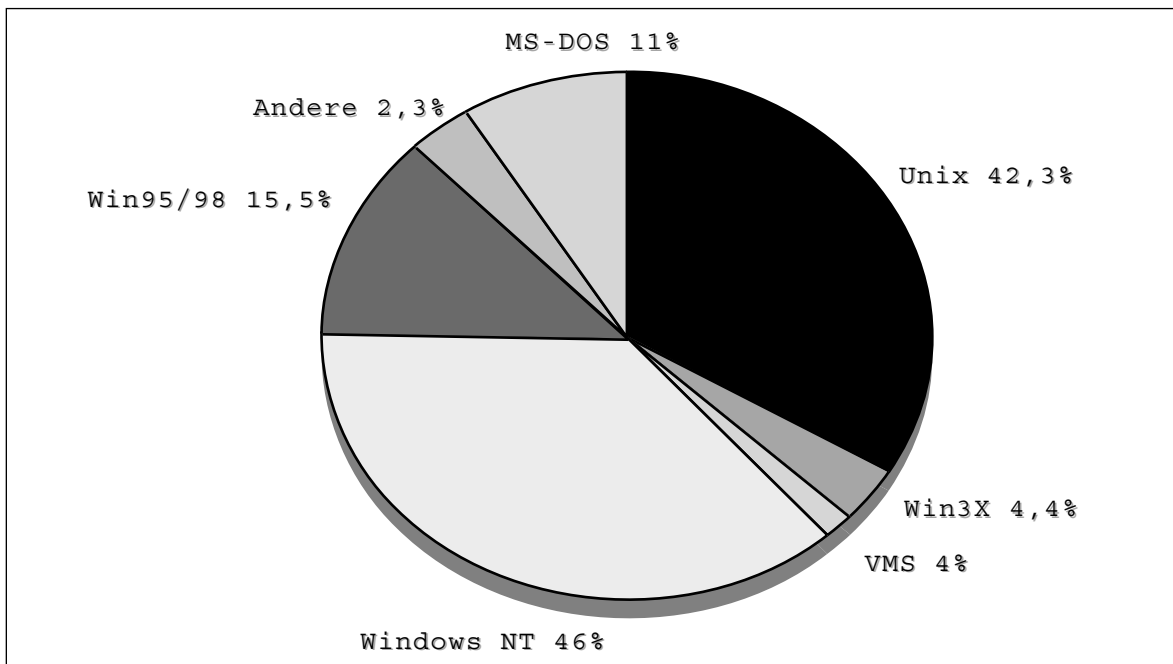


Bild 1.3 Prozentualer Einsatz von CAD-Systemen auf Rechnerkategorien /87/

entsprechender Großrechner. Dementsprechend dominieren CAD-Systeme die auf Workstations und PC's unter den Betriebssystemen UNIX und Windows NT installiert sind ca. 80 % des Marktes. Bild 1.3 zeigt eine Übersicht der von CAD-Systemen genutzten Betriebssysteme. Der CAD-Markt wird derzeit im wesentlichen von zwei Anbietern dominiert. Die Firmen IBM sowie Parametric Technologies besitzen mit ihren CAD-Systemen CATIA sowie ProEngineer ca. 30 % /87/ des deutschen CAD-Marktes. Die restlichen 70 % des CAD-Marktes sind von kleineren Anbietern besetzt, die aber jeweils nicht mehr als 5 % des Marktes besetzen.

Diese Marktsituation besitzt einen gewissen Einfluß auf die Weiterentwicklung der genannten CAD-Systeme. Um die Marktposition zu halten müssen die Systeme einen breiten Anwenderkreis ansprechen. Dies setzt voraus, daß sie ein hohes Maß an Funktionalität besitzen. Die Folge hiervon ist, daß die Systeme kompliziert zu bedienen und fehleranfällig sind. Der einzelne Anwender wiederum benötigt lediglich einen Bruchteil der vom System zur Verfügung gestellten Funktionen.

Diesem Trend entgegenwirkend ist der Beginn der Modularisierung der Systeme. Das System CADD5^{1/90/} besteht in seinem Kern aus einem Modellierer sowie einer Datenbank. Um diesen herum sind verschiedene Zusatzmodule angeordnet. Neben geometrischen Erweiterungen wie einem speziellen Flächenmodellierer gibt es einen CNC- sowie einen FEM- und einen Parametric Modul.

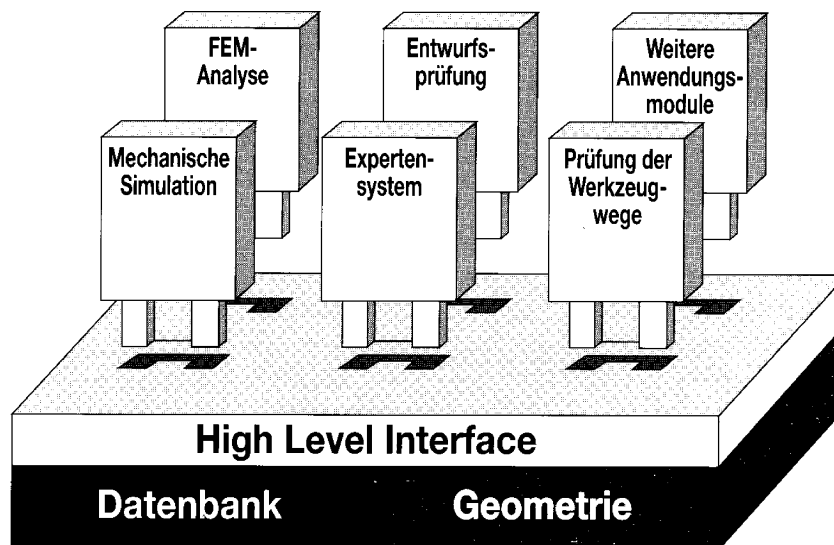


Bild 1.4 CADD5 bzw. CV-DORS Systemaufbau /96/

Im Bild 1.4 ist dargestellt, wie sich ein Systemaufbau auf der Basis der Standardsoftwarekomponenten Datenbank und Geometriemodellierer gestaltet. Auf deren Interfaces wird über ein sogenanntes High Level Interface zugegriffen. Dies bietet externen Modulen einen komfortablen Zugriff auf einen mächtigen Funktionsumfang. Ein individueller Konstruktionsprozess kann so von der Berechnung bis hin zur Prüfung des Werkstücks auf einer Softwarebasis ablaufen.

1. CADD5 wird inzwischen von der Firma Parametric Technologie gepflegt und vertrieben.

Ein weiteres Beispiel für dieses Konzept gibt die Firma SDRC mit ihrem System IDEAS. Es besteht aus den im Bild 1.5 dargestellten Modulen. Diese wurden zum größten Teil von Fremdfirmen entwickelt und werden von diesen auch als Einzelpakete vertrieben. SDRC entwickelte eine einheitliche Benutzeroberfläche sowie die entsprechenden Programme zum Transformieren der Daten von einem Modul zu einem anderen.

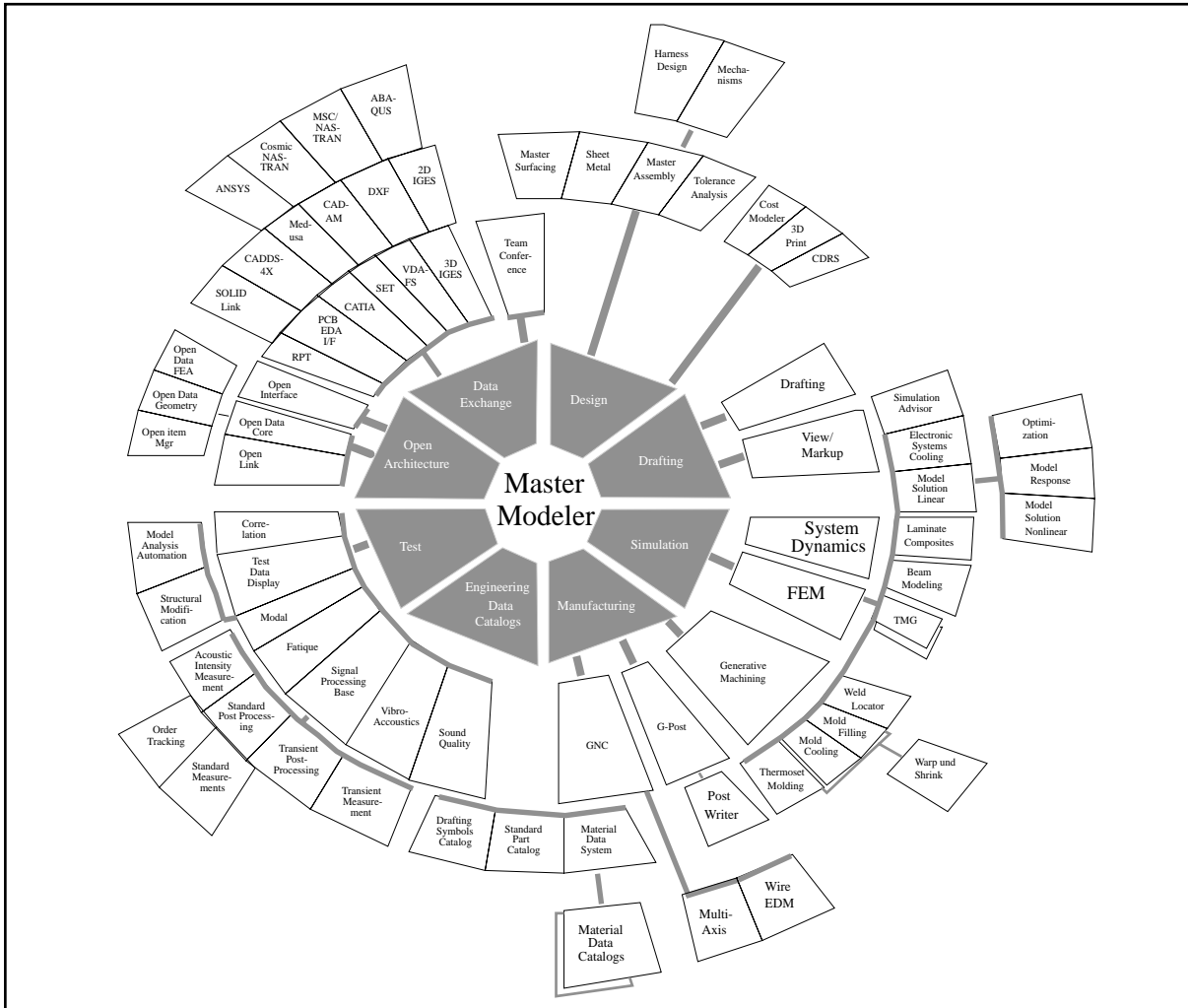


Bild 1.5 SDRC und Fremdanbieter der einzelnen Module /95/

Je nach Bedürfnissen kann sich der Kunde so sein benötigtes System zusammenstellen. Neben diesen derzeit im Einsatz befindlichen Systemen existieren weitere Forschungsvorhaben zur Entwicklung zukünftiger CAD-Systeme.

KRONOS /36 und 97/ steht als Projekttitel für ein logik- und geometrieverarbeitendes CAD-System. Das Projektziel besteht darin, ein Basissystem für intelligente CAD-Systeme zu schaffen. Die Integration von Produktdaten, Konstruktionsdaten, Konstruktionsmethodik und Wissen über Produktverhalten und -fähigkeiten sollen durchgeführt werden. Dabei wird auf logikbasierten Techniken des Knowledge-Engineering sowie der geometrischen Interpretation und Darstellung von Produktcharakteristiken aufgesetzt.

Zur Erforschung von rechnergestützten Konzepten und Techniken dient **KACTUS** /100/ (Modelling Knowledge About Complex Technical System for multiple Use). In diesem Projekt werden praxisrelevante Methoden erarbeitet und untersucht, die eine Verwendung von modelliertem Wissen über technische Produkte in unterschiedlichen Anwendungsdomänen unterstützen. Wissensmodelle sollen dabei auf Basis von Ontologien aufgebaut werden. Ontologien werden dabei als die explizite Spezifikation einer abstrakten und vereinfachten Sicht auf die Modellwelt verstanden.

Das Konstruktionssystem **mfk** /38, 39 und 74/ setzt auf 2D- und 3D-CAD-Systemen auf. Es soll herkömmliche Systeme nicht ablösen, sondern ergänzen. Koppelungen wurden zu den CAD-Systemen ICEMDDN, Pro/ENGINEER, SIGRAPH-2D, -3D und Design, MEDUSA sowie ICEMDUCT realisiert.

Das aus einem Synthese- und Analyseteil bestehende System ist über ein Produktmodell miteinander verbunden. Es bildet die funktionellen, topologischen und hierarchischen Beziehungen zwischen Baugruppen und Bauteilen ab. Der Syntheseteil erlaubt die von einem Fertigungsverfahren abhängige Erzeugung von Bauteilen und Produkten auf Basis von konstruktiven Elementen. Diese bestehen aus Gestaltelementen, Technologieelementen, Funktionselementen und Organisationselementen. Mit dem Analyseteil werden Konstruktionen beurteilt. Er besteht aus einer Wissensbasis und aus Informationsmodulen zur Analyseverarbeitung (notwendige Daten sammeln), Diagnose (Fehlerermittlung, Sollwerte errechnen), Beratung (Verbesserungs-, bzw. Lösungsvorschläge) und Korrekturphase (übernimmt Verbesserungsvorschläge automatisch in die Konstruktion).

Das nachfolgende Bild stellt den beschriebenen Aufbau des Konstruktionssystems mfk aus Komponenten dar.

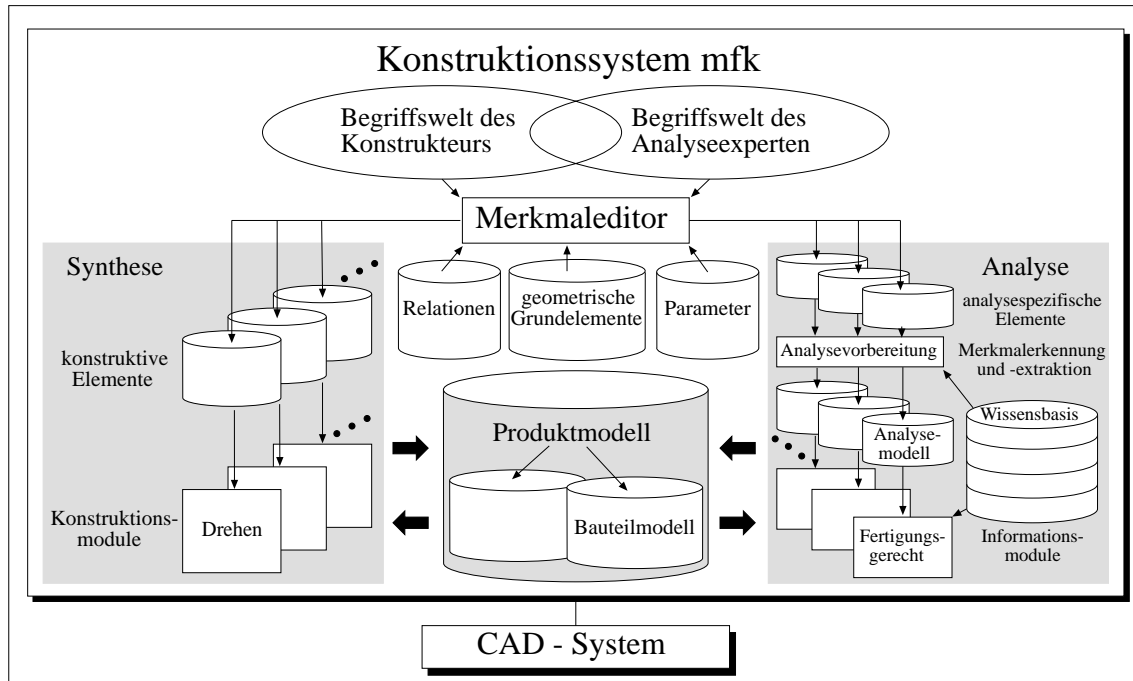


Bild 1.6 Erweiterter Aufbau des Konstruktionssystems mfk /38/

Weitere Forschungsaktivitäten im Rahmen des mfk sind: Konzipieren mit Wirkstrukturen, Drehteile, Blechteile, Gußteile und ein Merkmaleeditor.

Das an der Helsinki Universität of Technology entwickelte System **HutCAPP** /133/ basiert auf einer hybriden Softwarearchitektur. Sie besteht aus einem in „C“ implementierten geometrischen Modellersystem und einer in LISP geschriebenen Feature basierten Sprache. Dies ist einer der ersten objektorientierten Ansätze. Mit ihr werden alle benötigten Features definiert. Der Zugriff auf den Modellierer erfolgt in Form einer Library. Die Features besitzen Referenzen auf diese Library um ein entsprechendes geometrisches Modell zu erzeugen. Das gesamte Bauteil wird erzeugt, indem man die Featurehierarchie durchläuft, dabei alle Geometrien erzeugt und diese über die Booleschen Operationen verknüpft. Nach der Änderung eines Features muß die gesamte Hierarchie durchlaufen werden, um wieder ein korrektes geometrisches Modell zu erhalten.

First-Cut /129/ besteht aus einem Modell und einem sogenannten Schlüsselwerkzeug. Es erlaubt das Zusammenarbeiten mehrerer Personen beim Konstruieren und Fertigen eines Produktes. Der Kern von Next-Cut ist die Wissensbasis. Sie besitzt neben ihrer eigentlichen Aufgabe, dem Abspeichern von Daten und Wissen, die Aufgabe, den Informationsaustausch zu regeln (Wer mit Wem ?). Der Modellierungsprozess des Systems basiert im wesentlichen auf der Verwendung von Features. Der Konstrukteur wird dadurch gezwungen, nur in Fertigungsverfahren zu denken. Die genutzten Features verlangen zudem ein tiefgreifendes Wissen der zugrunde liegenden Fertigungsprozesse. Erweiterungsbausteine erlauben First-Cut :

- den Fertigungsprozeß (incl. Zusammenbau u. Spritzgießen) zu unterstützen,
- frühe Phasen der Konstruktion zu unterstützen,
- Entscheidungen aufgrund physikalischer und geometrischer, Gegebenheiten zu treffen,
- Inkompatibilitäten zwischen Konstruktion und Fertigung zu erkennen.

First-Cut besteht insofern aus Modellen und Modulen.

Um zu kürzeren Produktentwicklungszeiten bei gleichzeitiger Absenkung der Kosten und Anhebung der Qualität zu kommen wurde die Philosophie des **Concurrent Engineering** entwickelt. Sie beschränkt sich nicht auf die einzelnen Entwicklungsphasen des Produktes sondern über den gesamten Lebenszyklus eines Produktes /99/. Der Kern des Concurrent Engineering besteht in der abteilungsübergreifenden, interdisziplinären Arbeitsweise. Um diese zu realisieren werden Arbeitsgruppen gegründet, deren Mitglieder aus den verschiedensten Bereichen wie z.B. der Fertigung, der Konstruktion, des Marketing und des Vertriebs stammen. Durch diese Zusammensetzung ist gewährleistet, dass die Auswirkungen von Entscheidungen in einer bestimmten Phase auf die nachfolgenden Phasen des Produktlebenszyklus erkannt wird. Das Problem, dass ein optimales Ergebnis eines Entwicklungsschrittes ein

nicht optimales Ergebnis der gesamten Entwicklung bewirken kann, wird vermieden. Im nachfolgendem Bild 1.7 sind die Bereiche des Concurrent Engineering sowie seine Interaktionen dargestellt.

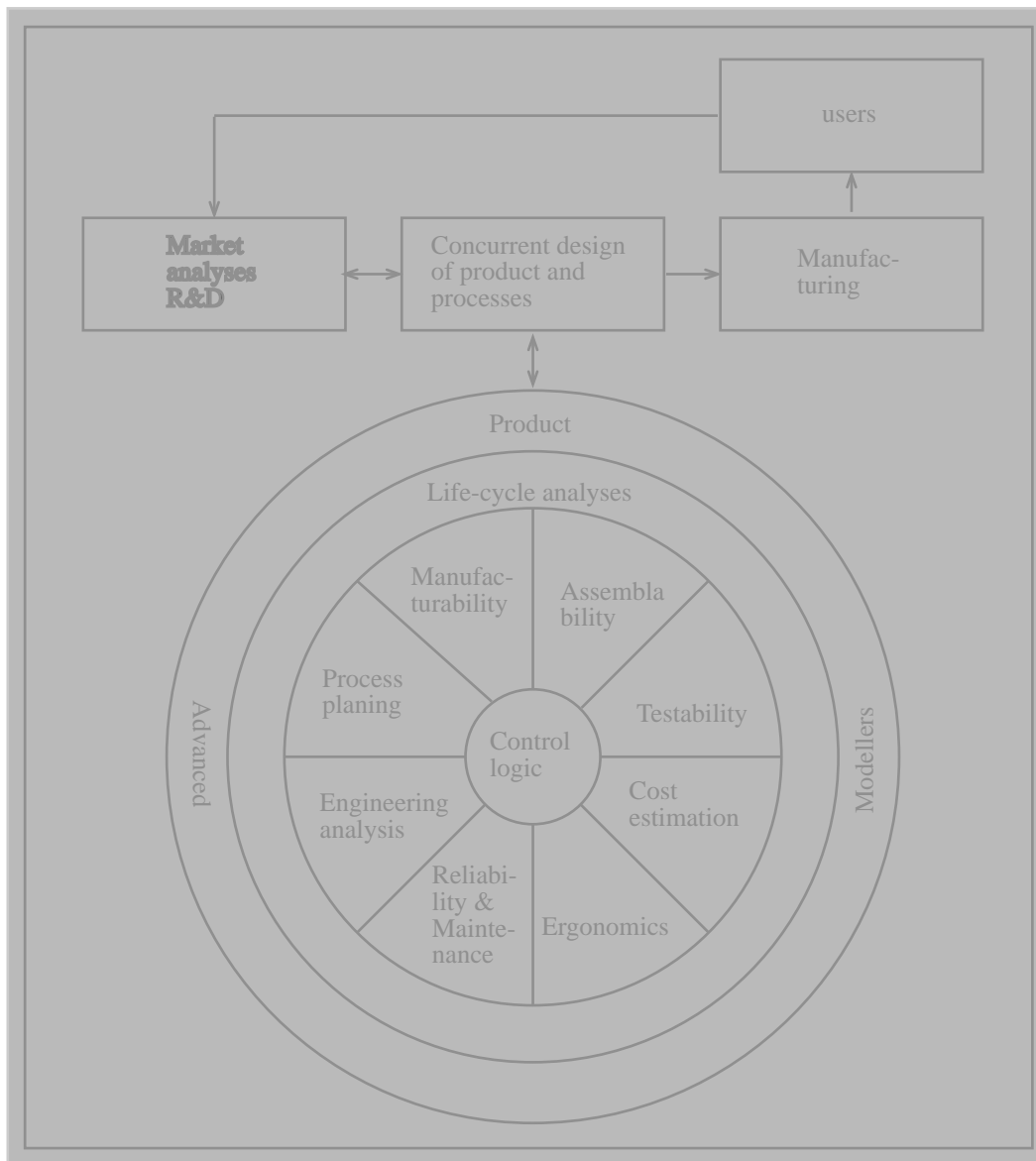


Bild 1.7 CE Wheel /116/

Unterstützung findet der Prozess des Concurrent Engineering durch die Methoden des **Wissensmanagements**. Diese stellen dem Produktentwicklungsprozeß /3 und 77/ durchgängig Wissen über alle dessen Bereiche zur Verfügung. Ein großer Nutzen entsteht beim Lösen von Problemstellungen durch die Möglichkeit, das Auffinden sowie die eingeschlagenen Lösungswege von ähnlichen Problemstellungen zu ermitteln.

In der Informationstechnologie sind die Bereiche der Geschäftsmodellierung, des computerunterstützten kooperativen Arbeitens, des Knowledge Engineering, der intelligenten Informationssysteme, des fallbasierten Schließens, des Data Warehousees, des Data Mining, des Information Retrieval, der Dokumentenmanagementsysteme der Mensch-Computer-Interaktion und der Agenten von Bedeutung. Das nachfolgende Bild 1.8 zeigt die primären Aufgaben des Wissensmanagement.

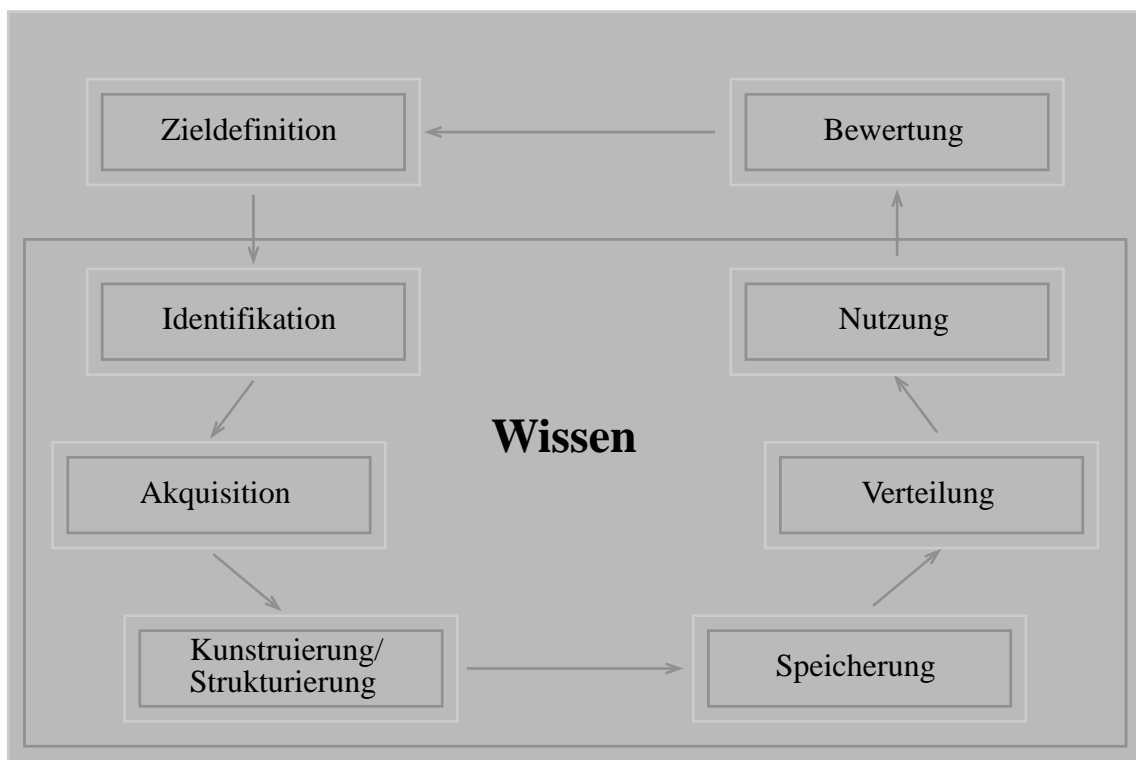


Bild 1.8 Primäre Aufgaben des Wissensmanagements /135/

Zunehmende Bedeutung erlangt das Wissensmanagement dadurch, dass Wissen als Wert der Produktentwicklung betrachtet wird. Wissen gilt als ein Erfolgsfaktor für Unternehmen. Dabei bewerten derzeit nur 20% die Nutzung ihres Wissens als gut /117/.

Das **ICAD**-System /101/ wurde spezifiziert, um den Entwicklungsprozeß zu automatisieren. ICAD basiert auf objektorientierten Technologien /33, 40, 118, 124, 128 und 150/. Mit ICAD besteht die Möglichkeit, den Entwurfsprozeß betreffende Produktdaten zu bearbeiten und Regeln zur Fertigung zu verwalten. Durch die generische Technik ist es möglich, eigene Prozesse in den Bereichen :

Datenein-/ausgabe, Datenkonvertierung, Berechnung, Produktkonfiguration, Verifikation der Herstellbarkeit, Interaktion und Datenaustausch mit externen Analysesystemen, Generierung der Materialkosten, Fertigungsinformationen und Zeichnungen zu automatisieren.

ICAD nutzt eine objektorientierte Programmiersprache, um dies zu erreichen. Die Modellierungsebene wird hierdurch abstrakter, gleichzeitig aber auch mächtiger. Durch Variation der Eingabeparameter können mit einem generischem ICAD-Modell durch die automatische Auswertung der Regeln sehr schnell Varianten von Produkten erzeugt und bewertet werden. Die als ICAD Design Language bezeichnete Sprache erlaubt es, Produktstrukturen zu definieren, Abhängigkeiten von Baugruppen zu generieren, Regeln des Zusammenbaus zu definieren, Fertigungsverfahren und Strategien zu erzeugen und auf die konventionellen Methoden der 2D und 3D Geometriemodellierung zuzugreifen.

Ein weiterer Baustein von ICAD stellt den Konstrukteuren Modellierungsoperationen zur Verfügung. Diese kann er nutzen, um regelbasierte Definitionen von komplexen Oberflächen, Körpern und Zeichnungen zu generieren. Zum kommunizieren mit dem Modell steht ein User-Interface zur Verfügung. Dies besteht aus den Subsystemen Browser, UI Toolkit, Kernel und GII. Der Browser dient zum Erzeugen eines Modells, über das UI Toolkit kann ein Anwender mit dem erstelltem Modell arbeiten. Der Kernel dient zur Steuerung des Modells durch einen PC im Batch-Betrieb. Über GII werden Daten mit CATIA ausgetauscht. Der Datenaustausch mit den Systemen AutoCAD, CADDs, Pro/Engineer und UG kann mit den Data Integration Tools durchgeführt werden.

WISKON /28/ wurde vom Fachgebiet Leichtbau-Konstruktion der Universität-Gesamthochschule Kassel im Jahre 1987 als eines der ersten Forschungsprojekte initiiert (**W**issensbasiertes **S**ystem für die mechanische **KON**struktion) und bis zum Jahre 1996 verfolgt. Eulenbach /28/ definiert die Zielsetzung des Systems wie folgt:

- **rechnergestützte Verfahren für die Konzept- und Entwurfsphase des Konstruktionsprozesses bereitzustellen,**
- **die Annäherung der Konstruktion an die Arbeitsplanung durch Nutzung einer ganzheitlich fertigungsorientierten Konstruktionsweise herbeizuführen und**
- **durch wissensbasierte Methoden die Entwicklung, die Konstruktion sowie die Arbeitsplanung zu unterstützen.**

In /28/ wurde das Grundkonzept für das Konstruktionssystem WISKON entwickelt. Die Hauptmodule des Systems wurden beschrieben. Kern des Systems ist der geometrische Modellierer GEOMOD. Er besteht aus einer hybriden Struktur, die sich aus einer Primitivkörperdarstellung und einer Randdarstellung zusammensetzt /11,12/. Sein API wurde darauf ausgelegt, von einem fertigungsorientierten Modellierer genutzt zu werden. Die rechnerinterne Darstellung wurde über das GKS-System visualisiert.

Auf dem GEOMOD-Kern setzt der fertigungsorientierte Modellierer FERMOD auf. Er erlaubt die ganzheitliche fertigungsorientierte Modellierung. Seine Implementation wurde prototypenhaft durchgeführt. Um FERMOD gruppiert sich TOPKON /140/. Es dient dazu, eine Funktions- und Strukturgliederung durchzuführen. Weiterhin kann mit ihm eine Produktbeschreibung durchgeführt werden. Allen Modulen übergeordnet ist WISMOD. Es ist eine in der Programmiersprache PROLOG ausgeführte Wissensverarbeitungs-komponente. Bild 1.9 zeigt die Struktur des Gesamtsystems WISKON.

Die für WISKON erarbeiteten Grundlagen wurden von dem konstruktionsmethodik- und wissensbasierten Konstruktionssystem WISKON/MA /67/ genutzt. Li stellte für die Phasen Planen, Konzipieren, Entwerfen und Ausarbeiten verschiedene rechnergestützte Hilfsmittel bei. Dies sind Optimierungsmethoden, die auf einer hybriden Wissensdarstellung des Konstruktionswissens aufsetzen. Sie wurden im Hinblick auf eine Integration in ein CAD-System entwickelt. Als wesentliches Ziel sieht Li die Steigerung der Entwurfsqualität bei gleichzeitiger Zeit- und Kosteneinsparung.

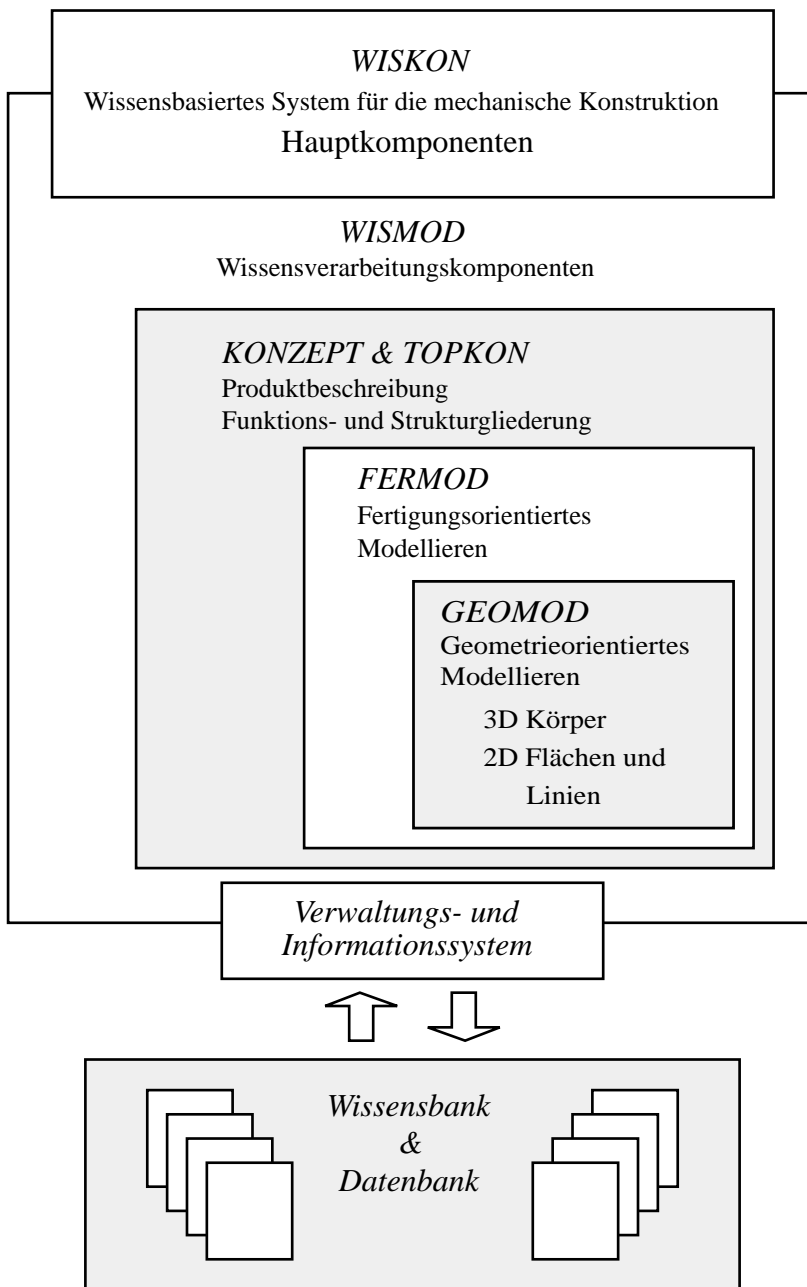


Bild 1.9 Hauptkomponenten des Konstruktionssystems WISKON /28/

Rembold gibt in /111/ einen weiteren Einblick über die Anwendung von Expertensystemen im Maschinenbau. Ein Projekt über die Entwicklung eines Expertensystems wird in /112/ dargestellt. Es soll vorhandene Planungsaufgaben, insbesondere die Erstellung von Arbeitsplänen für die Drehbearbeitung, aus dem Bereich der Fertigungstechnik durch Methoden der KI lösen. Um das Expertensystem zu erstellen wurde die KADS-Methode verwendet. Der expliziten Repräsentation des Expertenwissens wurde dabei besondere Beachtung geschenkt.

1.3 Zielsetzung und Gliederung der Arbeit

Die im Einsatz befindlichen CAD-Systeme unterstützen, wie einleitend bereits angedeutet, vornehmlich die Methoden der geometrischen Modellierung. Dies ist bei den derzeit gegebenen Anforderungen an die Konstruktion nicht mehr ausreichend. Zusätzliche Daten, Regeln und Vorschriften müssen berücksichtigt, weiterverarbeitet und dauerhaft zugänglich gemacht werden, um die Produktivität zu verbessern sowie dem Kostendruck auszuweichen. Dazu lassen sich außer dem geometrischen Modell weitere Modelle sowie deren Modellierungsmethoden ableiten.

Das Ziel der Arbeit besteht darin, ein fertigungsorientiertes Modellierungssystem auf der Basis von Standardsoftware zu entwickeln. Das fertigungsorientierte Modellieren steht hier stellvertretend für eine Reihe weiterer möglicher Modellierungsverfahren. Ausgewählt wurde der Bereich der Fertigung, da er traditionell eng mit der Konstruktionstechnik verbunden ist. Das Modellierungssystem soll es dem Konstrukteur auf einfachste Art und Weise ermöglichen, einen komplexen Modellaufbau problemorientiert durchzuführen. Zur Implementierung werden die folgenden Ziele verfolgt bzw. Arbeitsschritte durchgeführt:

Eine Analyse der Konstruktionssystematiken sowie deren Einfluß auf CAD-Systeme. Insbesondere sollen neben dem geometrischem weitere ableitbare Modelle ermittelt werden.

Die Ermittlung der zur fertigungsorientierten Modellierung geeigneten Verfahren wird in diesem Arbeitsschritt durchgeführt. Nicht alle Fertigungsverfahren eignen sich zur fertigungsorientierten Modellierung. Spanabhebende Verfahren, insbesondere mit geometrisch bestimmten Schneiden scheinen für die fertigungsorientierte Modellierung prädestiniert zu sein.

Die Ermittlung des prinzipiellen Systemaufbaus. Zur Zeit existieren verschiedene Möglichkeiten zur Realisierung eines CAD-Systems bzw. zur Integration von Daten und Methoden in ein CAD-System. Es soll die Einbringung von Fertigungsdaten in ein CAD-System unter-

sucht werden. Neben dem geometrischen und dem fertigungstechnischen Modell sollen das Kosten- und das Technologiemo- dell eines integrativen Produktmodells mit Daten versorgt bzw. deren Einbringung vorgesehen werden.

Die exemplarische Implementierung eines fertigungsorientierten Modellersystems. Dabei wird der Konstrukteur in die Lage versetzt, seine Überlegungen bezüglich der Fertigung des Bauteils im CAD-System abzulegen. Das CAD-System ist dann in der Lage, in etwa die Herstellkosten des Bauteils zu berechnen. Damit erhält der Konstrukteur ein Bewußtsein für das Kostenverhalten des Fertigungsprozesses. Der Fertigungsplaner kann auf diesen grundsätzlichen Überlegungen aufbauen. Er ergänzt die Informationen des Konstrukteurs wie z.B. die Verfahrswege, Oberflächenqualitäten und Toleranzen um technologische Daten wie die Spannkraft, die Frässtrategien oder die Schnittgeschwindigkeit.

Das entwickelte System besitzt neben seinen technischen Möglichkeiten auch Auswirkungen auf die Arbeitsweise und Arbeitsinhalte des Konstrukteurs und des Fertigungsplaners. Letztgenannte sollen in wesentlich früheren Stadien der Konstruktion in den Entwicklungsprozess eingebunden werden. Eine fertigungsgerechte Konstruktion soll von Beginn an unterstützt werden. Das noch weit verbreitete „Abteilungs“-System, d.h. eine Trennung von Konstruktion und Fertigung soll in ein konstruktives Miteinander übergehen.

2 Die Konstruktionsmethodiken

Aufgrund steigender Produktkomplexität und erweiterter Produkthanforderungen haben sich die Konstruktionsmethoden von einfachen Methoden hin zu umfangreichen Vorgehensmodellen entwickelt. Während in den Anfangszeiten der technischen Entwicklung einzelne herausragende Persönlichkeiten Meilensteine der technischen Entwicklung gesetzt haben, so sind dies seit Anfang des vergangenen Jahrhunderts ganze Ingenieurteams. Dies begründet sich darin, dass an technische Produkte zunehmend höhere Anforderungen gestellt wurden, die nur noch durch mechatronische, hydraulische oder pneumatische Komponenten befriedigt werden konnten. Darüber hinaus fanden z.B. Anforderungen wie die Recyclingfähigkeit sowie die Umweltverträglichkeit erst in den letzten Jahren Einfluß auf die Produktentwicklung. Um der immer verflochteneren und vielfätigeren Thematik der Produktentwicklung Herr zu werden, mußte eine methodische Vorgehensweise entwickelt werden.

Bei Erkens treten Mitte der 20er Jahre die ersten methodischen Gesichtspunkte des Konstruierens auf. Stichworte wie „schrittweises Vorgehen“, „stetiges Prüfen und Abwägen“ und „Ausgleich gegensätzlicher Forderungen“ werden begründet. Wögerbauer wird im allgemeinen als Begründer des methodischen Konstruierens gesehen. Er unterteilt die Gesamtaufgabe in Teilaufgaben. Diese wiederum werden in Betriebs- und Verwirklichungsaufgaben eingeteilt. Die Lösungen werden noch nicht systematisch ermittelt, sondern aus intuitiv gefunden Lösungen variiert. Es muß die erhaltene Lösungsvielfalt eingeschränkt werden. Wögerbauer erreicht dies durch das Prüfen und Bewerten. Merkmallisten entstehen, die bei weiteren Entwicklungen als Grundlage für Prüf- und Bewertungskriterien herangezogen werden.

Kesselring (Das Prinzip der minimalen Herstellkosten - Sparbau, das Prinzip vom minimalen Raumbedarf, das Prinzip vom minimalen Gewicht - Leichtbau, das Prinzip von den minimalen Verlusten, das Prinzip von der günstigen Handhabung), ebenso veröffentlichten Tschochner, Niemann, Matousek und Leyer wertvolle Ansätze zur Konstruktionssystematik, die auch heute noch Einfluß auf die einzelnen Arbeitsschritte des methodischen Konstruierens haben.

2.1 Anforderungen an die Konstruktion

Die Konstruktionsmethodik ist die Grundlage zur Entwicklung wettbewerbsfähiger Produkte geworden. In einem immer härter werdendem globalen Konkurrenzkampf sind selbst kleinste Mängel von enormer Wichtigkeit. Der Erfindergeist alleine reicht bei weitem nicht mehr aus, um konkurrenzfähige Produkte zu kreieren. Vielmehr müssen alle Forderungen an ein zu erstellendes Produkt systematisch abgearbeitet werden. Dabei ist es wichtig, alle Ergebnisse und Entscheidungen des Lösungsweges zu dokumentieren und nachvollziehbar zu gestalten welches auch eine Forderung der DIN ISO 9001 ist. Die Wichtigkeit der systematischen Vorgehensweise erkennt man an der Vielzahl und Unterschiedlichkeit der Anforderungen. Koller unterteilt die Forderungen in die Gruppen der „marktbedingten Forderungen“, der „umwelt- und gesellschaftsbedingten Forderungen“ sowie der „Erzeugungs-, Vertriebs- u. a. systembedingten Forderungen“. Die Gruppe der marktbedingten Forderungen umfaßt unter anderem die Eigenschaften wie die Lebensdauer, die Zuverlässigkeit, das Gewicht und das Design. Eigenschaften wie die „Lärm- und Schadstoffemission“, „Temperatur“, „Erschütterungen und Schwingungen“ sowie der Einfluß des „Menschen“ gehören zur Gruppe der Umwelt- und gesellschaftsbedingten Forderungen. Die Gruppe der Erzeugungs-, Vertriebs- u. a. systembedingter Forderungen umfaßt das „Entwickeln“, „Fertigen“, die „Montage“, „Lagerung und Transport“, „Vertrieb“, „Wartung“, „Reparatur“, „Recycling“ und „Beseitigung“. Im nachfolgenden Bild 2.1 sind zusätzlich zu diesen Forderungen die betriebsinternen Anforderungen und Schnittstellen an die Konstruktion dargestellt:

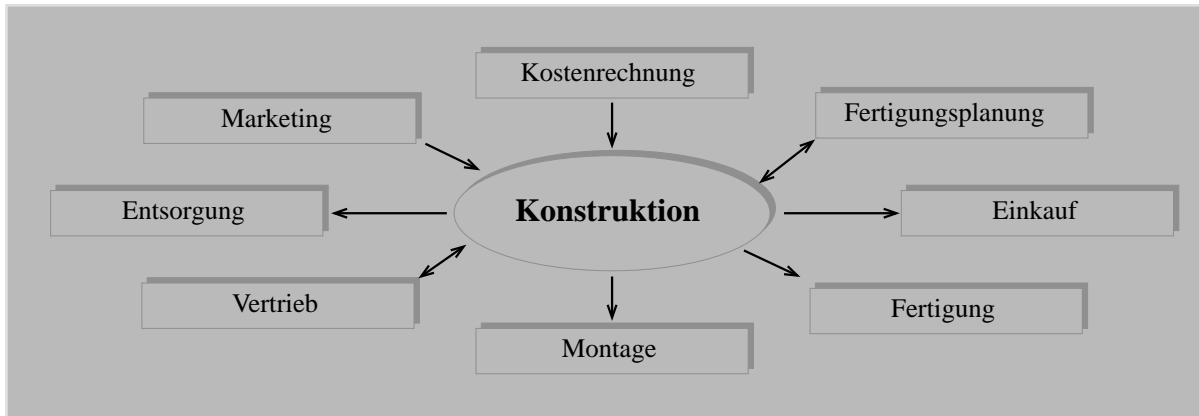


Bild 2.1 Einordnung der Konstruktion in die betriebliche Organisation /132/

Um diese Anforderungen erfassen zu können und ihnen gerecht zu werden, benötigt der Konstrukteur eine geeignete Methodik. Diese soll ihn einerseits in seinem Denken nicht so sehr einschränken, daß er sich mehr mit der Methodik als mit der eigentlichen Konstruktion beschäftigt, andererseits soll sein Handeln in einer nachvollziehbaren, geordneten Weise ablaufen. Derzeit gibt es vier wesentliche Anleitungen die sich mit dieser Thematik befassen. Neben den VDI-Richtlinien haben G. Pahl/W. Beitz, R. Koller, Rodenacker und K. Roth entsprechende Arbeiten verfaßt.

2.1.1 Die VDI-Richtlinien 2221, 2222 und 2223

Der Entwicklungs- und Konstruktionsprozeß hat einen wesentlichen Einfluß auf die Herstellung technischer Produkte. Dieser Prozeß ist geprägt durch eine große Vielfalt zu lösender Aufgaben, unternehmensspezifischer Bedingungen sowie marktseitiger und arbeitstechnischer Entwicklungstrends /VDI 2221/. Um diese Aufgabe lösen zu können bedarf es einer allgemeinen methodischen Arbeitsanleitung. Der VDI hat hierzu Anleitungen in Form von Richtlinien herausgegeben.

Wie in Bild 2.2 dargestellt ist die Richtlinie 2221 von einem übergeordnetem Charakter. Die auf ihr basierenden Richtlinien 2220, 2222 und 2223 behandeln jeweils Teilaspekte von ihr in einer ausführlichen Weise.

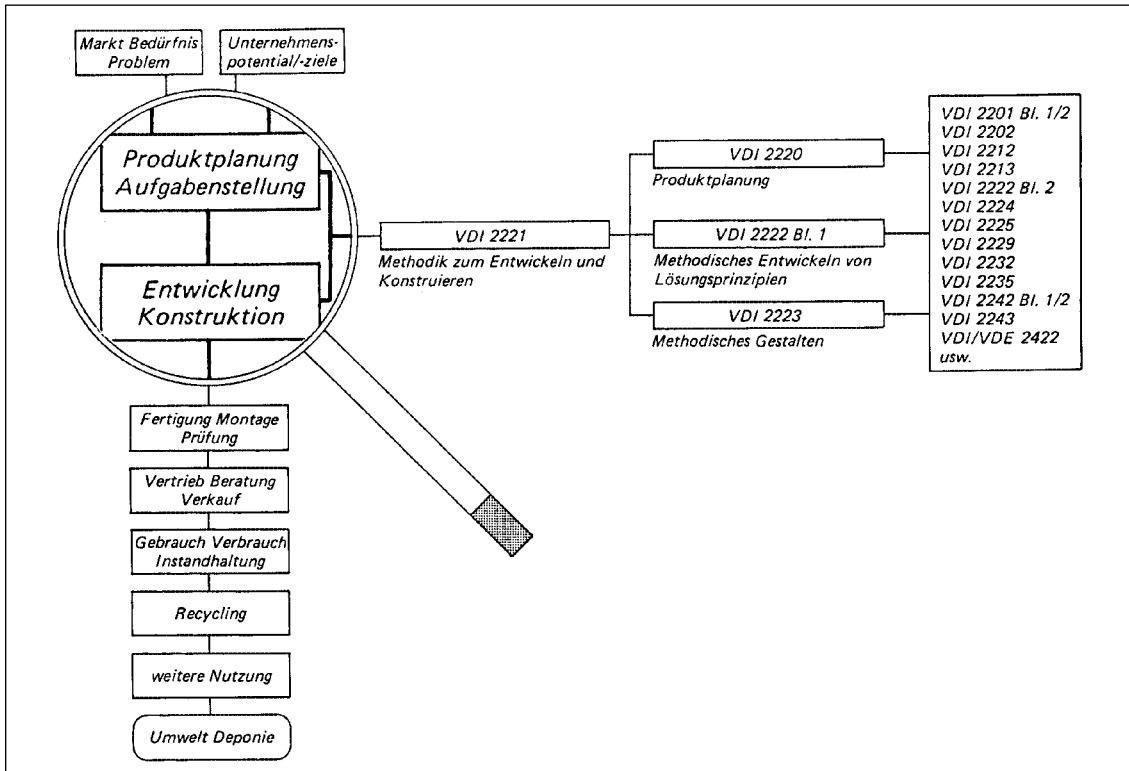


Bild 2.2 VDI Richtliniengerüst /147/

Die VDI 2221 definiert Arbeitsschritte und Arbeitsergebnisse die einer generellen Logik zur Durchführung des Prozesses der Findung der Aufgabenstellung sowie des Prozesses der Entwicklung und Konstruktion entsprechen. Sie dienen als Richtlinie im Arbeitseinsatz. Weitere untergeordnete Ziele der VDI 2221 sind:

- vorhandene Entwicklungs- und Konstruktionsabläufe inhaltlich und organisatorisch zu überdenken,
- die Datenverarbeitung optimal einzusetzen,
- den Einsatz der Datenverarbeitung beim Entwicklungs- und Konstruktionsprozeß in ein logisches Vorgehen zu integrieren,
- die entstandene Vielfalt von Methoden zur Durchführung einzelner Entwicklungs- und Konstruktionsschritte zusammenzustellen und zu ordnen,
- eine Begriffsdefinition durchzuführen,
- auf diese Richtlinie aufbauend sind weitere Richtlinien für spezielle Einsatzbereiche zu erstellen.

Basis der VDI 2221 ist die Systemtechnik. Sie definiert den Problemlösungsprozeß als Zusammenhang zwischen Zielsetzung, Planung, Durchführung, Kontrolle und Entscheidungen. Weitere definierte Begriffe sind die Problemanalyse, die Problemformulierung, die Systemsynthese, die Systemanalyse sowie die Beurteilung und Entscheidung. Aus diesen systemtechnischen Begriffen wurde ein allgemeiner Vorgehensplan für die Entwicklungs- und Konstruktionsprozesse abgeleitet.

Ausgangspunkt eines solchen Prozesses ist eine Aufgabenstellung. Sie liegt in Form der unternehmensinternen Produktplanung, der vom Kunden definierten Pflichtenhefte oder der innerbetrieblichen Konstruktionsaufträge für Betriebseinrichtungen oder Fertigungsvorrichtungen vor. Ein weiteres Bestimmungsmerkmal der Aufgabenstellung ist die Fertigungsart. Es muß zwischen Einzel- und Serien- bzw. Massenfertigung unterschieden werden. Neuentwicklung, Anpassung oder Weiterentwicklung sind ein weiteres Entscheidungskriterium. Diese Anforderungen enthalten bestimmte zu erreichende Ziele. Neben diesen Anforderungen existieren allgemeine Zwänge, die den Konstruktionsprozeß beeinflussen. Die VDI 2221 definiert die folgenden :

- die Wettbewerbssituation,
- den Kostendruck,
- den Termindruck,
- die Sonderwünsche,
- die Vorschriftenvielfalt,
- die Fremdfertigung,
- gestiegene Anforderungen und Komplexität der Produkte,
- das Eindringen der Mikroelektronik,
- den großen Forschungs- und Entwicklungsfortschritt.

Aus diesen Gegebenheiten wurde ein generell anwendbares Vorgehen beim Entwickeln und Konstruieren hergeleitet. Es besteht aus den in Bild 2.3 dargestellten sieben Schritten und den jeweiligen Ergebnissen dieser Arbeitsschritte. Alle Arbeitsschritte können zu jeweils mehreren Ergebnissen führen. Diese müssen gegeneinander abgewogen und bewertet werden. Mittels gezielter Optimierungsschritte kann ein Gesamtoptimum ermittelt werden.

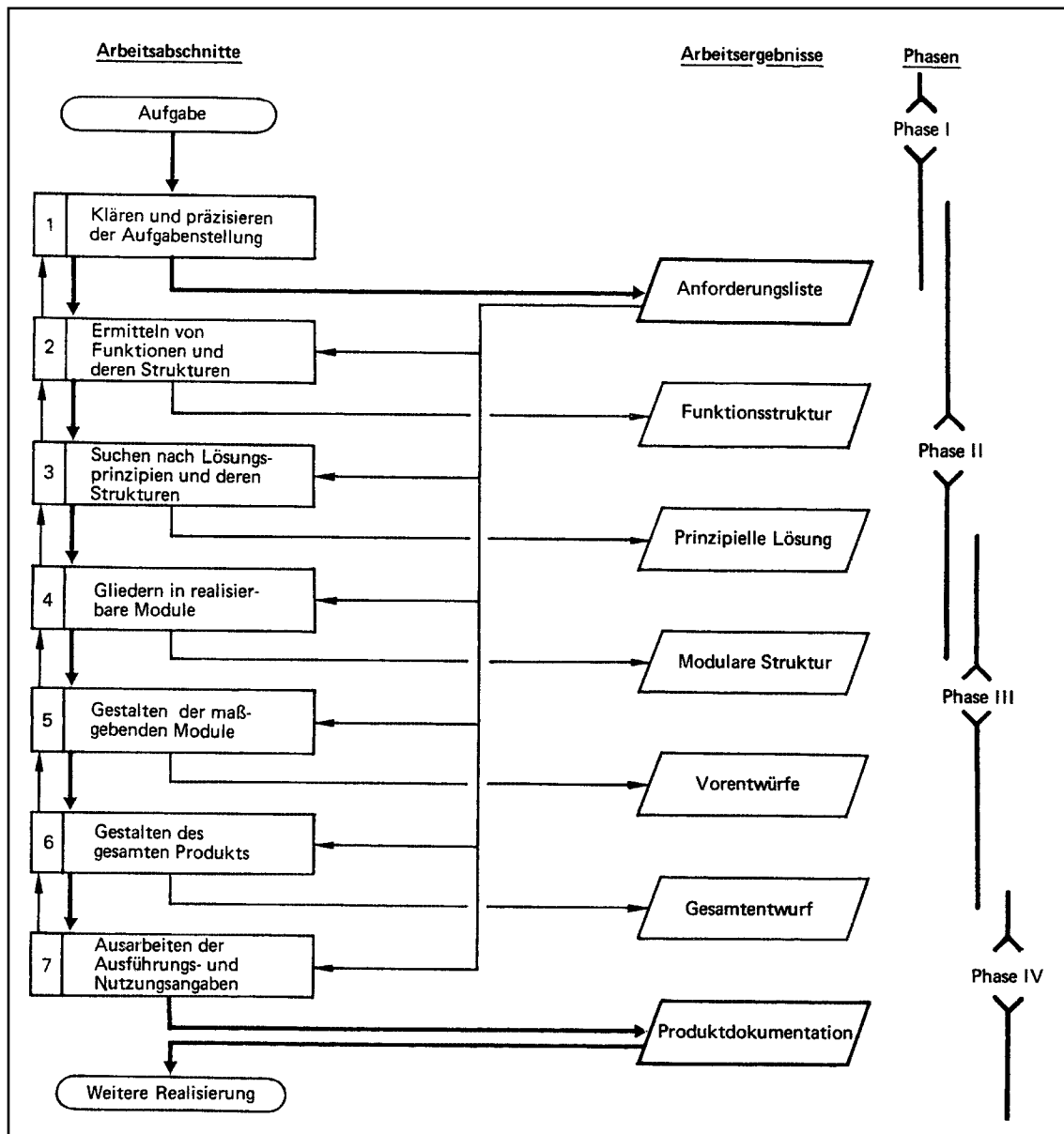


Bild 2.3 Generelles Vorgehen beim Entwickeln und Konstruieren /VDI 2221 /

Die Richtlinie VDI 2222 ist eine Vertiefung der ihr übergeordneten Richtlinie VDI 2221. Die ersten drei Themengebiete der VDI 2221, das „Klären und Präzisieren der Aufgabenstellung“, „das Ermitteln von Funktionen und deren Struktur“ sowie „Die Suche nach Lösungsprinzipien und deren Struktur“ werden vertiefend behandelt. Ausgehend von einer Entwicklungs- oder Konstruktionsaufgabe soll eine prinzipielle Lösung erarbeitet und dokumentiert werden. Besondere Aufmerksamkeit wird der systematischen Funktionsauswahl und der Variation der unterschiedlichen physikalischen Prinzipien gewidmet.

Das Blatt 2 der VDI-Richtlinie 2222 behandelt die Methodik des Erstellens und Anwendens von Konstruktionskatalogen. Davon ausgehend, daß einmal erarbeitetes Wissen für weitere Konstruktionstätigkeiten nur selten nutzbringend verfügbar gemacht wird, werden die Nutzung von Konstruktionskatalogen als auch ihre Erstellung als probates Mittel gesehen, um dieses Wissen verfügbar zu halten. Da die Informationstätigkeit bis zu 15 % /VDI 2222 Blatt 2/ der Konstruktionstätigkeit beansprucht, können Konstruktionskataloge einen wesentlichen Rationalisierungseffekt bewirken. Konstruktionskataloge in Form von Wälzlagerkatalogen, Tabellenbüchern, Normteilkatalogen und Wiederholteilkatalogen sind schon seit längerem im Einsatz. In der VDI 2222 Blatt 2 werden die Konstruktionskataloge in drei Kategorien eingeteilt :

- **Objektkataloge** sind /VDI 22 Blatt 2/ Konstruktionskataloge, die aufgabenunabhängig die für das Konstruieren notwendigen grundlegenden allgemeinen Sachverhalte, insbesondere physikalischer, geometrischer, technologischer und stoffkundlicher Natur, enthalten. Daten, wie die Oberflächen und Volumina wichtiger Körper, die physikalischen und technologischen Eigenschaften aber auch Lieferdaten von Profilen und physikalischer Effekte. Objektkataloge enthalten Informationen, die der Konstrukteur unabhängig von der Lösung eines bestimmten Problems benötigt.
- **Operationskataloge** sind /VDI 2222 Blatt 2/ Konstruktionskataloge, die Operationen (Verfahrensschritte) oder Operationsfolgen (Verfahren), die im Rahmen des methodischen Konstruierens von Interesse sind, und deren Anwendungsbedingungen und Einsatzkriterien enthalten. Demnach sind z. B. die Regeln zur Erzeugung verschiedener Konstruktionslösungen sowie Verfahren zur Lösung mathematischer und mechanischer Problemstellungen in Operationskatalogen enthalten.
- **Lösungskataloge** sind /VDI 2222 Blatt 2/ Konstruktionskataloge, die bestimmten Funktionen oder Aufgaben, ggf. ergänzt durch Randbedingungen, Lösungen zuordnen. Der Ordnungsgesichtspunkt für Lösungskataloge ist eine Funktion oder eine Klasse von Funktionen, die mit verschiedenen Mitteln realisiert werden kann.

Die drei Katalogarten unterstützen den Konstrukteur in allen Phasen seiner Arbeit. Die drei wesentlichen Ziele dieser Richtlinie bestehen darin, die Gestaltung und Erstellung von Katalogen zu beschreiben, das methodische Arbeiten mit Katalogen darzustellen sowie vorhandene Kataloge nutzbar zu machen. Die VDI Richtlinie 2223 besteht derzeit in der Form eines Entwurfes. Ihr Inhalt betrifft das Entwerfen von Produkten. Damit ist sie der VDI 2221 untergeordnet bzw. es wird in ihr der Bereich des Entwerfens ausführlicher spezifiziert. Dieser umfaßt die Aufgaben des „Gliederns in realisierbare Module“, das „Gestalten der maßgebenden Module“ sowie das „Gestalten des gesamten Produkts“. Diese drei Aufgaben sind in der Phase III des Entwicklungsprozesses zusammengefaßt und werden als Entwurfsprozeß bezeichnet. Der Entwurfsprozeß wird als ein methodisch durchführbarer Arbeitsprozeß definiert, der das Gestalten sowie das Planen, Steuern und Überwachen des Gestaltungsprozesses umfaßt. Das Gestalten besteht darin, die Gestalt- und Werkstoffeigenschaften von Gestaltungselementen festzulegen. Gestaltungselemente sind hierbei nicht immer nur Einzelteile und einzelne Flächen sondern auch Formelemente und Teilverbände. Die Begriffe Formelemente und Teilverbände sind in der Praxis gängige, aber nicht einheitlich benutzte Begriffe. Wird die Gestaltung auf der Basis funktionsrelevanter Gestaltungselemente durchgeführt werden oft die Begriffe Wirkfläche und Wirkraum verwendet. Ist die Komplexität eines Produktes das hervorstechende Merkmal, so wird oft in Baugruppen, Maschinen, Teilanlagen und Anlagen eingeteilt. Spielen Montage-, Vertriebs-, Gebrauchs- und Wartungseigenschaften eine Rolle, so wird von Montagegruppen, Zulieferkomponenten, Funktions- und Serviceeinheiten gesprochen. Eine einheitliche Definition des Begriffs Gestalt ist bisher nicht getroffen worden. Aufgrund der sehr unterschiedlichen Nutzungsbereiche werden sich immer wieder neue Definitionen bilden. Werkstoffeigenschaften bestehen aus der Werkstoffart, der Werkstoffbehandlung, der Werkstoffoberfläche und dem Werkstoffverbund /VDI 2223/. Die Dokumentation der gestalteten Elemente wird vom Konstrukteur in Zeichnungen und Stücklisten vorgenommen. Hat der Konstrukteur dies durchgeführt, so hat er einige weitere Produkteigenschaften festgelegt. Diese betreffen z.B. die Funktion, die Kosten, das Design und die Handhabung. Diese Eigenschaften werden auch als indirekt festgelegte Produkteigenschaften /DIN 2300/ bezeichnet. Bei der im nachfolgenden Bild gezeigten Querstiftverbindung verdeutlicht sich der Zusammenhang zwischen direkt und indirekt festgelegten Produkteigenschaften.

