

Institut für Maschinen- und Fahrzeugtechnik – Lehrstuhl für Maschinenelemente
Technische Universität München

Modell einer durchgängig rechnerbasierten Produktentwicklung

Andreas Dyla

Vollständiger Abdruck der von der Fakultät für Maschinenwesen
der Technischen Universität München zur Erlangung
des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. K. Bender

Prüfer der Dissertation: 1. Univ.-Prof. Dr.-Ing. B.-R. Höhn
2. Univ.-Prof. Dr.-Ing. U. Lindemann

Die Dissertation wurde am 17.01.2002 bei der Technischen Universität München eingereicht
und durch die Fakultät für Maschinenwesen am 25.04.2002 angenommen.

INHALTSVERZEICHNIS

1	Einleitung	1
2	Zielsetzung	4
3	Stand der Technik	5
3.1	Verfügbarkeit und Einsatzmöglichkeiten alleinstehender Rechnerprogramme	5
3.1.1	Berechnungs- und Simulationsprogramme	5
3.1.2	CAD-Systeme	6
3.2	Datenschnittstellen für den Datenaustausch zwischen Rechnerprogrammen	7
3.2.1	Systemspezifische und quasineutrale Datenformate zur Geometrieübertragung	8
3.2.2	Systemneutrale Datenformate zur Geometrieübertragung	8
3.2.3	Datenformate zur Übertragung allgemeiner Produktdaten	10
3.2.4	Innerhalb der vorliegenden Arbeit verwendete Datenformate	11
3.3	Rechnerintegrierte-, rechnerunterstützte- und programmverbindende Systeme	12
3.3.1	Methodenorientierte Ansätze	13
3.3.2	Objektorientierte Ansätze	15
3.3.3	Bewertung und Bedeutung vorhandener Systeme	16
3.4	Produkt- und Prozessmodelle	17
3.4.1	Grundlagen und Begriffe	17
3.4.2	Implementierung von Produkt- und Prozessmodellen	19
3.4.3	Methoden zur Prozessmodellierung	20
3.4.4	Beispiele für Produkt- und Prozessmodelle	20
3.5	Systeme für ein Produkt- und Prozessdatenmanagement	22
3.6	Grundlagen von STEP	24
3.6.1	Bestandteile von STEP	25
3.6.2	STEP-Anwendungsprotokoll 214	29
4	Datenaustausch zwischen Berechnung und Gestaltung	35
4.1	Darstellung der Problematik	36
4.2	Statischer Austausch von Produktdaten zwischen Programmen	37
4.2.1	Verknüpfung von Programmen über systemspezifische Konvertierungsroutinen	37
4.2.2	Verknüpfung von Programmen über systemneutrale Schnittstellen	38
4.2.3	Verknüpfung von Programmen über ein Produktmodell	39
4.3	Zeitlich-durchgängige Verknüpfung von Berechnung und Gestaltung	41
4.3.1	Kontinuierliche bidirektionale Kommunikation	41
4.3.2	Sequenzielle bidirektionale Kopplung	42
4.4	Methoden zur Entwicklung eines auf STEP basierenden Produktreferenzmodells	43
4.4.1	Vorteile durch die Verwendung bestehender Ansätze für die Produktmodellierung	43

4.4.2	Gegenüberstellung von Möglichkeiten zur Erstellung eines Produktreferenzmodells mit STEP	44
4.4.3	Resultierende Vorgehensweise zur Erstellung eines anwendungsbezogenen STEP-Produktreferenzmodells	46
4.5	Prozessabbildung	47
5	Entwicklung eines durchgängig rechnerbasierten Programmsystems zur Produktentwicklung	49
5.1	Übersicht und Anforderungen an einen programmsystemgeführten Konstruktionsablauf	49
5.2	Systemarchitektur	50
5.2.1	Basisfunktionen	52
5.2.2	Anwendungsfunktionen	52
5.2.3	Externe Anwendungen	53
5.2.4	Grafische Benutzeroberfläche	53
5.2.5	Daten- und Dateienverwaltung	54
5.2.6	Prozessmanagement-System (PMS)	56
5.2.7	Externe Datenschnittstelle	57
5.3	Zusammenspiel der Systemkomponenten während der Produktentwicklung	58
5.3.1	Kommunikation innerhalb des rechnerbasierten Programmsystems	59
5.3.2	Kommunikation mit externen Anwendungen	64
5.4	Kommunikation mit separaten externen Anwendungen	67
6	Entwicklung eines systemneutralen Produktmodells für Getriebe	69
6.1	Datentechnischer Inhalt des Produktmodells	69
6.1.1	Versionen und Sichten	69
6.1.2	Bauteile und Produktstruktur	69
6.1.3	Geometrie und Topologie	70
6.1.4	Produkteigenschaften und Getriebedaten	71
6.1.5	Externe Dokumente	72
6.1.6	Verknüpfung von Produktstruktur und Produkteigenschaften	72
6.2	Das AP214 als Grundlage für das Produktmodell für Getriebe	73
6.2.1	Kinematikstruktur des AP214 als Kern des Produktreferenzmodells	73
6.2.2	Produktstruktur des AP214 als Kern des Produktreferenzmodells	75
6.2.3	Bewertung und Auswahl	75
6.3	Realisierung des Produktmodells für Getriebe	76
6.3.1	Struktur des Produktreferenzmodells	76
6.3.2	Aus dem AP214 übernommene Anwendungsobjekte	82
6.3.3	Im Produktreferenzmodell für Getriebe neu definierte Anwendungsobjekte	83
6.3.4	Mechanismus der Abbildung des Produktreferenzmodells auf das AP214	84

7	Übertragung des rechnerbasierten Programmsystems zur Anwendung in der Getriebeentwicklung	87
7.1	Prozessmodell Getriebeentwicklung	87
7.2	Verwendete Programme zur Gestaltung und Berechnung von Getrieben	88
7.2.1	Auslegungsprogramme	88
7.2.2	CAD-Systeme	88
7.2.3	Nachrechnungs- und Simulationsprogramme	89
7.3	Anpassung der Benutzungsoberfläche	91
7.4	Integration externer Programme und des Produktreferenzmodells für Getriebe in das rechnerbasierte Programmsystem	92
7.5	Anpassung und Eingliederung des rechnerbasierten Programmsystems für Getriebe in vorhandene Strukturen	94
8	Anwendung und Funktion des rechnerbasierten Programmsystems an einem Beispielgetriebe	96
8.1	Vorstellung des Beispielgetriebes	96
8.2	Initialisierung des rechnerbasierten Programmsystems	98
8.3	Getriebeentwurf	98
8.3.1	Traditionelle Vorgehensweise	98
8.3.2	Getriebeauslegung mit dem interaktiven Getriebe-Auslegungsprogramm	99
8.3.3	Diskussion der mit dem Getriebe-Auslegungsprogramm erzielten Ergebnisse	102
8.3.4	Datenweitergabe nach Beendigung des interaktiven Getriebe-Auslegungsprogramms	103
8.4	Geometriemodellierung	105
8.5	Berechnung und Simulation	106
8.6	Visualisierung aller Daten und der Produktstruktur	107
8.7	Weiterverarbeitung und Übergabe von Geometriemodellen und Produktdaten	108
8.7.1	Nutzung mit integrierten externen Anwendungen	108
8.7.2	Nutzung mit separaten externen Anwendungen	109
8.8	Ergonomische Unterstützung im rechnerbasierten Programmsystem	110
9	Einordnung und Positionierung der rechnerbasierten Produktentwicklung	112
9.1	Einordnung und Weiterverwendung der Ergebnisse und entwickelten Methoden	112
9.2	Positionierung des rechnerbasierten Programmsystems in das Umfeld der Gestaltung und Berechnung	114
10	Zusammenfassung und Ausblick	118
10.1	Zusammenfassung	118
10.2	Ausblick	119
11	Literatur	121

1 Einleitung

„Ein Mann¹ der konstruieren will, der schaue erst mal und denke“ [N3]. Dieses Wort von G. Niemann gilt auch heute noch, selbst im Zeitalter der zunehmend rechnerunterstützten Produktentwicklung. Bild 1-1 zeigt einen Überblick über die Teilbereiche der komplexen Ingenieurarbeit und deren Verknüpfung. Nach NIEMANN et al. [N3] stellt die Konstruktion eine zentrale Aufgabe dar und erfordert ein Denken und Arbeiten in Regelkreisen, die sich durch dynamische Rückkopplungen von den frühen Entwicklungsphasen bis zum Recycling oder der Entsorgung des Produkts ergeben.

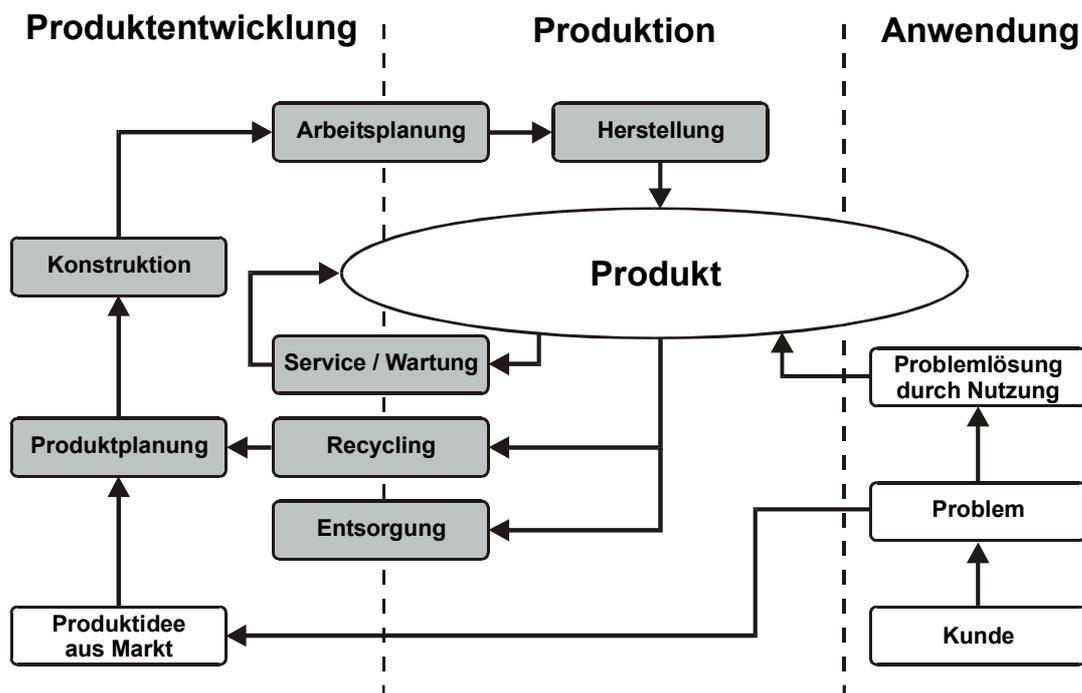


Bild 1-1: Teilaufgaben der Ingenieurarbeit (grau hinterlegt) im Produktlebenszyklus

Unter integrierter Produktentwicklung wird nach EHRENSPIEL [E1] ein ganzheitlicher Ansatz verstanden, um die Entwicklungszeit, Produktkosten und Produktqualität durch eine Integration der am Entstehungsprozess eines Produkts beteiligten Komponenten zu optimieren. Ziel ist es, nicht nur die Ergebnisse der Entwicklungsarbeit zu dokumentieren, sondern auch die eingesetzten Methoden, Vorgehensweisen, Entscheidungsgründe, Produktentwicklungshistorie und sich daraus ergebende Erfahrungen. Diesbezüglich versteht LINDEMANN [L3] unter „Innovation“ nicht nur die Innovation am Produkt selbst, sondern vor allem an den Prozessen zur Produktentstehung. Dies führt durch konsequenten Methodeneinsatz in Verbindung mit einer adäquaten Unternehmenskultur zu einer Verhaltensinnovation. Der ganzheitliche Ansatz der integrierten Produktentwicklung erfordert auch nach MEERKAMM

¹ „Das gilt natürlich auch für eine Frau, die konstruieren will“ [N3]. Für einen vereinfachten Sprachgebrauch wird im Folgenden von einer differenzierten Erwähnung abgesehen. Das Geschlecht des Entwicklers / Konstrukteurs spielt selbstverständlich keine Rolle.

[M4] ein geändertes Denkverhalten und die verstärkte Betrachtung der Wechselwirkung zwischen Produkt und Prozess.

Hier setzt die rechnerunterstützte Produktentwicklung an. KRAUSE [K11] betont, dass im Bereich der Produktentstehung die Informationstechnik nicht mehr nur als Hilfsmittel zur Unterstützung konstruktionsbezogener Aktivitäten begriffen werden darf, sondern auf dem Wege ist, die Möglichkeiten der konventionellen Produktentwicklung in Richtung einer neuartigen Produktentstehung im Rechner zu verändern. Vor dem Hintergrund zunehmend globaler Märkte und Kooperationen werden nach BENDER et al. [B8] diejenigen Unternehmen entscheidende Wettbewerbsvorteile erlangen, deren Informationssysteme sich in der Produktentwicklung schnell und flexibel auf die stetig ändernden Randbedingungen anpassen lassen.

Ein breites Spektrum vorhandener isolierter Lösungen zur rechnergestützten Produktentwicklung ist nach ANDERL&PHILIPP [A10] besonders dort zu beobachten, wo sich eine realitätsnahe Darstellung in einem analytischen mathematischen Modell abbilden lässt und die anzuwendenden Verarbeitungsschritte algorithmisierbar sind. Dies gilt im Bereich der mechanischen Konstruktion speziell für die Gestaltung, Berechnung und Simulation. Solange die einzelnen Softwaresysteme und Programme im Wesentlichen Inselösungen waren, war ihr Einfluss auf die zugrundeliegenden Prozesse gering. Mit der einsetzenden Verknüpfung der Inseln hat die Nutzung der Softwaresysteme eine Eigendynamik entwickelt, die es zu kontrollieren gilt. Aus diesem Grund wurde das Schwerpunktprogramm „Innovative rechnerunterstützte Konstruktionsprozesse: Integration von Berechnung und Gestaltung“ [B3] der Deutschen Forschungsgemeinschaft (DFG) ins Leben gerufen. Die Initiatoren des Schwerpunktprogramms BEITZ, KOLLMANN und LECHNER [B3] und dessen Koordinator MERTENS [M8] hängen die Zielsetzung des Schwerpunktprogramms sinngemäß an folgenden Fragestellungen auf:

- **Systemneutrale Datenschnittstellen:** Im Interesse der Kopplungsmöglichkeiten von unterschiedlichen CAD-Systemen und Berechnungs-/Simulationsprogrammen sind langfristig systemneutrale Datenschnittstellen erforderlich. Reicht hier die internationale Standardisierung der Produktdatentechnik, z.B. STEP [I1], für eine konsistente und widerspruchsfreie Verknüpfung aus oder müssen die Standards erweitert werden? Die Verwendung von Produktmodellen als erweiterte Datenschnittstelle und Integrationskern von neuartigen Konstruktionssystemen haben nach GRABOWSKI [G8] die besondere Fähigkeit, den Konstrukteur in allen Arbeitsschritten adäquat zu unterstützen. Nach VAJNA&WEGNER [V7] setzt die Integration der Tätigkeiten und Abläufe im Produktentstehungsprozess gar voraus, dass die Informationen in einem einheitlichen Datenmodell erfasst werden. ANDERL&TRIPPER [A12] drängen zur Nutzung des systemneutralen Produktmodells STEP, welches zur Zeit das umfassendste und einzige genormte Datenmodell darstellt.
- **Produktmodell:** Wie müssen digitale Geometriemodelle und Daten aus der Berechnung/Simulation hinsichtlich ihrer Operations- und Datenstruktur sowie der Datenspeicherung aufbereitet werden und aufgebaut sein, damit unterschiedliche Programme und Systeme darauf zugreifen, diese verändern und wieder zurückschreiben können? Nach GAUSEMEIER et al. [G3] liegt die Herausforderung in der Integration der Disziplinen, um die Gesamtfunktionalität des Produkts und dessen Entstehungsprozesses effizient zu

gewährleisten. Dazu ist die Koordination der Entwicklungstätigkeiten, die aufgabenspezifische Bereitstellung der CAE¹-Werkzeuge und Informationen zur Bearbeitung und Analyse des Produktmodells sowie der Austausch von Informationen zwischen den am Entwicklungsprozess beteiligten Programmen und Systemen notwendig. GRABOWSKI et al. [G9] betonen die Notwendigkeit der Verwendung eines einzigen Produktmodells als Vereinigungsmenge aller Daten und Modelle der ablaufenden Prozesse.

- **Prozessmanagement:** Wie ist die Datensicherung und die Speicherung verschiedener Gestaltungszustände entlang der zeitlichen Komponente der Produktentwicklung auszuführen, um trotz kontinuierlicher Detaillierung das Vorgehen und grundlegende Entscheidungen für spätere Änderungen oder Varianten zu sichern? Dies ist nicht nur für die Erzielung eines schnellen und redundanzfreien Prozessablaufs von Vorteil. Nach KRAUSE et al. [K14] zwingt die Gesetzgebung die Unternehmen zu einer ordentlichen Langzeitarchivierung der Daten und Dateien und empfiehlt die Dokumentation des Produkts und Produktmodells, ihren Entstehungsgeschichten und den damit verbundenen Prozessen.
- **Konstruktionsleitsystem:** Inwieweit ist es arbeitsmethodisch, programmtechnisch und aufwandsmäßig lohnend und machbar, mehrere Programme aus Berechnung/Simulation und Gestaltung von einem Konstruktionsleitsystem aus zu verwalten und einzeln oder integriert anzusteuern? Auch WEBER et al. [W4] stellen das Problem der zentralen Steuerung des Gesamtprozesses als Notwendigkeit in den Vordergrund. Nach KRAUSE et al. [K13] erfordert eine durchgängig digitale Rechnerunterstützung sogar eine gemeinsame Infrastruktur und einheitliche Programmoberflächen.
- **Frühe Phasen:** Wie kann der Konstrukteur in den frühen Phasen der Produktentwicklung sowohl Gestaltung als auch rechnerische Auslegung integriert selbst durchführen und in kurzer Zeit das Konstruktionsprodukt ganzheitlich formen und erfassen?

Die genannten Fragestellungen sind bisher nicht umfassend gelöst und können nicht isoliert betrachtet werden. Die Herausforderung besteht in der Entwicklung eines rechnerbasierten Gesamtsystems zur Integration der in der Produktentwicklung vorherrschenden Disziplinen und einer somit möglich werdenden zusammenhängenden Beantwortung der offenen Fragestellungen. Vorteile könnten sich dann in der Qualität des realen und virtuellen Produkts und vor allem in den Prozessabläufen ergeben. Zusätzlich kann eine schnellere und somit kostengünstigere Produktentwicklung erzielt werden.

¹ CAE steht für „Computer Aided Engineering“ und umfasst alle Programme, Systeme und Methoden, die an einem rechnerunterstützten oder –basierten Entwicklungsprozess beteiligt sind.

2 Zielsetzung

Die Arbeit soll einen Beitrag für einen effizienten Produktentwicklungsprozess liefern, um Entwicklungszeit und -kosten zu senken und gleichzeitig die Qualität der Prozesse und des Produkts zu erhöhen. Dies wird mit einem Modell für eine durchgängig rechnerbasierte Produktentwicklung realisiert. Mit „rechnerbasierter Produktentwicklung“ [H11] werden die folgenden Ziele verknüpft:

- Alle Vorgänge am Produkt und Prozess von den frühen Konstruktionsphasen bis hin zur Fertigstellung werden am Rechner durchgängig ausgeübt und digital dokumentiert. Speziell die frühe Entstehungsphase eines Produkts soll mit eingebunden werden.
- Die Forderung nach einer Abbildung aller Vorgänge und Integration der beteiligten Komponenten verlangt ein eindeutiges, redundanzfreies und konstruktionsphasen-übergreifendes Produkt- und Prozessmanagement.
- Die Verknüpfung der beteiligten Systeme, speziell zwischen den vorhandenen Programmen und Systemen aus Gestaltung und Berechnung, soll bidirektional und systemneutral erfolgen, so dass eine möglichst große Anzahl von CAE-Systemen integriert werden kann.
- Die Produkt- und Prozessdaten sowie die digitalen Dokumente sollen zentral gespeichert, verwaltet und organisiert werden.
- Eine Benutzungsoberfläche stellt die Schnittstelle zwischen Anwender und Rechner dar und ermöglicht eine Transparenz der Produktstruktur und der zugrundeliegenden beziehungsweise möglichen Prozesse sowie eine Erfassung des Zusammenhangs zwischen Produkt- und Prozessmodell und den digitalen Dokumenten.
- Die Umsetzung soll weitestgehend systemneutral erfolgen, damit die rechnerbasierte Produktentwicklung in möglichst vielen vorhandenen Systemarchitekturen und für beliebige Produkte eingesetzt werden kann.

Die rechnerbasierte Produktentwicklung zielt auf eine Beantwortung der in Kapitel 1 genannten offenen Fragestellungen des DFG-Schwerpunktprogramms [B3][M8] ab. Aufgrund einer angestrebten Gesamtverknüpfung sollen die Fragen nicht isoliert voneinander, sondern im zeitlich-durchgängigen Entwicklungsprozess unter einer bidirektionalen Integration der beteiligten Programme und Systeme in gegenseitigem Bezug beantwortet werden.

Die oben genannten Ziele sollen zunächst Allgemein und dann am Beispiel der Produktentwicklung von Zahnradgetrieben und der dort vorherrschenden engen Kommunikation zwischen Gestaltung und Berechnung verifiziert werden [D7]. Getriebe eignen sich hierfür besonders, da sie einerseits über eine komplexe Bauteilstruktur verfügen, die aufwendige Prozesse zu deren Erzeugung fordern, und andererseits eine Vielzahl von Insellösungen im Bereich der Berechnungs- und Simulationsprogramme und CAD-Systeme existieren. Diese gilt es sinnvoll und konstruktionsphasen-übergreifend zu verknüpfen. Die frühen Phasen der Getriebeentwicklung sollen zudem auf eine ausreichende Rechnerunterstützung geprüft und um ein Getriebe-Auslegungsprogramm erweitert werden.

3 Stand der Technik

3.1 Verfügbarkeit und Einsatzmöglichkeiten alleinstehender Rechnerprogramme

Während der Produktentwicklung ist der Umgang mit Programmen und Systemen zur Konstruktion von Produkten nicht mehr wegzudenken. Mittlerweile gibt es eine Vielzahl von Programmen und Systemen, die entweder von Routineaufgaben entlasten oder komplexe Vorgänge unterstützen und somit zu einer Beschleunigung der Produktentwicklung und einer hohen Vorhersagegenauigkeit von Produkteigenschaften beitragen.

3.1.1 Berechnungs- und Simulationsprogramme

Im Allgemeinen verfolgt der Einsatz von Berechnungs- und Simulationsprogrammen die folgenden Ziele [S13]:

- Ermittlung der Sicherheit des Bauteils gegen Versagen
- Prüfung der Produktfunktionalität
- Optimierung hinsichtlich geringem Gewicht und Leichtbauweise
- Optimale Materialausnutzung
- Sicherstellung einer wirtschaftlichen Konstruktion und Herstellbarkeit
- Bewertung äußerer Einflüsse

Eine Einteilung der Berechnungs- und Simulationsprogramme entlang der zeitlichen Achse des Entwicklungsprozesses ergibt Auslegungs-, Nachrechnungs- und Optimierungsprogramme.

Die **Auslegungsprogramme** werden zum Grobentwurf der Produkteigenschaften in den frühen Phasen der Produktentwicklung (nach [V13] auch Konzeptphase genannt) verwendet. Nach MEERKAMM et al. [M5] sind die frühen Phasen deshalb so wichtig, weil in ihnen etwa zwei Drittel der späteren Produkteigenschaften festgelegt werden. Die Rechnerunterstützung in den frühen Phasen lässt sich noch weiter untergliedern. LINDEMANN et al. [L6] stellen ein Softwarepaket zur Förderung der Kreativität des Konstrukteurs in den frühesten Phasen der Ideenfindung vor. Einen zeitlichen Schritt weiter im Entwurf beschreibt SCHULZ [S7] die Konstruktionsumgebung MOSAIK zur Rechnerunterstützung des funktionsorientierten Grobentwurfs. Bei wiederkehrenden Konstruktionen reicht ein Einstieg in vereinfachte Berechnungsansätze aus, in denen bis dato unbekannte Produktmerkmale sinnvoll angenommen werden, damit eine möglichst frühe Berechnung möglich wird. In HÖHN et al. [H9][H10] wird dies am Beispiel der Getriebe beschrieben.

Die **Nachrechnungsprogramme** dienen zur Abbildung der realen Physik und einer Verifizierung der darauf beruhenden Produkteigenschaften. Ziel ist die Feststellung, ob Einfluss-

größen in zumutbaren Grenzen liegen, um Aussagen über die Sicherheit der Bauteile zu treffen. Aus diesem Grund berufen sich viele Nachrechnungsprogramme auf standardisierte oder genormte Vorgehensweisen zum Nachweis der gewünschten Produkteigenschaften. BEERMANN&KISSLING [B2] beschreiben für Standardberechnungen des Maschinenbaus den Trend weg von FEM-Systemen hin zu Berechnungsprogrammen für Maschinenelemente, da diese weniger aufwendig in der Durchführung sind. Das physikalische Produktverhalten wird aus Geometrie und Einsatzbedingungen errechnet und auf Erfüllung der Anforderungen untersucht. Bei Nachrechnungen werden oft vereinfachte physikalische Modelle und eine Ersatzgeometrie verwendet, die auf die für die Berechnung notwendigen Geometrieinformationen vereinfacht ist. STEGER [S18] teilt die Nachrechnungsprogramme in Programme für elementare technische Berechnungen, Festigkeitsberechnungen, Verformungsberechnungen, Stabilitätsberechnungen, Lebensdauerberechnungen, Berechnung des dynamischen Verhaltens und thermische Berechnungen ein. Simulationsprogramme dienen seiner Auffassung nach zur Berechnung zeitabhängiger Eigenschaften beim Zusammenwirken mehrerer Elemente. WINTER et al. [W14] geben einen Überblick über verfügbare FVA¹-Programme und Systeme für die Nachrechnung von Getrieben. REN [R3] beschreibt ein Programm zur Entwicklung von Getrieben.

Optimierungsprogramme sind oft in Nachrechnungsprogrammen integriert oder eigenständig. Ausgewählte Konstruktionsparameter werden in Grenzbereichen selbsttätig solange variiert, bis ein Wert oder eine Funktion ein Optimum einnimmt oder Restriktionen erfüllt werden. Dies kann nach SPUR&KRAUSE [S13] zum Beispiel eine Gestaltoptimierung von Bauteilen sein, um höchstmögliche Steifigkeiten, niedrige Gewichte oder ausgeglichene Lastverteilungen zu erreichen. Optimierungsprogramme bestehen in der Regel aus Geflechtern von Parametrisierungen der Daten.

WEBER [W4] führt aus, dass der Begriff „Berechnung“ mehr und mehr durch „Simulation“ ersetzt wird und sich dies mit der zunehmenden Funktionalität von Programmen und Systemen zur komplexen Produktbetrachtung in möglichst frühen Entwicklungsphasen begründen lässt. Die Berechnungs- und Simulationsprogramme sind somit fortlaufenden Erweiterungen unterworfen, auch wenn sie bereits über einen sehr hohen Stellenwert und eine breite Anwendung und Akzeptanz verfügen.

3.1.2 CAD-Systeme

Die CAD-Programme lassen sich in zwei- und dreidimensionale Systeme unterscheiden. Dreidimensionale Systeme verfügen über mehr oder minder automatisierte zweidimensionale Darstellungsformen.

Die Anwendung von CAD-Systemen ist Stand der Technik [V1]. GRABOWSKI&GEIGER [G12] und VAJNA [V3] stellen Vorteile und Probleme der CAD-Nutzung gegenüber. SPUR&KRAUSE [S12] und VAJNA et al. [V6] vertiefen die Anwendung der CAD-Systeme. GRABOWSKI [G8] beschreibt die Notwendigkeit eines Produktmodells als Integrationskern

¹ FVA ist die Abkürzung der Forschungsvereinigung Antriebstechnik e.V., welche als Fachverband dem VDMA untergeordnet ist.

für CAD-Systeme. Eine damit möglich werdende feature-, parametrische- oder lösungsmusterbasierte Modellierung erlaubt eine sichere, schnelle und kostengünstige Produktentwicklung. VAJNA et al. [V4][V7] unterstreichen auch eine durch Features mehr an das Objekt angelehnte 3D-Produktmodellierung in den CAD-Systemen. Da CAD-spezifische Features nicht in anderen CAD-Systemen verwendet werden können, ist eine VDI-Richtlinie [V3] in Vorbereitung.

Zusammenfassend ist eine Veränderung der Methodik beim dreidimensionalen Konstruieren in CAD-Systemen hin zu einem mehr objektbezogenen Vorgehen zu beobachten.

CAD-Systeme verfügen in der Regel auch über Programmierschnittstellen, um z.B. Berechnungsdaten in parametrische Zusammenhänge mit der Geometrie zu setzen. COOPER [C5] beschreibt die einfache Erzeugung dreidimensionaler Zahnräder mit Pro/Engineer, VOGEL&BUNTE [V14] das integrierte Konstruieren, Berechnen und Optimieren anhand von Pro/Engineer und Pro/Mechanica.

Nachteil von CAD-gebundenen Erweiterungen für die Berechnung/Simulation oder objektbezogene Modellierung ist die fehlende Kompatibilität mit anderen CAD-Systemen oder angrenzenden Anwendungen. Durch Nutzung der in Kapitel 3.2.2 vorgestellten und vorhandenen CAD-Datenschnittstellen könnte zwar die reine Geometrie übertragen werden, nicht jedoch darüber hinausgehende Informationen, die in den CAD-Erweiterungen enthalten sind.

3.2 Datenschnittstellen für den Datenaustausch zwischen Rechnerprogrammen

Unter Datenschnittstellen sind die nach ANDERL [A6] definierten externen Schnittstellen von CAX¹-Programmen gemeint. Sie dienen dazu, einen Informationsaustausch zwischen dem CAX-System und seiner Umwelt beziehungsweise einem anderen CAX-System durchzuführen. Hiermit sind nach ANDERL et al. [A8] folgende Aufgaben verbunden:

- Informationsaustausch (bei Produktmodellen die Informationsabbildung)
- Konsistente Abbildung externer Informationen auf systeminterne Schnittstellen
- Benutzerfreundliche Anwendung
- Fehlersicherheit und -behebung
- Anpassbarkeit an anwendungsspezifische Anforderungen

Im Allgemeinen kann zwischen systemspezifischen und systemneutralen Datenschnittstellen unterschieden werden, die sich jeweils nur auf die Geometrie oder auf die anfallenden Daten des gesamten CAE-Prozesses beziehen können.

¹ CAX ist ein Sammelbegriff für alle CA(Computer-Aided)-Technologien.

3.2.1 Systemspezifische und quasineutrale Datenformate zur Geometrieübertragung

Die Realisierung des Datenaustausches zwischen zwei Systemen zur Geometriemodellierung wird durch Pre- und Postprozessoren realisiert, welche die Daten direkt in das jeweils andere Format schreiben bzw. es aus einem fremden, nativen Format auslesen [A6]. Gerade im Bereich des Datenaustausches zwischen komplexen Bauteilen, die in 3D konstruiert wurden, hat diese Art des Datenaustausches eine hohe Bedeutung. Mit speziell für bestimmte Systemversionen entwickelten Prozessoren lässt sich die Fehleranfälligkeit minimieren, da die Prozessoren nur für jeweils zwei Partnersysteme funktionsfähig sein müssen.

Unter quasineutralem Datenformat wird der Datenaustausch über den gleichen 3D-Modellierungskern von CAD-Systemen verstanden. Alle gängigen CAD-Systeme stützen sich zur 3D-Modellierung auf einen von derzeit drei verfügbaren Modellierungskernen (ACIS, Parasolid und Design Base). Nach SENDLER [S10] hat mit über 100 gängigen CAD-Anbietern ACIS die größte Bedeutung einer quasineutralen Schnittstelle. Als Nachteil der Verwendung einer quasineutralen Schnittstelle ist zu nennen, dass ein Datenaustausch mit anderen 3D-Modellierungskernen nicht üblich ist und deshalb auch keine Prozessoren angeboten werden. Bei Verwendung eines 3D-Modellierungskerns als Datenschnittstelle wird deshalb nur ein Teil der CAD-Systeme erfasst.

3.2.2 Systemneutrale Datenformate zur Geometrieübertragung

Für eine systemneutrale Geometrieübertragung gibt es die in Tabelle 3-1 gegenübergestellten Datenschnittstellen. ANDERL [A6] gibt einen Überblick über die wichtigsten CAD-Schnittstellen. Unter den vielen Verfügbaren wird im 2D-Bereich der Schnittstelle DXF, im 3D-Bereich den Schnittstellen STEP, IGES und VDAFS [K1] die größte Bedeutung zuteil. Die Intention von STEP ist es, nach und nach alle anderen Schnittstellenformate abzulösen. Dies zeigt sich nicht nur an wachsenden Implementierungen von STEP, sondern auch an dem „Einfrieren“ bestehender Versionen der Schnittstellen aus Tabelle 3-1. Dass seit Jahren keine neuen Versionen der anderen Schnittstellen herausgegeben werden, liegt nicht an deren Ausgereiftheit, sondern an der Tatsache, dass die Firmen ihre Bemühungen in die Integration von STEP konzentrieren. Die Bedeutung der anderen Schnittstellen ist somit nur noch Stand der Technik bei bestehenden und bewährten Abläufen in den Unternehmen. Für eine Neuorientierung von Arbeitsabläufen ist nur STEP von Bedeutung, da aufgrund des Produktmodell-Charakters von STEP nicht nur Geometriedaten sondern auch eine Vielzahl anderer Daten übertragen werden können. Diese Eigenschaft existiert nur bei STEP. Andere, einfachere Ansätze zur Übertragung von Produktdaten werden in Kapitel 3.2.3 beschrieben. Die zusätzliche Übertragung von Produktdaten im Vergleich zu einer rein auf Geometriedaten spezialisierten Schnittstelle ergibt insbesondere bei einem Datenfluss über die reinen CAD-Systeme hinaus Vorteile, da auch andere Programme und Systeme auf die Daten zugreifen können.

		IGES	SET	VDAFS	PDDI	VDAIS	CAD*I	DXF	STEP AP201	STEP AP214
Herkunft	erster Vorschlag	1979	1984	1983	1984	1986	1985	1986	1989	1989
	Standardisierung / Normierung	1981	1986	1983	1989	1989	1985	1986	1996	2001
	Herkunftsland und verantwortliche Behörde / Unternehmung	USA, NIST	Frankreich, Aeros	Deutschland, VDA	USA, NIST	Deutschland, VDA	Europa, ESPRIT	USA, Fa. Autodesk	USA, ISO	USA, ISO, Deutschland, VDA, ProSTEP
3D-Geometriefunktionen	Matrizen, Vektoren	+	+	o	+	+	+	o	o	+
	Punkte, Kanten	+	+	+	+	+	+	+	+	+
	analytische Kurven (z.B. Kreis)	+	+	-	+	+	+	+	+	+
	Freiformflächen (z.B. B-Spline)	o	+	+	+	+	+	+	+	+
	analytische Flächen (z.B. Ebenen)	+	+	-	+	+	+	+	+	+
	topologische Elemente (z.B. Eckpunkt.)	-	-	-	+	-	+	-	+	+
	Volumenprimitive (z.B. Quader)	+	+	-	-	-	+	+	-	+
	B-Rep Modelle	+	+	-	-	-	+	o	-	+
	CSG-Operationen	+	-	-	-	-	+	o	-	+
	Produktbeschreibende Features	-	-	-	-	-	-	-	-	o
einfache Parametrik	o	-	-	-	o	o	-	-	o	
2D-Zeichnungen	Zeichnung, Zeichnungsinformationen	+	+	-	-	+	o	+	+	+
	Ansichten	+	+	-	o	+	+	+	+	+
	Texte, Zeichensätze	+	+	-	o	+	+	o	+	+
	Symbole, Teilbilder	+	+	-	o	+	+	+	+	+
	Maßbilder, Schraffuren	+	+	-	-	+	+	+	+	+
grafische Darstellung (Farbe, Layer)	+	+	-	o	+	+	+	+	+	
Assoziativitäten / Relationen	Geometrie-Grafik	+	+	-	+	+	o	-	-	+
	Geometrie-Zeichnungselemente	-	-	-	-	-	o	-	-	+
	Geometrie-Fertigung	-	-	-	o	-	o	-	-	+
	Ebenen, Sichtbarkeit	+	+	-	+	+	+	+	+	+
	Gruppierung	+	+	+	+	+	+	-	-	+
	definierbare Assoziativitäten	+	+	-	+	+	+	-	-	+
	definierbare Eigenschaften, Properties	+	+	-	+	+	+	-	-	+
externe Referenzen	-	+	-	+	-	-	+	o	+	
Anwendung/Technologie	Toleranzen	-	-	-	+	-	-	-	+	+
	technische Angaben (z.B. Material)	-	-	-	+	-	-	o	+	+
	technische Elemente (z.B. Flansch)	-	-	-	+	-	-	-	-	+
	organisator. Angaben (z.B. Stückliste)	-	-	-	o	-	-	-	o	+
	Fertigungsinformationen (z.B. NC)	-	-	-	+	-	-	-	-	+
	FEM	+	+	-	-	-	+	-	-	+
	Kinematik	+	-	-	-	-	-	-	-	+
	mechanische Konstruktion	+	+	+	+	+	+	+	-	+
Elektrik, Elektronik	+	-	-	-	-	-	o	-	o	
Definition / Interpretation	Dateiinformatoren	+	+	+	+	+	+	+	+	+
	Voreinstellungen	+	+	-	+	+	+	-	-	+
	Interpretationsregeln	o	o	-	+	-	o	-	-	+
	benutzerspezifische Elemente	o	+	-	+	-	o	-	-	+
	Elementschema, -definition	-	-	-	+	-	-	-	-	+
	Alternative Elementrepräsentation	-	+	-	+	-	-	-	-	+

"+" berücksichtigt, "o" teilweise berücksichtigt, "-" nicht berücksichtigt

Tabelle 3-1: Gegenüberstellung von Datenschnittstellen

3.2.3 Datenformate zur Übertragung allgemeiner Produktdaten

Die systemneutrale Übertragung allgemeiner Produktdaten wird auf der Ebene eines Produktmodells erst durch STEP möglich. Die Beherrschung dieses umfassenden Werkzeugs und die erst in 2001 abgeschlossene Standardisierung der für den Maschinenbau relevanten Bereiche von STEP hat in der Vergangenheit viele Anwender dazu bewegt, eigene Datenaustauschformate zu definieren.

GRAVEL&POMMER [G13] beschreiben das Gear-Data-Exchange Format (GDE) für Verzahnungsdaten zur Anwendung in der Mess- und Prüftechnik von Zahnradern. Ziel ist der Datenaustausch zwischen Prüfmaschinen untereinander und zwischen Zahnradhersteller und Dienstleister in diesen Bereichen. Derzeitige Bemühungen dieses Schnittstellenformats gehen in Richtung XML¹ und werden in einem VDI-Arbeitskreis behandelt.

CHONG et al. stellen in Ihren Veröffentlichungen [C1] [C2] auch eine XML-Schnittstelle für den internetbasierten Austausch von Verzahnungsdaten vor.

Als Nachteil bei der Anwendung von XML ist die fehlende Standardisierung zu nennen. Da sich XML aus dem sich schnell verändernden Internet-Umfeld heraus entwickelt hat, kann sie langfristig nicht in Konkurrenz mit der STEP-Schnittstelle treten, die einheitlich und international als ISO-Norm festgeschrieben ist. Zudem stellt XML kein Produktmodell, sondern lediglich eine formale Spezifikation der Daten dar.

Zur Realisierung eines einheitlichen Datenaustausches in der Automobilindustrie hat der Verband der Automobilindustrie (VDA) die Empfehlung für den Datenaustausch mit ENGDAT (Engineering-Data-Message) europaweit ausgesprochen. Die ENGDAT-Richtlinie [V10] stützt sich auf die VDA-Empfehlung [V9] für den CAD/CAM Datenaustausch und beschreibt Form und Inhalt der ausgetauschten Informationen. ENGDAT stellt keine Schnittstelle zwischen Systemen dar, sondern kann als gebündeltes Paket von unterschiedlichen Dateien zu einer Anwendung gesehen werden, die gemeinsam verschickt und auf einem „Paketzettel“ protokolliert werden [E4][K5]. Es ist zum Beispiel denkbar, dass die Freiformgeometrie eines Bauteils als VDAFS-Datei mit einer IGES-Datei gepaart wird, welche ergänzende B-Rep Volumenmodelle des selben Bauteils enthält.

Bis auf STEP stützen sich die genannten Datenformate und Übertragungsmethoden auf konkrete Anwendungsbereiche, aus denen sie auch hervorgegangen sind. Eine Übertragung auf andere Produkte oder eine neutrale Darstellung als Referenz ist nicht möglich. Dieser Anforderung kommt nur STEP nach. Doch auch mit dem Einsatz von STEP existieren noch keine konkreten Anwendungen, sondern lediglich prototypische Umsetzungen [D4]. Eine Herausforderung besteht demnach darin, die Möglichkeiten mit STEP an einem konkreten Produkt anzuwenden und deren Leistungsfähigkeit weit über die Geometrie hinaus nachzuweisen.

¹ XML ist die Abkürzung der „Extensible Markup Language“ und ist ein Internet-basiertes, universelles Format, um Daten und Dokumente abzubilden und über das Internet auszutauschen.

3.2.4 Innerhalb der vorliegenden Arbeit verwendete Datenformate

Im dreidimensionalen Bereich und zur Produktmodellierung kommt das STEP AP214 [I5] zur Anwendung, welches aufgrund seiner Komplexität und flexiblen Anwendbarkeit ausführlich in Kapitel 3.6 vorgestellt wird.

Den zweidimensionalen Schnittstellen kommt in der vorliegenden Arbeit nur eine geringe Bedeutung zu, da sie ausschließlich für die Erstellung von Fertigungszeichnungen und Datenblättern eingesetzt werden. Dieser Vorgang ist über das DXF- und STEP AP201-Format möglich.

Nach JÄGER [J1] spielt das Data-Exchange-Format (DXF) im 2D-Bereich die Rolle eines Industriestandards beim Austausch von Geometriedaten. Die für die Übertragung zweidimensionaler Zeichnungen bedeutendste Schnittstelle [S8] wurde ursprünglich von der Firma Autodesk zur Verbindung der hauseigenen Produkte festgelegt. Aufgrund deren weiten Verbreitung kamen Hersteller anderer CAD-Systeme nicht umhin, auch eine DXF-Schnittstelle anzubieten. Die breite Anwendung im 2D-Bereich wird auch durch die Tatsache unterstrichen, dass Office- und DTP¹-Programme mit DXF-Schnittstellen ausgerüstet sind. Der Funktionsumfang von DXF beschränkt sich auf die Beschreibung der Zeichnungselemente innerhalb einer zweidimensionalen Zeichnung.

Die Möglichkeiten mit dem für den Datenaustausch von zweidimensionalen Zeichnungen eingerichteten STEP AP201 [I3] gehen darüber hinaus. Es spezifiziert ein STEP-Anwendungsprotokoll (Kapitel 3.6.1.3) zum Austausch zweidimensionaler technischer CAD-Zeichnungen und definiert die nominelle Gestalt eines Produkts durch zweidimensionale Geometrie und Annotation [G9]. Der Inhalt des Produktreferenzmodells von STEP AP201 und der Überschneidungsbereich mit DXF ist in Bild 3-1 dargestellt. GRABOWSKI et al. [G9] fassen den Inhalt des AP201 folgendermaßen zusammen:

- Repräsentation technischer Zeichnungen mit einem Maßstab.
- Hierarchische Strukturierung von Zeichnungen, Zeichnungsblättern und Zeichnungsansichten.
- Trennung des zweidimensionalen Gestaltmodells von dessen geometrischen Beschreibungen in der 2D Geometrie.
- Darstellung von Bemerkungen zu den Elementen des Produktreferenzmodells und administrativen Daten des Produkts.
- Elemente zur Zeichnungsverwaltung und Versionierung.

¹ DTP steht für „Desktop Publishing“ und bezeichnet die Erstellung von Präsentationsgrafiken

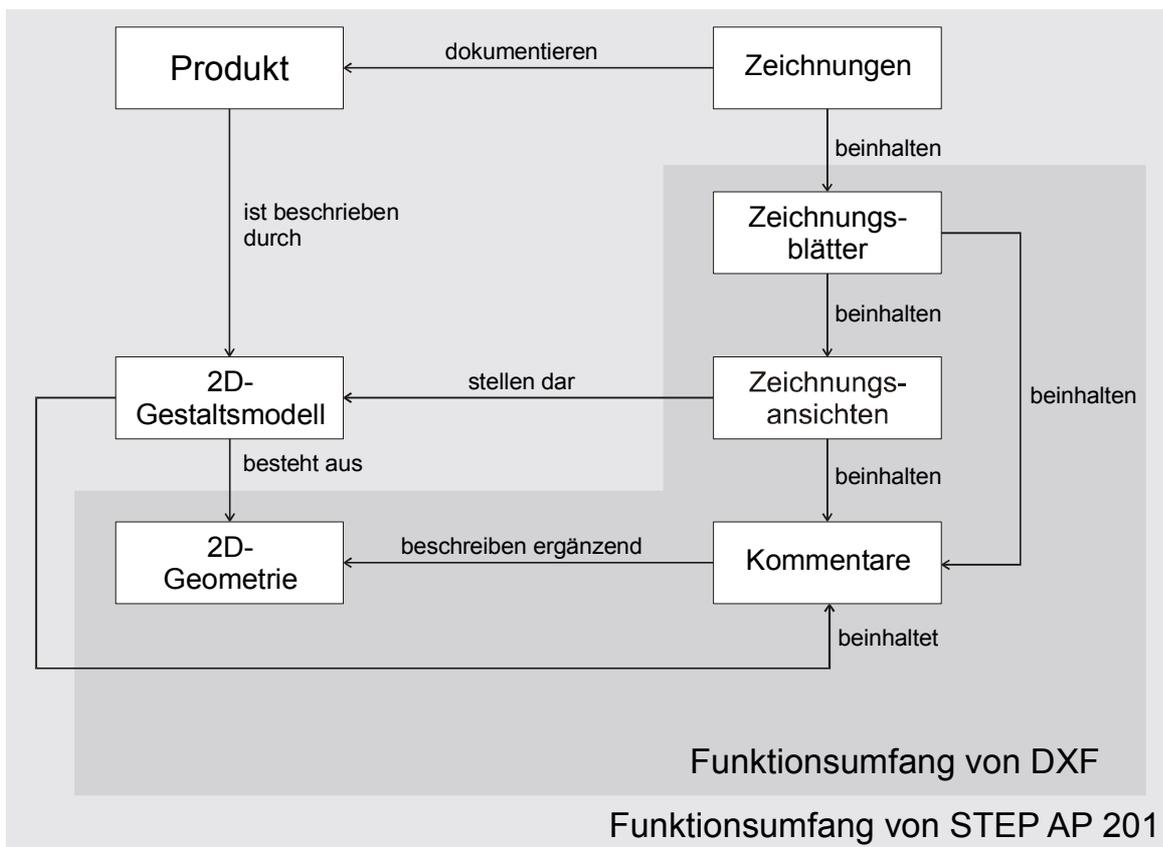


Bild 3-1: Inhalt des Produktpreferenzmodells STEP AP201 und der Überschneidungsbereich mit DXF

3.3 Rechnerintegrierte-, rechnerunterstützte- und programmverbindende Systeme

Eine Verkürzung der Entwicklungszeit gekoppelt mit einer Qualitätssteigerung kann innerhalb der rechnerunterstützten Produktentwicklung nur dann erreicht werden, wenn die beteiligten Programme und Systeme über die Geometriedaten hinaus eng und unmittelbar zusammenarbeiten. Für die Integration der Programme und Systeme haben sich unterschiedliche rechnerintegrierte, rechnerunterstützte und programmverbindende Systeme entwickelt, die nachfolgend strukturiert dargestellt werden.

Eine Unterscheidung kann nach der Art der Einbindung des Gestaltungsprozesses erfolgen. Folgende Möglichkeiten lassen sich, sortiert nach einem steigenden Aufwand für die Systemerstellung und -pflege, nennen:

- **Kein CAD-System eingebunden.** Damit sind in der Regel Programmsysteme gemeint, welche gekoppelte Programme aus der Berechnung/Simulation beinhalten und ausschließlich mit Ersatzgeometrie innerhalb der Programme arbeiten.

- **CAD-System nur zur Darstellung.** Das CAD-System wird nur zur Darstellung der Geometrie verwendet und unidirektional angesprochen. Ein Rückfluss der Geometrie aus dem CAD-System zurück in andere Programme und Systeme erfolgt nicht.
- **Vollständige Verschmelzung von CAD-System mit anderen Programmen/Systemen.** Damit ist in der Regel eine Erweiterung eines bestimmten CAD-Systems um Aufgaben aus der Berechnung/Simulation verbunden. Es entsteht ein nach [G3] eng gekoppeltes monolithisches Gesamtsystem.
- **Bidirektionale Verknüpfung von CAD-System und Programmen/Systemen über Schnittstellen.** Ein bidirektionaler Datenaustausch beschreibt die vollständige Integration eines CAD-Systems in den kreisförmig durchlaufenen Entwicklungsprozess unter einer vollständigen Separierung von CAD-System und beteiligten Programmen. Bei diesem nach [G3] indirekt gekoppelten CAE-Mechanismus erfolgt der Datenaustausch über Dateien oder ein gemeinsames Produktmodell.

Eine weitere Differenzierung der beschriebenen Ansätze ist hinsichtlich der Systemneutralität zum Zeitpunkt des Datenaustausches möglich:

- **Verwendung nativer Schnittstellen oder der Anwenderschnittstelle (API).** Durch eine Systemanpassung auf ein bestimmtes Anwendungssystem kann sich der Aufwand zur Systemerstellung verringern und ein größerer Leistungsbereich abgedeckt werden, da alle Möglichkeiten des entsprechenden Systems zur Verfügung stehen.
- **Verwendung systemneutraler Schnittstellen.** Dies bedeutet eine Einschränkung der Leistungsfähigkeit in der Datenschnittstelle als kleinster gemeinsamer Nenner der denkbaren Systeme und Ersatzsysteme. Auf der anderen Seite werden mehr verschiedene Programme und Systeme abgedeckt, was einer universelleren Verwendung entgegenkommt. Die Qualität der Datenschnittstelle hat einen maßgeblichen Einfluss auf die Möglichkeiten der Nutzung.

Im Folgenden wird gemäß MERTENS [M8] nach der Methode der Systemerstellung unterschieden in:

- **Methodenorientierte Ansätze** (Kapitel 3.3.1), die eine flexible und allgemeine Anwendung der Ergebnisse ermöglichen.
- **Objektorientierte Ansätze** (Kapitel 3.3.2), die zunächst konkrete Produkte oder Verfahren zur exemplarischen Entwicklung von Methoden heranziehen, um die Ergebnisse anschließend zu verallgemeinern.

3.3.1 Methodenorientierte Ansätze

Zu den methodenorientierten Ansätzen gehören Referenzmodelle und -architekturen, die für bestimmte Bereiche vorgeschlagen werden. Am bekanntesten ist das CAD-Referenzmodell [A1][K7][S13], welches eine offene, modulare und flexible Systemarchitektur zur Förderung eines integrierten Produktmodellierungsprozesses darstellt und welches die Grundlage für

viele andere integrierte Systemarchitekturen ist. Das von GAUSEMEIER et al. [G3] eingeführte KOMFORCE¹-Referenzmodell stellt ein vier-Ebenen-Modell dar und verspricht eine allgemeine Kopplung von beliebigen CAE-Werkzeugen durch „Intelligente Objekte“ unter Nutzung eines gemeinsamen STEP-Produktmodells. Das sehr allgemein gehaltene KOMFORCE-Referenzmodell dient für viele andere Rechnersysteme als Vorlage.

Tabelle 3-2 gibt einen Überblick über verfügbare Rechnersysteme, die methodenorientierte Vorgehensweisen zur Integration der Subsysteme beinhalten, und deren Beispielanwendungen.

Initiator	Bezeichnung und Beispielanwendung	Einbindung der Gestaltung	Systemneutralität	Literatur
Beitz	Durchgängig flexibles Konstruktionssystem	kein CAD-System eingebunden	nein	FELDHUSEN [F1]
Dietz	rechnergestütztes Beratungssystem zur Konstruktion lärmarmen Produkte	Erweiterung eines CAD-Systems	nein	DIETZ et al. [D1]
Ehrlenspiel Lindemann	INKA – Integrierter Konstruktionsarbeitsplatz	Integration von Pro/Engineer	STEP-Ausgabe möglich	EHRENSPIEL [E1] EHRENSPIEL et al. [E2] LINDEMANN&AMFT [L5] AMBROSY [A4]
Franke	MOSAIK – Modulare Systemarchitektur zur Integration von Konstruktionswerkzeugen. Beispiel: Wälzlager und Wellen	kein CAD-System eingebunden	nur auf Systemebene (RPC, CORBA, OLE)	PETERS [P1]
Gausemeier	CORSICA – integriertes Ingenieursystem	Integration versch. CAD-Systeme	STEP-Ausgabe möglich	HAHN [H4]
Klose	Entwurfssystem zur Unterstützung früher Phasen durch funktional-technische Objektmodellierung	Erweiterung von EMS (Intergraph)	nein	KLOSE et al. [K8] STEGER [S18]
Meerkamm	mfk – Konstruktionsassistenzsystem; Beispiel: Antriebsstrang, PKW-Differential	Integration von Pro/Engineer	STEP-Ausgabe möglich	KOCH&MEERKAMM [K9] MEERKAMM et al. [M6] WARTZACK&MEERKAMM [W1]
Mertens	Berechnungskompetenzzentrum	über API beliebige CAD-Systeme	STEP-Ausgabe möglich	MERTENS [M8] WÖLFLE [W17]
Weck	Ingenieurgerechter Konstruktionsarbeitsplatz; Beispiel: Wellenende mit Pressverband	Erweiterung von CATIA	nein	REPETZKI [R4] WECK et al. [W6]
Welp	INPRO – objektorientierter Integrationsprozess	Integration von I-DEAS	nein	WELP&BRAUN [W11]

Tabelle 3-2: Verfügbare Rechnersysteme des methodenorientierten Ansatzes zur Verknüpfung von Berechnung und Gestaltung

¹ KOMFORCE bezeichnet den „Kommunikations- und Forschungskreis für Integrationstechnologien in Computer Aided Design und Engineering“.

3.3.2 Objektorientierte Ansätze

Unter „objektorientiert“ wird in diesem Zusammenhang die Erstellung von Systemen verstanden, die sich zunächst auf ein konkretes Produkt des Maschinen- und Anlagenbaus beziehen und im Anschluss eine Übertragung der Ergebnisse auf andere Anwendungsfälle zulassen.

Tabelle 3-3 gibt einen Überblick über derzeit verfügbare Rechnersysteme und deren Basisanwendungen. Neben den wissenschaftlichen Ansätzen existieren zum Teil auch in den Unternehmen integrierte Systeme. Zum Beispiel bietet ein Wälzlagerhersteller ein Online-Beratungs- und Berechnungszentrum mit CAD-Schnittstelle für Wälzlager an [L7][L8][S1]. HAASIS [H3] stellt weiterhin die Konstruktion von Getrieben mit einem speziell angepassten CAD-System vor.

Initiator	Bezeichnung und Basisanwendung	Einbindung der Gestaltung	Systemneutralität	Literatur
Brüser	Konstruktionssystem für Planetengetriebe	Erweiterung von EUCLID	nein	KUNTZ [K18]
Feldmann Geiger	Integrierte Ingenieurumgebung für den Entwurf mechatronischer Produkte	über ACIS Modellkern	STEP	KREBS [K15]
Franke	Entwurfsumgebung für Mikrokomponenten	über STEP Schnittstelle	STEP	GERMER et al. [G7]
Gausemeier	EUMECH - Entwicklungsumgebungen Mechatronik; wissenschaftliche Referenzarchitektur	vielfach möglich	u.a. STEP	GAUSEMEIER&LÜCKEL [G5]
Günthner	Datenbankbasiertes Programmsystem für Gittermast-Fahrzeugkrane	nur Ersatzgeometrie	nein	SCHRÖDER [S6]
Höhn	Verteiltes Softwaresystem zum Entwerfen, Gestalten und Nachrechnen von Zahnradgetrieben	über STEP AP214	STEP AP214 kompatibel	HÖHN et al. [H9][H10][H13]
Leidich	Internet-basiertes Programmsystem für Kurvengetriebe	über STEP	STEP und VRML	KERTSCHER&LEIDICH [K4]
Mewes	Konstruktionsumgebung für die Entwicklung von Kalandern	Erweiterung von Pro/Engineer	nein	RIEST [R5] RIEST&MEWES [R6]
Predki	Rechnersoftware für Planetengetriebe	Erweiterung von Unigraphics	nein	CHRIST [C3] PREDKI&CHRIST [P4]
Spur	„Kisssoft“ Prototyp für Stirnradberechnung	im Berechnungsprogramm integriert	nein	KISSLING&ZIEBEIL [K6] SPUR&ZIEBEIL [S14] WESER [W13]
Strohmeier Vajna	Konstruktionssystem für integrierte Apparatekonstruktion	über STEP am Beispiel Pro/Engineer	STEP	JANDELEIT et al. [J2]
Winter Höhn	STIPSY, GETSY – Getriebeprogrammsysteme für Zahnradgetriebe	nur Ersatzgeometrie	nein	RUDZEWSKI [R8] WINTER et al. [W14] [W15]

Tabelle 3-3: Verfügbare Rechnersysteme des objektorientierten Ansatzes zur Verknüpfung von Berechnung und Gestaltung

Die objektorientierten Ansätze haben sich größtenteils aus industrienahen Anwendungsfällen mit konkreten Beispielen ergeben. THEIßEN [T1][T2] beschreibt den konventionellen, praxisbewährten Ansatz zur Integration von Berechnung und Gestaltung von Großgetrieben bei einem mittelständischen Getriebehersteller. Die Vorgehensweise kann zum Beispiel die Ausgangsbasis für integrierte Rechnerarbeitsplätze sein. In der Vergangenheit wurden an der

FZG¹ zwei Getriebeprogrammsysteme entwickelt, die es sich zum Ziel gesetzt haben, Berechnungsprogramme datentechnisch zu koppeln und den Benutzer durch die systematische Anwendung der Berechnungsprogramme im Problemlösungszyklus Getriebe zu führen. Diese beiden Projekte STIPSY (Stirnradprogrammssystem [W15]) und GETSY (Getriebeprogrammsystem [R8]) waren durch eine einheitliche Datenbasis und Oberfläche gekennzeichnet.

Die im Sonderforschungsbereich 336 (SFB) [L1] entwickelten Methoden und Werkzeuge für die integrierte Produktentwicklung wurden im Transferbereich 2 (TFB) in zwei Projekten bei fünf Unternehmen auf Praxistauglichkeit geprüft und dahingehend optimiert. Ziel war eine Umsetzung in ein firmenspezifisches System zur Parallelisierung der Aufgaben durch eine simultane Bearbeitung und eine Eigenschaftsfrüherkennung, das heißt eine Wissensverlagerung von der Montageplanung in die Konstruktion [L2]. Dies sollte nach BENDER et al. [B6][B7] durch eine bereichsübergreifende Zusammenfassung der zahlreich in den Unternehmen vorliegenden Insellösungen in eine „integrierte Engineering-Umgebung“ (iEU) erreicht werden. Als Ergebnisse wurden in allen fünf Unternehmen bei den relativen Zielgrößen wie Transparenz der Prozesse, methodische Unterstützung, umfassender Methodeneinsatz und verbesserte Abstimmungsprozesse ausschlaggebende Verbesserungen erzielt. Dies ist vor allem auf die integrierte Vorgehensweise der Systeme in den Unternehmen zurückzuführen.

3.3.3 Bewertung und Bedeutung vorhandener Systeme

Bei den meisten rechnerintegrierten, rechnerunterstützten und programmverbindenden Systemen handelt es sich entweder um Referenzsysteme, die so allgemein gehalten sind, dass eine konkrete Anwendung viel Anpassungsarbeit bedarf, oder um prototypische Forschungsansätze für die Entwicklung eines bestimmten Produkts. Die Systeme unterscheiden sich in ihren Reifegraden und stellen oftmals nur Ansätze im Konzeptstadium dar. Bei bereits im Prototypenstadium verwendbaren Systemen lassen sich folgende Diskrepanzen erkennen:

- Viele Systeme sind nur eine Erweiterung bestimmter CAD-Systeme um Berechnungs- und Simulationsfunktionen, was keine Innovation mehr darstellt.
- Da im Rahmen der Forschungsarbeiten der Gesamtsysteme auch Programme für die Berechnung und Simulation entwickelt worden sind, ist eine Industrietauglichkeit mangels Implementierungen und Testläufen oft nicht ohne weitere Bemühungen und einer firmenspezifischen Anpassung möglich.
- Eine wahre Systemneutralität ist bei vielen Systemen nicht gegeben. Viele Systementwickler verstehen unter „Systemneutralität“ nur die Möglichkeit, bestehende Datenschnittstellen, z.B. der CAD-Systeme, zu nutzen, um Daten systemneutral auszugeben. Da sich dies jedoch nur auf die Geometrie bezieht und in vielen Systemen keine systemneutralen Produktmodelle erzeugt werden, liegt keine wahre Systemneutralität vor.
- Die Übertragbarkeit und Anpassung der Systeme auf andere Produkte oder Produktparten ist nur umständlich möglich.

¹ FZG ist die Abkürzung der Forschungsstelle für Zahnräder und Getriebbau der Technischen Universität München.

- Die Integration und Kommunikation mit anderen Programmen und Systemen der vorhandenen informationstechnischen Umgebung ist nicht oder nur eingeschränkt möglich.
- Die geringe Übersichtlichkeit, Systemsteuerung und Benutzungsoberfläche gestatten keine intuitive Bedienung der Systeme.

3.4 Produkt- und Prozessmodelle

Durch modellhafte Abbildungen realer Systeme können komplexe Zusammenhänge leicht veranschaulicht werden. Dadurch wird ein Systemdenken unterstützt. Durch eine Abstraktion wird der für das jeweilige Anwendungsgebiet notwendige Informationsbedarf abgedeckt.

3.4.1 Grundlagen und Begriffe

Ein **Modell** stellt ein Abbild eines Produkts, Systems oder Vorgangs dar. Da je nach Anwendungszweck des Modells unterschiedliche Detaillierungsgrade angebracht sind, ist der Abbildungsprozess vom Modell zum Original nicht eindeutig. Folgende Merkmale können nach STACHOWIAK [S16] allgemeinen Modellen zugeordnet und auch auf digitale Produktmodelle übertragen werden:

- **Verkürzungsmerkmal:** Das Verkürzungsmerkmal bestimmt die abzubildenden Informationsanteile. Je nach den aus Entwicklersicht benötigten Eigenschaften des Originals werden Produktmodelle mehr oder weniger genau im Rechner abgebildet.
- **Abbildungsmerkmal:** Durch das Abbildungsmerkmal wird die Repräsentation der abgebildeten Produktinformationen festgelegt und auf eine rechnerinterne Ablage überführt.
- **Pragmatisches Merkmal:** Die Verbindung zwischen Original und Modell ist abhängig vom Nutzer, dem Zeitintervall sowie der Einschränkung auf bestimmte gedankliche bzw. tatsächliche Operationen. Das pragmatische Merkmal kommt demnach der Beschreibung des Anwendungszusammenhangs entgegen.

Für die Modellierung der digitalen Produktentwicklung ist zu beachten, dass diese aus zwei zusammenhängenden Teilen, dem Produkt- und Prozessmodell, besteht, die in enger Wechselwirkung zueinander stehen.

Ein **Prozess** ist eine logische Abfolge von Tätigkeiten und Aktivitäten, die durch inhaltliche Zusammenhänge verbunden sind. Informationen und Entscheidungen wirken auf den Prozess ein. Die Erreichung eines eindeutigen Ziels mit einem definierten Ergebnis ist das Ende eines Prozesses.

Nach AMBROSY [A4] beschreiben **Prozessmodelle** die dynamischen Aspekte eines Systems mit den zugrundeliegenden Methoden und werden zur Abbildung, Simulation und Analyse verwendet. Ein Prozessmodell kann ein, mehrere oder versionierte Produktmodelle enthalten.