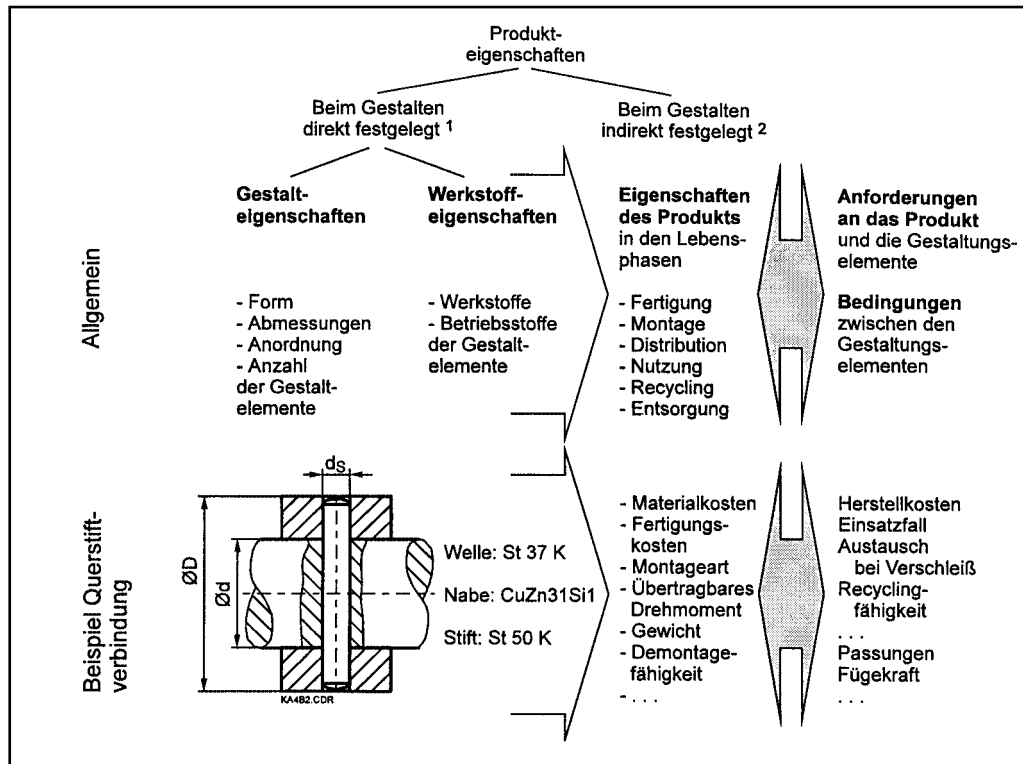


Bild 2.4 Direkt und indirekt festgelegte Produkteigenschaften / VDI 2223/

Die Abhängigkeit der Herstellkosten von den Material- und Fertigungskosten ist beispielsweise direkt ersichtlich. Zur Durchführung des Gestaltungsprozesses nutzt der Konstrukteur die Tätigkeiten des Analysierens, des Variierens, des Berechnens, des Experimentierens und des Beurteilens und Entscheidens. In diesen Tätigkeitsphasen kann er auf verschiedene Modelle, Methoden und Werkzeuge zugreifen. Bei der Tätigkeit des Analysierens kann er beispielsweise auf die FMEA-Methode, die Fehlerbaumanalyse, diverse Standardberechnungsmethoden oder die FEM zugreifen.

Wie bereits am Anfang dieses Abschnittes erwähnt besteht der Entwurfsprozeß aus dem Gestaltungsprozeß sowie dessen Planung, Steuerung und Überwachung. Unter Betrachtung der Tätigkeiten des Gestaltungsprozesses ergeben sich die Anforderungen für den Entwurfsprozeß wie folgt:

Er muß eine multidisziplinäre Arbeitsmethodik und ein leistungsfähiges Planungs- und Steuerungssystem für Termine, Projektkosten und Kapazitäten in die Entwicklung integrieren / VDI 2223/.

Das Entwerfen wird in diesem Sinne planbar, gestaltbar und nachvollziehbar.

2.1.2 Der Konstruktionsprozeß nach Pahl und Beitz

Für die beiden Autoren besteht die wesentliche Aufgabe des Ingenieurs darin, technische Probleme mit Hilfe naturwissenschaftlicher Erkenntnisse zu lösen und sie unter den jeweils gegebenen Einschränkungen stofflicher, technologischer und wirtschaftlicher Art in optimaler Weise zu verwirklichen /9 und 104/. Diese schöpferisch geistige Tätigkeit beruht auf dem Grundwissen der Naturwissenschaften sowie der Fertigungstechnik und der Konstruktionslehre. Dabei ist die Lösungsfindung eine Optimierungsaufgabe unter Berücksichtigung verschiedenster Randbedingungen und Randbereiche. Aufgrund verschiedener Ausgangssituationen beim Auslösen eines Konstruktionsauftrages werden unterschiedliche Organisationsformen der Zusammensetzung und des Zusammenwirkens der beteiligten Abteilungen sowie unterschiedliche Konstruktionsarten definiert. Bei den Konstruktionsarten wird in Neukonstruktion, Anpassungskonstruktion sowie Variantenkonstruktion unterschieden. Hieraus wird ersichtlich, daß ein starres, schematisches Vorgehen beim Konstruieren nicht möglich ist. Zu schnell wechseln die Randbedingungen. Immer komplexer werdende Ansprüche und Einflüsse an die Konstruktionsaufgabe können nur mit einer Konstruktionsmethodik befriedigt werden. Diese soll /104/:

„Ein problemorientiertes Vorgehen ermöglichen, d.h. sie muß prinzipiell bei jeder konstruktiven Tätigkeit branchenunabhängig anwendbar sein, erfindungs- und erkenntnisfördernd sein, mit Begriffen, Methoden und Erkenntnissen anderer Disziplinen verträglich sein, Lösungen nicht zufallsbedingt erzeugen, Lösungen auf verwandte Aufgaben leicht übertragen lassen, geeignet sein für den Einsatz elektronischer Datenverarbeitungsanlagen, lehr- und erlernbar sein und den Erkenntnissen der Arbeitswissenschaft entsprechen.“

Um diesen Anforderungen gerecht zu werden, fügen sie den sieben Vorgehensstufen der VDI 2221 die notwendigen Entscheidungsschritten zu und ordnen dies den Konkretisierungsphasen Konzipieren, Entwerfen und Ausarbeiten zu. Das in Bild 2.5 gezeigte Schema entsteht.

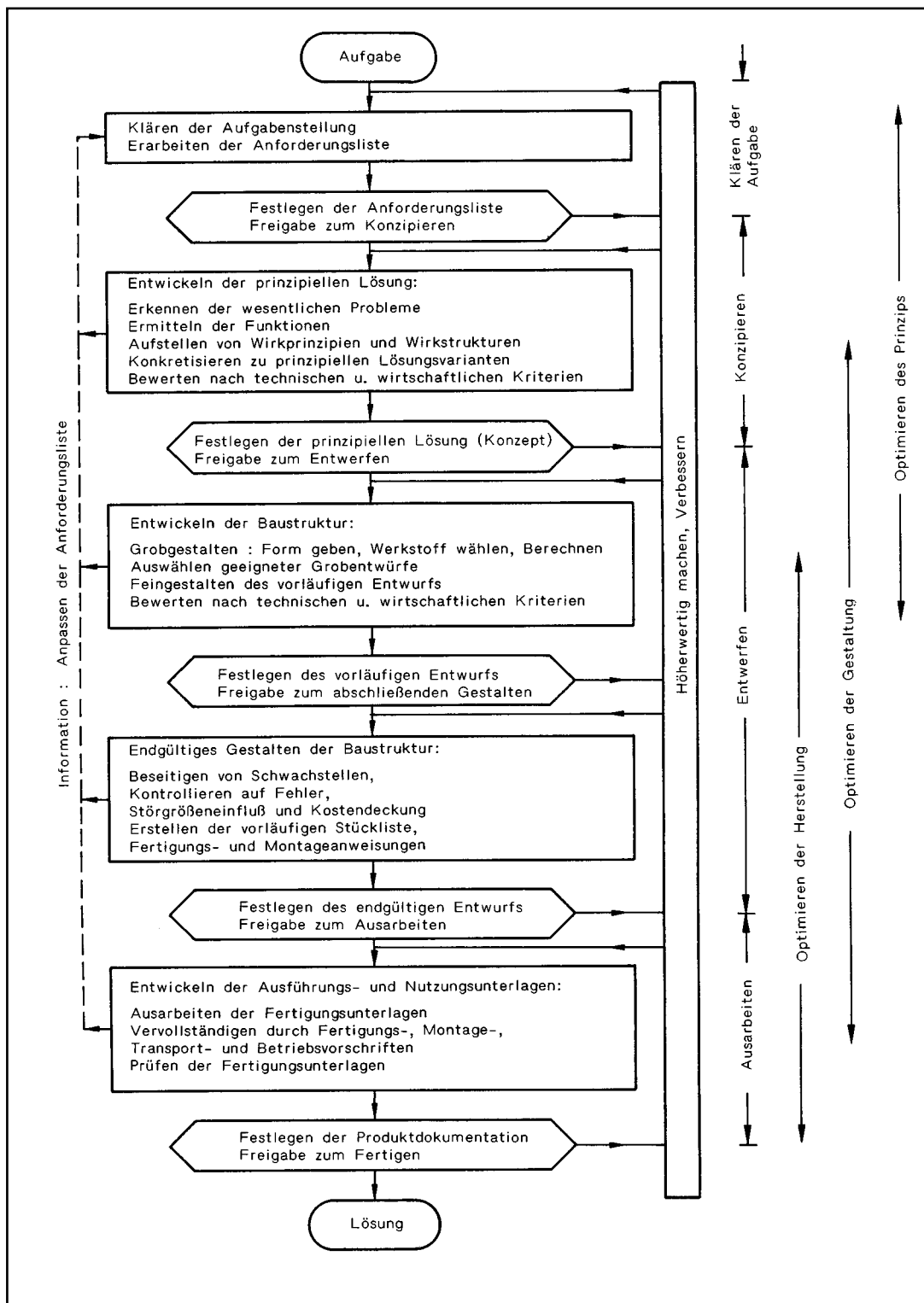


Bild 2.5 Arbeitsschritte beim Konstruieren /104/

Die einzelnen Schritte des Schemas werden in der Praxis mehr oder weniger bewußt durchlaufen. Ein unter Termindruck arbeitender Konstrukteur wird diese Schritte unbewußter abhandeln als ein Konstrukteur, der eine Neukonstruktion ohne großen Zeitdruck bearbeitet. Bei Optimierungsaufgaben einer Anpassungskonstruktion werden nicht alle Schritte durchlaufen. Diejenigen aber, die durchlaufen werden, werden mit größter Sorgfalt bearbeitet und gut dokumentiert.

2.1.3 Der Konstruktionsprozeß nach Koller

Für Koller besteht eine der Hauptaufgaben darin, die von der Unternehmensleitung, der Produktplanung, der Forschungsabteilung und weiteren Abteilungen vorgegebenen Daten in Konstruktionsergebnisse umzusetzen. Konstruktionsergebnisse sind vollständige Beschreibungen von Produkten. „Arbeitsvorbereitung, Fertigung und Montage“ produzieren diesen Beschreibungen entsprechende Produkte /59/.

Die Entwicklung einer allgemeinengültigen, produktneutralen Konstruktionslehre beginnt in den 60er Jahren. In die Wege geleitet haben dies einerseits ein immer größer werdendes Produktspektrum, andererseits das Aufkommen leistungsfähiger Computer.

Die Erforschung der Konstruktionsmethodik ist für Koller die wesentliche Voraussetzung zur Analyse, Beschreibung und Automatisierung von Konstruktionsprozessen. Durch sie werden qualitativ wesentlich bessere Produkte, schutzfähige Lösungen oder Umgehungen geschützter Lösungen /59/ geschaffen.

Beim Konstruieren unterscheidet er folgende Arten von Tätigkeiten bzw. Vorgängen:

- Synthesevorgänge, deren Ergebnisse alternative Lösungen sind,
- Analyse- oder Prüfvorgänge, welche die gefundenen Lösungsalternativen bezüglich gestellter Forderungen auf Brauchbarkeit prüfen,
- Bewerten und Selektieren (Ausscheiden) von weniger geeigneten Lösungen und Details,
- Verbessern nicht genügend tauglicher Lösungen oder Details durch erneute bzw. wiederholte Syntheseprozesse.

Die drei erstgenannten Schritte sind der Hauptbestandteil des Konstruktionsprozesses. Sie werden bei jedem Konstruktionsschritt durchlaufen. Konstruktionsschritte sind die Funktionssynthese mit dem Ergebnis der Funktionsstruktur, die Effektsynthese mit dem Ergebnis der Effektenstruktur, die Effekträgersynthese mit dem Ergebnis der Effekträgerstruktur, die qualitative Gestaltsynthese mit dem Ergebnis der qualitativen Entwürfe sowie die quantitative Gestaltsynthese mit dem Ergebnis der quantitativen Entwürfe.

Das nachfolgende Bild 2.6 veranschaulicht die einzelnen Prozeßschritte, ihre Ergebnisse sowie die einzelnen Entscheidungsmöglichkeiten nach jedem Konstruktionsschritt.

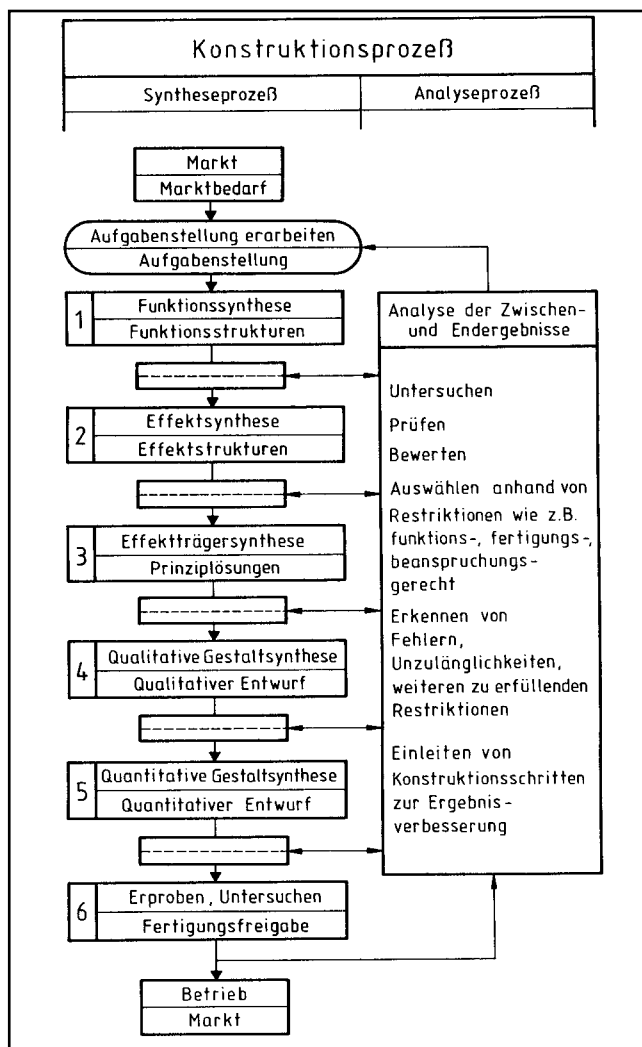


Bild 2.6 Konstruktionsschritte der verschiedenen „Konkretisierungsebenen“ /59/

Es ist das Ziel dieser Entscheidungsschritte Lösungen, die nicht zu einer optimalen Gesamtlösung führen können, auszuschließen. Oftmals müssen jedoch mehrere Lösungen, von denen

nicht vorhergesagt werden kann, ob sie zur optimalen Lösung führen, verfolgt werden. Koller unterscheidet nicht nach den in der Praxis üblichen unterschiedlichen Konstruktionsarten wie „Anpassungskonstruktion“, „Variantenkonstruktion“, „Neukonstruktion“, „Top-Down-Konstruktion“ und „Grob-Fein-Gestaltung“. Für ihn scheint es sinnvoller, danach zu unterscheiden, ob ein Konstruktionsprozeß bzw. dessen Beschreibungen sich für die Konstruktion beliebiger Produkte oder nur einer Produktart eignen. Es erfolgt eine Unterscheidung in produktneutrale bzw. produktunabhängige sowie produktabhängige bzw. produktspezifische Konstruktionen. Weiterhin unterscheidet Koller Konstruktionsprozesse die methodisch ablaufen und solche die intuitiv ablaufen.

Es findet eine weitere Einteilung in originäre oder primäre sowie in nachvollzogene oder sekundäre Prozesse statt. Die Bezeichnung originär verwendet er für Prozesse, die zum erstenmal für ein neu entstandenes Produkt abgelaufen sind. Sie ist weiterhin durch einen sehr hohen Anteil an festzulegenden Parameterwerten gekennzeichnet. Der sekundäre Konstruktionsprozeß legt nur noch relativ wenige Parameter fest. Er dient beispielsweise zur Erzeugung von Varianten.

2.1.4 Der Konstruktionsprozeß nach Roth

Der Schwerpunkt der Arbeiten von Roth /114/ besteht darin: „für die Konstruktion wichtige und immer wieder brauchbare Lösungen sowie Baugruppen und deren Variationsmöglichkeiten in übersichtlichen Tabellen, aber auch in Tafeln besonderer Art, den sogenannten Konstruktionskatalogen zusammenzufassen, nach einheitlichen Gesichtspunkten zu ordnen und nach spezifischen Gesichtspunkten greifbar zu machen“. Dabei müssen vielfältig verflochtene konstruktive Zusammenhänge in ihre Bestandteile zerlegt werden.

Die Zuordnung von benötigten Konstruktionsunterlagen zu Methoden oder Vorgehensweisen muß erfolgen. Dies kann nur mit einer methodischen Vorgehensweise durchgeführt werden. Sie gliedert sich in mehrere Teilschritte in denen die jeweiligen Konstruktionskataloge Verwendung finden:

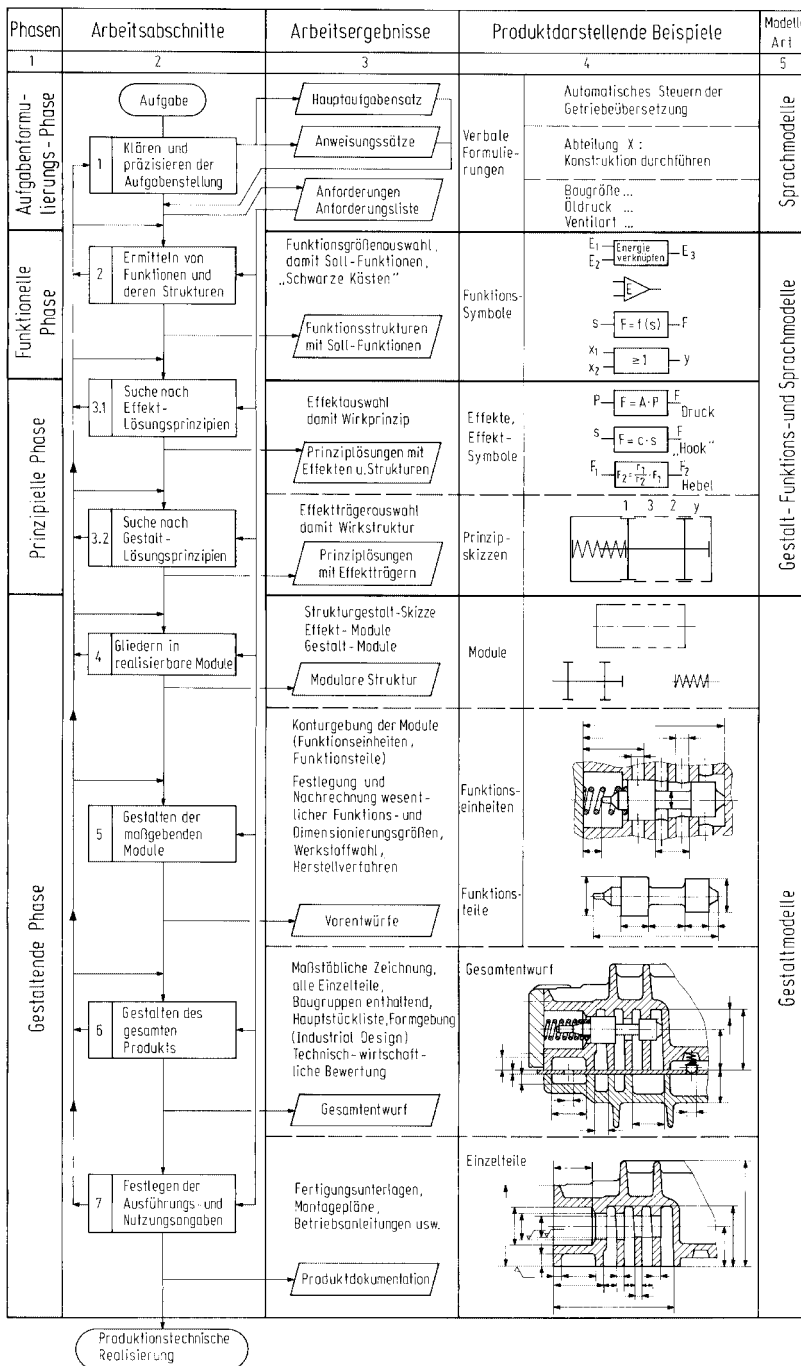


Bild 2.7 Die Konstruktionsmethodik nach Roth /114/

Das von Roth entwickelte algorithmische Auswahlverfahren zur Konstruktion mit Katalogen (AAK) stellt ein Beispiel bzw. einen Rahmen zur Anwendung der Kataloge und tabellarischen Sammlungen dar. Das Verfahren gliedert er zunächst in die Aufgabenformulierungsphase mit der Tätigkeit des Formulierens, die funktionelle Phase mit der Tätigkeit des Entwickelns, die

prinzipielle und die gestaltende Phase. Am Ende jeder Phase erfolgt eine Bewertung der erzielten Ergebnisse. Je nach Bewertung wird eine Phase nochmals durchlaufen. Am Anfang der Aufgabenformulierungsphase erfolgt die Erfassung von Konstruktionsdaten zunächst in verbaler und notizenhafter Form. Sie beinhaltet noch keine vollständige Erfassung aller relevanten Produktdaten. Am Ende dieser Phase sollten nach /114/ alle Tätigkeiten, Veranlassungen und Handlungen festliegen, die zur Erstellung der Konstruktions- und Fertigungsunterlagen notwendig sind. Weiterhin sollten alle chemischen, physikalischen und technischen Eigenschaften des Produktes vollständig definiert sein. Die Anforderungsliste, die zusätzliche, für das Produkt relevante Anforderungen enthält, sollte jetzt vollständig sein.

In der funktionalen Phase werden zuerst die Beziehungen zwischen Ein- und Ausgangsgrößen ermittelt. Eine Untergliederung der Gesamtfunktionalität in Teilfunktionalitäten erfolgt. Am Ende dieser Phase sollten alle noch effektenungebundenen Funktionen ermittelt sein. Ihre Funktionsstruktur sollte jetzt festgelegt werden. Die prinzipielle Phase ordnet den Funktionen reale Effekte (z.B. Hebel- und Keileffekt) zu. In dieser Phase werden zahlreiche Varianten durch Variation der Effekte als auch durch Variation der Geometrie entworfen und bewertet. Die gestaltende Phase unterteilt Roth in die geometrisch-stoffliche und die fertigungstechnische Produktgestaltung. Am Anfang dieser Phase steht eine aus den vorangegangenen Konstruktionsschritten erarbeitete Skizze. Diese wird nach geometrischen, dimensionierungs- und werkstofftechnischen Gesichtspunkten aus Körpern zusammengesetzt. Unter Berücksichtigung der Kosten sollten die Oberflächen möglichst einfach herstellbar sein. Montage und Fertigung nehmen hier ihren Einfluß, indem sie den Konstrukteur beispielsweise veranlassen Teile in mehrere Einzelteile aufzugliedern. Im Bild 2.7 sind diese Schritte, die daraus resultierenden Arbeitsergebnisse sowie zugehörige produktdarstellende Beispiele gezeigt.

Aus diesem für Neukonstruktionen gültigem Schema wurde ein weiteres abgeleitet, der variable Ablaufplan. Er trägt der Änderungs- und Variantenkonstruktion Rechnung und bietet in jeder Konstruktionsphase Einstiegsmöglichkeiten. Hierzu werden die Konstruktionsarten „Herstell-Einstiegs- bzw. Herstellungskonstruktion“, „Gestalt-Einstieg- bzw. Gestaltungskonstruktion“, „Prinzip-Einstiegs- bzw. Prinzipkonstruktion“, „Funktions-Einstieg- bzw. Funktionskonstruktion“ und „Neukonstruktion“ unterschieden.

2.2 Zusammenfassung der Konstruktionsmethoden

Die vorgestellten Konstruktionsmethoden besitzen die folgenden Gemeinsamkeiten: Allen Modellen ist gemein, daß sie in verschiedene Phasen unterteilt sind. Am Ende jeder Phase findet eine Bewertung der erzielten Ergebnisse statt. Aufgrund dieser erfolgt eine Wiederholung des vorangegangenen Schrittes, ein Weitergehen im Ablauf oder ein Abbruch des Prozesses.

Das Prinzip des „teile und herrsche“ wird von allen Methoden verwendet. Komplexe Aufgaben, Funktionen oder Maschinen werden in einfachere Teilaufgaben, Teilfunktionen, Baugruppen und Einzelteile aufgespalten. Für diese werden Lösungen gefunden, falls nicht bereits vorhanden. Diese werden zum Gesamtprodukt zusammengesetzt.

Die Generierung von Varianten ist ein weiteres Merkmal aller Konstruktionsmethoden. Der Konstrukteur sollte mehrere Alternativen einzelner Bauteile erzeugen. Die einzelnen Varianten werden gegeneinander abgewogen und bewertet. So sollen möglichst alle Lösungen erfaßt werden. Diese Methodik wird oft auch als „Konstruieren in die Breite“ bezeichnet.

Eine weitere Gemeinsamkeit aller Methoden ist der erste Schritt der jeweiligen Vorgehensweise. Er besteht in der Ermittlung der Aufgabenstellung. Welche Daten und Arbeitsschritte erfaßt und definiert werden müssen ist zu aufgabenspezifisch um sie detailliert in einer Konstruktionssystematik beschreiben zu können. Anhand von Beispielen wird ein Eindruck vermittelt, was in diesem ersten Konstruktionsschritt zu beachten ist.

Im zweiten Schritt der Konstruktionsmethodiken werden die prinzipiellen Lösungen ermittelt. Roth teilt diese Phase in drei Schritte ein, das Ermitteln der Funktionen und allgemeinen Funktionsstruktur, die Zuordnung von Effekten zu Funktionen sowie die Erstellung der Prinzipskizze. Die VDI 2221 spiegelt dieses Schema wieder. Bei Pahl/Beitz findet die typische Unterteilung in drei Arbeitsschritte nicht statt, der Inhalt ihres Konzeptionsschrittes enthält aber die Inhalte der drei einzelnen Schritte der VDI2221 bzw. der Systematik von Roth. Koller bezeichnet diesen Schritt als Funktionssynthese. Bei ihm findet die Zuordnung von Funktionen und Effekten in diesem Schritt noch nicht statt. Rodenacker entwickelt in dieser Phase zuerst den „logischen Wirkzusammenhang“, anschließend den „physikalischen Wirkzusammenhang“.

Ablaufplan nach						
Konstrukt-Phasen (Roth)	Roth	Pahl/Beitz	VDI -2221	Koller	Rodenacker	
	Aufgabe	Aufgabe	Aufgabe	Produktplanung	Aufgabe	
Aufgabenformulierung	Formulieren Klären d. Aufgabenstellung Hauptaufgabe Anweisung, Anforderungsliste	Klären der Aufgabe Klären der Aufgabenstellung Anforderungsliste	Formulieren der Aufgabe 1 Klären und präzisieren der Aufgabenstellung	Marktanalyse Erarbeiten der Aufgabenstellung	Geforderter Wirkzusammenhang	
	Funktion entwickeln Funktion ermitteln, Allgemeine, Logische Funktionsstruktur	Konzipieren Entwickeln der prinzipiellen Lösung Funktionen Wirkprinzipien Wirkstrukturen Prinzipielle Lösungen Varianten Technisch - wirtschaftliches Bewerten	Prinzip finden 2 Ermitteln von Funktionen und deren Strukturen 3.1 Suche nach Lösungsprinzipien Effektebene 3.2 Suche nach Lösungsprinzipien Gestaltenebene	Funktionssynthese Zweck- o. Hauptfunktionen, Gliedern der Teil- und Grundfunktionsstrukturen, Technisch - wirtschaftliches Bewerten	Funktion Logischer Wirkzusammenhang Physikalisches Geschehen Physikalischer Wirkzusammenhang	
Wirkprinzip	Prinziplösung entwickeln Funktionen mit Effekten, Prinzipiellösungen entwickeln, Spezielle Funktion					
	Effektträger zur Prinzipialskizze entwickeln Techn.-wirtschaftl. Bewert.			Qualitative Synthese Zuordnen und variieren von Effekten, Effektträger variieren, Prinzip darstellen, Auswählen der Lösungen für Gesamtkonzept, Gestalten, Entwerfen	Wirkort Kinematischer Wirkzusammenhang Konstruktiver Wirkzusammenhang	
Wirkstruktur						
				Quantitative Sythese Berechnen, Bemessen, Experimentelle Untersuchungen, Erprobung, Verbesserung, Detaillieren, Arbeitspläne erstellen Fertigungs- Montageunterlagen	Fertigungstechnischer Wirkzusammenhang	
Geometrisch- Stoffliche Gestaltende	Gestalten Struktur- und Formgestalten Strukturgestalt- Skizze Konturen und Querschnitte entwerfen Werkstoffe, Festigkeit, Montierbarkeit, Benennung Funktionsintegration Gesamtentwicklung Techn.- wirtschaftl. Bewert.	Entwerfen Baustruktur entwickeln Grobgestalten : Form, Werkstoff, Berechnen Feingestalten Techn.-wirtschaftl. Bewert. Baustruktur endgültig gestalten Schwachstellen, Störgrößen, Kostendeckung, Stücklisten, Fertigungsanweisungen	Gestalten 4 Gliedern in realisierbare Module 5 Gestalten der maßgebenden Module 6 Gestalten des gesamten Produkts			
	Fertigungsgestalten Schwachstellenanalyse Fertigungs- montage- transport-, recycling- gerecht usw. gestalten Endgültiger Entwurf, Detaillieren, Tolieren Herstellunterlagen, Montage-, Betriebs- Prüfvorschriften	Ausarbeiten Ausführungs- und Nutzungsunterlagen, Fertigungsunterlagen Montage-, Transport, Betriebs-, Prüfvorschriften	7 Festlegen der Ausführungs- und Nutzungsangaben			
Herstellungstechnische Gestaltende						
Produktdokumente						

Bild 2.8 Gegenüberstellung verschiedener Konstruktionsmethoden /114/

Den dritten Schritt, die gestaltende Phase, unterteilt Roth in die Unterschritte Gestalten und Fertigungsgestalten. Im ersten Unterschritt werden u.a. die Strukturgestalt, Skizzen, Konturen, Werkstoffe festgelegt. Beim Fertigungsgestalten findet u.a. eine Prüfung im Hinblick auf die Schwachstelle, Fertigungs-, Recycling- und Montagegerechtheit statt. Pahl/Beitz unterteilen diesen dritten Schritt in die beiden Schritte Entwerfen und Ausarbeiten. Die Inhalte dieser entsprechen denen Roths. Die VDI 2221 unterteilt die Phase des Gestaltens in vier Unterschritte mit wiederum demselben Inhalt wie die beiden vorgenannten Autoren.

Koller nutzt andere Elemente als auch eine unterschiedliche Bezeichnungsweise. Mit der Definition der „Qualitativen Synthese“ und der „Quantitativen Synthese“ beim Gestalten unterscheidet er sich. In der qualitativen Synthese erfolgt bei Koller das Variieren und Zuordnen von Effekten, die Darstellung des Prinzips, die Auswahl der Lösungen für das Gesamtkonzept sowie das Gestalten und Entwerfen. In der Phase der quantitativen Synthese erfolgen Berechnungen, Bemessungen, Erprobung und Versuch sowie das Detaillieren.

Bei Rodenacker steht der kinematische Wirkzusammenhang als erster Bearbeitungspunkt der gestaltenden Phase. Ihm folgt die Ermittlung des konstruktiven Wirkzusammenhangs. Den Abschluß bildet die Ermittlung der fertigungstechnischen Wirkzusammenhänge.

Bild 2.8 zeigt die Gemeinsamkeit und Unterschiede der verschiedenen Methoden in den verschiedenen Prozeßschritten. Die vorgestellten Konstruktionsmethoden unterscheiden sich mehr oder weniger stark voneinander. Eine „richtige“, für alle Probleme gültige Konstruktionsmethodik wird sich nicht entwickeln lassen. Zu unterschiedlich sind die Anforderungen. Einerseits werden die Anforderungen über die unterschiedlichsten Produktgruppen definiert, andererseits besitzen die Konstruktionsabteilungen bereits individuelle, vom Einsatzgebiet und Umfeld der Firma abhängige Erfahrungen. Darüberhinaus besitzt jeder Konstrukteur seine Erfahrungen und Vorlieben, die ihn die eine oder andere Konstruktionsmethodik bevorzugen lassen.

2.3 Ableitbare Modelle

Aus den vorgestellten Konstruktionsmethoden lassen sich für deren einzelne Phasen verschiedene Modelle ableiten. Diese Modelle wurden u. a. in /5, 28, 39, 83, 102 und 143/ definiert. Für die nachfolgenden Betrachtungen der Zuordnung dieser Modelle zu den einzelnen Konstruktionsphasen wurde die VDI-Richtlinie 2221 zugrunde gelegt. Anfangsschritt ist das *Klären und Präzisieren der Aufgabenstellung*. Dieser Arbeitsschritt wird durch ein Spezifikationsmodell repräsentiert. In ihm sind die Forderungen und Wünsche des Kunden, Einschränkungen des Herstellers, die Kennzeichnung der Konstruktionsart, die Sachmerkmale des Produktes sowie der Ablauf der Produktentwicklung enthalten.

Für die Phase des *Ermittelns von Funktionen und deren Struktur* existiert ein Funktionsmodell. In ihm werden die Struktur der Haupt-, Neben- und Hilfsfunktionen sowie die Wirkstrukturen

und die genutzten Wirkprinzipien ermittelt und abgelegt. Dieses Modell enthält auch noch Daten aus der Phase des *Suchens nach Lösungsprinzipien und deren Struktur* sowie der Phase des *Gliederns in realisierbare Module*. Der Phase des *Gestaltens der maßgebenden Module* und der Phase des *Gestaltens des gesamten Produkts* lassen sich mehrere Modelle zuordnen. Das Geometriemodell enthält die geometrische Beschreibung der Einzelteile des Produktes, der Abhängigkeiten geometrischer Grundelemente, der Anordnung der Einzelteile zueinander sowie die Erzeugnisgliederung und Stücklisten. Im technischen Gestaltmodell sind Gewinde, Verzahnungen und u.a. Passfedernuten. Daten über Werkstoffe und deren Behandlungszustand, Oberflächenqualitäten, Passungen und Toleranzen kennzeichnen das Technologiemodell.

Im Parametriemodell werden variable geometrische Größen, mathematische Abhängigkeiten der Parameter sowie Konstruktionslogiken in Form von Entscheidungstabellen und Regelbasen abgelegt. Die Phase des *Ausarbeitens der Ausführungs- und Nutzungsunterlagen* wird durch das Darstellungsmodell unterstützt. In ihm sind die Daten der erforderlichen Zeichnungsunterlagen, notwendige Schnitte und ihre Spezifikationsdaten und die Bemaßung enthalten. Weitere Unterstützung liefert das Berechnungsmodell. Es stellt die Daten der wesentlichen Auslegungsgrößen, die Abhängigkeiten zwischen den Auslegungsgrößen, die verwendeten Vorschriften, Formeln und Verfahren für die Auslegung, idealisierte Modelle für die Nachrechnung sowie charakteristische Berechnungsergebnisse zur Verfügung. Das Fertigungsmodell enthält die Daten der verwendeten Halbzeuge und Rohteile, der Kennung von Eigen-, Fremd-, Einzel- und Serienfertigung, der Fertigungselemente, der relevanten Zwischenzustände der Fertigung, die erforderlichen Fertigungsmaschinen, Werkzeuge und Spannmittel, erforderliche Betriebsmittel, kalkulierte Fertigungs- und Rüstzeiten sowie Fertigungs- und Montagelogik in Form von Entscheidungstabellen und Regelbasen. Ein weiteres Modell ist das Kostenmodell. Es enthält Informationen über die Kostenstruktur, die Materialkosten, die Herstellkosten sowie über die Grenzstückzahlen.

Die vorgestellten Daten beinhalten alle wesentlichen Informationen sowie Methoden, die in den Konstruktionsmethodiken Verwendung finden. Sie alle sind Teilmodelle des integrativen Produktmodells. Der Begriff des Produktmodells entstand aus der Notwendigkeit, Datenmodelle zu entwickeln und zu kennzeichnen, die während des ganzen Konstruktionsprozesses nutzbar sind. In der Literatur wurde der Begriff des „Phasenbezogenen Produktmodells“ sowie

des „Kohärenten Produktmodells“ entwickelt. Ersteres zeichnet sich dadurch aus, daß es aus verschiedenen Teilmodellen besteht. Letzteres bietet ein einheitliches, zentrales Datenmodell über alle Konstruktions- und Fertigungsphasen. Es besitzt den Vorteil, daß es phasenunabhängig ist und somit keine Datenverluste durch Modelltransformationen entstehen. Dies beinhaltet aber auch den Nachteil komplexer Datenstrukturen sowie großer Datenmengen. Die Nachteile des kohärenten Modells sind zugleich die Vorteile des phasenbezogenen Produktmodells. Es besitzt einen direkten Phasenbezug, geringe Datenmengen sowie einfache Datenstrukturen. Der Nachteil liegt im Datenverlust beim Transformieren der Daten über die einzelnen Phasenübergänge. Im Bild 2.9 ist das integrative Produktmodell sowie seine zugehörigen Partialmodelle dargestellt /28/.

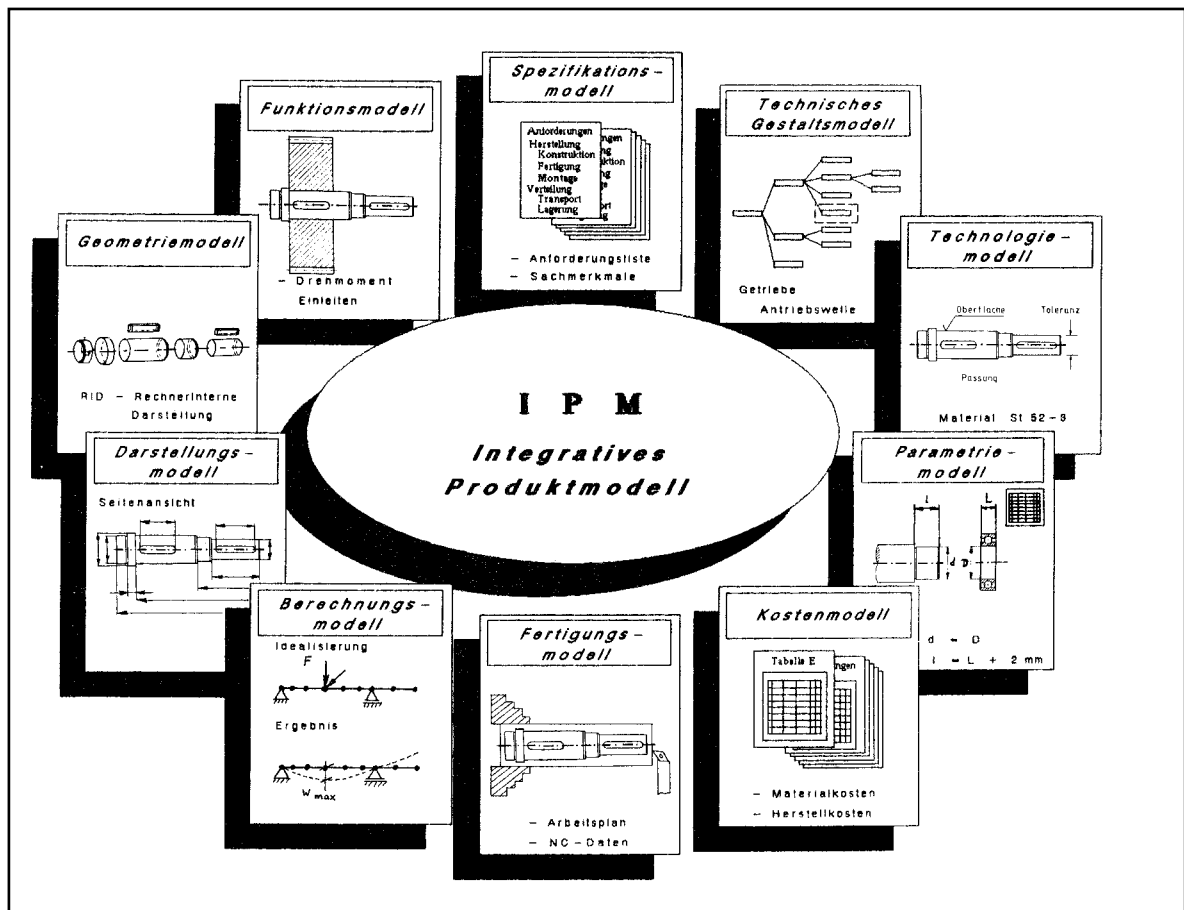


Bild 2.9 Teilmodelle des integrativen Produktmodells /28/

Die Daten der Teilmodelle müssen nicht notwendigerweise in einem einzelnen Konstruktions-schritt erarbeitet werden. Vielmehr werden die Teilmodelle während der einzelnen Konstrukti-

onsschritte mit den jeweils anfallenden Daten gefüllt. Eine vollständige Füllung sollte erst nach Ablauf des gesamten Konstruktionsprozesses erreicht sein.

2.4 Anforderungen an EDV-Systeme

Die Konstruktionslehre und die Konstruktionssystematik sind komplexe Themengebiete. Es existieren keine prinzipiellen Vorgehensweisen zur Entwicklung bzw. Weiterentwicklung von technischen Produkten. Es zeigt sich, daß in vielen Firmen unterschiedliche Konstruktionssystematiken vertreten werden. Fast jeder Konstrukteur hat eine eigene individuelle Vorgehensweise. Diese kann man mit einer mehr oder weniger großen Abweichung in die in den Kapiteln 2.1.1 bis 2.1.4 beschriebenen Vorgehensweisen eingeordnet werden. Aufgrund der vielfältigen Anforderungen /1, 34, 37, 41, 63, 75, 119, 122, 131, 152 und 154/ an eine EDV gestützte Konstruktionssystematik läßt sich erkennen, daß es kein universelles EDV-System /141/ geben wird, das allen Anforderungen genügen kann. Vielmehr wird es eine große Anzahl verschiedenster EDV-Systeme geben, die den Konstrukteur bei seiner Arbeit unterstützen. Derzeit verzeichnen EDM-Systeme stark steigende Verkaufszahlen, was einen großen Bedarf an Strukturierung und Organisation von verschiedenen Daten erkennen läßt.

In Bild 2.10 sind Tätigkeiten des Konstrukteurs sowie Rechnergestützte Hilfsmittel abgebildet. Hier finden sich auch die einzelnen Modelle des integrativen Produktmodells wieder. Wird pro Modul eine Datei eines Bauteils benötigt, so ist der steigende Bedarf an EDM-Systemen erklärbar. Aus der vorangegangenen Diskussion der Konstruktionsmethodik zeigen sich allerdings noch weitere Anforderungen, die nicht nur Modelle sondern auch Prozesse betreffen.

Die VDI 2221 stellt die Forderung, die Möglichkeiten der Datenverarbeitung optimal einzusetzen. Eine durchgängige integrierende Datenverarbeitung im Gesamtunternehmen soll ermöglicht werden.

Eine der Anforderungen aus der VDI 2222 Blatt 1 lautet:

In der VDI 2222 werden Konstruktionsmodelle beschrieben, die das Wissen so strukturieren und ordnen, daß eine zukünftige Übertragung auf den Rechner möglich und sinnvoll ist. Weiterhin soll eine Unterstützung der Variation von Lösungen stattfinden. Ein weiterer Wunsch ist, eine skizzenhafte Beschreibung von Modellen zu ermöglichen. Eine „orientierende Berechnung“, um Lösungsvarianten auszuschließen wird gefordert. Eine Unterteilung des Konstrukti-

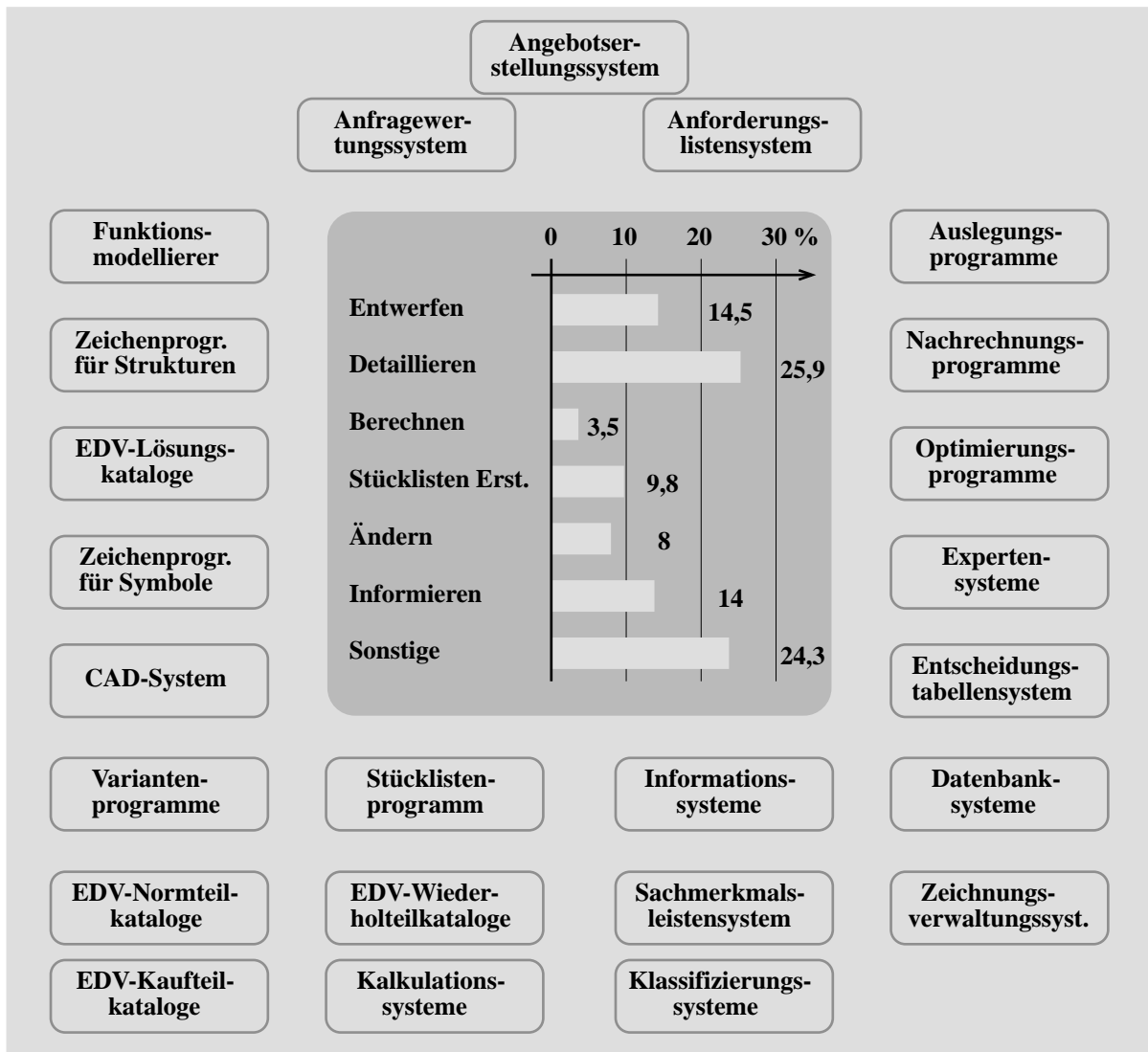


Bild 2.10 EDV-gestützte Hilfsmittel /28/

onsprozesses soll ermöglicht werden. Die Berücksichtigung von z.B. Zoll-, Transport- und ergonomischen Gesichtspunkten muß stattfinden.

In der VDI 2222 Blatt 2 wird festgestellt: „die gezielte Sammlung konstruktionsbezogener Daten ist darüber hinaus ein erster Schritt, um den Einsatz moderner Informationsträger oder von DV-Anlagen vorzubereiten“. Die Forderung nach „Objektkatalogen die u.a. auch Lieferdaten von Profilen ... enthalten“ wird als sinnvoll erachtet.

Eine der Anforderungen aus der VDI 2223 ist die Ermöglichung des „Festlegens der Werkstoffeigenschaften“. Eine weitere Forderung betrifft die Betrachtungsweise des gesamten Produktlebenszyklusses bereits in der Konstruktion:

„Der Zwang zu Grenzleistungen, der Kostendruck, werkstofftechnische Fortschritte und technologische Innovation in Fertigung und Montage beeinflussen ganz erheblich den Gestaltungsprozeß und führen zu einem Vorgehen, bei dem der Konstrukteur weit mehr als früher über die Fertigung und Montage hinaus den gesamten Lebenslauf des Produktes in seine Arbeit einbeziehen muß“.

Für Pahl und Beitz ist eine der zentralen Forderungen die „Informationsbeschaffung über Normen, Zuliefermaterial, Werkstoffe ...“.

Koller stellt die Anforderung „Fertigungsbedingte Forderungen (Vorgaben): Bauteile sollen bestimmte Abmessungen nicht überschreiten“.

Die Anforderungen von Roth lauten u.a. „Definition der Herstell-Einstiegs- bzw. Herstellungs-konstruktion sowie deren Nutzung im variablen Ablaufplan“.

Betrachtet man alle Anforderungen, so kann man die folgenden wesentlichen Anforderungen an EDV-Systeme im Konstruktionsbereich stellen:

Der EDV-gestützte Konstruktionsprozeß muß den gesamten Produktlebenszyklus berücksichtigen, die Informationsbeschaffung (Normen, Kataloge, Vorschriften, Richtlinien) muß unterstützt werden, die Informationsspeicherung und Verwaltung /57/ aller erarbeiteten Daten muß weltweit über Netzwerke ermöglicht werden, Variationen von Lösungen/Teillösungen müssen abgespeichert und bewertet werden können, die Prozeßsteuerung (Konstruktionsprozeß) muß ermöglicht werden, in frühen Phasen des Konstruktionsprozesses eine Kostenabschätzung durchführen zu können, Verschmelzung mit der kommerziellen EDV, das Teamwork der verschiedenen beteiligten Abteilungen zu Fördern um Probleme frühzeitig zu erkennen und zu beseitigen.

Am wahrscheinlichsten kann man diese Ziele erreichen, indem man ein System entwickelt, das die Steuerung der verschiedenen vorhandenen EDV-Systeme übernimmt und neu zu erstellende Systeme integriert sowie die entsprechenden Modelldaten verwaltet. Das nachfolgende Bild veranschaulicht die Problematik.

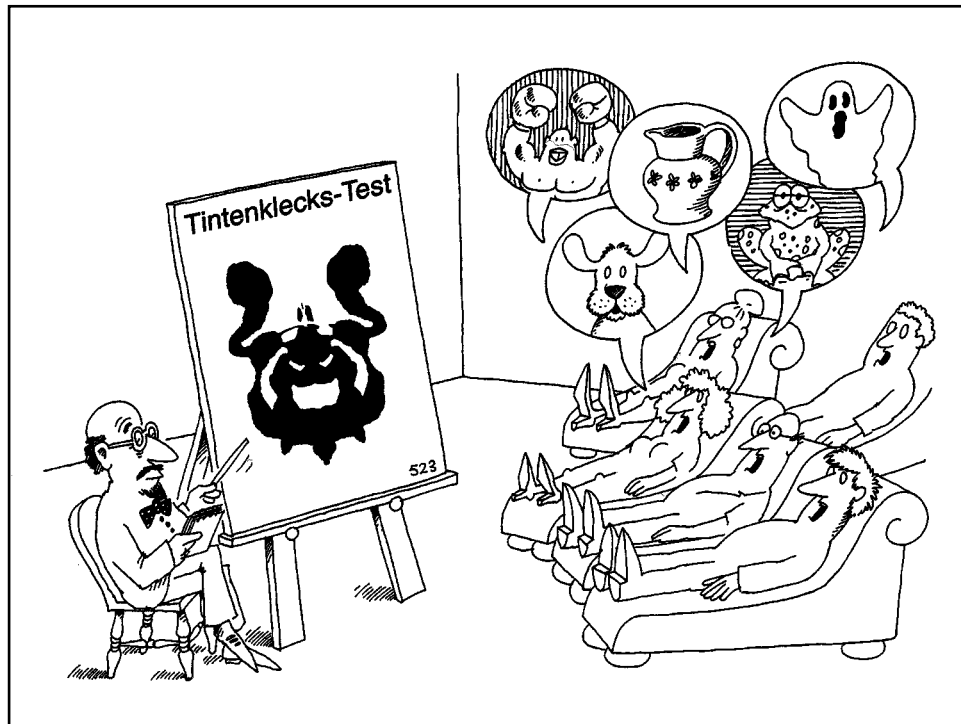


Bild 2.11 Tintenkleckstest /13/

Obwohl alle Personen (am Produktentwicklungsprozeß beteiligten Abteilungen) das gleiche Bild (Produkt) sehen, so haben sie doch alle unterschiedliche Interpretationen und Vorstellungen (Modelle/Daten) ein und desselben Gegenstands. Jede beteiligte Person besitzt aus ihrem Arbeits- und Lebensbereich eine spezielle, in ihr Schema passende Interpretation oder Sichtweise dieses Gegenstands. Aus ihrer Sichtweise sind alle Personen auch an unterschiedlichen Daten und Eigenschaften interessiert. Übertragen auf die Konstruktionstechnik lässt sich feststellen, daß das geometrische Modell ein zu interpretierendes Modell ist. Aus der Sicht des jeweiligen Bearbeiters besitzt die geometrische Darstellung eine jeweils aus dem Kontext des Bearbeiters definierte Bedeutung.

3 Fertigungstechnische Grundlagen

Nachdem verschiedene Konstruktionsmethodiken erläutert sowie deren Umsetzung in rechnergestützte Hilfsmittel vorgestellt wurden, erfolgt die Beschreibung und Gliederung der Fertigungstechnologie. In diesem Kapitel werden die notwendigen Grundlagen des Fertigungsprozesses sowie die Anwendbarkeit dieses Prozesses auf die in Kapitel 1 vorgestellte Philosophie dargelegt. Anschließend werden die einzelnen Fertigungsverfahren, die sich zur fertigungsorientierten Modellierung eignen, vorgestellt. Eine Gliederung der Werkzeugmaschinen folgt. Den Abschluß dieses Kapitels bildet die Beschreibung der Einbettung dieser Maschinen in den betrieblichen Datenfluß. Es wird gezeigt, wie aus einem CAD-System Steuerungsdaten an eine Werkzeugmaschine übertragen werden.

3.1 Gliederung der Fertigungsverfahren

Die DIN 8580 ist eine Gliederung der Fertigungsverfahren im Bereich der Metallverarbeitung. Sie besteht aus sechs Hauptverfahren. Diese sind: Das Urformen, das Umformen nach DIN 8582, das Trennen, das Fügen nach DIN 8593, das Beschichten und das Ändern der Stoffeigenschaften.

Das Urformen ist das Herstellen eines festen Körpers vorgegebener Gestalt aus einem Werkstoff, der in einem formlosen Ausgangszustand vorliegt. Während des Umformvorganges ändert der Werkstoff seinen Zustand. Er erhält seinen festen Zusammenhalt über das gesamte Werkstückvolumen. Relevante Fertigungsverfahren hierzu sind z.B. Schwerkraftgießen, Druckgießen, Schleudergießen, Stranggießen sowie Sintern und Pressen. Diese Verfahren eignen sich nicht für die Methode des fertigungsorientierten Modellierens, da die vorgegebenen Formen geometrischer Natur sind.

Kapitel 3 Fertigungstechnische Grundlagen

Die DIN 8580 bezeichnet das Formgeben von Werkstücken durch bildsames (plastisches) Formändern des Werkstoffes im festen Zustand. Sowohl die Masse als auch der Zusammenhalt des Materials bleiben bestehen. Geringe Materialverluste sowie kurze Fertigungszeiten sind ein hervorragendes Merkmal. Die Umformverfahren werden /22/ in die Verfahren des Druckumformens, Zugdruckumformens, Zugumformens, Biegeumformens sowie des Schubumformens eingeteilt. Moderne Blechkonstruktion mit 3D-Formfeatures zeigt, wie sich dieser Bereich teilweise durch fertigungsorientierte Konstruktionsweisen erschließt.

Für die fertigungsorientierte Modellierung ist das Verfahren des Trennens bzw. dessen Unterverfahren von großer Bedeutung. Im nachfolgenden Bild ist die Einordnung der Verfahren sowie eine Untergliederung in Unterverfahren dargestellt.

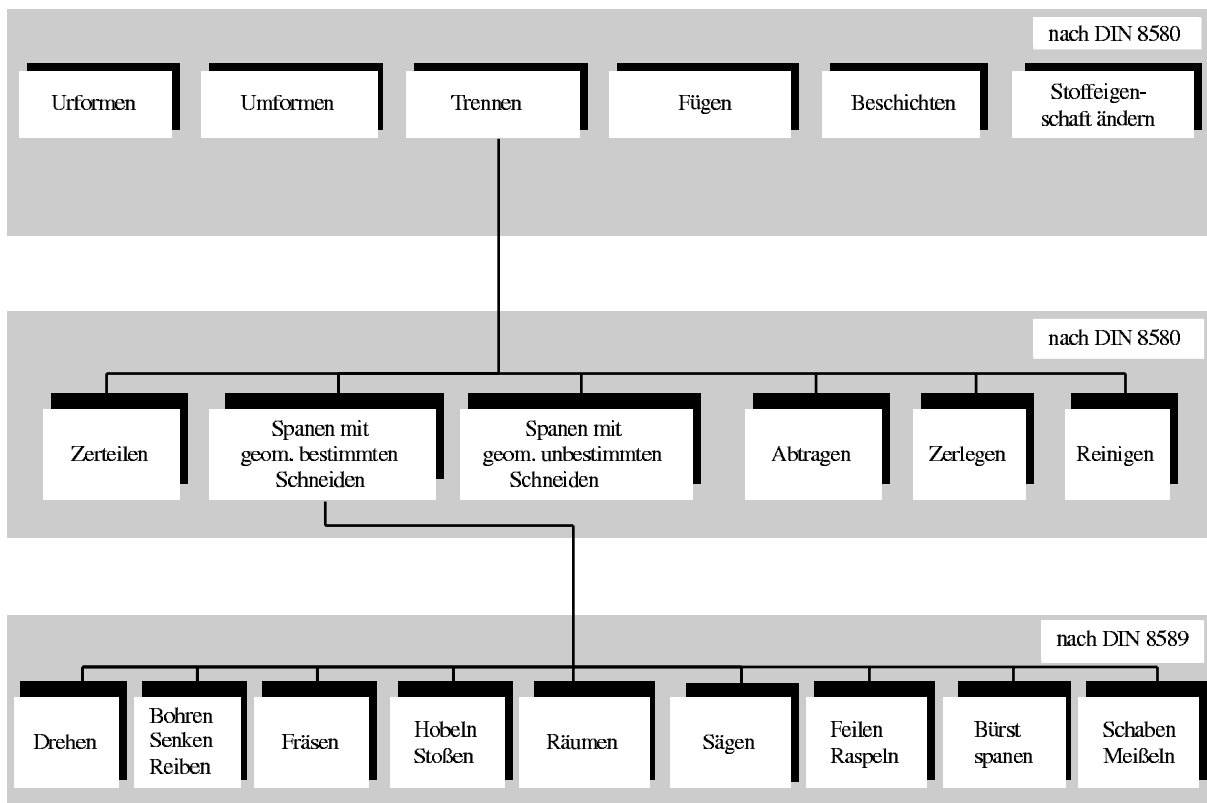


Bild 3.1 Fertigungsverfahren zur Metallverarbeitung nach DIN 8580 /151/

Es gliedert sich in die Verfahren:

Des Zerteilens, des Spanens mit geometrisch bestimmten Schneiden, des Spanens mit geometrisch unbestimmten Schneiden, des Abtragens, des Zerlegens und des Reinigens.

Das mechanische Trennen von Materialien ohne Spanbildung bezeichnet man als Zerteilen. Die DIN 8588 teilt die zerteilenden Verfahren in Scherschneiden, Messerschneiden, Beißschneiden, Spalten, Reißen und Brechen auf. Mit Ausnahme des Scherschneidens haben die aufgeführten Verfahren bei der Metallverarbeitung eine eher untergeordnete Bedeutung.

Das Verfahren des Scherschneidens ist für die fertigungsorientierte Modellierung sehr gut geeignet.

In der DIN 8589 ist der Begriff des Spanens definiert: „Spanen ist Trennen, bei dem von einem Werkstück mit Hilfe der Schneide eines Werkzeugs Werkstoffschichten in Form von Spänen zur Änderung der Werkstückform und/oder Werkstückoberfläche mechanisch abgetrennt werden.“

Bild 3.2 zeigt das Prinzip der Zerspantung anhand einer geometrisch bestimmten Schneide. Die Kienzle Formel beschreibt die an der Schneide angreifenden Zerspangkäfte. Das Verfahren wird unterteilt in die Fertigungsverfahren: Drehen, Bohren, Senken und Reiben, Fräsen, Hobeln, Stoßen, Räumen, Sägen, Feilen, Raspeln, Bürstspanen und Schaben.

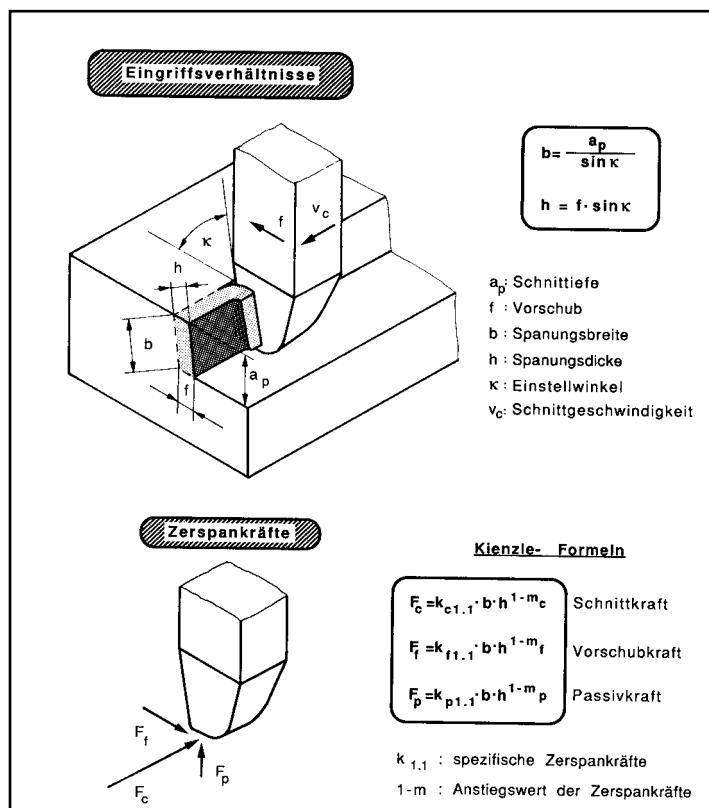


Bild 3.2 Kienzle-Formel /151/

Zu den Fertigungsverfahren, die das Spanen mit geometrisch unbestimmter Schneide realisieren gehört das „Schleifen mit rotierendem Werkzeug, das Bandschleifen, das Hubschleifen, das Hohnen, das Läppen, das Strahlspanen sowie das Gleitspanen. Bei diesen Verfahren /22/ sind die Anzahl der Schneiden, die Geometrie der Schneidkeile sowie die Lage der Schneiden zum Werkstück unbestimmt. Das Hohnen und Läppen eignen sich sehr gut als zu implementierende fertigungsorientierte Verfahren, da lediglich die Oberflächenrauigkeit verbessert wird. Geometrische Operationen müssen nicht erfolgen.

Die Verfahren des Abtrags unterteilen sich in thermische, chemische sowie elektrochemische Abtragverfahren. Von großer Bedeutung ist die Funkenerosion.

Beim chemischen Abtrag reagieren Säuren oder Laugen mit dem Werkstoff. Sie lösen so Volumen aus dem Werkstoff. Die Bearbeitung findet im Bad oder durch Aufsprühen statt. Nicht zu bearbeitende Teile werden durch Masken, die im Siebdruck- oder Photolackverfahren aufgebracht werden, geschützt.

Der elektrochemische Abtrag verwendet das Prinzip des Ladungs- und Stoffaustausches. Das Werkstück wird als Anode, das Werkzeug als Kathode gepolt. Die Anode geht in Lösung über. Die Kathode erfährt keinen Abtrag. Die Hauptanwendungsgebiete sind das elektrochemische Senken sowie das elektrochemische Entgraten.

Bei der funkenerosiven Bearbeitung erfolgt der Materialabtrag durch elektrische Entladungen zwischen Werkzeug und Werkstück. Die Entladung bewirkt eine hohe Temperaturentwicklung. Durch diese schmelzen Metallpartikel an der Entladungsstelle, verdampfen teilweise und werden durch mechanische und elektromagnetische Kräfte aus der Schmelze gerissen. Das Dielektrikum ist das Arbeitsmittel, das das abgetragene Material abtransportiert. Gleichzeitig dient es als Kühlung.

Das Zerlegen oder besser Demontieren von Werkstücken dient dazu, Teile auszutauschen oder Maschinen transportfähig zu machen. Eine bessere Entsorgung bzw. ein Recyclen von Teilen wird durch eine leichte Demontierbarkeit unterstützt. Aus dem Blickwinkel der fertigungstechnischen Modellierung besitzt dieses Verfahren keinerlei Bedeutung.

Zur Entfernung unerwünschter Teile von der Oberfläche eines Werkstückes werden die Verfahren des Reinigens verwendet. Die Verfahren des Bürstens, Strahlens, Waschens und Beizens gehören hierzu. In der fertigungsorientierten Modellierung finden diese Verfahren keine Verwendung.

Technische Produkte werden aus Einzelteilen und Baugruppen zusammengesetzt. Das Zusammensetzen kann in die Teilbereiche des Zusammenlegens, Füllens, An- und Einpressens, durch Urformen, Umformen oder das Stoffeverbinden gelöst werden. Aus konstruktiver Sicht unterteilt man in kraftschlüssiges, formschlüssiges und stoffschlüssiges Verbinden. Aus der Sicht der fertigungsorientierten Modellierung besteht das Interesse der Sicherung von Toleranzdaten und Stoffdaten. Diese können zur Kostenfrüherkennung herangezogen werden.

Die DIN 8580 definiert *Beschichten als das Aufbringen einer festhaftenden Schicht aus formlosem Stoff auf ein Werkstück*. Der formlose Stoff kann in pastenförmigem, flüssigem, breiigem oder gasförmigem Zustand vorliegen. Häufige Verwendung finden die Verfahren des Aufdampfens (gasförmiger Zustand), Lackierens und Auftragsschweißens (flüssiger und pastenförmiger Zustand). Von größter Bedeutung sind die Verfahren des Galvanisierens (Verchromen, Vernickeln, Verzinken und Vergolden). Diese Verfahren dienen vornehmlich dem Zweck, den Oberflächenschutz zu verbessern, gewünschte Gleiteigenschaften zu erhalten, das Verschleißverhalten zu optimieren oder einfach nur den optischen Eindruck zu verbessern. Da die genannten Verfahren keinen wesentlichen Einfluß auf die Geometrie eines Werkstückes besitzen, sind sie aus Sicht der fertigungstechnischen Modellierung nicht relevant.

Aus der Verfahrensgruppe „Stoffeigenschaften ändern“ sind die Verfahren des Härtens und Vergütens gebräuchlich. Dabei werden die Stoffeigenschaften durch Verändern des Gitteraufbaus erzeugt. Für die fertigungsorientierte Modellierung sind diese Verfahren nicht nutzbar, da deren Auswirkungen beim Konstruieren eines Werkstückes nicht exakt genug erfaßt werden können. Es können lediglich Aufmasshinweise angebracht werden.

3.2 Die Gliederung der Werkzeugmaschinen

Das Normblatt 69651 legt die Gliederung, Bezeichnung und Einteilung von Werkzeugmaschinen fest. Sie ist an die Norm 8580 angelehnt worden. Berücksichtigt werden die Verfahren des Fügens, Trennens und Umformens. Das in Bild 3.3 schematisiert dargestellte Normblatt 69651 definiert eine Werkzeugmaschine als „...mechanisierte und mehr oder weniger automatisierte Fertigungseinrichtung, die durch relative Bewegungen zwischen Werkzeug und Werkstück eine vorgegebene Form oder Veränderung am Werkstück erzeugt“.

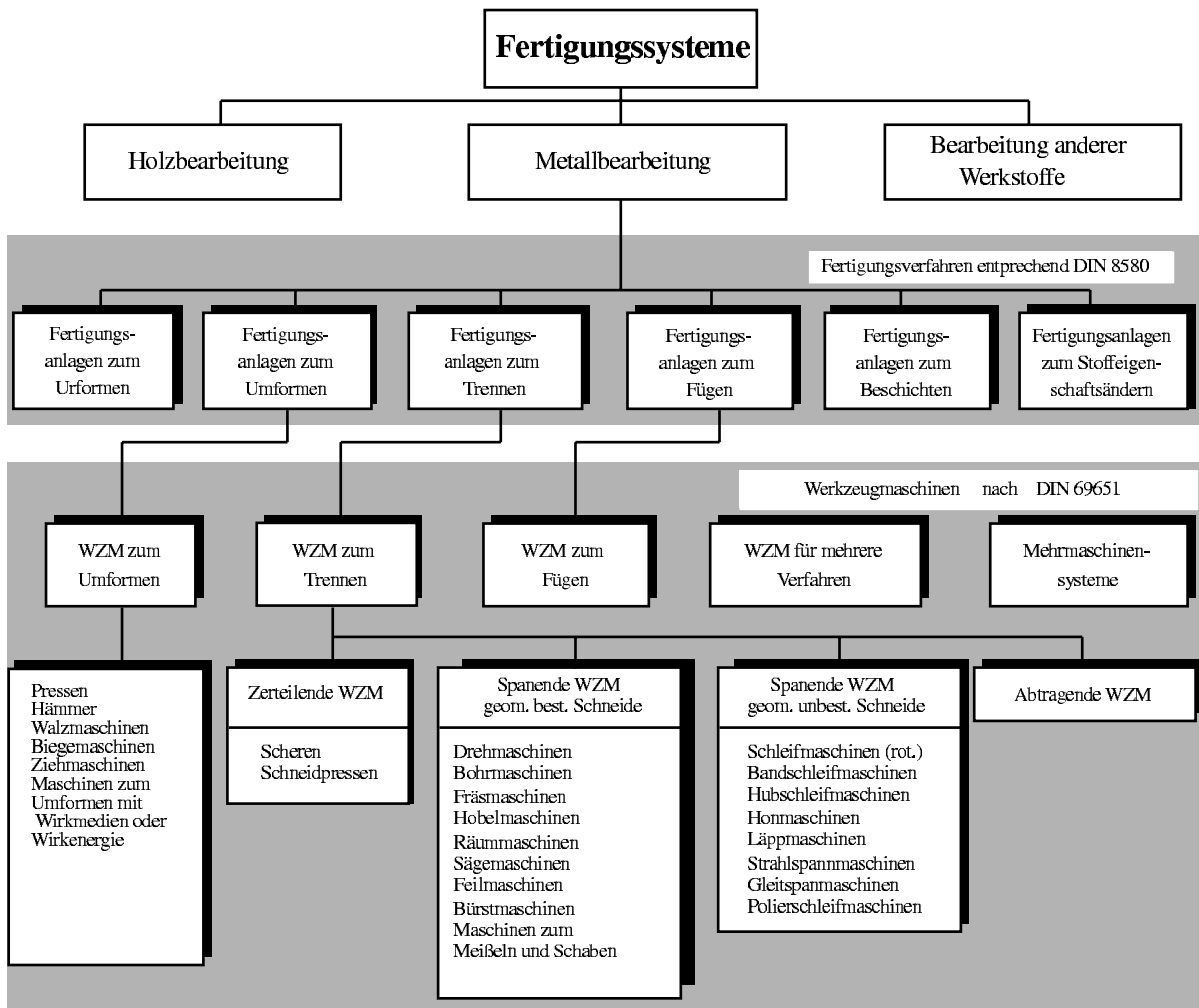


Bild 3.3 Gliederung der Werkzeugmaschinen nach DIN 69651 /151/

Durch den Automatisierungsgrad und das Fertigungsverfahren erfolgt die Bezeichnung einer Werkzeugmaschine. Typische Vertreter von Einzelmaschinen, mit denen nur ein Fertigungsverfahren durchgeführt werden kann sind z.B. Drehmaschinen. Die Bezeichnung erfolgt in diesem Fall nur nach dem Fertigungsverfahren. Können von einer Werkzeugmaschine mehrere

Verfahren durchgeführt werden, so stellt man der Bezeichnung den Wortteil „Bearbeitungs“ voran. Als Beispiel sei hier die Bearbeitungsmaschine für prismatische Teile genannt. Maschinen, die nur zu einem Zweck gebaut wurden, z.B. dem Fräsen von Zahnrädern, werden als Einzweckmaschinen bezeichnet. Sollen Maschinen noch weiter unterteilt werden, so wird häufig die Bauform als Kriterium herangezogen.

Die Bezeichnung der Fertigungssysteme erfolgt nach deren Automatisierungsgrad. Die in der Regel mikroprozessorgesteuerten Maschinen zeichnen sich durch einen automatischen Werkzeugwechsel sowie einen eigenen Werkzeugspeicher aus. Die Bearbeitungszentren werden oft zu größeren und komplexeren Einheiten, den Transferstraßen, mit Werkzeuglager und Montageeinrichtungen sowie Meßstationen kombiniert. Aufgrund der zu fertigenden Losgrößen sowie der geänderten wirtschaftlichen Rahmenbedingungen (kürzere Produktlebenszeiten) besteht ein Trend hin zu flexiblen Fertigungssystemen. Diese komplexen Systeme werden über einen eigenen Rechner gesteuert.

3.2.1 Die Steuerung von Werkzeugmaschinen

Werkzeugmaschinen der aktuellen Produktion werden in der Regel numerisch gesteuert (NC). Informationen (Befehle und Daten) für die Werkzeugmaschine werden an diese in digitaler Form übermittelt. Als Datenformat hat sich das Format CLDATA (Cutter Location Data) durchgesetzt. Es enthält Befehle und Daten zur Geometrie, Technologie und Maschinenhilfsfunktionen. In den Anfangsjahren der NC-Steuerungen wurden die Informationen auf einem Computer erstellt. Im Anschluß wurden die Informationen auf einen Lochstreifen übertragen, der dann von der NC-Maschine eingelesen und verarbeitet wurde. Durch die zunehmende Verbreitung der Netzwerke wurden die Lochstreifen überflüssig. Die NC-Programme werden direkt über das Netz an die NC-Maschine übertragen. Neben den eigentlichen NC-Programmen werden auch Werkzeugdaten übermittelt sowie Betriebs- und Prozeßdaten von der Maschine abgefragt. Als Netzwerke kommen sogenannte Local Area Networks zum Einsatz. Auf ihnen kommunizieren die verschiedenen Rechner und NC-Einheiten über ein Protokoll. Die ISO (International Standards Organisation) und die CCITT (Consultative Committee for International Telegraph and Telephone) haben das OSI (Open Systems Interconnection) - Referenzmodell festgelegt, das aus sieben Schichten besteht.

Diese sieben Schichten /130/ unterstützen jeweils eine ganz bestimmte Funktion die durch ein definiertes Protokoll implementiert wird. Je niedriger die Schichten desto Hardwarenäher ist das Protokoll, je höher die Schicht desto applikationsnäher ist die Schicht. Die sieben Schichten werden oft in zwei Gruppen unterteilt.

Die erste Gruppe beinhaltet die Layer eins bis drei. Diese realisieren die Hardwaregerechte Kommunikation zweier miteinander verbundener Geräte. Die zweite Gruppe besteht aus den Layern vier bis sieben. Diese unterstützen Protokolle, die es ermöglichen, daß zwei Applikationen miteinander kommunizieren können.

Jede Schicht des Modells existiert als ein unabhängiges Modul. Die Module der einzelnen Ebenen können jederzeit durch ein anderes ersetzt werden, ohne daß sich Auswirkungen auf die darüber oder darunter liegenden Schichten ergeben sollten. Im nachfolgendem Bild 3.4 ist der Aufbau des Schichtmodells dargestellt.

Die **physikalische Schicht** kontrolliert das Senden und Empfangen von Bitströmen über physikalische Medien. Eine Strukturierung der Bitströme findet nicht statt. In dieser Schicht werden lediglich die elektrischen, mechanischen, optischen und funktionalen Schnittstellen für das physikalische Medium beschrieben.

Die **verbindende Schicht** ist dafür zuständig, daß eine fehlerfreie Übertragung der Datenframes erfolgt. Sie setzt, wie dargestellt, auf der physikalischen Schicht auf. Alle über dieser Schicht liegenden Ebenen können von einer physikalisch fehlerfreien Übertragung der Daten ausgehen.

Für die Kontrolle eines Subnetzes ist die **Netzwerkschicht** vorhanden. Sie legt die Transportwege der Daten im Subnetz fest. Dabei werden die Prioritäten der aktiven Dienste beachtet. Das Beenden, das Aufbauen sowie das Erhalten der Verbindung während eines Datentransportes ist Aufgabe der Netzwerkschicht.

Zur Kontrolle der Nachrichten dient die **Transportschicht**. Sie kontrolliert, daß die Nachrichten in der Reihenfolge abgeliefert werden, in der sie gesandt wurden. Weiterhin wird sichergestellt, daß Nachrichten weder verloren gehen, noch daß sie mehrmals vorhanden sind.

Die **Sitzungsschicht** dient zum Aufbau von Sitzungen. Diese findet zwischen Prozessen statt, die auf verschiedenen Computern ablaufen. Sie unterstützt die Datenübertragung im Nachrichtenmodus.

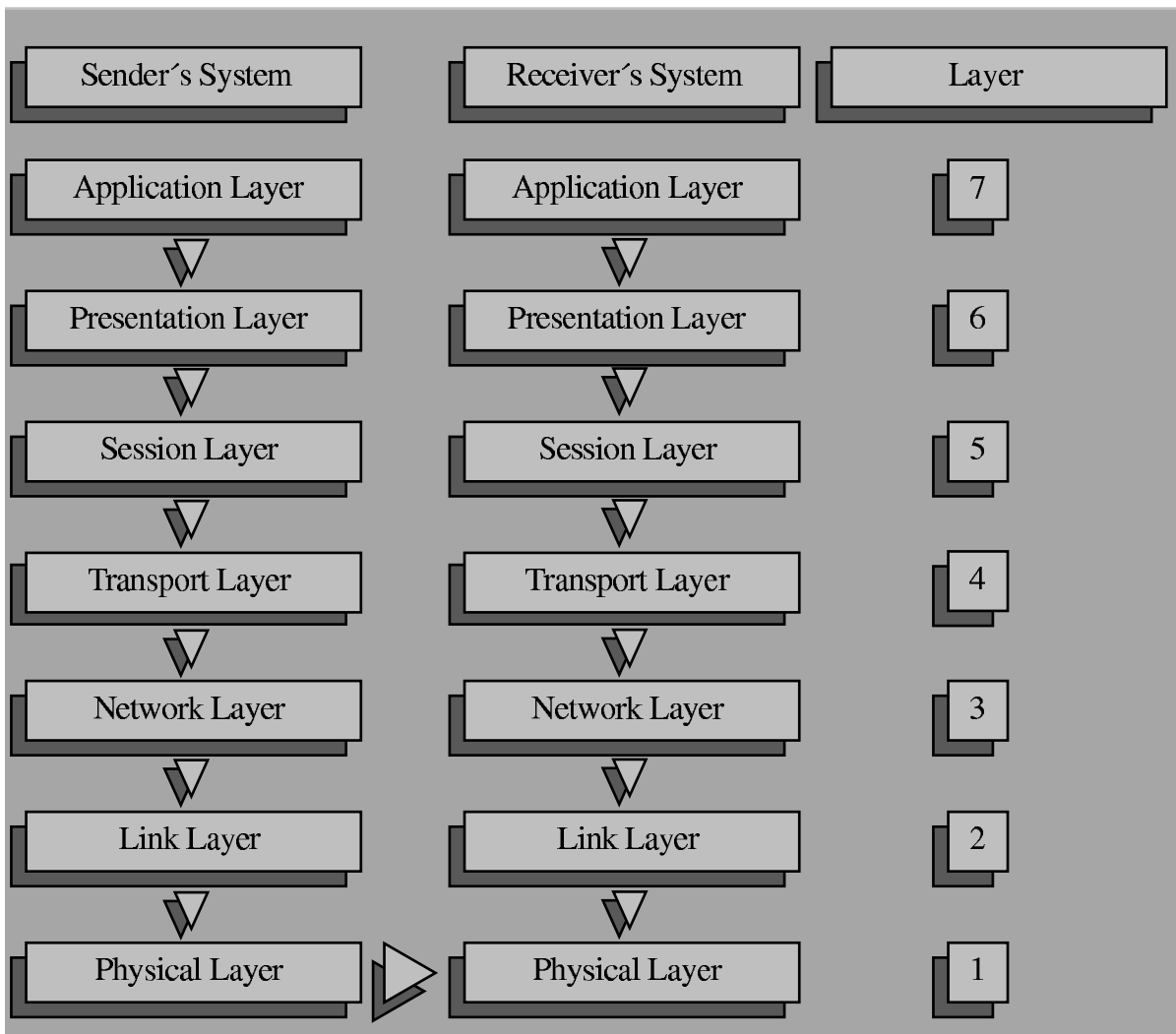


Bild 3.4 ISO/OSI 7-Schichten Modell /91/

Zur Übersetzung von Daten im Netzwerk wird die **Präsentationsschicht** genutzt. Der sendende Computer übersetzt durch seine Präsentationsschicht die aus der Anwendungsschicht kommenden Daten. Die Präsentationsschicht des empfangenden Computers übersetzt dieses Format in ein seiner Anwendungsschicht bekanntes.

Benutzer- und Anwendungsprozesse greifen durch die **Anwenderschicht** auf Netzwerkdienste zu. Dazu stellt die Anwenderschicht den zugreifenden Prozessen Funktionen wie Verzeichnisdienste, Netzwerk-Management, Management elektronischer Nachrichten, Unterstützung der Interprozesskommunikation, Remote-Zugriffe auf Drucker und Dateien sowie die Simulation virtueller Terminals.

4 CAD Systemaufbau

Seit der Entwicklung der ersten CAD- bzw. Zeichensysteme haben sich deren Aufgabengebiete enorm erweitert. Stand zu Beginn der Entwicklung die Abbildung des Zeichenbrettes im Rechner im Vordergrund, so wurden zunehmend neue Modellierungsmethoden wie das parametrische Modellieren und Aufgabengebiete wie die Feature-Technologie entwickelt. Dies stellte hohe Anforderungen an die Softwareentwicklung. Da die vielfältigen Sachverhalte nicht mehr mit monolithischen Softwarepaketen zu bewältigen waren, wurden entsprechend aufeinander abgestimmte Module entwickelt.

Im Einsatz befindliche CAD-Systeme besitzen einen konzeptionell ähnlichen Systemaufbau. Um einen Modellierkern herum sind verschiedene Module angeordnet. Hierzu gehören einerseits Standardmodule wie eine grafische Library, eine Benutzeroberfläche und ein Datenbanksystem. Andererseits gehören hierzu aufgabenspezifische Module. Der Erfolg des CAD-Systems AutoCAD /62/ basiert vornehmlich auf dieser Anordnung. Um das von Autodesk geschaffene System haben eine große Anzahl von Softwareentwicklern branchenspezifische Lösungen implementiert. Hierzu gehören Module für die Architektur, die Elektrotechnik sowie Teilbereiche des Maschinenbaus. Erst diese Implementationen generieren den wirtschaftlichen Erfolg, der mit einem allgemeinen, „von der Stange“ gekauftem CAD-System nicht erreicht werden kann. Die aufgabenspezifische, der geometrischen Modellierungsebene übergeordnete Modellierungsebene ist beim Einsatz des Systems die Entscheidende. Durch sie entstehen die Zeitvorteile beim Erzeugen eines Modells.

Gleichzeitig ist der Datenaustausch zwischen verschiedenen Systemen qualitativ verbessert worden. Formate wie IGES, VDAFS und STEP /26, 48 und 43/ bieten derzeit einen Datenaustausch zwischen unterschiedlichen Systemen ohne allzugroße Datenverluste. Der Aufwand an Nacharbeiten hält sich meist ebenfalls in Grenzen. Zudem bietet STEP /2 und 42/ durch die Übertragung von Produktdaten zusätzliche effektive Möglichkeiten.

4.1 Der Systemaufbau von Standardsystemen

Wie bereits einleitend erwähnt, besitzen alle CAD-Systeme einen ähnlichen Aufbau. Kern eines jeden CAD-Systems ist der grafische Modellierer /26, 35, 51, 92, 126 und 127/. In ihm werden alle geometrischen und topologischen Daten abgelegt. Er führt alle mathematischen Berechnungen durch und bietet eine Schnittstelle, um von einer Programmiersprache auf ihn zugreifen zu können. Diese wird von CAD-Erweiterungen genutzt. Dargestellt werden die rechnerinternen Daten des Modellierers durch eine grafische Library. Sie visualisiert die Daten auf die verschiedensten Arten auf den Ausgabegeräten Bildschirm, Drucker und Plotter. Die Interaktion zwischen CAD-System und Anwender erfolgt über ein User-Interface. Dieses stellt das Betriebssystem zur Verfügung, auf dem das CAD-System implementiert ist.

4.1.1 Modellierkerne

Bei den 3-D-Modellierern wurde bisher zwischen Systemen unterschieden, die auf der B-Rep (**B**oundary **R**epresentation) Methode beruhen und zwischen Systemen, die auf der CSG-Methode (**C**onstructive **S**olid **G**eometry) beruhen. Ein CSG-Modell wird aus primitiven Körpern aufgebaut. Zu diesen zählen der Quader, des Ellipsoid, der Zylinder, der Kegel und der Torus. Die Körper können über die Booleschen Operationen zu komplexen Bauteilen zusammengefügt werden. Die rechnerinterne Speicherung erfolgt in einer Baumstruktur. Die Entstehungsgeschichte des Geometriemodells bleibt auf diese Weise erhalten. Änderungen am Modell sind leicht durchführbar. Bei der Visualisierung wird das CSG in ein B-Rep-Modell überführt. Ein B-Rep Modell besteht aus einer Liste berandeter Flächen. Es besitzt seinen Ursprung in der Freiformflächenmodellierung. Hierbei erfolgen Kurven- und Flächendarstellungen in einer parametrischen Darstellung. Die Kurven und Flächen werden durch Stützpunktpolygone und polynomische Basisfunktionen erzeugt und verändert.

Die bekanntesten Vertreter dieser Darstellung sind die NURBS (Nonuniform Rational B-Spline Curves) [29 und 107]. Sie gestatten es, Kegelschnitte, Quadriken und Rotationsflächen exakt darzustellen. Durch die Definition über Gewichte und Stützpunkte haben sie einen großen Einsatzbereich, die Manipulation von Kurven und Flächen betreffend. So ist das Einfügen und Löschen von Knoten möglich, Kurven können geteilt werden, der Grad des Polygons ist beeinflussbar und NURBS-Kurven verhalten sich invariant bei der Skalierung, Rotation und Translation. NURBS besitzen jedoch den Nachteil eines deutlich erhöhten Speicherplatzbedarfs.

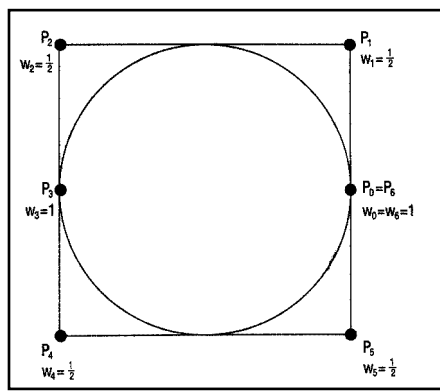


Bild 4.1 Ein durch einen NURB dargestellter Kreis

Im Bild 4.1 ist ein Kreis durch einen NURB dargestellt. Der Speicherplatzbedarf eines Kreises besteht aus seinem Mittelpunkt, einem Richtungsvektor und dem Radius. Dies ergibt einen Bedarf von sieben numerischen Werten. Der Nurb benötigt hingegen 38 numerische Werte. Ein weiterer Nachteil der NURBS ist der Verlust der Typinformationen eines geometrischen Elements. Der Modellierer weiß nicht mehr, ob er einen Kreis oder eine Ellipse bearbeitet. Die Datenstruktur eines B-Rep-Modellierers beinhaltet neben den Geometriedaten die topologischen Daten. Dies sind die geschlossenen Konturzüge der Flächen, die Definition von Kanten (Edges) und Ecken (Vertex).

Moderne Modellierer unterstützen die B-Rep- und die CSG-Modellierung. Beide werden zu einem hybriden Modellierer [45 und 136] verbunden. Dies hat den Vorteil, daß die Verfügbarkeit der Erzeugungsstruktur gegeben ist. Die Praxis des Maschinenbaus erfordert es z.B aufgrund des Modellerstellungsaufwandes, daß Modellierer gleichzeitig Draht-, Flächen- und Volumenmodelle bearbeiten können. Dies macht es notwendig, neue Modellierer zu entwic-

keln, die diese sogenannten Non-Manifold-Modelle benutzen. Diese Mischmodelle werden auch als Zellkomplexe bezeichnet. Eine 0d-Zelle entspricht einem Punkt, eine 1d-Zelle einer Kante, eine 2d-Zelle einer ebenen Fläche und eine 3d-Zelle einem berandeten Volumen.

4.1.2 Der ACIS Modellierkern

Die Anzahl der sich im Einsatz befindlichen Modellierkerne beschränkt sich auf eine kleine Zahl. Der bekannteste und kommerziell erfolgreichste ist der ACIS-Kern /88/. Er wurde von der Firma Spatial Technologie¹ entwickelt. Die Mitarbeiter der Firma Spatial waren bereits in den 70'er Jahren an der Entwicklung des damals verwendeten Modellierkerns ROMOLUS beteiligt. Der ACIS-Kern ist somit eine logische, keine programmtechnische Weiterentwicklung, die auf dem Erfahrungsschatz von ROMOLUS basiert.

Der ACIS Geometric Modelierer ist ein Framework zur Unterstützung aller Bereiche der geometrischen Datenverarbeitung. Es ist in C++ /45, 78 und 133/ implementiert worden und besteht aus einer entsprechenden Klassenhierarchie. Diese gliedert sich in drei wesentliche Teile. Zum einen besteht die Unterstützung eines Dateiformates in ASCII als auch in binärer Form. Diese dienen dem Sichern der Modelle sowie zum Datenaustausch. So können CAD-Systeme, die nicht auf dem ACIS-Kern basieren, Daten über dieses Softwareprodukt importieren und verarbeiten. Die Dateien werden unter dem Begriff SAT-Files geführt. Der zweite Teil betrifft die komfortable Anbindung von Erweiterungen und Einbettung von Komponenten im Kern. Im ACIS-Sprachgebrauch werden diese als optionale und interne HUSKS bezeichnet. Der dritte Teil besteht aus dem eigentlichen Modellierkern. Die wichtigste Komponente des Kerns stellt Funktionen zur Bearbeitung von Geometrie und Topologie zur Verfügung. Weitere Komponenten erlauben das Bearbeiten von Splines, die Nutzung der mengentheoretischen Funktionen, die Nutzung komplexer Operationen wie das Verrunden und Sweepen, die Nutzung von Darstellungsfunktionen, die Darstellung durch einfache Schattierungsalgorithmen sowie die Nutzung plattformabhängiger Schattierungssoftware. Der Kern ist ein Solid-Modeler. Neben der B-Rep Struktur wird auch die Non-Manifold-Geometrie unterstützt. Flächenmodelle können ebenfalls bearbeitet werden. Im nachfolgendem [Bild 4.2](#) ist der Aufbau des Systems aus Komponenten verdeutlicht. Neben dem ACIS-Kernel existieren Klassen als direk-

1. Die Firma Spatial Technologie wurde zwischenzeitlich von der Firma Dassault übernommen

tes Interface (Direct Object Interface) um auf den Kernel zugreifen zu können. Das API zum Zugriff auf parametrische Kurven und Flächen ist vom allgemeinen API getrennt. Auf das API kann durch C++ über die direkte Schnittstelle zugegriffen werden. Für die einzelnen Komponenten wie z.B. grafische Interaktion, Geometrie und Darstellung existieren eigene API's. Diese bieten wiederum eine direkte Schnittstelle zu C++. Sie bilden das neue allgemeine API.

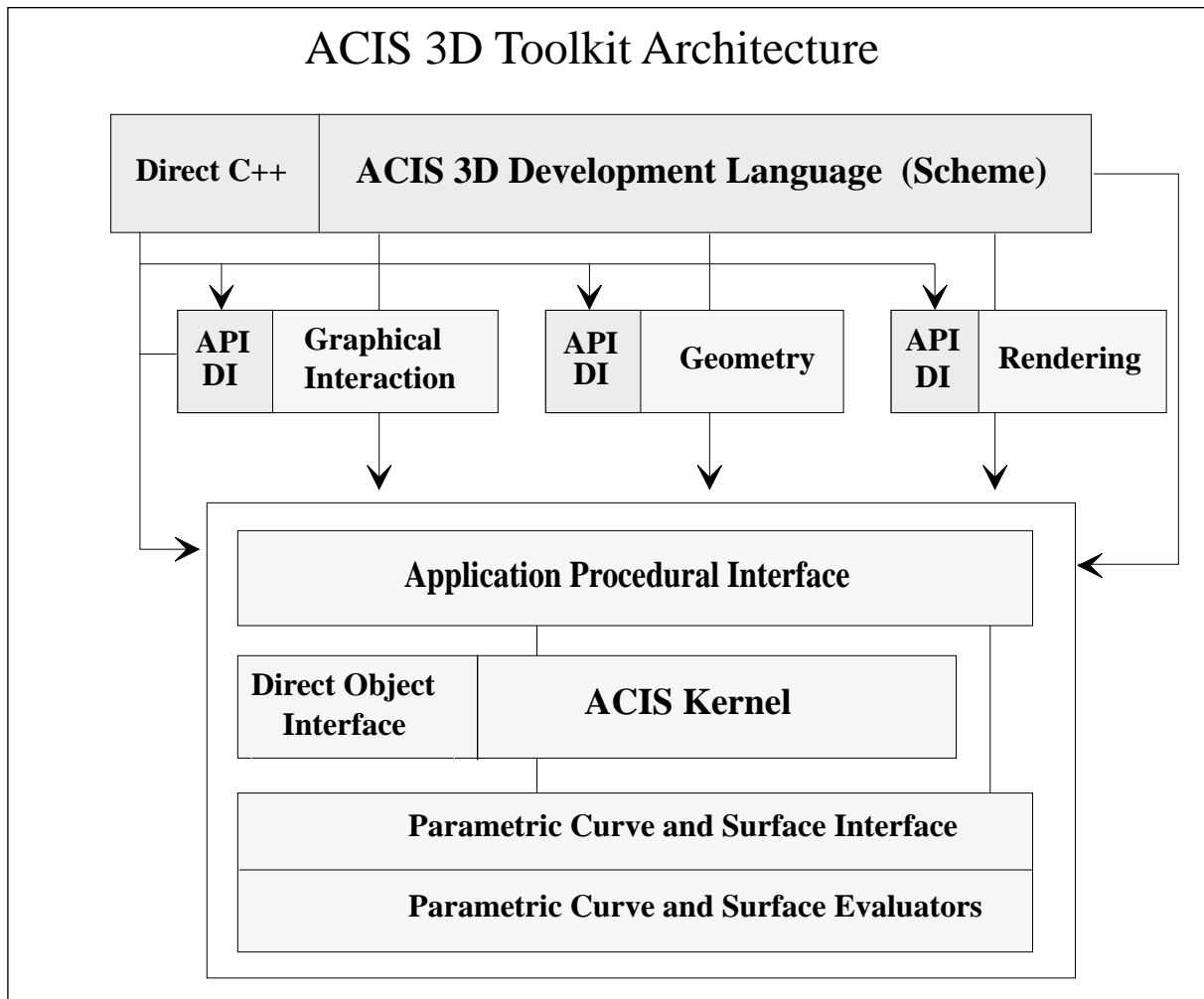


Bild 4.2 ACIS 3D Toolkit /88/

Wurzel der C++ - Objekthierarchie des ACIS-Kernel ist die Klasse „ENTITY“. Sie implementiert das Laden und Speichern von Objekten in Dateien. Alle mächtigeren ACIS-Objekte, die permanent gesichert werden müssen, sind von dieser Klasse abgeleitet. Von der Klasse „ENTITY“ wurden drei Gruppen von Klassen abgeleitet. Die erste Gruppe ist die Klasse der geometrischen Objekte. Sie besteht aus der Klasse Punkt (APOINT), Kurve (CURVE), para-

metrische Kurve (PCURVE) und Fläche (SURFACE). Von der Klasse Kurve wurden die Klasse Gerade, Ellipse, Schnittkurve und zusammengesetzte Kurve abgeleitet. Die Klasse Fläche dient als Basisklasse für die Klassen Ebene, Kegel, Kugel, Torus, Spline und Netzfläche. Letztgenannte besteht aus einem Facettennetz. Sie dient zur Darstellung von großen Datenmengen. Die zweite Gruppe repräsentiert die topologischen Objekte. Zu ihnen zählen der Körper, eingeschlossene Hohlräume, zusammenhängende Faces (Shells), Subshells, durch Kanten begrenzte Flächen (Faces), zusammenhängende Coedges (Wire), das zusammenhängende Gebiet einer Fläche (Loop), den Zusammenhang zwischen Fläche und Kante repräsentierenden Elementen (Coedge), Kanten (edge) und Eckpunkte (vertex). Die dritte abgeleitete Gruppe sind die Attribute. Dies können Systemattribute oder vom Nutzer implementierte Attribute sein. Das [Bild 4.3](#) zeigt die grafische Darstellung.

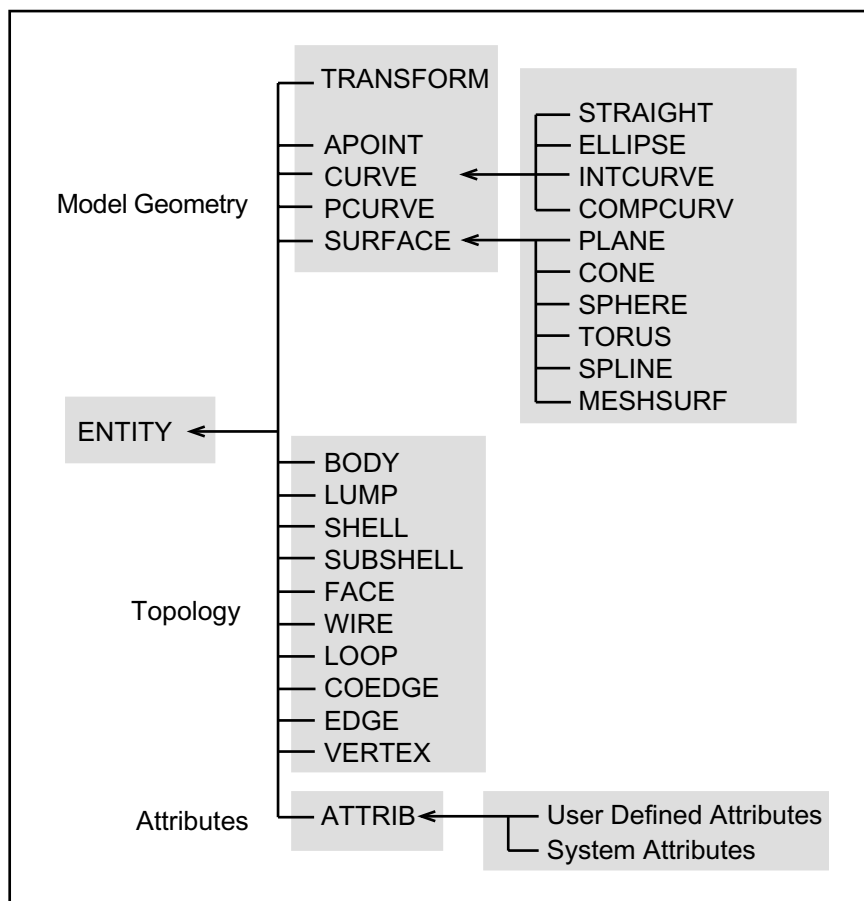


Bild 4.3 ACIS-Klassenhierarchie /88/

Im nachfolgenden Teil dieses Abschnittes wird die CV-DORS Klassenhierarchie vorgestellt. Sie besitzt viele Ähnlichkeiten, aber auch einige wesentliche Unterschiede zur ACIS-Hierarchie. So besitzt der CV-DORS - Modellierer die Klassen Geometrie und Datenbank. Von der Klasse Geometrie wurden die Klassen Topologie und Entity abgeleitet. Die Klasse Topologie besitzt einen ähnlichen Aufbau wie ihre entsprechende ACIS-Klasse. Die Klasse Entity wird zunächst in mehrere allgemeine Linien- und Flächenklassen aufgeteilt. Diese verzweigen anschließend in der nächsten Hierarchiestufe in spezielle Klassen. Die Klasse Kurve wird durch den CV-DORS-Modellierer z.B. weiter in die Klassen Linie, Kegelschnitte und NURB unterteilt.

Die im Bild 4.3 dargestellte Objekthierarchie muß den verschiedensten Anforderungen genügen. Sie muß die rechnerinterne Darstellung von Volumenkörpern (Solids), Flächen (Surfaces) und Drahtmodellen (Wireframe) ermöglichen. Diese können zudem gleichzeitig in einem Modell existieren. Dabei müssen Modelle verarbeitet werden bei denen z.B. ein Solid eine gemeinsame Kante mit einem Drahtmodell besitzt. Hieraus resultieren enorme Anforderungen an die Verwaltung der Datenstruktur, die in der Möglichkeit mündet, unvollständige (nonmanifold) Geometrien verwalten zu können.

Das Bild 4.4 zeigt die komplizierten Relationen der einzelnen Klassen. Zu erkennen ist ein hierarchischer Aufbau. Die Klasse Body stellt den Zugriff auf die Gesamtstruktur dar. Sie besitzt Referenzen auf die nächst höherwertigen Elemente wie die Klasse Wire oder Lump. Diese referieren auf die Elemente Coedge bzw. die Klasse Lump kann zusätzlich aus Shells und Faces bestehen. Diese referieren wiederum auf Coedges. Coedges referieren auf Edges, diese auf Vertexes. Die drei letztgenannten sowie die Klasse Face besitzen referenzen auf die geometrischen Elemente Fläche, Kurve bzw. Polykurve und Punkt.

Die angezeigte Klasse „box“ dient dazu, um jedes ACIS Object eine umhüllende Box zu legen. Anhand dieser Box kann mit wenig Rechenaufwand die Lage und Ausdehnung eines Körpers zu einem anderen bestimmt werden. Überschneiden sich die zwei umhüllenden Boxen zweier Körper nicht, so muß nicht erst ein komplizierter Schnittalgorithmus durchlaufen werden um dies zu ermitteln.

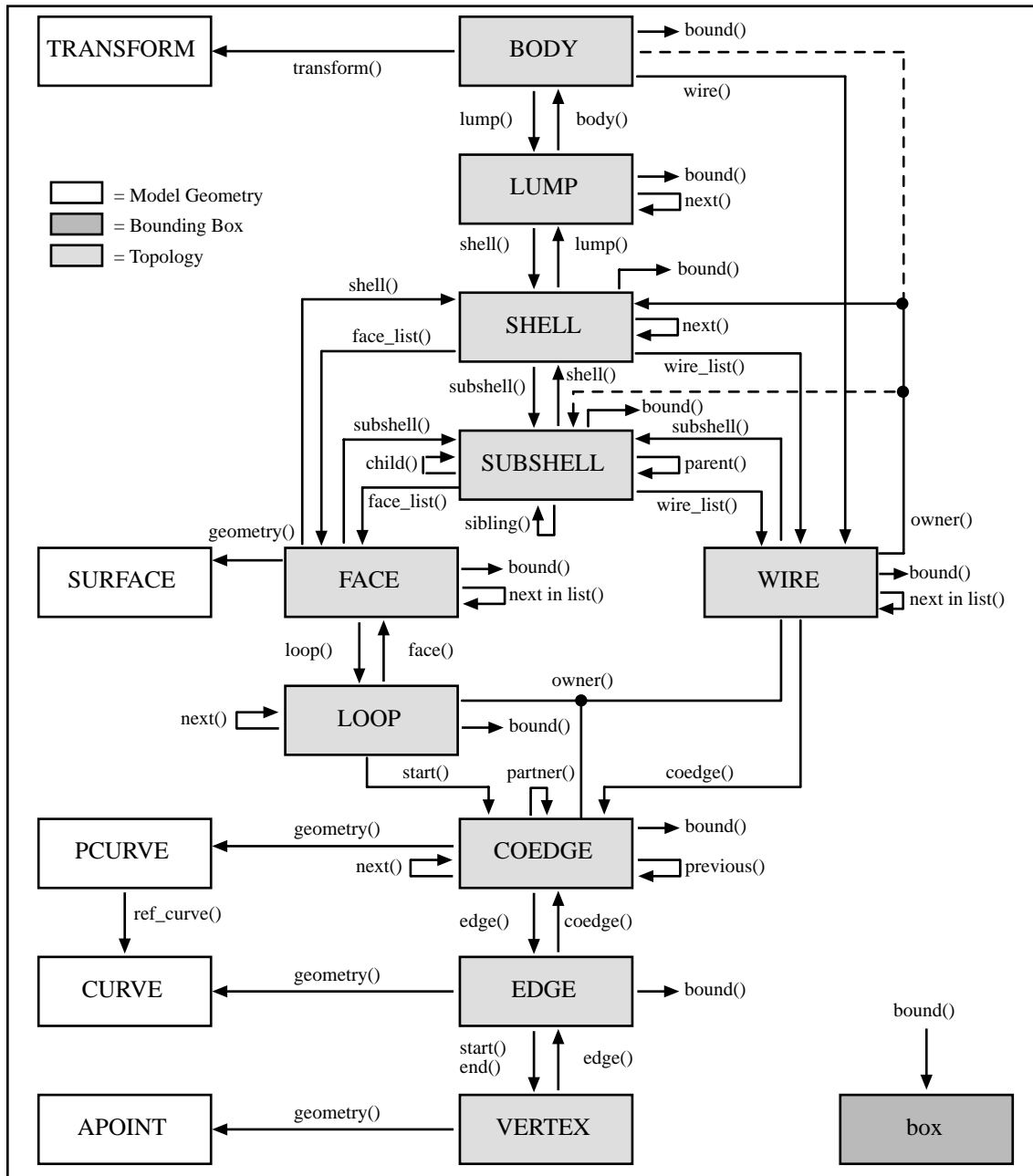


Bild 4.4 Klassenrelationen /88/

Um einen Modellierkern flexibel zu gestalten muß er erweiterbar sein. Spatial Technology hat das Konzept der sogenannten Husks entwickelt. Durch sie ist es möglich, die für den CAD/CAM-Bereich notwendigen Zusatzmodule einzubinden. Diese werden einerseits von Spatial selbst, andererseits von Drittanbietern entwickelt und vermarktet. Auf diese Weise ist es möglich, auf eine breite nutzbare Basis verschiedene, weniger oft genutzte Zusatzmodule aufzusetzen.

zen. Von Spatial selbst werden Module zum Verrunden von Flächen, zum erweiterten Rendern von Modellen, zum Deformieren von Körpern sowie verschiedener Module zum Exportieren der Daten in das IGES, STEP und VDA-FS Format angeboten. Im nachfolgenden Bild ist schematisch dargestellt, wie HUSKS auf ACIS zugreifen können.

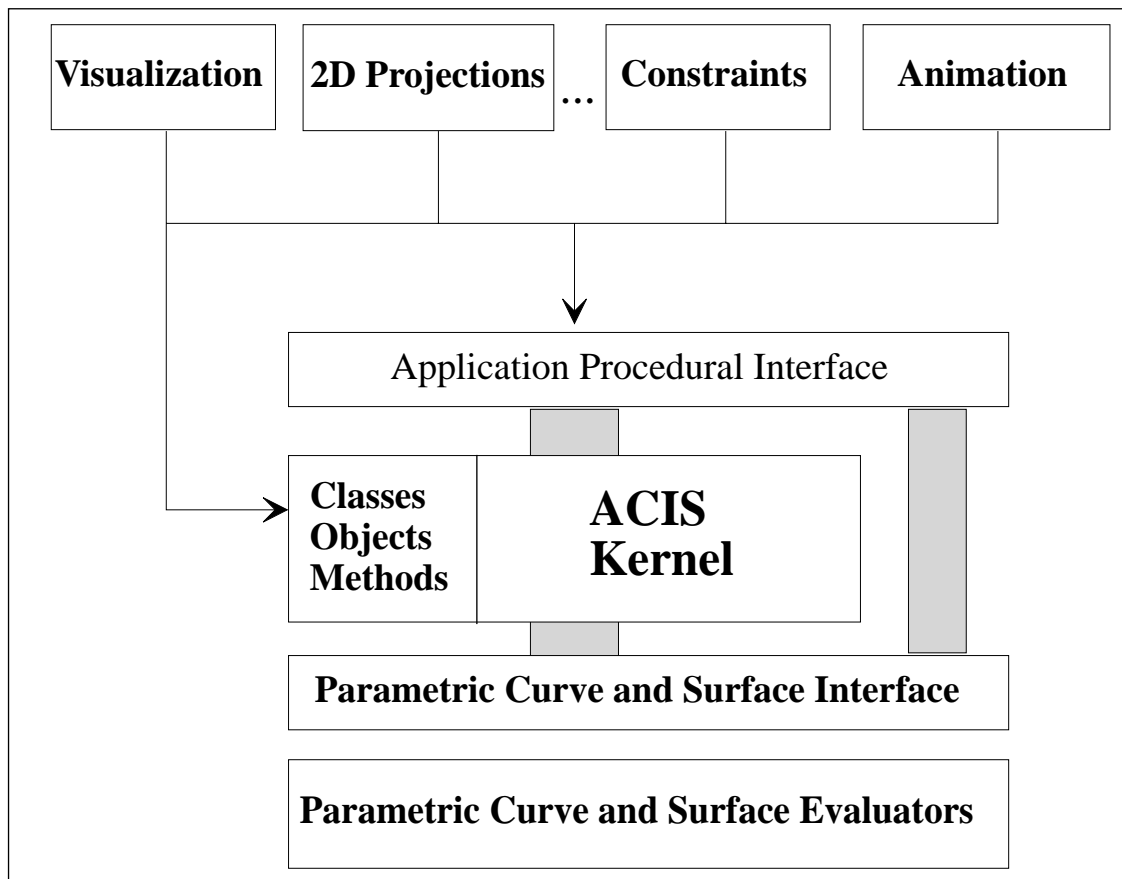


Bild 4.5 ACIS Husks Around the ACIS Kernel /88/

Um HUSKS zu erzeugen setzte man mit seinem in C oder C++ implementierten Programm auf dem ACIS-API oder den ACIS Klassen auf. Neben den geometrischen und mathematischen Funktionen sind Funktionen zur einheitlichen Fehlerbehandlung implementiert. Um Namenskonflikte zwischen den verschiedenen HUSKS zu vermeiden werden Kennwörter des Entwicklers in das Klassen- und das prozedurale Interface eingesetzt. Mit eigenen Programmen kann auf das API der HUSKS zugegriffen werden.

4.1.3 Die CV-DORS-Klassenhierarchie

Im Gegensatz zum ACIS-Kernel und seiner Implementierung wurde der CV-DORS Modellierer /96/ vor dem Hintergrund entwickelt, als Grundgerüst für das CAD-System CADDs /90/ zu dienen. Auf ihm sollten die verschiedensten Erweiterungen wie FE-Pakete, Expertensysteme und Simulationswerkzeuge aufsetzen und direkt mit der CADDs-Datenbank kommunizieren (siehe auch Bild 1.4). Der CV-DORS-Modellierer ist in der objektorientierten Programmiersprache C++ implementiert. Sein Interface gliedert sich in drei Teile. Ein Teil ist die FORTRAN-Schnittstelle, ein zweiter Teil die C-Schnittstelle. Der dritte Teil ist die Implementierung einer C++ - Schnittstelle. Alle Schnittstellen besitzen, mit kleinen Ausnahmen, den gleichen Leistungsumfang.

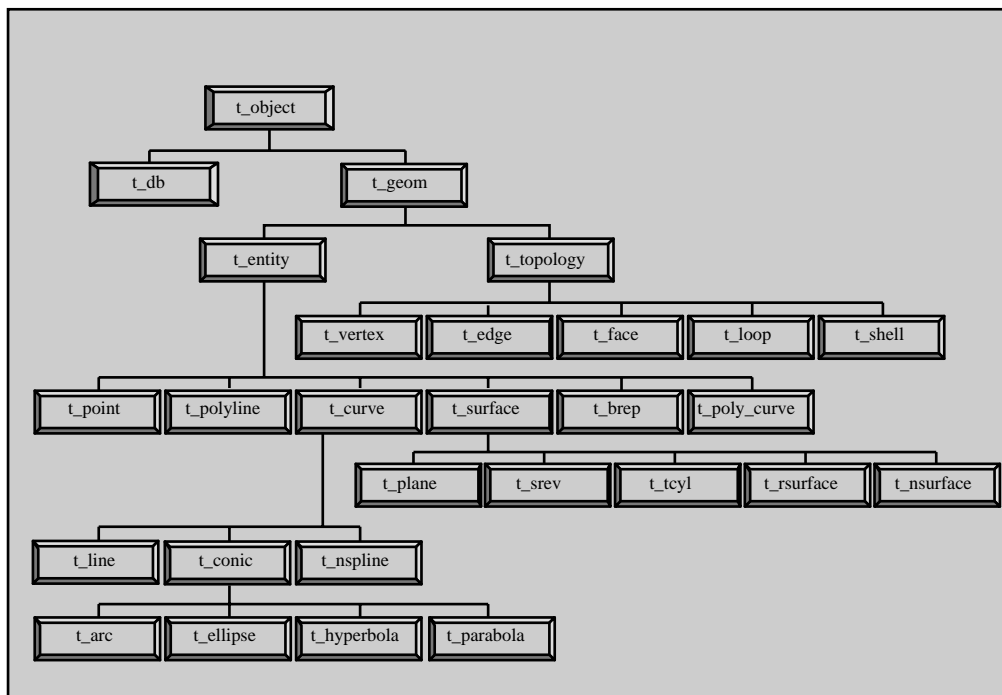


Bild 4.6 CV-DORS Objekthierarchie /96/

Wurzel der Objekthierarchie ist *t_object*. Es besitzt die Memberfunktion *error* zum Ermitteln des internen Status. Neben dieser Funktion sind lediglich der Konstruktor und Destruktor definiert. Von dem Objekt *t_object* sind alle weiteren abgeleitet. Diese gliedern sich in eine Datenbankklasse, eine Klasse zum Behandeln der globalen Systemvariablen, einer Geometrieklasse, einer Eigenschaftsklasse sowie einer Listenklasse und einigen weiteren Hilfsklassen, die mathematische Funktionen zum Behandeln von Vektoren, Matrizen und Koordinatensystemen

beinhalten. Bild 4.6 stellt den geometrischen Teil und die Datenbankklasse „t_db“ der Objekthierarchie dar. Die Klasse „t_geom“ ist der wichtigste und größte Teil der Klassenhierarchie. Sie ist der Kern, um eigene anwendungsspezifische Erweiterungen an CADD5 anzubinden.

4.1.3.1 Die Klasse Geometrie

Das mächtigste Objekt ist die Klasse *t_geom*. Sie dient der rechnerinternen Darstellung der Geometrie und gliedert sich in die Klassen *t_entity* und *t_topology*. Erstgenannte implementiert die geometrischen Darstellungen von Punkten, Polygonen, Kurven (Linie, Kegelschnitte und NSplines) und Flächen.

Als Flächen können Ebenen, Rotations- und Verschiebeflächen, Regelflächen aus zwei vorgegebenen Kurven und NURBS-Flächen verwendet werden. Das Geometrieobjekt besitzt die Memberfunktionen zum Ermitteln des zur Geometrie gehörigem Datenbankzeigers, Funktionen zum Setzen eines Attributzeigers sowie Funktionen zum Ermitteln und Setzen des Layers, eines Markers und eines Namens.

Die abgeleitete Klasse *t_entity* besitzt zudem Funktionen um das entsprechende geometrische Element in eine Datenbank zu schreiben, es aus einer Datenbank auszulesen, einen Namen zu setzen und auszulesen, den Geometriotyp zu ermitteln, es anzuzeigen, es zu kopieren sowie zu ermitteln, ob es Teil eines *Solid* ist.

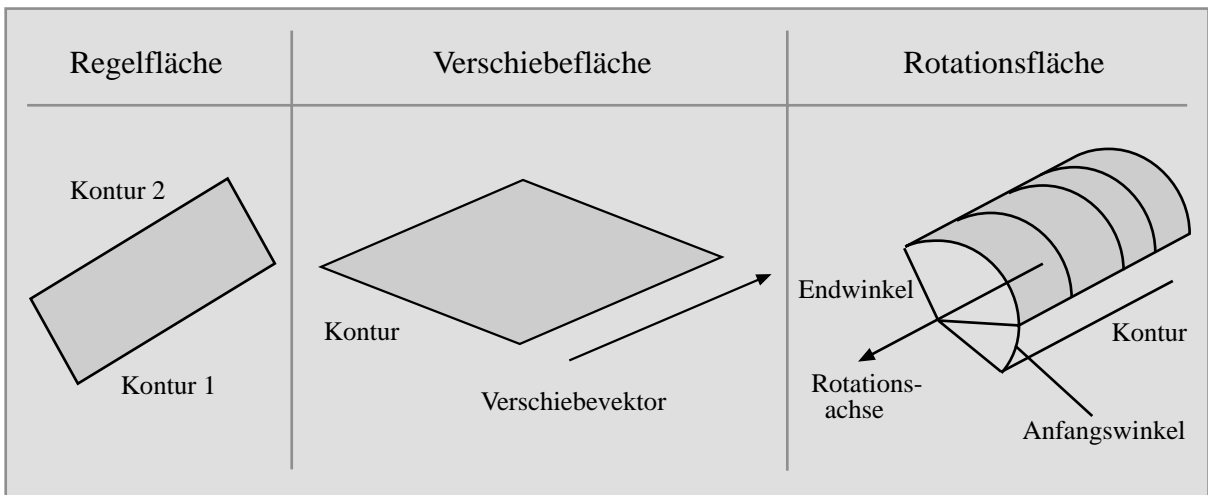


Bild 4.7 Methoden zur Erzeugung von Flächen

Von *t_entity* wurde die Klasse *t_curve* abgeleitet. Diese beinhaltet Funktionen zur Handhabung der in Parameterform dargestellten, von *t_curve* abgeleiteten Elemente *t_line*, *t_conic* und

t_nspline. Funktionen zur Schnittpunktberechnung, zum trimmen, zum durchlaufen, zum Erweitern sowie zur Offsetbildung und zur Ableitung sind in dieser Klasse implementiert.

Die entsprechende Klasse zur Bearbeitung von Flächen ist die Klasse *t_surface*. Von ihr sind die Klassen *t_plane*, *t_srev*, *t_cyl*, *t_rsurface* sowie *t_nsurface* abgeleitet worden. Bild 4.7 zeigt drei gebräuchliche Methoden zum Erzeugen von Flächen. Neben den verschiedenen Konstruktoren enthält die Klasse *t_surface* Funktionen zur Schnittlinienberechnung, Parameterberechnung, Ableitungsbildung und Projektion.

Mit der von *t_entity* abgeleiteten Klasse *t_brep* können die geometrischen Primitiven Zylinder, Quader, Kegel, Kugel, Torus, Prisma sowie Körper, die durch Verschieben bzw. Rotieren einer Kontur entstehen, erzeugt werden. Desweiteren können Körper aus Flächen zusammengesetzt (engl.: sew) werden.

Ein *t_brep* muss nicht geschlossen sein. Er kann auch eine offene Struktur (Shell) bilden. Daneben gibt es Funktionen zum Anbringen von Radien, Ausrundungen und Fillets. Die mengentheoretischen Funktionen des Subtrahierens, Addierens und der Schnittmengenbildung werden von der Memberfunktion *boolean* durchgeführt. Neben einer Flächenberechnung kann die Lage des Punktes zu einem *t_brep* ermittelt werden (innerhalb, ausserhalb oder auf der Oberfläche).

4.1.3.2 Die Klasse Topologie

Das Objekt *t_topology* sowie die aus ihm abgeleiteten Objekte *t_vertex*, *t_edge*, *t_face*, *t_loop* und *t_shell* stellen die logischen Verknüpfungen der geometrischen Elemente eines Körpers untereinander her. In der Klasse wurden Funktionen zum Ermitteln des zur Topologie gehörigen *t_brep*, des zugehörigen Layers sowie Funktionen zum Ermitteln der Kanten-, Flächen-, Ecken-, Loops- und Shelllisten implementiert.

Loops sind als eine Liste zusammenhängender und gerichteter Kanten definiert, die eine Fläche umschließen. Mit Shells wird eine Menge von logisch zusammengesetzten Flächen bezeichnet. Das nachfolgende Bild veranschaulicht die getroffenen Definitionen. Die Zusammenhänge und Abhängigkeiten der Loop, der Shell, der Face, der Kante und der Ecke werden verdeutlicht.

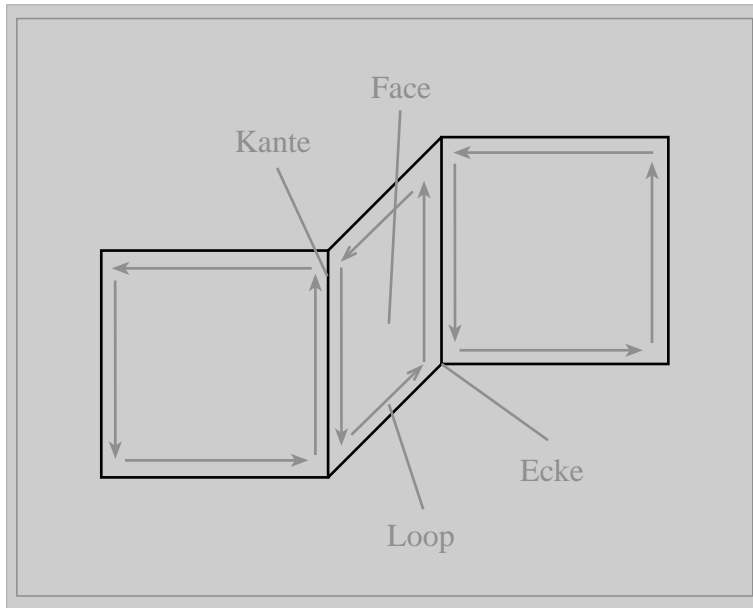


Bild 4.8 Offene Shell

Eine Kante begrenzt eine Fläche. In der Computergrafik wird oft der Begriff Halbkante verwendet. Unter Verwendung dieses Begriffes wird eine Fläche von einer gerichteten Menge von Halbkanten begrenzt. Zwei gegenläufige Halbkanten begrenzt. Zwei gegenläufige Halbkanten bilden eine „richtige“ Kante. Korrespondierend zum topologischen Begriff der Kante ist der geometrische Begriff der Kurve (Gerade, Kreis, Ellipse, Ellipsensegment, NURB, ...). Dem topologischen Begriff der Face entspricht der geometrische Begriff der Oberfläche (Ebene, Ellipsoid, NURB, ...).

4.1.3.3 Klassen zur Listenverwaltung

Eine weitere wichtige Klasse ist t_object_list . Von ihr sind die Klassen t_geom_list , t_ent_list , $t_poly_curve_list$ und t_triple_list abgeleitet. Sie dienen dazu, die entsprechenden Elemente in Listen zu verwalten. Häufige Verwendung finden die Listen bei der Schnittpunkt- und Schnittkurvenberechnung.

Auch beim Durchwandern der Topologie eines t_brep werden sie genutzt. Die Listen wurden in der Form der doppelt verketteten Listen implementiert. Neben den Zeigern auf den Nachfolger und Vorgänger besitzen sie einen Zeiger auf ein Datenelement. Das nachfolgende [Bild 4.9](#) veranschaulicht die Elemente der Listenklasse sowie die Verknüpfungen der Listenelemente mit den Datenelementen (Objekten).

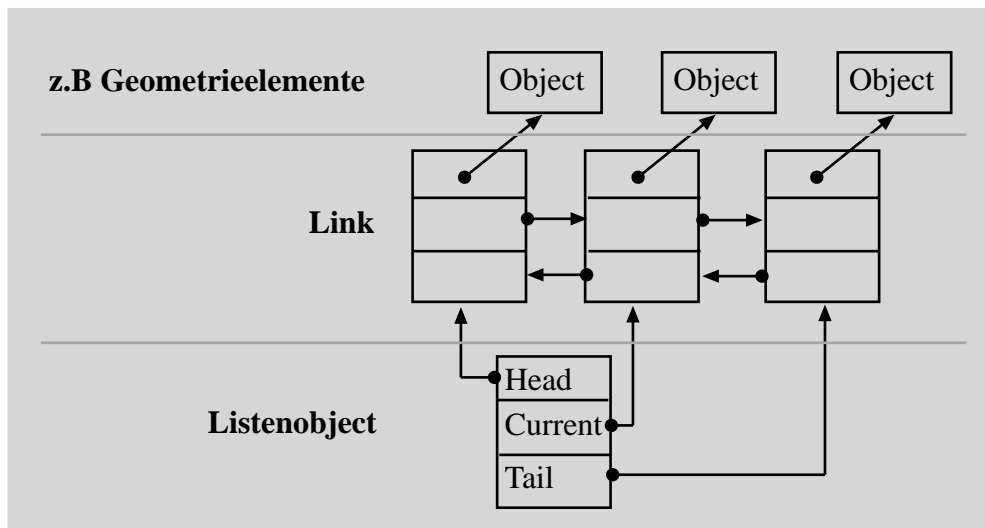


Bild 4.9 Der Aufbau doppelt verketteter Listen in CV-DORS /95/

Durch die Funktionen `head()`, `tail()`, `next()` und `previous()` wird ein Zeiger durch die Liste bewegt. Mit der Funktion `add()` kann ein neues Element vor oder nach dem aktuellem Zeiger eingefügt werden. `Remove()` entfernt das aktuelle Listenelement. Um das zum Link gehörige Datenelement zu ermitteln wird die Funktion `retrive()` genutzt. Mit der Funktion `total_links()` kann die Anzahl der Listenelemente ermittelt werden. Zum Durchlaufen einer Liste kann aber auch abgefragt werden, ob das aktuelle Listenelement der Kopf oder das Ende ist.

4.1.3.4 Die Datenbankklasse

Die Klasse `t_db` stellt Funktionen zur Bearbeitung von CADDs-Datenbanken zur Verfügung. Neben dem Öffnen, Schließen, Löschen und Anlegen von Datenbanken mit den verschiedensten Möglichkeiten sind besonders die Funktionen zum Durchsuchen einer geöffneten Datenbank von Interesse. Es werden die Möglichkeiten zur Verfügung gestellt, nur Elemente eines bestimmten Layers oder nur Elemente eines bestimmten Typs einzulesen.

4.1.3.5 Mathematische Funktionen

Die mathematischen Funktionen sind nicht in einer sondern in mehreren voneinander unabhängigen Klassen implementiert. So sind die Klassen `t_pair`, `t_triple`, `t_frame`, `t_vektor`, `t_xform` und `t_matrix` von der Klasse `t_objekt` abgeleitet. Die Klasse `t_pair` dient dazu, die Distanz zweier Punkte in der Ebene zu berechnen. Mit der Klasse `t_triple` werden Punktoperationen im Raum vorgenommen. So kann aus Vektoren, Feldern und drei Koordinatenwerten ein `t_triple`

Objekt erzeugt werden. Die entsprechenden Gegenfunktionen, das Ermitteln dieser Daten aus einem `t_triple` Objekt ist ebenfalls möglich. Implementiert sind die Klassenfunktionen zum Transformieren des durch `t_triple` definierten Punktes von einem Koordinatensystem in ein anderes. Eine Funktion zum Ermitteln der Distanz zweier Punkte im Raum ist ebenfalls implementiert. Um mit Koordinatensystemen zu arbeiten wird die Klasse `t_frame` genutzt. Neben sechs Konstruktoren sind Funktionen zum Zuweisen, zum Ermitteln der Koordinatenachsen und des Koordinatensystems sowie zum Verschieben des Koordinatensystems implementiert. In den Klassen `t_matrix` und `t_vektor` sind die mathematischen Funktionen der Matrix- und Vektorrechnung realisiert. Dabei wurden einige Sonderfunktionen zur Bearbeitung von zwei- und dreidimensionalen Vektoren und Matrizen implementiert. Diese beiden Klassen werden durch die Klasse `t_xform` ergänzt. In ihr sind die Funktionen zum Spiegeln, Verschieben, Rotieren und Skalieren ausgeführt. Die Klasse arbeitet mit vierdimensionalen quadratischen Matrizen. Um eine Verschiebung durchzuführen wird ein Object der Klasse `t_xform` definiert: `t_xform mein_xform()`; Ein Verschiebevektor kann durch `t_vector verschiebung(1.0,2.0,0.0)` definiert werden. Die Erstellung der Verschiebmatrix erfolgt durch: `mein_xform.translate(verschiebung)`; Ein Punkt, der durch `t_point mein_punkt(10.0,10.0,0.0)` definiert wurde, kann jetzt durch die Operation `mein_punkt.xform(mein_xform)` verschoben werden. Sollen mehrere Verschiebe-, Rotations-, Spiegel- und Skalieroperationen durchgeführt werden, so können sie zu einer Matrix kombiniert und anschliessend auf ein geometrisches Objekt angewandt werden.