Dem Ansatz des **Produktmodells** (auch integriertes Produktmodell genannt) liegen folgende spezialisierte Anforderungen zu Grunde:

- Abbildung von Produktdaten (organisatorisch, technisch und technologisch) aller Produktlebensphasen.
- Vereinigung von verschiedenen physikalischen und technischen Produkteigenschaften.
- Verschiedene Sichtweisen sind je nach Anwendungsgebiet möglich. Ziel ist es, den Blickwinkel verschiedener Anwender des Produktmodells in allen Produktlebensphasen möglichst genau zu treffen [G9].

Für die vorliegende Arbeit wird die Definition eines Produktmodells nach WASMER [W2] herangezogen: Dort wird ein Produktmodell als Informationsmodell verstanden, welches alle im jeweiligen Kontext relevanten Informationen eines Produkts abbildet. Es ist aus formal beschriebenen Modellschemata, inklusive Definitionen der beschriebenen Sachverhalte, aufgebaut. Dadurch werden Datenstrukturen festgelegt, mit denen individuelle Produkte im Rechner abgebildet werden können. Im Folgenden wird auf die begriffliche Detaillierung „digitales“ oder „integriertes“ Produktmodell verzichtet und unter einem „Produktmodell“ stets ein im Rechner abgebildetes Modell eines Produkts oder Systems mit den oben genannten Eigenschaften verstanden.

Ein **Produktreferenzmodell** ist die logisch abstrahierte Abbildung aller möglichen Datenstrukturen eines Produkts oder einer Komponente während aller Prozesse und beinhaltet strukturelle und prozessbezogene Aspekte des Untersuchungsobjekts. RULAND [R9] umfasst das Referenzmodell als die allgemeingültige, logisch abstrahierte Darstellung sowohl funktionaler wie auch informationsbezogener Sachverhalte eines zu betrachtenden Bereichs. Ein Referenzmodell beinhaltet demnach keine Werte oder konkrete Daten. Erst bei der Instanzenbildung (s. u.) wird das Referenzmodell zu einem Produktmodell für eine bestimmte Baugruppe, wie z.B. ein definiertes Getriebe. Sinngemäß das Gleiche gilt auch für ein Prozessreferenzmodell. Im Folgenden wird unter einem **Referenzmodell** das nicht mit Instanzen versehene Produkt- oder Prozessmodell verstanden.

Unter **Instanzenbildung** wird die Ausprägung des Produktmodells mit konkreten Werten verstanden. Ein Produkt- oder Prozessreferenzmodell wird hierzu mit genauen Daten und gegebenenfalls Einheiten versehen, die einen bestimmten Anwendungsfall repräsentieren. Der sich durch eine Instanzenbildung ergebende Zusammenhang zwischen Realität, Modell, Produkt- und Produktreferenzmodell zeigt Bild 3-2 anhand eines Getriebes.

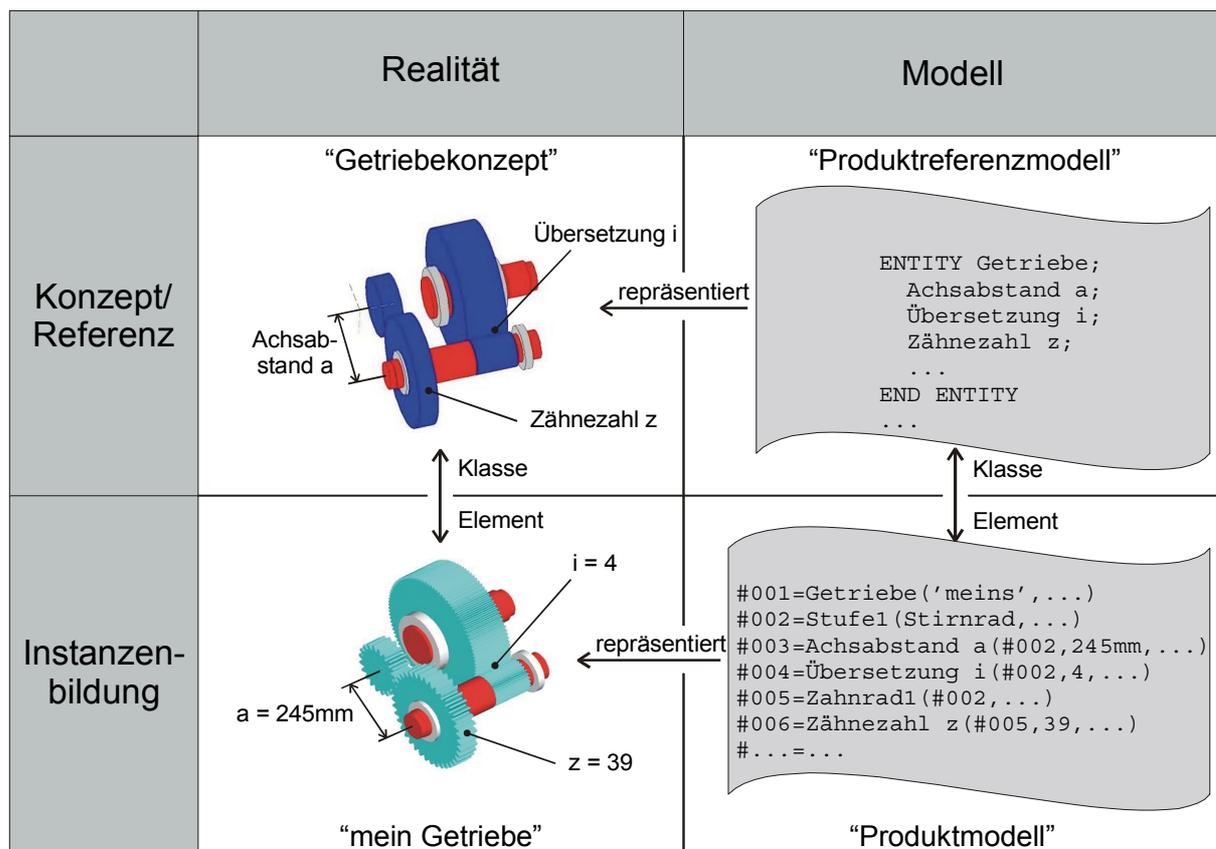


Bild 3-2: Zusammenhang zwischen Produkt- und Produktreferenzmodell bei der Instanzenbildung

3.4.2 Implementierung von Produkt- und Prozessmodellen

Nach MOHRMANN&SPECK [M10] stellt das Produktmodell eine Integrations- und Implementierungsplattform für die abzubildenden Prozessketten dar. Um ein Produktmodell operativ einsetzen zu können, wird es implementiert. Ein Programmsystem muss dann Produktdaten nach der Spezifikation im Produktreferenzmodell lesen, schreiben und manipulieren können [S4]. Zur Realisierung gibt es zwei Möglichkeiten:

- **Dateibasierter Datenaustausch** auf Basis eines Produktreferenzmodells über Prä- und Postprozessoren (nach STAUB [S17] auch Ex- beziehungsweise Importer genannt). Im Mittelpunkt steht das Datenaustauschformat, vergleiche Kapitel 3.2. Ein rechnerübergreifender Datenaustausch ist durch den Austausch von Dateien problemlos möglich. Die Definition der physikalischen Austauschdatei für STEP findet sich in [I4].
- **Gemeinsamer Datenzugriff auf eine gemeinsame Datenbank**, die auf dem Produktreferenzmodell basiert. Im Mittelpunkt steht eine Programmierschnittstelle zum Zugriff auf die Produktdaten. Bei Anwendung von STEP kommt das STEP-Data-Access-Interface (SDAI) [I6] zur Anwendung.

Die nachfolgenden Untersuchungen stützen sich aus folgenden Gründen auf einen dateibasierten Datenaustausch:

- Universellere und einfachere Handhabung
- Realisierung verteilter Systeme leichter möglich
- Eine Integration bestehender Schnittstellendateien vorhandener Programme und Systeme über Prozessoren sehr gut möglich

3.4.3 Methoden zur Prozessmodellierung

Nach BENDER [B6] haben sich in den letzten Jahren eine ganze Reihe von Modellierungsmethoden, wie z.B. die Architektur integrierter Informationssysteme (ARIS), die ereignisgesteuerte Prozesskette (EPK), die semantische Objektmodellierung (SOM), Structured Analysis and Design Technique (SADT) und objektorientierte Technologien wie die Unified Modeling Language (UML) etabliert.

Als wesentlicher Vertreter von Methoden zur grafischen Modellierung von Abläufen hat sich die SADT-Methode nach MARCA&GOWAN [M3] durchgesetzt. Aufgrund des hohen Bekanntheitsgrades wird sie beispielsweise auch zur Modellierung der Funktionsstrukturen von Produkten eingesetzt [E1].

Das Prinzip der SADT ist in Bild 3-3 dargestellt. Die Aktivitäten und Prozesse werden durch Rechtecke, die Ein- und Ausgangsinformationen als Pfeile dargestellt, die mit einer Aktivität oder einem Prozess in Verbindung stehen. Von oben kommen Steuerinformationen hinzu, die einen Ist- oder Sollzustand beschreiben, eine Aktivität auslösen oder koordinieren. Von unten wirken auf die Aktivitäten Methoden und Hilfsmittel ein, die zu deren Ausführung notwendig sind.

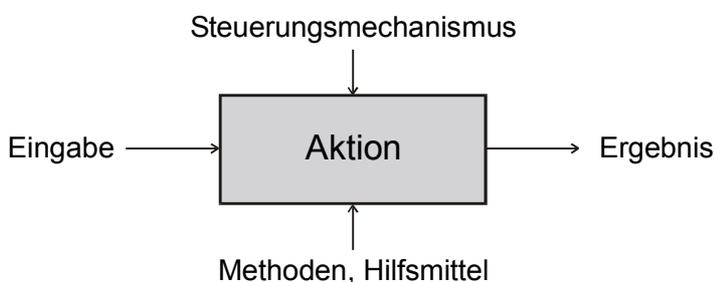


Bild 3-3: Structured Analysis and Design Technique (SADT)

In der vorliegenden Arbeit stellt die SADT-Methode die Basis zur Definition der Abläufe dar.

3.4.4 Beispiele für Produkt- und Prozessmodelle

In Tabelle 3-4 sind repräsentative Ansätze für verschiedene Arten von Produkt- und Prozessmodellen beschrieben, die im Rahmen unterschiedlicher Forschungsaktivitäten entwickelt worden sind.

Neben den wissenschaftlichen Ansätzen sind in den Unternehmen neben den Standards wie STEP auch eigens entwickelte Produktdatenmodelle im Einsatz. Zum Beispiel stellen BOPP

et al. [B9] ein sich bereits in Anwendung befindliches, prozessorientiertes Produktdatenmodell für Druckmaschinen vor.

Initiator	Bezeichnung und Basisanwendung	Einbindung der Gestaltung	Systemneutralität oder Referenzarchitektur	Literatur
Anderl	MechaSTEP – STEP Datenmodell zur Abbildung mechatronischer Systeme	über STEP-Basismodelle	neues STEP-AP	DONGES et al. [D4] GAUSEMEIER&LÜCKEL [G5]
Anderl	Das Produktmodell der ISO 10303 (STEP)	Definition der STEP-Basismodelle	STEP-Referenz	ANDERL&TRIPPNER [A12] ISO 10303 [I1]
Anderl	CoMET – Co-Operative Modelling Environment; Referenzarchitektur	nicht relevant	Referenzarchitektur	ANDERL&PÜTTER [A11]
Awizus Anderl	Abbildung der Schmiedeprozesskette mit STEP	über STEP-Basismodelle	STEP-Erweiterung	ANDERL et al. [A8] TRAUTHEIM [T6]
Bender	Referenzmodell zur integrationsgerechten Konzeption von Produktdatenmanagement	nicht relevant	Referenzarchitektur	WIRTZ [W16]
Ehrlenspiel Lindemann	Produktmodell für einen rechnerintegrierten Arbeitsplatz	integriertes CAD-System	CAD-Programmierschnittstelle	WELLNIAK [W10]
Ehrlenspiel Lindemann	AUTOSYS – rechnergestütztes Produktmodell für die PKW-Entwicklung	Informationssystem ohne Gestaltung	nicht notwendig	STEINMEIER [S23]
Grabowski	DICAD – Produktmodellansatz: Dialogorientiertes integriertes und intelligentes CAD-System	über die Referenzarchitektur	Referenzarchitektur zukünftiger CAD-Systeme	POLLY [P2] STAUB [S17]
Höhn	Empfehlung eines Produktmodells für Getriebe auf Basis von STEP AP214	über STEP-Basismodelle	Projektion auf das STEP AP214	DYLA&HÖHN [D5] DYLA et al. [D6] HÖHN et al. [H15]
Tönshoff	Datenmodell für systemneutralen Datenaustausch auf Basis technischer Elemente	angepasstes CAD-System Solid Works	Erweiterung von STEP	TÖNSHOFF et al. [T3][T4]
Warnecke	BAMOF – Bauteil-Modell-FBK; ein prozess- und integrationsgerechtes Produktmodell am Beispiel der Gießwerkzeugentwicklung	feste CAD-Anbindung	STEP-Schnittstelle geplant	RADTKE [R1]

Tabelle 3-4: Beispiele für ausgeführte Produkt- und Prozessdatenmodelle und Referenzmodelle

Eine Produktmodellierung ist die Voraussetzung für innovative rechnerunterstützte Entwicklungsprozesse. Zur Erstellung von Produktmodellen zeigt Tabelle 3-4 einerseits universell anwendbare Referenzmodelle und andererseits für bestimmte Anwendungen erstellte Produktmodelle, die in die jeweilige Systemumgebung implementiert sind.

Die Entwicklung von Prozess- und Prozessreferenzmodellen ist noch nicht so ausgereift, wie die der Produktmodelle. Dies bezieht sich insbesondere auf die Erfassung und Abbildung der Prozesse. Grund hierfür ist insbesondere, dass in erster Linie Produktmodelle unternehmensübergreifend ausgetauscht werden und dass sich Prozessmodelle aufgrund verschiedener Vorgehensweisen in den Unternehmen sehr voneinander unterscheiden.

3.5 Systeme für ein Produkt- und Prozessdatenmanagement

Mit der Kopplung von Gestaltung und Berechnung fällt mit der Handhabung der anfallenden Daten, Dateien und Dokumente ein zentrales Problem eines rechnerbasierten Programmsystems an.

Nach ANDERL&PHILIPP [A10] untersucht das Produktdatenmanagement den systematischen und effizienten Einsatz der Rechnertechnologie in allen Lebensphasen des Produkts mit dem wesentlichen Schwerpunkt in der Phase der Produktentwicklung. Zur Verwaltung dieser Daten haben sich aus den anfänglich für CAD-Stücklisten gedachten Engineering-Document-Management-Systemen (EDM) die Produktdatenmanagement-Systeme (PDM) entwickelt, die eine Integrationsbasis für die gesamte Prozesskette der Produktentwicklung darstellen [W16]. PDM-Systeme sind technische Datenbank- und Kommunikationssysteme, die dazu dienen, Informationen über Produkte und deren Entstehungsprozesse konsistent zu speichern, zu verwalten und transparent für alle relevanten Bereiche eines Unternehmens zur Verfügung zu stellen [V12].

Die unmittelbar mit dem Produkt- und Prozessdatenmanagement [G4] in Verbindung stehenden PDM-Systeme bilden nach EIGNER&ZAGEL [E5] die Voraussetzung für eine virtuelle Produktentwicklung. Für die vorliegende Arbeit sind sie insofern von Bedeutung, da sie ein beispielhaftes Mittel darstellen, um unterschiedliche Daten, Dateien und Dokumente digital zu handhaben. Die beinhalteten Methoden, Vorgehensweisen und insbesondere die Architektur von PDM-Systemen können für ein rechnerbasiertes Programmsystem von Interesse sein.

Bild 3-4 zeigt eine aus [V12], [S5] und [S13] zusammengestellte Systemarchitektur für PDM-Systeme. Diese unterteilt sich grob in anwendungsbezogene und systemübergreifende Funktionen. Schwerpunkt ist neben dem Datenspeicher und den darauf angewendeten Funktionen die Werkzeugumgebung zur Systemanpassung. Der größte Aufwand bei der Einführung von PDM-Systemen in Unternehmen besteht in der Anpassung („Customizing“) der PDM-Systeme an die firmenspezifische informationstechnische Umgebung und an die daran gekoppelten Prozessabläufe [A2][B1][W8].

BENDER et al. [B8] gehen mit ihren Engineering-Informationssystemen (EIS) [B6][B7] über den Inhalt des PDM-Gedankens hinaus und definieren den Begriff „KEIM – Kontinuierliches Engineering-Informationssystemmanagement im Product Life Cycle“ als einen ganzheitlichen integrierten Lösungsansatz, der Kooperationsprozesse mittels geeigneter Informationssysteme schneller, flexibler und effizienter unterstützen kann. Hierbei soll auch das Management der mentalen Veränderungsprozesse in den Denk- und Handlungsweisen der Mitarbeiter gefördert werden.

Einen neuen Ansatz für das Produktdaten- und Prozessmanagement stellt EIGNER [E4] mit dem „PDM²“-Gedanken vor. Unter „PDM hoch zwei“ versteht er „Product Development and Lifecycle Management“. Dies beschreibt ein Informationssystem, welches im Gegensatz zu früher die angezeigten Informationen nicht mehr vollständig physikalisch speichert, sondern aus betriebsspezifischen Fremdsystemen oder dem Internet referenziert. Durch Methoden eines durchgängigen Konfigurations-Managements wird mit „PDM²“ ein vollständiges Management aller Daten über die Lebensphase von Produkten erreicht.

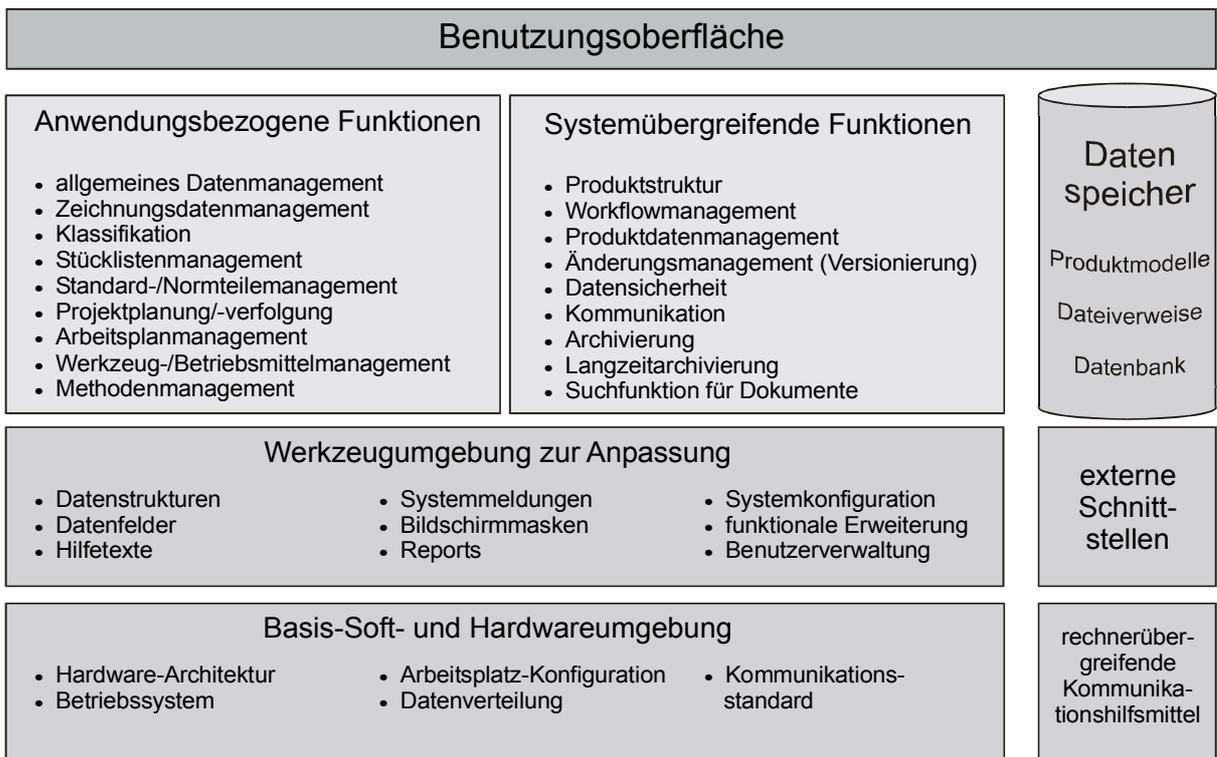


Bild 3-4: Typische Systemarchitektur eines PDM-Systems

TRAUTHEIM [T5] und MACHNER [M1][M2] stellen die Bedeutung von STEP für PDM-Systeme dar. Das Anwendungsprotokoll AP214 stellt schon jetzt bei einigen PDM-Systemen den datentechnischen Kern dar. Für STEP als Lösungsansatz im Bereich der PDM-Technologie lässt sich folgendes anführen:

- STEP beschreibt geometrische und allgemeine Produktdaten, welche auch alle notwendigen Daten der PDM-Systeme beinhalten.
- STEP trennt Inhalt (Produktdatenmodell) und Beschreibungsmittel (EXPRESS, siehe Kapitel 3.6.1.1) und macht damit die logische Beschreibung des Anforderungsmodells unabhängig von der Implementierungsmethode.
- STEP reflektiert international harmonisierte Anforderungen an ein kompatibles Produktdatenmanagement.
- Mit dem STEP-PDM-Schema existiert eine Anwendungsprotokoll-übergreifende Methode zur Abbildung von PDM-Daten.
- Durchgeführte PDM-Benchmarktests auf STEP-Basis zwingen die Systemanbieter zur Kompatibilität auf PDM-Ebene.

Die Bedeutung von STEP im PDM-Bereich und in Datenbankanwendungen [L11] unterstreicht die Verwendung eines STEP-Produktreferenzmodells für das zu erstellende rechnerbasierte Programmsystem.

3.6 Grundlagen von STEP

STEP (Standard for the Exchange of Product Model Data) ist ein Synonym für die Normenreihe ISO 10303 [I1]. Mit STEP wird das Ziel verfolgt, eine eindeutige, durch Rechner interpretierbare Darstellung aller Daten des Produktlebenszyklus eines Produkts zu ermöglichen [A12]. Bild 3-5 zeigt einen Überblick über den Inhalt der STEP-Norm mit den korrespondierenden Nummern der ISO 10303.

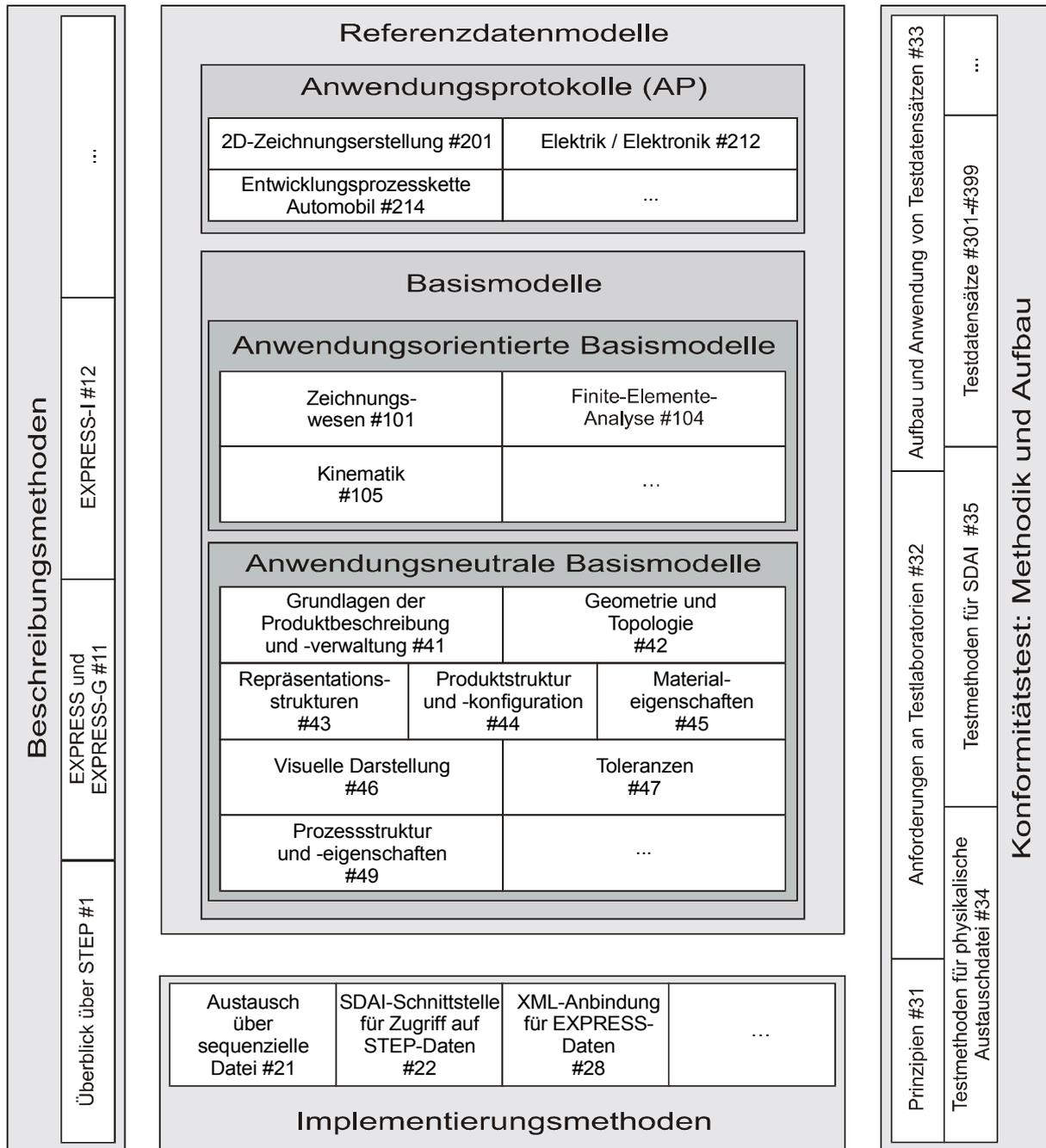


Bild 3-5: Bestandteile von STEP mit ihren zugehörigen Nummern der ISO 10303

Im Folgenden werden alle STEP- oder Getriebe-Produktmodell-spezifischen, englischsprachigen Begriffe *kursiv* geschrieben. Bezeichnungen von STEP- oder Getriebe-Produktmodell-Objekten (Kapitel 6) sind der Vorgabe gemäß mit einem Unterstrich versehen (z.B. *item_version*). In den EXPRESS_G Diagrammen beginnen STEP AP214-spezifische Begriffe mit einem kleinen, Getriebe-Produktmodell-spezifische Begriffe mit einem großen Buchstaben.

3.6.1 Bestandteile von STEP

STEP ist kein fertiges Produktreferenzmodell für den Datenaustausch, sondern ist vielmehr ein Methodenkatalog und Baukasten für die Beschreibung anwendungsspezifischer Produktreferenzmodelle, den sogenannten Anwendungsprotokollen (*Application Protocols*) mit Hilfe von Basismodellen (*Integrated Resources*), die in STEP definiert sind [G11].

STEP ist in mehrere Teile aufgliedert, die nummeriert als Teil der ISO 10303 veröffentlicht sind (Bild 3-5). In STEP sind die Modelle zur Produktdatenbeschreibung selbst, sowie weitere Standards über Beschreibungsmethoden (*Description Methods*), Implementierungsmethoden (*Implementation Methods*) und Methoden für Konformitätstests (*Conformance Testing Methodology and Framework*) enthalten [R10]. Durch die Normung der Schnittstelle selbst und der zugehörigen Beschreibungsmethoden werden Nachteile anderer Standards, etwa im Bereich der formalen Beschreibung oder der Anwendung von Testkriterien, ausgeschlossen [H5].

3.6.1.1 Die Beschreibungssprache EXPRESS

EXPRESS ist eine formale Beschreibungssprache für Produktreferenzmodelle und gehört zu den genormten Beschreibungsmethoden von STEP [I2]. Neben EXPRESS ist auch eine grafische Darstellung, EXPRESS-G genannt, definiert. EXPRESS ist eine objektorientierte Datenbeschreibungssprache und keine funktionale Programmiersprache. Ihr Hauptaugenmerk liegt auf der Definition von Datenstrukturen [S2]. Dennoch enthält EXPRESS funktionale Elemente, um beispielsweise Regeln für den Zusammenhang von Entitäten oder Algorithmen für die Datenaufbereitung bereitzustellen. Unter Entität (*Entity*) ist ein einzelnes formales Objekt zu verstehen, auf das Methoden anzuwenden sind. Entitäten sind die eigentlichen Informationsträger. EXPRESS-G verwendet die grafisch verwertbare Teilmenge der EXPRESS-Definitionen und dient zur grafischen Darstellung der definierten Datenstrukturen. Hierfür gibt es Datentypen, Attribute und Beziehungen [L11] (Bild 3-6).

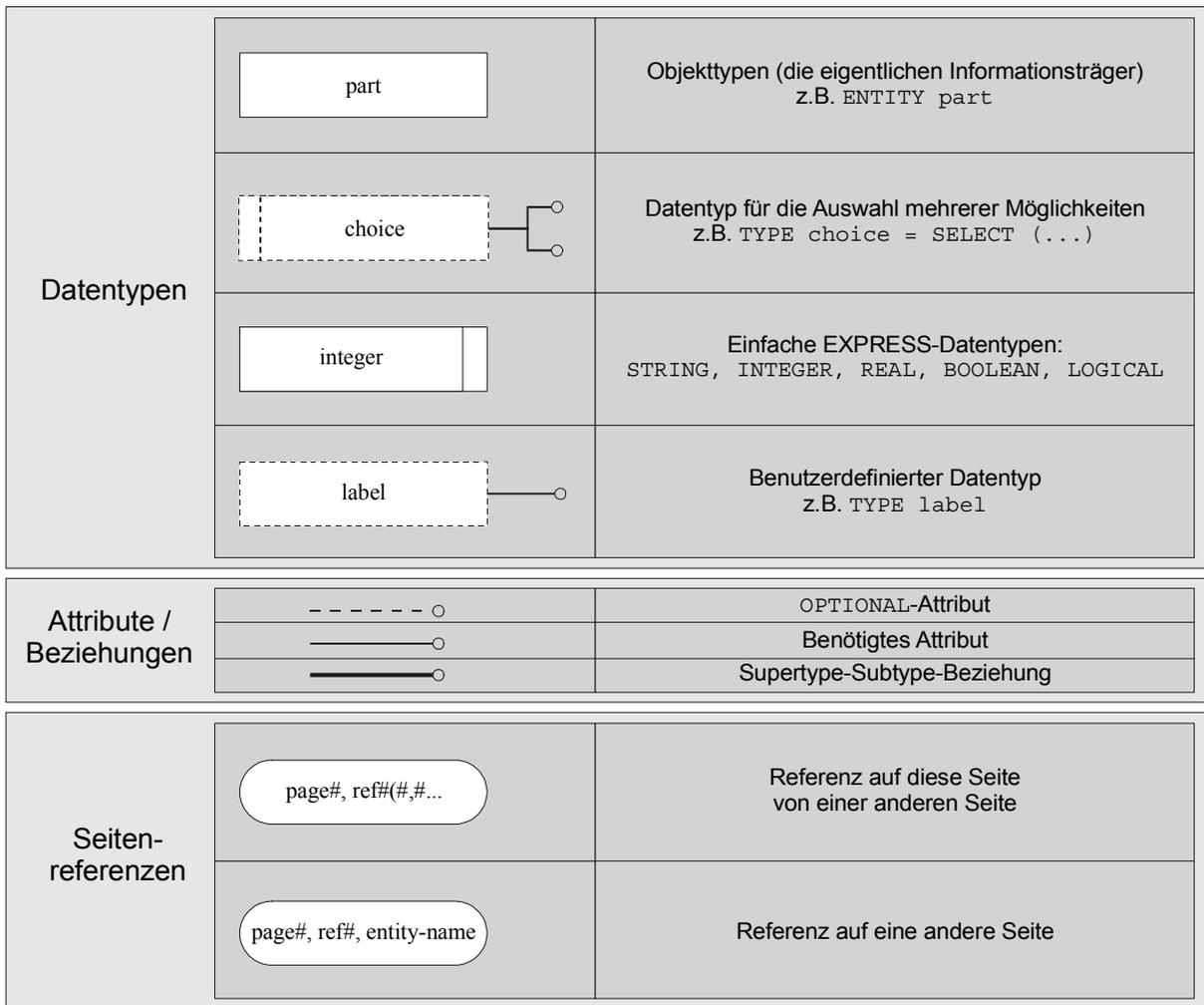


Bild 3-6: Elemente zur grafischen Beschreibung mit EXPRESS-G

EXPRESS und EXPRESS-G sind besonders für die Beschreibung von Produktreferenzmodellen geeignet [G11]. In STEP sind die Basismodelle und Anwendungsprotokolle in der Sprache EXPRESS und EXPRESS-G definiert. Damit wird die eindeutige und bereits durch Rechnerwerkzeuge interpretierbare Beschreibung von konzeptionellen Datenmodellen ermöglicht [W12]. Bild 3-7 zeigt einen Vergleich zwischen einer EXPRESS-G-Darstellung und dem äquivalenten EXPRESS-Code.

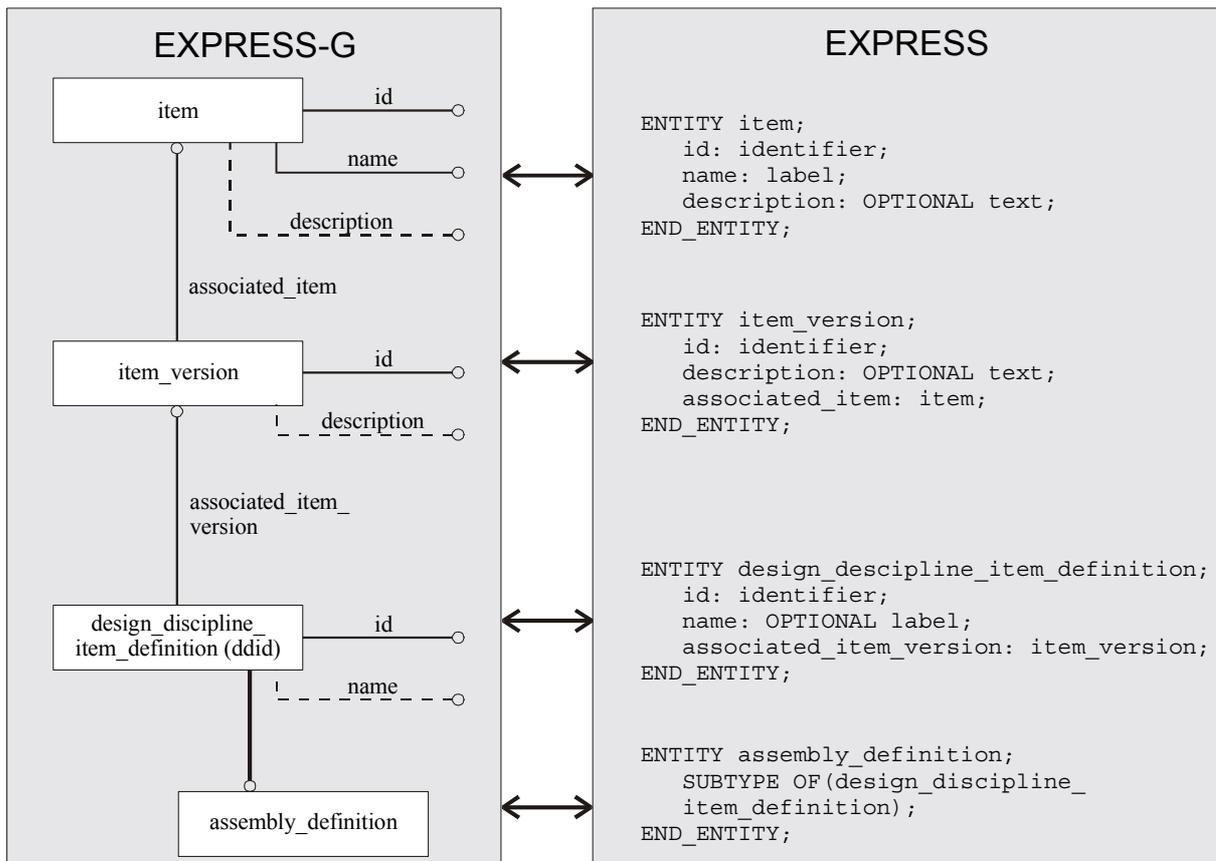


Bild 3-7: EXPRESS-G Datenmodell und äquivalenter EXPRESS-Code

3.6.1.2 STEP Basismodelle

Die Basismodelle bilden den Kern von STEP und enthalten die Grunddaten, welche in den unterschiedlichsten Anwendungen vorkommen können [E6]. Basismodelle lassen sich in anwendungsneutrale und anwendungsorientierte Basismodelle unterteilen (Bild 3-5). Die anwendungsneutralen Basismodelle (*Integrated Generic Resources*) definieren Datenmodelle für verschiedene Produktdaten, sind aber nicht auf einen bestimmten Anwendungsfall bezogen.

Die anwendungsorientierten Basismodelle (*Application Resources*) kombinieren die anwendungsneutralen Basismodelle für spezifische aber dennoch neutrale Anwendungen wie beispielsweise das Zeichnungswesen. Die so entstehenden Konstrukte sind so neutral formuliert, dass sie für beliebige Produkte und Anwendungen verwendbar sind.

Unter „Konstrukten“ werden bestimmte Anordnungen und Zusammenhänge von STEP-Entitäten zur Erreichung einer bestimmten Aufgabe verstanden. Diese Konstrukte sind meist in den Basismodellen in ihrer grundsätzlichen Ausprägung definiert und werden in den Anwendungsprotokollen an eine konkrete Anwendung angepasst.

3.6.1.3 Anwendungsprotokolle

Anwendungsprotokolle (AP) sind die konkreten Beschreibungen von Produktreferenzmodellen für bestimmte Produktparten. Da ein Anwendungsprotokoll aufgrund seiner Spezialisierung nicht alle vorhandenen Basismodelle benötigt, werden Regeln für die Abbildung der Konstrukte des anwendungsspezifischen Produktreferenzmodells auf die nötigen Basismodelle definiert. Das dadurch entstehende Datenmodell aus zusammengesetzten Basismodellen ist die Grundlage für die eigentliche Implementierung. Diesen Zusammenhang zeigt Bild 3-8.

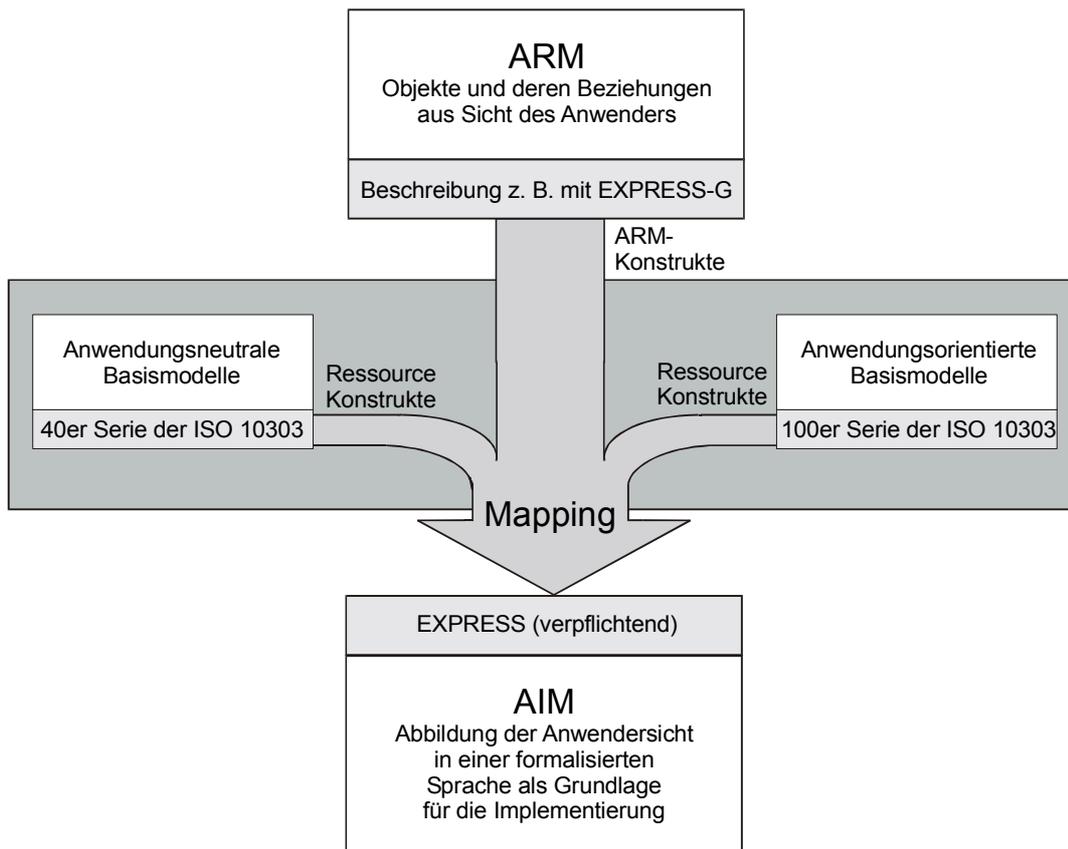


Bild 3-8: Der Mapping-Mechanismus zwischen ARM und AIM

Ein Anwendungsprotokoll besteht aus folgenden Teilen:

- **AAM (*Application Activity Model*):** Das AAM beschreibt die Funktionalitäten, die durch das Datenmodell abgedeckt werden und dient in erster Linie zur Abgrenzung des Gültigkeitsbereichs des STEP-Anwendungsprotokolls. Für die Modellierung des AAM dient die SADT-Methode, vergleiche Kapitel 3.4.3.
- **ARM (*Application Reference Model*):** Das ARM beschreibt das eigentliche Datenmodell mit allen benötigten Datenklassen und ist somit die informationstechnische Lösung aus Sicht der Anwender für das spezifizierte AAM. Es stellt aber nicht die Grundlage für die Implementierung dar. Daher kann es mit verschiedenen, in Kapitel 3.4.3 genannten Beschreibungsmethoden definiert werden. Eine Definition in EXPRESS und EXPRESS-G ist jedoch üblich und zweckmäßig.

- **AIM (*Application Interpreted Model*):** Das AIM besteht nur noch aus Konstrukten der Basismodelle und ist deshalb der eigentliche systemneutrale Kern für den Datenaustausch. Dies bildet die Grundlage für die systemneutrale Implementierung. Das AIM ist die Abbildung des ARM unter Verwendung der Basismodelle und wird aus Kompatibilitätsgründen ausnahmslos mit EXPRESS beschrieben.
- **Mapping:** Das „*Mapping*“ ist die Abbildungsvorschrift, mit der das ARM unter Verwendung der Basismodelle in das AIM übergeführt wird. Die Definition und Dokumentation erfolgt in Tabellenform unter Nennung der entsprechenden Objekte und Abbildungsregeln.

3.6.2 STEP-Anwendungsprotokoll 214

Das AP214 [I5] mit dem Titel „*Core Data for Automotive Mechanical Design Processes*“ wurde ursprünglich als Informationsmodell für die Abbildung von Produkt- und Betriebsmitteldaten des Entwicklungsprozesses der Automobilindustrie entwickelt, hat sich mittlerweile aber als Referenzmodell für den gesamten Maschinenbau durchgesetzt [G10]. Mit der umfassenden Beschreibung aller relevanten Daten der Produktlebensphasen sind Vorteile wie Systemunabhängigkeit, Datendurchgängigkeit, redundanzfreie Datenbeschreibung und Langzeitarchivierung [H8] verbunden.

3.6.2.1 Inhalt des AP214

Das AP214 deckt unter anderem folgende Bereiche durch seine Datenklassen ab:

- Produktstruktur (mit Versionen und Sichten)
- Geometrie (Volumenmodelle, Kantenmodelle, 2D-Zeichnungen, Toleranzen, ...)
- Materialangaben mit Oberflächeneigenschaften
- Dokumente (verknüpfte externe Dokumente und Dateien)
- Kinematik

Für die vorliegende Arbeit sind insbesondere die Produktstruktur, die Geometrie, die Produkteigenschaften und die verknüpften Dokumente des AP214 relevant. Zum Verständnis des in Kapitel 6 beschriebenen Produktreferenzmodells für Getriebe ist ein Verständnis der folgenden Strukturen und deren Elemente notwendig.

3.6.2.2 Produktstruktur

Im AP214 werden Bauteile und Baugruppen im zeitlichen Verlauf in jeweils neuen Versionen abgelegt. Für jede Version können beliebig viele Sichten erzeugt werden, die nur für sie relevante Daten beinhalten. Die Definition von Sichten kann beispielsweise für die Berechnung/Simulation, die Montage und das Recycling vorgenommen werden. Bauteile lassen sich über die Sichten sowohl als einfache Bauteile oder als hierarchische Baugruppen definieren. Folgende Objekte sind nach Bild 3-9 im AP214 dafür vorgesehen:

- *Item*: Das *Item* enthält alle Attribute und Produktinformationen, die versionsunabhängig sind.
- *Item_version*: Die *Item_version* repräsentiert die Version eines *Items*. Ein *Item* kann mehrere Versionen besitzen.
- *DDID (Design_discipline_item_definition)*: Das *DDID* repräsentiert eine bestimmte Sicht auf die Produktdaten. Durch die Sicht können Daten gefiltert werden. Eine *Item_version* kann mehrere Sichten enthalten. Das *DDID* enthält Unterklassen für die Repräsentation von Baugruppen (*Assembly_definition*), Informationen über Verbindungsarten zwischen Bauteilen / Baugruppen (*Mating_definition*) sowie weitere Sichten auf das Produkt (z.B. *Collection_definition*).
- *Assembly_component_relationship*, *Item_definition_instance_relationship*, *Item_instance* (Bild 3-13): Mit Hilfe der Klassen *Assembly_component_relationship*, *Item_definition_instance_relationship*, *Item_instance* werden die Einbau- und Zugehörigkeitsbeziehungen von Bauteilen oder Baugruppen zu anderen Baugruppen definiert. Durch Unterklassen der *Assembly_component_relationship* kann die Beziehung näher spezialisiert werden, zum Beispiel nach bestimmten hierarchischen Beziehungen oder bewusst unter Verzicht auf eine Hierarchie.

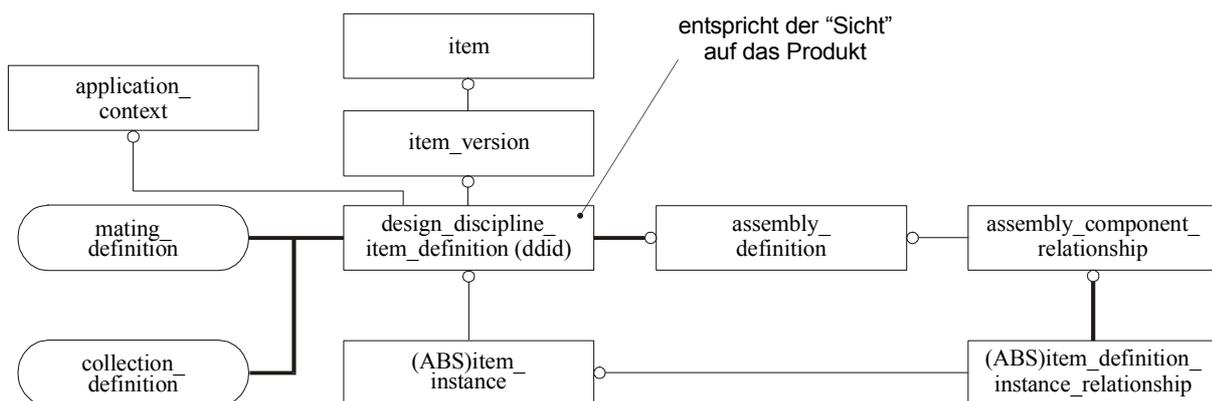


Bild 3-9: Wichtige Elemente und ihre Beziehungen der Produktstruktur des AP214

3.6.2.3 Geometrie

Für die Geometriedefinition im AP214 stehen Konstrukte für die Topologie der Gestalt und die eigentliche Geometrie zur Verfügung. Dadurch können für die Gestalt der Bauteile und Baugruppen verschiedene Geometrierepräsentationen (Volumenmodelle, Flächenmodelle, 2D-Zeichnungen, ...) definiert werden. Dies gewährleistet eine hohe Flexibilität bei der Geometriedarstellung. Die modellierte Gestalt wird direkt mit der entsprechenden Sicht (vergleiche *DDID*, Kapitel 3.6.2.2) für das beschriebene Bauteil verknüpft. Folgende Objekte sind nach Bild 3-10 dafür vorgesehen:

- *Item_shape*: Das *Item_shape* repräsentiert die Gestalt eines Items.
- *Shape_element*: Das *Shape_element* repräsentiert einen Teilbereich der Gesamtgestalt.
- *Geometric_model*: Das *Geometric_model* enthält Unterklassen für die Definition der verschiedenen Geometrierepräsentationen (*CSG_model*, *B_rep_model*, ...).
- *Shape_description_association*: Über die *Shape_description_association* werden die Geometrien (*Geometric_model*) mit der entsprechenden Gestalt (*Item_shape* oder *Shape_element*) verknüpft.

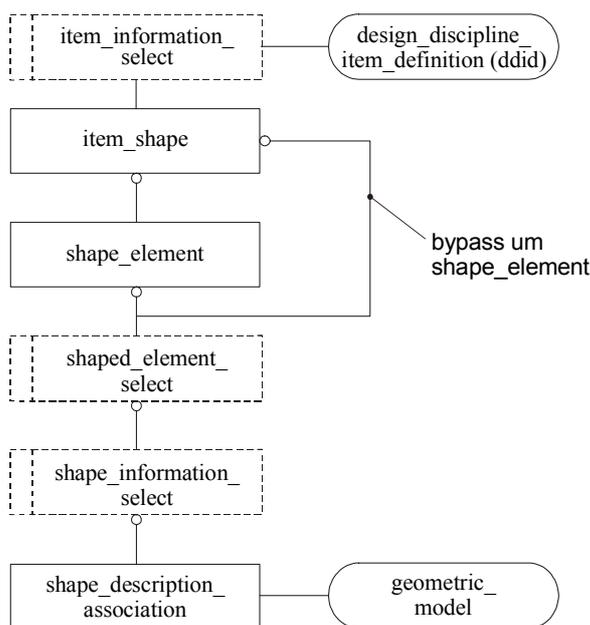


Bild 3-10: Wichtige Elemente und ihre Beziehungen der Geometrie im AP214

3.6.2.4 Produkteigenschaften

Zur Abbildung von nichtgeometrischen Produkteigenschaften (*Properties*), die sich nicht auf Materialangaben, Oberflächeneigenschaften oder Toleranzen beziehen (für diese Daten sind spezielle Konstrukte im AP214 vorhanden), wird eine Struktur zur Verfügung gestellt, mit der sich beliebige Daten an die verschiedensten Objekte anknüpfen lassen. Folgende Objekte sind nach Bild 3-11 dafür zuständig:

- *Property*: Ein *Property* enthält Unterklassen für die Definition verschiedener Eigenschaften eines Items. Ein *Property* klassifiziert ein *Property_value*.
- *Property_value*: Ein *Property_value* repräsentiert den eigentlichen Wert mit Bezeichnung und/oder Einheit der definierten Eigenschaft. Ein *Property_value* enthält Unterklassen zur Definition des Wertes als Zeichenfolge oder Werteliste sowie numerische Werte. Für die numerischen Werte kann nochmals nach einzelnen Werten, Wertebereichen oder der Angabe von Grenzwerten unterschieden werden.
- *Property_value_representation*: Ein *Property_value_representation* ist die Darstellung einer Produkteigenschaft und enthält den entsprechenden Wert (*Property_value*) sowie dessen Klassifikation (*Property*). Das *Property_value_representation* wird dem zu beschreibenden Objekt zugeordnet (etwa einem Bauteil, einer Sicht, einer Einbaubeziehung, ...).
- *Item_property_association*: Ein *Item_property_association* stellt die Verbindung zwischen der *Property_value_representation* und dem zu beschreibenden Objekt her.

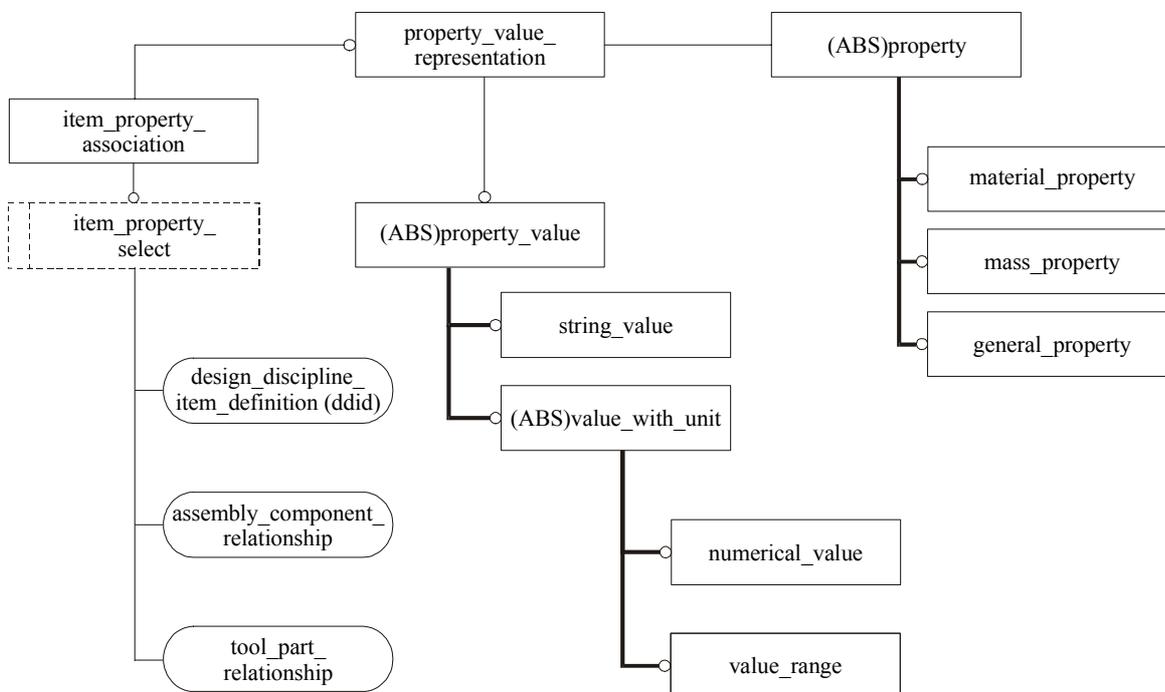


Bild 3-11: Wichtige Elemente und ihre Beziehungen der Properties im AP214

3.6.2.5 Dokumente

Im AP214 können Dokumente definiert werden, die mit dem Produktmodell verknüpft sind. Ein Dokument kann im AP214 eine digitale Datei, eine Zeichnung oder ein realer Gegenstand sein. Dokumente sind wie die Produkte versioniert (*Document_version*). Folgende Objekte sind nach Bild 3-12 für die Definition von Dokumenten im AP214 vorgesehen:

- *Document*: Ein *Document* ist der Überbegriff für beliebige Produktdaten in Dokumenten- oder Dateienform, die in beliebiger Anzahl in diesem gesammelt sind und über das *Document* identifiziert werden können.
- *Document_representation*: Eine *Document_representation* enthält Informationen über die Darstellung des Dokuments und weitere konkreten Dokumente, die damit verbunden sind.
- *Document_file*: Ein *Document_file* enthält Informationen über ein konkretes Dokument (eine Datei, eine Zeichnung, ...), das Teil einer *Document_representation* ist.
- *Document_assignment*: Über ein *Document_assignment* wird die *Document_representation* dem entsprechenden Objekt (*DDID*, *Item*, *Item_shape*, ...) zugeordnet, dem das Dokument zur Verfügung gestellt wird.

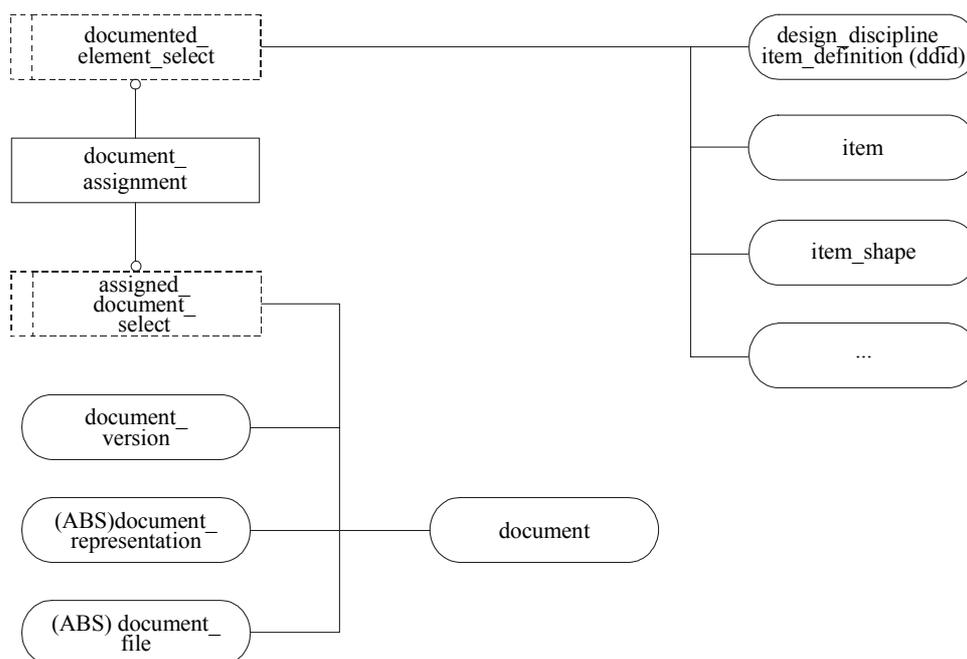


Bild 3-12: Wichtige Elemente und ihre Beziehungen der Dokumente im AP214

4 Datenaustausch zwischen Berechnung und Gestaltung

Das zentrale Problem der durchgängig rechnerbasierten Produktentwicklung ist die Verknüpfung zwischen Programmen zur Gestaltung und Berechnung im zeitlich durchgängigen Konstruktionsprozess (Bild 4-1). Sowohl im Entwurf als auch bei der Ausarbeitung wird fortlaufend zwischen Berechnung/Simulation und Gestaltung gewechselt. Zur Realisierung der Verknüpfung ist sowohl der Prozess des fortlaufenden Datenaustausches als auch die hierfür verwendete Methode maßgebend.

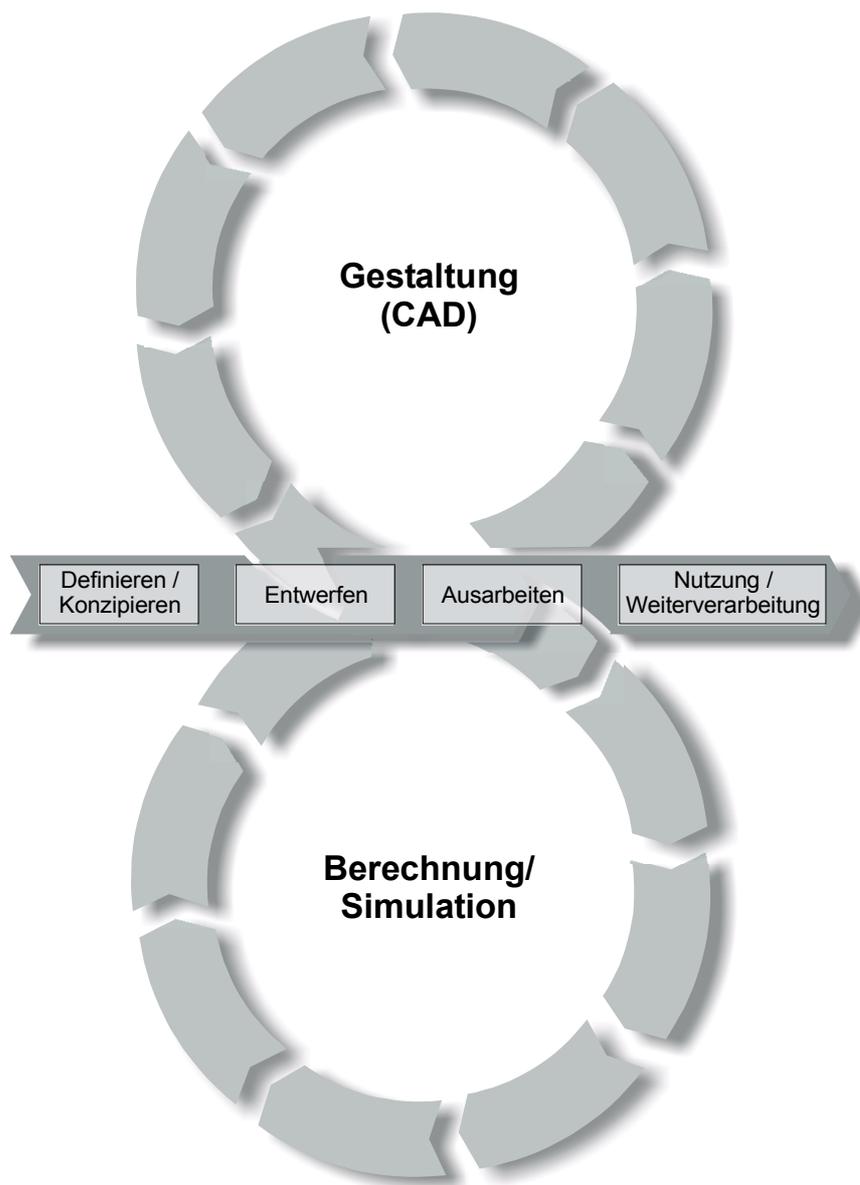


Bild 4-1: Kreislauf zwischen Gestaltung und Berechnung/Simulation auf der zeitlichen Achse der Produktentwicklung

Die Komplexität des Datenaustausches hängt von

- den eingesetzten Systemen,
- dem Inhalt der auszutauschenden Daten und
- der Methode, was der gewünschten Qualität der Produktdaten im Austauschprozess entspricht, ab.

Werden Daten speziell zwischen einem Gestalt- und Berechnungs-/Simulationsmodell ausgetauscht, so reicht der rein geometriebezogene Dateninhalt nicht mehr aus, sondern es müssen leistungsfähige Datenschnittstellen oder Produktmodelle eingesetzt werden.

4.1 Darstellung der Problematik

In der Produktentwicklung sind Berechnung und Gestaltung zwei grundlegende Aufgaben, die eng miteinander verwoben sind [A5]. Auf der einen Seite werden die geometrischen Eigenschaften des Produkts in einem Geometriemodell abgebildet und auf der anderen Seite gleichzeitig Voraussagen über die physikalisch/technischen Eigenschaften des entstehenden Produkts auf Basis der Geometrie getroffen. Dies geschieht mit Berechnungs- und Simulationsprogrammen.

Für das gestalterische Arbeiten stehen leistungsfähige CAD-Systeme zur Verfügung. Auf der Seite der Berechnung/Simulation existieren Programme zur Unterstützung der anfallenden Auslegungen, Nachweise und Optimierungen. Durch den Umgang mit diesen Hilfsmitteln wird aus technischer Sicht eine ganzheitliche Betrachtung des entstehenden Produkts durchgeführt. Bislang musste der Konstrukteur zwischen den Gestalt- und den Berechnungsmodellen von Hand wechseln, die Modelle aneinander anpassen und aktualisieren. Der nicht unerhebliche Datenaustausch in beide Richtungen zwischen den CAD-Modellen und den Berechnungen wurde zu einem sehr großen Teil über eine manuelle Datenweitergabe bewältigt, was nicht nur zu einem Zeitverlust, sondern auch zu Übertragungsfehlern führte.

Ein rechnerunterstützter Datenaustausch schafft Abhilfe, indem die beiden Ebenen Gestaltung und Berechnung sich gegenseitig „verstehen“ und in eine Datenübertragung nicht mehr manuell eingegriffen werden muss. Dadurch wird die Qualität erhöht und ein schneller Datenaustausch ohne Redundanzen gewährleistet. Der beim Zusammenwirken von CAD-Systemen und Berechnungsprogrammen stattfindende, digitale Datenaustausch muss zur Unterstützung des Entwicklungsprozesses in beide Richtungen stattfinden. Dies wird allgemein als bidirektionaler Datenaustausch bezeichnet [A5]. Hinzu kommt auch noch die zeitliche Komponente. Den Gesamtzusammenhang verdeutlicht Bild 4-2. Der bidirektionale Datenaustausch zwischen zwei Systemen wird statisch zu einem bestimmten Zeitpunkt betrachtet. Bei Berücksichtigung des zeitlichen Ablaufs der Konstruktion kommt die zeitlich-durchgängige, d.h. phasenübergreifende Verknüpfung hinzu. Zur Erfassung ist eine Prozessabbildung (Kapitel 4.5) notwendig.