4.1.4 Grafische Programmierschnittstellen

CAD-Systeme nutzen zur Ausgabe ihrer rechnerinternen Daten Programmbibliotheken. Zur Ansteuerung der Peripheriegeräte, Bildschirm mit Grafiksystem, Drucker und Plotter gibt es Systeme wie das GKS, PHIGS, OpenGL oder HOOPS (Hierarchical Object Oriented Picture System). Die Systeme sind unabhängig von der speziellen Anwendung CAD, sie werden in vielen Bereichen der Technik und Medizin zur Visualisierung von Daten eingesetzt. Ihre Systemfunktionalität wird dem Anwender über ein in mehreren Programmiersprachen verfügbares API zur Verfügung gestellt. Die Plattformunabhängigkeit ist ein weiteres Merkmal der Systeme. Meist steht der vollständige Funktionsumfang auf den verschiedensten Hardware-

plattformen vom PC über Workstations bis hin zu Supercomputern zur Verfügung. Einige Systeme sind skalierbar. Je nach zugrundeliegender Hardware ist ihr Befehlsumfang eingeschränkt. Die Erweiterbarkeit von Grafiksystemen um neue Objekte stellt ein weiteres Qualitätsmerkmal dar.

Das **GKS** /7,115/ ist die erste genormte Grafikschnittstelle. Die ISO 7942 beschreibt ihre Funktionen. Sie beschränkt sich zunächst auf die zweidimensionale Grafik. Leider hielt die Entwicklung des GKS nicht den rasch steigenden Anforderungen aus Industrie und Wirtschaft stand. Dabei wurden im GKS erstmals Konzepte realisiert, die in der Computergrafik richtungsweisend waren. So wurde das Konzept eines virtuellen grafischen Gerätes, das als Workstation bezeichnet wurde, implementiert. Dies konnte in verschiedenen Leistungsstufen in das GKS eingebunden werden. Die Workstations wurden in Ein- und Ausgabegeräte, für die temporäre Sicherung von Segmenten zuständige Geräte und für den Import und Export von Metafiles geeignete Geräte eingeteilt. Eine weitere Neuerung waren die Segmente. In diese konnten alle grafischen Primitive eingefügt werden, die ausgegeben werden sollten. Sie entsprachen der Realisierung der Display-Listen. Ein Mangel der Segmente betraf deren Editierbarkeit. Ausser dem Zufügen von Elementen in ein geöffnetes Segment waren keine weiteren Funktionen auf ein geöffnetes Segment erlaubt. Es konnten auch keine weiteren Segmente aus einem Segment referenziert werden. Die grafischen Primitive Polygon, Polymarke, Text, Füllgebiet, Zellmatrix und ein verallgemeinertes Darstellungselement wurden unterstützt. Die Entwicklung und Normung von GKS-3D erfolgte 1988 durch die ISO 8805.

Das **PHIGS** /8,15,115/ Grafiksystem wurde zur gleichen Zeit wie das GKS entwickelt. Im Gegensatz zu diesem wurde es von Beginn an als 2D und 3D System konzeptioniert. Es wurde 1989 durch die ISO/IEC 8592 genormt. Das „Programmer’s Hierarchical Interactive Graphics System“ folgte einer vorteilhafteren Philosophie als das GKS und erweiterte dessen Konzepte zum Teil erheblich. Der Schwerpunkte der Veränderungen betraf die Segmentstruktur des GKS. PHIGS unterstützte die Erzeugung und Modifizierung hierarchischer grafischer Strukturen. Kern dieser Neuerung sind PHIGES Strukturen. Sie bestehen aus Funktionen zur Generierung der grafischen Primitiven als auch aus Referenzen auf andere Strukturen. Weiterhin sind die Strukturen editierbar. So können in den Listen enthaltene Elemente gelöscht, modifiziert oder ersetzt werden.

OpenGL /8,50/ wurde von der Firma Silicon Graphics entwickelt. Die Firma benötigte für ihre Workstations eine leistungsfähige Software im Bereich der Animation und photorealistischen Darstellung. OpenGL entwickelte sich rasch zum Industriestandard und wurde auf die verschiedensten Hardwareplattformen portiert. Neben den UNIX- Systemen werden das MacOS, Windows 95/98/2000/NT, Linux und das BeOS unterstützt. Das Ausführungsmodell basiert auf dem Client/Server Modell. Das Anwendungsprogramm (Client) sendet Kommandos. Diese werden von OpenGL (Server) interpretiert. Hierbei spielt es keine Rolle, ob Client und Server auf einem oder auf verschiedenen Computern ablaufen. Das Programminterface kann über die Sprachen C, C++, Fortran, ADA und Java angesprochen werden. OpenGL selbst folgt einem prozeduralen Programmierstil. Es erlaubt den direkten Zugriff auf Transformationsmatrizen, Gleichungen zur Lichtberechnung können direkt manipuliert werden, Antialiasing Methoden können bestimmt werden und Methoden zum Anzeigen von Pixelfeldern sind direkt beeinflussbar. Die grafischen Primitiven Punkt, Liniensegment und Polygon werden über sogenannte Vertexes (Scheitelpunkt) oder Gruppen von Vertexen definiert. Den Vertexen können Attribute wie die Farbe, Koordinaten und Eckenverweise zugeordnet werden. Das nachfolgende Bild zeigt, wie OpenGL die zu interpretierenden Kommandos abarbeitet:

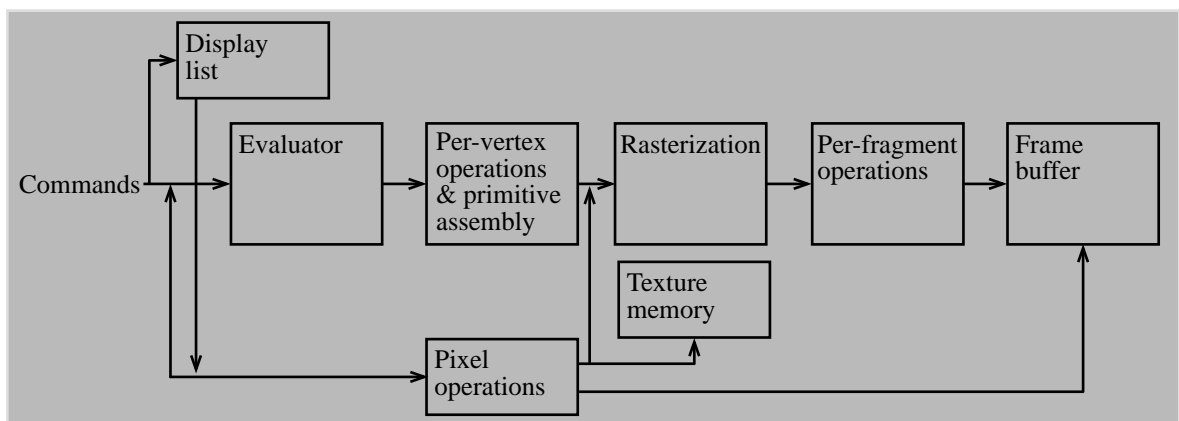


Bild 4.10 Die OpenGL Prozesse zu Visualisieren /50/

Trifft ein Befehl ein, kann er sofort abgearbeitet oder in einer Displayliste gespeichert werden. Der Evaluator dient dazu, Kurven und Flächen in Polynome umzurechnen. Im nachfolgenden Schritt werden die Vertexen transformiert und geclipped. Die Rasterization transformiert die vorhandenen dreidimensionalen Daten in zweidimensionale.

```
glNewList(10, GL_COMPILE);           // Displayliste oeffnen
glColor3f(1,0,0);                   // Farbe setzen : Rot
    glRecti(164,33,364,600);         // Rechteck definieren
glEndList();                          // Displayliste schliessen

glNewList(20, GL_COMPILE);           // Neue Displayliste oeffnen
    glColor3f(1,0,0)
    glPolygonMode(GL_BOTH, GL_LINE);
    glBegin(GL_POLYGON);             // Kreis als Polygon definiert
        for(i=0; i <100; i++) {
            cosine = 25 * cos(i * 2 * PI/100.0);
            sine = 25 * sin(i * 2 * PI/100.0);
            glVertex2f(cosine,sine);
        }
    glEnd();                          // Alle Kreispolyg. sind definiert
    glBegin(GL_QUADS);
        glColor(0,1,1);
        glVertex2i(100,100);
        glVertex2i(100,200);
        glVertex2i(200,200);
        glVertex2i(100,200);
    glEnd();
glEndList();

glNewList(30, GL_COMPILE);
    glCallList(10);
        glColor(1,1,1);
        glRecti(400,100,500,300);
    glCallList(20);
glEndList();
```

Bild 4.11 OpenGL Displaylistenstruktur /48/

Die Per-fragment operations beinhalten im wesentlichen Operationen zur logischen Bitmanipulation. Displaylisten sind das zentrale Strukturierungsmittel von OpenGL.

Bild 4.11 zeigt die Verwendung einer Displayliste. In Displaylisten werden sowohl die grafischen Primitiven als auch die Attribute eingefügt. Durch das Verweisen auf Displaylisten innerhalb von Displaylisten können hierarchische Modelle aufgebaut werden. Das Hinzufügen von eigenen Attributen und Objekten ist jedoch nicht möglich.

Haupteinsatzgebiet von **HOOPS** /8,14,93,94/ ist der technisch - wissenschaftliche Bereich. Zahlreiche CAD-Systeme wie z.B. CADD5, die Systeme von Strässle und Nestler sowie AutoCAD setzen auf HOOPS auf. HOOPS gliedert sich in vier Teile, das HDI (HOOPS Device Interface), das HOOPS Graphic API (Application Programming Interface), das GUI

(Graphic User Interface) sowie die HOOPS Datenbank. Die wirtschaftlichen Anwendervorteile von HOOPS liegen in der breiten Unterstützung einer Vielzahl von Rechnerarchitekturen und der vollständigen Portabilität zwischen den einzelnen Rechnerarchitekturen.

Das nachfolgende Bild gibt einen Überblick über die unterstützten Hardwareplattformen, deren Betriebssysteme und die von den Betriebssystemen unterstützten grafischen Schnittstellen.

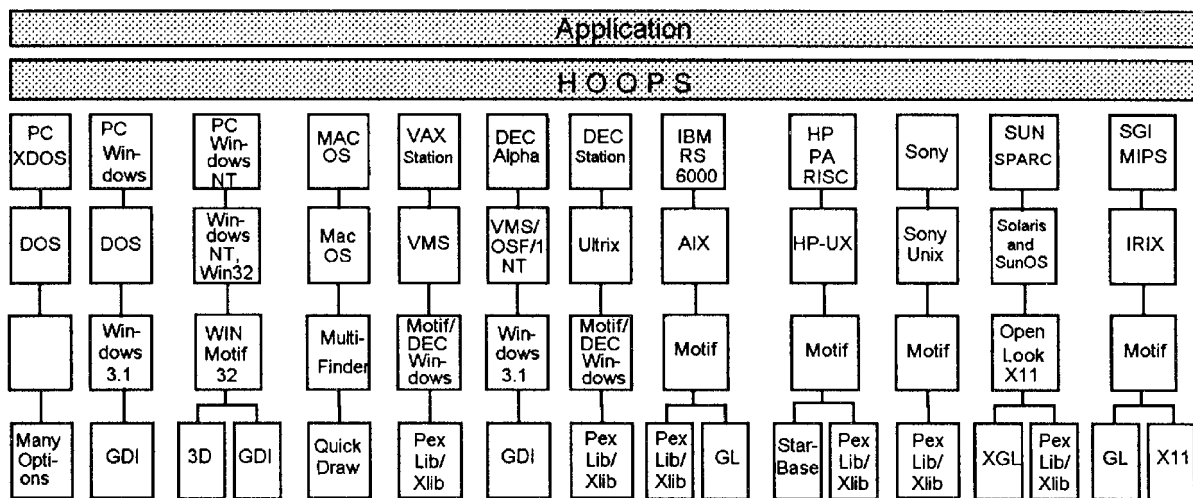


Bild 4.12 Von HOOPS unterstützte Rechnerplattformen /14/

Für Anwendungsprogrammierer zeichnet sich HOOPS durch seinen deklarativen Programmierstil und eine leicht verständliche Segmentstruktur aus. Segmente sind die grundlegenden Einheiten der Strukturierung der Daten. Segmente können ineinander verschachtelt werden, so daß Hierarchien aufgebaut werden können. Segmente vererben hierbei ihre Attribute an untergeordnete Segmente. In Segmenten können die geometrischen Primitiven, Kameras, Lichter, Darstellungsattribute sowie Benutzerinformationen abgelegt werden. Zu den Attributen gehören auch Windows. Das Wurzelsegment beinhaltet definitionsgemäß ein Window mit den Koordinaten (-1.0, -1.0) für die „linke untere Ecke“ und (1.0, 1.0) für die „rechte obere Ecke“. Ein Window wird mit dem Befehl `HC_Set_Window(XMin,XMax,YMin,Ymax)` definiert. In diesem wird das zugehörige Segment abgebildet.

Zu den geometrischen Grundelementen gehören neben dem Marker, Linie, Polyline, Kreis, Ellipse und Polygon auch komplexere Elemente wie eine Shell, ein Mesh, ein Image und Text sowie Lichtquellen. Bevor diese Elemente erzeugt werden, müssen ihre Attribute definiert wer-

den. Dies sind die geometrischen Transformationen, die Farbe und die Sichtbarkeit. Die Abbildung des dreidimensionalen Objektraumes auf den zweidimensionalen Bildraum erfolgt durch Kameras. Hierbei kann zwischen den Möglichkeiten der perspektivischen, der orthogonalen und der gestreckten (ist eine orthogonale Projektion, bei der das Sichtfeld exakt in das zugehörige Segmentwindow paßt) Projektion unterschieden werden. Das nachfolgende Bild zeigt ein Beispiel zum besseren Verständnis der getroffenen Definitionen.

```
int main(){
float q[3],r[3],s[3];
q[0]=0.0;q[1]=0.0;q[2]=0.7;
r[0]=0.0;r[1]=-0.7;r[2]=0.0;
s[0]=0.7;s[1]=0.0;s[2]=0.0;
HC_Open_Segment(,?Picture");
  HC_Open_Segment(,Drahtmodell");
    HC_Set_Window(-1.0,0.0,-1.0,0.0);
    HC_Insert_Circle(q,r,s);
    HC_Set_Visibility(,faces=off");
  HC_Close_Segment();
  HC_Open_Segment(,Flächenmodell");
    HC_Set_Window(0.0,1.0,-1.0,1.0);
    HC_Open_Segment(,Streifen");
      HC_Set_Window(-1.0,1.0,0.33,1.0);
      HC_Insert_Circle(q,r,s);
      HC_Set_Face_Pattern(,==");
    HC_Close_Segment();
    HC_Open_Segment(,Karos");
      HC_Set_Window(-1.0,1.0,-1.0,-0.33);
      HC_Insert_Circle(q,r,s);
      HC_Set_Face_Pattern(,,<>");
    HC_Close_Segment();
    HC_Insert_Circle(q,r,s);
  HC_Close_Segment();
HC_Close_Segment();
HC_Pause();
return(0);
```

Bild 4.13 HOOPS-Objekthierarchie /14/

Zu erwähnen bleibt noch die komfortable Eingabetechnik von HOOPS. Hierbei wird zwischen den Eingabetypen Taste, String, Locator, Auswahl und Timer unterschieden. Alle zugelassenen Ereignisse werden in der Reihenfolge ihrer Ankunft in einer Warteschlange verwaltet. Von dort wird das „älteste“ Ereignis aus der Schlange entnommen und verarbeitet.

Eine Neuentwicklung ist das Grafiksystem **QuickDraw 3D** /86/ der Firma Apple. Seine Anwendungsbereiche sind nicht Workstations sondern High-End PCs, die zunehmend in grafisch intensiven Arbeitsbereichen eingesetzt werden. QuickDraw 3D deckt ein großes Einsatzgebiet, die interaktive 3D-Modellierung, die Darstellung von 3D-Objekten, die Animation sowie die Datenvisualisierung ab. Es besteht aus zwei Komponenten. Eine ist ein sogenannter Viewer. Er dient zur Visualisierung von 3D-Modellen und besitzt eine Funktionalität, die Zoom-, Verschiebe- und Rotationsfunktionen sowie als Darstellungsmethoden verschiedene Schattierungsmethoden und ein Hidden-Line Verfahren umfasst. Diese Funktionalität ist bereits in einige Web-Browser integriert worden. Die andere Komponente ist das API sowie deren implementierten Funktionen. Die grafischen Funktionen von QuickDraw3D sind darauf ausgerichtet /92/ :

- die Grafikfunktionen leicht erweiterbar zu gestalten, d.h. es können eigene Renderer integriert werden,
- Geometrien um benutzerdefinierte Daten zu erweitern,
- ein Cross-Plattform-Tool, das zur Zeit die Betriebssysteme Windows 95/98/NT, MacOS sowie Rhapsody (Apple Unix) unterstützt,
- Hardware-Beschleuniger einzusetzen,
- ein plattformübergreifendes umfassendes Meta-File-Format zu schaffen.

Diese Anforderungen führten zu den im nachfolgendem Bild dargestellten funktionalen Komponenten von QuickDraw 3D. Das Anwendungsprogramm greift auf die einzelnen voneinander unabhängigen Komponenten zu. Derzeit besteht die Möglichkeit, die Dateioperationen sowie die Grafikattribute an die eigenen Bedürfnisse anzupassen, d.h. sie zu erweitern. In zukünftigen Versionen ist geplant, die Geometrieelemente , die Schnittstelle zur Grafikhardware, den Darstellungsmodul und den Schattierungsmodul erweiterbar zu gestalten. Dies gestattet es dem Anwender auf ein sehr flexibles und leicht anpassbares System zugreifen zu können.

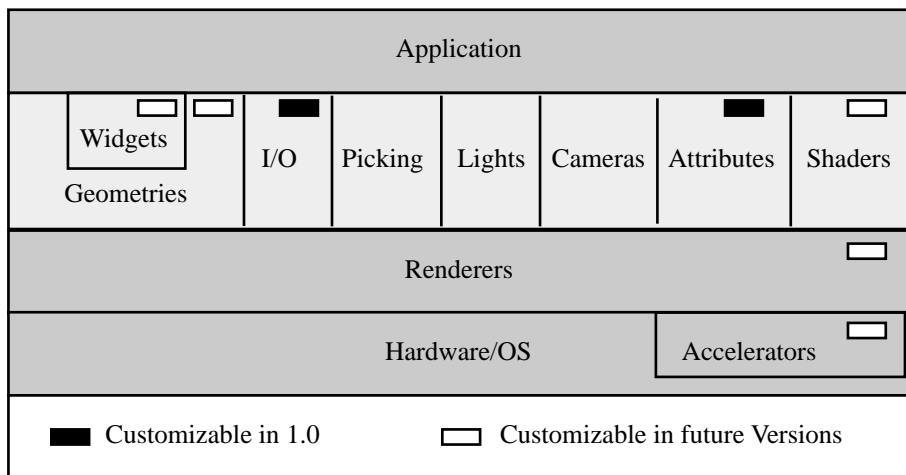


Bild 4.14 Quick Draw 3D funktionale Komponenten /86/

Implementiert wurde die Programmbibliothek in der Programmiersprache „C“. Das API sowie ihr innerer Aufbau gestaltet sich objektorientiert. Alle Klassen sind in eine hierarchische Struktur eingeordnet. Die Methoden der Vererbung sowie des Überschreibens von Funktionen werden unterstützt. Wurzel des Objektbaums ist das Objekt selbst. Es stellt die Methoden des Erzeugens, Zerstörens, des Duplizierens, des Submit (Verfügbar seins) sowie verschiedener Typinformationsfunktionen zur Verfügung.

Abgeleitet wurden die Objekte Element, Pick Objekt, Shared Objekt und View Objekt. Elementobjekte werden genutzt, um Applikationsspezifische Daten aufzunehmen und in einem Mengenobjekt zu verwalten. Das Pick Objekt dient dazu, Informationen die beim Identifizieren (Auswählen eines Elementes mit einem Cursor) eines Elementes notwendig sind zu verwalten. Mächtigstes Objekt ist das Shared Objekt. Von ihm sind Klassen zum Dateibearbeitung, zur Texturbearbeitung, zur Mengenverarbeitung(Set), Darstellung, Eingabe und Gemometrie abgeleitet.

Im View Objekt sind alle Informationen zusammengefasst, die das Erscheinen sowie die Position von Objekten zur Darstellungszeit beeinflussen. Das nachfolgende Bild zeigt den vollständigen Objektbaum.

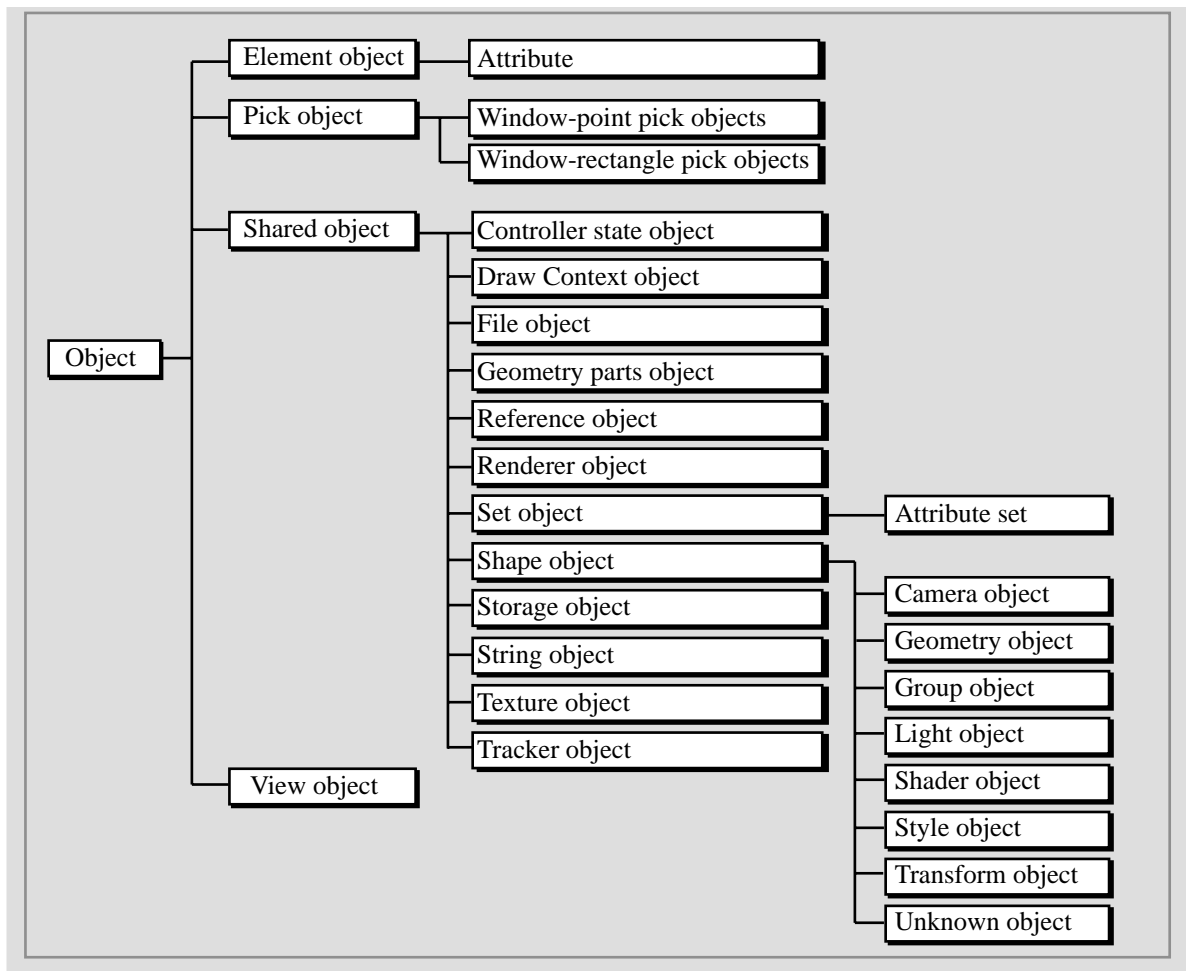


Bild 4.15 Quick Draw 3D Objekthierarchie /86/

Zu diesen aus dem kommerziellen Umfeld bekannten Systemen existieren einige neue Ansätze, die sich fast alle im Forschungsstadium befinden. Sie sind vorwiegend objektorientierter Natur.

Zu ihnen zählen die Systeme GEO++, GROOP, YART, IRIS INVENTOR und GRAMS. Die Systeme sind in objektorientierten Sprachen, meist C++ implementiert und besitzen auch nur eine objektorientierte Sprachanbindung. Trotz dieser guten Voraussetzung wird ein wesentliches Ziel, ein offenes und erweiterbares Grafiksystem zu erstellen nicht immer erreicht.

4.1.5 Benutzeroberflächen

Moderne Benutzeroberflächen sind Ereignisorientiert. Sie beruhen auf dem Modell-View-Controller (MVC) Prinzip, das in den siebziger Jahren /136/ entwickelt wurde. Die am meisten verwendeten Fenstersysteme wie die Microsoft Windows Familie, das MacOS sowie alle Desktop Aufsätze der UNIX-Versionen beruhen auf diesem Konzept. Die Desktop Aufsätze der UNIX-Workstations basieren auf dem X-Window Modell. Es wurde am MIT entwickelt. Durch die ISO ist es genormt.

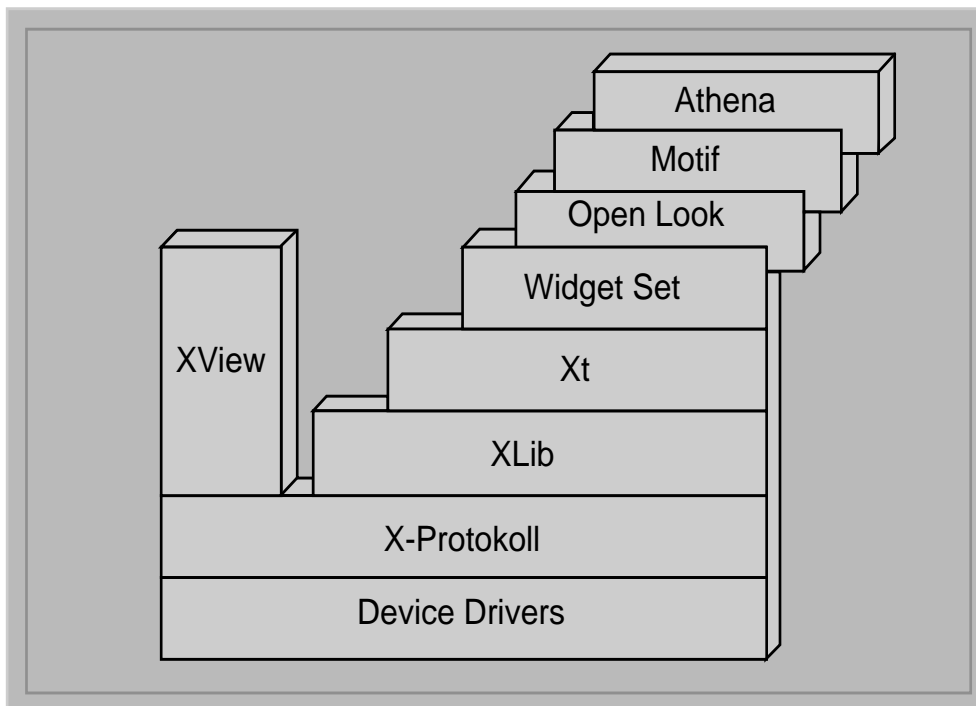


Bild 4.16 Das X-Window Modell /103/

Bild 4.16 zeigt den aus einzelnen, aufeinander basierenden Komponenten bestehenden Aufbau. Als Widget Set existieren „OPEN LOOK“, „Athena“ , „MOTIF“ und der „CDE“. Der CDE Widget Set hat sich als Standard durchgesetzt. Mit dem CDE (Common Desktop Environment) existiert für alle UNIX-Plattformen eine gleiche Benutzerschnittstelle. Moderne CAD-Systeme setzen mit dem Design ihrer Benutzerschnittstelle darauf auf. Sie ist in mehrere Bereiche unterteilt. Optischer Mittelpunkt ist das grafische Fenster, in dem die RID visualisiert wird.

Um dieses herum sind mehrere Befehlsfenster mit unterschiedlichen Befehlsgruppen angeordnet. In diesen wiederum befinden sich Buttons, Pull-Down- oder Pop-Up Menüs durch die der Konstrukteur Befehle ausführen kann.

Einige Systeme besitzen zusätzlich eine Kommandozeile, in der der Konstrukteur Befehle direkt absetzen kann. Mit dem Befehl:

„INSert LINE HORIZONTAL LENGTH 20 : Model Loc dd;“

wird in dem CAD System CADD5 beispielsweise eine horizontale Linie der Länge 20 in ein Modell eingefügt. Diese „Sprachen“ sind für einen Ungeübten auf den ersten Blick sehr komplex. Hat man einen gewissen Satz an Befehlen gelernt, so kann man schnell neue Befehle aus den bereits gelernten ableiten. Die Befehlseingabe „von Hand“ ist nach einer gewissen Einarbeitungszeit wesentlich effizienter als die Befehlseingabe über Buttons, Dialogfelder und Menüs.

Vorteil der letztgenannten Methode ist die Möglichkeit, die zur Eingabe benötigten Parameter grafisch in einem Anwendungsfall darstellen zu können. Beide Methoden haben ihre Vor- und Nachteile. Grundsätzlich sollte eine Methode durchgängig unterstützt werden. Der dauernde Wechsel zwischen Maus- und Tastatureingaben ist die schlechteste Lösung. Da jeder Benutzer eine eigene, bevorzugte Arbeitsweise hat, findet hier keine Bewertung darüber statt, welche Arbeitsweise die „richtige“ ist. Eine Koexistenz beider Methoden scheint aber die sinnvollste zu sein.

Wesentlich entscheidender ist, wie sich der Konstrukteur seine eigene Arbeitsumgebung gestalten kann. Hier bietet z.B. das CAD-System CADD5 die Möglichkeit, „Buttons“ zu verschieben. So können oft benötigte Befehle in einem Fensterbereich abgelegt werden. Befehlsfolgen können zusammengefasst und auf Buttons abgelegt werden. Diese können dann mit beliebig zu gestaltenden Piktogrammen belegt werden.

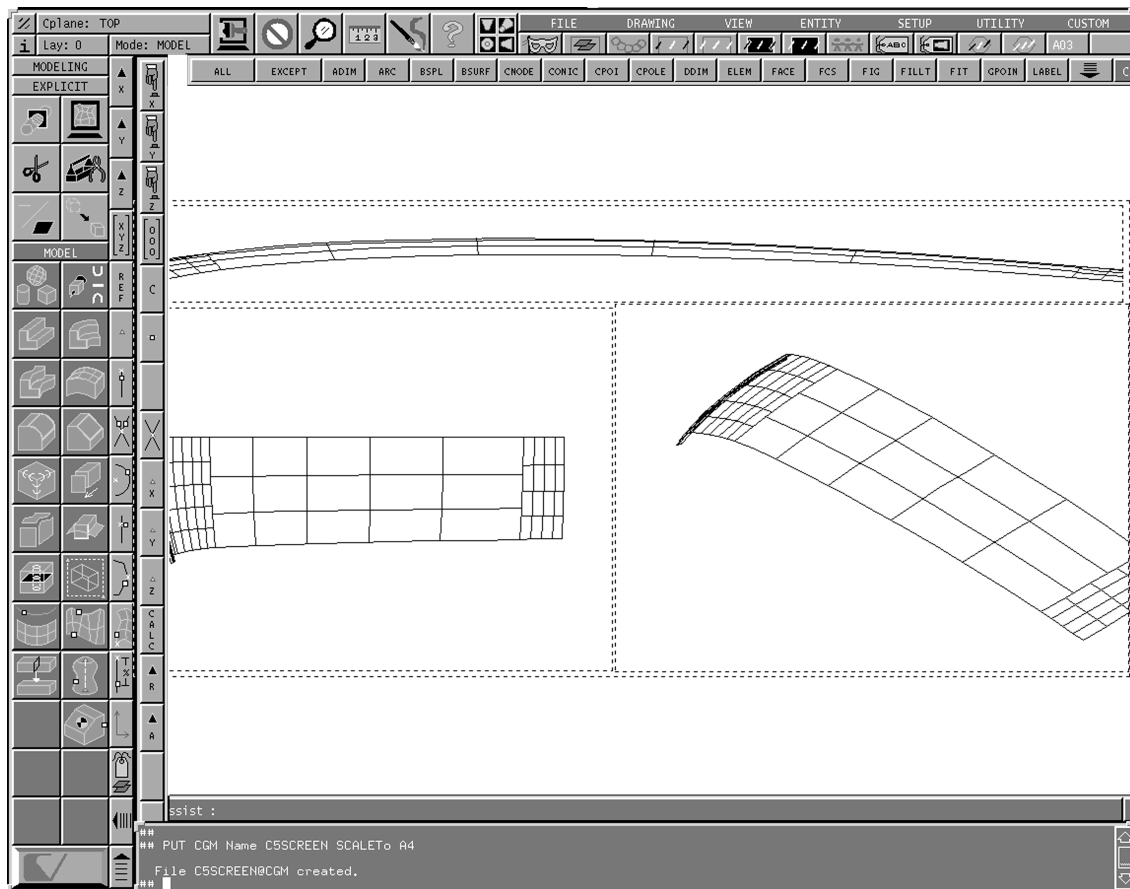


Bild 4.17 CADD55 Benutzeroberfläche

Bild 4.17 zeigt die Benutzeroberfläche von CADD55. Im linken unteren Menübereich ist die Funktionsgruppe „MODEL“ selektiert. Sie bietet die grundlegenden Funktionen zum Erzeugen und Bearbeiten von Körpern. Die unteren fünf Buttons können noch mit eigenen, oft benötigten und diese Funktionsgruppen ergänzenden Funktionen belegt werden.

4.1.6 Erweiterungen

Die Anbieter von CAD-Systemen bieten zu ihren Systemen Erweiterungen an. Typische Erweiterungen sind FE-Pakete und NC-Module. Diese greifen direkt auf die Datenstruktur des Modellierkerns zu. Eine Konvertierung von Daten über eine Geometrieschnittstelle entfällt. In der nachfolgende Tabelle sind die Zusatzmodule von CADD55 aufgelistet.

Tabelle 1: Zusatzmodule zu CADD5 /90/

Interference Checking	Milling	Systems Lab
CV GP	Physical Properties	Mechanical Simulation Package
CV NC	Design Optimization	Rapid Prototyping Interface
Turning	Parametric Tolerance Modelling	Advanced Wire EDM
Fabrication	Table Driven Design	CV 3D Harness Design (Kabelbaum)
3D Harness Layout	Stress Lab	CV Electrical System Engineer
Plastic Lab	Therm Lab	

Der Konstrukteur wird in wesentlich weiteren Bereichen als nur der geometrischen Modellierung unterstützt. Er kann sein Modell sofort berechnen oder eine Überführung der Daten in NC-Programme aus seiner gewohnten Programmumgebung durchführen. Zum Teil standardisierte Konstruktionsmethoden wie beispielsweise die Kabelbaumkonstruktion werden unterstützt.

4.1.7 Software-Schnittstellen

Die von einer Konstruktionsabteilung für die verschiedenen Aufgaben genutzten Softwarepakete stammen meist von verschiedenen Anbietern. So ist es üblich, Daten aus einem CAD-System in ein FE-Programm oder ein CAP-Programm zu übertragen. Durch die gegebene Wirtschaftsstruktur findet ein reger Datenaustausch von Geometriedaten zwischen Zulieferern und ihren Auftraggebern statt. Diese benutzen nicht immer die gleichen Programmpakete. Standardisierte Schnittstellen zum Datenaustausch sind IGES (Initial Graphical Exchange Specification), VDAFS, DXF und SET.

Tabelle 2: Geometrieschnittstellen und ihr Anwendungsgebiet /90/

IGES	Regelgeometrie
VDAFS	Freiformflächengeometrie
DXF	Allgemeine Geometriedaten
SET	Allgemeine Geometriedaten
STEP	Produktdaten

Die Datenübertragung ist leider nicht immer verlustfrei [71,72 und 134]. So werden die genormten Standards in verschiedenen CAD-Systemen im Rahmen der Norm unterschiedlich interpretiert. Ein weiterer Verlustfaktor sind die unterschiedlichen Systemphilosophien. 3D-Systeme benutzen Views, um ihr internes Datenmodell auf dem Bildschirm darzustellen. Werden diese Daten über Schnittstellen in ein anderes System eingegeben, das das Konzept der Views nicht „kennt“, so entstehen Fehlinterpretationen. Im nachfolgenden Bild ist der Prozeß des Einlesens von VDAFS Daten dargestellt. Im VDAFS-Format abgelegte Geometriemodelle lassen sich aufgrund der eindeutig definierten Spezifikation des VDAFS-Formats sehr gut zwischen den verschiedensten CAD-Systemen austauschen.

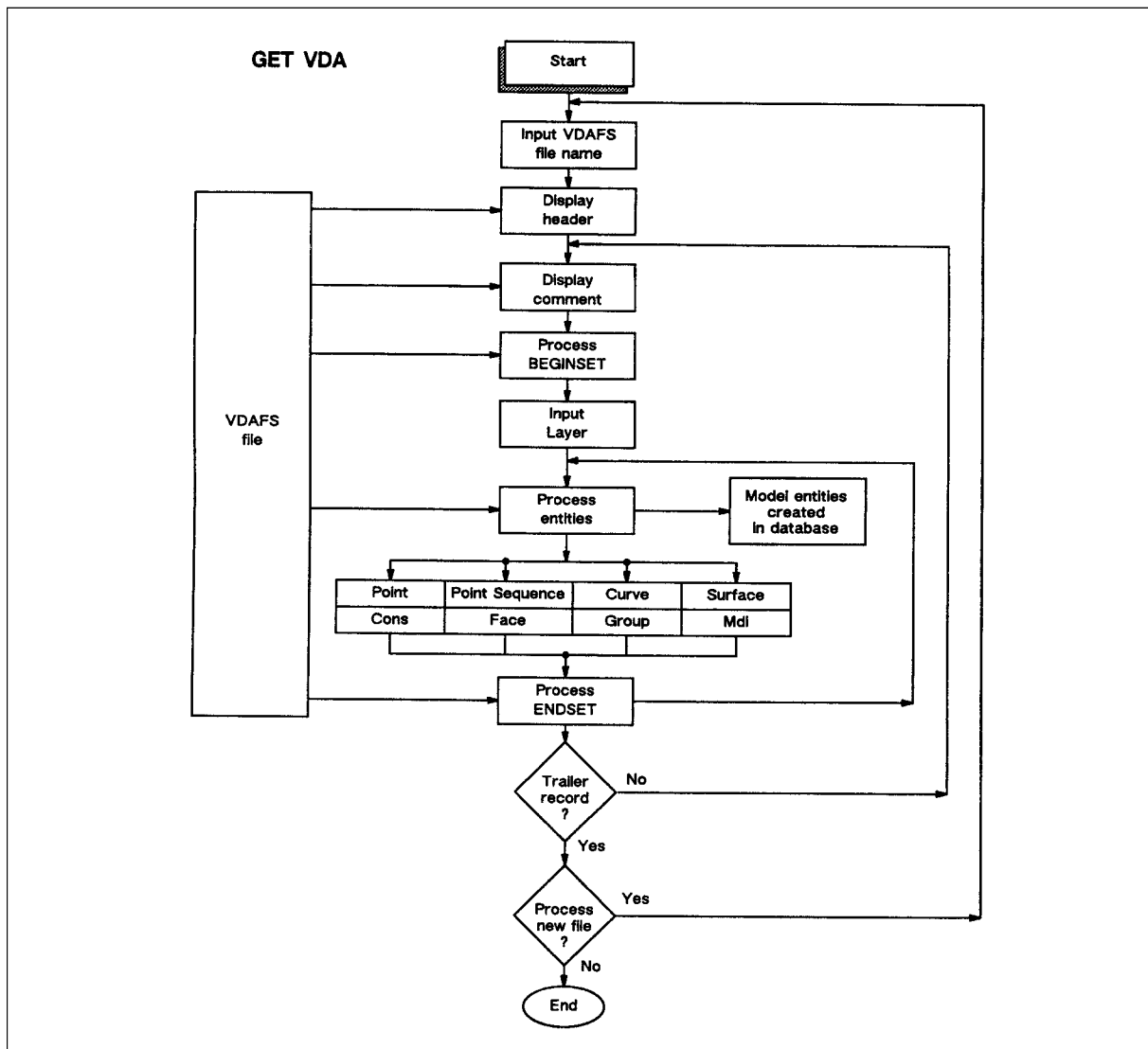


Bild 4.18 Der Prozeß des Einlesens von VDAFS-Daten [96/

Die Menge der über die VDAFS-Spezifikation ausgetauschten Daten hat bereits einen Anteil, wie in Bild 4.19 dargestellt, von 30% am gesamten Datenaustauschvolumen erreicht. Weitere 8% des Datenaustauschvolumens werden über die VDAIS-Schnittstelle abgewickelt. „Marktführer“ ist eine der ältesten Spezifikationen, das IGES-Format. Es wurde mit der Version 1.0 im Jahre 1981 als ANSI-Norm Y 14.26 M /140/ veröffentlicht. Die aktuelle Version besitzt die Nummer 5.1.

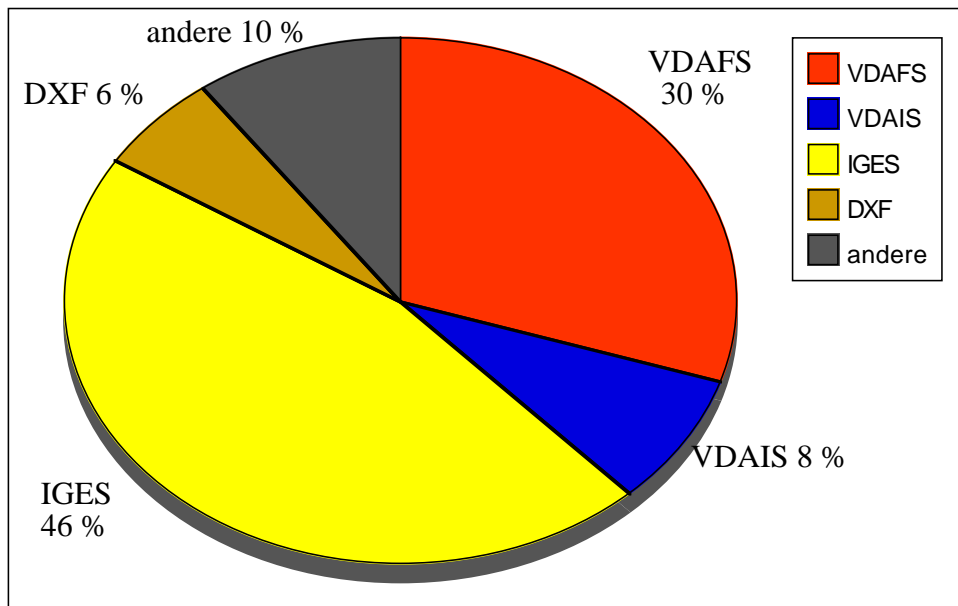


Bild 4.19 Der Anteil der einzelnen Schnittstellen am gesamten Datenaustauschvolumen /87/

Größter Schwachpunkt aktueller Schnittstellen ist die fehlende Möglichkeit der Übertragung von Daten, die nicht geometrischer Natur sind. Dieser soll mit der Definition von STEP (Standard for the Exchange of Product data) beseitigt werden. STEP /46 und 48/ soll es ermöglichen, alle für ein Produkt notwendigen Daten in einem CAD - System neutralen Format abzulegen. Das AP 214 kann beispielsweise Daten aus den Bereichen Geometrie&Topologie, Zeichnungen mit Bemaßung, Produktkonfiguration, Produkteigenschaften, Oberflächeneigenschaften, Toleranzen, Externe Referenzen, Finite-Element-Daten, Kinematik, Formelemente und Arbeitsplan übertragen.

STEP ist von der ISO in der Normenreihe ISO 10303 genormt. Bild 4.20 spiegelt den Aufbau von STEP wieder. Die Normung umfasst nicht nur die eigentlichen Produktdaten bzw. Modelle zu deren Beschreibung sondern auch Beschreibungsmethoden, Implementierungsmethoden

sowie Methoden zum Überprüfen der Konformität. Dieser Aufbau wurde gewählt, um die Mängel der im obigen Abschnitt beschriebenen Schnittstellen wie eine nicht ausreichende formale Beschreibung (IGES) sowie ungenügende bzw. nicht vorhandene Testkriterien auszuschließen. STEP ermöglicht es, anwendungsspezifische Produktmodelle durch die Verwendung von Grundbausteinen zu repräsentieren.

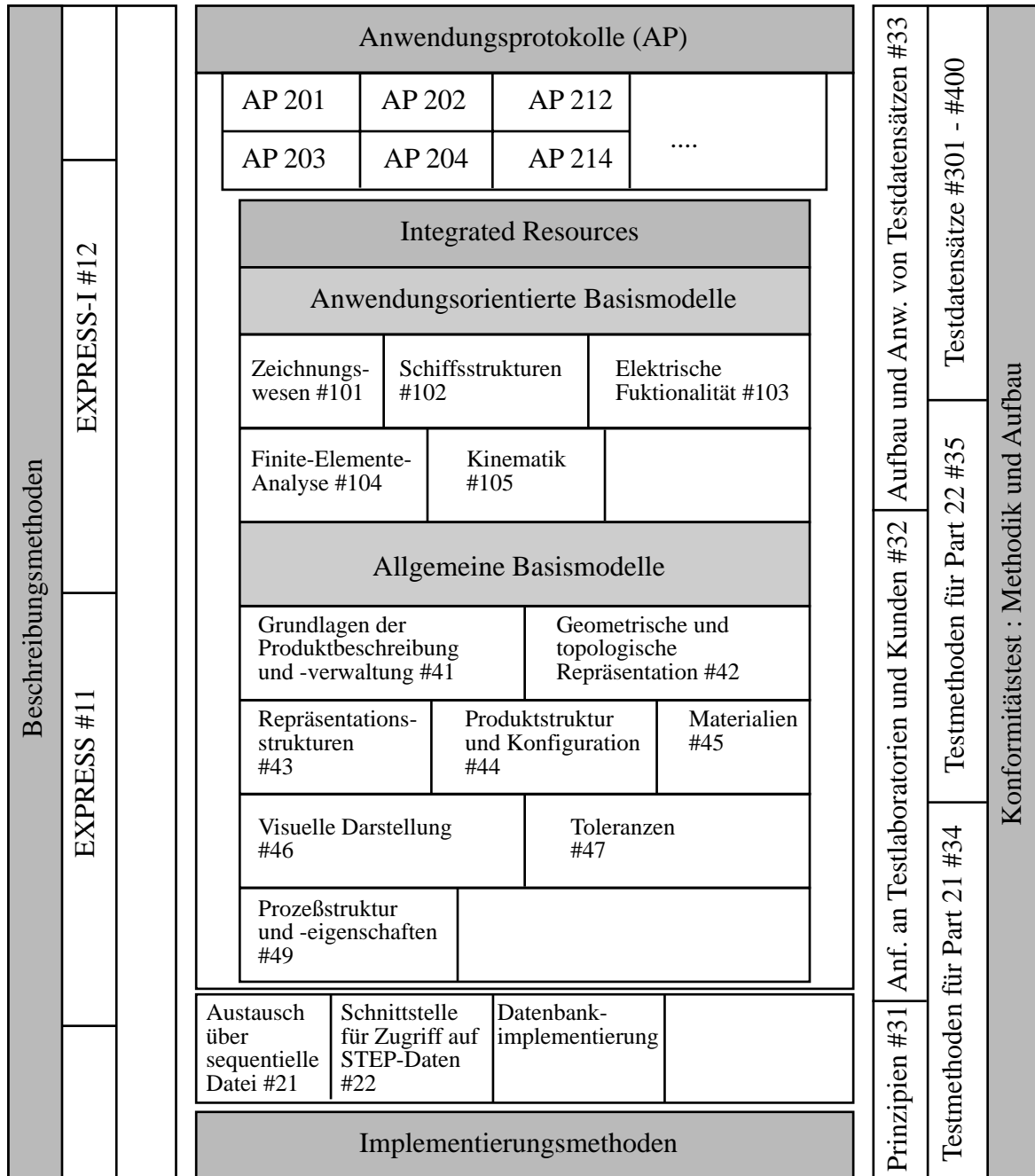


Bild 4.20 STEP-Übersicht /98/

Beschreibungsmethoden:

Die Beschreibungsmethoden werden durch EXPRESS implementiert. EXPRESS ist eine Spezifikationsprache und in der ISO 10303-11 genormt. Sie dient zur logischen Beschreibung von Informationsmodellen. Die formale, eindeutige und vollständige Beschreibung eines Produktmodells wird ermöglicht. Das nachfolgende Bild zeigt die EXPRESS-Beschreibung von zwei Kreisen.

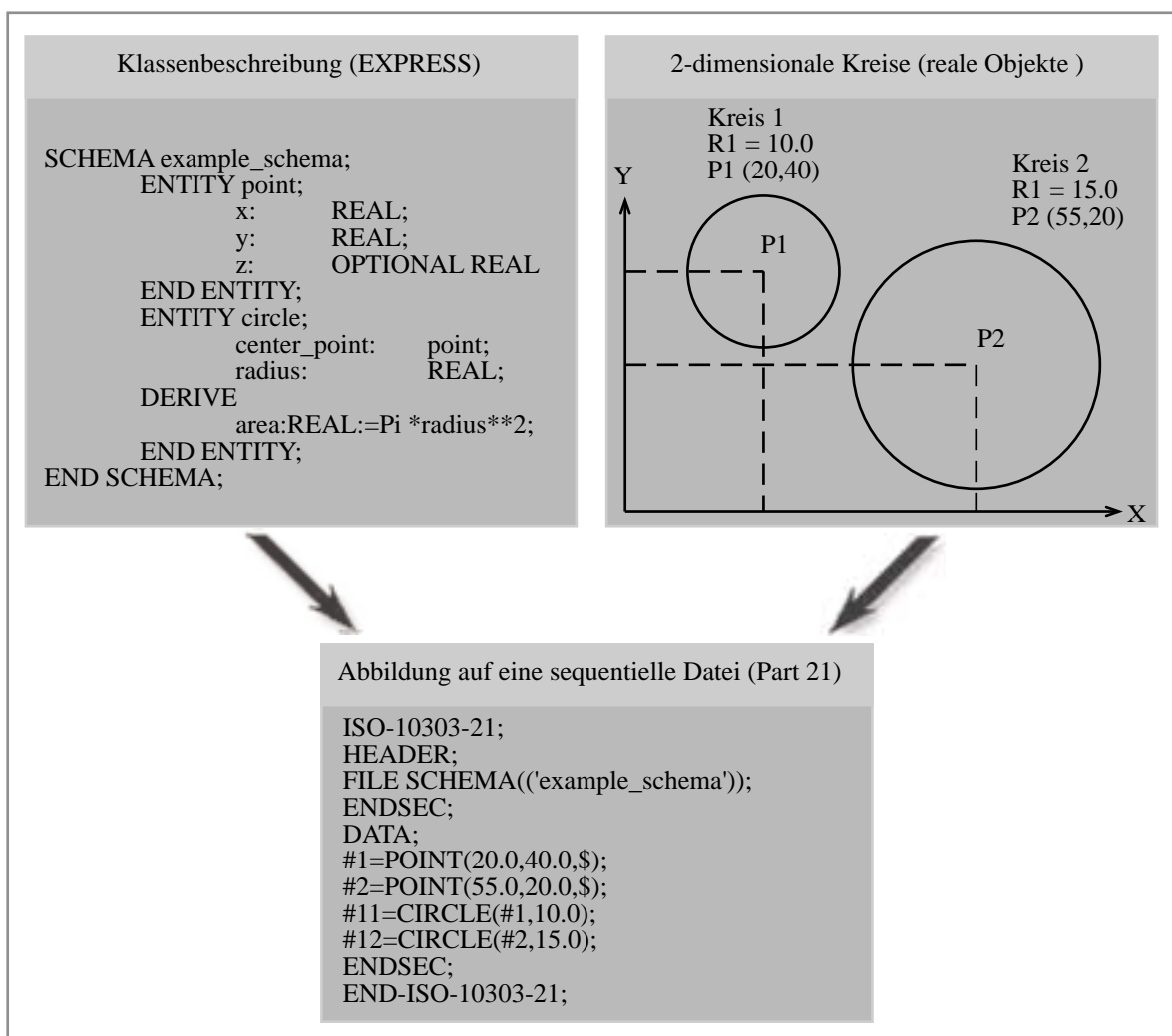


Bild 4.21 EXPRESS-Beispiel /98/

Implementierungsmethoden:

Ein in EXPRESS beschriebenes Informationsmodell ist zunächst unabhängig von einer speziellen Implementierung. Mit Softwarewerkzeugen kann ein solches Modell auf die verschiedensten Zielimplementationen abgebildet werden. Die folgenden Bereiche werden in STEP betrachtet:

- Physical File Exchange für den Austausch von Produktdaten durch eine sequentielle Datei (ISO 10303-21),
 - ein Applications Programming Interface für den Zugriff auf Produktdaten, Bezeichnung: Standard Data Access Interface (SDAI) SDAI ist in der Sprache C++ eingebunden. Neben C++ (ISO 10303-23) können die Sprachen C (ISO 10303-24), FORTRAN (ISO 10303-25) sowie IDL (ISO 10303-26),
 - Data Sharing ist der Begriff für die Möglichkeiten des Zugriffs von Anwendungsprogrammen auf Datenbasen, die im EXPRESS-Schema beschrieben wurden
- und
- die Langzeitarchivierung von Produktdaten (Produkthaftung).

Konformitätsprüfung:

Um sicherzustellen, daß auf STEP basierende Software ein hohes Maß an Kompatibilität besitzt, wurden nicht nur die Schnittstellen und Regeln definiert nach denen die Software zu erstellen ist, sondern auch entsprechende Prüfmethoden. In der ISO 10303-31 und folgenden sind Testverfahren sowie die Anwendung dieser Testverfahren beschrieben.

Anwendungsneutrale Basismodelle:

Um die Produktdaten unabhängig von einer speziellen Anwendung zu beschreiben wird die Sprache EXPRESS verwendet. STEP besteht aus einigen Basismodellen, die durch EXPRESS realisiert wurden. Basis- oder auch Informationsmodelle der Geometrie und Topologie, der Visualisierung, der Materialeigenschaften, der Toleranzen, der Formelemente und der Strukturinformationen sind bereits vorhanden. Zudem besteht noch die Möglichkeit, die logische Abfolge von Prozeßaktivitäten zu beschreiben.

Anwendungsorientierte Basismodelle:

Diese Modelle ergänzen die Basismodelle. Durch sie werden die Basismodelle in einen speziellen Kontext gesetzt. Ein geometrisches Objekt wie die Linie wird durch das Modell Draughting (ISO 10303-101) in den Bezug zu einer technischen Zeichnung gebracht und stellt eine Maßlinie dar. Ein weiteres zugehöriges Modell ist das Kinematik - Modell(ISO 103-105). In der Entwicklung befinden sich Modelle zur Definition von Schiffsstrukturen, elektronischen und elektrischen Verbindungen, der Finiten Element Analyse sowie des Bauingenieurwesens.

Anwendungsprotokolle:

Die ISO 10303-201 und folgende sind anwenderspezifische Produktmodelle in STEP. Ihre Aufgabe ist es, die für einen anwendungsspezifischen Fall benötigten Elemente in Verbindung mit den STEP Basismodellen zu setzen. Dies geschieht durch die Definition von Regeln und die Spezialisierung von Objekten wie es z.B. bei der Linie der Fall ist, die zu einer Maßlinie wird.

Im Anwendungsprotokoll ISO 10303-214 /89/ werden die Produkt- und Betriebsmitteldaten der Entwicklungsprozeßketten in der Automobilindustrie beschrieben. Unterstützt werden zur Zeit die Datenklassen der Produktstruktur (organisatorische Daten, Baugruppenstruktur, Stücklistendaten), Geometrie und Topologie (Kanten-, Flächen- und Volumenmodelle), Darstellungen der Modelle (Farbe, Linienstärke, schattierte Darstellungen), Toleranzen, Bauteil- und Baugruppenbeschreibungen in Form von Zeichnungen, die Kinematik, Materialangaben und Oberflächeneigenschaften.

4.2 Möglichkeiten der anwenderspezifischen Erweiterbarkeit

Die Erweiterungsmöglichkeiten, die CAD-Systeme für verschiedene Anforderungen bieten sind entscheidend für ihre Effizienz im industriellen Einsatz. Erst wenn das CAD-Paket von „der Stange“ durch verschiedene Anpassungsmöglichkeiten an den Bedarf der Konstrukteure angepasst worden ist, kann sein ganzes Leistungspotential ausgenutzt werden. Im allgemeinen bieten CAD-Systeme vier Möglichkeiten der Erweiterbarkeit. Diese sind die MAKRO-Programmierung, die Erstellung von Variantenprogrammen, die Anwendungsprogrammierung sowie die Möglichkeiten der Generierung von Features. Im betrieblichen Einsatz werden diese Methoden leider noch zu selten genutzt, um die Qualität sowie die Geschwindigkeit des Kon-

struktionsprozesses zu verbessern. In /7/ ist ein entsprechendes System zur Unterstützung des Entwurfsprozesses von Schiebe-Hebedächern für Personenkraftwagen beschrieben. Dieses System zeigt recht deutlich, welches Potential in der Anwendungsprogrammierung /110 und 120/ steckt. Sobald der Konstrukteur problemgerechte Werkzeuge zur Lösung seiner Aufgabe erhält kann er die Geschwindigkeit sowie die Qualität des Konstruktionsprozesses enorm steigern.

4.2.1 Die Generierung von MAKROS

Die Makroprogrammierung ist eine der einfachsten Methoden der Erweiterung von CAD-Systemen. Sie war auch eine der ersten Möglichkeiten. Durch sie werden standardisierte Geometrien oder Zeichnungselemente, die in unveränderter Form immer wieder vorkommen, fest implementiert. Bei der Erzeugung eines Makroprogramms werden alle Eingaben des Anwenders in einer Datei abgelegt. Sie kann mit einem Editor verändert werden. Auf diese Art und Weise können verschiedene Versionen eines Bauteils generiert werden. Die Arbeitsweise ist aber recht umständlich. Erzeugt man eine große Anzahl von Makros, so ist diese Methode auch nicht sehr übersichtlich. Ein weiterer Nachteil besteht darin, daß komplexe geometrische Modelle durch diese Methode nur sehr schwer zu modifizieren oder anzupassen sind. Geschwindigkeitseinbußen gegenüber Variantenprogrammen sollen nicht unerwähnt bleiben. Neben dieser Methode existieren in vielen CAD-Systemen Schnittstellen zu sogenannten Makro-Programmiersprachen. Dies sind, von der Syntax betrachtet, recht einfache Sprachen. Meist sind sie an einen BASIC- oder FORTRAN-4 Dialekt angelehnt und um Befehle zur Erzeugung von geometrischen Elementen /15, 32, 47, 55, 58, 65, 66, 69, 70 und 109/ erweitert. Oftmals fehlen komfortable Möglichkeiten der Dialogführung sowie der Speicherung von Daten. Dadurch sind sie schnell erlernbar und für kleinere Probleme einsetzbar. Haupteinsatzgebiet der Makroprogrammierung sind die Erstellung von Normteilgeometrien (Schrauben) sowie die Generierung von Zeichnungsrahmen, Toleranz- und Schriftfeldern. Im nachfolgenden Bild [4.22](#) sind einige typische Elemente einer Makroprogrammiersprache dargestellt :

Der Anfang es Programms besteht in der Deklaration von Variablen. Hier werden zuerst Variablen wie der aktuelle Layer, die aktuelle Konstruktionsebene und die aktuelle Farbe definiert und belegt. Am Programmende bzw. bei einem fehlerbedingten vorzeitigen Abbruch werden

```

DECLARE REAL fehlnum,fehlzeil <# Zur Fehlerbehandlung
DECLARE REAL AktLayer <# Aktueller Layer
...
DECLARE VECTOR NetVorn(4) <# Netzkoordinateneckpunkte vorn
DECLARE VECTOR NetUnten(4) <# Netzkoordinateneckpunkte unten
...
DECLARE ENTITY ActCpl <# CPL-Elementzeiger
DECLARE ENTITY SecCPL <# Reference-CPL

ERTRAP fehler <# Fehlerbehandlung aktivieren

GETPRM(ACS)ActCpl
GETPRM(LAYER)AktLayer
<# SELECT(ACS)DefCPL

GOSUB READPA <# Einlesen der benoetigten Parameter
GOSUB CALCNET <# Eckpunkte der Netze berechnen

WHEN MitNetz .EQ. 1
  GOSUB DRAWNET <# Netze zeichnen
ENDWHEN

GOSUB MKCPL <# Erzeugen der benoetigten CPL's
#999 CONTINUE
SELECT(ACS) ActCpl
SELECT(LAYER) AktLayer

```

Bild 4.22 Auschnitte eines CVMAC-Makro zum Erzeugen eines Gitters

diese CAD-Systemvariablen zurückgesetzt, um den Zustand des Systems bei Ausführung des Programms wieder herzustellen. Im Anschluss werden spezielle, nur für CAD-Makroprogrammiersprachen typische Variable wie Vektoren und Matrizen deklariert. Diese werden in nachfolgenden Programmierschritten mit Werten belegt. Aus diesen werden Gometrien erzeugt. Der abschließende Programmschritt besteht im Rücksetzen der CAD-Systemparameter. Der gesamte Programmaufbau gestaltet sich im allgemeinen recht einfach.

4.2.2 Die Variantenprogrammierung

Die Variantenprogrammierung erfolgt im Gegensatz zur MAKRO-Programmierung über eine High-Level-Programmiersprache. Diese greift auf ein durch das CAD-System zur Verfügung gestelltes API auf dessen Kern zu. Als Programmiersprachen werden FORTRAN 77, C, C++ sowie Lisp verwendet. Durch die Variantenprogrammierung werden alle Ausprägungen bestimmter Teile oder Baugruppen aufgrund eines Parametersatzes erzeugt. Typische Beispiele

sind Programme zur Implementierung von Baureihen. Programme zur Auslegung von Getrieben (Größen- und Gestaltvarianten) und Wellen (Größenvarianten) seien an dieser Stelle erwähnt. Neben der reinen Gestalterzeugung werden in Variantenprogrammen Berechnungen sowie Nachweise wie z.B. Festigkeitsnachweise durchgeführt. Somit ist Konstruktionswissen in einem Variantenprogramm enthalten.

Ein Nachteil der Variantenprogrammierung ist, daß ein Programmierer benötigt wird bzw. eine entsprechende Auftragsprogrammierung extern vergeben werden muß. Desweiteren erfordern die Variantenprogramme eine gewisse Systempflege (Anpassung an einen neuen CAD-System Releasesstand, neues Betriebssystem, neue Compiler), was zu einem nicht unerheblichen Wartungsaufwand führen kann. Neuere CAD-Systeme sind um Parametrikmodule erweitert worden. Diese gestatten es, Größenvarianten zu erzeugen. Dabei legt der Konstrukteur beim Aufbau seines Modells Abhängigkeiten zwischen einzelnen Geometrieelementen fest. Diese werden bei späteren Maßänderungen beibehalten. Trotz einiger dargestellter Nachteile ist das Potential zur Automatisierung wesentlich größer als bei der Makroprogrammierung.