4.2 Statischer Austausch von Produktdaten zwischen Programmen

Unter statischem Produktdatenaustausch von beispielsweise zwei Programmen wird die zweifach unidirektionale Kopplung über Datenschnittstellen zu einem oder mehreren zeitlich unabhängigen Zeitpunkten verstanden. „Unidirektional“ beschreibt, dass die Hin- und Rückführung der Datei unabhängig voneinander ist und in keinem Kontext zueinander steht. In Bild 4-2 sind die Möglichkeiten hierfür von oben nach unten in aufsteigender Komplexität beim Datenaustausch aufgezeigt.

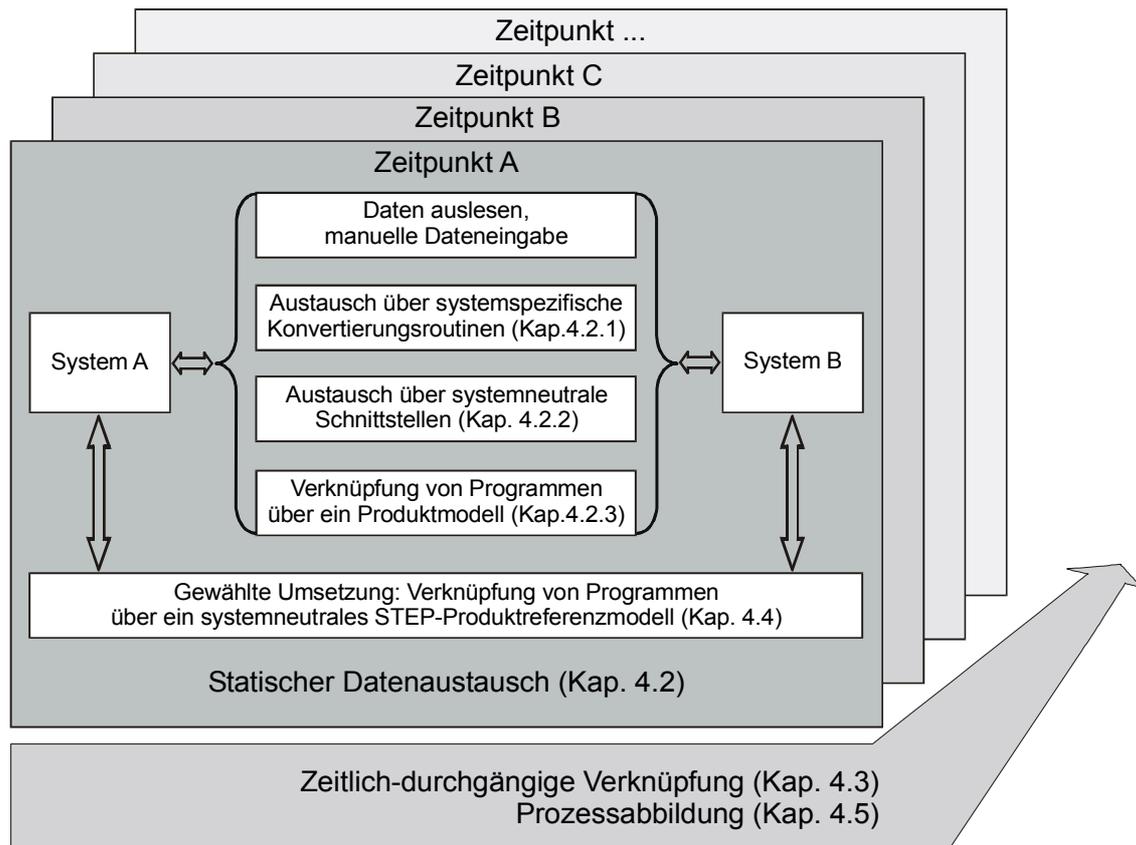


Bild 4-2: Szenario des Datenaustausches zwischen zwei Programmen im zeitlichen Kontext

Die Produktentwicklung im zeitlich durchgängigen Konstruktionsprozess wird in Kapitel 4.3 untersucht. Dort wird gezeigt, welche Möglichkeiten es gibt, ein Produktmodell entlang der Zeitschiene der Produktentwicklung zu führen.

4.2.1 Verknüpfung von Programmen über systemspezifische Konvertierungsroutinen

Für die Verknüpfung von zwei Programmen ist für jedes ein Prozessor notwendig, der aus dem nativen Format über einen Konvertierungsprozess das native Format des anderen Programms ausgibt. Sind mehr als zwei Programme am Datenaustausch beteiligt, werden eine Vielzahl von Prozessoren benötigt. Bild 4-3 zeigt alle durch Pfeile angedeutete Prozessoren in einem Beispielszenario und gibt deren Berechnungsformel an. Der erhebliche Erstellungs-

und Pflegeaufwand der Prozessoren macht die Bedeutung von systemneutralen Schnittstellen deutlich.

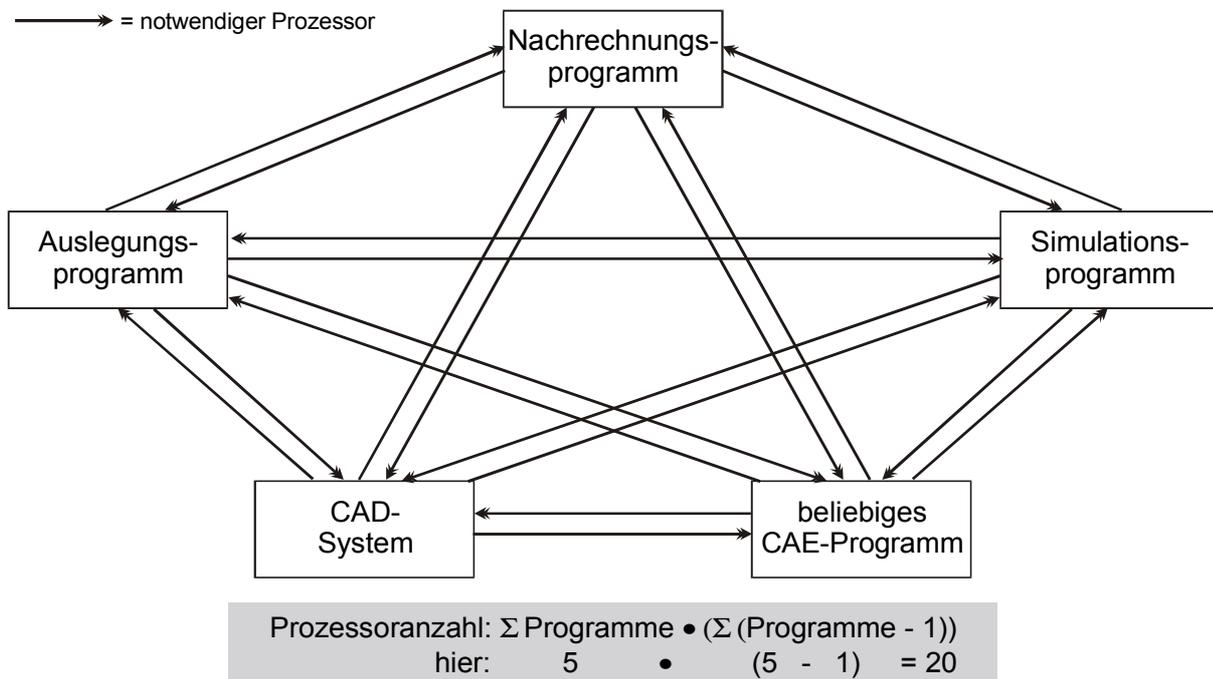


Bild 4-3: Prozessoranzahl beim Datenaustausch ohne systemneutrale Schnittstelle

4.2.2 Verknüpfung von Programmen über systemneutrale Schnittstellen

Für den Datenaustausch zwischen zwei Programmen werden je ein Pre- und Postprozessor benötigt [A6]. Ein Preprozessor wandelt das systemeigene (=native) Datenformat in ein Schnittstellenformat um. Der Postprozessor des anderen Programms setzt das Schnittstellenformat in sein natives Datenformat um. Wird das in Bild 4-3 gezeigte Austauschzenario um ein systemneutrales Schnittstellenformat (Kapitel 3.2) für den Datenaustausch erweitert (z.B. STEP), ergibt sich bei nur fünf zu verknüpfenden Programmen bereits mindestens eine Halbierung der benötigten Anzahl von Prozessoren. Bild 4-4 zeigt diesen Zusammenhang. Die Pre- und Postprozessoren des jeweiligen Programms wandeln das native Datenformat in STEP-Daten um bzw. umgekehrt. Für den Austausch der STEP-Dateien sind keine weiteren Prozessoren notwendig. Daraus ergibt sich die benötigte Prozessoranzahl im Systemverbund als doppelte Anzahl der beteiligten Programme. Eine weitere Senkung der Prozessoranzahl wird in der Regel dadurch erreicht, dass beteiligte Programme (z.B. CAD-Systeme) bereits mit systemneutralen Schnittstellen ausgestattet sind.

Die in Bild 4-4 verwendete STEP-Austauschdatei kann als unidirektional und nur für den Zeitpunkt des einfachen Datenaustausches zwischen zwei Programmen als existent betrachtet werden. Im Systemverbund existiert keine zentrale Datenhaltung oder Datenspeicherung. Der Einsatz eines Produktmodells schafft Abhilfe und unterstützt einen bidirektionalen Datenaustausch.

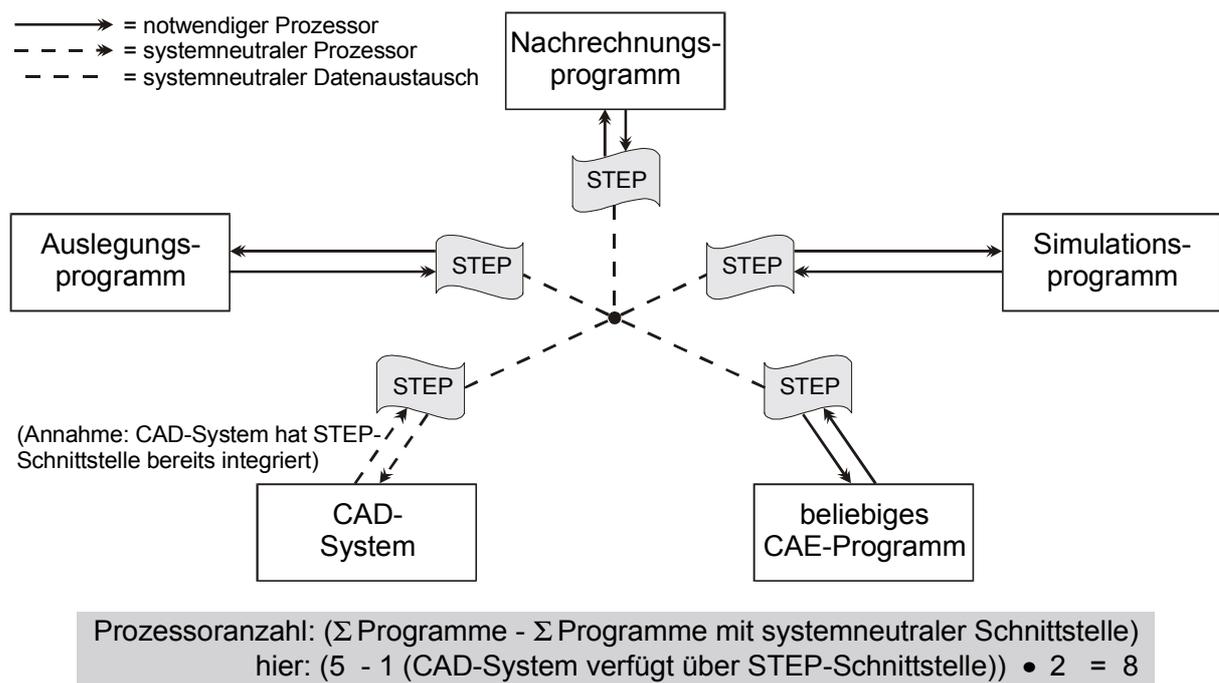


Bild 4-4: Prozessoranzahl bei Datenaustausch über eine systemneutrale Schnittstelle

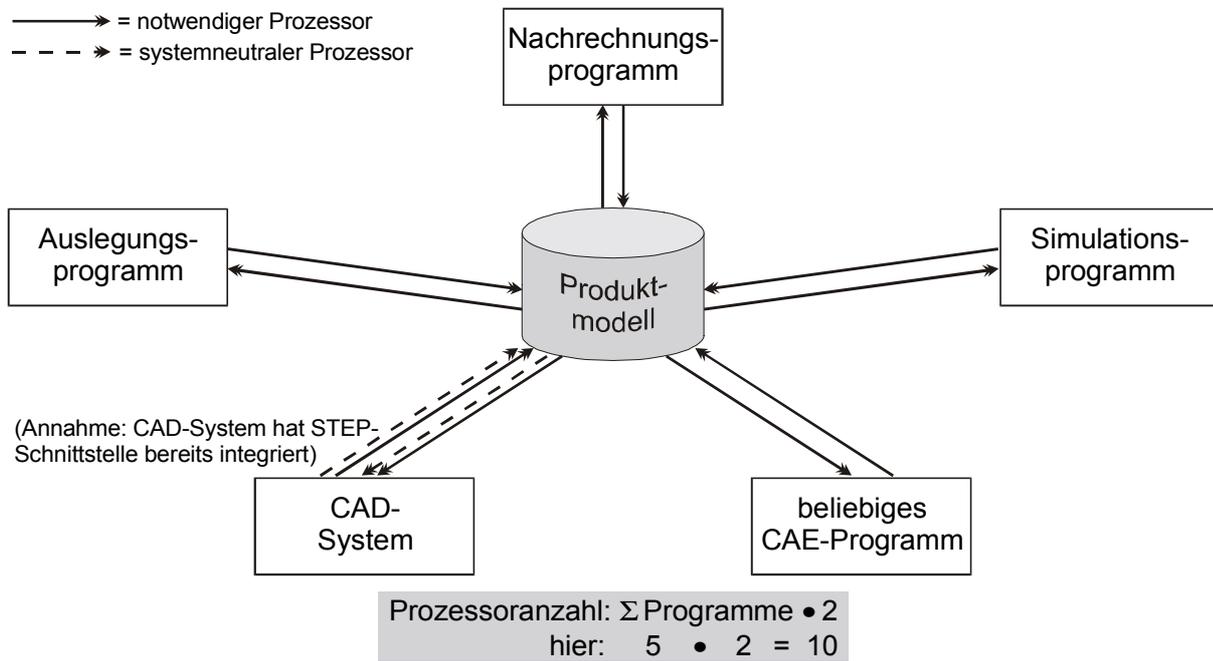
4.2.3 Verknüpfung von Programmen über ein Produktmodell

Ein Produktmodell steht als zentrales Bindeglied zwischen den beteiligten Programmen (Bild 4-5). Durch dessen Einsatz kann zwar im Vergleich zu Bild 4-4 aufgrund eines notwendigen Vorhandenseins eines Pre- und Postprozessors je Programm die Prozessoranzahl nicht weiter gesenkt werden, es sprechen aber andere Gründe für den Einsatz eines Produktmodells, auch wenn dessen Einsatz mehr Anstrengungen erfordert. Folgende Vorteile können mit einem Produktmodells erzielt werden, was jeweils die beschriebenen Arbeiten nach sich zieht:

- Kontrollierter Datenaustausch zwischen den beteiligten Programmen. Prozessoren zum Datenein- und Datenauslesen mit entsprechenden Kontrollmechanismen werden zusätzlich notwendig.
- Nur ein gültiges Datenmodell zu jedem Zeitpunkt vorhanden. Eine Versionierung innerhalb des Produktmodells wird notwendig.
- Produktmodell vereint alle möglichen Austauschdaten der beteiligten Programme. Hierfür muss ein Produktreferenzmodell (Definition in Kapitel 3.4.1) erstellt werden.

Im Prinzip erfüllt jede singuläre Austauschdatei zwischen zwei Programmen die Anforderungen eines zeitlich fixierten Produktmodells, sofern sie logisch strukturierte Produktdaten enthält. Im Zusammenhang mit der Verknüpfung mehrerer Programme geht die Definition eines Produktmodells darüber hinaus. Ein Produktmodell wird als zentrales Organ zur Verwaltung und Speicherung aller für den Datenaustausch vorgesehenen Produktdaten definiert. Mit der Einführung eines Produktmodells werden Funktionen notwendig, die das Ein- und Auslesen

der Daten in das Produktmodell übernehmen und kontrollieren und das Produktmodell aufgrund seiner Definition im Produktreferenzmodell in sich konsistent halten.



bei Verwendung eines systemneutralen Produktmodells gilt:

Prozessoranzahl: $(\Sigma \text{ Programme} - \Sigma \text{ Programme mit systemneutraler Schnittstelle}) \cdot 2$
 hier: $(5 - 1 \text{ (CAD-System verfügt über STEP-Schnittstelle)}) \cdot 2 = 8$

Bild 4-5: Prozessoranzahl bei Datenaustausch mit einem zentralen Produktmodell

Die Erstellungsmöglichkeiten eines auf STEP basierenden Produktreferenzmodells werden in Kapitel 4.4 umfassend erläutert, die Umsetzung eines Produktreferenzmodells für Getriebe ist in Kapitel 6 beschrieben.

4.3 Zeitlich-durchgängige Verknüpfung von Berechnung und Gestaltung

Wird die zeitliche Komponente zu dem statischen Austausch von Produktdaten zwischen Programmen hinzugenommen, ergibt sich eine zeitlich-durchgängige Verknüpfung von Berechnung und Gestaltung. Hierfür gibt es prinzipiell zwei Möglichkeiten: die kontinuierliche und sequenzielle bidirektionale Kommunikation. Die Grundprinzipien dieser beiden Kommunikationstechnologien unter besonderem Einbezug der Bidirektionalität wurden von AMFT et al. [A5] näher beleuchtet und werden nachfolgend umrissen.

4.3.1 Kontinuierliche bidirektionale Kommunikation

Zur bidirektionalen Integration von Berechnung und Gestaltung entlang des Entwicklungsprozesses bietet sich die Verschmelzung beider Teilmodelle an. Werden Berechnung und Gestaltung gemeinsam und redundanzfrei in einem integrierten Modell abgelegt, so wird über Verknüpfungen jede Veränderung der Geometrie auf die Berechnung übertragen und umgekehrt. Jedes neue Berechnungsergebnis kann über Abhängigkeiten seine eigenen Eingabeparameter beeinflussen. Es entsteht eine ganzheitliche Sicht auf Physik und Gestaltung. Bild 4-6 zeigt, wie Berechnung und Gestaltung kontinuierlich in Verbindung stehen und gemeinsam auf der Zeitachse des Entwicklungsprozesses entlanggeführt werden.

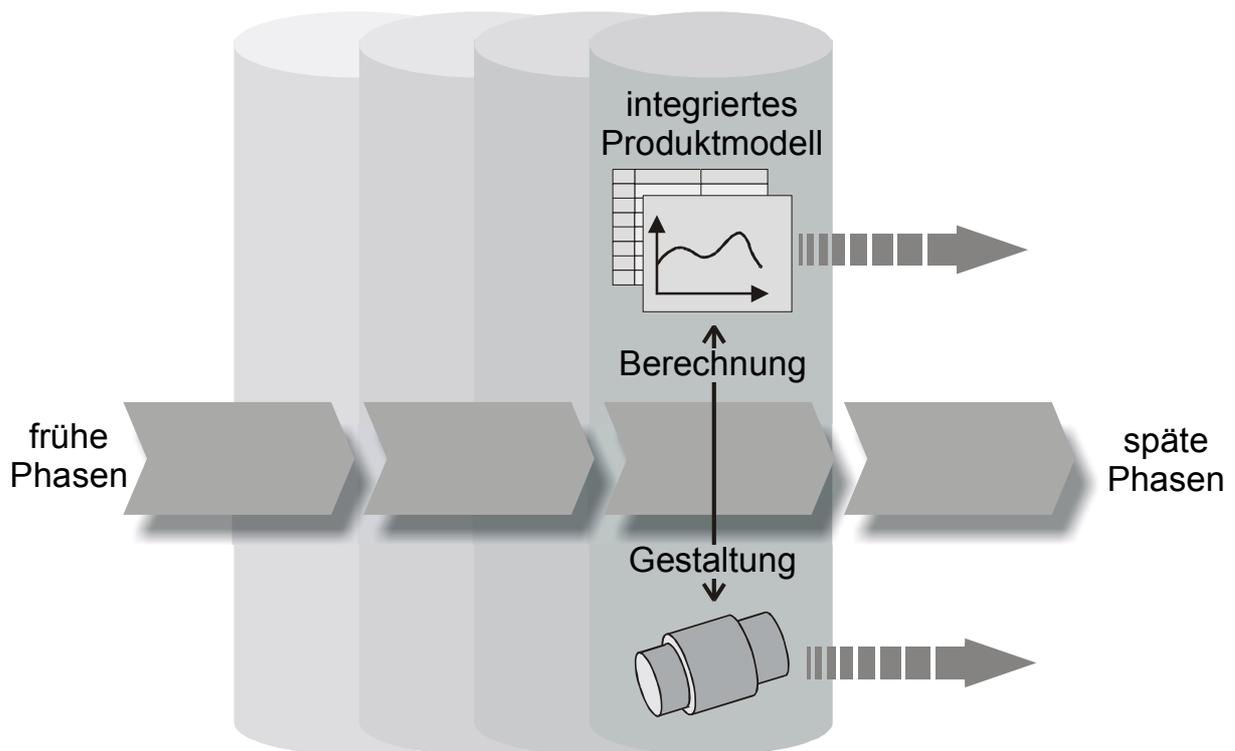


Bild 4-6: Kontinuierlich bidirektionale Kommunikation

Die kontinuierliche bidirektionale Integration fordert ein hochintegriertes redundanzfreies System, welches Berechnung und Gestaltung als Einheit behandelt. Das komplexe Produktmodell ist kontinuierlichen Änderungen unterzogen, da jede Parametervariation ganzheitlich

und simultan Einfluss auf alle geometrischen und physikalischen Eigenschaften der Ein- und Ausgabe nimmt. Die Gleichbehandlung von geometrischen und nichtgeometrischen Parametern öffnet dem Konstrukteur aber eine ganzheitliche Sicht auf das Modell. Selbst bei der Abbildung einfacher Formelberechnungen entstehen bereits sehr komplexe und reaktive Produktmodelle. Mechanismen zu deren Handhabbarkeit und Optimierung unter Wahrung einer Übersichtlichkeit stellen die besondere Herausforderung dieser Integrationsform dar.

Realisiert werden kann die kontinuierlich bidirektionale Kopplung nur, wenn eine vollständige Integration der Programme aus Gestaltung und Berechnung in einem integrierten System erfolgt. Dies muss auf systemtechnischer Ebene erfolgen. Denkbar ist z.B. die feste Integration von Berechnungsmodulen in CAD-Systeme (z.B. [P4]) oder aber die Verknüpfung von CAD-System und Berechnungswelt über einen mathematischen Solver, der in direktem Kontakt zu CAD- und Berechnungsroutinen steht (z.B. [L5]).

Das daraus entstehende, starre aber schnell-reaktive System ist aufgrund des festen Integrationskonzepts zum Zeitpunkt des Datenaustausches als nicht systemneutral zu betrachten. Das Produktmodell ist in sich geschlossen. Aus diesen Gründen eignet sich die kontinuierlich bidirektionale Kommunikation nicht zur Verwendung in der vorliegenden Arbeit.

4.3.2 Sequenzielle bidirektionale Kopplung

Als zweite Möglichkeit kann der bidirektionale Datenaustausch in zwei singuläre Vorgänge aufgeteilt sein. Der Datenaustausch ist in beide Richtungen und zu beliebigem Zeitpunkt möglich, im Moment des Datenaustausches ist er allerdings unidirektional. Bild 4-7 zeigt das Prinzip.

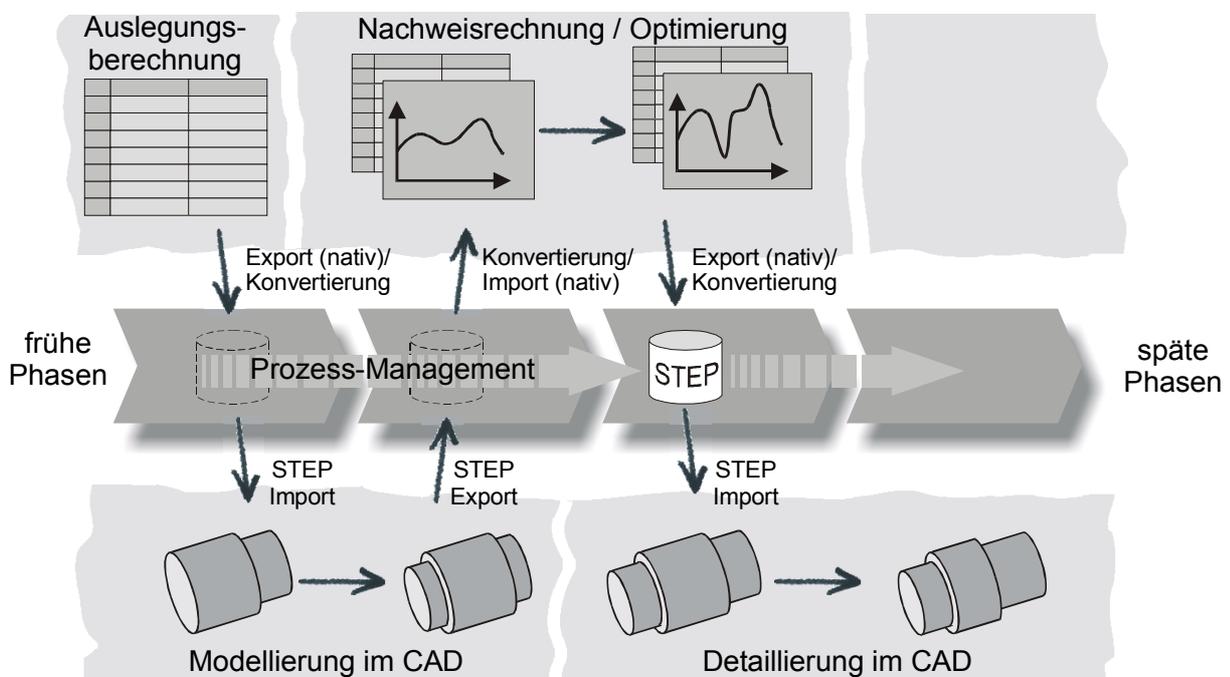


Bild 4-7: Sequenzielle bidirektionale Kopplung

Nach jedem Transfer wird der Datenfluss unterbrochen und das Modell, mit dem auf der Berechnungsseite oder auf der Gestaltungsseite gearbeitet wird, detailliert. Das jeweils andere Modell bleibt von den Arbeiten zunächst unberührt. Die Modelle auf beiden Seiten stehen eigenständig und außer im Augenblick der Datenübertragung ungekoppelt nebeneinander. Einen Abgleich der Modelle übernimmt ein Prozessmanagement-System.

Die Pfeile, die den Datenaustausch zwischen Berechnungs- und Gestaltungsseite symbolisieren, zeigen durch ihre Pfeilspitze die Unidirektionalität und Singularität der Kopplung. Die Bruchlinien zwischen den Modellen symbolisieren die Entwicklungssprünge, die sich durch einen Wechsel aus der Gestaltung zur Berechnung und umgekehrt ergeben.

Die sequenzielle bidirektionale Integration setzt den Schwerpunkt auf die Schnittstellen, Prozessoren und das Produktmodell, die alle für den Datenaustausch notwendig sind. Unter Verwendung eines einheitlichen, im Idealfall systemneutralen Produktmodells lassen sich bestehende Berechnungs- und Gestaltungssysteme beliebiger Komplexität gut miteinander verbinden. Der Datenaustausch und Abgleich mit dem Produktmodell erfolgt nur bei einem Wechsel zwischen Gestaltung und Berechnung und führt zu einer Trägheit im System. Dies ermöglicht eine Überschaubarkeit der Aktionen. Eine Integration weiterer Programme und Systeme ist zudem aufgrund der klar definierten Schnittstellen und Anschlüsse an das Produktmodell leicht möglich. Zur Realisierung des Datenaustausches entlang der Zeitachse wird ein Prozessmanagement benötigt, wie es zum Beispiel in Kapitel 5.2.6 beschrieben wird.

4.4 Methoden zur Entwicklung eines auf STEP basierenden Produktreferenzmodells

Die Verwendung eines Produktmodells hat sich für den in Kapitel 4.2 geschilderten, statischen Produktdatenaustausch als vorteilhaft erwiesen. Eine zeitlich-durchgängige Verknüpfung von Berechnung und Gestaltung lässt sich ohne den Einsatz eines Produktmodells nur sehr schwer realisieren.

4.4.1 Vorteile durch die Verwendung bestehender Ansätze für die Produktmodellierung

Voraussetzung für den Produktmodell-basierten Datenaustausch ist das Vorhandensein eines Produktreferenzmodells (Kapitel 3.4.1). Die Entwicklung eines eigenen Produktreferenzmodells wird wesentlich vereinfacht, wenn auf bereits vorhandene Konstrukte zur Produktmodellierung zurückgegriffen wird. Eine weitestgehende Anwendung bestehender Produktreferenzmodelle (z.B. STEP) wird angestrebt. Damit sind folgende Vorteile verbunden:

- Das Produktreferenzmodell ist **in sich schlüssig**, logische Fehler werden durch die Verwendung einer ausgereiften Produktmodellstruktur vermieden.
- Das Produktreferenzmodell ist **vollständig**. Bei der Übertragung eines Produktreferenzmodells auf den eigenen Anwendungsfall ist man gezwungen, das bestehende Produktreferenzmodell in seiner gesamten Komplexität zu verstehen und dessen Grobstruktur zu übertragen.

- Das in Produktreferenzmodellen neutral ausgedrückte **Expertenwissen** kann genutzt werden. Damit kann eine sehr große Zahl von Anwendungsfällen abgedeckt werden.
- Je nach Art der Produktmodellierung (Kapitel 4.4.2) kann eine sehr große **Kompatibilität** mit dem ursprünglichen Produktreferenzmodell erreicht werden.
- Es können **Anschlusspunkte** an das eigene Produktreferenzmodell definiert werden, ab denen bestimmte Konstrukte (z.B. die Geometrie) des standardisierten Produktreferenzmodells gültig sind.

Die Verwendung bestehender Teilproduktmodelle findet sich auch in dem internen Aufbau von STEP wieder. So bauen verschiedene Anwendungsprotokolle (siehe Kapitel 3.6.1.3) auf den gleichen Basismodellen (siehe Kapitel 3.6.1.2) auf. Dadurch wird eine Kompatibilität zwischen den Anwendungsprotokollen auf Basis von Konstrukten der Basismodelle erreicht. Es stellt sich die Frage, welche Möglichkeiten die komplexe und nur schwer zu durchblickende STEP-Norm [11] bietet, eigenständige Produktmodelle zu erstellen und gleichzeitig eine möglichst hohe Kompatibilität zum Standard aufrechterhalten.

4.4.2 Gegenüberstellung von Möglichkeiten zur Erstellung eines Produktreferenzmodells mit STEP

Die Entwicklung von Produktreferenzmodellen zielt darauf ab, eine größtmögliche Kompatibilität mit bestehenden Produktreferenzmodellen herzustellen. Hierbei nimmt STEP aus den in Kapitel 3.6 für allgemeine Produktmodelle angeführten und folgenden, zusätzlich für STEP geltenden, Gründen eine Vorreiterrolle ein:

- Daten unterschiedlichster Quellen werden unabhängig von ihrer Anforderung oder Anwendung gespeichert.
- Produkt- und Geometriedaten können gemeinsam mit Administrativ-, Meta- und Berechnungs-/Simulationsdaten logisch verknüpft werden.
- Es gibt zwei Ebenen für Daten: Die in Kapitel 3.6.1.3 vorgestellte Anwendungs- (ARM) und die Austauschene (AIM). Da beide über das „*Mapping*“ gegenseitig logisch verbunden sind, ermöglicht dies eine schnelle Datenintegration und einen anwendungsübergreifenden Datenaustausch.
- Da sich die AIM-Ebene ausschließlich auf die STEP Basismodelle bezieht, können über das eigene Produktreferenzmodell hinaus Daten ausgetauscht werden.