4.2.3 Die Anwendungsprogrammierung

Es ist eine der Aufgaben der Anwendungsprogrammierung /6, 25, 65, 66, 69, 70, 123, 152 und 153/, die Prozeßschritte der Konstruktion programmtechnisch nachzubilden. Anwendungsprogramme lassen sich in zwei Gruppen einteilen. Eine Gruppe besteht aus Standardpaketen. Mit ihnen können CAD-Systeme um Kinematikmodule, Optimierungsmodule, Animationsmodule und Berechnungsmodule erweitert werden. Bild 1.5 sowie Tabelle 1 geben einen Überblick. Die zweite Gruppe betrifft den jeweiligen betriebsspezifischen Anwendungsfall. Seine Aufgabe umfasst Teilbereiche mehrerer Standardanwendungen sowie zusätzliche Restriktionen, die beispielsweise aus der betrieblichen Fertigung, geometrischen Randbedingungen oder betrieblichen Kostenvorgaben resultieren. Gerade in der Entwurfsphase könnte der Konstrukteur eine Vielzahl computergestützter Hilfsmittel nutzen, um einen schnellen Prozeßdurchlauf mit einer hohen Qualität zu erreichen. Die Nutzung mehrerer Standardprogramme birgt jedoch die Nachteile des Datenverlustes bei der Transformation in die unterschiedlichen Programme. Ein weiteres Manko besteht in der unterschiedlichen Bedienung der Standardprogramme. Ein Anwendungsprogramm besitzt den Vorteil, daß es die Benutzerschnittstelle des CAD-Systems

nutzen kann. Der Konstrukteur kann in seiner gewohnten Arbeitsumgebung weiterarbeiten. Ein Nachteil der Entwicklung von entsprechenden Softwaresystemen ist, daß zunächst ein hoher finanzieller Aufwand erforderlich ist. Dieser amortisiert sich aber schon nach relativ kurzer Zeit. Die Vorteile eines schnell durchführbaren Konstruktionsprozesses, eines reproduzierbar durchführbaren Prozesses, eines in einer kurzen Zeitspanne wiederholt durchführbaren Prozesses sowie eines dokumentiert durchführbaren Prozesses kommen hinzu. Gerade eine erhebliche Zeiteinsparung ist bei den derzeitigen wirtschaftlichen Gegebenheiten ein enormer Wettbewerbsvorteil. Für das im nachfolgendem Bild dargestellte Schiebe-/Hebedachsystem wurde in /6/ ein entsprechendes Programmpaket zur Minimierung der Bauhöhe sowie zur Verkürzung des Entwurfsprozesses entwickelt.

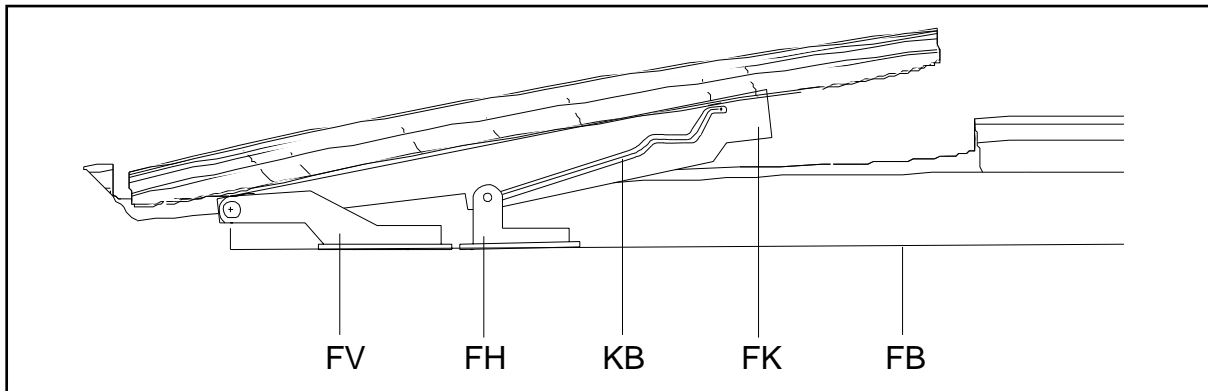


Bild 4.23 Elemente eines Schiebe-/Hebedachsystems/6/

Eingangsparameter des Optimierungsprozesses sind Funktionsparameter die den Bewegungsablauf festlegen, Konstruktionsparameter die die konstruktive Auslegung im Hinblick auf Steifigkeits- oder Fertigungsgesichtspunkte regeln sowie geometrische Randbedingungen in Form der Dachgeometrie und des Dachausschnittes. Als eine weitere Restriktion ist der betriebliche Kostenrahmen des zu entwickelnden Produktes zu betrachten. Die Zahl der Entwurfsvariablen konnte auf 38 beschränkt werden. Von diesen finden in der konventionellen Schiebe-/Hebedachauslegung nur eine Auswahl Verwendung. Das Programmsystem stellt alle Bauteile in der ersten Phase in ihrer idealisierten Form durch Parameterprogramme dar. Mit dem idealisierten System wird eine kinematische Simulation des gesamten Funktionsablaufes und eine darauf basierende iterative Ermittlung der Bewegungsbahnen durchgeführt. Im nachfolgenden Bild ist eine Lösungsfläche dargestellt.

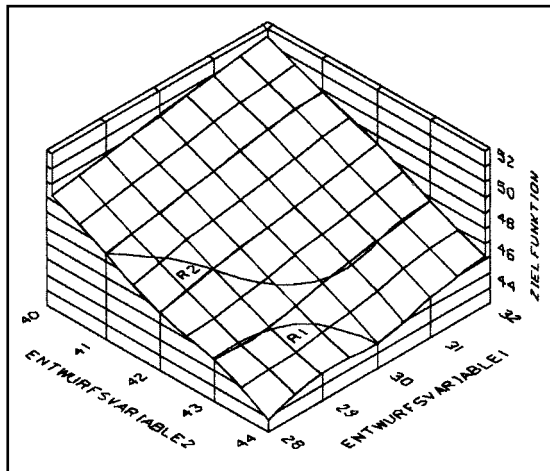


Bild 4.24 Abhängigkeiten von Entwurfsvariablen /6/

Diese ergibt sich durch die Variation zweier wesentlicher Entwurfsvariablen unter Beibehaltung der Werte der übrigen Entwurfsvariablen. Die beiden Kurven R1 und R2 stellen Restriktionen in Form von Kollisionen dar. Neben der Ermittlung der Parameter dient die idealisierte Geometrie der Bauteile als Ausgangspunkt für die Weiterverarbeitung in Form der automatisierten Zeichnungserstellung oder NC-Datengenerierung für Prototypen. Durch eine Erweiterung der Modelle um technologie- und kostenrelevante Parameter sind sie darüber hinaus als Grundlage sowohl für eine integrierte strukturmechanische Analyse als auch für eine simultane Kostenabschätzung geeignet.

4.2.4 Parametermodelle

Die Erstellung komplexer geometrischer Modelle ist keine triviale Aufgabe, auch unter Ausnutzung moderner Modellierer mit mächtigen Funktionen und ausgeklügelten Hilfesystemen. Man benötigt mehrere Tage bis Wochen, um ein kompliziertes technisches Objekt als geometrisches Modell zu erstellen. Betrachtet man hierzu die Gegebenheit, daß ein großer Teil der Konstruktion den Bereich der Anpassungskonstruktion betrifft, so erscheint es sinnvoll, die vorhandenen Modelle weiter zu nutzen. Hierzu müssen Arbeitsweisen entwickelt werden, die vorhandene Modelle den neuen Randbedingungen anpassen. Auf der Basis des „Kopierens“ und „Einfügens“ von Grundkörpern oder vorhandenen Bauteilen und Baugruppen kann der mögliche Gesamtvorteil nicht vollständig ausgeschöpft werden. Die Parametertechnologie wurde entwickelt, um dies zu erreichen. Um ein Bauteil mit einem parametrischen Modellierer

zu erstellen, wird zuerst die Topologie erstellt. Diese besteht aus primitiven grafischen Elementen. Dies können Konturen oder durch die Booleschen Operationen verknüpfte Primitivkörper sein. Diese werden meist mit einem eigenem Modul, dem sogenannten Sketcher erzeugt. Ergebnis dieses ersten Schrittes ist ein Modell, das die logischen Abhängigkeiten des Bauteils enthält. Diesem Modell können in einem zweiten Arbeitsschritt weitere Zwangsbedingungen in Form von mathematischen Gleichungen zugeführt werden. Der parametrische Modellierer versucht im dritten Arbeitsschritt ein Gleichungssystem zu erstellen, das allen bisher definierten Zwangsbedingungen genügt. Ist dieser Schritt erfolgreich abgeschlossen, kann der Konstrukteur durch Verändern der Variablen im Gleichungssystem verschiedenste Instanzen seines Modells erzeugen. Das nachfolgende Bild zeigt den beschriebenen Ablauf anhand eines Beispiels.

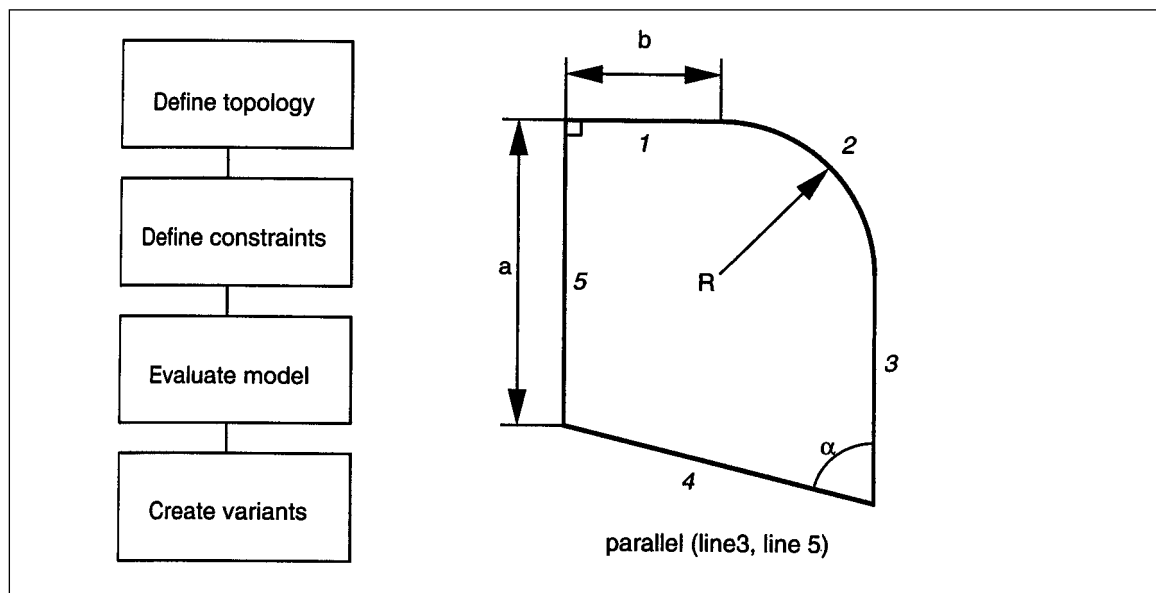


Bild 4.25 Beispiel einer parametrischen Definition /137/

Parametrische Modelle sind aber nicht einfach zu erstellen und einfach zu handhaben. Sind komplexe Einzelbauteile mit vielen Zwangsbedingungen sowie Abhängigkeiten zwischen mehreren Bauteilen vorhanden, so ist es oft sehr schwierig, den Einfluß einzelner Variabler auf das gesamte Modell zu erkennen. Hier wird eine sehr gute Dokumentation benötigt, die den Aufbau des Parametermodells sowie die logischen Abhängigkeiten sehr genau und detailliert beschreibt. Ohne eine solche Beschreibung ist es sonst einfacher und geht schneller von staten, das Parametermodell neu zu implementieren.

4.2.5 Die Erzeugung von Features

Features sind Informationsstrukturen für Gestalt- oder andere Attribute eines Produktes /4, 27, 60 und 145/. Diese Attribute enthalten neben den Gestaltinformationen weitere Informationen wie z.B. zur Herstellung /51 und 121/ und zum Zusammenbau. Bild 4.26 vermittelt einen Überblick.

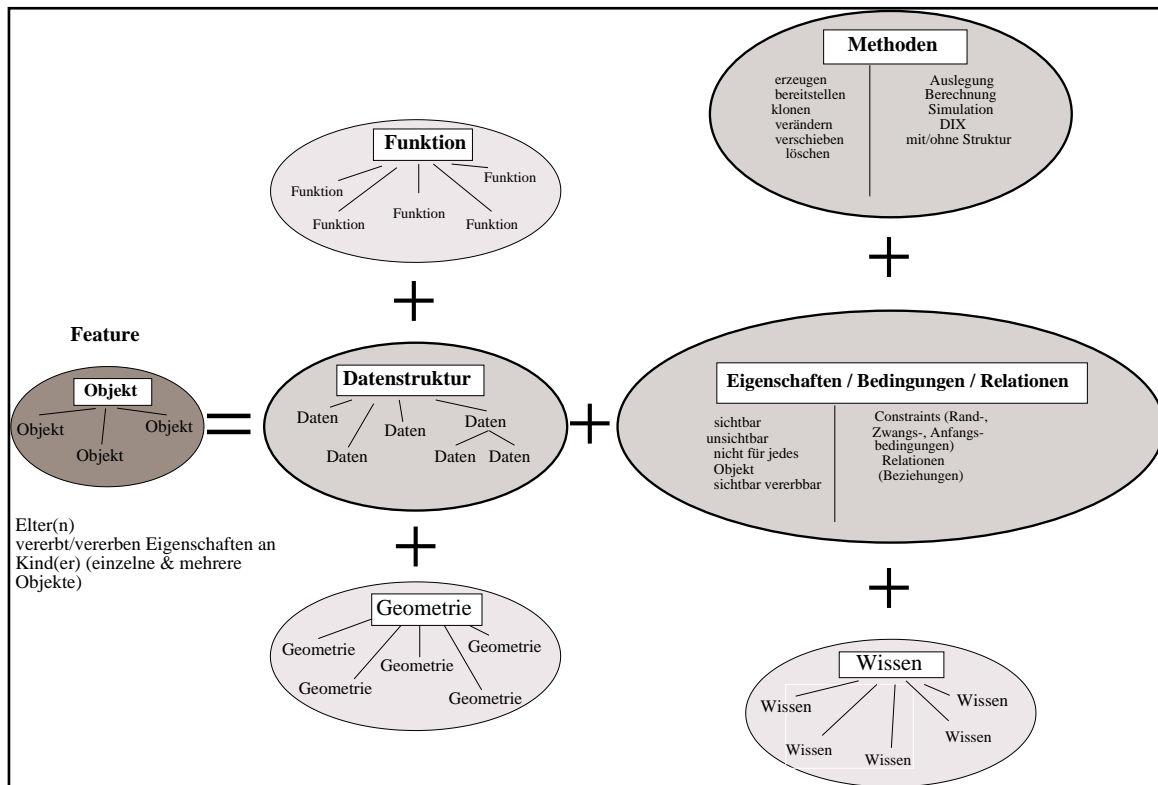


Bild 4.26 Definition von Features /142 /

Die Feature Technologie kann aber auch dazu dienen, Daten von Berechnungen, der Prozeßplanung, des Betriebszustandes sowie der Wartung für ein Produkt verfügbar zu machen. Nach Shah muß ein Feature die folgenden Eigenschaften besitzen: es ist physikalischer Bestandteil eines Teils, es hat eine generische Form, es enthält Ingenieurwissen und es besitzt berechenbare Eigenschaften. Die Feature Technologie findet auf zwei Arten Verwendung: in der Fertigungsplanung als Feature Recognition und in der Konstruktion als Design by Features. Die Methode des Feature Recognition ermittelt automatisch oder halbautomatisch, d.h. im Dialog mit einem Anwender, aus bereits vorhandenen Modellen Features. Dem zur Folge ist das Haupteinsatzgebiet der Bereich des CAM. Man findet aber auch Systeme im Bereich des CAD.

Mit der Methode des „Design by Feature“ wird ein Modell erstellt in dem aus einer Featurebibliothek entsprechende Features ausgewählt und auf eine vorhandene Geometrie angewandt werden. Features beziehen sich in diesem Zusammenhang auf einen rein geometrischen Aspekt. Vorteil dieser Methode ist die drastisch sinkende Zahl der Eingaben. Anstatt sich mit den Eingaben von geometrischen Primitiven zu beschäftigen, kann der Konstrukteur auf mächtige Elemente zugreifen.

Die Erzeugung von Features erfolgt interaktiv. Ihre rechnerinterne Repräsentation ist in allen CAD-Systemen unterschiedlich. Einheitliche Standards existieren noch nicht. Im Rahmen von PDES/STEP und dem ESPRIT (CAD*I) Projekt wurde ein produktneutrales Form-Feature Modell (FFIM) und ein neutrales Dateiformat entwickelt.

Der geometrische Teil eines Features /52 und 82/ setzt meist auf der CSG-Darstellung auf. Im Gegensatz dazu setzen CAP-Systeme meist auf einem Flächenmodell auf. Der Trend entwickelt sich aber hin zu hybriden B-rep/CSG Repräsentationen.

Zur Repräsentation des Wissens existieren keine Standards. Bisher wurde versucht, dieses Wissen durch Regeln zu beschreiben. Die Programmiersprache PROLOG ist ein extremer Vertreter dieses Denkansatzes. Aktuellere Forschungsergebnisse der Informatik weisen eine andere Richtung. Sie kombinieren Regeln und Daten, d.h. sie fassen beide als Einheit auf. Objektorientierte Programmiersprachen sind das Ergebnis. Sie stellen Wissen in Form von Klassen dar. Klassen bestehen aus Daten und den auf sie anwendbaren Methoden.

Die Erstellung von Features ist ein komplexer Vorgang. Meist steht nur die Verwendung eines Programmierinterfaces zur Verfügung. Die Generierung von Features erfordert dadurch ein hohes Maß an Programmierfähigkeiten. Desweiteren muß der Programmierer über ausgezeichnete Kenntnisse in der Konstruktion sowie der Fertigungstechnik verfügen.

5 Systemarchitekturen

Nachdem in den vorangegangenen Kapiteln die Grundlagen zur Realisierung des fertigungsorientierten Modellierens dargestellt wurden, erfolgt nun eine Beschreibung, wie diese in der gegenwärtigen EDV-Landschaft erfolgen kann. Dabei wird das System WISKON erläutert. Es ist der wohl kompromissloseste Ansatz zur Realisierung eines Modellierers. Seit seiner Implementierung haben sich einige Voraussetzungen geändert. 3D Modellierer befanden sich noch in der Entwicklungsphase. Sie waren in CAD-Systeme integriert und nicht frei verfügbar. Bei durchgeführten Tests zeigten sich Schwächen bei den verschiedensten Verknüpfungsoperationen. Derzeit existieren auf dem freien Markt mehrere Modellierkerne, die den Anforderungen genügen. Durch ihr objektorientiertes Interface gestatten sie eine schnelle und sichere Entwicklung von komplexen Anwendungssystemen.

In kaum einem Bereich hat sich der technische Fortschritt derart rasant weiterentwickelt wie in der Computergeometrie. Die grafischen Ausgabesysteme haben eine enorme Leistungssteigerung erfahren. Die ersten Systeme wie das GKS waren auf den verschiedenen Hardwareplattformen nicht zufriedenstellend implementiert. Ihre Leistungsfähigkeit war den damaligen Verhältnissen angepaßt, aber für das geplante Vorhaben, einen fertigungsorientierten Modellierer zu entwickeln, zu gering. Aktuelle am Markt erhältliche Systeme bieten die benötigte Leistung, sind weitgehend von der Hardware unabhängig und besitzen eine hohe Qualität.

Einen großen Fortschritt hat die Weiterentwicklung der Programmiersprachen gemacht. WISKON wurde in C implementiert, einer damals und auch heute noch in vielen Bereichen zeitgemäßen Sprache. Die Entwicklung bzw. kommerzielle Verbreitung objektorientierter Programmiersprachen hat die

Entwicklung von Programmen um ein vielfaches beschleunigt. Die Wiederverwendung und Nutzung von vorhandenen Ressourcen ist enorm gestiegen. Dies bedeutet aber nicht, daß in Zukunft in diesem Bereich kein weiterer Entwicklungsbedarf besteht.

Nicht zuletzt hat die rasante Entwicklung der Hardware sowohl bezüglich ihrer Leistungsfähigkeit als auch ihres rapiden Preisverfalls dazu beigetragen, immer komplexere Grafiksoftware für immer breitere Anwendungsschichten zur Verfügung zu stellen, was zu einem breiteren Markt und mehr Entwicklerressourcen führte. Diese Marktentwicklung schränkt aber aufgrund des entstandenen Datenbestandes sowie der erlernten Arbeitsweise die Weiterentwicklung bzw. Neuentwicklung von CAD-Systemen ein. So haben es CAD-Systeme, die zwar eine effiziente aber neue Arbeitsweise besitzen, relativ schwer, sich am Markt durchzusetzen. Kriterium für ein neues CAD-Konstruktions-System wird es sein, eine gewohnte Arbeitsweise beizubehalten sowie den vorhandenen Datenbestand zu übernehmen.

Zur Implementierung eines Modellierers stehen verschiedene Möglichkeiten zur Diskussion. Eine Möglichkeit besteht darin, so wie im Forschungsprojekt WISKON geschehen, ein vollständig neues System zu entwickeln, ohne Standardsoftware zu nutzen. Der Vorteil liegt darin begründet, daß alle zu implementierenden Komponenten individuell aufeinander abgestimmt werden können. Weiterhin ist man unabhängig von der Entwicklung der verwendeten Tools. Der Nachteil liegt eindeutig in dem enormen Entwicklungsaufwand begründet. Der Trend, daß selbst weit verbreitete und wirtschaftlich erfolgreiche Systeme zu weiten Teilen auf Standardsoftware beruhen, stützt diese These. Konvertierungsprogramme und Schnittstellen müßten implementiert werden, um vorhandene Datenbestände zu nutzen bzw. Daten mit am Markt vorhandenen Systemen austauschen zu können.

Die zweite Möglichkeit ist, aus Standardsoftware ein System aufzubauen. Der Entwicklungsaufwand besteht darin, die verwendeten Tools zu verknüpfen sowie das Modellierungssystem zu entwickeln. Der Nachteil dieser Methode liegt darin, daß man von den einzelnen Weiterentwicklungen der genutzten Softwarepakete abhängig ist. Vorhandene Daten könnten ohne die Entwicklung von Schnittstellen genutzt werden. Es stünden bereits einige Datenformate zum Im- oder Exportieren zur Verfügung.

Das Aufsetzen auf einem vorhandenen CAD-System ist die dritte Möglichkeit. Hierbei ist der Entwicklungsaufwand am geringsten. Desweiteren ist man auch nur von einem Softwarepaket

abhängig, über dessen Programminterface man das Modellierungssystem implementiert. Bei der Implementierung des fertigungsorientierten Modellierers wurden die beiden letztgenannten Möglichkeiten genutzt. Zunächst wurde ein System implementiert, das aus dem CV-DORS Standalone-Paket, das den Modellierkern, eine CADD5 kompatible Datenbank sowie das Grafiksystem HOOPS beinhaltet sowie der OPEN LOOK Widget Set verwendet. In einem späteren Stadium wurden die entwickelten Softwarebausteine in die CADD5 Umgebung portiert, was relativ problemlos von statten ging.

5.1 Das System WISKON

Modernen Programmierrichtlinien folgend, ist das Konstruktionssystem WISKON modular, wie in [Bild 1.9](#) dargestellt, aufgebaut. Es besteht im wesentlichen aus den Hauptmodulen „WISMOD“, „KONZEPT&TOPKON“, „FERMOD“, „GEOMOD“ und dem „Verwaltungs- und Informationssystem“. Alle Komponenten sind aufeinander abgestimmt. Das System basiert auf einem kohärenten Modell.

5.1.1 Geomod

Kern des Systems WISKON ist der geometrische Modellierer. Er besteht aus zwei Teilen, einem Modellierer für 3D-Geometrien sowie einem Sketcher für 2D-Geometrien. Beide Module sollten die gleiche Datenbasis nutzen. Der 2D-Modellierer sollte vornehmlich dazu dienen, Konturen zu erzeugen, aus denen durch die Operationen des Sweepens oder Rotierens Volumenkörper erzeugt werden. Ein weiterer Einsatzbereich sollte die gebräuchliche Zeichnungserstellung, insbesondere die Vermaßung sein. Aufgrund von Ressourcenmangel kam der 2D-Modellierer über ein Projektierungsstadium nicht hinaus. Der 3D-Modellierer basiert sowohl auf der Primitivkörperdarstellung (**C**onstructive **S**olid **G**eometry) als auch auf der Randdarstellungsform (**B**oundary **R**epresentation). Als Primitivkörper wurden der Quader, Zylinder, Kegel, Kugel und der Torus /12/ implementiert. Die Primitivkörperdarstellung wird hauptsächlich zum Modellaufbau verwendet. Im nachfolgenden Bild sind die topologischen Zusammenhänge eines Quaders dargestellt.

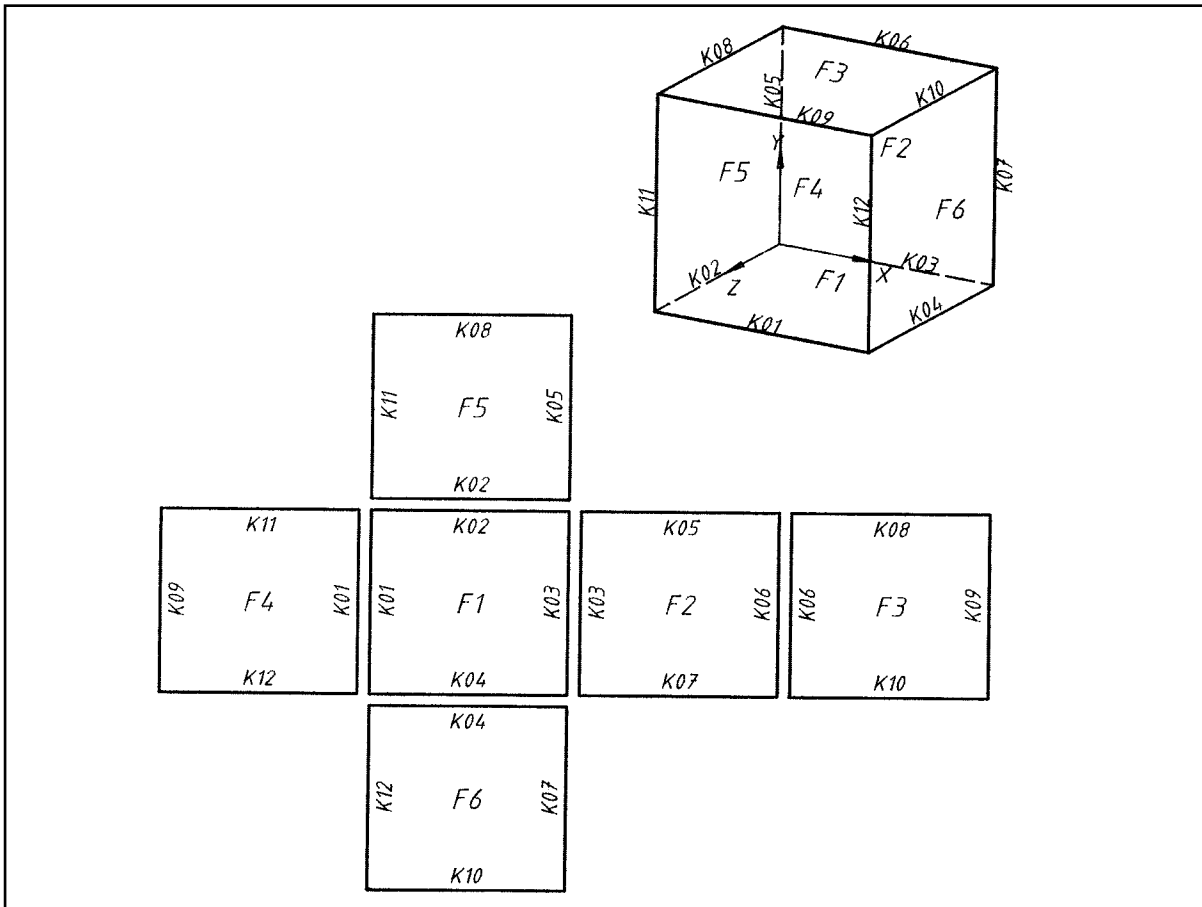


Bild 5.1 Darstellung eines Quaders /12/

Zur Erzeugung der Rotations- und Sweepkörper wird die BRep-Darstellung benötigt. Ausgangspunkt ist die Datenstruktur Körper. An sie angebunden ist die Datenstruktur Shell. Sie dient zur Verwaltung von Flächen- und Drahtgeometrie. Die Drahtgeometrien wurden nicht implementiert. Die Flächengeometrien können als geschlossenene oder geöffnete Körper verarbeitet werden. Flächen definieren durch ihren Schnitt mit einer anderen Fläche eine Kante (Edge). Bei geschlossenen Körpern wird jede Fläche durch einen geschlossenen Kantenzug begrenzt. Bei nicht geschlossenen Körpern werden Flächen durch Kanten und Halbkanten begrenzt. Um „Löcher“ in einer Fläche topologisch zu repräsentieren werden „Loops“ verwendet. Jeder geschlossene Kantenzug innerhalb einer Fläche stellt eine Loop dar. Flächen besitzen eine Referenz auf ihre mathematische Definition, die Ebene.

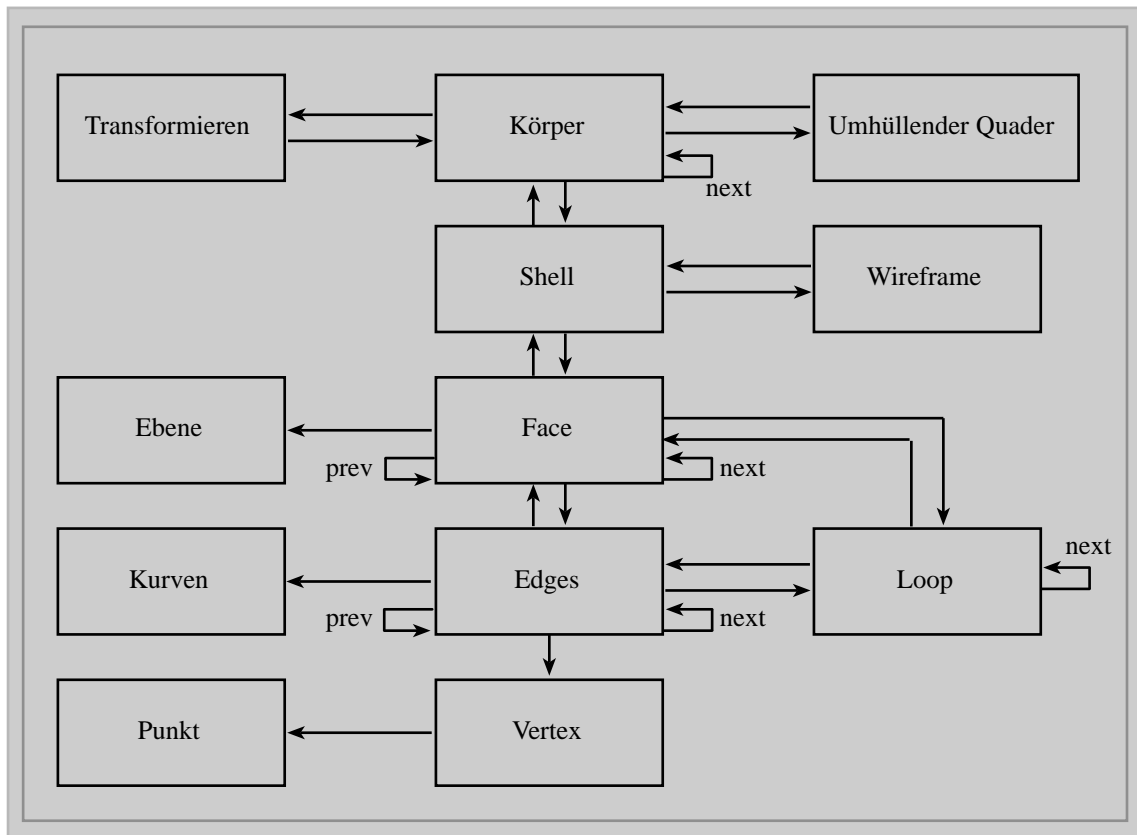


Bild 5.2 Rechnerinterne Darstellung von GEOMOD3

Kanten besitzen neben ihrer topologischen Eigenschaft eine Referenz auf ihre geometrische Darstellung als auch einen Verweis auf die sie begrenzenden Eckpunkte. Im Bild 5.2 sind diese Abhängigkeiten verdeutlicht.

Als Ausgabesystem für den geometrischen Modellierer wurde eine Submenge des GKS-Systems implementiert. GKS stellte seinerseits den Standard für grafische Bibliotheken dar. Auf der zur Verfügung stehenden Hardware (Grafikkontroller der Firma Westward, Typ 3220W) wurde das GKS leider sehr fehlerhaft implementiert.

So wurde eine eigene, GKS kompatible Library geschrieben. Diese wurde in einen gerätebezogenen und einen programmbezogenen Teil aufgespalten. Der letztgenannte diente dazu, die GKS-Befehle zu realisieren. Der erstere setzte diese Befehle auf die von der Hardware vorgegebenen Befehle um.

5.1.2 Fermod

Um den geometrischen Modellierer wurde der fertigungsorientierte Modellierer FERMOD angeordnet. Vom Blickwinkel der Konstruktion wurden für den fertigungsorientierten Modellierer die folgenden Anforderungen formuliert /28/:

- der gestaltbeschreibende Charakter der Modellierung muß weitgehend erhalten bleiben,
- der Eingabeaufwand für die Modellierung muß so gering wie möglich sein,
- die fertigungstechnischen Minimalvorgaben dürfen den Konstrukteur nicht überfordern,
- die fertigungstechnischen Angaben müssen im Sinne einer ganzheitlichen Betrachtungsweise erweiterbar sein.

Aus diesen sowie den Anforderungen der Arbeitsvorbereitung wurden die Hauptfunktionen des fertigungsorientierten Modellierers FERMOD3 entwickelt:

- Erzeuge Rohteil,
- Erzeuge Bearbeitungselemente,
- Erzeuge Technologieelemente,
- Modifizieren und Prüfen der Fertigungsbeschreibung,
- Berechnen von Teil- und Fertigungseigenschaften,
- Speichern und Laden der Modellinhalte,
- Darstellen von Rohteil, Fertigteil, Spankörper und Zwischenzuständen.

Um den Modellierkern GEOMOD3 angeordnet ergibt sich der in [Bild 5.3](#) dargestellte Modulaufbau. Die Erzeugung von Rohteilen erfolgt über die Auswahl eines Halbzeuges, Schmiedeteiles, Gußteils oder Schweißteils. Halbzeuge werden über entsprechende Tabellenwerke sowie die zugehörige Angabe der Abschnittslänge bestimmt. Schmiede- und Gußteile werden über die geometrische Modellierung dem System zugänglich gemacht.

Im Modul Modifizieren & Prüfen werden die Bearbeitungselemente, Technologievorgaben, Arbeitsfolgen und Abhängigkeiten bearbeitet. Es können einerseits einzelne Parameter oder Attribute von Elementen geändert werden andererseits kann auch die Reihenfolge von Bearbeitungselementen verändert werden. Um im letztgenannten Fall konsistente Datenmodelle zu erhalten bedarf es leistungsfähiger Algorithmen zur restrukturierung des Datenmodells.

Die Gruppe der Technologieelemente umfaßt die Speicherung der Oberflächenangaben (Rauhtiefen und Bearbeitungsmerkmale), Toleranzen (Form-, Richtungs-, Orts-, Lauf-, Maß- und Koordinatentoleranzen) und Wirkzusammenhänge.

Die Gruppe der Bearbeitungselemente umfaßt das Ausführen von Dreh-, Bohr- und Fräsverfahren. Als freie Formen sind die Verfahren des Drehens und FräSENS implementiert. Bearbeitungselemente sind hierbei Informationseinheiten, die neben den gestalterischen Merkmalen auch technologische und fertigungstechnische Merkmale besitzen.

Die Gruppe Berechnen dient zur Ermittlung von mechanischen Größen wie des Volumens, der Schwerpunkte und Trägheitsmomente als auch der Ermittlung von Relativkosten.

Das Modul Darstellen dient zum Anzeigen des rechnerinternen geometrischen Modells, der technologischen Daten sowie organisatorischer Daten wie die Liste der Fertigungsdaten und des Primitivkörperbaums.

Die Komponente Speichern & Laden speichert alle Modelldaten ab. Hierzu zählen nicht nur die geometrischen Daten sondern auch die Rohteil- und technologischen Daten. Der Baustein GEOMOD2 wird zur Erzeugung von 2D-Konturen verwendet. Diese nutzt FERMOD um sie an GEOMOD3 weiterzuleiten und Volumenkörper zu erstellen.

Ausgangspunkt der Anwendung der fertigungsorientierten Operationen ist der Eingabebaustein. Mit ihm werden die benötigten Rohteile selektiert und die anzuwendenden Fertigungsverfahren ausgewählt. Er generiert die dazu notwendigen Menüs.

Mit diesen Eingaben stößt der Modellgenerator GEOMOD2/3 zur Erzeugung der notwendigen Geometrien an. Der Modellgenerator hat die Aufgabe, verschiedene Eingabemittel wie die Tastatur, Tablets und Mäuse zu bedienen, Eingabemasken am Bildschirm aufzubauen sowie ein Eingabeprotokoll zu erzeugen bzw. dieses zu interpretieren und auszuführen.

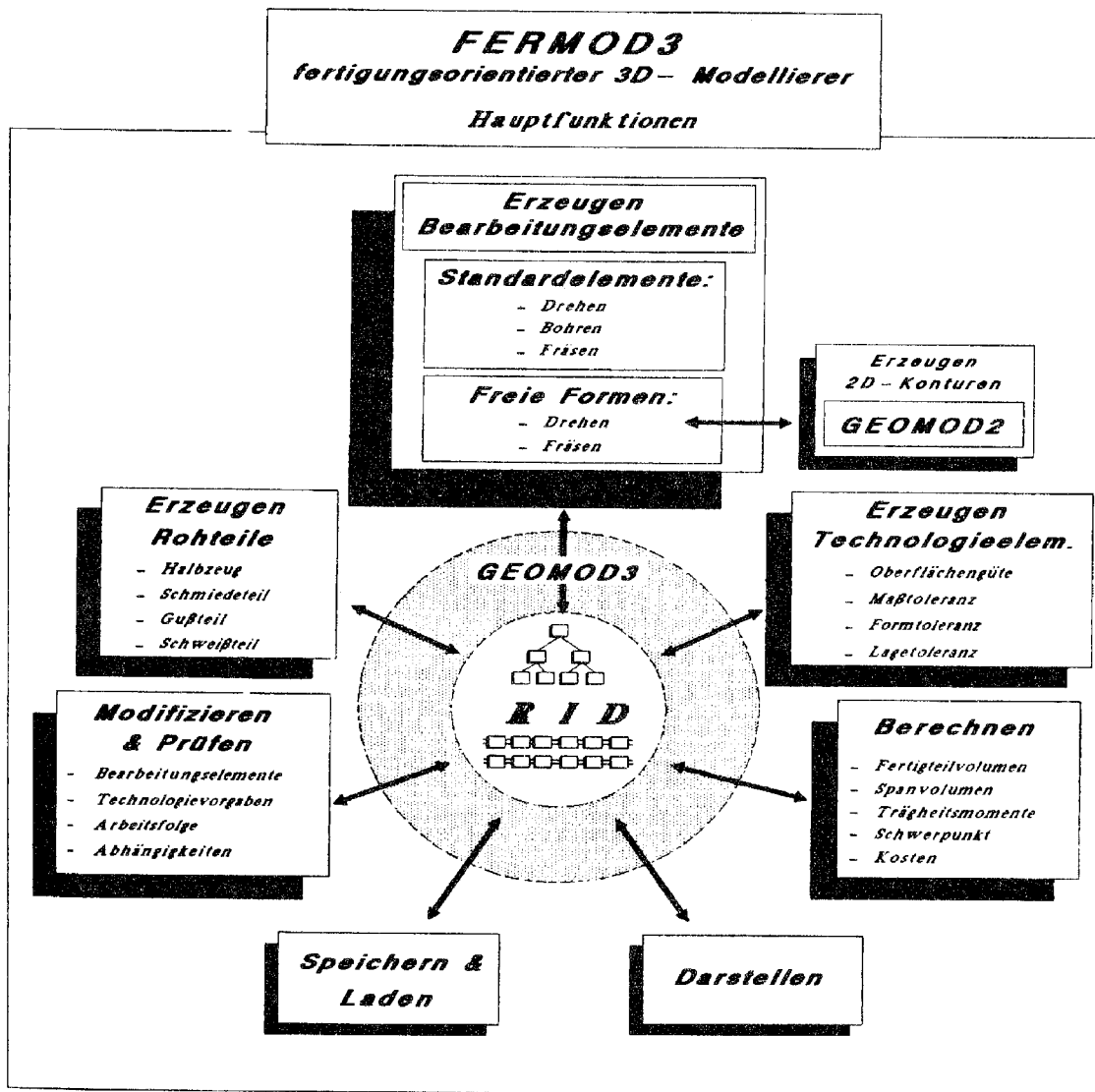


Bild 5.3 Hauptfunktionen des fertigungsorientierten Modellierers FERMOD3 /28/

Sind die Geometrien erzeugt, so generiert der Modellgenerator die notwendigen Fertigungsinformationen. FERMOD verwaltet alle erzeugten Daten. Der Darstellungsgenerator erzeugt die grafischen Ausgabeinformationen aus der rechnerinternen Darstellung und leitet sie an das Ausgabesystem weiter.

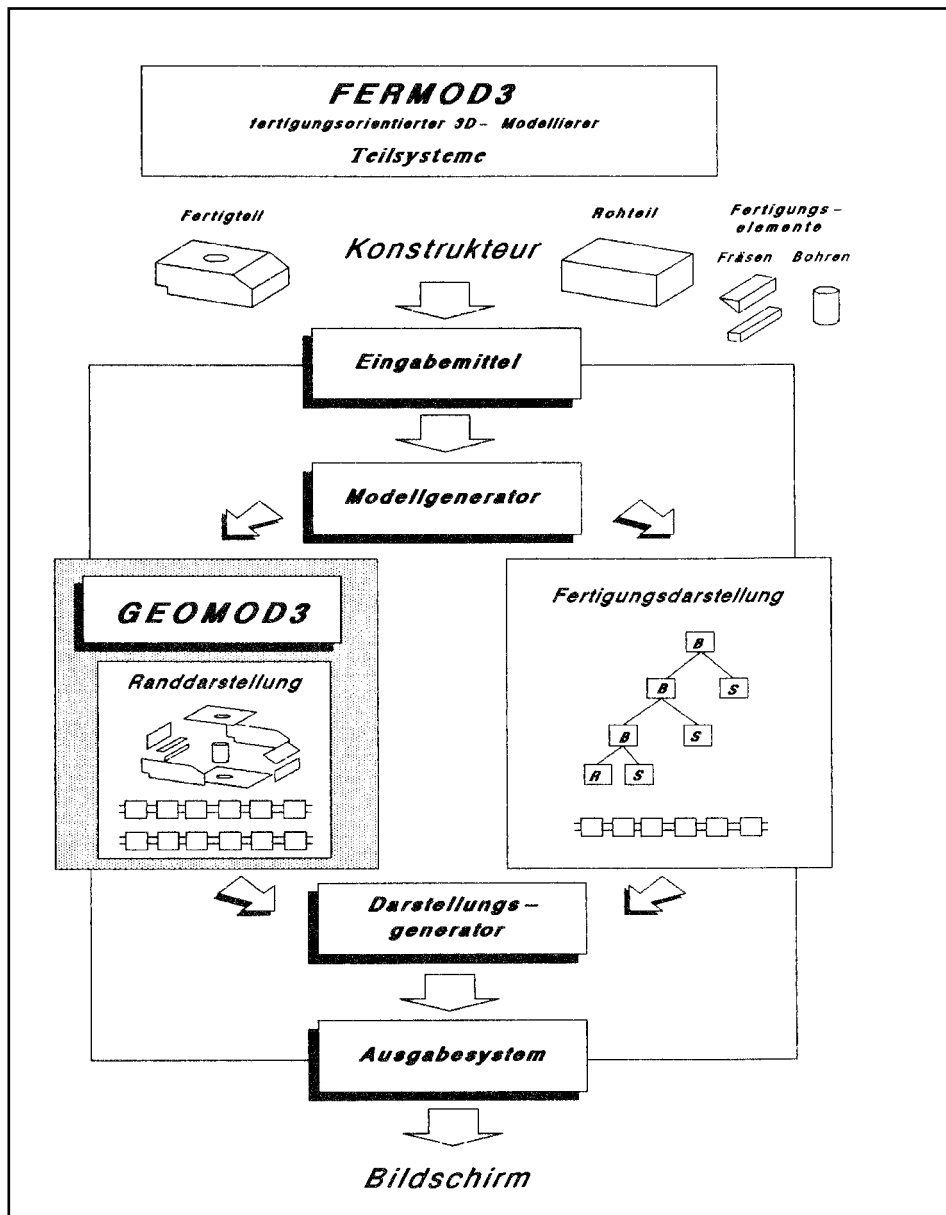


Bild 5.4 Teilsysteme des Bausteins FERMOD3 /28/

Bild 5.4 zeigt den Modellgenerator als zentralen Bestandteil von FERMOD3. Er wird von diesem angestoßen indem das FERMOD-Eingabesystem aktiviert wird. Benötigte Daten werden entweder über ein Menüsystem oder aus einer Protokolldatei eingelesen. Mit den Hauptfunktionen des in Bild 5.5 dargestellten Modellgenerators werden zuerst Roh-teile, Bearbeitungselemente oder Technologieelemente erzeugt.

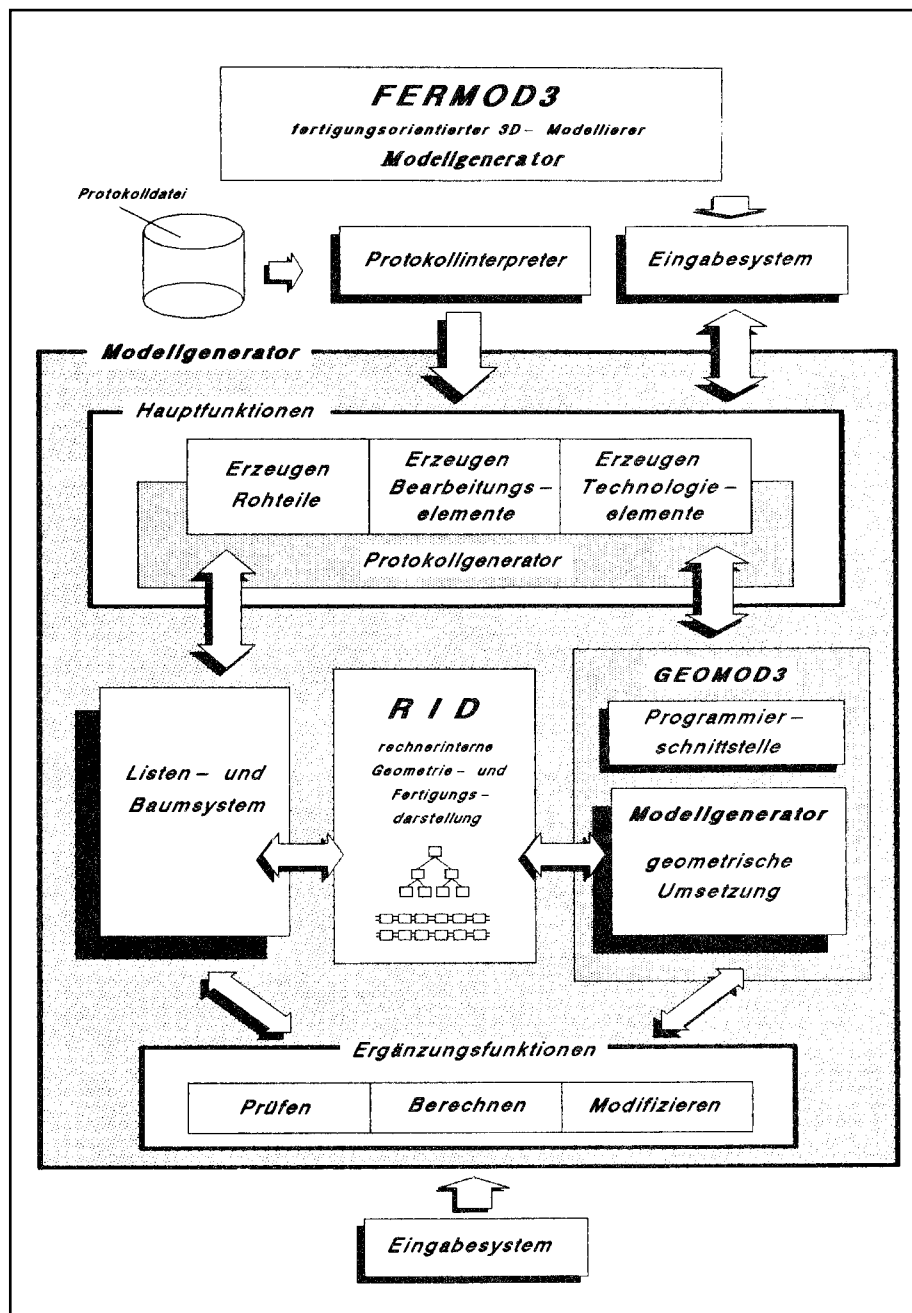


Bild 5.5 FERMOD3-Modellgenerator /28/

Deren geometrischen Darstellungen werden durch die von GEOMOD3 zur Verfügung gestellte Programmierschnittstelle (API) erzeugt. Der Modellgenerator fügt diese Daten in sein Listen- und Baumsystem ein. Wurde ein Rohteil erzeugt, werden durch den Baustein „Erzeugen Bearbeitungselemente“ Fertigungsverfahren ausgeführt. Durch Zugriff auf GEOMOD3 werden

Spankörper erzeugt. Diese werden zusammen mit einer Referenz auf das Halbzeug und einer mengentheoretischen Operation an GEOMOD3 weitergegeben. GEOMOD3 erzeugt einen neuen Körper und liefert ihn an den Modellgenerator zurück. Das nachfolgende Bild zeigt die internen Verknüpfungen der geometrischen und fertigungstechnischen Datenelemente.

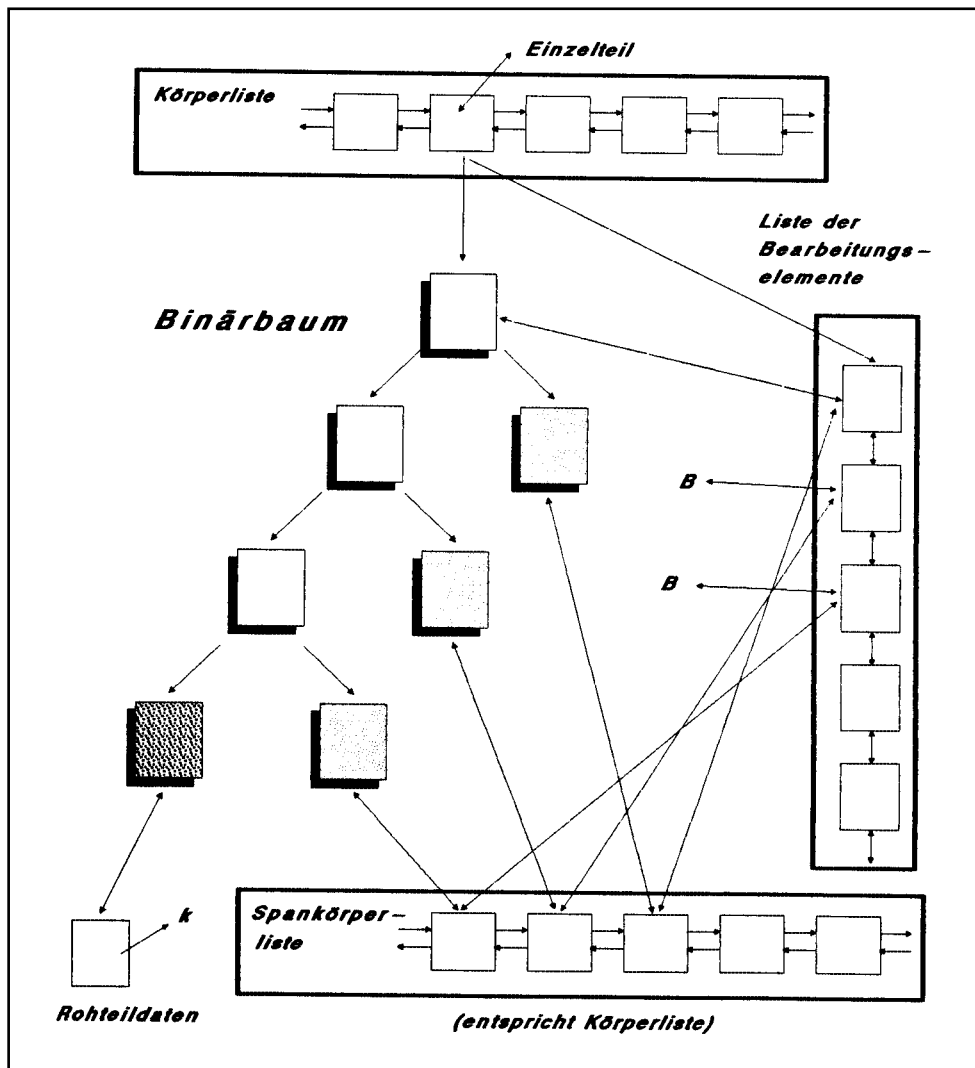


Bild 5.6 Integration von Fertigungsdaten in das GEOMOD3 Datenmodell /28/

Der Modellgenerator kann jetzt weitere Berechnungen wie z.B. die Berechnung des Spanvolumen durchführen und Technologieelemente an die neu entstandene Struktur anbinden. Anschließend kann ein neuer Bearbeitungsschritt durchgeführt werden. Die Gruppe der Ergänzungsfunktionen wird aus dem Prüfen, Berechnen und Modifizieren gebildet.

5.1.3 Konzept & Topkon

Dieser Modul dient zur Unterstützung der frühen Konstruktionsphase. In ihm wurden Hilfsmittel zum Aufbau einer Produktstruktur sowie dessen Nutzung implementiert. Zur Produktstruktur gehört hier u.a. die Wirk- oder Verbindungsstruktur sowie die Funktionsstruktur. Letztgenannte ist der wesentliche Bestandteil des Moduls KONZEPT. Der Modul TOPKON erfüllt neben seinen technologischen Aufgaben verwaltungstechnische. Um ein Bauteil zu erzeugen oder zu modifizieren, muß der Konstrukteur zuerst den Modul TOPKON aufrufen. Mit diesem wählt er das zu verändernde oder zu erzeugende Bauteil aus.

5.1.4 WISMOD

Dieser Modul steuert den gesamten Programmablauf des wissensbasierten Konstruierens. Alle notwendigen Voreinstellungen und Initialisierungen werden in diesem Modul durchgeführt. Eine Datenbankschnittstelle zur Verwaltung von Projekt-, Produkt- und Zeichnungsdaten ist hier integriert. Die Hauptaufgabe des Moduls WISMOD besteht jedoch darin, dem Konstrukteur Werkzeuge zur Bearbeitung von Konstruktionswissen zur Verfügung zu stellen. Hierbei sollten Objekt- und Regeleditoren sowie Inferenzmechanismen zur Herleitung von neuem Konstruktionswissen aus bereits vorhandenem implementiert werden.

5.2 Die Implementierung durch Standardsoftware

Alle Module des Konstruktionssystems WISKON wurden in „C“ bzw. „PROLOG“ (WISMOD) implementiert. Sie waren Einzelanfertigungen. In den nachfolgenden Kapiteln wird die Implementierung eines fertigungsorientierten Modellierers mit Standardsoftware untersucht. Dabei wird ein Konzept basierend auf den Komponenten CV-DORS, HOOPS und Motif sowie ein Konzept das auf der Einbettung in CADD5 basiert vorgestellt.

Der Kern der Implementierung, das CV-DORS-Interface, blieb gleich so daß nur geringfügige Änderungen vorgenommen werden mußten. Vorteil des ersten Konzeptes ist die ungebundene Erstellung der Benutzeroberfläche. Bei der Einbettung in CADD5 sind durch dessen Benutzerinterface bereits starke Einschränkungen vorgegeben.

5.2.1 Der Systemaufbau mit Einzelkomponenten

Der Modellierer CV-DORS (Computervisions Developers Open Resource Software) besteht aus zwei Teilen. Ein Teil ist der geometrische Modellierkern. Er ermöglicht die Erstellung und Manipulation rechnerinterner geometrischer Modelle. Der andere Teil ist der sogenannte „interactive Viewer“. Er besteht aus einem eingeschränkten Interface zu der Grafikbibliothek HOOPS und dient zur Visualisierung der rechnerinternen Modelle. Zur Erstellung von Userinterfaces bietet er nur ein sehr eingeschränktes Interface, das sich lediglich zu einfachsten Zwecken eignet, an. [Bild 5.7](#) veranschaulicht den implementierten Systemaufbau. Dargestellt sind die Hauptkomponenten des Gesamtsystems sowie deren Kommunikation.

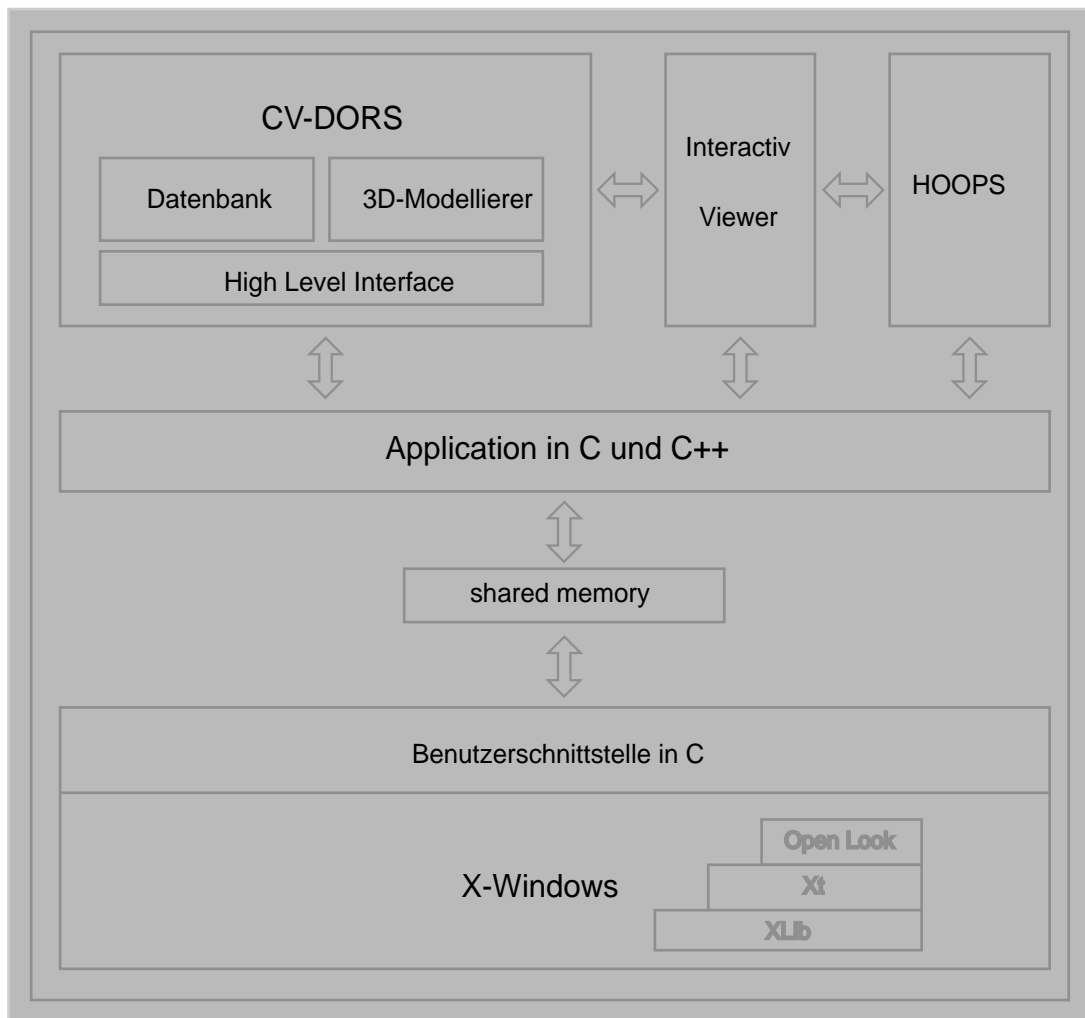


Bild 5.7 Systemaufbau

Als Programmierschnittstelle stellt CV-DORS eine „C“- , „FORTRAN“- und eine objektorientierte „C++“-Schnittstelle zur Verfügung. Über die letztgenannte wurden die eigenen Pro-

gramme angebunden bzw. eigene Datentypen in das System eingebunden. Ein Problem von CV-DORS ist, daß es keine Möglichkeiten zur Programmierung von grafischen Benutzerschnittstellen bietet. Um zu einer ansprechenden Benutzeroberfläche zu gelangen wurde das X-Windows System mit dem Widget Set OPEN LOOK genutzt. Da weder CV-DORS noch der „interactive Viewer“ Möglichkeiten zur Behandlung der Eingabebeques beschreiben, kam es bei der Implementierung eines entsprechenden Programmes zu Problemen bei der Identifizierung von Elementen im „interactive Viewer“. Aus diesem Grund wurde der Programmteil zur Steuerung der Benutzeroberfläche in einem eigenem Programm realisiert. Die Kommunikation beider bestehender Programme erfolgt über „shared memory“.

5.2.1.1 Die Schnittstelle zwischen CV-DORS und dem Benutzerinterface

Die Schnittstelle wurde durch die Methode des sogenannten „shared memory“ realisiert. Beim Starten der beiden Programme werden vom Betriebssystem zwei Speicherblöcke reserviert. Deren Adresse wird von den Programmen über eine zuvor zu vergebende Codenummer ermittelt. Ein Speicherblock dient zur Übertragung der Daten, der andere dient zur Kommunikation der beteiligten Programme. Die Daten zur Erzeugung von Geometrien und zur Steuerung der Darstellung des rechnerinternen Modells werden nach dem in [Bild 5.8](#) dargestellten Protokoll ausgetauscht:

```
typedef struct Com_Punkt { double p[6]; } Com_Punkt;
typedef struct Com_Line { double coords[6]; } Com_Line;
typedef struct Com_Kreis { double Radius ,MittelPunkt[3]; } Com_Kreis;
typedef struct Com_Koordinate { double coords[3]; } Com_Koordinate;
typedef struct Com_CharPuffer { char Puffer[45]; } Com_CharPuffer;
typedef struct Com_Plandrehen {
    double Durchmesser,Tiefe,OberflGueteKreis,OberflGueteMantel;
} Com_Plandrehen;
typedef struct Com_RohteilRund{
    double XPos,YPos,ZPos,Laenge,Durchmesser;
    char Ausrichtung,UseXYZPos;
} Com_RohteilRund;
typedef struct Com_SetView{
    char ViewName,Projektion,Ansicht;
    double XMin,XMax,YMin,YMax;
} Com_SetView;
struct Com_Buffer{
    short user,type;
    union {
        Com_CharPuffer com_charpuffer;Com_Koordinate com_koordinate;
        Com_Kreiscom_kreis;Com_Linecom_line;Com_Punktcompunkt;
    } Com_GrElems;
};
```

Bild 5.8 Programm-
kommunikation

In dem zweiten Puffer legen die Programme Informationen darüber ab, ob sie einen Befehl ausgeführt haben bzw. ob sie einen neuen Befehl im Datenpuffer abgelegt haben. Dieser Puffer kann die Zustände „Com_Sperren_Puffer“, „Com_Freigeben_Puffer“, „Com_Geaenderter_Puffer“, „Com_Geaenderter_Puffer2“ und „Com_Programm_Ende“ annehmen. Sie sind ausreichend, um eine Kommunikation der beiden Programme zu gewährleisten. Beide Programme werden getrennt über eigene Shells gestartet. Bei der entsprechenden Integration des Systems in CADDS5 war dieser Schritt überflüssig.

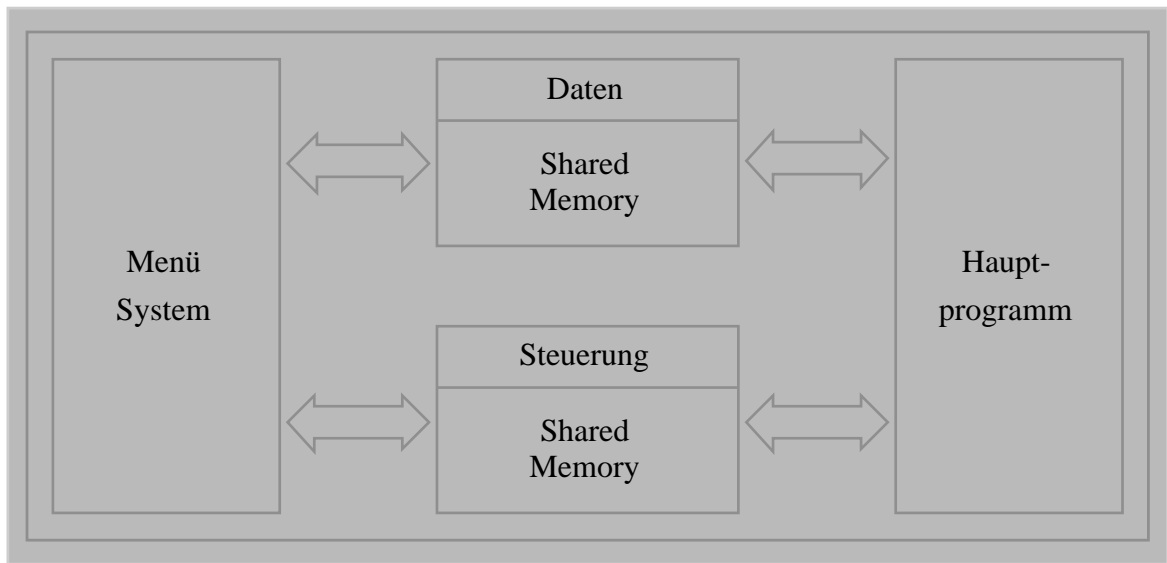


Bild 5.9 Die Kommunikation zwischen CV-DORS und HOOPS

Das obige Bild veranschaulicht den Sachverhalt. Es zeigt die an der Kommunikation beteiligten Programme sowie deren Kommunikationswege, den Datenbereich und den Steuerbereich.

5.2.1.2 Die Schnittstelle zwischen CV-DORS und HOOPS

Die zwischen dem Modellierer CV-DORS und dem Visualisierungswerkzeug HOOPS bestehende Schnittstelle wurde von der Firma Computervision erstellt und als „interactive Viewer“ bezeichnet. Sie bietet eine Schnittstelle zur Visualisierung der rechnerinternen Daten. Die Möglichkeiten zur Behandlung der Windows sowie der Event-Queues wurden nicht offengelegt. Daher wurde eine eigene Schnittstelle definiert und implementiert.

Sie besteht aus drei Teilen. Der erste Teil nutzt den „interactive Viewer“ zur Visualisierung der rechnerinternen Daten. Dies erfolgt durch Befehle wie:

```
t_geom::layer()      // Elemente in einen Layer legen
t_entity::transform() // Elemente verschieben/rotieren
t_entity::display()  // Elemente anzeigen
```

Die vom „interactive Viewer“ in „C“ implementierte Display - Listensteuerung wurde durch eine eigene, objektorientierte Steuerung in „C++“ ersetzt. Grund hierfür war die mangelnde Flexibilität der vorhandenen Steuerung. Eine wichtige Funktionen wie das Entfernen von Listenelementen aus der Displayliste existierte nicht und hätte mit den vorhandenen Datenstrukturen nur unter großem Aufwand implementiert werden können.

Der zweite Teil dieser Schnittstelle dient dazu, das grafische Fenster zu steuern. Er greift wiederum auf die Schnittstelle des „interactive Viewer“ zu. Die Projektionsart, die Anzahl, die Größe und die Darstellungsart (Hidden-Line, schattiert ...) eines grafischen Fensters werden mit diesem Teil der Schnittstelle bestimmt. Attribute wie die Belegung der Layer mit Farben können ebenfalls vorgenommen werden. Im nachfolgenden ist eine kleine Auswahl von implementierten Funktionen aufgelistet:

```
t_gr_draw_create()    // Eine Zeichnung wird erzeugt
t_gr_view_create()    // Ein „Fenster“ auf das Modell wird erzeugt
t_gr_set_color()      // Farbe setzen
t_gr_set_display_mode() // Setzen des Anzeigemodus
```

Der dritte Teil der Schnittstelle nutzt Befehle der Grafikbibliothek HOOPS. Als Subset wurden die Befehle zur Gestaltung der Ansicht implementiert. Zu diesen gehörten u.a.:

```
QSet_Camera_Position()
Orbit_Camera()
Set_Color()
Zoom_Camera()
Update_Display()
```

Die Befehlsbeschreibungen sind der einschlägigen Literatur /2/ zu entnehmen. Von den in /2/ definierten Abschnitten Attribute, Berechnungen, Eingabe, Geometrie, Segmente und System werden von dem implementierten Programm Funktionen der Abschnitte Attribute und System verwendet.

5.2.1.3 Das Menüsystem

Basierend auf der X-Windows Architektur /30 und 84/ und dem „OPEN LOOK“ Widget-Set wurde das Hauptmenü mit den im nachfolgenden Bild 5.10 dargestellten Auswahlpunkten erstellt :

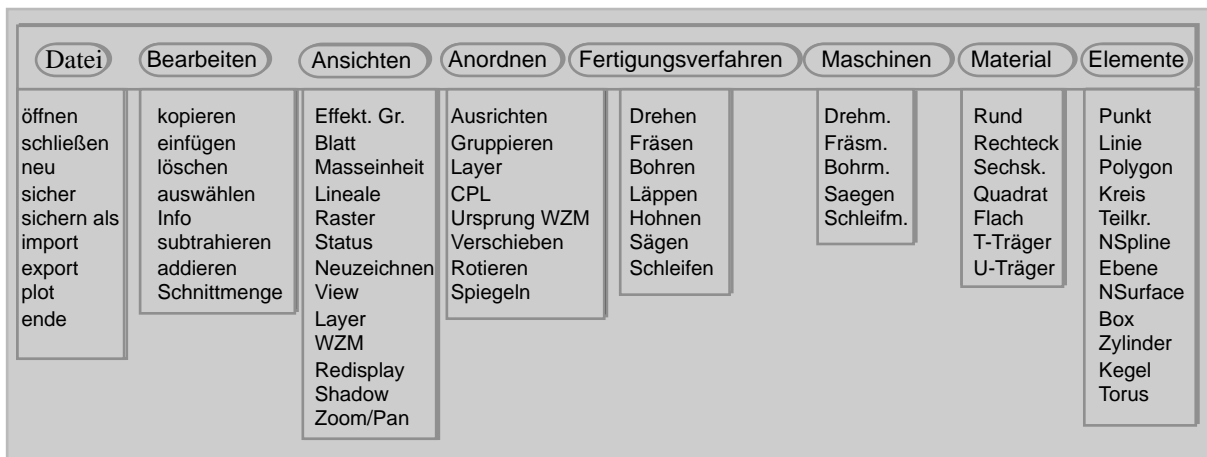


Bild 5.10 Hauptmenü

Das Menü lehnt sich an die allgemeinen Definitionen und Richtlinien zur Gestaltung eines Benutzerinterfaces an. Die Gestaltung findet sich in ähnlicher Form in vielen Programmen, angefangen bei Textverarbeitungen, Tabellenkalkulationen über Datenbanken bis hin zu technisch wissenschaftlichen Programmen. Bei all diesen Programmen lassen sich gleiche Funktionen an gleichen Menüpositionen ausführen. Im vorliegenden Menüsystem finden die Elemente des Menübalkens, der Pull-Down- sowie der Pop-Up-Menüs Verwendung. Innerhalb dieser werden Editierfelder, Check-Boxes, Check-Buttons und Buttons sowie Tear-Drop Elemente verwendet.

Der Auswahlpunkt „Datei“ der Menüleiste öffnet ein Pulldown-Menü mit Auswahlmöglichkeiten, die notwendig sind, um Modelle in Dateien zu sichern bzw. um Modelle aus Dateien einzulesen. Die Option „Plot“ gestattet die vielfältige Ausgabe des Modells auf Plottern und

Druckern. Die Menüpunkte „Export“ und „Import“ gestatten es, Modelle in Fremdformaten abzulegen bzw. sie aus diesen einzulesen. Mit dem Unterpunkt „Ende“ wird das Programm beendet.

Mit dem Auswahlpunkt „Bearbeiten“ lassen sich ausgewählte Elemente manipulieren. Zu den aus Standard - Anwendungen bekannten Optionen „kopieren“, „einfügen“, „löschen“ und „auswählen“ erscheinen die Optionen „addition“, „subtraktion“ und „Schnittmenge“. Mit ihnen werden die mengentheoretischen Operationen auf zwei oder mehr Elemente angewandt. Der Auswahlpunkt „Ansichten“ enthält Unterpunkte, die das Ausgeben des rechnerinternen Modells auf Ausgabegeräte wie Bildschirm, Plotter und Drucker definieren. Das Einstellen von Rasterpunkten, die Einblendung von Linealen und des Zeichnungsstaus sind nur einige Beispiele.

Mit dem Punkt „Anordnen“ werden Menüpunkte eingeblendet, die das Ausrichten von Elementen gestattet. Das Gruppieren von Elementen sowie das Ablegen von Elementen auf bestimmten Layern kann durch einen weiteren Menüpunkt bestimmt werden. Desweiteren können Elemente verschoben, rotiert und gespiegelt werden.

Unter dem Auswahlpunkt „Fertigungsverfahren“ können die Verfahren „Drehen“, „Fräsen“, „Bohren“, „Läppen“, „Hohnen“, „Sägen“ und „Schleifen“ ausgewählt werden. Die ersten drei Fertigungsverfahren wurden implementiert und mit Untermenüs versehen. „Maschinen“ bietet ein Pulldown-Menü aus dem die Werkzeugmaschinen ausgewählt werden können. Als Auswahlpunkte stehen Dreh-, Fräs-, Bohr- und Schleifmaschinen sowie Sägen zur Verfügung.

Die Materialauswahl steht durch den Auswahlpunkt „Material“ zur Verfügung. Hierunter sind die Halbzeuge eingefügt. Als Formen können Rund-, Rechteck-, Sechskantmaterial sowie „T“, „L“, „Z“ und „I“-Profile selektiert werden.

Mit dem letzten Auswahlpunkt „Elemente“ werden grafische Primitive ausgewählt. Hierzu zählen sowohl die Linienelemente als auch die Volumenelemente. Zu jedem Auswahlpunkt erscheint ein Popup-Menü in dem die einzelnen Auswahlmöglichkeiten weiter spezifiziert werden. Bei der Auswahl von Linien kann beispielsweise weiter in horizontale oder vertikale Linie spezifiziert werden. Stellvertretend sei hier die im nachfolgendem [Bild 5.11](#) dargestellte Struktur des Menüs „Drehen“ mit seinen Untermenüs gezeigt:

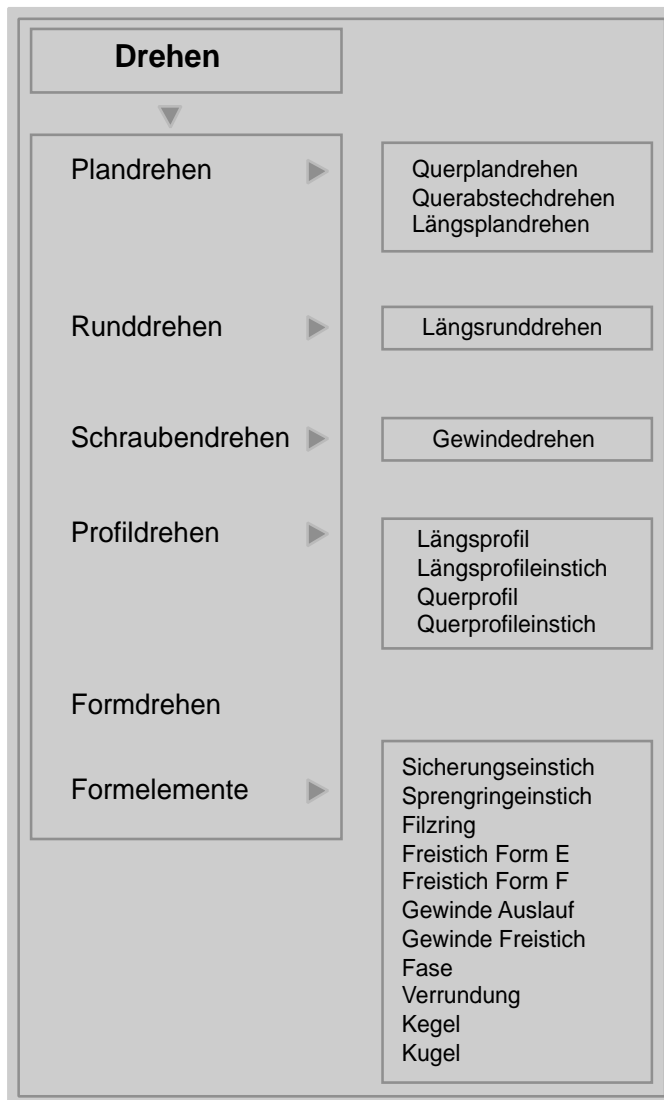


Bild 5.11 Das Menü „Drehen“

Neben dem Fertigungsverfahren „Drehen“ sind analog zum dargestellten Menübaum Menüs für die Fertigungsverfahren „Fräsen“ und „Bohren“ realisiert worden. Nach der Auswahl eines Menüpunktes der Untermenüs erscheint ein wie in [Bild 5.14](#) dargestelltes Popup-Menü. In dieses „Datenblatt“ sind die Werte für ein Fertigungsverfahren einzutragen. Im Editierfeld „Durchmesser“ ist einzutragen, auf welchen Durchmesser ein Werkstück abgedreht werden soll. Der Abstand „a“ bezeichnet die Breite, die abgenommen werden soll. Zusätzlich zu diesen zwingend notwendigen Informationen können Zusatzinformationen über die Oberflächen-güte der Stirn- und Mantelfläche eingegeben werden.

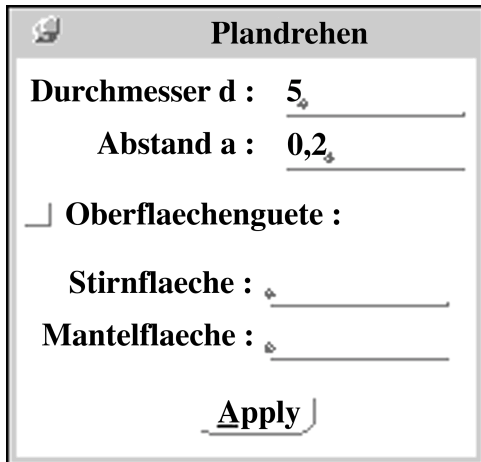


Bild 5.12 Menü Plandrehen

Die Menüs zur Materialauswahl wurden ebenfalls eingearbeitet. [Bild 5.13](#) zeigt ein Pop-up-Menü zur Auswahl eines Rundmaterials. Die Angaben „Durchmesser“ sowie „Laenge“ müssen eingegeben werden. Durch die Check-Box „Position“ und den Check-Button „X-Achse“ können zusätzliche Informationen über die abgelegt Position und Lage abgelegt werden.

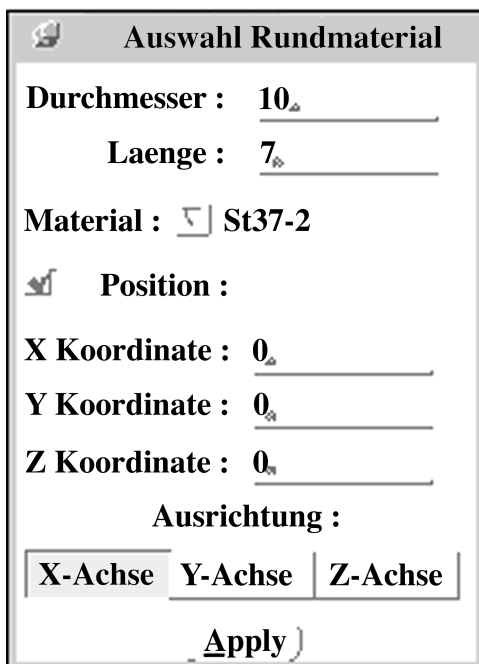


Bild 5.13 Menü Rundmaterial

Um das grafische Fenster zu gestalten und zu manipulieren wurden ebenfalls Pop-up-Menüs generiert. Diese werden aus dem Hauptmenü „Ansichten“ durch Auswahl eines Untermenüs aufgerufen. [Bild 5.14](#) zeigt das Menü zum Einstellen der Projektion, der Projektionsebene sowie der Ansicht eines grafischen Fensters. In diesem kann über ein Teardrop-Menü ein View

selektiert werden. Zu diesem können die Bildraumkoordinaten sowie eine Projektionsart und die Ansichtsart ausgewählt werden. Als Projektionsarten stehen die orthografische sowie die parallele Projektion zur Auswahl.

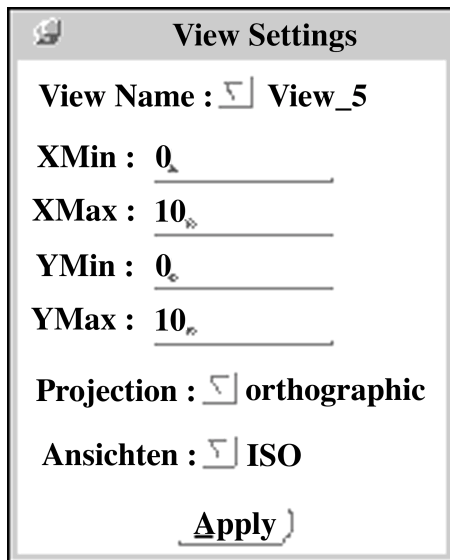


Bild 5.14 Fenstereinstellungen

Folgende Ansichten können durch ein Menü gewählt werden: die Vorderansicht, die Rückansicht, die Ansicht von links und von rechts, die Draufsicht und die Ansicht von unten. Die gewählte Ansicht kann in der isometrischen oder dimetrischen Projektion dargestellt werden.

5.2.1.4 Das grafische Fenster

Das grafische Programmsystem HOOPS wurde zur Darstellung der rechnerinternen Modelle genutzt. Die Aufteilung des Bildschirms erfolgt standardmäßig wie in [Bild 5.15](#) dargestellt. Das UNIX-Window ist in vier Views unterteilt. Diese sind nach der 3-Tafel Ansicht angeordnet. Die einzelnen Views können aber durch das in [Bild 5.14](#) gezeigte Popup-Menü beliebig verändert werden. Eine weitere Möglichkeit die Abbildung des Modells in die Views zu bestimmen liegt darin, die Kamerapositionen beliebig zu definieren. Kameras bilden die in einem dreidimensionalen kartesischen Koordinatensystem definierten Objekte in das zweidimensionale des Bildschirms ab.

Durch diese Möglichkeiten wurden dem Programm die Funktionen des Zoomens, Verschiebens sowie Rotierens zur Verfügung gestellt. Die entsprechenden Funktionen sind über ein Popup-Menü zugänglich.

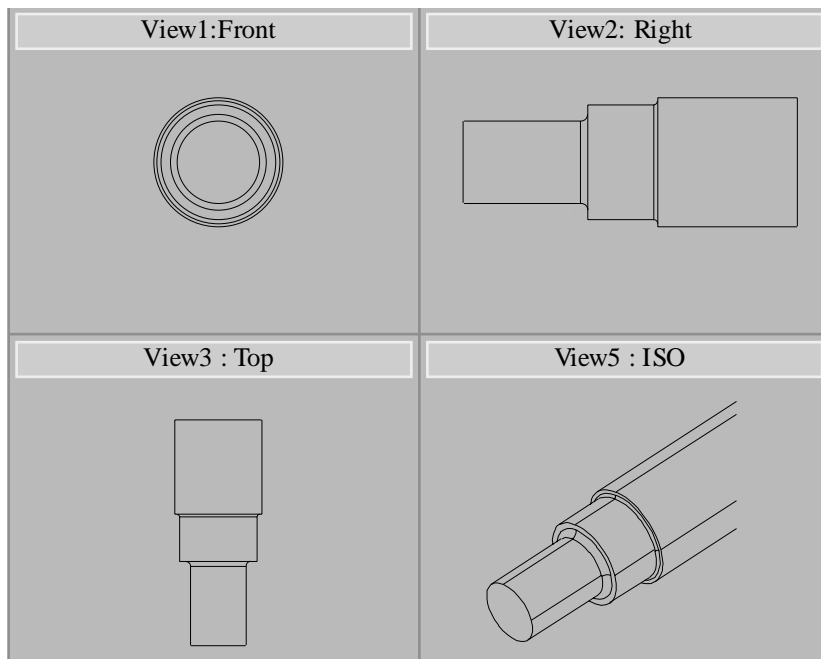


Bild 5.15 Das Grafikfenster

Neben der Ausgabe auf einen Bildschirm besteht die Möglichkeit der Ausgabe über einen Drucker oder Plotter. Die Ausgabe auf einen Drucker wird in zwei Schritten durchgeführt. Im ersten Schritt wird eine Datei erzeugt:

```
HC_Open_Segment („?driver/postscript/doily.ps“);  
HC_Include_Segment(„picture“);  
HC_Update_Display();  
HC_Close_Segment()  
HC_Delete_Segment („?driver/postscript/doily.ps“);
```

Die erzeugte Datei wird in einem zweiten Schritt an einen Drucker gesendet und entsprechend umgesetzt. Neben dem im Beispiel gezeigtem Postscript steht eine große Anzahl weiterer Formate zur Verfügung. Die Formate von Hewlett Packard und das CGM sind die am weitesten verbreiteten unter ihnen.

5.2.2 Die Einbindung in ein CAD-System

Diese Methode besitzt drei wesentliche Vorteile gegenüber den bisher dargestellten. Der erste besteht darin, daß die Anwender in ihrer gewohnten Umgebung weiter arbeiten können. Erlernte Befehle und Arbeitsweisen können unverändert übernommen werden. Ein anderer Grund besteht darin, daß bereits vorhandene Daten ohne Verluste genutzt werden können. Weiterhin ist von Vorteil, daß der CAD-Anbieter sein System auf neue Hardwareplattformen und neue Betriebssystemversionen portiert. Ein erheblicher Aufwand an selbst zu leistender Arbeit wird dadurch eingespart.

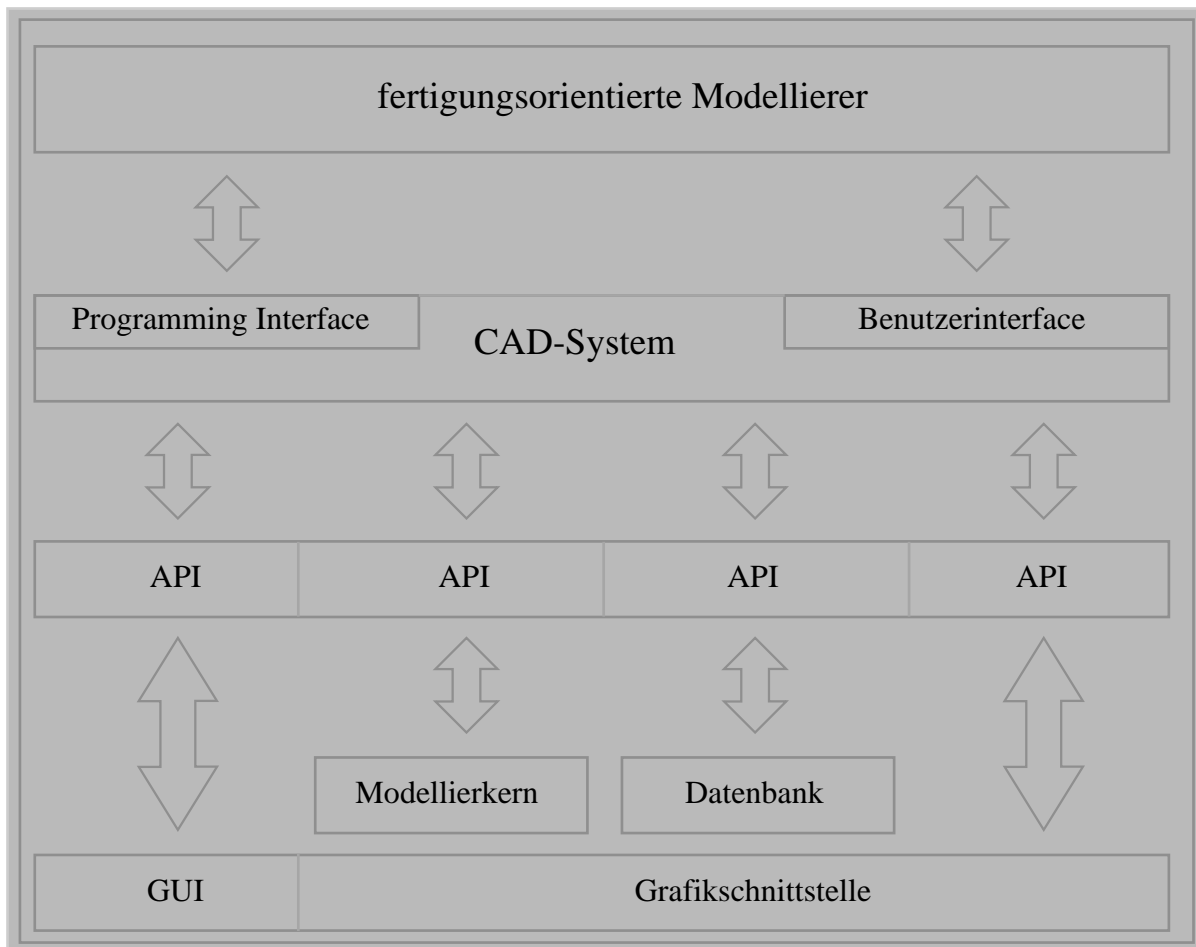


Bild 5.16 Systemaufbau

Andererseits kann dieser Vorteil auch in einen Nachteil umschlagen. Implementiert der CAD-Anbieter sein System nicht schnell auf neue Hardwareplattformen bzw. paßt er es nicht schnell an sich rasch ändernde Hardware- und Softwaregegebenheiten an, so verpasst man eventuel

sich abzeichnende neue Trends bzw. Märkte. Ein weiterer Nachteil besteht darin, daß man sich der Systemphilosophie des CAD-Systems anpassen muß. Andererseits kennen sich die Anwender bereits in dieser Umgebung aus. Ein weiterer Nachteil ist, daß die Datenstrukturen des CAD-Systems nicht ausreichen, um die benötigten Informationen eines Konstruktionssystems aufzunehmen. Neueste Entwicklungen in der Softwareindustrie könnten dieses Argument allerdings bald entkräften. Das vorangegangene Bild zeigt die prinzipielle Integration des fertigungsorientierten Modellierers in ein CAD-System. Die einzubindende Software kommuniziert mit zwei Systemkomponenten des CAD-Systems. Eine ist das Programmierinterface des CAD-Systems, das andere die Benutzerschnittstelle.

5.2.2.1 Das CAD-Programmierinterface

Das API eines CAD-Systems gliedert sich in mehrere Untersysteme. Der wichtigste Bereich betrifft die Möglichkeit der Nutzung von geometrischen Funktionen. Dazu gehören die Erzeugung der geometrischen Elemente sowie Funktionen zur Bearbeitung der selben. Trimmfunktionen und Schnittpunktberechnungen sind diesem Bereich zuzuordnen.

Ein zweiter großer Bereich betrifft die Nutzung der CAD-Datenbank. Suchfunktionen und Filterfunktionen sind nutzbar, um eigene Datensätze zu generieren. Zur Weiterverarbeitung der CAD-Daten ist in verschiedenen Bereichen oft nur ein kleiner Teil dieser Daten notwendig. Mit Hilfe von Filterfunktionen können diese generiert werden.

Die Veränderung der Systemumgebung kann mit einer Ansammlung von Unterprogrammen vorgenommen werden. Die Funktionen dienen dazu, die Längeneinheit festzulegen, den Layern Farben zuzuordnen, Ansichtsfenster zu definieren und Koordinatensysteme ein oder auszublenden.

5.2.2.2 Die Verwendung des Benutzerinterfaces

Die einzubindenden Programme werden üblicherweise über das Benutzerinterface des CAD-Systems aufgerufen. In den vorhandenen Systemen sind bestimmte Icons (Schaltelemente) oder Menüs nicht belegt. Sie können mit einem Ressourcen-Editor oder Customizer bearbeitet werden. Diese Werkzeuge gestatten es, ein Bild (Bitmap) auf ein Icon bzw. einen Text in ein Menü zu legen. Beim betätigen des Elementes wird eine durch den Customizer definierte Aktion (Unterprogrammaufruf) ausgeführt.

Einige CAD-Systeme besitzen zudem noch eine Kommando- oder Befehlszeile. In dieser werden, einer Syntax folgend, Befehle eingegeben. Ein Kommandoprozessor interpretiert die Eingaben und generiert aus diesen Befehlen die Programmaufrufe. Die Befehlsstruktur ist in sogenannten verb-noun-tables festgelegt. Diese können mit einem Editor erweitert oder angepaßt werden, um eigene Befehle zu generieren.



```
$:GENerate POInt TAG ON STGT DIST 20 : TAG POI1
```

Bild 5.17 CADD5-Kommandozeile

Der Aufbau der Befehle ist in [Bild 5.17](#) anhand eines Beispiels dargestellt. Erster Teil eines Befehls ist das sogenannte Befehlswort, im Beispiel GENerate. Der Befehlsinterpreter erzeugt große Buchstaben solange ein Befehlswort noch nicht eindeutig ist. Sobald es eindeutig wird, generiert er kleine Buchstaben. Der Anwender braucht in diesem Fall den Befehl auch nicht vollständig auszuschreiben. Im Anschluß an das Befehlswort wird das Subjekt angegeben. Ist seine Eingabe abgeschlossen, können verschiedene Modifikatoren angegeben werden. Im Beispiel soll der zu erzeugende Punkt auf einem Element mit der Kennzeichnung „STGT“ liegen. Der Modifikator DIST zeigt an, daß der neue Punkt im Abstand von 20 mm zu dem Punkt mit der Bezeichnung POI1 erzeugt werden soll. Diese Art der Befehlseingabe erscheint zunächst sehr komplex und unübersichtlich. Mit zunehmender Einarbeitung und Übung erweist sie sich allerdings als überaus mächtig und sehr schnell. Nachdem man einen gewissen Befehlsumfang sowie Modifikatorenumfang erlernt hat, lassen sich diese sehr leicht auf neue Elemente anwenden. Neue Befehle werden intuitiv erzeugt.

Das nachfolgende Bild zeigt den systematischen Aufbau der CADD55 Benutzeroberfläche. Dieser ist ein typischer Vertreter modern gestalteter Benutzerinterfaces. Hauptbestandteil ist der Bereich zur grafischen Ein- und Ausgabe. Um ihn herum gruppieren sich Menüleisten und Spalten. Diese sind in verschiedene Bereiche unterteilt. In der oberen Menüleiste befinden sich vornehmlich Befehle zum Steuern der CAD-Systemumgebung. Die Zuweisung von Farben zu Layern, die Definition von Ansichten, das Abspeichern von Dateien sowie die Positionierung dieser Ansichten im Zeichenbereich seien hier genannt. Die Icons im oberen linken Eckbereich enthalten Betriebssystem bedingt Befehle zur Steuerung des gesamten Fensters. Dazu gehören

Befehle wie das verkleinern, das Darstellen als Icon und das verschieben des Fensters. Mit dem obersten Menüpunkt des sich auf der linken Seite befindlichen Menüs kann ausgewählt werden, ob man in den Zeichenmodus, den Modellmodus oder den Parametermodus wechseln möchte. Der mit „Environment globals II“ bezeichnete Bereich enthält oft benötigte geometrische Funktionen. Unter diesem Bereich befindet sich das Icon „Task set status/access“. In diesem befinden sich themenorientierte Auswahlmöglichkeiten. Ist das geometrische Modellieren im Menü „Environment status/access“ ausgewählt, so kann in diesem Menü ausgewählt werden, ob Körper oder Flächen erzeugt werden sollen oder ob die mengentheoretischen Operationen ausgeführt werden sollen.

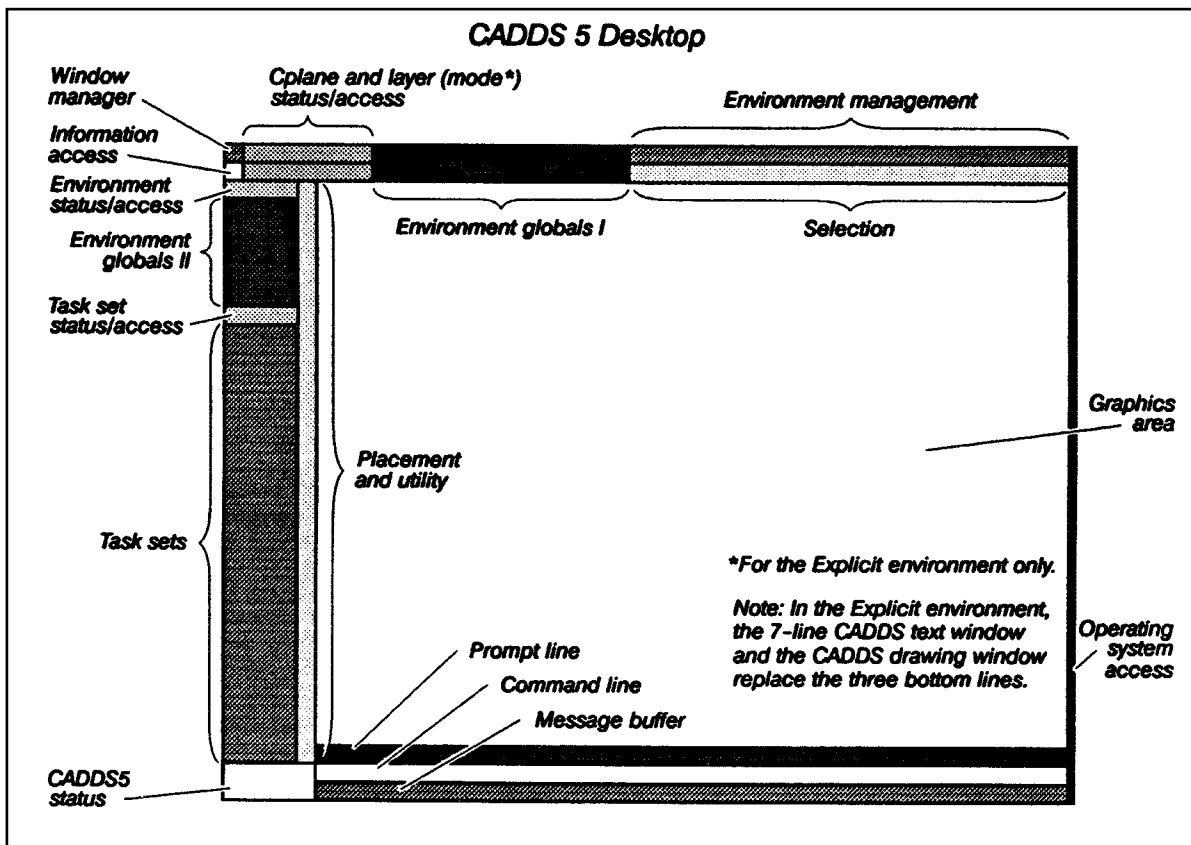


Bild 5.18 CADD5-Desktop

Der mit „Task Sets“ bezeichnete Rechteckbereich beinhaltet Funktionen aus dem in „Task set status/access“ spezifiziertem Themengebiet. Bei der Erzeugung von Grundkörpern werden die Icons zur Generierung von Quader, Zylinder, Kegel, Torus und Ellipsoid angezeigt. Auswahlmöglichkeiten zur Erzeugung von Körpern durch Rotations- oder Translationsbewegungen

können ebenfalls aufgerufen werden. Zwischen „Task sets“ und grafischem Bereich befindet sich ein Menü mit der Auswahlmöglichkeit von Positionierungsmöglichkeiten. Im unteren Bereich des Bildschirms befinden sich drei Menübereiche, die in Art von drei Zeilen angeordnet sind. Die obere Zeile ist die Datenzeile und dient zur Eingabe von Daten. Die Kommandozeile zeigt die Befehle an, die ausgeführt werden wenn Menüs bedient werden bzw. über die Tastatur Befehle eingegeben werden. In der untersten Zeile werden Systemmeldungen angezeigt.

Die dargelegte Struktur findet sich in ähnlicher Form in vielen CAD-Systemen wieder. Um das zentrale Fenster des grafischen Bereichs gruppieren sich Menüzeilen und -spalten. Dabei gibt es unveränderliche Menüs, die oft benötigte Funktionen enthalten und veränderliche Menüs. Diese strukturieren die einzugebenden Daten und Entscheidungen in einer hierarchischen Form und bilden sie in Menüs und Untermenüs ab.

Neben den Menüs dient zunehmend der grafische Cursor als intelligente Eingabehilfe. Das nachfolgende Bild zeigt die Möglichkeiten des IDEAS-Modelliersystems.

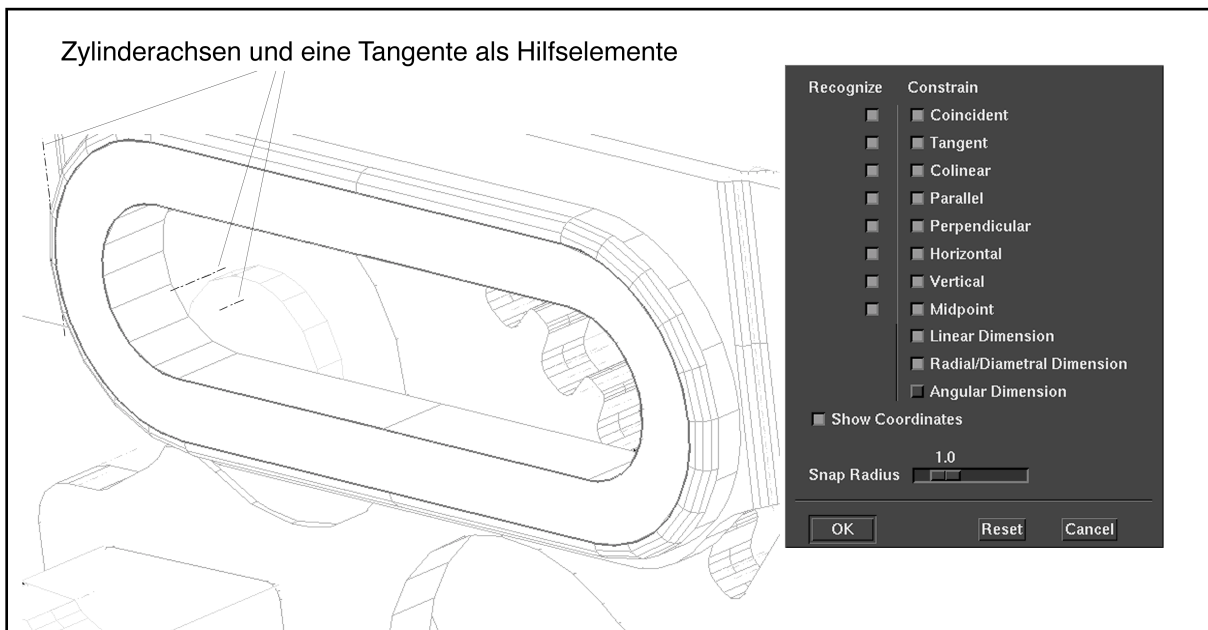


Bild 5.19 SDRC Menü zum Anzeigen von Hilfselementen

Im Dialogfeld kann durch Check-Boxen eine Auswahl von anzuzeigenden Hilfselementen selektiert werden. Diese können u.a. zur Erzeugung von Schnittpunkten, Mittelpunkten und Tangenten verwendet werden. Flächen die in der Konstruktionsebene liegen werden farblich gekennzeichnet.

Diese Möglichkeiten werden von CADD5 nicht unterstützt. Die Implementierung der Befehle zur fertigungsorientierten Modellierung erfolgte über die Verb-Noun-Tables und die damit verbundene Generierung von Befehlen in der Kommandozeile. Die Einarbeitung der Befehlsstruktur in Menüs wurde nicht durchgeführt.

6 Die Fermodklassen

Im vorangegangenen Kapitel wurden prinzipielle Wege zur Implementierung verschiedener Module und Modelle zum Aufbau und zur Erweiterung von CAD-Systemen dargelegt. In diesem Kapitel erfolgt die detaillierte Beschreibung zur Implementierung eines fertigungsorientierten Modellierers. Die dazu notwendigen Klassen sowie deren Implementation werden vorgestellt. Grundlage der Implementation ist die CV-DORS Klassenhierarchie. In diese wurden zur Erstellung des fertigungsorientierten Modellierers drei weitere Klassen eingefügt. Der Modellierungshistorie folgend wurde die Klasse „Halbzeug“ eingefügt. Sie dient zur Auswahl von Roh- und Halbzeugen. Zur Bearbeitung der ausgewählten Halbzeuge wurde die Klasse „Fertigung“ erstellt. Sie stellt eine Anzahl von Verfahren, die zur Fertigung von Werkstücken notwendig sind zur Modellierung zur Verfügung. Es wurde eine Auswahl der gebräuchlichsten und der am häufigsten verwendeten Verfahren realisiert. Die dritte implementierte Klasse ist die Klasse der Werkzeugmaschinen. Sie dient zur Darstellung der wichtigsten geometrischen und technologischen Daten einer Werkzeugmaschine. So können beispielsweise Aufspannverhältnisse dargestellt und eine Bewegungssimulation durchgeführt werden. Die zugefügte Klassenhierarchie stellt im allgemeinen eine Featurehierarchie dar. Im Gegensatz zu anderen Systemen werden die Features auf eine andere Art und Weise generiert. Sie bilden die Fertigungsmöglichkeiten und Erfahrungswerte der Fertigung vollständig ab. Der Konstrukteur kann sein Modell nicht nur partiell sondern vollständig unter direktem Zugriff von Erfahrungswerten und Fertigungsnormen durchführen.

Der Modellaufbau erfolgt vom Rohteil über die Anwendung verschiedener Fertigungsverfahren zum Fertigteil, was bei anderen Systemen so nicht zwingend vorgegeben ist. Der Konstrukteur soll aber nicht mit zu komplexen Eingaben überfordert werden. Technologische Daten sind daher in der Regel wahlweise anzugeben.

6.1 Halbzeuge

Typisches Ausgangsprodukt jeder Fertigung sind zu bearbeitende Halbzeuge, fertigzustellende Gußteile oder nachzuarbeitende Fertigteile. Für die beiden letztgenannten ist eine Systematisierung für den fertigungsorientierten Modellierungsprozeß nicht notwendig.

Um ein Einzelteil zu fertigen muß zunächst ein zu bearbeitendes Teil vorhanden sein. Dies ist typischerweise ein Halbzeug. Es läßt aufgrund seiner Formgebung eine grobe Gliederung zu. Eine detaillierte Gliederung ist durch die Vielzahl der möglichen Form - Werkstoffpaarungen nicht möglich. Für die fertigungsorientierte Modellierung wurden die Halbzeuge zunächst in die Gruppen „Profile“, „Bleche“ und „Rohre“ eingeteilt. Da sich die Implementation des Modellierers zunächst nur auf die spanende Fertigung konzentriert, spielt die Klasse „Blech“ nur eine untergeordnete Rolle.

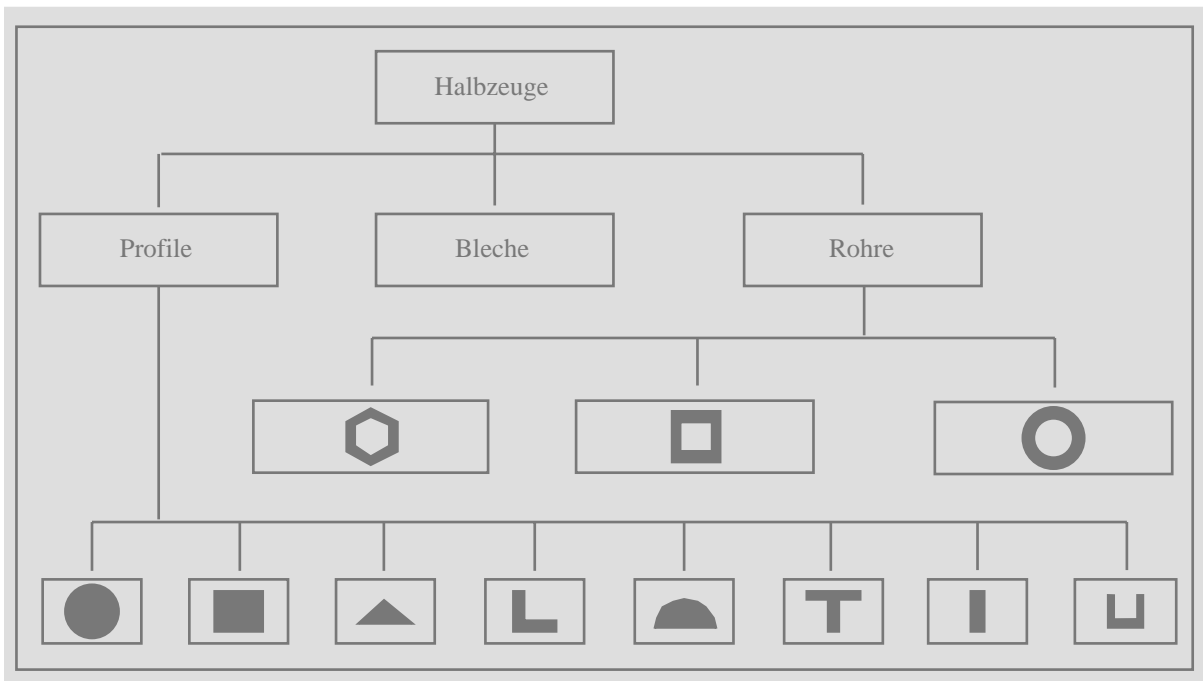


Bild 6.1 Gliederung Halbzeuge

Äus dem Blickwinkel der Fertigungstechnik gewinnt die Blechbearbeitung durch 3D-CAD Systeme und durch featurebasierende CAD Systeme allerdings zunehmend an Bedeutung. Die aus wirtschaftlichen Gründen immer komplizierter gestalteten Biegeteile sind mit den traditionellen Methoden nicht mehr herstellbar. Erst auf 3D-CAD Systemen basierende Abwicklungs- und Biegesimulationsprogramme erlauben die Fertigung der geforderten Bauteile /76/.

Von der Klasse „Profile“ sowie der Klasse „Rohre“ wurden weitere Klassen abgeleitet. In [Bild 6.1](#) ist der prinzipielle Aufbau der Klassenhierarchie Halbzeuge dargestellt. Durch sie kann eine Auswahl der gebräuchlichsten Halbzeuge vorgenommen werden.

6.1.1 Die Klasse „t_halfzeug“

Die Klasse *t_halfzeug* wird von der Basisklasse *t_object* der CV-DORS Klassenhierarchie abgeleitet. Damit erbt sie die Methoden des Kopierens, des Entfernens (Löschen) und der Fehlerbehandlung. *t_halfzeug* dient selbst wiederum als Basisklasse. Für die von ihr abgeleiteten Klassen *t_profile*, *t_bleche* und *t_rohre* werden in ihr Informationen über den Werkstoff, die Länge, die Toleranzfelder, die Dichte sowie die Oberflächenqualität in geschützten Variablen abgelegt. Das nachfolgende [Bild 6.2](#) zeigt den Aufbau der Klassenhierarchie Halbzeuge.

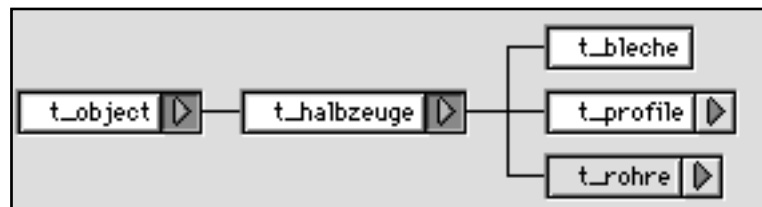


Bild 6.2 Die Klasse *t_halfzeug* als Basisklasse

Die Funktionen zum Ändern, Auslesen und Setzen dieser Variablen stehen zur Verfügung. Die virtuellen Funktionen zur Berechnung des Volumens, der Querschnittfläche, des axialen Widerstandmoments, des polaren Widerstandmoments sowie des Flächenträgheitsmoments wurden implementiert. Die letztgenannten Werte stehen für mechanische Betrachtungen zur Verfügung.

```
class t_halfzeuge:t_object{
protected :
    char * mName;           // Bezeichnung des Halbzeuges z.B. Blech, Rohr, Sechskantrohr
    char * mWerkstoff;     // z.B. Ust37-2 oder 1.0036
    char * mHerstellverfahren; // gezogen, ...
    char * mToleranz;      // Toleranzfelder z.B. h9, h12 ...
    char * mOberflaechenart; // Poliert, Blank...
    char * mDINBezeichnung; // Rohr : DIN 2391
    double mLaenge;
    TQ3GroupObject mModel;
public :
    t_halfzeuge();
    ~t_halfzeuge();
    virtual double mElementVolume(){return 0.0; };
    virtual double mQuerschnittsflaeche(){return 0.0; };
    virtual double mAxialesFlaechentraegheitsmoment(){return 0.0; };

    void GetHalbzeugeName( char * pHalbzeugName, int pStrLength);
    void GetDINBezeichnung( char * pDINBezeichnung, int pStrLength);
    void GetHalbzeugeWerkstoff( char * pWerkstoffName, int pStrLength);
    void GetHalbzeugeHerstellverfahren( char * pWerkstoffName, int pStrLength);
    void GetHalbzeugeToleranz( char * pWerkstoffName, int pStrLength);
    void GetHalbzeugeOberflaechenart( char * pWerkstoffName, int pStrLength);
    TQ3GroupObject GetGeometrieObject();

    void SetHalbzeugeName( char * pHalbzeugName, int pStrLength);
    void SetDINBezeichnung( char * pDINBezeichnung, int pStrLength);
    void SetHalbzeugeWerkstoff( char * pWerkstoffName, int pStrLength);
    void SetHalbzeugeHerstellverfahren( char * pWerkstoffName, int pStrLength);
    void SetHalbzeugeToleranz( char * pWerkstoffName, int pStrLength);
    void SetHalbzeugeOberflaechenart( char * pWerkstoffName, int pStrLength);
    void SetGeometrieObject(TQ3GroupObject pMmodel);
};
```

Bild 6.3 Die Klasse Halbzeug als C++ Klasse

Bild 6.3 zeigt die Klassendefinition der Basisklasse *t_halfzeug* mit den kommentierten Member-Variablen sowie den verfügbaren Funktionen.

6.1.2 Die Klasse „t_profile“

Von der Klasse *t_halfzeug*, der Basisklasse der Halbzeuge, wurden die Klassen *t_profile*, *t_blech* sowie *t_rohre* abgeleitet. Die Klasse *t_profile* dient wiederum als Basisklasse für die Klassen *t_profile_stange*, *t_profile_rechteck*, *t_profile_vierkant* sowie *t_profile_X*. Wobei X stellvertretend für I-, T- und U-Profile steht. Bild 6.4 veranschaulicht den Klassenaufbau der Klasse *t_profile* sowie der von ihr abgeleiteten Klassen. Allen Profilen gemein sind die in der

Klasse *t_halfzeug* getroffenen Angaben zum Werkstoff, zur Oberflächenbeschreibung, zu den Toleranzen sowie die Angabe zur Abschnittslänge. Die von der Klasse *t_profile* abgeleiteten Unterklassen ergänzen diese Daten um geometrie- und berechnungsspezifische Daten. Die entsprechenden Parametrisierungen der Profilegeometrien sowie die Generierung der Profilkörper findet in den jeweiligen Unterklassen statt. Neben den geometrischen Daten und den auf diese Daten angewandten profilspezifischen Funktionen werden von den jeweiligen Profilen Berechnungsdaten und -Funktionen verwaltet. Zu diesen zählen Funktionen zur Berechnung des Volumens, der Profilfläche und des Flächenträgheitsmomentes.

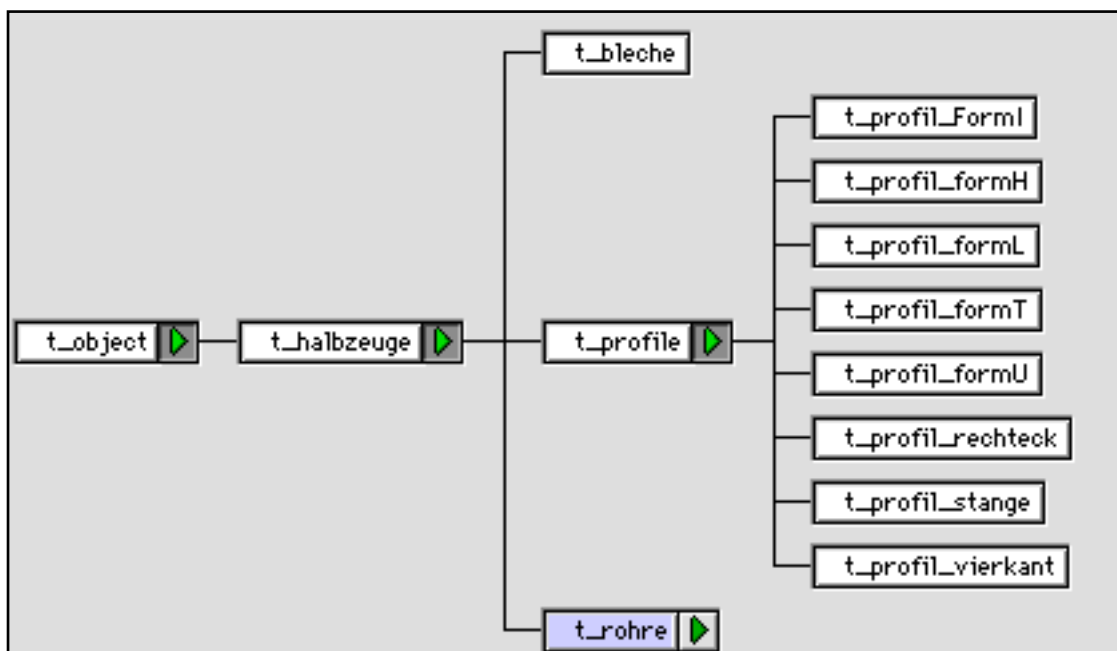


Bild 6.4 Die Klasse *t_profile* sowie von ihr abgeleitete Klassen

Das Bild 6.4 spiegelt den Einfluß des Klassenaufbaus der Klasse *t_profile* auf die Struktur der Eingabemenüs wieder. Es zeigt die Implementation einer Auswahlmöglichkeit der verschiedenen Profile. Durch den Hauptauswahlpunkt Halbzeuge der Menüleiste erfolgt über den Untereintrag Profile das Anzeigen eines Dialogfeldes. In diesem können durch ein Auswahlelement die verschiedenen Eigenschaftsblätter zu den jeweiligen Profilen selektiert werden. Die Eigenschaftsblätter enthalten die notwendigen Eingabemasken für die zur Verfügung stehenden Profilarten. Die Eingabe der geometrischen Daten ist zwingend vorgeschrieben, die Eingaben der Zusatzinformationen über den verwendeten Werkstoff müssen nicht zwangsweise erfolgen.



Bild 6.5 Auswahlmöglichkeiten von Profilen

Alle verwendeten Profile wurden parametrisiert. Die Parameter bzw. Parametersätze können dabei aus Normen, firmenspezifischen Normen und Zulieferkatalogen übernommen werden. Im nachfolgenden Bild ist das Modell eines parametrisierten I-Profils (Doppel-T-Träger) /10 und 54/ dargestellt. Als variable Größen gehen die Abschnittslänge, die Breite, die Stegdicke und die Gesamthöhe des Profils ein.

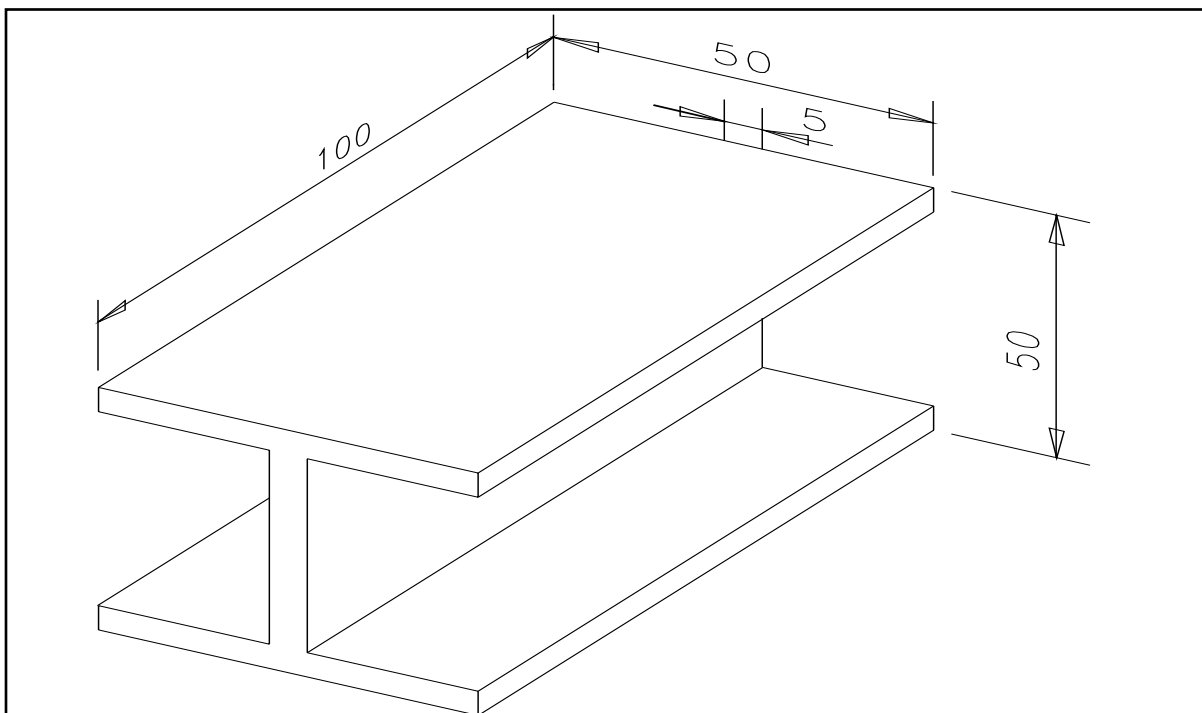


Bild 6.6 Die Parametrisierung eines I-Profils /125/

Zur Erzeugung des Halbzeuges werden zuerst Systemparameter wie die Farben und der Layer ausgelesen und anschließend mit eigenen, gewünschten Werten belegt. Ein lokales Koordinatensystem wird erzeugt. In diesem wird die 2d-Kontur durch ein Parameterprogramm generiert. Ein Verschiebevektor, beginnend im Bezugspunkt der Kontur und in die Richtung des Spannmittels zeigend, wird erzeugt. Seine Länge entspricht der Abschnittslänge des Halbzeuges. Entlang des definierten Vektors wird die Kontur verschoben. Ein prismatischer Körper entsteht. Das Koordinatensystem sowie eventuell weitere geänderte Voreinstellungen wie die Farben und der Layer werden rückgesetzt.

6.1.3 Die Klasse „t_bleche“

Die Klasse *t_bleche* besitzt keine weiteren, von ihr abgeleiteten Klassen. Sie dient zur Aufnahme der geometrischen Abmessungen des zu erzeugenden Werkstückes (Zuschnitt), den Abmaßen des Rohteils (Blechtafel), dem zu erzeugendem Fertigteil sowie der Bearbeitungsfolge mit ihren jeweils entsprechenden Parametern. Die moderne Herstellung komplexer Blechteile weicht von der fertigungsorientierten Modellierung ab. Bei der fertigungsorientierten Modellierung wird von einem Rohteil ausgehend eine Bearbeitungsreihenfolge von Fertigungsverfahren auf dieses Rohteil angewendet, um zu einem Fertigteil zu gelangen. Bei der Entwicklung einer Blechkonstruktion erzeugt man in einem ersten Schritt ein Modell des gebrauchsfertigen Blechbiegeteils. Dieses Modell wird im anschließendem Schritt auseinandergefaltet. Die Abwicklung entsteht. Die Abwicklung ist aber im Sinne der fertigungsorientierten Modellierung das Rohteil auf das eine Bearbeitungsfolge angewendet werden kann. Dieser Widerspruch kann durch das Feature Based Design gelöst werden. In /139 und 76/ werden Softwaresystem zur Blechbearbeitung vorgestellt. Das nachfolgende Bild zeigt verschiedene in den entsprechenden Systemen verwendete Gestaltelemente.

Nach dem Modellaufbau durch Features wird die Abwicklung des Modells automatisch generiert. Mit dieser muß sichergestellt werden, daß das erzeugte Modell auch hergestellt werden kann. Hierzu bedarf es einer Biegesimulation. Sie führt mit der Abwicklung die ermittelten Biegungen durch und prüft dabei die Kollisionsfreiheit.

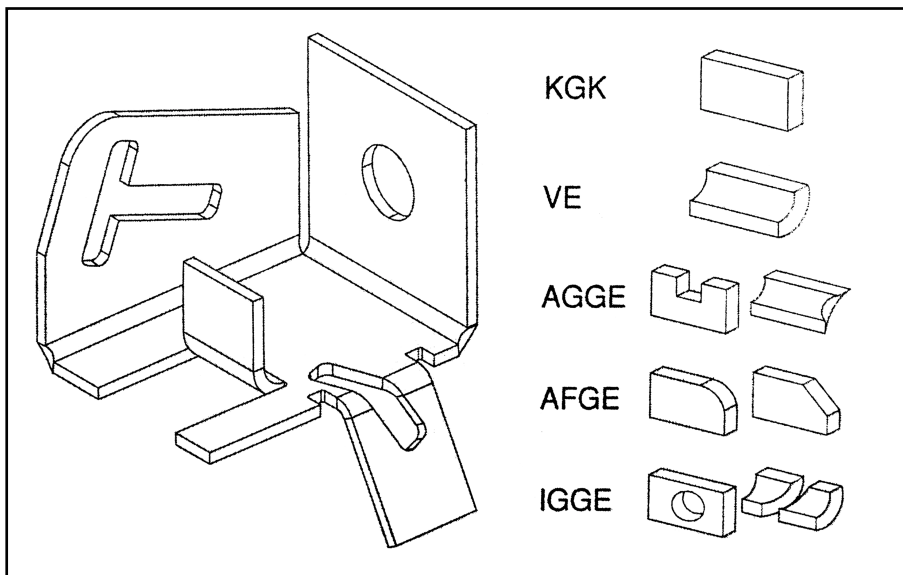


Bild 6.7 Verwendete Gestaltelemente eines Biegeteils /76/

An die verwendeten Features /73/ besteht hierbei die Anforderung, daß sie firmenspezifisch erweitert werden können bzw. firmenspezifische Vorgaben nutzen. Hierzu gehört die Definition von Stegbreiten, Lochdurchmessern und Biegeradien.

6.1.4 Die Klasse „t_rohre“

Die Klasse *t_rohre* wird von den Klassen *t_rohre_sechskant*, *t_rohre_rechteck* und *t_rohre_kreis* als Basisklasse genutzt. Sie sind in der Form der öffentliche Klassen von *t_rohre* abgeleitet.