Sollen Daten, die über den Umfang der STEP-Norm hinausgehen oder diese weiter detaillieren in einem auf STEP basierenden Produktmodell abgebildet werden, so muss das zu erstellende STEP-Produktreferenzmodell erweitert werden. Dies ist prinzipiell auf folgenden Wegen möglich.

- **Erstellung eines neuen Anwendungsprotokolls:** Basierend auf den Basismodellen (Kapitel 3.6.1.2) wird ein neues Anwendungsprotokoll erstellt und in Anlehnung an die ISO dokumentiert.
- **Erweiterung eines bestehenden Anwendungsprotokolls:** Ein vorhandenes Anwendungsprotokoll wird um fehlende Objekte aus den Anforderungen des neuen Produktreferenzmodells erweitert. Dabei sollte nur die ARM-Ebene, nicht jedoch die AIM-Ebene ergänzt werden, um eine Kompatibilität zum Anwendungsprotokoll weiter zu gewährleisten.
- **Projizierung eines bestehenden Anwendungsprotokolls auf den Anwendungsfall:** STEP-Anwendungsprotokolle bieten offene Konstrukte an, die eine sehr hohe Flexibilität aufweisen und zur Einbindung fremder Daten und Strukturen verwendet werden können.

Tabelle 4-1 zeigt eine Übersicht der drei Vorgehensweisen in Bezug auf Aufwand, Kompatibilität, Normierungsmöglichkeiten und Akzeptanz bei Anwendern und Entwicklern.

	Erstellung eines neuen Anwendungsprotokoll	Erweiterung eines bestehenden Anwendungsprotokolls	Projizierung auf ein bestehendes Anwendungsprotokoll
ARM-Ebene	neu zu entwickeln	Erweiterungen möglich	Änderungen nicht notwendig
AIM-Ebene	neu, unter Bezug auf STEP Basismodelle	sollte nicht verändert werden	wird nicht verändert
Mapping	neu: ARM auf AIM	Erweiterung des bestehenden Mappings ARM auf AIM	kein Mapping notwendig. Nur Zuordnung vom Produktmodell zu den ARM-Objekten des AP's
STEP-Kompatibilität der physikalischen Austauschdatei	100% mit diesem AP, bedingt mit Anderen	mit dem gewählten AP, falls AIM-Ebene unverändert, sonst nur bedingt	100% kompatibel auf ARM- und AIM-Ebene
Attribute/ Properties	neu anzugeben, Verknüpfung neu zu definieren	neu anzugeben, auf bestehenden Mechanismen aufbauend	neu anzugeben, auf bestehenden Mechanismen aufbauend
Aufwand und Komplexität	hoch	mittel	niedrig
Akzeptanz im STEP-Umfeld	in der Regel niedrig	mittel	hoch
Normierung möglich als	Teil der ISO 10303	Technical Specification (TS), PAS, Richtlinie, Erweiterung eines genormten AP's unwahrscheinlich	Technical Specification (TS), PAS, Richtlinie
Umsetzungsdauer	langsam	mittel	schnell
STEP-Konformität	nur mit diesem AP	neue Objekte nur mit dem erweiterten AP	100% mit dem gewählten AP der ISO 10303 und den damit verbundenen „Integrated Resources“
Implementierungsaufwand	mittel bis hoch	mittel	niedrig

Tabelle 4-1: Vergleich der Vorgehensweisen für die Erstellung eines STEP-Produktreferenzmodells

Die Neuentwicklung eines Anwendungsprotokolls ist mit dem größten Aufwand verbunden und sollte nur dann realisiert werden, wenn sich das Anwendungsgebiet in keines der bestehenden Anwendungsprotokolle integrieren lässt. Die hierfür notwendige Prüfung wird durch den immer noch hohen Abstraktionsgrad der Anwendungsprotokolle erschwert. Die Zusammenhänge der in den Anwendungsprotokollen beinhalteten Mechanismen und Möglichkeiten sind oft nicht sofort ersichtlich. Bei einigen Projekten wird deshalb zu schnell der Entschluss

gefasst, ein neues Anwendungsprotokoll zu entwickeln, ohne sich dem hohen Aufwand bei der Erstellung und den Hürden bei der Normierung bewusst zu sein.

Weniger aufwendig ist die Erweiterung eines bestehenden Anwendungsprotokolls um branchenspezifische Belange. Hierbei ist zu beachten, dass nur Veränderungen auf ARM-Ebene durchgeführt werden sollten. Die neu hinzukommenden Objekte werden dann auf das bestehende AIM des Anwendungsprotokolls „gemappt“, um auf der Ebene des Datenaustausches eine hundertprozentige Kompatibilität zum gewählten Anwendungsprotokoll herzustellen. Die soeben geschilderte Vorgehensweise birgt jedoch den Nachteil der Veränderung eines Normdokuments.

Dies kann vermieden werden, wenn ein bestehendes Anwendungsprotokoll nicht verändert sondern lediglich auf die Belange des Anwendungsgebietes projiziert wird. Dies bedeutet eine Anpassung der im Anwendungsprotokoll enthaltenen Mechanismen auf die Zusammenhänge des spezifischen Produktreferenzmodells der Anwendung. Dabei entsteht ein neues Produktreferenzmodell, welches zunächst unabhängig von STEP sein kann. Die systemneutrale Umsetzung erfolgt durch eine Übertragung des Produktreferenzmodells für das Anwendungsgebiet auf die ARM-Ebene des gewählten Anwendungsprotokolls. Sowohl das ARM, das „Mapping“ auf die AIM-Ebene als auch die Austauschdatei des Anwendungsprotokolls werden nicht verändert und entsprechen weiterhin dem ISO-Standard. So wird eine hundertprozentige Kompatibilität zu dem jeweiligen Anwendungsprotokoll erreicht.

4.4.3 Resultierende Vorgehensweise zur Erstellung eines anwendungsbezogenen STEP-Produktreferenzmodells

Bild 4-8 zeigt die sich aus Kapitel 4.4.2 abgeleitete Vorgehensweise zur Entwicklung eines anwendungsbezogenen STEP Produktreferenzmodells.

Zunächst gilt es zu prüfen, ob eine Projizierung eines bestehenden Anwendungsprotokolls auf den Anwendungsfall möglich ist. Gerade dieser Schritt wurde bisher bei Projekten zur Erstellung von Produktreferenzmodellen nicht sorgfältig genug durchgeführt (z.B. Gefahr eines voreiligen Entschlusses zur Erstellung eines Produktreferenzmodells für die Schmiedeindustrie [A8]), bedarf er doch einer genauen Studie der bestehenden Anwendungsprotokolle. Dies umfasst insbesondere das Verstehen der Zusammenhänge im genormten STEP-Produktreferenzmodell, worüber nur wenig Literatur verfügbar ist [A12].

Falls sich kein Anwendungsprotokoll zur Projizierung eignet, sollte zuerst versucht werden, Veränderungen auf ARM-Ebene des gewählten Anwendungsprotokolls vorzunehmen. Ergibt sich auch hier eine unzureichende Abbildung der gewünschten Zusammenhänge im Produktreferenzmodell, ist auch eine Erweiterung der AIM Ebene des STEP-Anwendungsprotokolls denkbar, was allerdings die Kompatibilität auf der Ebene des Datenaustausches einschränkt.

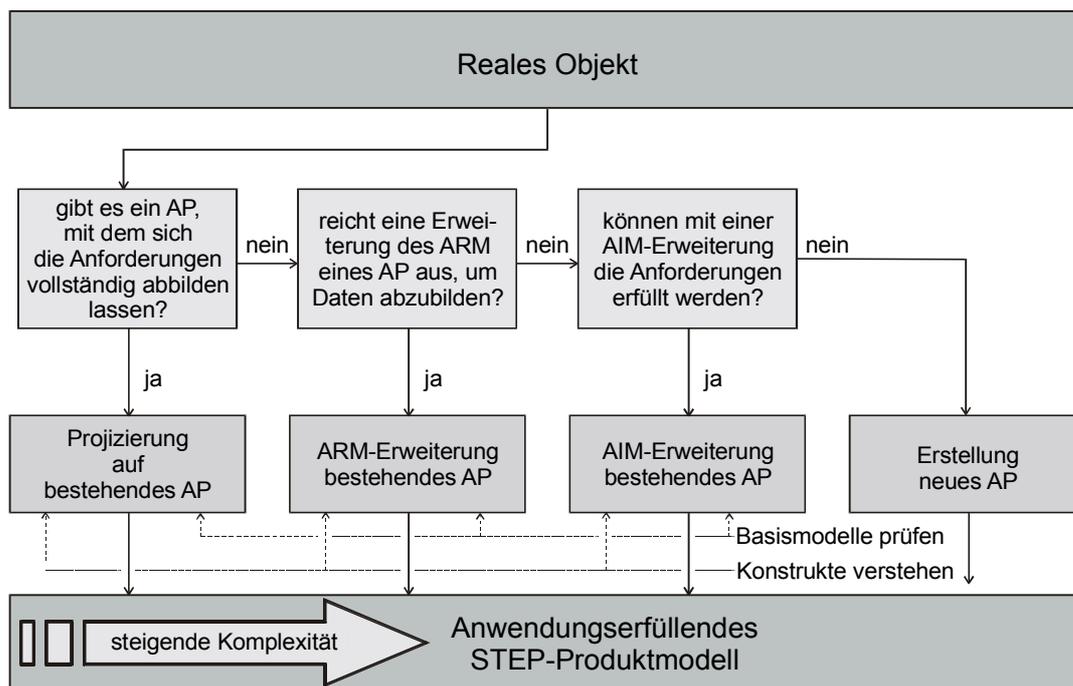


Bild 4-8: Vorgehensweise zur Entwicklung eines anwendungsbezogenen STEP-Produktreferenzmodells

Stellt sich trotz einer sorgfältigen Recherche in Hinblick auf eine Erweiterung der bestehenden Anwendungsprotokolle heraus, dass sich keines für den Anwendungsfall eignet, muss ein neues Anwendungsprotokoll geschaffen werden. Dies kann mit einem mehrjährigen Aufwand verbunden sein und wird nur bei wenigen Produkten notwendig sein, da die STEP-Norm aufgrund ihrer Neutralität bei den Basismodellen einen sehr großen Anwendungsbereich abdecken kann.

4.5 Prozessabbildung

Ein Prozess wird durch viele Einzelbausteine (Aktivitäten genannt) charakterisiert, die durch logische Verknüpfungen verbunden werden. Gemäß der in Kapitel 3.4.3 beschriebenen SADT- Methode besteht eine Aktivität aus der Datenbeschaffung, der Verarbeitung und der Datenweitergabe. Zum Zeitpunkt der Aktivität werden Steuerungsinformationen und Methoden eingeleitet. Setzt man dieses Aktivitätenmodell in zeitlichen Bezug zu einem Produktmodell, ergibt sich Bild 4-9. Dort wird der Unterschied zwischen Produkt- und Prozessmodellierung deutlich. Ein Produktmodell ist nur zu einem bestimmten Zeitpunkt gültig und wird nach einer Aktivität durch das sich ergebende neue Produktmodell ersetzt. Das alle Aktivitäten verwaltende Prozessmodell muss in der Lage sein, die Prozessschritte, Methoden und die Ein- und Ausgabeinformationen aufzuzeichnen und den jeweiligen Stand des Produktmodells zu dokumentieren. Diese Aufgabe wird im rechnerbasierten Programmsystem von dem Prozessmanagement-System übernommen (Kapitel 5.2.6).



Bild 4-9: Bezug zwischen Aktivität und Produktmodell

Die während der Produktentwicklung anfallenden Daten lassen sich je nach Bezugsobjekt, Aktivität oder Verwendungszweck in folgende Datenklassen unterscheiden:

- Die **Produktdaten** stellen alle digital erfassbaren Daten dar, die in Bezug zum jeweiligen Produkt gesetzt werden können. Gemeint sind die Daten eines Produkts zu einem bestimmten Zeitpunkt, die sich entweder auf die Produktstruktur, auf die Geometrie oder auf das physikalische Verhalten beziehen. In den Produktdaten verankert sind auch die Metadaten, die alle Daten über Struktur und Kontext des Produkts abbilden.
- Die **Prozessdaten** beziehen sich grundsätzlich auf einen Modifikationsprozess der Produktdaten und umfassen alle im Prozessablauf für die Durchführung, Steuerung, Koordination und Kommunikation von Aktivitäten vorkommenden Daten und deren Methoden.

Die Herausforderung für die Erstellung eines rechnerbasierten Programmsystems ist die Handhabung sowohl der Produkt- als auch der Prozessdaten während der Produktentwicklung.

5 Entwicklung eines durchgängig rechnerbasierten Programmsystems zur Produktentwicklung

Das rechnerbasierte Programmsystem (vergleiche HÖHN et al. [H14]) für die Entwicklung allgemeiner Produkte dient zur Verwirklichung der rechnerbasierten Produktentwicklung. Nach KRAUSE et al. [K13] erfordert der durchgängig virtuelle Produktentwicklungsprozess ein Konstruktionsleitsystem mit einer gemeinsamen Infrastruktur und einer einheitlichen grafischen Benutzungsoberfläche. Hierfür und für eine Integration beliebiger Programme und Systeme aus der Gestaltung, Berechnung und den angrenzenden Bereichen ist das rechnerbasierte Programmsystem vorgesehen.

5.1 Übersicht und Anforderungen an einen programmsystemgeführten Konstruktionsablauf

In der immer komplexer werdenden Produktentwicklung gibt es inzwischen eine Vielzahl von Rechnerprogrammen (für Getriebe siehe Kapitel 7.2) zur Problemlösung unterschiedlicher Aufgaben. Ein großes Hindernis besteht in der Verwaltung der anfallenden Daten und Dateien und dem sich daraus ergebenden Datenaustausch. Oftmals wird es dem Anwender selbst überlassen, sich um den Datenaustausch während des Entwicklungsprozesses zu kümmern. Dies hat nicht nur eine unnötige zeitliche Verlängerung des Entwicklungsprozesses zur Folge, sondern ist insbesondere mit einer hohen Fehleranfälligkeit behaftet, was zu Datenredundanzen und Unübersichtlichkeit bei der Versionsverwaltung führen kann.

Hier setzt das Programmsystem zur rechnerbasierten Produktentwicklung an und stellt Methoden vor, diese Diskrepanzen zu vermeiden. Der Entwicklungsprozess differiert bei einer groben Betrachtung nach Produkt und Branche nur gering. Ziel ist nach Bild 1-1 die Entwicklung eines Produkts für eine Problemlösung durch dessen Nutzung. Bei der Entwicklung werden nach SPUR&KRAUSE [S13] folgende Schritte in Kreisläufen bearbeitet:

- Klären und Präzisieren der Aufgabenstellung
- Suchen nach Lösungsprinzipien
- Gestalten der maßgebenden Module
- Gestalten des Produkts
- Ausarbeiten der Ausführungsangaben

Bild 5-1 verdeutlicht diesen Ablauf als Prozessmodell mit Hilfe der SADT-Methode (Kapitel 3.4.3). Die zweite Ebene veranschaulicht das Vorgehen, das insbesondere durch Kreisläufe und Rücksprünge bei der Produktentwicklung charakterisiert ist.

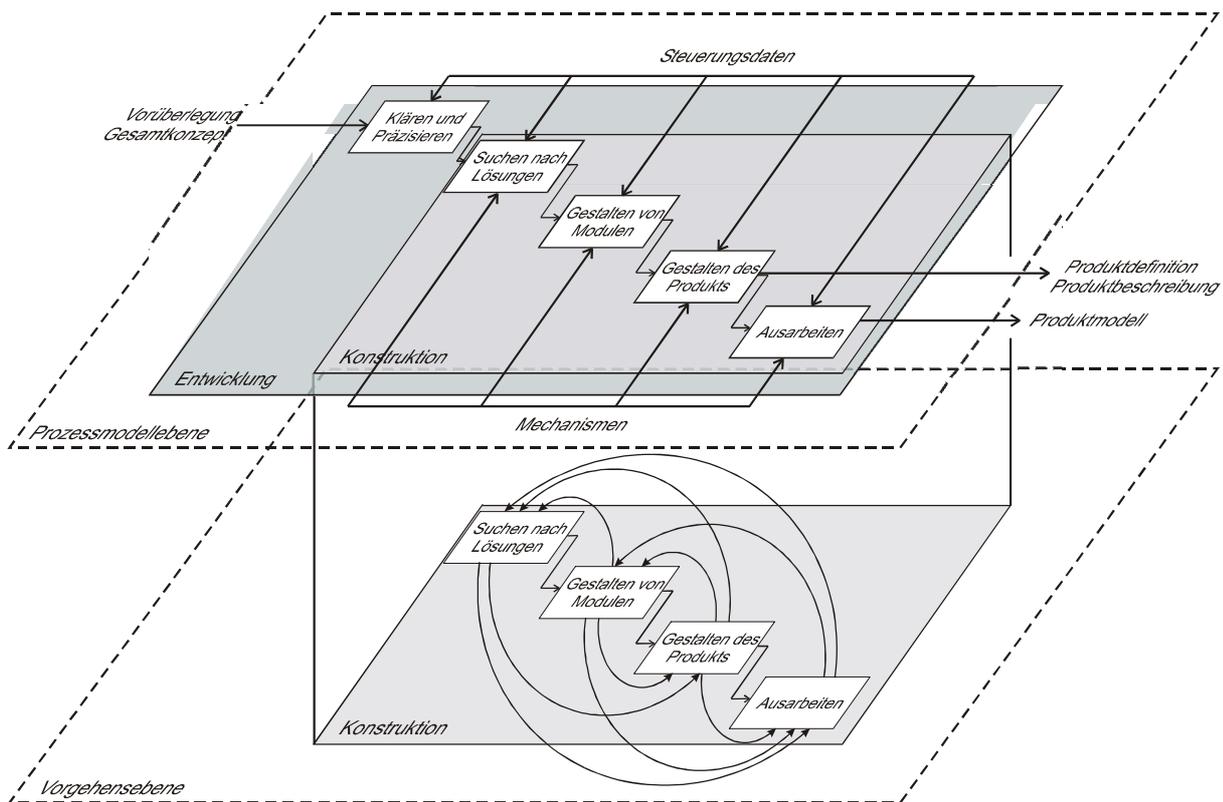


Bild 5-1: Prozessmodell und Vorgehensebene des allgemeinen Entwicklungsprozesses

Soll dieser Entwicklungsprozess nun rechnerbasiert stattfinden, sind beim Entwurf eines Programmsystems folgende Anforderungen zu beachten:

- offene Schnittstellen zur Einbindung bestehender und etablierter Rechnerprogramme
- Bereitstellung eines einheitlichen, möglichst systemneutralen Produktmodells
- Möglichkeit zur Verwaltung und Konvertierung aller während der Produktentwicklung entstehenden Daten und Dateien
- Langzeitarchivierung des Produktmodells und des Produktentwicklungsprozesses
- ergonomisch gestaltete grafische Benutzeroberfläche
- effiziente Darstellung der Produkt- und Prozessstruktur und der erzielten Ergebnisse

5.2 Systemarchitektur

Die Architektur des rechnerbasierten Programmsystems bietet eine benutzer- und aufgabenangepasste Unterstützung für den gesamten Konstruktionsprozess an. Eine wichtige Zielsetzung besteht in der Konzeption einer modularen, flexiblen und anpassbaren Systemarchitektur. Die in Bild 3-4 gezeigte Systemarchitektur von PDM-Systemen liefert für das Produktdaten-, Produktstruktur- und Workflowmanagement wichtige Anregungen. Bild 5-2 vermittelt einen Überblick über die zugrunde liegenden Komponenten der Systemarchitektur des rech-

nerbasierten Programmsystems und der verwendeten Abkürzungen. Eine Unterstützung erfolgt sowohl bei der Entwicklung selbst als auch bei der Wiedergewinnung abgelegter Dokumente und Daten. Der Daten- und Dateienfluss wird während der Entwicklung rechnerbasiert gesteuert, überwacht und verwaltet, und somit der Entwicklungszeitraum entscheidend verkürzt und Übertragungsfehler vermieden.

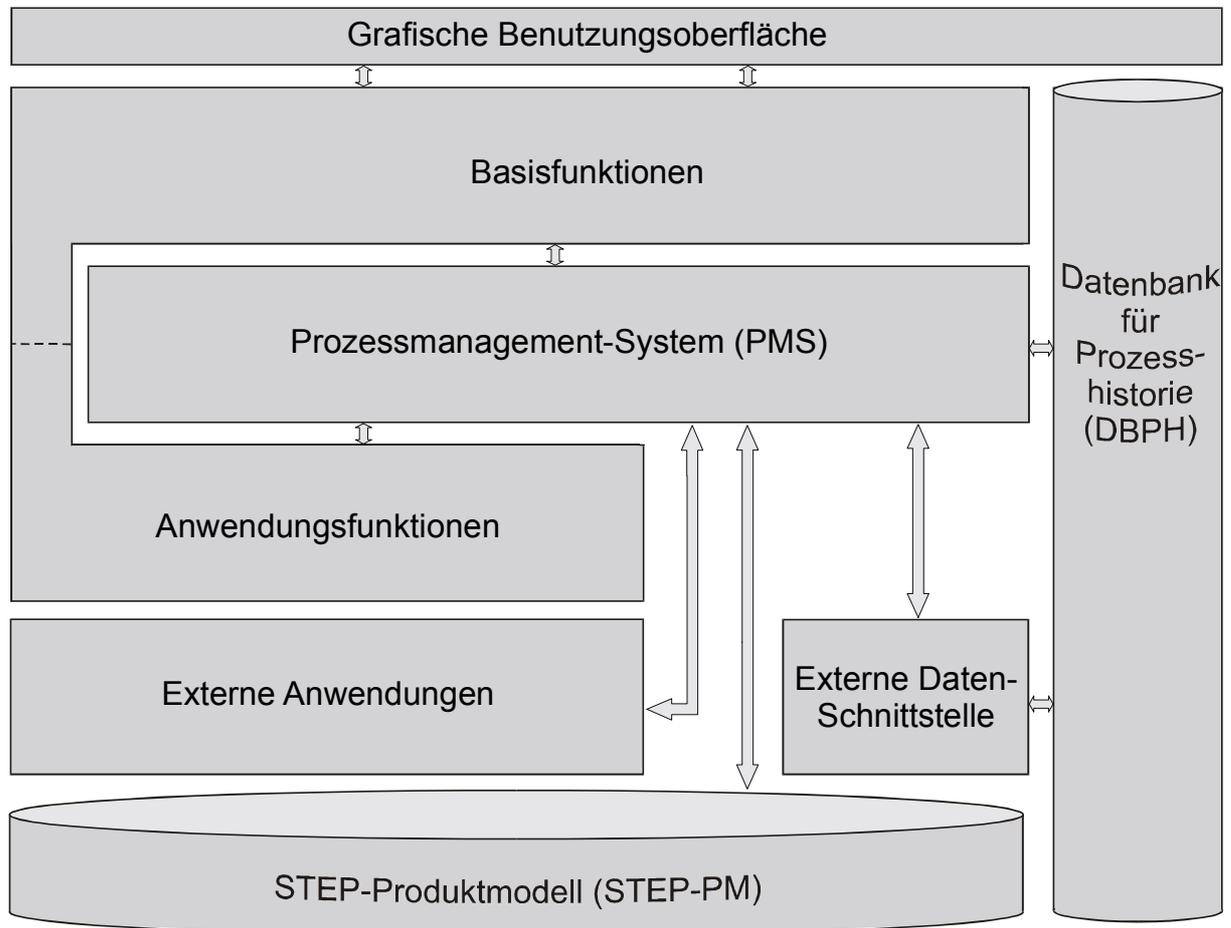


Bild 5-2: Systemarchitektur des rechnerbasierten Programmsystems

Alle Entwicklungsschritte sind durch eine Trennung von Produkt- und Prozessdaten eindeutig und einfach nachzuvollziehen. Durch diese Trennung bleibt das Produkt autonom und isoliert vom Prozess. Die Steuerung des rechnerbasierten Programmsystems erfolgt über Basis- und Anwendungsfunktionen. Das Prozessmanagement-System übernimmt die Datenkontrolle und den Datenaustausch.

Obwohl die Architektur eines Softwaresystems bei der Anwendung in der Regel nicht wahrgenommen wird, beeinflusst sie doch die Arbeitsweise und die Qualität der Ergebnisse. In den Kapiteln 5.2.1-5.2.7 werden die in Bild 5-2 dargestellten Systemkomponenten und ihre Funktionen näher erklärt.

5.2.1 Basisfunktionen

Die Basisfunktionen dienen zum einen der Systemkonfiguration, Systemanpassung und internen Steuerung des Programmsystems, zum anderen ermöglichen sie einen Überblick über den Entwicklungsprozess, den aktuellen Entwicklungsstand, die Produktstruktur, den Zusammenhang zwischen Baugruppe, Einzelteil und Dokument. Durch die Basisfunktionen wird eine dynamische Konfiguration der Systemumgebung erreicht und eine lückenlose Visualisierung aller anfallenden Daten möglich.

Die Gliederung der einzelnen Basisfunktionen und ihre Bedeutung im Entwicklungsprozess werden in Kapitel 7.1 näher erläutert. Eine Auflistung aller in der Systemarchitektur integrierten Basisfunktionen zeigt Bild 5-3.

Basisfunktionen									
3D-Produktstruktur	3D-Geometrieviewer	Datei-Viewer	Karteikartensystem	2D-Strukturbaum	Suchfunktionen	Systemfunktionen	Systemkonfiguration	Projektverwaltung	Tabelle der Prozesshistorie

Bild 5-3: Basisfunktionen

5.2.2 Anwendungsfunktionen

Die Integration, Steuerung und Überwachung der externen Anwendungen ist die grundlegende Aufgabe der Anwendungsfunktionen. Sie gestalten sich weitgehend flexibel. Flexibel bezeichnet die Anpassung des produktneutralen rechnerbasierten Programmsystems über die Anwendungsfunktionen an den jeweiligen Produktbereich und die produktspezifischen Programme. Die Anwendungsfunktionen werden über die grafische Benutzeroberfläche angesprochen und finden ihre Ausführung durch das Prozessmanagement-System.

Die Anwendungsfunktionen stellen eine schnelle und eindeutige Ausführung der Anwenderbefehle sicher. Sie kümmern sich unabhängig vom Datenformat um die Datei- und Datenbeschaffung aus dem STEP-Produktmodell und der Datenbank für Prozesshistorie und deren Rückführung nach der Bearbeitung. Sie starten und beenden externe Programme und dokumentieren in einem Ereignisprotokoll deren Ansteuerung. Eine Zusammenfassung der einzelnen Anwendungsfunktionen liefert Bild 5-4.

Anwendungsfunktionen						
Anbindung externer Anwendungen an das STEP-PM	Aufruf Hilfe-datei	Verwaltung von Daten in externen Programmen	Dateibeschaaffung über PMS	Dateirückführung über PMS	externer Programmstart über PMS	externes Programmende über PMS

STEP-PM ... STEP-Produktmodell
PMS ... Prozess-Managementsystem

Bild 5-4: Anwendungsfunktionen

5.2.3 Externe Anwendungen

Externe Anwendungen bezeichnen bestehende, eigenständige Programme (CAD-Systeme, Berechnungs- und Simulationsprogramme, ...), welche aus der grafischen Benutzungsoberfläche gestartet werden. Externe Programme differieren je nach Produkt und Anwendung. Das rechnerbasierte Programmsystem ist offen für die Einbindung neuer Programme in das System und für die Weitergabe von Daten und Dateien (Kapitel 5.4).

Die Aufgabe des rechnerbasierten Programmsystems besteht nicht darin, in den Ablauf der externen Anwendungen einzugreifen. Es stellt lediglich der externen Anwendung die benötigten Daten und Dateien zur Verfügung und stellt eine fehlerfreie Datenrückführung nach Beendigung der Anwendung sicher. Die externen Anwendungen bleiben in sich geschlossen und werden nicht verändert. Die Kommunikation erfolgt ausschließlich über Austauschdateien, an die Konverter des rechnerbasierten Programmsystems gekoppelt werden. In Bild 5-5 sind die externen Anwendungen exemplarisch für die Getriebeentwicklung aufgeführt.

Externe Anwendungen							
Berechnungs- und Simulationsprogramme	externe Datenbanken (z.B. Werkstoff, Normteile)	2D-CAD-Systeme 3D-CAD-Systeme	Datei-Editoren	PDM-Editor	LRZ-Grafik-Viewer	nativer 3D-CAD Viewer	Sonstige: CAx, Office, ...

LRZ-Grafik-Viewer ... Softwaretool des Leibniz Rechenzentrums der bayerischen Akademie der Wissenschaften

Bild 5-5: Externe Anwendungen

5.2.4 Grafische Benutzungsoberfläche

Die in Bild 5-6 schemenhaft dargestellte grafische Benutzungsoberfläche stellt die Schnittstelle zwischen Mensch und Maschine dar. Die Akzeptanz von Anwendersoftware wird maßgebend durch die Gestaltung der Benutzungsoberfläche beeinflusst. Aus diesem Grund orientiert sich die Oberflächenbeschaffenheit an der Systemarchitektur. Basis- und Anwendungsfunktionen werden klar getrennt. Mit verschiedenen Interaktionselementen, wie Buttons, Toolbar, Menübar und Grafiken, wird die Kommunikation zwischen Anwender und der grafischen Benutzungsoberfläche erleichtert und der Überblick gewährleistet. Die Umsetzung mit diesen Hilfsmitteln ist unter ergonomischen Gesichtspunkten erfolgt.

Auch bei einem komplizierten Entwicklungsprozess darf der Benutzer den Überblick nicht verlieren. Durch die Möglichkeit, mehrere Basisfunktionen gleichzeitig zu nutzen, wird die Effizienz der Visualisierung von Daten und Dateien erheblich gesteigert und eine Navigation bei der Produktentwicklung erleichtert.

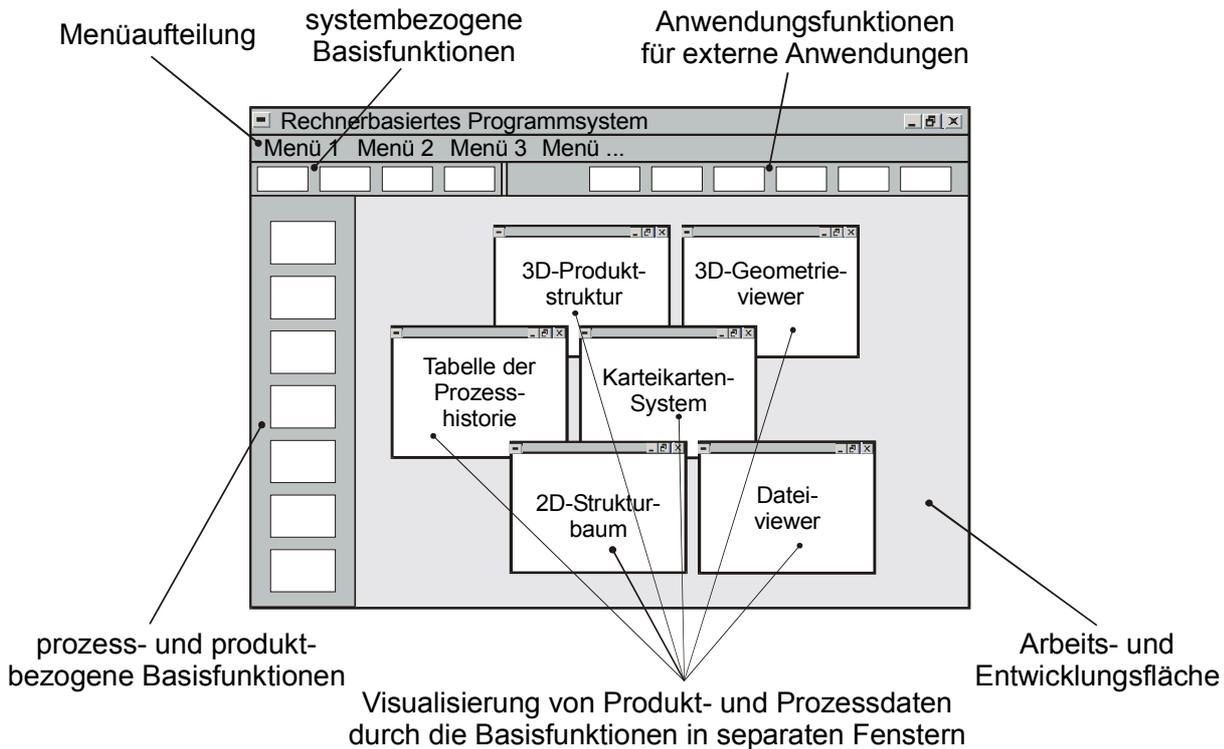


Bild 5-6: Schematische Darstellung der grafischen Benutzeroberfläche

5.2.5 Daten- und Dateienverwaltung

In einem rechnerbasierten Entwicklungsprozess unter Anwendung verschiedener externer Programme entstehen eine Vielzahl von Daten und Dateien. Aufgrund der Verschiedenartigkeit und Vielfalt der externen Anwendungen werden die unterschiedlichsten Daten- und Dateiformate produziert. Eine strukturierte Daten- und Dateiverwaltung dieser nativen Daten- und Dateiformate im zeitlichen Entwicklungsprozess wird notwendig.