Der in Bild 6.8 gezeigte Klassenaufbau ergibt sich. Die Gemeinsamkeiten dieser Klassen besteht neben den in der Klasse *t_halfzeuge* definierten Eigenschaften darin, daß sie einen Durchmesser und eine Wandstärke besitzen.

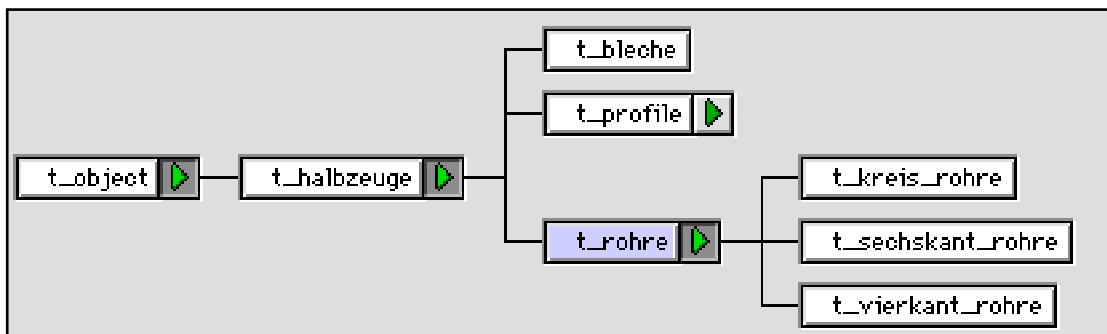


Bild 6.8 Objekthierarchie der Klasse *t_rohre*

Diese Werte sind als geschützte Variable der Basisklasse *t_rohre* definiert. Die entsprechenden Funktionen zum Setzen und Auslesen dieser Werte sind ebenfalls in der Klasse *t_rohre* implementiert worden. Als gemeinsame Funktionen sind die Methoden zum Berechnen der Querschnittsfläche, des Trägheitsmomentes sowie des Volumens definiert und erstellt worden.

Das nachfolgende [Bild 6.9](#) zeigt die Möglichkeiten der Auswahl der Halbzeuge vom Typ Rohr. Über das Auswahlfenster des Dialogs können die Eigenschaftsblätter des Kreisrohrs, des Vierkantrohrs sowie das Sechskantrohrs ausgewählt werden. Gezeigt ist das Eigenschaftsblatt des Kreisrohres.



Bild 6.9 Auswahl und Definition eines Halbzeuges

Diese enthalten die notwendigen Datenfelder des jeweiligen Halbzeuges. Angaben zur Geometrie wie die Abschnittslänge, der Durchmesser sowie die Wandstärke müssen getätigt werden. Die Angabe zum Werkstoff sowie die Angabe zur DIN ist wahlweise und muß nicht ausgefüllt werden. Nachfolgend ist die Implementation der Klasse Kreisrohre dargestellt. Sie ist von der Basisklasse Rohre abgeleitet.

```
class t_kreis_rohre : public t_rohre
{
public :
    t_kreis_rohre();
    ~t_kreis_rohre();
    double mQuerschnittsflaeche();
    double mAxialesFlaechentraegheitsmoment();
    double mElementVolume();
};
```

Bild 6.10 Die Implementation der Klasse Rohre mit Kreisquerschnitt

Da der Durchmesser sowie die Wandstärke des Rohres bereits in der Basisklasse definiert wurde, müssen lediglich die von der Profilart abhängigen Funktionen zur Berechnung der Querschnittsfläche, des Flächenträgheitsmomentes sowie des Volumens implementiert werden.

6.2 Fertigungsverfahren

In der vorliegenden Arbeit war es aufgrund der Vielzahl der Fertigungsverfahren nicht möglich, alle existierenden in dem fertigungsorientiertem Modellierer zu implementieren. Es wurde eine Auswahl der zu implementierenden Verfahren getroffen. Dabei wurde insbesondere berücksichtigt, wie oft die Verfahren im industriellen Einsatz Verwendung finden. Ein weiteres Auswahlkriterium ist die Eignung eines Verfahrens für die fertigungsorientierte Modellierung. Bei einigen, wie dem Schneidverfahren, ist die geometrische Modellierung vollkommen ausreichend. Einige Verfahren wie z.B. die Blechverarbeitung stellen aufgrund ihrer speziellen Anforderungen, siehe Kapitel 6.1.3, eine Sonderrolle für die fertigungsorientierte Modellierung dar.

Durch die Klasse Fertigung werden die drei Hauptverfahren des Bohrens, Drehens und FräSENS für Modellierungszwecke verfügbar gemacht. Um eine Klassenhierarchie zu entwickeln wurden die genannten Fertigungsverfahren zunächst systematisiert /9/.

Hierzu wurden die DIN 8589 T0, T1, T2, T3 und T6 auf alle wichtigen spanabhebenden Fertigungsverfahren im Hinblick auf die fertigungsorientierte Modellierung untersucht. Für alle Fertigungsverfahren und ihre Unterverfahren wurden Datenblätter ermittelt. Die Datenblätter wurden nach folgenden Kriterien erstellt:

Hauptverfahren, Unterverfahren, Verfahrenselement, Parameter, deren Tolerierung, Wertebereiche und Standardwerte, entstehende Spankörper, gegebenenfalls Nebenkörper, entstehende Oberflächen, deren Rauheitsgrade (Oberflächengüte) Werkzeuge und Einsatzgrenzen.

Ergebnis diese Arbeitsschrittes war eine Gliederung der Fertigungselemente in Hauptverfahren, Unterverfahren und Verfahrenselemente. Eine Gruppe Formelemente wurde zu jedem Hauptverfahren hinzugefügt.

In Bild 6.11 sieht man die Klassenhierarchie ohne die Wurzelklasse *t_object* sowie ohne die abgeleiteten Klassen der Verfahrenselemente.

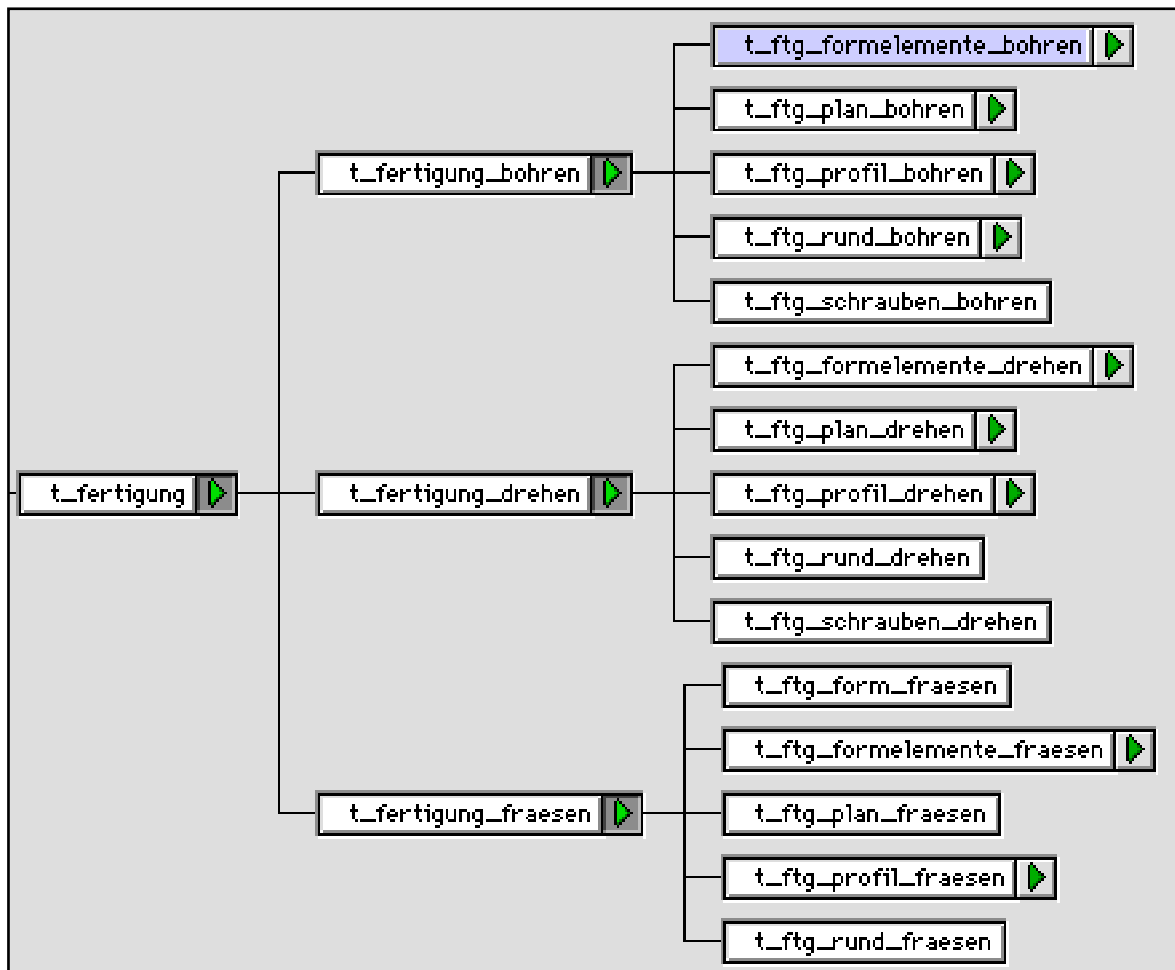


Bild 6.11 Die Klassenhierarchie *t_fertigung*

Zu jedem Verfahrenselement und Formelement wurde eine Parametertabelle erstellt. In ihnen sind die zur fertigungsorientierten Modellierung notwendigen Daten und Berechnungsverfahren enthalten. Die Wurzel dieser Klassenhierarchie ist die Klasse *t_fertigung*, die von der Klasse *t_object*, der Wurzel der CV-DORS- Klassenhierarchie abgeleitet wurde. Als Unterverfahren wurden die Klassen *t_drehen*, *t_fraesen* und *t_bohren* von der Klasse *t_fertigung* abgeleitet. Von diesen drei Hauptverfahren wurde die jeweiligen Unterverfahrens- und Formelementeklassen abgeleitet.

Eine weitere Unterteilung der Unterverfahren in Verfahrenselemente wurde durchgeführt. Für das Unterverfahren Profildrehen des Hauptverfahrens Drehen wurden z.B. die Verfahrenselemente Längsprofildrehen, Längsprofileinstechdrehen, Querprofildrehen und Querprofileinstechdrehen implementiert. Als Formelemente des Hauptverfahrens Bohren wurden unter anderem die Durchgangs-, die Sack-, die Stufen- und die Gewindebohrung realisiert.

6.2.1 Das Fertigungsverfahren Drehen

Das Fertigungsverfahren Drehen wird in die Unterverfahren Plandrehen, Runddrehen, Schraubendrehen Profildrehen und Formdrehen unterteilt. Von diesen Unterverfahren wird das Querplandrehen, das Querabstechdrehen, das Längsrunddrehen sowie aus der Gruppe der Formelemente der Sicherungseinstich dargestellt.

Aus den zu diesen Verfahren ermittelten Parametertabellen wurden Programme zur Generierung der entsprechenden Geometrien entwickelt. Datenstrukturen zur Aufnahme der Werte sowie zur Verwaltung der Geometriedaten wurden geschaffen. Ein Ablauf zur Modellierung wurde festgelegt.

Bild 6.12 zeigt wie sich die technologische Gliederung des Fertigungsverfahrens Drehen auf die Menügestaltung ausgewirkt hat. In der Hauptmenüauswahl Modellierung können die beiden Popup-Menüs „Drehen“ sowie „Formelemente des Drehens“ selektieren werden.

Beide Auswahlpunkte erzeugen jeweils ein Dialogfenster. Diese besitzen ein sogenanntes „Cardlayout“. Hierzu wird eine Auswahlleiste sowie ein Bereich für Eigenschaftsblätter auf dem Dialogfenster positioniert. Die jeweiligen Unterverfahren besitzen jeweils ein Eigenschaftsblatt und sind durch die Auswahlleiste anwählbar.

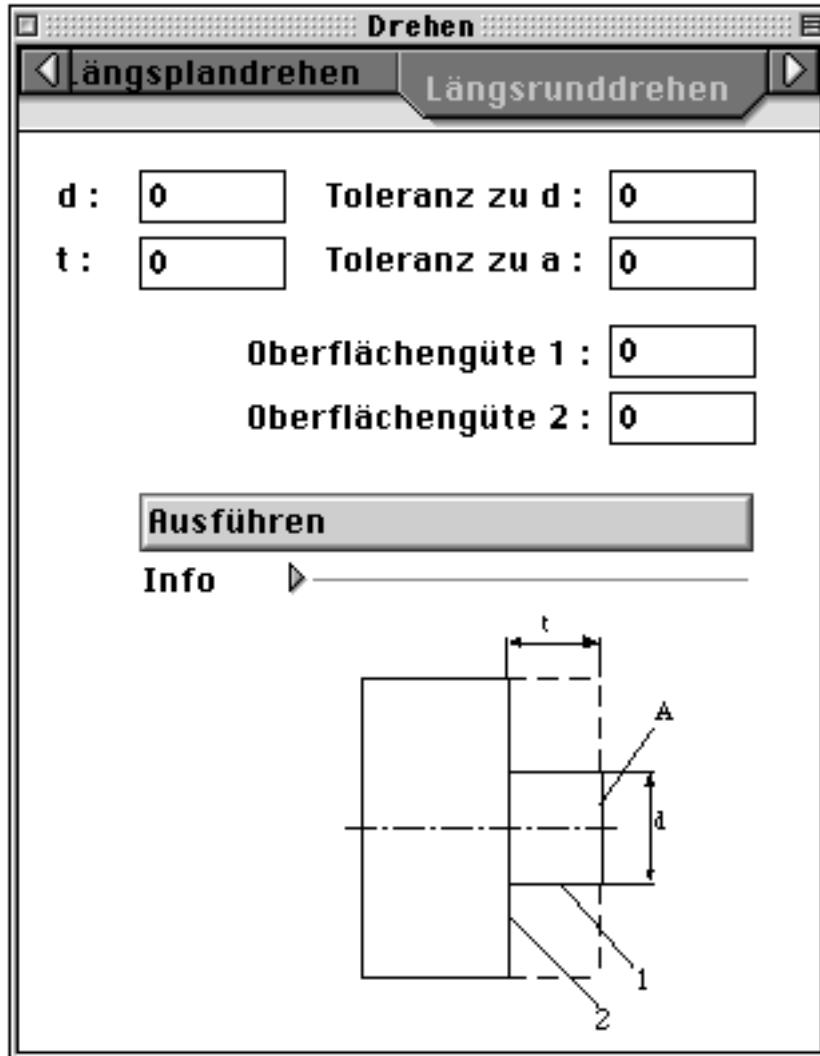


Bild 6.12 Menüauswahl Längsrunddrehen mit ausgeklappten Informationsblatt

Im [Bild 6.12](#) ist der Aufbau des Fensters zum Ausführen des Fertigungsverfahren Längsrunddrehen veranschaulicht. Wie bei den Halbzeugen so gilt auch bei den Fertigungsverfahren, dass die geometrisch notwendigen Angaben angegeben werden müssen. Die Eingaben zur Toleranz und zur Oberflächengüte sind wahlweise einzugeben.

6.2.1.1 Das Unterverfahren Querplandrehen

Beim Verfahren des Plandrehens wird das Werkzeug senkrecht zur Drehachse des Werkstückes bewegt. [Bild 6.13](#) zeigt beispielhaft das Datenblatt für das implementierte Drehverfahren Querplandrehen. Als notwendige Parameter dieses Fertigungsverfahrens werden eine Bezugsfläche, eine Längenangabe sowie eine Durchmesserangabe benötigt. Die Bezugsfläche erfolgt durch das Identifizieren einer Fläche. Der Parameter Längenangabe sowie der Parameter

Durchmesser werden durch Eingabefelder im Eigenschaftsblatt ermittelt. Zu diesen beiden Parametern werden die jeweiligen entsprechenden Toleranzen benötigt. Diese sind mit Standardwerten vorinitialisiert, können aber im Eigenschaftsblatt verändert werden. Zur vollständigen Definition des Fertigungsverfahrens verwaltet das Programmsystem die Oberflächengüten der entstehenden Oberflächen.

Hauptverfahren : Drehen Unterverfahren : Plandrehen Verfahrenselement : Querplandrehen	
Parameter :	A: Bezugsfläche d: zu erzeugender Durchmesser a: Abstand zu A
Toleranzangaben :	d, a
Spankörper :	$d = 0, V = (\pi * a * D^2) / 4$ $d > 0, V = (\pi * a * (D^2 - d^2)) / 4$
erzeugte Oberfläche :	1. Kreisfläche, Kreisringfläche 2. Zylindermantel
Oberflächengüte 1 :	$R_z 250 - R_z 2,5$
2 :	$R_z 160 - R_z 1,6$

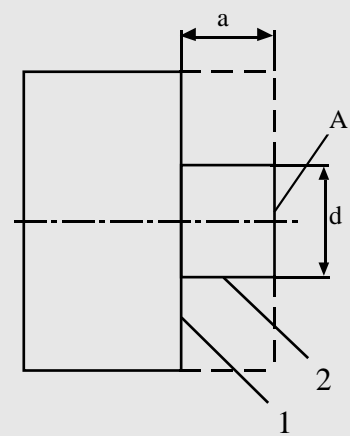


Bild 6.13 Datenblatt Drehen/Plandrehen/Querplandrehen /64/

Als Spankörper entsteht, abhängig von der Durchmesserangabe entweder ein Zylinder oder eine Hülse. Diese beiden Körper werden automatisch vom System generiert. Der Zylinder kann direkt vom Modellierer durch die Klasse `t_brep::t_brep` erzeugt und vom zu bearbeitenden Werkstück subtrahiert werden. Zur Erzeugung der Hülse wird zunächst ein rechteckiger Querschnitt mit dem Radius $d/2$ zur Drehachse des Werkstückes generiert.

Die Konturelemente werden durch die Funktion `t_sweep_profile` in ein Profilelement gewandelt. Die ursprünglichen Konturelemente bleiben erhalten. Das erzeugte Profilelement wird beim Aufrufen des Konstruktors `t_brep` an diesen mit den entsprechenden Winkelangaben übergeben.

Der Spankörper Hülse wird erzeugt. Zur Berechnung des Spanvolumens wird die Schnittmenge zwischen dem Werkstück sowie dem erzeugtem Spankörper gebildet. Die Memberfunktion `t_brep::get_mass_properties` liefert die benötigten Daten zum Volumen.

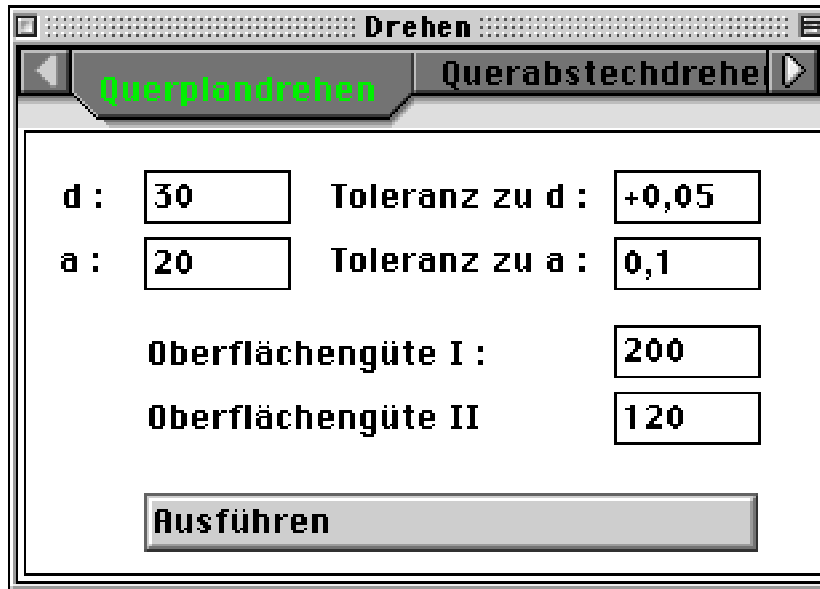


Bild 6.14 Dateieingaben zum Verfahren Querplandrehen

Ausgelöst werden die beschriebenen Aktionen durch das in [Bild 6.14](#) gezeigte Menü. In ihm werden die geometrischen Daten eingegeben. Diese müssen zwangsweise vorhanden sein. Wahlweise können Werte für die Toleranzen und Werte für die Oberflächengüte eingetragen werden.

6.2.1.2 Das Verfahren des Querabstechdrehens

In [Bild 6.15](#) ist das Verfahren des Querabstechdrehens dargestellt. Das Verfahren dient dem Zwecke des Abtrennens des Werkstückes oder von Werkstückteilen. Neben dem Spankörper entsteht ein Nebenkörper. Um den Nebenkörper zu erzeugen wird ein Zylinder erzeugt, dessen Durchmesser größer als der des Nebenkörpers ist.

Seine Länge ist größer als das Maß a und das Maß b . Der Ursprung dieses Zylinders liegt auf der Drehachse. Er ist um das Maß a vom Ende des Werkstückes hin zum Ursprung des Werkstückes verschoben. Durch die Bildung der Schnittmenge dieses Zylinders mit dem Werkstück entsteht der Nebenkörper. Dieser wird zur weiteren Verwendung in einer internen Datenstruktur abgelegt. Zur Bearbeitung des Werkstückes wird der Zylinder vom Werkstück subtrahiert. Dabei wird das Werkstück modifiziert.

Hauptverfahren : Drehen Unterverfahren : Plandrehen Verfahrenselement : Querabstechdrehen	
Parameter :	A: Bezugsfläche b: Einstechbreite a: Abstand zu A
Toleranzangaben :	b, a
Spankörper :	$V = (\pi * b * D^2) / 4$ $V = (\pi * a * D^2) / 4$
erzeugte Oberfläche :	1. Kreisfläche
Oberflächengüte :	1. $R_z 250 - R_z 2,5$

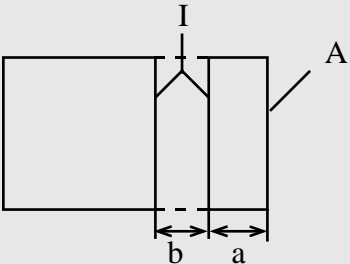


Bild 6.15 Parametertabelle Drehen/Plandrehen/Querabstechdrehen /64/

Der entstandene Körper entspricht jetzt dem zu erzeugendem Werkstück inklusive des Spankörpers. Um den Spankörper zu erzeugen wird der Zylinder jetzt um den Betrag b vom Ende des Werkstückes hin zum Ursprung des Werkstückes verschoben. Dann wird wiederum die Schnittmenge mit dem Werkstück gebildet. Die Schnittmenge entspricht dem Spanvolumen. Um die Werkstückgeometrie zu verändern wird der Zylinder vom Werkstück subtrahiert. Die notwendigen Parameter dieses Verfahrens sind die Bezugsfläche, die Einstechbreite b sowie die Länge a des Nebenkörpers.

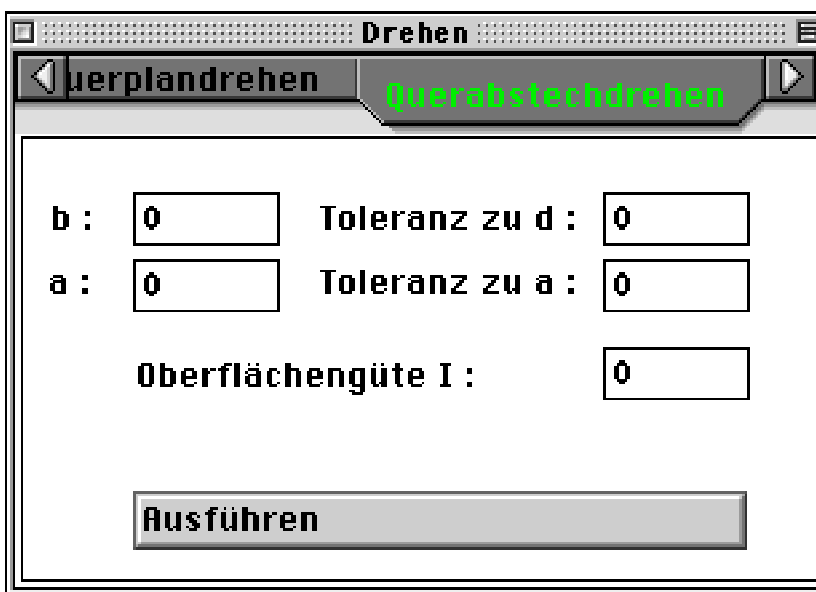


Bild 6.16 Das Menü Querabstechdrehen

Zu den beiden Längenangaben benötigt dieses Verfahren jeweils eine Toleranzangabe. Der Wert der Oberflächengüte wird sowohl in der Datenstruktur des Werkstückes als auch in der Datenstruktur des Nebenkörpers verwaltet.

Durch das in Bild 6.16 dargestellte Menü werden die Daten zum Durchführen des Verfahrens Querabstechdrehen eingelesen. Notwendige Eingaben sind die Daten zur Geometrie. Die Oberflächengüte sowie die Toleranz können wiederum wahlweise in das Datenblatt eingetragen werden.

6.2.1.3 Das Längsplandrehen

In Bild 6.17 ist das Datenblatt zum Längsplandrehen dargestellt. Es ist ein Plandrehen mit Vorschub parallel zur Drehachse des Werkstückes. Dabei ist die Schneide des Drehmeißels mindestens so breit wie die am Werkstück entstehende ringförmige ebene Fläche. Als Spankörper entsteht eine Hülse. Diese wird wie bereits beim Verfahren Querplandrehen beschrieben erzeugt. Das Spanvolumen wird über die Schnittmenge der beiden Körper berechnet. Die neu entstandene Geometrie des Werkstückes wird durch die Subtraktion der Hülse erreicht.

Hauptverfahren : Drehen Unterverfahren : Plandrehen Verfahrenselement : Längsplandrehen	
Parameter :	A: Bezugsfläche t: Tiefe
Toleranzangaben :	t
Spankörper :	$V = (\pi * t * (D^2 - d^2)) / 4$
erzeugte Oberfläche :	1. Kreisringfläche
Oberflächengüte I. :	Rz 250 - Rz 2,5

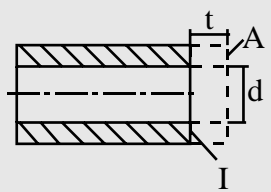


Bild 6.17 Datenblatt Drehen/Plandrehen/Längsplandrehen /64/

Die Bezugsfläche A sowie die Tiefe t sind die beiden notwendigen Parameter des Fertigungsverfahrens Längsdrehen. Zur Tiefe t kann eine Toleranz vorgegeben werden, was jedoch nicht zwingend notwendig ist. Zu der erzeugten Kreisringfläche kann die zugehörige Oberflächengüte über ein Eigenschaftsblatt eingetragen werden.

6.2.1.4 Profildrehen

Bei diesem Verfahren wird ein Profilwerkzeug genutzt, um ein Profil auf das Werkstück abzubilden. Dabei wird dieses Unterverfahren in die Verfahrenselemente Längsprofildrehen, Längsprofileinstechdrehen, Querprofildrehen und Querprofileinstechdrehen unterteilt. Beim Längsprofildrehen wird der Vorschub parallel zur Drehachse des Werkstückes ausgeführt. Die Schneide des Profilmeißels entspricht der Breite der zu erzeugenden Fläche. Das Längsprofileinstechdrehen dient dazu, auf der Stirnseite des Werkstückes ringförmige Einstiche zu erzeugen. Beim Querprofileinstechdrehen erfolgt der Vorschub quer zur Drehachse des Werkstückes. Die Schneide des Profilmeißels ist so breit wie die zu erzeugende Fläche. Das Querprofileinstechdrehen dient dazu, ringförmige Einstiche an der Zylinderfläche eines Werkstückes anzubringen. Der Vorschub des Werkzeugs erfolgt senkrecht zur Drehachse.

Um Spankörper für die genannten Profilverfahren zu erzeugen, wird zuerst eine 2d-Kontur erzeugt. Ihre Position und Lage entspricht der zu erzeugenden Werkstückkontur. Die Konturenenden werden je nach Verfahren in Richtung der Drehachse oder quer zur Richtung der Drehachse über die Kontur des Werkstücks verlängert und geschlossen. Die so erzeugte geschlossene Kontur wird durch Rotation um die Drehachse des Werkstückes zu einem Körper. Zur Erzeugung des Spankörpers wird die Schnittmenge mit dem Werkstück gebildet.

Hauptverfahren : Drehen Unterverfahren : Profildrehen Verfahrenselement : Querprofileinstechdrehen	
Parameter :	A: Bezugsfläche a: Mittenabstand von A $a > 0$ t: Tiefe $t > 0$
Toleranzangaben :	t, a
Spankörper :	abhängig von Profilform
erzeugte Oberfläche :	abhängig von Profilform
Oberflächengüte I. :	Rz 160 - Rz 4

Bild 6.18 Parameter des Verfahrenselement Querprofileinstechdrehen /64/

Das Werkstück wird durch Subtraktion des Körpers auf seine gewünschte Gestalt gebracht. In Bild 6.18 ist das Verfahrenselement Querprofileinstechdrehen mit seinen Parametern als typischer Vertreter des Profildrehens dargestellt. Notwendige Parameter sind wiederum eine Bezugsfläche, ein Abstand sowie eine Tiefe. Der Abstand und die Tiefe beziehen sich bei diesem Fertigungsverfahren auf die Profilkanten bzw. die Profilmitten. Die Toleranzangaben zu den Werten Abstand und Tiefe sind wahlweise zu bestimmen.

Das Erzeugen der Kontur kann auf zwei verschiedene Arten erfolgen. Eine Möglichkeit besteht in der Auswahl einer Kontur nach der DIN. Hierbei können z.B. die Formelemente Sprengringeinstich und Sicherungsringeinstich ausgewählt werden. Aus vordefinierten Tabellen können die Parametersätze in dieses Fertigungsverfahren übertragen werden.

Die zweite Möglichkeit besteht darin, eine 2d-Kontur über die geometrieorientierte Funktionalität des Programmsystems zu erzeugen. Hierbei wird durch einen „Sketcher“ die Kontur aus 2d-Elementen erzeugt oder aus einer Bibliothek abgerufen und modifiziert. Als Schnittstelle zum fertigungsorientierten Modellierer dient eine Liste der erzeugten Geometrieelemente. Diese Liste kann wahlweise, falls sie nicht aus einer Bibliothek geladen wurde, als Muster abgelegt werden. Der Aufbau werksinterner Normen wird damit unterstützt.

Die Oberflächengüte bezieht sich bei diesem Verfahren auf die erzeugte Profilfläche. Sie ist wahlweise einzugeben.

6.2.1.5 Das Längsrunddrehen

Bei dem Verfahren des Runddrehens erfolgt die Bewegung des Werkzeugs koaxial zur Achse des Werkstücks. Bei der Durchführung des Verfahrens ist zunächst die Bezugsfläche A zu identifizieren. Weitere notwendige Angaben zur Durchführung dieses Verfahrens sind die Angabe des Durchmessers d auf den das Werkstück abgedreht werden soll sowie die Längenangabe t , die der Abschnittslänge entspricht. Zur Abschnittslänge und dem Durchmesser kann jeweils eine Toleranzangabe erfolgen. Die Oberflächengüten des entstehenden Zylindermantels sowie der entstehenden Kreisringfläche können vorgegeben werden. Alle Werte werden über ein entsprechendes Eigenschaftsfeld vorgegeben. Im nachfolgendem Bild 6.19 sind die Parameter sowie ihre Bedeutung dargestellt.

Hauptverfahren : Drehen Unterverfahren : Runddrehen Verfahrenselement : Längsrunddrehen	
Parameter :	A: Bezugsfläche d: zu erzeugender Durchmesser $d > 0$ t : Tiefe $t > 0$
Toleranzangaben :	d, t
Spankörper :	$V = (\pi * t * (D^2 - d^2)) / 4$
erzeugte Oberfläche :	1. Zylindermantel 2. Kreisringfläche
Oberflächengüte :	1. $R_z 250 - R_z 1$ 2. $R_z 160 - R_z 4$

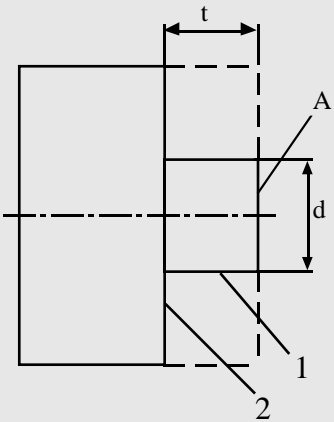


Bild 6.19 Parametertabelle Drehen/Runddrehen/Längsrunddrehen /64/

Der Spankörper der bei diesem Fertigungsverfahren entsteht ist eine Hülse. Zu deren Erzeugung werden zunächst zwei Kreise in der identifizierten Ebene A generiert. Ihr Mittelpunkt liegt in der Drehachse des Werkstücks. Die beiden Kreise werden entlang eines Vektors der Länge t koaxial zur Werkstückachse in Richtung der Einspannung bewegt. Die Schnittmenge der entstehenden Hülse mit dem Werkstück wird als Spanvolumen abgelegt.

In einer zweiten Operation wird die Hülse von der Werkstückgeometrie subtrahiert. Dadurch entsteht die Werkstückgeometrie nachdem das Fertigungsverfahren angewendet worden ist. Zum Einlesen der benötigten Daten wurde ein entsprechendes Eigenschaftsfeld generiert. In diesem müssen die notwendigen geometrischen Daten wie die Abschnittslänge t und der Durchmesser d eingetragen werden.

Die beiden zugehörigen Toleranzwerte können wahlweise eingegeben werden. Auch die Eingaben der Oberflächenrauigkeiten zu den entstehenden Flächen ist nicht zwangsweise notwendig. Der Konstrukteur besitzt aber zu diesem Zeitpunkt die Möglichkeit, die entsprechenden Informationen zu sichern.

6.2.1.6 Die Formelemente

Ein Vertreter der Gruppe der Formelemente des Fertigungsverfahrens Drehen ist der Sicherungseinstich. Das zugehörige Datenblatt ist in Bild 6.20 dargestellt. Die Spankörper werden durch Rotieren einer 2d-Kontur um die Rotationsachse des Werkstückes erzeugt. Zur Konturerzeugung stehen die im Kapitel Profildrehen vorgestellten Verfahren zur Verfügung.

Hauptverfahren : Drehen Formelement : Sicherungseinstich	
Parameter :	A: Bezugsfläche t : Tiefe $t > 0$ b : Einstichbreite $b > 0$ rr : Rundungsradius $rr > 0$ a : Abstand zu A $a > 0$
Toleranzangaben :	t,b,a
Spankörper/Oberfläche :	Abhängig von der Profilform
Oberflächengüte :	1. Rz 160 - Rz 4

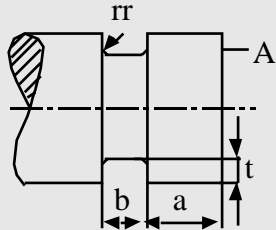
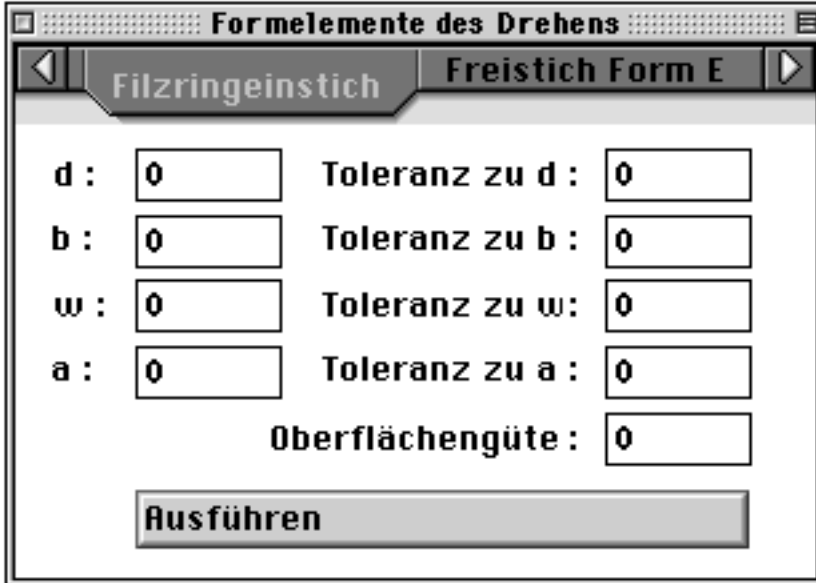


Bild 6.20 Parametertabelle Drehen/Formelemente /64/

Als Beispiel eines zugrunde liegenden Parametersatzes zeigt Bild 6.21 den vollständigen Parametersatz des Fertigungsverfahrens Filzringeinstich. Um ein Fertigungsverfahren der Klasse Formelemente durchzuführen müssen alle geometrischen Parameter eingegeben werden.



Formelemente des Drehens

Filzringeinstich Freistich Form E

d : Toleranz zu d :

b : Toleranz zu b :

w : Toleranz zu w :

a : Toleranz zu a :

Oberflächengüte :

Ausführen

Bild 6.21 Das Menü Filzringeinstich

Entsprechende Parameterprogramme für die in Bild 6.22 dargestellten Konturen wurden implementiert. Für den Aufbau der Parameterprogramme wurde eine schematische Vorgehensweise definiert. Diese ergab sich zum einem aus den Gegebenheiten der Programmiersprache C++ sowie zum anderen aus den Anforderungen des CV-DORS Interfaces.

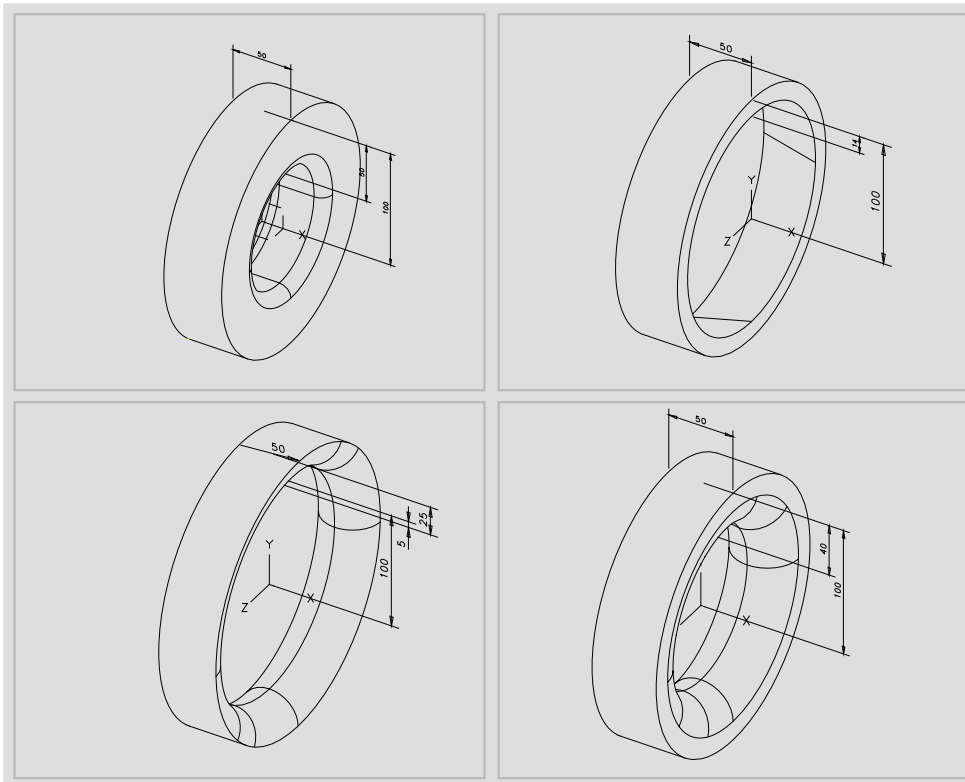


Bild 6.22 Parametrisierte Konturen /125/

Ein weiterer Vorteil ist die schnellere Erstellung von Programmen sowie eine bessere Verifikation dieser. Der Programmkopf besteht aus der Definition und Kommentierung der benötigten Variablen. Im Anschluß hieran werden die globalen Variablen `Frame`, `Layer` und `Farben` gesichert. Diese Werte werden jetzt den eigenen Bedürfnissen angepaßt. Die Erzeugung der benötigten Konturelemente wird mit Hilfe der Parameter durchgeführt. In einem ersten Schritt müssen die einzelnen Konturelemente erzeugt werden.

Dies geschieht mit Funktionen wie `t_arc::t_arc` oder `t_ellipse::t_ellipse`. Mit der Funktion `t_sweep_profile` wird aus diesen geometrischen Primitiven ein Konturelement erzeugt. Die geometrischen Primitiven werden in eine Liste eingefügt. Diese wird als Parameter an die Funktion `t_sweep_profile::t_sweep_profile` übergeben, die das gewünschte Konturelement erzeugt. Es ist darauf zu achten, das die Konturelemente eine geschlossene Kontur bilden.

```

// ----- INCLUDE DATEIEN -----
#include <stdio.h>
#include <math.h>
#include <cadds/insert/cds_mdl/entity.hxx>
// ----- EXTERNE FUNKTIONEN -----
// ----- EXTERNE VARIABLE -----
// ----- DEFINES -----
#define 2PI PI*2
// ----- VORWAERTSDECLARATIONEN DER VERWENDETEN FUNKTIONEN -----
START_EXTERN_C
int Pattern_();
END_EXTERN_C
// ----- MODUL VARIABLE -----
int Pattern_()
{
//   Initialisierung der Variablen
  int error = 0; ...
//   Setzen der Konstruktionsebene
//   Sichern des aktuellen Layers
//   Veraenderliche Variablen
  double Koerper_Anfangs_Punkt_X = 50.0; ...
//   Erzeugen der Raumpunkte und Winkel fuer den Koerper
  t_triple Koerper_Anfangs_Punkt
    (KoerperAnfangsPunktX,KoerperAnfangsPunktY,KoerperAnfangsPunktZ); ...
//   Erzeugen der Kontur durch einen Vollkreis
  t_arc *Kreis1 = new t_arc(Kreis_Mittel_Punkt,Kreis_Radius,a1,a2);
//   Uebergeben des Segments in eine Liste
  t_curve_list *Kreis_Liste = new t_curve_list;
  Kreis_Liste->add(Kreis1);
//   Extrudieren der Kontur mit sweep Befehl
  t_sweep_profile *Kreissweepen = new t_sweep_profile(*Kreis_Liste,error_code);
  if(!Kreissweepen) {printf ("t_sweep Fehler new \n");return(1);}
//   interne Darstellung der Kontur
  t_brep *Rot_Kreissweepen =
    new t_brep(*Kreissweepen,KoerperAnfangsPunkt,KoerperEndPunkt,Winkel,endcap,error );
  if(!Rot_Kreissweepen) {
    printf ("t_breb Fehler new \n");
//   Rucksetzen der Konstruktionsebene
//   Rucksetzen des Layers
    return(1);
  }
  Rot_Kreissweepen->write();Rot_Kreissweepen->display();
//   Rucksetzen der Konstruktionsebene
//   Rucksetzen des Layers
  return( 0 );
}

```

Bild 6.23 Der prinzipielle Aufbau der Parameterprogramme

Ferner muß ein Umlaufsinn eingehalten werden. Die Konturelemente müssen im Uhrzeiger- oder im Gegenuhrzeigersinn angeordnet werden. Mit dem durch die Funktion *t_sweep_profile* erzeugten Profil wird der gewünschte Körper durch den Konstruktor *t_brep::t_brep* erzeugt. Die globalen Parameter erhalten ihre Ursprünglichen Werte zurück. Der erzeugte Körper wird an das aufrufende Programm als Körper vom Typ **t_brep* zurückgeliefert. Das vorangegangene [Bild 6.23](#) zeigt den schematischen Programmaufbau. Neben den beschriebenen Sicherungseinstich wurden die Formelementverfahren des Sprengringeinstich, des Filzringeinstich, des Freistichs Form E und F, der Fase, der Verrundung, des Gewindes mit Gewindeauslauf, des Gewinde mit Gewindefreistich sowie des Kegels parametrisiert und implementiert.

6.2.2 Das Fertigungsverfahren Fräsen

Das Verfahren Fräsen besitzt die Unterverfahren Planfräsen, Rundfräsen, Profilfräsen, Formfräsen und die Gruppe der Formelemente. Das Profilfräsen wird in die Verfahrenselemente Längsprofilfräsen und Rundprofilfräsen aufgeteilt. Zu den Formelementen gehört das Fräsen von Nuten, T-Nuten, Radian, Ecken, Fasen, n-Außenkant, Paßfeder, Scheibefeder, Frästasche und Fräsinsel.

Planfräsen ist als Fräsen mit geradliniger Vorschubbewegung zur Erzeugung ebener Flächen definiert. Der Spankörper wird durch die zu bearbeitende Fläche sowie die Spantiefe gebildet. Die programmtechnische Implementierung erfolgt analog. Durch identifizieren der Bezugsfläche sowie das Eingeben der Spantiefe, der Toleranzen und der Oberflächengüte wird das Verfahren durchgeführt. Dabei wird der Spankörper durch verschieben der Bezugsfläche entlang eines Vektors erzeugt. In einem zweiten Schritt wird er vom Grundkörper subtrahiert.

Fräsen mit kreisförmiger Vorschubbewegung zur Erzeugung kreiszylindrischer Flächen wird als Rundfräsen bezeichnet. Der Spankörper entsteht durch Verschieben eines Kreisring entlang eines Vektors. Programmtechnisch wird der Spankörper analog zu dieser Vorgehensweise erzeugt. Als Parameter gehen die Bezugsfläche, der zu erzeugende Durchmesser, die Fräsbreite, der Abstand zur Bezugsfläche sowie die zu diesen Angaben gehörigen Toleranzwerte und die Oberflächengüte ein.

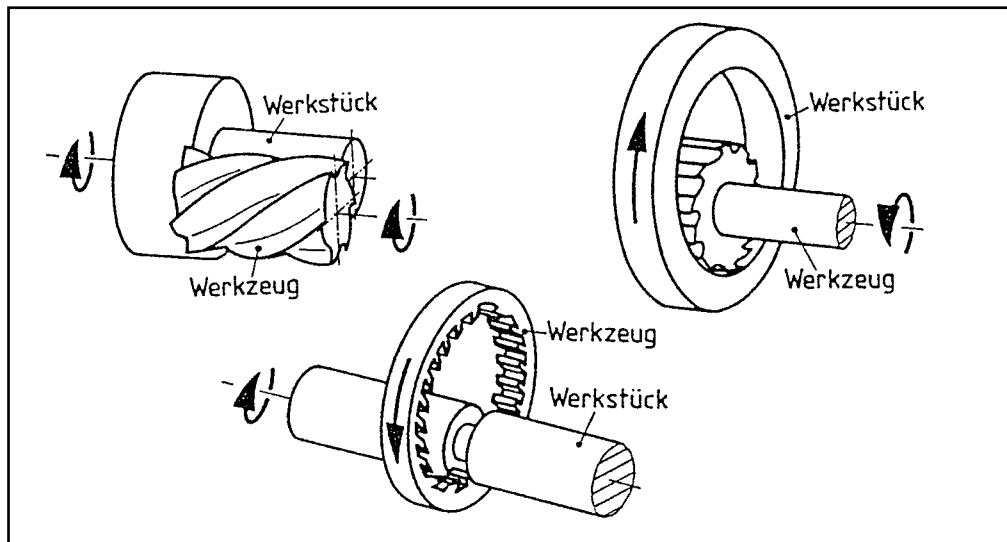


Bild 6.24 Werkzeug-Werkstück Kombinationen beim Rundfräsen /64/

Im vorangegangenen Bild sind die Möglichkeiten der Bewegungen zwischen Werkstück und Werkzeug dargestellt. Einen Einfluß auf die Erstellung der Spankörper besteht nicht.

Ganz allgemein betrachtet ist Profilfräsen eine Art des Fräsens, bei der sich das Profil des Fräasers auf das Werkstück abbildet. Beim Längsprofilfräsen wird eine geradlinige Vorschubbewegung zur Erzeugung gerader Profilflächen durchgeführt, beim Rundprofilfräsen erfolgt eine kreisförmige Vorschubbewegung zur Erzeugung von rotationssymmetrischen Profilflächen. Die Erzeugung des Spankörpers beim Längsprofilfräsen erfolgt durch Ableiten der 2d-Kontur des Profilfräasers. In einem ersten Schritt wird ein Körper durch Verschieben dieser Kontur entlang eines Vektors erzeugt. Am Endpunkt des Verschiebevektors wird eine in der Konturebene liegende Rotationsachse erzeugt. Um diese wird die 2d-Kontur rotiert, wodurch ein Körper entsteht. Dieser wird zu dem vorher Erzeugtem addiert. Der entstandene Körper bildet den endgültigen Spankörper. Beim Rundprofilfräsen wird die zu erzeugende 2d-Kontur um die Rotationsachse des Werkstückes rotiert. Der so generierte Körper ist der Spankörper.

Das Formfräsen kann in verschiedene Unterverfahren unterteilt werden. Beim Freiformfräsen erfolgt die Vorschubbewegung von Hand. Dieses Verfahren wird oft beim Gravierfräsen benutzt. Da seine technisch-wirtschaftliche Bedeutung gering ist, wurde es nicht implementiert. Bedeutungsvoller ist das Nachformfräsen. Es wird beim Gravieren mit Schablonen, dem Gesenkfräsen sowie dem Fräsen von Kurvenscheiben verwendet. Beim kinematischen Form-

fräsen erfolgt die Vorschubbewegung durch ein mechanisches Getriebe. Beispiele hierfür sind das Oval-Formfräsen, das Trochoiden-Formfräsen, das Polygonprofil-Fräsen und das Hinterfräsen von Werkzeugschneiden bzw. -zähnen. Als letztes Verfahren sei das NC-Formfräsen erwähnt. Bei diesem Verfahren wird die Vorschubbewegung durch eingegebene Daten gesteuert. Ein Vertreter der Formelemente des Fräsens ist die T-Nut. Seine 2d-Kontur mit den Parametern zur Bestimmung der Geometrie ist in Bild 6.25 dargestellt.

Hauptverfahren : Fräsen Formelement : T-Nut	
Parameter :	A : Bezugsfläche b1: Nutenbreite b2: Schaftbreite t1 : Nuttiefe t2: Schafttiefe l : Nutlänge a : Abstand zu A
Toleranzangaben :	d, a
Spankörper :	abhängig von Nutform
erzeugte Oberfläche :	I,II
Oberflächengüte I,II :	$R_z 160 - R_z 1,6$

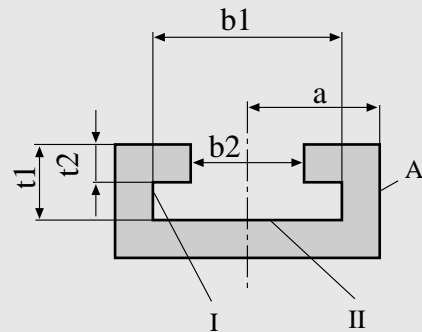


Bild 6.25 Parametrisierte T-Nut /64/

Neben den geometrischen Parametern sind die wahlweise einzugebenden technologischen Daten dargestellt. Der Spankörper wird durch verschieben der 2d-Kontur entlang eines Vektors erzeugt. Endet die T-Nut im Körper, so wird ein weiterer Körper durch rotieren der Kontur um eine im Endpunkt des Verschiebevektors stehende Rotationsachse erzeugt. Dieser wird zum ersten Spankörper addiert. Durch Subtraktion dieses Gesamtkörpers wird das Fertigungsverfahren ausgeführt.

Analog der Erzeugung von T-Nuten können Nuten, Ecken und Fasen generiert werden. Beim Formelement des Radius wird in einem erstem Schritt ein Quader erzeugt. An diesen wird an der Unterseite tangential ein Zylinder erzeugt. Beide Körper werden addiert. Sie bilden den Spankörper.

Das Formelement n-Außenkant dient vornehmlich zum Anbringen von Schlüsselflächen an Halbzeugen. Die Anzahl n kann einen Wert von 2 bis 8 annehmen. Als Spankörper werden n Quader erzeugt, die im entsprechenden Winkel und Abstand um die Rotationsachse des Werkstücks positioniert werden. Durch die Schnittmengenberechnung aller Quader mit dem Werkstück wird das Spanvolumen berechnet. Durch Subtraktion der Quader vom Werkstück wird die neue Geometrie erzeugt.

Um eine Paßfedernut zu Erzeugen wird zuerst ein Quader erzeugt. An beiden Enden werden Zylinder mit dem Durchmesser der Breite des Quaders generiert. Die Zylinderachsen liegen in den Ebenen der Quaderenden. Alle drei Körper werden addiert. Die Schnittmenge dieses Körpers mit dem Werkstück ergibt das Spanvolumen. Die gewünschte Geometrie wird durch subtrahieren dieses Körpers vom Werkstück generiert.

Die Erzeugung einer Scheibefedernut wird durchgeführt, indem zunächst ein Zylinder erzeugt wird, dessen Höhe der Breite der Scheibefedernut entspricht. Die Schnittmenge dieses Zylinders mit dem Werkstück entspricht dem Spankörper. Die Werkstückgeometrie nach Anwendung des Fertigungsverfahrens erhält man, indem man den Zylinder vom Werkstück subtrahiert.

Frästasche und Fräsinsel sind Formelemente, die durch Formfräsen erzeugt werden. Für sie gelten die selben Aspekte, wie bereits unter dem Unterverfahren Formfräsen beschrieben.

6.2.3 Das Fertigungsverfahren Bohren

Unter dem Hauptverfahren des Bohrens sind die Verfahren des eigentlichen Bohrens, des Senkens und des Reibens zusammengefasst. Beim Bohren verläuft die Vorschubbewegung in Richtung der Drehachse des Werkzeuges. Die Drehachse und die Achse der Erzeugten Innenflächen sind identisch. Beim Senken wird senkrecht zur Drehachse liegend eine Planfläche oder eine Kegelfläche erzeugt. Das Reiben dient zur Erhöhung der Oberflächengüte. Es ist ein Aufbohren mit geringer Spanungsdicke.

Das Bohren wird in die Verfahren des Plansenkens mit den Verfahrenselementen des Planansenkens und des Planeinsenkens, des Rundbohrens, des Schraubenbohrens, des Profilbohrens und die Formelemente des Bohrens unterteilt. Das Rundbohren besteht aus den Verfahrenselementen des Bohrens ins Volle, des Kernbohrens, des Aufbohrens und des Rundreibens. Das

Unterverfahren Schraubenbohren besteht lediglich aus dem Verfahrenselement des Gewindebohrers. Das Profilbohren wird in die Verfahrenselemente des Profilbohrers ins Volle, des Profilaufbohrers, des Profilsenkens und des Profilreibens aufgliedert. Unter der Gruppe der Formelemente sind die Verfahren der Durchgangs- und Sackbohrung, der Senkbohrung, der Stufensenkung, Stufenbohrung und Gewindebohrung zusammengefaßt. Für alle genannten Unterverfahren und Formelemente wurden Parametertabellen erstellt. Im nachfolgendem Bild ist die Parametertabelle des Formelementes Stufenbohrung dargestellt.

Hauptverfahren : Bohren, Senken, Reiben	
Formelement : Stufenbohrung	
Parameter :	A: Bezugsfläche d: Bohrungsdurchmesser $ds > d > 0$ ds: Senkungsdurchmesser $ds > 0$ t: Tiefe $t > 0$ ts: Senkungstiefe $t > ts > 0$ ws: Senkungswinkel $180^\circ \geq ws \geq 0^\circ$ bdf: Bodenform $bdf = 120^\circ, 180^\circ$ x,y: Position (x,y,z): Richtungsvektor
Toleranzangaben :	t,ts,d,ds,ws,x,y,(x,y,z)
Spankörper :	abhängig von Bohrungsform
erzeugte Oberfläche :	I, II
Oberflächengüte I :	$R_z 250 - R_z 16$
II :	$R_z 40 - R_z 4$

Bild 6.26 Parametertabelle Stufenbohrung /64/

Alle Parameterprogramme zur Erzeugung der Spankörper erzeugen in ihrem ersten Schritt die 2d-Kontur des Werkzeuges. Diese wurde durch eine Rotation um die Drehachse in einen Volumenkörper überführt.

6.2.4 Die Erzeugung von Spankörpern

Zur Implementierung der Fertigungsverfahren wurden Spankörper erzeugt, die von einer vorhandenen Geometrie, z.B. einem Halbzeug, subtrahiert werden. Die Spankörper dienen zur Ermittlung des Spanvolumens sowie zur Generierung des Werkstückes nach der Durchführung eines Fertigungsverfahrens.

Das wesentliche Ziel ihrer Verwendung besteht darin, einen sicheren Modellaufbau zu gestatten. Geometrische Modellierer reagieren auf die unterschiedlichen Möglichkeiten der Modell-erzeugung recht verschieden. Da der Modellaufbau weitgehend automatisch erfolgt, wurden einige Varianten der Generierung von Spankörpern getestet. Prinzipiell lassen sich die Spankörper auf zwei verschiedene Arten generieren. Eine besteht darin, eine ebene Kontur zu erzeugen und diese durch einem Vektor zu verschieben, oder um eine Achse zu rotieren.

Die zweite Methode nutzt die geometrischen Primitivkörper. Aus ihnen werden mit den Mengenoperationen komplexere Körper erzeugt. Am wenigsten fehleranfällig gestaltete sich die Generierung der Spankörper wenn aus 2d-Konturen durch „Sweepen“ oder „Rotieren“ Volumen erzeugt wurden /79, 80, 81, 85, 106, 109 und 113/.

Ein weiteres Ziel der Generierung von Spankörpern besteht darin, das Spanvolumen zu berechnen. Dadurch könnten Verfahren zur Ermittlung der Herstellkosten ermittelt werden. Die Kosten für ein Bauteil könnten so näherungsweise ermittelt werden. Der Bedarf an Fertigungskapazitäten kann abgeschätzt werden. .

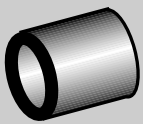
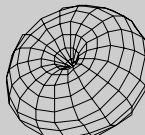
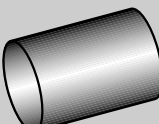
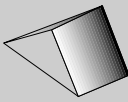
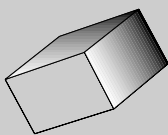
Hülse	Torus	Zylinder	Fase	Box
				
Verwendung				
Querplandrehen Längsplandrehen Längsrunddrehen Sicherungseinstich Kernbohren Aufbohren Rundfräsen	Querprofil- einstechdrehen Sicherungseinstich Sprengringseinstich	Querabstechdrehen Querplandrehen Planeinsinken Bohren ins Volle Scheibfedernut	Kegeldrehen	Planfräsen Nutenfräsen Ecken

Bild 6.27 Spankörper und ihre Verwendung

Mit den aus dem vorangegangenen Abschnitt bearbeiteten Parametertabellen wurden die benötigten Spankörper erstellt. Bild 6.27 gibt einen Überblick über geometrische Körper und ihre Verwendung als Spankörper.

6.2.5 Der Aufbau der Programme zur Erzeugung von Spankörpern

Die Spankörperprogramme beruhen auf einer Parametrisierung der zur fertigungsorientierten Modellierung notwendigen Spankörper. Von einer zu identifizierenden oder zu definierenden Bezugsfläche ausgehend wird ein lokales Koordinatensystem erzeugt. In diesem werden zunächst die Konturelemente des Spankörpers erzeugt. Sie werden durch den CV-DORS Befehl `t_sweep_profile` in eine Kontur umgewandelt. Es ist dabei notwendig, daß die Kontur stetig ist. Weiterhin muß die Kontur im Uhrzeigersinn oder im Gegenuhrzeigersinn ausgerichtet sein. Aus der Kontur wird der Körper durch Rotation oder Verschiebung erzeugt. Von dem Körper und dem zu bearbeitendem Werkstück wird die Schnittmenge gebildet. Der entstehende Körper ist der reale Schnittkörper. Er stellt das Spanvolumen dar und wird in der Datenbank verwaltet. Im Anschluß wird der ursprüngliche Spankörper von dem Werkstück subtrahiert. Das Werkstück erhält damit die durch den Fertigungsschritt erzeugte Form. Beim Querplandrehen z.B. wird ein Zylinder als Spankörper benötigt. Dieser wird direkt über den CV-DORS Befehl :

```
t_brep *Spankörper = new t_brep(CYLINDER, radius, cline_init_pnt, cline_fine_pnt, error);
```

generiert. Sein Referenzzeiger wird in die Klasse Fertigung eingetragen. Der Fertigungsbaum enthält somit alle Spanvolumen. Die Ermittlung des gesamten Spanvolumen erfolgt durch das Durchlaufen des Baumes.

6.2.6 Werkzeugmaschinen Daten

Um die Anwendung der Fertigungsverfahren zu kontrollieren wurden die dazu notwendigen Elemente von Werkzeugmaschinen als CAD-Modell implementiert. Für verschiedene Bearbeitungsmaschinen wurden die Daten des Arbeitsraumes ermittelt sowie zugehörige Objektstrukturen definiert, in der diese Daten abgelegt werden.

Neben dem Arbeitsraum sind die Geometriedaten der Spannlagelage sowie des Spannmittels erstellt worden. Diese dienen in der gegenwärtigen Implementationsstufe zur Kollisionskontrolle /68/. Verfahren zur Berechnung von Spankräften können noch hinzugefügt werden. Die

Geometrie der Spannmittel wurde dabei stark vereinfacht, um die Datenmenge nicht zu sehr anwachsen zu lassen.

```
class Einspannung {
    t_triple Anschluss,Ausrichtung;
    t_entity *Spannwerkzeug;
public :
    Einspannung(),
    ~Einspannung();
    void Anzeigen( int );
    t_entity *Kollision( t_entity *);
};
```

Bild 6.28 Definition der Klasse Einspannung

Die Spannmittel wurden als CADD55 Modell erzeugt. Durch Einlesen eines Modells und anschließendes Identifizieren der Geometrie wird der Datenbankzeiger generiert und mit dem fertigungsorientierten Modellierer verknüpft. Es ist geplant eine entsprechende Bibliothek auf der Basis von Variantenprogrammen zu erstellen.