Die erzeugten Daten können in zwei zusammenhängende Teile getrennt werden:

- Die **Produktdaten** werden im STEP-Produktmodell (STEP-PM),
- die **Prozessdaten** in der Datenbank für die Prozesshistorie (DBPH) abgelegt.

Die logische Verknüpfung der beiden Datenbanken erfolgt über das Prozessmanagementsystem und ist aufgrund gegenseitiger Abhängigkeiten zwischen Produkt- und Prozessdaten notwendig.

5.2.5.1 STEP-Produktmodell (STEP-PM)

Produkt- und Prozessmodelle wurden ausführlich in Kapitel 3.4 vorgestellt. Für den statischen Produktdatenaustausch zwischen Programmen hat sich in Kapitel 4.2 ein Produktmodell als vorteilhaft erwiesen. Das STEP-Produktreferenzmodell ist derzeit das einzige, welches aufgrund seiner Normierung [I1] als systemneutral bezeichnet werden kann. Wie die

Produktdaten sind auch Metadaten im STEP-Produktmodell abgelegt. Werden diese Daten erneuert oder geändert, wird das STEP-Produktmodell aktualisiert. So liefert es dem Anwender zu jedem Zeitpunkt den aktuellsten Entwicklungsstand des Produkts. Visualisiert wird dieser Stand durch Basisfunktionen. Das STEP-Produktmodell (STEP-PM) des rechnerbasierten Programmsystems vereint alle Produktdaten aus den externen Programmen inklusive einer Versionierung. Der Aufbau und Inhalt des STEP-Produktreferenzmodells ist in Bild 5-7 dargestellt. Wie in Kapitel 3.2.4 beschrieben wurde, eignet sich derzeit nur STEP als Basis für das Produktmodell, welches alle Anforderungen im rechnerbasierten Programmsystem abdecken soll. Zur Erstellung eines erweiterten STEP-Produktmodells für bestimmte Anwendungsfälle stehen die in Kapitel 4.4 beschriebenen Vorgehensweisen zur Verfügung. Die Erstellung eines Produktreferenzmodells für Getriebe wird in Kapitel 6 beschrieben.

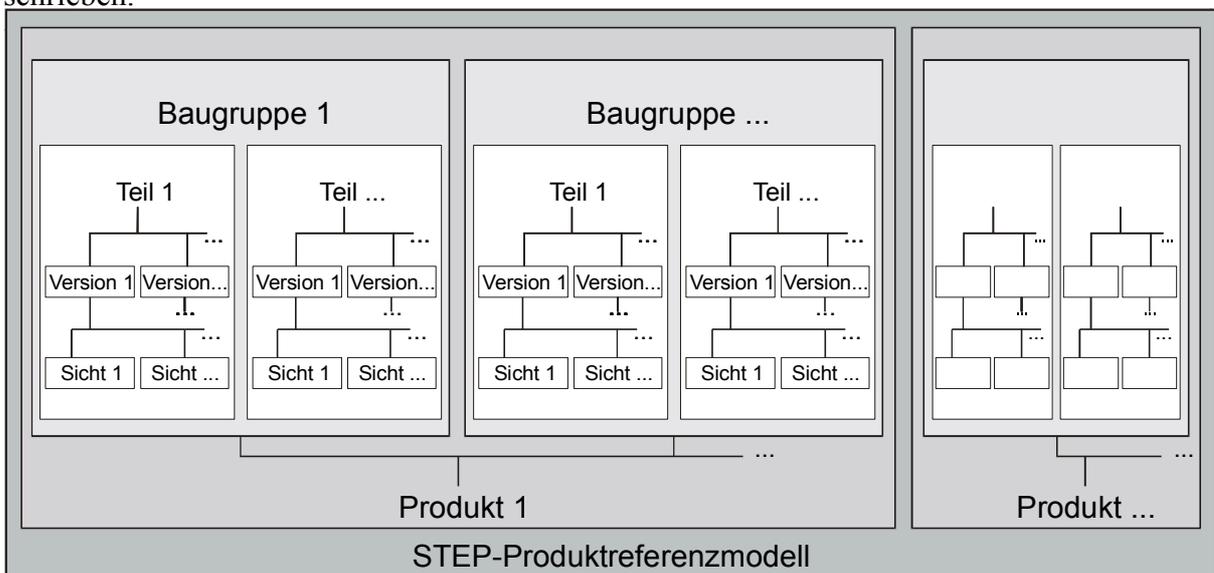


Bild 5-7: Produktstruktur im STEP-Produktreferenzmodell

Die Ordnung im STEP-Produktmodell gestaltet sich hierarchisch nach Teil (Einzelteil, Baugruppe, Produkt), Teileversion und Teilesicht. Zu der Teilesicht gehören die Dokumentenverweise, welche sich auf in der Datenbank für Prozesshistorie abgespeicherte Dokumente beziehen. Direkte Verknüpfungen zwischen STEP-Produktmodell und Datenbank für Prozesshistorie bestehen nicht. Diese beiden Bausteine der Systemarchitektur werden indirekt durch das Prozessmanagement-System verbunden.

5.2.5.2 Datenbank für Prozesshistorie (DBPH)

Alle Dateien und das STEP-Produktmodell werden vom Prozessmanagement-System sortiert und je nach Datenformat und zeitlicher Reihenfolge in der Datenbank für Prozesshistorie abgelegt (Bild 5-8). So wird einerseits eine schnelle Dateifindung bei Bedarf gewährleistet und andererseits eine Archivierung aller anfallenden Daten und Dateien sichergestellt.

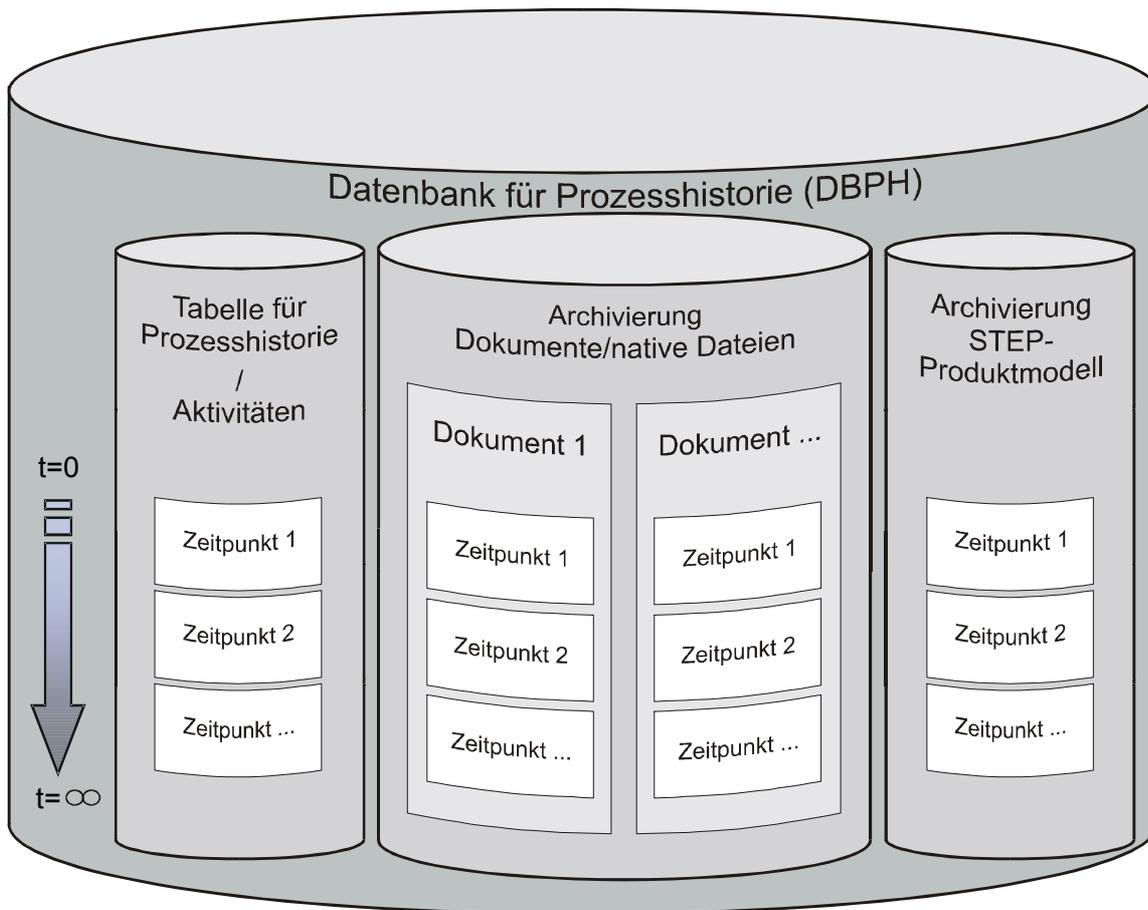


Bild 5-8: Datenbank für Prozesshistorie

Die Datenbank für Prozesshistorie hat drei Hauptaufgaben:

- Chronologische **Speicherung aller Daten und Dateien** sämtlicher externer Anwendungen. Unabhängig von externen Programmen und Datenbanken kann so auf alle erarbeiteten Daten und Dateien zurückgegriffen werden.
- **Speicherung des STEP-Produktmodells** vor jeder Änderung. Die Entwicklung wird dadurch transparent und kann einfach nachvollzogen werden. Der Zugriff auf einzelne Produktentwicklungsstufen ist möglich.
- **Aufzeichnung aller Aktivitäten** des Benutzers und wichtiger Systemaktivitäten in der Tabelle für Prozesshistorie. So kann leichter der Weg der Entwicklung zurückverfolgt und in ein früheres Entwicklungsstadium zurückgesprungen oder es als Ausgangsbasis für neue Projekte verwendet werden.

5.2.6 Prozessmanagement-System (PMS)

Das Prozessmanagement-System ist die Kommandozentrale des rechnerbasierten Programmsystems. Seine Hauptaufgabe ist die Steuerung des Datenflusses. Um Datenredundanz zu vermeiden, wird jeder Daten- und Dateitransfer über das Prozessmanagement-System gere-

gelt. Hierfür verknüpft es sowohl das STEP-Produktmodell mit der Datenbank für Prozesshistorie, wie auch die Basis- und Anwendungsfunktionen mit der Datenablage und den externen Anwendungen. Das Prozessmanagement-System überwacht und dokumentiert alle System- und Benutzeraktivitäten und sendet unter gegebenen Umständen Fehlermeldungen an die grafische Benutzungsoberfläche. Wird vom Anwender aus der grafischen Benutzungsoberfläche ein externes Programm gestartet, beschafft das Prozessmanagement-System die notwendigen Daten, übernimmt gegebenenfalls die Konvertierung der Datenformate und legt die Daten nach Beendigung wieder ab.

Der komplette Wirkungsbereich des Prozessmanagement-Systems ist auf den ersten Blick schwer zu überschauen. Bild 5-9 vermittelt nur die einzelnen Funktionen. Die komplexe Wirkungsweise des Prozessmanagement-Systems im Zusammenspiel mit den anderen Komponenten der Systemarchitektur beschreibt Kapitel 5.3.

Prozessmanagement-System (PMS)											
Aufruf/ Ende externer Program- me	Daten- beschaf- fung aus STEP-PM und DBPH	Daten- rück- führung in STEP- PM	Konvertie- rung der STEP- Daten in externe Dateien	Konvertie- rung der externen Dateien in STEP- Daten	Kontrolle der Benutzer und System- aktivitäten	Rückmel- dung bei Fehlern an Benutzer	Verwal- tung der Prozess- historie	Prozess- über- wachung	Versions- und Reifegrad- verwaltung	Archivie- rung von Daten und Dateien	Anlegen von Dateien in DBPH

Bild 5-9: Prozessmanagement-System

5.2.7 Externe Datenschnittstelle

Die externe Datenschnittstelle ermöglicht einerseits den Datenzugriff von außen auf Daten und Dateien im Systemverbund. Andererseits kann die externe Datenschnittstelle fremde Daten und Dateien in das rechnerbasierte Programmsystem einlesen, verarbeiten oder nur als Dokumentenverknüpfung in der Datenbank für Prozesshistorie ablegen. Das Ein- und Auslesen wird durch das systemneutrale STEP-Format erleichtert, für native Datenformate sind Konverter obligatorisch. Die Anordnung der externen Datenschnittstelle im Gesamtsystem zeigt Bild 5-10, ihre Funktionsweise erklärt Kapitel 5.4.

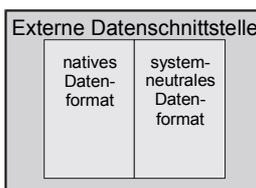


Bild 5-10: Externe Datenschnittstelle

5.3 Zusammenspiel der Systemkomponenten während der Produktentwicklung

Im Folgenden werden die abstrakten Komponenten der Systemarchitektur aus Kapitel 5.2 in einen projekttaulaufbezogenen Zusammenhang gesetzt und aufgezeigt, wie das Zusammenspiel dieser funktioniert. Bild 5-11 zeigt die gesamte Systemarchitektur im Überblick.

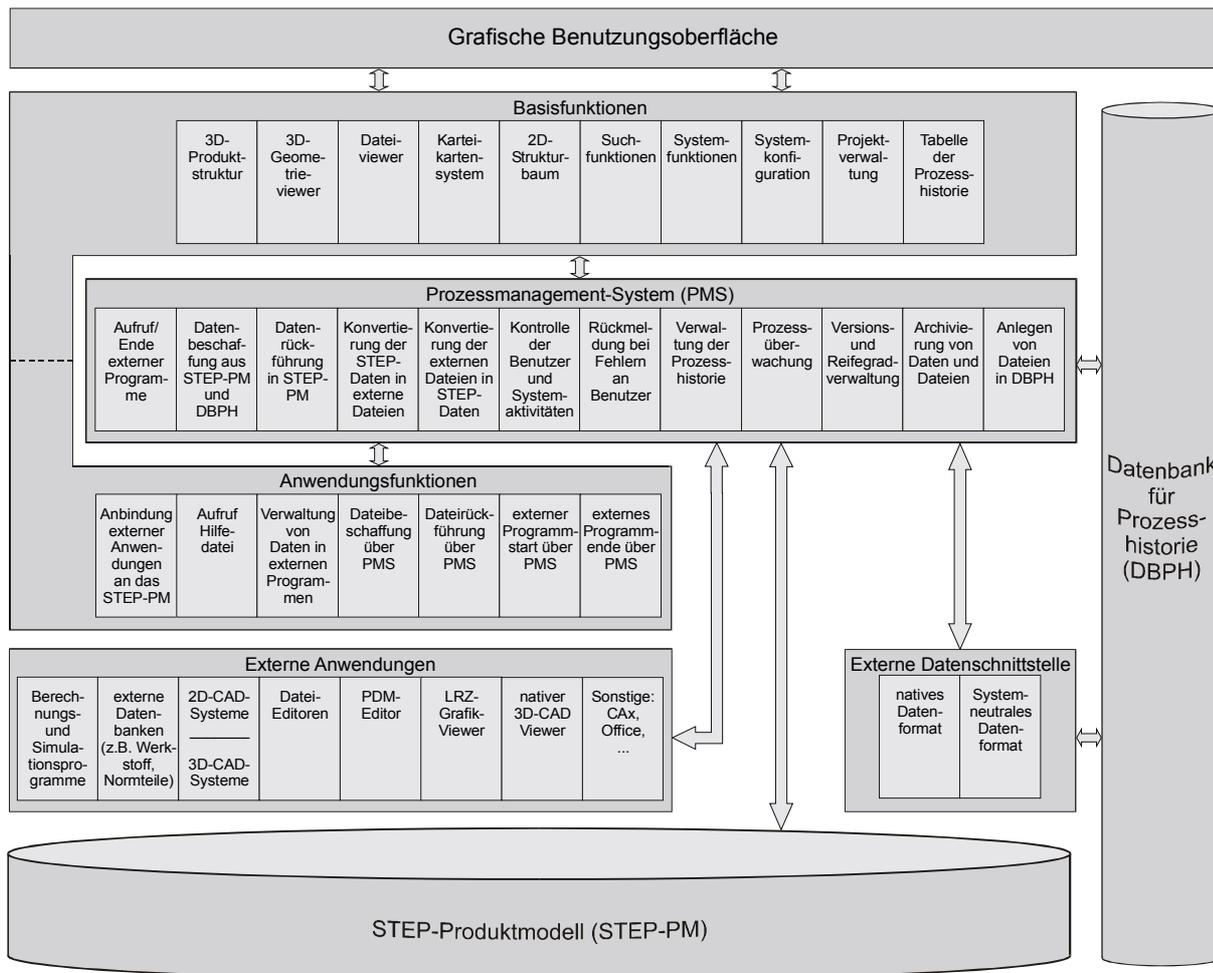


Bild 5-11: Detaillierte Systemarchitektur des rechnerbasierten Programmsystems

Eine Benutzeraktivität in der grafischen Benutzeroberfläche zieht zwangsläufig eine Systemaktivität nach sich. Das rechnerbasierte Programmsystem unterscheidet grob zwischen programminternen und programmexternen Anwendungsprozessen. Die externen Anwendungen dienen der Verwendung und Erzeugung von Produktdaten. Die internen Anwendungen ändern vorhandene Daten nicht, sondern sorgen für Verwaltungs-, Visualisierungs- und Steuerungsvorgänge. Eine Sonderstellung kommt der externen Datenschnittstelle zu. Mit ihr können Daten und Dateien unabhängig vom Rest des Programmsystems ausgelesen bzw. eingefügt oder verändert werden.

Im Mittelpunkt der Systemaktivitäten steht das Prozessmanagement-System. Es hat die Aufgabe, den Datenfluss zu steuern und zu überwachen. Bild 5-12 vermittelt einen groben Überblick über den Datenfluss des Systems.

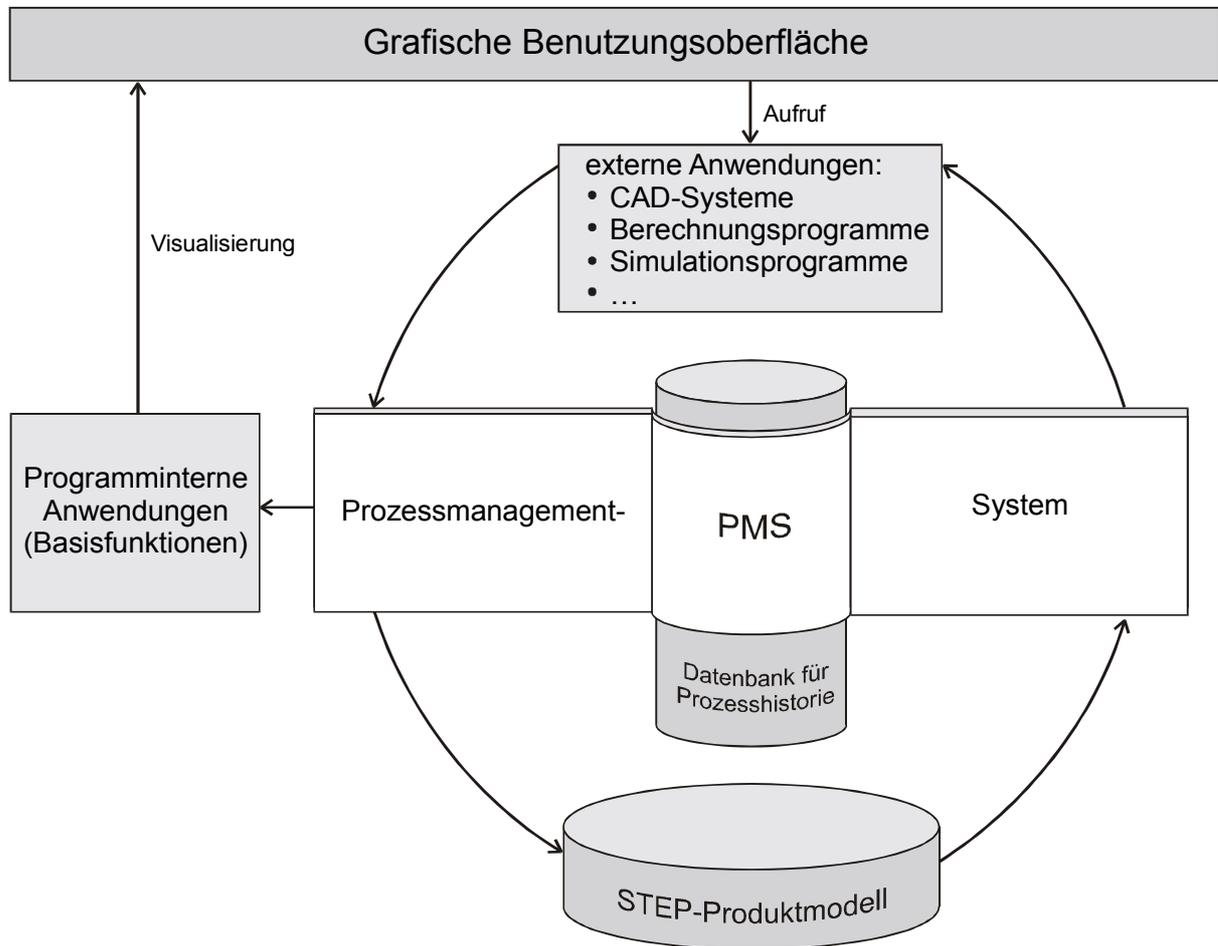


Bild 5-12: Übersicht über den Datenfluss

5.3.1 Kommunikation innerhalb des rechnerbasierten Programmsystems

Unter Kommunikation innerhalb des rechnerbasierten Programmsystems ist das Zusammenspiel mehrerer Funktionen innerhalb der Komponenten der Systemarchitektur zu verstehen. Eine in der Benutzungsoberfläche begonnene Aktivität setzt im Hintergrund laufende Systemaktivitäten in Gang. Die Aktivität beschreibt die Bedienung des rechnerbasierten Programmsystems während der Produktentwicklung. Dabei steuert und überwacht das Prozessmanagement-System den Datenfluss. Der interne Datenfluss hat die Aufgabe, die Basisfunktionen mit Daten zu versorgen.

Die Basisfunktionen sind maßgebend für die internen Dienstleistungen des rechnerbasierten Programmsystems und werden nachfolgend beschrieben.

5.3.1.1 Gliederung der Basisfunktionen

Die Basisfunktionen gliedern sich ihrer Funktion nach in produktmodellbezogene, prozessbezogene und systembezogene Basisfunktionen. Bild 5-13 zeigt die Zuordnung der einzelnen Basisfunktionen zu den genannten Gliederungspunkten.

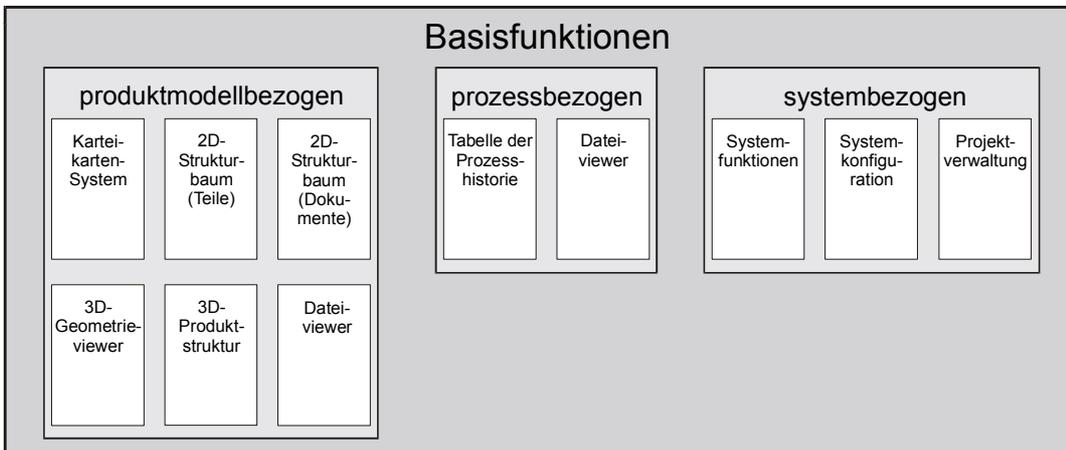


Bild 5-13: Aufteilung der Basisfunktionen

Der Grund für die Dreiteilung der Basisfunktionen ist zum einen ihre Aufgabe zur Systemsteuerung, zum anderen die Visualisierung von Prozess- und Produktdaten. Da sich die systembezogenen Basisfunktionen (drucken, speichern, ...) des rechnerbasierten Programmsystems von der gängigen Systemsteuerung nicht unterscheiden, werden diese als bekannt vorausgesetzt und nicht weiter behandelt.

Bei der Datenbeschaffung für die Visualisierung der Daten oder Dateien in der grafischen Benutzungsoberfläche ist der Ablageort im STEP-Produktmodell oder der Datenbank für Prozesshistorie und das damit verbundene Datenformat entscheidend. Deshalb werden produktmodell- und prozessbezogene Basisfunktionen benötigt.

5.3.1.2 Interner Produktmodelldatenfluss zur Visualisierung von Produktdaten

Die produktmodellbezogenen Basisfunktionen informieren den Benutzer über die Produktstruktur und den aktuellen Produktentwicklungsstand und werden deshalb auch als Visualisierungsfunktionen bezeichnet. Eine Übersicht über den produktmodellbezogenen Datenfluss liefert Bild 5-14.

Der jeweils aktuelle Datenbestand des Produkts ist im STEP-Produktmodell abgelegt. Soll darauf zugegriffen werden, wird das STEP-Produktmodell durch die Anwendungsfunktion „Dateibeschaﬀung“ kopiert und an das Prozessmanagement-System gesendet. Dieses durchsucht die Datei nach den zur Visualisierung benötigten Daten, beschafft diese und konvertiert sie in ein von den Basisfunktionen lesbares internes Datenformat. Dieses Datenformat wird durch die Basisfunktionen innerhalb der Benutzungsoberfläche grafisch umgesetzt. Die möglichen Darstellungsformen sind in Bild 5-6 enthalten und werden in Kapitel 5.3.1.4 beschrieben.

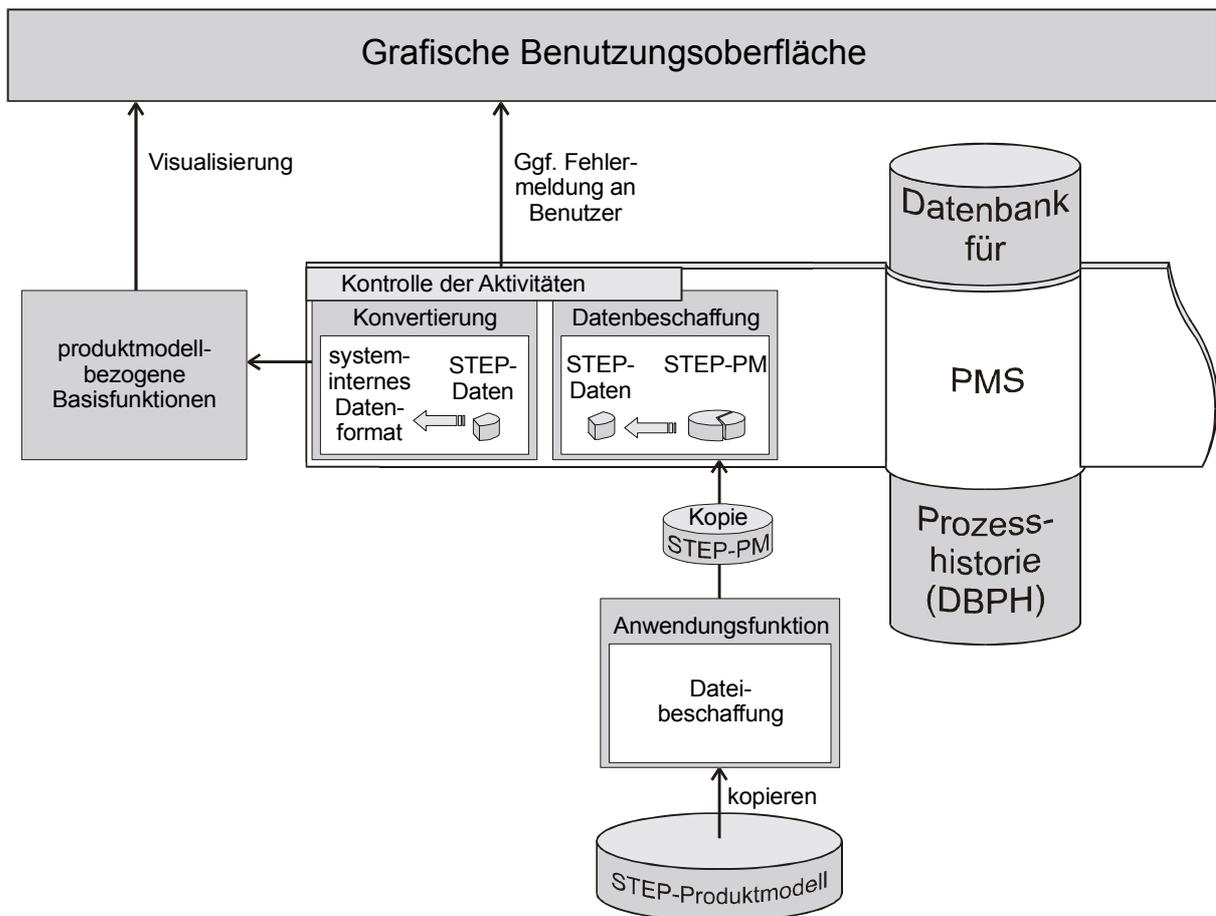


Bild 5-14: Produktmodelldatenfluss innerhalb des rechnerbasierten Programmsystems

Die Kontrolle der Systemaktivitäten übernimmt das Prozessmanagement-System selbst. Ist eine Aktivität fehlerbehaftet, meldet das Prozessmanagement-System dies der Benutzeroberfläche.

Der programminterne Datenfluss verläuft unidirektional. Die Daten werden aus dem STEP-Produktmodell kopiert und verfallen, wenn die Basisfunktionen beendet werden. Das STEP-Produktmodell wird dabei nicht verändert, da die Daten lediglich visualisiert werden. So werden immer die aktuellsten Daten für die Basisfunktionen verwendet.

5.3.1.3 Interner Prozessdatenfluss zur Visualisierung der Prozessdaten

Der Prozessdatenfluss ist dem Produktmodelldatenfluss in weiten Teilen ähnlich (Bild 5-15). Auch hier ist der Datenfluss unidirektional. Statt des STEP-Produktmodells dient die Datenbank für Prozesshistorie als Quelle für die Daten und Dateien. Dort sind verschiedene Datenformate abgelegt.

Die Anwendungsfunktion „Dateibesorgung“ übernimmt wiederum eine Kopie der Datenbank für Prozesshistorie. Das Prozessmanagement-System durchsucht die Datenbank und beschafft die benötigte Datei. Nach der Bestimmung des Datenformats entscheidet das Prozessmanagement-System, ob die Daten konvertiert werden müssen oder ob diese sofort zur

Visualisierung an die prozessbezogenen Basisfunktionen weitergereicht werden können. Ab hier läuft der Prozessdatenfluss synonym zum Produktmodelldatenfluss aus Kapitel 5.3.1.2.

Das Prozessmanagement-System übernimmt eine weitere wichtige Aufgabe, indem es eine interne Archivierung aller Dateien und des STEP-Produktmodells während dem Entwicklungsprozess in die Datenbank für die Prozesshistorie vornimmt.