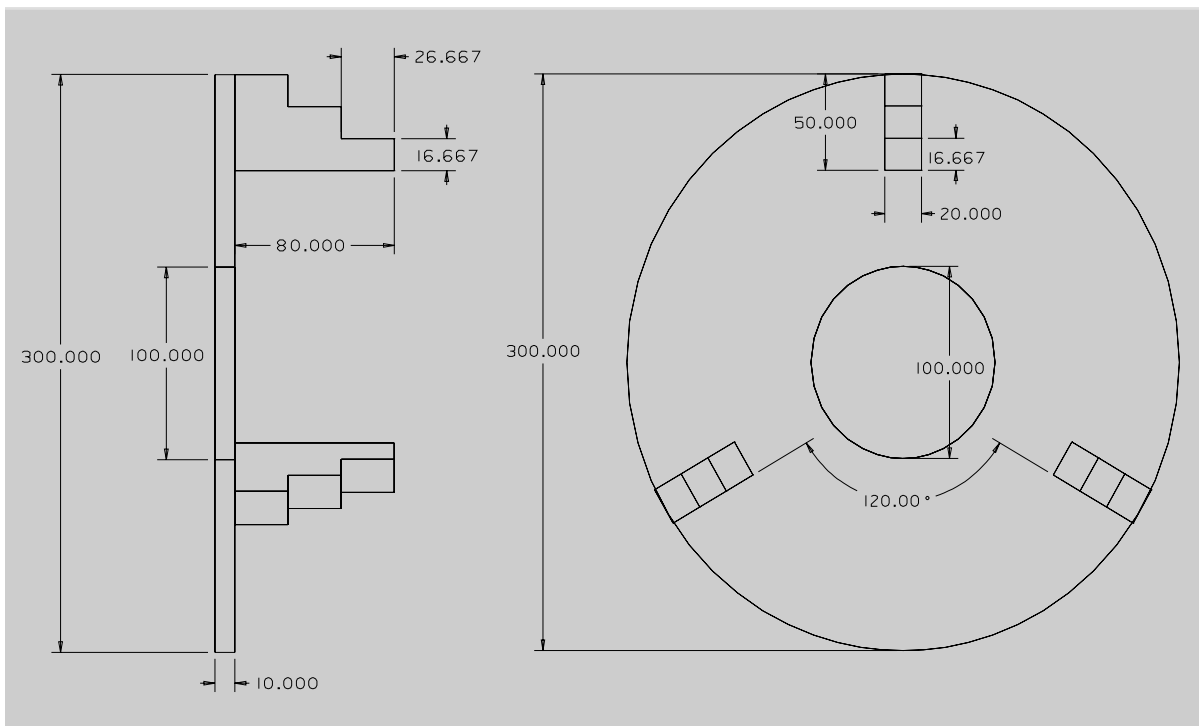


Bild 6.29 Parametrisiertes 3-Backenfutter /125/

Als Varianten implementiert wurden bisher ein Schraubstock zum Aufspannen von Werkstücken auf einer Fräsmaschine sowie das in [Bild 6.29](#) gezeigte Dreibackenfutter. Neben den Spannmitteln wurden entsprechende Klassen für die Arbeitsbereiche von Dreh-, Bohr und

Fräsmaschinen definiert. Die Arbeitsbereiche für Dreh- und Fräsmaschinen wurden durch die nachfolgend aufgeführten Klassen definiert:

```
class Arbeitsbereich {
public :
    Arbeitsbereich();
    ~Arbeitsbereich();
    virtual int inside( t_entity *);
}
class DrehmaschinenArbeitsbereich : Arbeitsbereich {
    double Spitzenhoehe, Drehdurchmesser, Drehlange;
public :
    DrehmaschinenArbeitsbereich( double, double,double);
    ~DrehmaschinenArbeitsbereich();
    int inside( t_entity *);
}
class FraesmaschinenArbeitsbereich : Arbeitsbereich {
    double BereichInX,BereichInY,BereichInZ;
public :
    FraesmaschinenArbeitsbereich( double, double ,double);
    ~FraesmaschinenArbeitsbereich();
    int inside( t_entity *);
}
```

Bild 6.30 Definition der Klasse Arbeitsbereich

Die Klasse „Arbeitsbereich“ wird von der Klasse „Werkzeugmaschine“ genutzt. Zusätzlich besitzt „Werkzeugmaschine“ noch die Klasse „Einspannung“. Durch verschieden konkrete Objekte der beiden Klassen lassen sich die verschiedensten Werkzeugmaschinen erzeugen. Die derzeitige Implementation der Klasse Werkzeugmaschine gestaltet sich wie folgt :

```
class Werkzeugmaschine {
    char * mName;
    t_entity * mGrafModel;
    Einspannung * WZMEinspannung;
    Arbeitsbereich * WZMArbeitsbereich;
public :
    Werkzeugmaschine( int Typ, t_entity* pModel);
    ~ Werkzeugmaschine();
    char * GetWZMName();
    int PositionBearbeitbar();
    int Anzeigen( int );
}
```

Bild 6.31 Die Definition der Basisklasse Werkzeugmaschine

Mit der Funktion „Anzeigen()“ kann die grafische Darstellung einer Werkzeugmaschine ein- bzw. ausgeschaltet werden. Die Funktion „PositionBearbeiten()“ ermittelt, ob ein Fertigungsverfahren Kollisionsfrei durchgeführt werden kann.

6.3 Die Benutzeroberfläche

Ergonomischen Richtlinien folgend, wurde die Benutzeroberfläche aufgebaut. Sie besteht im wesentlichen aus drei Bereichen, einem Fenster für die grafische Darstellung, einer Menüleiste und einer Anzahl von Untermenüs. Das grafische Fenster beinhaltet den eigentlichen Bereich der grafischen Darstellung sowie eine Anzahl von Bedienelementen zur Steuerung der Darstellung. Die Menüleiste dient zur prinzipiellen Programmsteuerung. Durch sie werden Untermenüs ein- und ausgeblendet sowie elementare Funktionen wie das Öffnen und Schliessen von Dateien ausgeführt. Die Untermenüs sind auf schwebenden Palettenfenstern positioniert und beinhalten Eigenschaftsfelder zum Auswählen von Unterauswahlpunkten und zum Abfragen von detaillierten Informationen. Diese Struktur ergibt einen flachen und damit sehr übersichtlichen Menübaum.

6.3.1 Das Fermod-Hauptfenster

Das Hauptfenster des fertigungsorientierten Modellierers gliedert sich in zwei Bereiche. Im oberen, größerem Bereich befindet sich das grafische Fenster, das die visualisierte Darstellung des rechnerinternen Modells enthält. Im kleinerem unterem Bereich sind Steuerelemente zur Abbildung des Modells im Fenster enthalten.

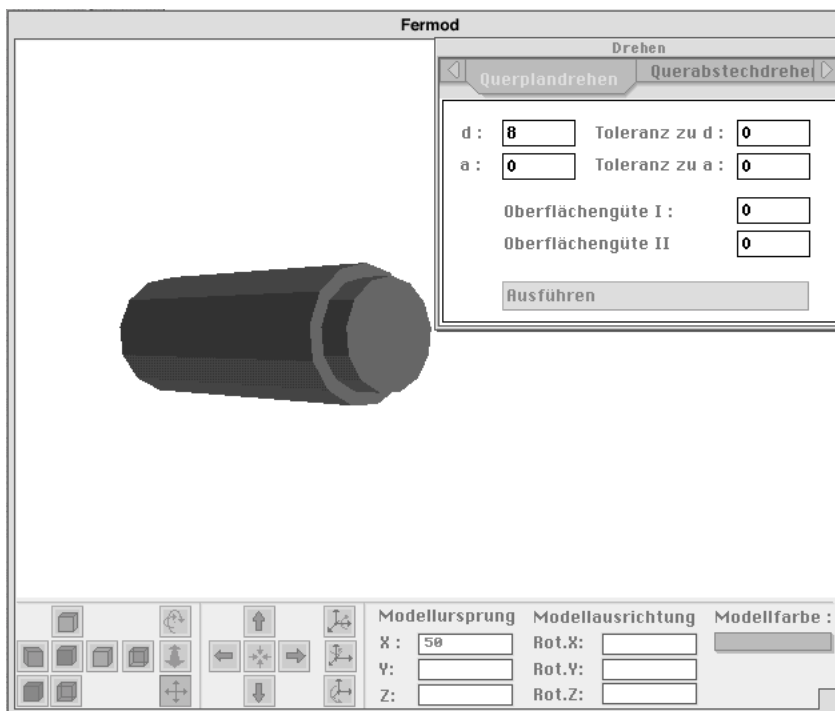


Bild 6.32 Fermod-Hauptfenster

des „realen“ Modells entsteht auf der „Viewplane“ und wird auf dem Bildschirm dargestellt. Die beiden Ebenen „Hither Plane“ und „Yon Plane“ dienen zum begrenzten des Modellraumes. Alle Elemente die sich, vom Standpunkt der Kamera aus betrachtet, vor der „Hither Plane“ oder hinter der „Yon Plane“ befinden werden bei der Darstellung nicht berücksichtigt.

Neben der dargestellten Perspektivprojektion verwenden Softwareprodukte aus dem technischen Bereich sehr oft die Parallelprojektion. Diese bildet Modelle zwar nicht so realistisch auf dem Bildschirm ab wie die Perspektivprojektionen, erhält dafür aber die Parallelität von Kanten. Entfernungen werden durch die Parallelprojektion im Gegensatz zur Perspektivprojektion nicht verzerrt dargestellt. Besonders häufig findet die isometrische Projektion Verwendung. Sie legt zunächst die beiden Ursprünge des Modell- und des Bildkoordinatensystem übereinander. Dann werden die X- und die Y-Achse in einen Winkel von 30 Grad zur Bildebene gedreht. Alle Längen werden im Verhältnis 1:1:1 abgebildet /47/.

Beim Zoomen wird ein Würfel im Modellraum berechnet, der dem sichtbaren Bereich des Modells entspricht. Ein Vektor von der Kameraposition zum Würfelschwerpunkt wird bestimmt. Ein zweiter Vektor von der Kameraposition zur rechten oder linken Kante des Würfels in der Ebene des zuerst berechneten Vektors wird gebildet. Dieser Wert entspricht dem Öffnungswinkel der Kamera. Zu diesen wird ein Wert addiert, der sich aus der Mausbewegung sowie der Entfernung der Kamera vom Würfelschwerpunkt ergibt.

Beim Rotieren wird wiederum ein Würfel im Modellraum berechnet der dem sichtbaren Bereich des Modells entspricht. In den Schwerpunkt dieses Würfels wird eine Kugel gelegt. Deren Radius entspricht dem Abstand des Würfelschwerpunktes von der Kameraposition. Durch bewegen der Maus verändert der Benutzer die Kameraposition auf dieser Kugel um den Würfelschwerpunkt. Dabei bleibt die Blickrichtung der Kamera erhalten. Die Bewegungen der Maus werden in Abhängigkeit von der Entfernung der Kamera zum Modell in entsprechende Winkelwerte umgerechnet. Hierdurch wird die „Bewegungsempfindlichkeit“ der Kamera gesteuert.

Bei starken Vergrößerungen des Modells werden die differentiellen Mausbewegungen in kleine Winkeländerungen umgerechnet. Nachfolgend ist der Algorithmus zur Berechnung der Rotation um die X-Achse dargestellt :

```
void CMainGraphicView::RotateCameraAboutX(DocumentPtr theDocument, float dX)
{
    TQ3ViewObject      theView;
    TQ3CameraObject    theCamera;
    TQ3Vector3D        xVector,zVector,tmpVector ;
    TQ3Matrix4x4       rotationMatrix;
    TQ3ViewAngleAspectCameraData  data;
    TQ3Point3D         rotLoc;

    // Get the view and the camera objects.
    theView = theDocument->fView;
    Q3View_GetCamera(theView, &theCamera);
    Q3ViewAngleAspectCamera_GetData(theCamera,&data);
    Rect  theFrame;
    CalcLocalFrameRect( theFrame );
    dX = ( 180.0 * dX ) / (float)(theFrame.right - theFrame.left);
    dX = ( dX * 3.14 / 180.0);
    Q3Point3D_Subtract(&data.cameraData.placement.cameraLocation,
                      &data.cameraData.placement.pointOfInterest,&zVector);
    Q3Vector3D_Cross(&zVector,&data.cameraData.placement.upVector,&xVector);
    Q3Vector3D_Normalize(&xVector,&tmpVector);
    xVector = tmpVector;
    // Create the rotation matrix for rotating about the x axis.
    Q3Matrix4x4_SetRotateAboutAxis(&rotationMatrix,
                                  &data.cameraData.placement.pointOfInterest , &xVector, dX);
    Q3Point3D_Transform(&data.cameraData.placement.cameraLocation,
                      &rotationMatrix,&rotLoc);
    data.cameraData.placement.cameraLocation = rotLoc;
    Q3Point3D_Subtract(&data.cameraData.placement.pointOfInterest
                      ,&data.cameraData.placement.cameraLocation,&zVector);
    Q3Vector3D_Cross(&zVector,&xVector,&tmpVector);
    Q3Vector3D_Normalize(&tmpVector,&zVector);
    data.cameraData.placement.upVector = zVector;
    Q3ViewAngleAspectCamera_SetData(theCamera,&data);
    Q3View_SetCamera(theView,theCamera);
    Q3Object_Dispose(theCamera);
}
```

Bild 6.34 Algorithmus zum Rotieren der Kamera um die X-Achse

Um die Funktion des Verschiebens zu realisieren wird zunächst ein Würfel im Modellraum berechnet. Dieser umfasst den sichtbaren Bereich des Modells. Vom Schwerpunkt des Würfels wird ein Vektor zur Kameraposition berechnet. Rechtwinklig zu diesem Vektor wird eine Ebene erzeugt. Der Vektor stellt somit den Normalenvektor der Ebene dar. Auf dieser Ebene wird die Kamera verschoben sobald der Benutzer Mausbewegungen bei gedrückter Maustaste ausführt.

6.3.2 Die Menüleiste

Bei dem in Kapitel 5 behandelten Softwaresystem wurde bereits der Aufbau einer Hauptmenüleiste, basierend auf dem X-Windows-System mit dem „OPEN LOOK“ Widget-Set vorgestellt. Dieser wurde modifiziert. Durch die Einführung der Eigenschaftsblätter konnte die Auswahlhierarchie einfacher und übersichtlicher gestaltet werden. Als Rückwirkung auf die Menüleiste entstanden andere Menüeinträge. Geblieben sind das „Datei-“ und das „Editieren-Menü“. Die Funktionen des „Ansichten-Menü“ wurden fast vollständig durch im Hauptfenster platzierte Icons ersetzt. Zusatzfunktionen wie die Auswahl eines Schattierungsalgorithmus (Phong oder Lambert), die Darstellungsart (Draht, Flächen, Volumen), Lichteinstellungen, die Layerbelegung und die Farbdefinition wurden unter dem neuen Menüpunkt „Darstellung“ angeordnet.

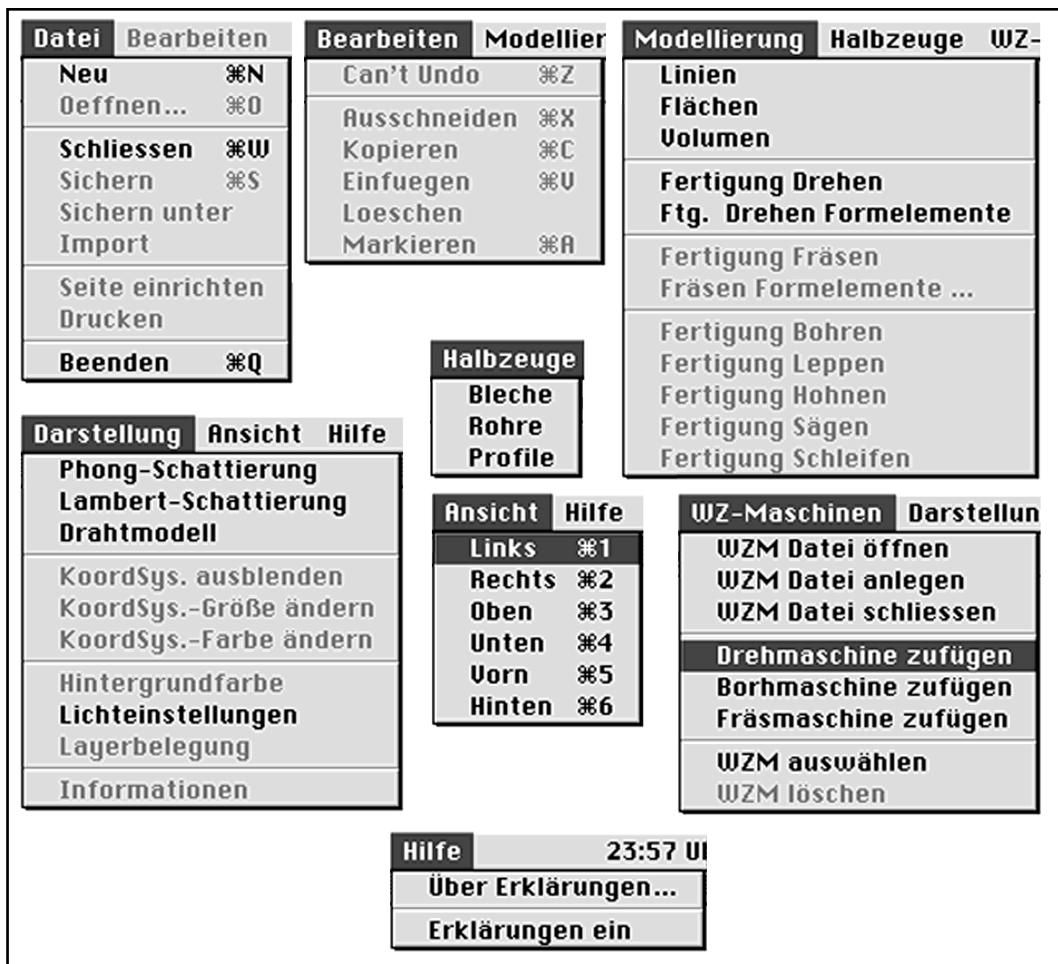


Bild 6.35 Übersicht der Menüoptionen des Hauptmenüs

Neben der Menüauswahl „Darstellung“ existiert die Menüauswahl „Ansicht“. Mit ihr können standardmäßig vorgegebene Ansichten (Ansicht von links, rechts, ...ISO) ausgewählt werden. Zusätzlich zu diesen Menüeinträgen existieren die Menüeinträge Modellierung, Halbzeuge und WZ-Maschinen.

Im Menü Modellierung können sowohl die geometrischen als auch die fertigungsorientierten Modellierverfahren /43/ ausgewählt werden. Das Menü Halbzeuge bietet die Möglichkeit, ein Halbzeug, eingeteilt in die Gruppen Bleche, Rohre und Profile, auszuwählen.

Durch das Menü WZ-Maschinen können Werkzeugmaschinen selektiert, Daten modifiziert und neue Werkzeugmaschinen zugefügt werden.

Eine Übersicht aller Menüleisteneinträge gibt das Bild 6.35. Zusätzlich zu den dargestellten Menüs existiert ein „Hilfemenü“, das von den meisten Frameworks zur Erstellung von Benutzeroberflächen automatisch angelegt wird. Neben der Versionsnummer und dem Copyright kann eine kontextsensitive Hilfe sowie eine auf einem Index basierende Hilfe aufgerufen werden.

6.3.3 Die Untermenüstruktur

Bei der Gliederung der Unterfunktionen wurde oft auf die sogenannten Eigenschaftsblätter oder auch Property-Sheets zurückgegriffen. Sie dienen dazu, eine Gruppe von Eingabefeldern, Selektionsmöglichkeiten und Einstellmöglichkeiten zusammenzufassen und übersichtlich darzustellen. Für den Anwender besitzt dies den Vorteil, das er sofort alle Eingabe- und Einstellmöglichkeiten einsehen kann. Die Eigenschaftsfenster wurden auf „schwebenden Fenstern“, die oft auch als „floating Windows“ oder „Toolwindow“ bezeichnet werden, angeordnet. Neben dem Eigenschaftsblatt befindet sich das „Auswahlmenü“ zum Auswählen eines gewünschten Eigenschaftsblattes sowie ein Infofenster auf dem Toolwindow. Mit Hilfe des Infofensters können Skizzen oder Hilfetexte ein- oder ausgeblendet werden.

Der Vorteil der dargestellten Anordnung liegt darin, das eine relativ große Anzahl von Auswahlpunkten bei der gleichzeitigen Darstellung einer großen Anzahl von Informationen angezeigt werden kann. Im Bild 6.36 ist der Aufbau eines solchen Fensters dargestellt:

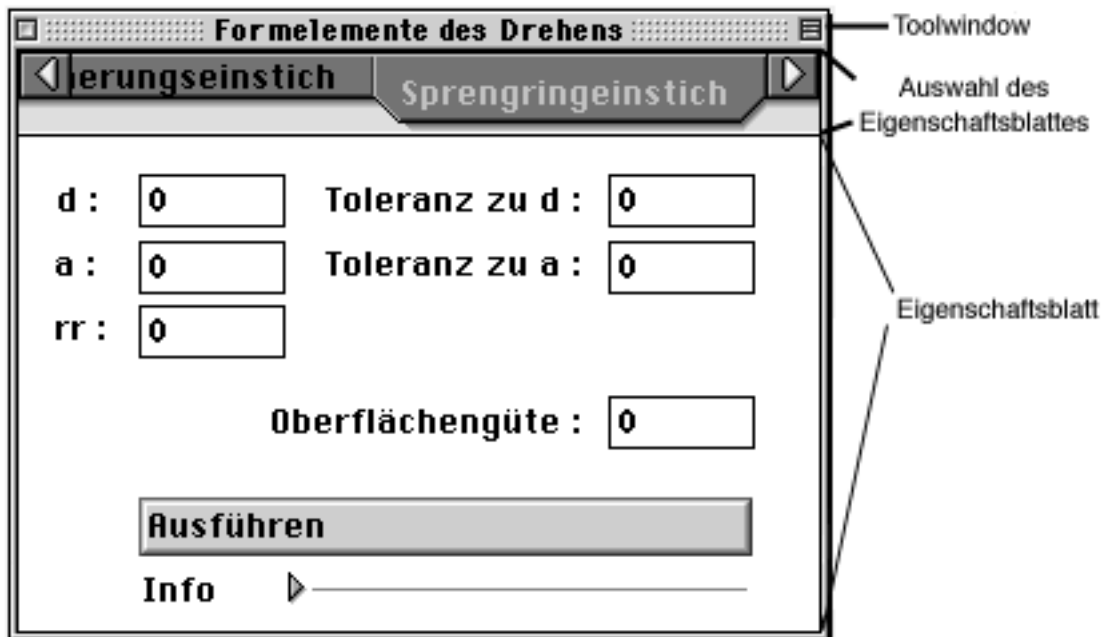


Bild 6.36 Die Elemente eines Tool-Fensters

Zu fast allen Menüeinträgen der Menüleiste existiert ein solches Toolwindow. Es bietet eine komfortable Darstellung und Auswahlmöglichkeit von Unterauswahlmöglichkeiten. Im [Bild 6.36](#) wird das Toolwindow der Formelemente des Drehens gezeigt. Eingestellt ist das Eigenschaftsfeld des „Sprengringeinstichs“.

6.4 Der Arbeitsablauf

Im Gegensatz zu geometrieorientierten Modellierungsverfahren beginnt die fertigungsorientierte Modellierung mit der Auswahl eines Halbzeuges. Hierbei werden bereits Daten gewonnen, die bei geometrischen Verfahren in nachfolgenden Arbeitsschritten ermittelt werden müssen. Wichtige Daten wie die Rohteilabmessungen und der Werkstoff des Rohteils sowie technologische Daten werden bereits erfasst. Die fertigungsorientierte Modellierung folgt der Realität und bildet das Modell durch Subtraktion der Spankörper vom Halbzeug. Dies ist analog zu dem realem Ablauf wie er an der Werkzeugmaschine durchgeführt wird. Hierbei können bereits technologische- und Kostendaten erfasst werden. Die geometrische Modellierung beginnt meist mit der Erzeugung einer Kontur, die durch Rotieren oder Verschieben zu einem Körper wird bzw. sie baut ein Modell aus Primitivkörpern auf. Hierbei können die genannten Daten nicht generiert werden.

In /125/ wird die Vorgehensweise der fertigungsorientierten Modellierung exemplarisch an zwei Beispielen dargestellt. Im ersten wird eine Welle mit mehreren Stufen, einer Zentrierbohrung, einer Nut sowie mehrerer Radien an Übergängen erzeugt. In den 21 zur Herstellung der Welle benötigten Schritte zeigt Sen welche technologischen und geometrischen Daten eingegeben werden können bzw. müssen. Im letzten Schritt werden die hieraus zusätzlich ermittelbaren Daten wie die Fertigungskosten und die Maschinenbelegungszeiten berechnet. Im zweiten Beispiel wird die fertigungsorientierte Modellierung analog des ersten Beispiels im Bereich der Fräsbearbeitung dargestellt.

6.5 Die Erweiterbarkeit

Der effiziente Einsatz eines CAD-Systems ist im wesentlichen davon abhängig, wie schnell sich individuelle Konstruktionsprozesse mit ihm durchführen lassen. Durch die verschiedensten Maßnahmen lassen sich CAD-Systeme an den Konstruktionsprozeß anpassen. Bisher vorgestellt wurden die Methoden der „Anpassung der Benutzeroberfläche“, der „Makroprogrammierung“, der „Variantenprogrammierung“, der „Anwendungsprogrammierung“ sowie der „Features“. Auch ein sich auf einem hohen Abstraktionsgrad befindliches System wie ein fertigungsorientierter Modellierer muß Methoden oder Werkzeuge bieten, um ihn an die verschiedensten betrieblichen Konstruktionsprozesse anzupassen, bzw. Automatisierungen vorzunehmen. Zur Automatisierung bieten sich zwei Möglichkeiten an:

Die erste besteht in der Nutzung der vorhandenen Klassenbibliothek. Die darin enthaltenen Klassen dienen als Basis für den Aufbau einer eigenen Klassenbibliothek bzw. die Klassen werden in eigenen Programmen genutzt. Dieser Gedanke basiert auf der Anwendungsprogrammierung. Er setzt auf einer höheren, komplexeren Ebene auf als die derzeitigen Anwendungsschnittstellen. Der Anwendungsprogrammierer wird in die Lage versetzt, mit relativ geringem Aufwand Modelle mit hohem Dateninhalt zu erzeugen.

Die zweite Möglichkeit der Erweiterung besteht darin, eine Makroprogrammiersprache zu entwickeln, die die Erstellung einer Ablaufsteuerung gestattet. Bei dieser Methode protokolliert der Anwender zunächst seine Arbeitsschritte. Diese werden einer Syntax folgend in einer Datei abgelegt. Mit einem Editor ist der Anwender in der Lage, diese Dateien zu modifizieren. So sind auf eine einfache Art und Weise Varianten eines Bauteils zu generieren.

6.6 Datenschnittstellen

Bei der Nutzung der fertigungsorientierten Modellierung werden wesentlich mehr Daten erzeugt als bei der Nutzung der geometrischen Modellierung. Die zum internen und externen Datenaustausch genutzten Schnittstellen müssen diesen gestiegenen Anforderungen genügen. Zwei wesentliche Forderungen sind dabei zu erfüllen:

- Alle erzeugten Daten für die verschiedensten Programmpakete nutzbar zu machen, die zu einer weiteren Bearbeitung des erzeugten Modells notwendig sind. Datenverluste sollten nach Möglichkeit nicht auftreten. Zu den nachgeschalteten Bereichen gehören beispielsweise die FE-Analyse und die NC-Fertigung.
- Protdaten über den gesamten Produktlebenszyklus zu verwalten um die wiederholte Eingabe von Daten in den verschiedenen Produktzyklen zu vermeiden.

Die im Kapitel 4.1.7 beschriebenen Schnittstellen wie IGES, VDAFS und SET werden diesen Ansprüchen nicht gerecht. Sie sind lediglich Geometrieschnittstellen. Technologische Informationen und Materialeigenschaften können durch sie nicht beschrieben werden. Für die fertigungsorientierte Modellierung sind sie dahingehend nutzbar, eine Auswahl der Daten für externe Programme zur Verfügung zu stellen. Prädestiniert für die Datenspeicherung der fertigungsorientierten Modellierung ist STEP. Mit den anwendungsneutralen Basismodellen (Part 41 .. 49) besteht die Möglichkeit, erzeugte Daten aus den folgenden Bereichen zu sichern bzw. an andere Systeme zu übertragen :

- Stammdaten (z.B. Freigabe, Version),
- Strukturdaten (z.B. Zusammenbauten, Stücklisteninformation),
- Konfiguration und Aufträge,
- Arbeits- bzw. Methodenpläne,
- Form Features,
- technische Eigenschaften wie Masse, Trägheitsmomente, Material und Recyclingeigenschaften, Toleranzen und Oberflächeneigenschaften sowie zu Teilbereichen aus Berechnung bzw. Simulation.

Kapitel 6 Die Fermoklassen

Mit den anwendungsorientierten Basismodellen können weitere Daten, die durch die fertigungsorientierte Modellierung erzeugt wurden, gesichert werden.

International genormt sind die Anwendungsmodelle des „Draughting“ und „Kinematics“. Die Applikationen der „Finite Element Analyse“ sowie einiger weiterer befinden sich in der Entwicklung. Von der fertigungsorientierten Modellierung können die Anwendungsmodelle des „Draughting“ sowie der „Finite Element Analyse“ genutzt bzw. mit Daten versorgt werden. Im nachfolgendem Bild ist der Fortschritt der Leistungsfähigkeit des AP214 gezeigt. Es ist ein speziell auf den Produktentwicklungsprozeß der Automobilindustrie zugeschnittenes Protokoll. Viele der durch die fertigungsorientierte Modellierung erzeugten Daten können in ihm abgelegt werden.

In verschiedenen Schritten wurde die Leistungsfähigkeit erhöht. Bereits existierende und oft genutzte Standards wie VDAFS und IGES wurden integriert.

		1995	1996	1997	1998	1999
Spezifikation		Working Draft	1. Committee Draft	2. Committee Draft	Draft International Standard (DIS)	Final DIS Internat. Standard
Implementierung	Stammdaten					
	Geometrie	Erste ProSTEP Impl.	VDAFS + Solids + Presentation + Structure (AP214 CC1)	IGES + VDAFS + Solids + Tolerances (AP214 CCx)	Geometry + Product Structure + Configuration (AP214 CCx)	
	- Volumengeometrie					
	- Flächengeometrie					
	- Drahtgeometrie					
	Präsentation					
	Technische Daten					
	Meßdaten					
	Produktstruktur					
	Form Features					
	Analysedaten					
	Simulationsdaten					
	Technologie Daten					
Dokumente						
Qualitätssicherung		Spezifikation von Testkriterien			Prozessoren test	

Bild 6.37 Leistungsumfang und Normung des AP 214 /98/

In einem relativ kurzem Zeitraum hat das AP214 ein hohes Niveau erreicht. In Vergleichstests mit bereits seit längerem existierenden Schnittstellen erzeugt es meist ein kleineres Datenvolumen und erreicht eine höhere Qualität der übertragenen Daten /98/. Dies betrifft sowohl die kürzeren Nachbearbeitungszeiten und die Anzahl der fehlerfrei übertragenen Elemente als auch einen fehlerfreien Modellaufbau im Zielsystem.

7 Zusammenfassung

Für die Beschreibung eines Produktes werden primär immer noch Zeichnungen verwendet. Sie dienen als Kommunikations- und Entscheidungsgrundlage. Dabei müssen Zeichnungen interpretiert werden. Das ist allerdings ein Grund für Miss- und Fehlinterpretationen. Zeichnungen enthalten nicht alle notwendigen Informationen, die während des Lebenszyklusses eines Produktes benötigt werden. Die vollständigen Spezifikationen und Definitionen eines Produktes sind in tausenden von einzelnen Zeichnungen enthalten. Konsistenzprüfungen werden nicht durchgeführt. Eine Wiederverwendung einzelner Zeichnungen kann daher nicht erfolgen.

Die Modellierung mit CAD-Systemen setzt auf dieser Philosophie auf und unterstützt lediglich das geometrische Modellieren. Ausgereifte Funktionalität zum 2D-3D geometrischen Modellieren sowie zum Zeichnen, Visualisieren und Analysieren geometrischer Problemstellungen stehen zur Verfügung. Der Anwender ist allerdings gezwungen, auf einem sehr niedrigen geometrischen Level zu arbeiten. Er benutzt von Anfang an mit Punkten, Linien, Ebenen und Körpern Elemente, die er exakt bestimmen muß, und die rein geometrischer Natur sind. Dies ist eine sehr ineffiziente Arbeitsweise, da der Konstrukteur auf einer anderen gedanklichen Ebene modelliert.

Er muß aus seinem gedanklich komplexen Bauteil, daß zum Beispiel ein „Kegeliges Wellenende mit Außengewinde DIN 1448“ ist, elementare grafische Elemente wie Linien und Kreissegmente machen und diese anschließend wieder zu komplexeren geometrischen Elementen zusammensetzen.

Eine weitere Entwicklung der CAD-Systeme wurde durch die zunehmende Vernetzung von Computern untereinander sowie mit Werkzeugmaschinen ausgelöst. Durch den einfach zu realisierenden physikalischen Datenaus-

tausch auf den verschiedensten Hardwareplattformen entstand die Nachfrage nach kompatiblen Programmschnittstellen zwischen CAD, CAM und CAP-Systemen. So lassen sich Werkzeugmaschinen direkt über ein Netzwerk von einem CAD/CAM-System steuern. Erste Ansätze der CAD/CAM-Integration erfolgten. Auch wirtschaftliche Aspekte /56/ trugen hierzu bei. Die vorliegende Arbeit hat diesen Trend weiterentwickelt. Sie lässt die Bereiche Konstruktion und Fertigung weiter zusammenwachsen.

Im zweiten Kapitel wurden die anerkannten Konstruktionsmethodiken vorgestellt. Ihre Berücksichtigung durch CAD-Systeme wurde dargelegt. Ein Ergebnis dieses Arbeitsschrittes ist, daß es neben den von CAD-Systemen unterstützten Modellen weitere gibt. Gerade bei zunehmender Produktkomplexität und steigenden Qualitätsanforderungen nimmt der Einfluß dieser Modelle auf den Entwicklungsprozeß stark zu. So müssen deren Einflüsse in zunehmendem Maße während der Konstruktion berücksichtigt werden. Dies bedeutet, das CAD-Systeme weit über das geometrische Modellieren hinaus gehen sollten. Weitreichendere Modelle, deren Daten und Abhängigkeiten sind zu verarbeiten.

In dieser Arbeit wurde gezeigt, wie ein Fertigungsmodell mit einem geometrischem Modell verknüpft werden kann. Hierzu wurden zunächst die bekannten Konstruktionsmethodiken sowie die Fertigungstechnik auf ihre gegenseitigen Abhängigkeiten untersucht. Auf diesen Grundlagen aufbauend wurde der fertigungsorientierte Modellierer konzipiert.

Um dieses Konzept zu realisieren wurden drei verschiedene Methoden der möglichen Softwarekonstellationen ermittelt. Die Möglichkeit der Erstellung individueller Software wurde bereits im Konstruktionssystem WISKON untersucht. Sie besitzt den Vorteil, das die Software für den fertigungsorientierten Modellierer ideal gestaltet werden kann. Als Nachteil ist der hohe Ressourcenverbrauch bei der Softwareerstellung zu nennen.

Die zweite Möglichkeit besteht in der Kombination verschiedener am Markt erhältlicher Softwarepakete. Hier kann für die einzelnen Aufgaben ein ideales Softwareprodukt genutzt werden. Der Aufwand liegt in der Verknüpfung der einzelnen Pakete sowie bei der Implementation des fertigungsorientierten Modellierers.

Die dritte Möglichkeit besteht in der Einbettung des Modellierers in ein bestehendes CAD-System. Diese besitzt den geringsten Implementierungsaufwand. Die Möglichkeiten der individuellen Gestaltung der zu entwickelnden Software und deren Benutzeroberfläche ist jedoch

sehr stark eingeschränkt und durch das CAD-System vorgegeben. Die Abhängigkeiten vom CAD-System bezüglich dessen Weiterentwicklung erscheint als großer Nachteil.

Es ist das Hauptziel des fertigungsorientierten Modellierers einen im Hinblick auf die Fertigung vollständigen und problemorientierten Modellaufbau durchführen zu können. Die Einbeziehung der Fertigung bereits in diesem frühem Stadium bewirkt eine kosten- und fertigungsgerechte Konstruktion.

Der wirtschaftlichste Einsatz eines Modellierungssystems findet dann statt, wenn dieses durch verschiedene Maßnahmen an die individuellen betrieblichen Bedürfnisse angepasst wird. Durch die Zuordnung von Fertigungsinformationen zu Geometrie bietet der Modellierer die Möglichkeit, mächtige Features aufzubauen. Diese wiederum können zu Featurekatalogen zusammengefasst werden. Das vollständige Wissen eines Betriebes bezüglich der Herstellung eines Einzelteiles kann so erfasst und zugänglich gemacht werden.

Ein weiteres Ziel zukünftiger Modellersysteme besteht in der Generierung, der Austauschbarkeit und dauerhaften Sicherung der erzeugten Daten und Methoden. Derzeit besteht mit STEP und EXPRESS eine leistungsfähige und flexible Möglichkeit, diese Daten zu sichern und mit anderen Systemen auszutauschen.

Der allgemeine Trend bzw. die Zukunft der CAD-Systeme wird derzeit durch zwei wesentliche Merkmale bestimmt. Ein Merkmal ist, dass CAD-Systeme wesentlich mächtigere Modellierungsfunktionen zur Verfügung stellen müssen. Das System Pro/Engineer sei an dieser Stelle als Beispiel genannt. Gleichzeitig darf diese mächtigere Funktionalität aber nicht zu einer komplizierten und nicht mehr handhabbaren Bedienung führen. Sich derzeit im Einsatz befindliche Systeme zeigen in diesem Bereich sehr gute Ansätze.

Das andere Merkmal des zukünftigen Trends ist das weitere Zusammenwachsen von verschiedenen Bereichen wie CAD/CAM, CAE und PPS. Der Datenaustausch über Standardschnittstellen wie z.B. STEP, IGES und dem VDAFS-Formaten funktioniert sehr gut. Neuere Formate wie das ACIS SAT-Format und die Formate des Parasolid-Kernels haben mittlerweile einen gewichtigen Anteil am Datenaustausch zwischen verschiedenen Systemen. Ein weitergehender Trend hierbei ist die Integration von FEM- und Mehrkörpersimulationssystemen in CAD-Systeme. Mit diesen Systemen können bereits in frühen Konstruktionsphasen virtuelle Bauteile auf ihre physikalischen und mechanischen Eigenschaften hin untersucht werden. Der

Konstruktionsprozeß in die „Breite“ wird hierdurch unterstützt. Zur vollständigen Unterstützung des frühen Konstruktionsprozesses fehlen dann noch Methoden zur prinzipiellen Funktionsfindung. In /53/ wird dargelegt, wie Kreativitätstechniken sowie die Methoden TRIZ, QFD, FMEA, DoE und SPC diese Konstruktionsphase unterstützen.

EDM-Systeme sorgen sowohl für einen Datenabgleich als auch für eine Strukturierung der Daten aus den einzelnen genannten Bereichen. Durch die bestehende Vernetzung aller im Betrieb bestehenden EDV-Systeme werden EDM-Systeme zu einem zentralem Prozessteuerungssystem.

8 Literaturverzeichnis

- / 1 / Abeln, O.
Modelle wissensbasierter Konstruktionssysteme-
Schnittstellen versus Integration
VDI 903, VDI-Verlag, Düsseldorf, 1991
- / 2 / Abeln, O.; Meerkamm, H.; Krause, D. und Storath, E.
The Reference Model for CAD-System - on the way
to new Architecture
Workshop Design-Konstruktion 22, ICED '93, Den Haag, 1993
- / 3 / Antoni, C.H. und Sommerlatte, T.
Report Wissensmanagement
Wie deutsche Firmen ihr Wissen profitabel machen
Symp@sion publishing, 2000
- / 4 / Bär, T. und Weber, C.
Neues aus dem Bereich der Feature-Technologie
CAD-CAM Report, Nr. 10, 1996
- / 5 / Bauert, F.; Weise, E. und Salem, N.
Modellierungsmethoden für Systeme zur
rechnergestützten Gestaltung
Konstruktion 42(19990)97-107, Springer-Verlag
- / 6 / Becker, Th.; Berthold, A.
Automatisierter Entwurfsprozeß bei einem Systemlieferanten
CAD/CAM-Report Nr.6 1996
- / 7 / Beier, Ekkehard
Objektorientierte 3D-Grafik
International Thomson Publishing GmbH, Bonn, 1994
- / 8 / Beitz, W.
Bewertungsmethoden als Entscheidungshilfe zur Auswahl
von Lösungsvarianten
Konstruktion 24(1973)493-498, Springer Verlag

- / 9 / Beitz, W.; Birkhofer, H. und Pahl, G.
Konstruktionsmethodik in der Praxis
Konstruktion 44(1992)391-397, Springer-Verlag
- / 10 / Beitz, W. und Hüttner, K.-H.
Taschenbuch für den Maschinenbau/Dubbel
15. korrigierte und erweiterte Auflage
Springer-Verlag, Berlin, Heidelberg, 1983
- / 11 / Berthold, A.
Beschreibung eines auf der Programmiersprache „C++“
basierenden objektorientierten Ansatzes für die
geometrieorientierten Module GEOMOD2
und GEOMOD3 des Konstruktionssystems „WISKON“
Diplomarbeit, LbK, FB 15
Universität-GH Kassel, 1990
- / 12 / Berthold, A.; Richter, C.
Entwicklung des CAD-Programmbausteins GEOMOD3 zur
geometrischen Modellierung von 3D-Objekten
Diplomarbeit, LbK, FB 15
Universität-GH Kassel, 1988
- / 13 / Booch, G.
Objektorientierte Analyse und Design,
Addison-Wesley, Bonn, 1994
- / 14 / Christov, L; Gorzelanczyk, M.
3D-Graphik mit HOOPS
Springer-Verlag, 1994
- / 15 / Cunningham, S.; Craighill, N. K.; Fong, M. W. und Brown, J. R.
Computer Graphics using Object-oriented programming
John Wiley & Sons, New York, 1992
- / 16 / DIN 4766 T1: Herstellverfahren der Rauheit von Oberflächen.
Erreichbare gemittelte Rauhtiefe Rz
Beuth-Verlag, Berlin
- / 17 / DIN 4766 T2: Herstellverfahren der Rauheit von Oberflächen.
Erreichbare Mittenrauhwerte Ra
Beuth-Verlag, Berlin

- / 18 / DIN 69651, Teil 1-6: Werkzeugmaschinen für die Metallverarbeitung
Beuth-Verlag, Berlin 1982

- / 19 / DIN 8580: Fertigungsverfahren-Begriffe, Einteilung (Entwurf)
Beuth-Verlag, Berlin, 1985

- / 20 / DIN 8582: Fertigungsverfahren Umformen-Einordnung,
Unterteilung, Alphabetische Übersicht
Beuth-Verlag, Berlin, 1985

- / 21 / DIN 8588: Fertigungsverfahren Zerteilen-Einordnung,
Unterteilung, Begriffe
Beuth-Verlag, Berlin, 1985

- / 22 / DIN 8589 T0: Fertigungsverfahren: Spanen
T1: Fertigungsverfahren: Drehen
T2: Fertigungsverfahren: Bohren, Senken, Reiben
T3: Fertigungsverfahren: Fräsen
T6: Fertigungsverfahren: Sägen
Beuth-Verlag, Berlin, 1981-1982

- / 23 / DIN 8593: Fertigungsverfahren Fügen-Einordnung,
Unterteilung, Begriffe
Beuth-Verlag, Berlin, 1985

- / 24 / Ehrlenspiel, K.
Industrieprobleme in Entwicklung und Konstruktion sowie
Folgerungen gemäß einer Umfrage
Konstruktion 45, Springer-Verlag 1993

- / 25 / Ehrlenspiel, K.; Ambrosy, St. und Wach, J.
Integration der Entwicklung von Produkt und Montageanlage
VDI 993.1, VDI-Verlag, Düsseldorf, 1994

- / 26 / Engel, B. und Gschwind, E.
CAD-Kerne: Leistungsfähig, offen und flexibel
CAD-CAM Report, Nr. 2, 1994

- / 27 / Engeli, M. und Taiber, G.
Feature-Based Modelling - Resultat eines tiefgreifenden Wandels
VDI-Z 136, 1994, Nr. 5

Kapitel 8 Literaturverzeichnis

- / 28 / Eulenbach, D.
Konstruktionssystem WISKON
VDI Verlag Reihe 20 Nr. 32 Düsseldorf 1990
- / 29 / Farin, G.
Curves and Surfaces for Computer Aided Geometric Design
Academic Press, San Diego, 1990
- / 30 / Flanagan, D.
X Toolkit Intrinsic Reference Manual
Third Edition for X11, Release 4 and Release 5
O'Reilly & Associates, Inc., Sebastopol
- / 31 / Franke, H.-J. und Weigel, K.
Entwurfserfahrung mit CAD-Bewertung gegenwärtiger Systeme und
notwendige Entwicklungsschwerpunkte
VDI Berichte Nr.8 1990
- / 32 / Franz, L.
Geometrisches und technisches Modellieren in CAD/CAM-Systemen
Maschinenbautechnik, Berlin (38), 1989
- / 33 / Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.
Entwurfsmuster: Elemente wiederverwendbarer objektorientierter Software
Addison-Wesley (Deutschland) GmbH, Bonn, 1996
- / 34 / Gausmeier, J.; Frank, Th. und Humpert, A.
Impulse für die CAD-Technik durch integrierte CASE-Systeme
Konstruktion 46, Springer-Verlag 1994
- / 35 / Gossard, D. C.; Zuffante, R. P. und Sakurai, H.
Representing Dimensions, Tolerances and Features in MCAE Systems
IEEE Computer Graphics & Applications, March 1988
- / 36 / Grabowski, H.; Pocsai, Z. und Staub, G.
Integriertes Logik und geometriverarbeitendes CAD-System (KRONOS)
http://irats1.ira.uka.de/13V_HTML/FORSCHUNGSVORHABEN1995/1247
- / 37 / Grunau, M.
Aufbereitung von CAD-Daten für die Fertigung
CAD-CAM Report Nr. 9, 1994

- / 38 / Hagen, G.
Erkennen und Verstehen durch Datenstrukturanalyse beim Konstruieren mit
CAD-Möglichkeiten zur Analyse von Informationen
Dissertation, Technische Fakultät der Universität Erlangen-Nürnberg, 1992
- / 39 / Hagen, G.; Müller, B.; Nesner, N. und Weber, J.
Produktmodell statt Zeichnung
CAD-CAM Report, Nr. 9, 1994
- / 40 / Hansen, T. L.
The C++ Answer Book
Addison-Wesley Publishing Company, Reading, Massachusetts, 1990
- / 41 / Helmerich, R. und Wawer, V.
Turbolader für die Konstruktion :
Integration von CAD und EDM
CAD-CAM Report Nr. 1, 1996
- / 42 / Helpenstein, H. J.
STEP als Verbindung zwischen Systemen
Konstruktion 45, Springer-Verlag 1993
- / 43 / Heuer, T.
Definieren einer Benutzerschnittstelle für ein zu entwickelndes
CAD-System mit Schwerpunkt in der Unterstützung
von Fertigungsverfahren
Studienarbeit, LbK, FB 15
Universität-GH Kassel, 1997
- / 44 / Hoffmann, C.M.
Geometric & Solid Modelling
Morgan Kaufmann Publishers Inc., Palo Alto, 1989
- / 45 / Hoffmann, C.M.; Hopcroft, J.E. und Karasick, M.S.
Robust Set Operations on Polyhedral Solids
IEEE Computer Graphics & Applications, November 1989
- / 46 / Hoimyr, N.
CAD/CAM and the Exchange of Product data
<http://Cadd.cern.ch/presentations/CsC96>
- / 47 / Hoschek, J. und Lasser, D.
Grundlagen der geometrischen Datenverarbeitung
B. G. Teubner, Stuttgart, 1989

Kapitel 8 Literaturverzeichnis

- / 48 / Jäger, K.-W.
High-lights der nächsten CAD/CAE-Software-Generation
CAD-CAM Report, Nr. 6, 1990
- / 49 / Kalay, Y. E.
Worldview : An Integrated Geometric-Modelling/Drafting System
IEEE CG&A, February 1987
- / 50 / Kempf, R. und Frazier C.
OpenGL Reference Manual
Silicon Graphics, 1997
- / 51 / Kimmig, H.
3D-CAD-Bearbeitung und EDM-Einsatz
CAD-CAM Report Nr. 3, 1995
- / 52 / Kimmig, H.
Feature-basierende Modellierung
CAD-CAM Report, Nr.6, 1996
- / 53 / Klein, B.
Triz/Tips
Theorie des erfinderischen Problemlösens
Vorlesung Wintersemester 2000/2001
LbK, FB 15
Universität-GH Kassel, 2000
- / 54 / Klein, M.
Einführung in die DIN-Normen
B.G. Teubner, Stuttgart und Beuth Verlag, Berlin und Köln, 1980
- / 55 / Klein, R.
Algorithmische Geometrie
Addison-Wesley-Longman, 1997
- / 56 / Klimmer, M.
Wirtschaftliche Aspekte der Integration von rechnergestützter Konstruktion
und NC-Programmierung
VDI-Z 135, 1993 Nr.11/12
- / 57 / Knechtel, U.
Produktdaten-Management im World Wide Web
CAD-CAM Report, Nr. 6, 1997

- / 58 / Kobori, K. I.; Futagami, N. und Nishioka, I.
Automated generation of simply connected solid
objects from wire-frame data using operations on graphs
The Visual Computer, Nr. 2, Springer-Verlag 1986
- / 59 / Koller, Rudolf
Konstruktionslehre für den Maschinenbau
Springer, Berlin Heidelberg, 1994
- / 60 / Krause, F. -L.; Gisela, M.; Rieger, E.; Stephan, M. und Ulbrich, A.
Features als semantische Objekte
CAD-CAM Report, Nr. 5 und Nr. 7, 1994
- / 61 / Krause, F.-L.; Schlingheider, J.
Entwickeln und Konstruieren mit wissensbasierten
Softwarewerkzeugen - ein Überblick
VDI Berichte 903 Seite 205-230
VDI-Verlag, Düsseldorf, 1991
- / 62 / Kurtz, T.
Entwicklung portabler CAD-Applikationen
CAD-CAM Report, Nr. 6, 1994
- / 63 / Langner, Th.
Voraussetzungen für profitable CAD/CAM-Anwendungen
Konstruktion 45, Springer-Verlag 1993
- / 64 / Laubert, T.
Grundlagen des Konstruierens unter besonderer
Berücksichtigung der Fertigung
Diplomarbeit, LbK , FB 15
Universität-GH Kassel 1988
- / 65 / Lee, Y. T. und Requicha, A. G.
Algorithms for Computing the Volume and Other Integral Properties of Solids.
I. Known Methods and Open Issues
Communications of the ACM, Vol. 25, Number 9, September 1982
- / 66 / Lee, Y. T. und Requicha, A. G.
Algorithms for Computing the Volume and Other Integral Properties of Solids.
II. A Family of Algorithms Based on Representation Conversion
and Cellular Approximation
Communications of the ACM, Vol. 25, Number 9, September 1982

- / 67 / Li, Zhengang
Rechnergestützte Hilfsmittel für eine wissensunterstützte Konstruktionssystematik
VDI-Verlag, Düsseldorf, 1995
- / 68 / Linner, S.
Entwicklungszeiten verkürzen durch graphische Simulation
VDI 993.1, VDI-Verlag, Düsseldorf, 1994
- / 69 / Lutz, M.
Methode zur Sicherung der geometrischen Kondition bei der Punktlage-Bestimmung
ZwF 83, Carl Hanser Verlag 1988
- / 70 / Lyche, T. und Schumaker, L. L.
Mathematical Methods in Computer Aided Geometric Design II
Academic Press, San Diego, 1992
- / 71 / Maihack, S.
CAD-Standards: IGES und VDAIS in der Praxis
CAD-CAM Report, Nr. 10, 1995
- / 72 / Maihack, S.
Schnittstellentools für neutrale CAD-Schnittstellen
CAD-CAM Report Nr. 9, 1994
- / 73 / Meerkamm, H.
CAD und Fertigungsgerechtes Konstruieren
Workshop Design-Konstruktion 19, ICED '90, Dubrovnik, 1990
- / 74 / Meerkamm, H.
Rechnerunterstützung beim fertigungsgerechten Konstruieren -
Möglichkeiten und Stand der Technik
Workshop Design-Konstruktion 20, ICED '91, Zürich, 1991
- / 75 / Meerkamm, H.; Finkenwirth, K. und Räse, U.
Fertigungsgerechtes Konstruieren mit CAD-Systemen
Konstruktion 42(1990)293-298, Springer-Verlag
- / 76 / Meerkamm, H.; Krause, D. und Rösch, S.
Analysemöglichkeiten für Blechbiegeteile im Konstruktionssystem mfk
CIM-Management, Nr. 10 Oldenbourg-Verlag, 1994

- / 77 / Meinsen, S.
Wissensmanagement
- Ein zukunftsweisendes Managementkonzept gewinnt an Bedeutung -
<http://www.tob-online.de/artikel/wissen.htm>
- / 78 / Meyers, S.
Effektiv C++ programmieren
50 Möglichkeiten zur Verbesserung Ihrer Programme
Addison-Wesley, Bonn, 1995
- / 79 / Miller, J. R.
Analysis of Quadric-Surface-Based Solid Models
IEEE Computer Graphics & Applications, January 1988
- / 80 / Mortenson, M. E.
Computer Graphics
An Introduction to the Mathematics and Geometry
Industrial Press Inc. New York, 1989
- / 81 / Mortenson, M.E.
Geometric Modelling
John Wiley & Sons, 1997
- / 82 / Müller, C.
Featurebasierende Konstruktion
CAD-CAM Report Nr. 3 1995
- / 83 / Müller, J.; Praß, P. und Beitz, W.
Modelle beim Konstruieren
Konstruktion 44(1992)319-314, Springer-Verlag 1992
- / 84 / Myers, B. A.
A Taxonomy of Window Manager User Interfaces
IEEE, Computer Graphics & Applications, September 1988
- / 85 / Nievergelt, J. und Preparata, F. P.
Plane-Sweep Algorithms for Intersecting Geometric Figures
Communications of the ACM, Number 10, October 1982
- / 86 / N. N.
3D Graphics Programming with QuickDraw 3D
Addison-Wesley, Bonn, 1995

- / 87 / N. N.
3. neutrale Marktstudie über Mechanik-CAD:
Elektronisches Zeichenbrett auf dem Rückzug
<http://www.cadcircle.de/infos/cpresse-6.htm>
- / 88 / N. N.
ACIS 3D Toolkit
Spatial Technology Inc., Boulder Colorado, Version 5, 1999
- / 89 / N. N.
Anwendungsprotokoll 214
ProSTEP GmbH
http://www.prostep.de/d_ap214.html, 1999
- / 90 / N. N.
CADDSS5 Systemhandbücher
Computervision 1996
- / 91 / N. N.
Data Interchange
<http://www.dip.co.uk/articles/communic/comms.html>
- / 92 / N. N.
EMS 3, Intergraph setzt auf Durchgängigkeit
CAD-CAM Report, Nr. 8, 1994
- / 93 / N. N.
HOOPS GRAPHICS SYSTEM COOKBOOK
ITHACA Software
1301 Manna Village Parkway, Manela CA
- / 94 / N. N.
HOOPS GRAPHICS SYSTEM TUTORIAL
ITHACA Software
1301 Manna Village Parkway
Manela CA
- / 95 / N. N.
I-DEAS: Model Solution and Optimization, User Guide
Structural Dynamics Research Corporation (SDRC), 1990
- / 96 / N. N.
Installing and Using CV-DORS ISSM
Computervision 1996

- / 97 / N. N.
Integriertes Logik und geometriverarbeitendes CAD-System (KRONOS)
http://www.informatik.uni-karlsruhe.de/13V_HTML/FORSCHUNGSVORHABEN/00228235.html
- / 98 / N. N.
Produktdatentechnologie und STEP
ProSTEP GmbH
<http://www.prostep.de/wis51.htm>, 1999
- / 99 / N. N.
Mit ESPRIT auf dem Weg zum Concurrent Design
CAD-CAM Report, Nr. 1, 1993
- / 100 / N. N.
Modelling Knowledge About Complex Technical Systems for Multiple Use
<http://www.uni-karlsruhe.de/ibk/FORDAT/projekte/KA0196.htm>
- / 101 / N. N.
THE ICAD SYSTEM
http://www.ktiworld.com/our_products/icad.shtml
- / 102 / Nolte, R.
Durchgängige Kinematik- und Kinetostatiklösung für
Einsteiger und Profis
CAD/CAM 1/96
- / 103 / Nye, A. und O'Reilly, T.
X Toolkit Intrinsic Programming Manual
O'Reilly & Associates, Inc., Sebastopol
- / 104 / Pahl, G. und Beitz, W.
Bewertungsmethoden als Entscheidungshilfe zur Auswahl von Lösungsvarianten
Konstruktion 25(1973)29-32, Springer Verlag
- / 105 / Pahl, G. und Beitz, W.
Konstruktionslehre
2. Auflage, Springer Verlag, Berlin, 1986
- / 106 / Pareigis, B.
Analytische und projektive Geometrie für die Computer-Graphic
B. G. Teubner, Stuttgart, 1990

- / 107 / Piegler, L.
On NURBS: A Survey
Computer Graphics & Applications, IEEE, 1991, Nr. 1
- / 108 / Potthast, A.
Datenaustausch zwischen STEP und ACIS
CAD-CAM Report, Nr. 3, 1996
- / 109 / Press, W.W. H.; Flannery, B. P.; Teukolsky, S. A. und Vetterling, W. T.
Numerical Recipes in C
Cambridge University Press, 1988
- / 110 / Reiß, M.
Konstrukteur- und konstruktionsgerechte 3D-Modellierung mit
Hilfe von Mischmodellen
Konstruktion 46, Springer-Verlag 1994
- / 111 / Rembold, U.
Die Anwendung von Expertensystemen im Maschinenbau
FdZ 30
- / 112 / Richter, M.M.; Bernardi, A.; Klauck, Ch. und Legleitner, R.
Akquisition und Repräsentation von technischen Wissen für Planungsaufgaben im
Bereich der Fertigungstechnik
VDI 903, VDI-Verlag, Düsseldorf, 1991
- / 113 / Rogers, D.F. und Adams, J.A.
Mathematical Elements for Computer Graphics
McGraw-Hill, 1976
- / 114 / Roth, K.
Konstruieren mit Konstruktionskatalogen I
Springer Verlag, Berlin, 1982
- / 115 / Salmon, R. und Slater, M.
Computer Graphics Systems & Concepts
Addison-Wesley, 1987
- / 116 / Salomons, O.
Computer Aided Design
<http://www.opm.wb.utwente.nl/staff/Otto/thesis/contents.html>

- / 117 / Schaal, H.-G.
Wissensmanagement in der Praxis
<http://www.experteam.de/browser/data/ber576.text.htm>
- / 118 / Schäfer, S.
Objektorientierte Entwurfsmethoden
Addison-Wesley, Bonn, 1994
- / 119 / Schütz, B.; Ratfisch, U. und Beitz, W.
Die „Kontaktmethode“ - eine Methode zur Vereinfachung rechnergestützter
Gestaltvariationen durch technisches Anordnen
Konstruktion 44(1992)43-50, Springer-Verlag
- / 120 / Schütz, B.; Scholz, E. und Tünte, M.
Anwendung von 3D-CAD-Systemen - aber wie ?
Universelle 3D-Werkzeuge versus Branchenlösungen
Konstruktion 46, Springer-Verlag 1994
- / 121 / Schützer, K. und Schulz, H.
Integrierte Konstruktion, Arbeitsplanung und automatische NC-Programmierung
durch Anwendung von Fertigungselementen
Konstruktion 47(1995)383-387, Springer-Verlag
- / 122 / Schulte, M. und Weber, C.
Stand und Integration in der Forschung auf den Gebieten
Konstruktionsmethodik und CAD
Konstruktion 45, Springer-Verlag, 1993
- / 123 / Schwenke, H. und Tegel, O.
Wissensverarbeitung in CAD-Anwenderprozeduren
Konstruktion 44, Springer-Verlag, 1992
- / 124 / Sedgewick, R.
Algorithms
Addison-Wesley, Publishing Company, Reading, Massachusetts, 1988
- / 125 / Sen, K.
Fertigungsorientierte Konstruktion
Studienarbeit, LbK, FB 15
Universität-GH Kassel, 1999
- / 126 / Sandler, U.
Flexibler Modellierer vereinfacht die 3D-Konstruktion
CAD/CAM, Nr. 3, 1994

- / 127 / Sendler, U.
Kaskaden von Objekten
CAD/CAM 1/96
- / 128 / Sengupta, S.; Korobkin, C. P.
C++ Object-Oriented Data Structures
Springer 1994
- / 129 / Shah, J. J. und Mäntylä, M.
Parametric and Feature-Based CAD/CAM
John Wiley & Sons, 1995
- / 130 / Solonon, D.
Inside Microsoft Windows NT
Der offizielle Führer zur Architektur von Windows NT 4.0
Microsoft Press, München, 1998
- / 131 / Specht, D.; Forkel, M. und Göbler, Th.
Semantische Modellierung von technischen Objekten und Handlungen
VDI 903, VDI-Verlag, Düsseldorf, 1991
- / 132 / Spur, G. und Krause, F.-L.
Das virtuelle Produkt: Management der CAD-Technik
Carl Hanser Verlag, München, Wien, 1997
- / 133 / Stroustrup, B.
Die C++ Programmiersprache
Addison Wesley 1992
- / 134 / Stubbe, H.-C.
Lösungsansatz für den Datenaustausch
CAD-CAM Report, Nr. 2, 1994
- / 135 / Studer, R. und Erdmann, M.
Wissensmanagement
<http://www.aifb.uni-karlsruhe.de/wbs1/Lehrveranstaltungen/wm.thml>
- / 136 / Tempke, M. und Bittner, J.
Unigraphics: 3D-CAD nicht nur zum Abheben
CAD-CAM Report, Nr. 6, 1995

- / 137 / Tilove, R. B.
Set Membership Classification :
A Unified Approach to Geometric Intersection Problems
IEEE Transaction on Computers, No. 10, October 1980
- / 138 / Turner, J. U.
Accurate Solid Modeling Using Polyhedral Approximations
IEEE Computer Graphics & Applications, May 1988
- / 139 / Ulrich, K.
Moderne Blechkonstruktion mit 3D-Formfeatures
CAD-CAM Report, Nr. 1, 1996
- / 140 / Umbach, F.
Entwicklung des Programmbausteins TOPKON zum Aufbau und
zur Darstellung von Produktstrukturen im wissensbasiertem
Konstruktionssystem WISKON
Diplomarbeit, LbK, FB15,
Universität-GH Kassel, 1989
- / 141 / Vajna, S.
Process Threads - nur ein neuer Name für bekannte Vorgehensweisen ?
CAD-CAM Report, Nr. 6, 1997
- / 142 / Vajna, S.
Zukunftsperspektiven für CAx-Technologien
CAD-CAM Report, Nr. 6, 1997
- / 143 / Vajna, S. und Wegner, B.
Stand und Tendenzen der rechnerunterstützten Modellierung
CAD-CAM Report, Nr. 2, 1996
- / 144 / VDI 2216: Datenverarbeitung in der Konstruktion - Einführungsstrategien
und Wirtschaftlichkeit
VDI-Verlag, Düsseldorf, 1996
- / 145 / VDI 2218: Feature-Technologie (Entwurf)
VDI-Verlag, Düsseldorf, 1999
- / 146 / VDI 2221: Methodik zum Entwickeln und Konstruieren technischer
Systeme und Produkte
VDI-Verlag, Düsseldorf, 1986

- / 147 / VDI 2222 Blatt 1:Konstruktionsmethodik : Konzipieren technischer Produkte
VDI-Verlag, Düsseldorf, 1997
- / 148 / VDI 2222 Blatt 2:Konstruktionsmethodik :
Erstellung und Anwendung von Konstruktionskatalogen
VDI-Verlag, Düsseldorf, 1977
- / 149 / VDI 2223: Konstruktionsmethodik -Entwurf - :
Methodisches Entwerfen technischer Produkte
VDI-Verlag, Düsseldorf, 1999
- / 150 / Voss, G.
Object-Oriented Programming: An Introduction
Osborne McGraw-Hill, Berkeley, 1991
- / 151 / Weck, M.
Werkzeugmaschinen, Fertigungssysteme
Band1 Maschinenarten, Bauformen und Anwendungsbereiche
VDI Verlag, Düsseldorf, 1999
- / 152 / Welp, E. G.
Mit Strategie zum erfolgreichen CAD-Einsatz - ein Erfahrungsbericht
Konstruktion 45, Springer-Verlag, 1993
- / 153 / Winkler, M.
Entwicklung von Anwendersoftware für den Entwurf komplexer Baugruppen
Konstruktion 44, Springer-Verlag, 1992
- / 154 / Wüstenberg, D.
Kreativität in der Konstruktion
Konstruktion 45, Springer-Verlag, 1993

Abkürzungen

Benutzte Abkürzungen :

API	Application Programming Interface
CE	Concurrent Engineering
BREP	Boundary Representation
CGM	Computergraphics Metafile
CLData	Cutter Location Data
CSG	Constructive Solid Geometry
DFA	Design for Assembly
DFM	Design for Manufacturing
DFX	X = Assembly, Manufacturing, Life-Cycle, Environment
DXF	Drawing Exchange Format
FBDS	Feature Based Design System
GKS	Grafisches Kernsystem
HOOPS	Hierarchical Object Oriented Picture System
IGES	Initial Graphical Exchange Specification
KACTUS	Modelling Knowledge About Complex
KRONOS	Integriertes Logik und Geometrie verarbeitendes CAD-System
	Technical System for multiple Use
MAP	Manufacturing Automation Protocol
NURBS	Nonuniform Rational B-Spline Curves
OpenGL	Open Graphics Library
PHIGS	Programmer's Hierarchical Interactive Graphics System
RID	Rechnerinterne Darstellung
SDAI	Standard Data Access Interface
SET	Standard d'Echange et de Transfer
STEP	Standard for the Exchange of Product Data
VDAFS	Flächenschnittstelle des Verbandes der deutschen Automobilindustrie
VDAIS	IGES Subset des Verbandes der deutschen Automobilindustrie