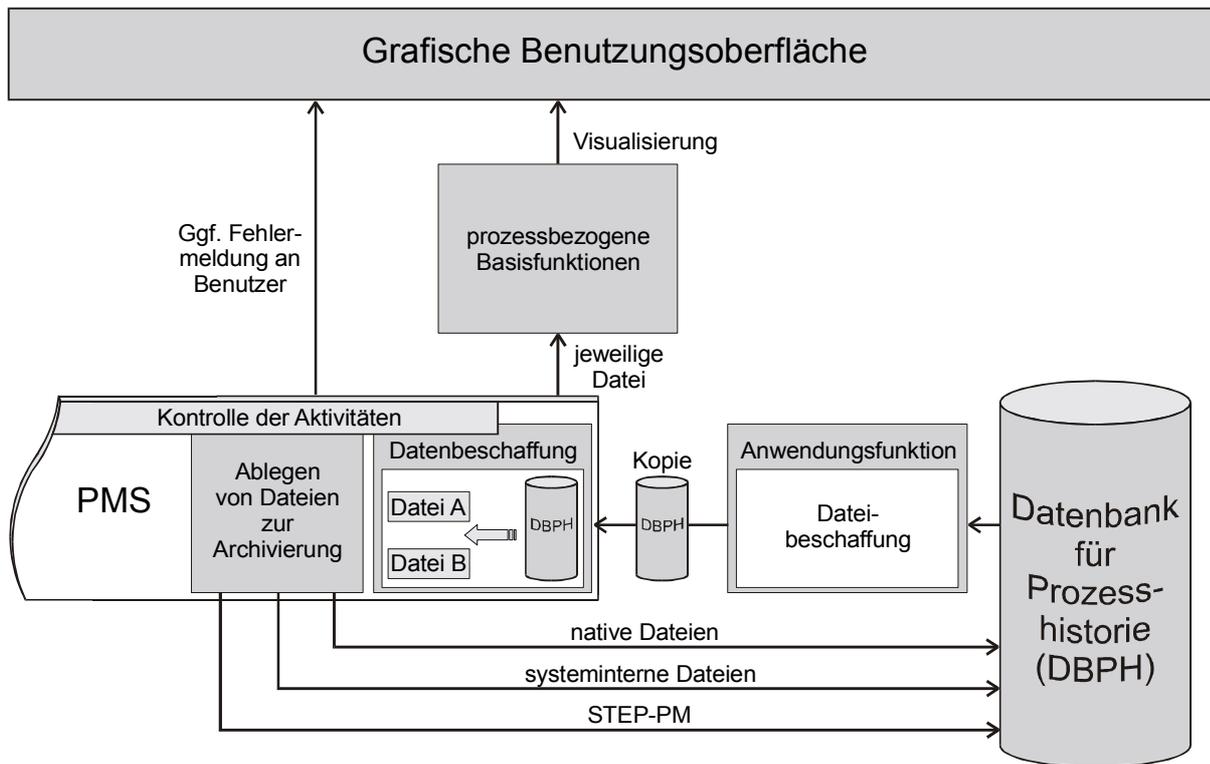


Bild 5-15: Prozessdatenfluss innerhalb des rechnerbasierten Programmsystems

5.3.1.4 Visualisierungsmöglichkeiten von Dateien, Produkt- und Prozessdaten

Unter Visualisierung wird die Transformation von Informationen aus der ursprünglichen Anordnungsweise in eine dem Menschen verständliche Form verstanden. Handelt es sich um einfach verknüpfte Informationen, sind diese im allgemeinen leicht nachzuvollziehen und die Visualisierung gestaltet sich ebenfalls einfach. Bei komplexen Zusammenhängen, wie beispielsweise der versionierten Produktstruktur in Bezug zu ihren Dokumenten, ist die Visualisierung der Information schwierig. Ziel des rechnerbasierten Programmsystems ist es, mit Visualisierungstechniken komplexe Zusammenhänge von Produkt und Prozess darzustellen.

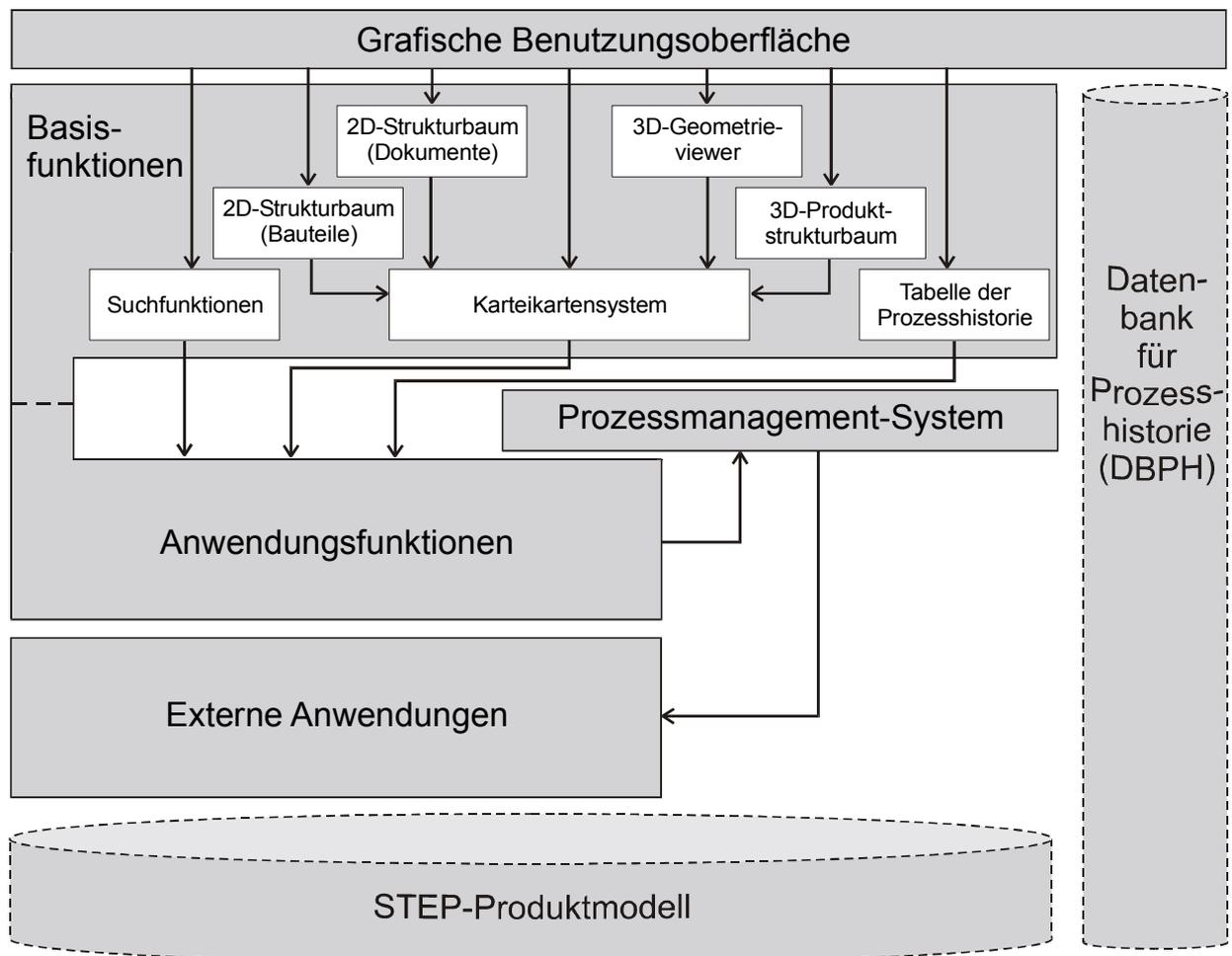


Bild 5-16: Zusammenspiel der Basisfunktionen

Die in Bild 5-16 abgebildeten, sieben verschiedenen Visualisierungsformen sind miteinander gekoppelt, um ein Höchstmaß an Übersichtlichkeit zu gewährleisten:

- Das **Karteikartensystem** vereint alle Informationen zu einem Bauteil, einer Baugruppe und des gesamten Produkts in Tabellen und Text. Ziel ist es, alle Produktdaten in verschiedenen Karteikarten darzustellen. Aus den Karteikarten können die anderen Visualisierungsfunktionen aufgerufen werden und umgekehrt.
- Der **3D-Produktstrukturbaum** ermöglicht die Darstellung der zweidimensionalen Produktstruktur und die Zusammenhänge der einzelnen Bauteile und Baugruppen mit Hilfe der dritten Dimension. Ein mausgesteuertes Navigieren durch den 3D-Produktstrukturbaum ermöglicht es, die funktionalen Bauteilzusammenhänge zu erfassen.
- Der **3D-Geometrieviewer** („Viewer“ werden in [N4][N5] genauer spezifiziert) stellt die Anordnung der Bauteile zur Produktstruktur dar, je nach Wunsch bewegt oder unbewegt.
- Die **2D-Strukturbäume** für Dokumente und Produkte ermöglichen einen Überblick über alle Bauteile, ihre Versionen und Sichten und die zugehörigen Dokumente, die ebenfalls versioniert werden.

- Die **Tabelle der Prozesshistorie** zeichnet unter Mithilfe des Prozessmanagement-Systems den Entwicklungsprozess auf, nimmt somit eine Archivierung vor, und ermöglicht den Schritt zurück zu einer früheren Entwicklungsstufe. Nach Ende eines Projekts kann die Tabelle der Prozesshistorie langzeitarchiviert werden, was nach [H8] und [K14] von steigender Bedeutung ist.
- Mit Hilfe der **Suchfunktion** hat der Anwender die Möglichkeit, Daten und Dateien im rechnerbasierten Programmsystem zu suchen.
- Der **Dateiviewer** gehört nicht zum innovativen Zusammenspiel der Basisfunktionen des rechnerbasierten Programmsystems. Er ermöglicht lediglich die Betrachtung aller Dateiformate in ihrer ursprünglichen Form.

5.3.2 Kommunikation mit externen Anwendungen

Die eigentliche Weiterentwicklung des Produkts geschieht mit Hilfe der externen Anwendungen, welche Veränderungen am STEP-Produktmodell vornehmen. Für das rechnerbasierte Programmsystem selbst ist der Vorgang innerhalb der externen Anwendungen nicht von Bedeutung. Es ist lediglich wichtig, welche Daten für den Programmstart benötigt werden, ob hierfür ein Konverter eingeschaltet werden muss, und welche Daten das externe Programm nach dessen Ende zurückliefert.

Auch bei den externen Anwendungen übernimmt das Prozessmanagement-System die Kontrolle der Systemaktivitäten und die Mitteilung eventueller Fehler. Als zentrale Einheit des Programmsystems kommt ihm bei der Veränderung der Daten eine weitere Aufgabe zu. Das Prozessmanagement-System verfolgt die Prozesshistorie und legt die notwendigen Daten in der Datenbank für Prozesshistorie ab.

Das externe Programm wird aus der Benutzungsoberfläche unabhängig vom benötigten Datenformat gestartet. Die Anwendungsfunktion „Dateibeschafter“ kopiert das STEP-Produktmodell und sendet es an das Prozessmanagement-System. Es wird nach einem externen Datenfluss mit systemneutralem (Kapitel 5.3.2.2) oder nativem Datenformat (Kapitel 5.3.2.3) unterschieden. Durch eine Daten- und Dateirückführung nach einer Beendigung der externen Anwendung liegt ein aktualisiertes STEP-Produktmodell vor. Falls das System bei diesem Vorgang feststellt, dass bei den rückgeführten Daten Unstimmigkeiten auftreten, versucht das rechnerbasierte Programmsystem den Fehler zu identifizieren und es wird eine Anwenderentscheidung in Bezug auf das weitere Vorgehen verlangt. Dieser Vorgang muss vom Anwender sehr sensibel durchgeführt werden, da veränderte Daten oder leichtfertig gelöschte Daten einen weitreichenden Einfluss haben können. Deshalb sind Widersprüche bei den Daten nur auf falsche Eingaben oder eine nicht korrekte Integration der externen Anwendung in das rechnerbasierte Programmsystem zurückzuführen.

5.3.2.1 Gliederung der externen Anwendungen

Die in das rechnerbasierte Programmsystem einbindbaren externen Anwendungen lassen sich nach Funktion und Verwendung im Entwicklungsprozess unterscheiden (Bild 5-17). Die externen Anwendungen bleiben unverändert, der Datenaustausch erfolgt ausschließlich über

Ein- und Ausgabedateien. Die einzelnen Anwendungen werden nicht hier, sondern erst in Kapitel 7.2 am Beispiel der Getriebe näher erläutert.

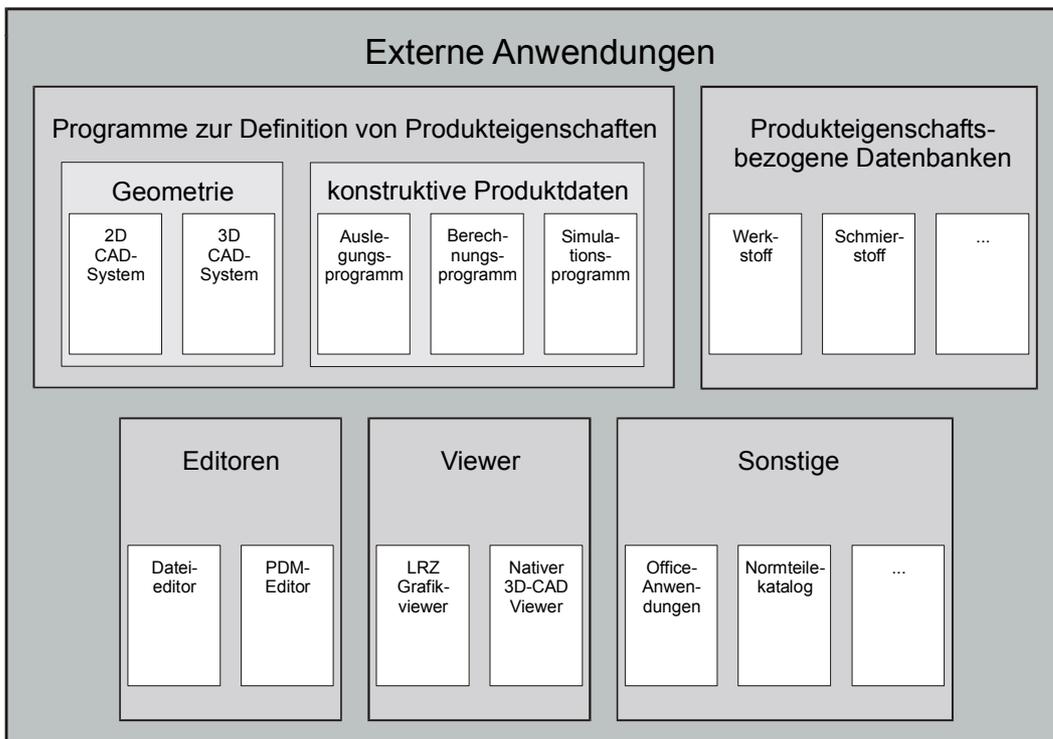


Bild 5-17: Gliederung der externen Anwendungen

Für die Einbindung eines externen Programms in das rechnerbasierte Programmsystem ist es von Vorteil, wenn es das systemneutrale STEP-Format verwendet, da in diesem Fall kein Konverter für das rechnerbasierte Programmsystem erstellt werden muss. Die externe Kommunikation des rechnerbasierten Programmsystems wird in den Datenaustausch mit systemneutralem und nativem Datenformat unterschieden.

Den externen Datenfluss mit dem systemneutralen STEP-Datenformat und vergleichend mit einem nativem Datenformat zeigt Bild 5-18. Der Mechanismus wird nachfolgend beschrieben.

5.3.2.2 Externer Datenfluss mit systemneutralem Datenformat

Das Prozessmanagement-System beschafft die benötigten Daten aus dem STEP-Produktmodell und sendet sie im STEP-Format mit Hilfe der Anwendungsfunktion „externer Programmstart“ an die externe Anwendung. Weiterhin legt es eine Kopie dieser „ausgeschnittenen“ STEP-Datei in der Datenbank für Prozesshistorie zur Archivierung ab und vervollständigt die Tabelle für die Prozesshistorie.

Nach der Beendigung des externen Programms und der Ausführung der daraufhin aktivierten Anwendungsfunktion „externes Programmende“ werden die geänderten STEP-Daten zurück an das Prozessmanagement-System geschickt. Dies übernimmt über einen aufwendigen Integrationskontrollmechanismus die Rückführung der Daten in das zuvor ausgeschnittene

STEP-Produktmodell. Davor wird eine Kopie der geänderten Datei in der Datenbank für Prozesshistorie abgelegt und die Tabelle für Prozesshistorie vervollständigt.

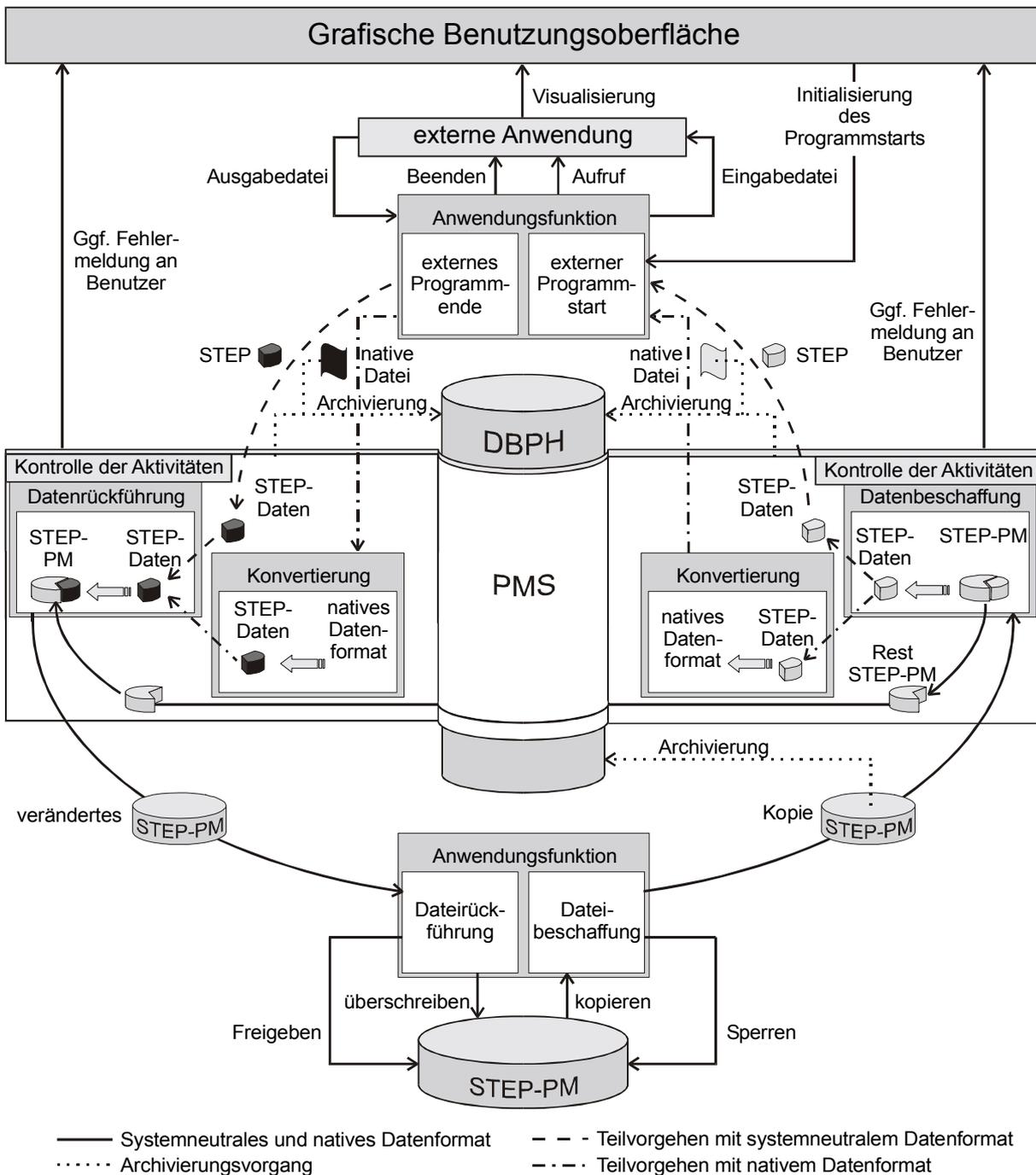


Bild 5-18: Datenfluss mit externen Anwendungen über ein systemneutrales oder natives Datenformat

Sind die Systemaktivitäten fehlerfrei beendet, überschreibt die Anwendungsfunktion „Dateirückführung“ das veraltete STEP-Produktmodell und der Kreis des externen Datenflusses ist geschlossen. Ist eine Aktivität fehlerbehaftet, findet die „Dateirückführung“ nicht statt und

das alte STEP-Produktmodell bleibt erhalten. Der Anwender oder das System kann nun den Fehler beheben und über weitere Aktivitäten entscheiden.

Während des gesamten Kreislaufs bleibt das STEP-Produktmodell zur Vermeidung einer redundanten Dateneingabe für andere Anwendungen gesperrt und wird erst bei der Datenrückführung wieder freigegeben.

5.3.2.3 Externer Datenfluss mit nativem Datenformat

Der externe Datenfluss mit nativem Datenformat ist dem Datenfluss mit systemneutralem Datenformat sehr ähnlich. Zusätzlich werden Zwischenschritte für die Konvertierung und Rückkonvertierung der Daten in das systemneutrale Format des STEP-Produktmodells eingefügt (Bild 5-18).

Wird vom externen Programm ein spezielles Datenformat angefordert, legt das Prozessmanagement-System eine Kopie des gesamten STEP-Produktmodells in der Datenbank für Prozesshistorie ab, da es von einer Änderung dessen ausgeht. So wird eine lückenlose Prozesshistorie gewährleistet. Der Datenbeschaffung folgt die Konvertierung der STEP-Daten in das native Datenformat. Bevor diese Daten an die externe Anwendung weitergegeben werden, wird eine Kopie der nativen Datei in der Datenbank für Prozesshistorie abgelegt und die Tabelle für Prozesshistorie vervollständigt. Das Gleiche geschieht mit der geänderten oder neu erstellten nativen Datei nach Beendigung des externen Programms. Nach der Rückkonvertierung und der Datenrückführung im Prozessmanagement-System schließt sich der Kreis des externen Datenflusses wieder mit der Überschreibung des veralteten STEP-Produktmodells.

5.4 Kommunikation mit separaten externen Anwendungen

Bei der Verwendung des rechnerbasierten Programmsystems können je nach vorherrschender Entwicklungsumgebung und angeschlossenen informationstechnischen Systemen jederzeit sowohl die Daten des STEP-Produktmodells, als auch die Daten und Dateien der einzelnen Entwicklungsphasen aus der Datenbank für die Prozesshistorie isoliert und an separate externe Anwendungen übergeben werden. Unter separater externer Anwendung ist ein in die informationstechnische Struktur eingebundenes, aber vom rechnerbasierten Programmsystem separat laufendes Programm zu verstehen, welches nicht als externe Anwendung in das rechnerbasierte Programmsystem integriert ist. Beispiele für separate externe Anwendungen und der Mechanismus sind in Bild 5-19 aufgeführt.

Über die externe Datenschnittstelle können Daten, Datenformate und Dateien ein- oder ausgelesen, verändert und gelöscht werden. Da das Programmsystem hier keine Kontrolle über die Aktivitäten des Benutzers der externen Datenschnittstelle hat, muss der Zugang geschützt werden.

Der Anwender benutzt die Suchfunktion des Prozessmanagement-Systems um zu den sowohl im STEP-Produktmodell als auch in der Datenbank für Prozesshistorie abgelegten Daten zu gelangen. Die Dateibesorgung, Datenbeschaffung, Änderung und Rückführung übernehmen die Anwendungsfunktionen und das Prozessmanagement-System. Möchte der Anwender Da-

ten oder Dateien über die externe Datenschnittstelle im STEP-Produktmodell oder in der Datenbank für Prozesshistorie anlegen, benutzt er dazu ebenfalls Anwendungsfunktionen und Prozessmanagement-System.

Die wesentlichen Unterschiede zum Datenfluss mit in das rechnerbasierte Programmsystem eingebundenen externen Anwendungen (Bild 5-18) sind die nicht mögliche Konvertierung der Daten und die ausbleibende Systemkontrolle durch das Prozessmanagement-System. Bild 5-19 zeigt den Weg des Datenflusses bei Verwendung der externen Datenschnittstelle.

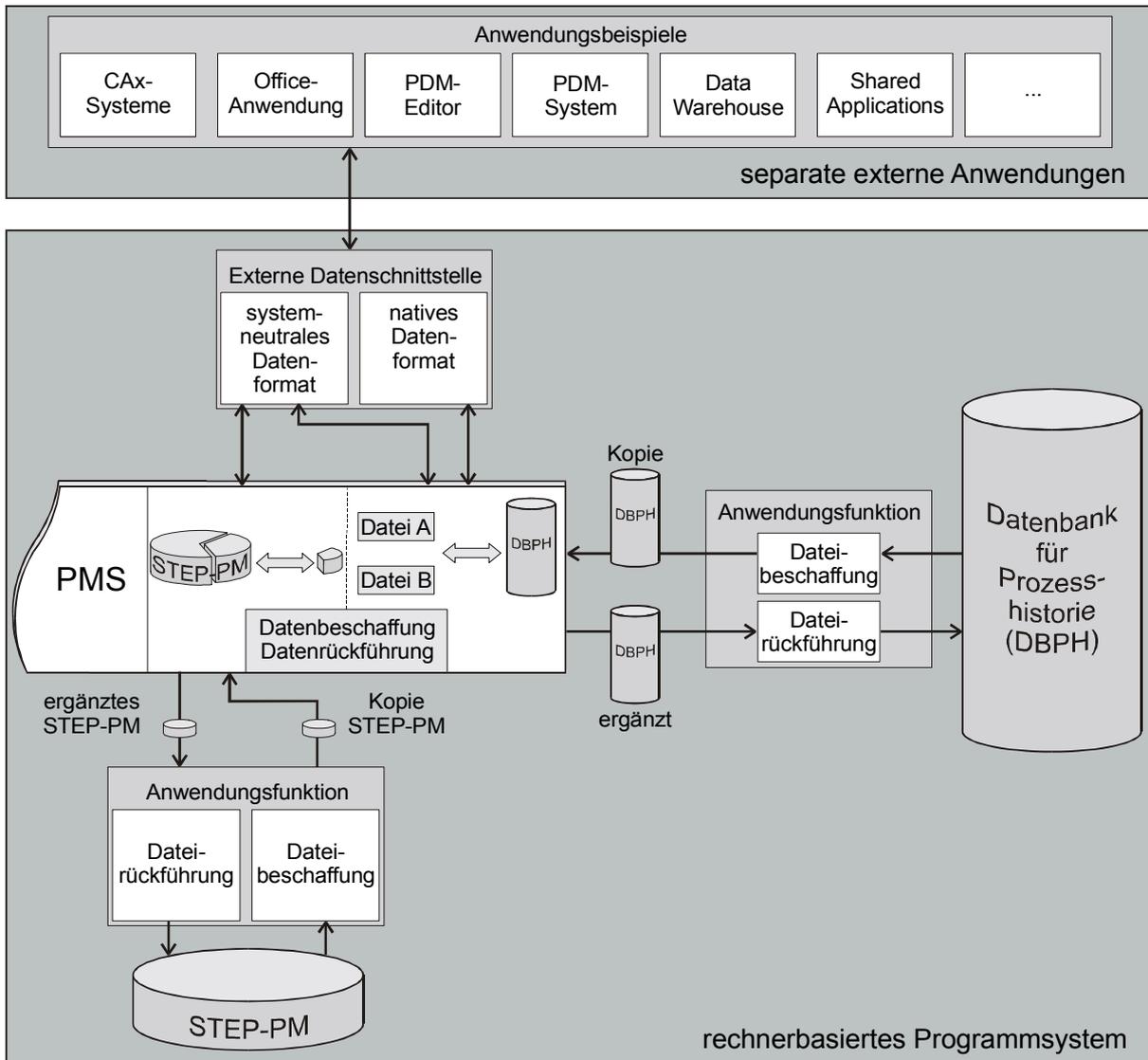


Bild 5-19: Datenaustausch über die externe Datenschnittstelle

6 Entwicklung eines systemneutralen Produktmodells für Getriebe

In der Getriebeentwicklung sind viele Programme im Einsatz, die selbst bei ähnlichem Inhalt unterschiedliche Formate für die Dateneingabe und Datenausgabe benötigen. Dadurch wird der redundanz- und fehlerfreie Datenaustausch erschwert, da eine Umwandlung in andere Dateiformate oder das manuelle Auslesen der benötigten Daten erforderlich ist. Ein systemneutrales Datenformat in Form eines Produktreferenzmodells ist daher notwendig, um alle relevanten Daten der Prozesskette Getriebekonstruktion in einem einheitlichen Format zur Verfügung zu stellen. Zur Erreichung einer Offenheit für unterschiedliche Anwendungen ist die Beschreibung eines systemneutralen Produktreferenzmodells für Getriebe von Vorteil. Dabei werden die speziellen Anforderungen der Getriebekonstruktion berücksichtigt und es wird die Erweiterung um ergänzende Produktdaten sichergestellt.

6.1 Datentechnischer Inhalt des Produktmodells

Das Produktreferenzmodell deckt alle relevanten Daten aus den Bereichen der Getriebekonstruktion, -berechnung und -simulation ab. Dazu gehören die produktbeschreibenden Daten (Geometrie, Produktstruktur, ...; vergleiche Kapitel 3.6.2.2-3.6.2.4) und die speziell für die Berechnung und Simulation erforderlichen Daten. Die Ergebnisse aus durchgeführten Berechnungen und Simulationen können ebenfalls beschrieben werden. Eine Zuordnung der Ein- und Ausgabedateien zu den Objekten wird über den STEP-Dokumentenmechanismus (Kapitel 3.6.2.5) erreicht.

6.1.1 Versionen und Sichten

Die Unterscheidung des beschriebenen Produkts, welches ein einzelnes Bauteil, eine Baugruppe oder das gesamte Getriebe sein kann, nach verschiedenen Versionen sowie die Definition mehrerer Sichten, welche den Versionen zugeordnet sind, ist für die Datenstrukturierung und deren Management von Vorteil. Die Sichten (Bild 3-9) filtern den gesamten abgebildeten Dateninhalt nach entsprechenden Verwendungszwecken oder verschiedenen Abschnitten im Produktlebenszyklus (Entwurf, Konstruktion, Berechnung/Simulation, ...).

6.1.2 Bauteile und Produktstruktur

Den Kern des Datenmodells bilden Maschinenelemente, welche ein Getriebe typischerweise enthält, und ihre Bauteilstruktur zueinander. Im einzelnen sind folgende Maschinenelemente definiert:

- Fundament und Gehäuse
- Lager (Gleit- und Wälzlager)

- Welle
- Dichtung
- Zahnrad (Stirnrad, Kegelrad, Schnecke und Schneckenrad, Schraubrad)
- Räder, Scheiben, Kupplungen, Bremsen und Welle-Nabe-Verbindungen zur Kraftübertragung
- Umschlingungsmittel und Riemen
- Herstellwerkzeug für Zahnräder
- Beliebig vom Anwender zu definierendes Maschinenelement (*User_defined_object*)

Das Werkzeug wird als Sonderfall wie ein Bauteil behandelt. Dadurch können neben den eigentlichen Herstellwerkzeugdaten auch spezielle Informationen, die sich auf die Paarung zwischen Werkzeug und Zahnrad beziehen, abgebildet werden.

Neben der Definition von Baugruppen, die Bauteile zu logischen Einheiten gruppieren, besteht die Möglichkeit zwischen den Maschinenelementen, unabhängig ihrer hierarchischen Baugruppenzugehörigkeit, funktionale Zusammenhänge zu definieren. Typische Bauteil- und Baugruppenbeziehungen, die neben einer normalen Baugruppenstruktur definiert werden können, sind zum Beispiel:

- Gehäuse - Lager
- Gehäuse - Dichtung
- Welle - Lager
- Welle - Zahnrad
- Zahnradpaare (Stirn- und Kegelradgetriebe, Planetengetriebe, ...)
- Zahnrad - Werkzeug
- Getriebe - Schmierstoff

Durch die eindeutige funktionelle Charakterisierung der enthaltenen Maschinenelemente und ihrer Beziehungen werden bereits Informationen über die funktionalen Zusammenhänge abgelegt, die unabhängig von spezifischen Benennungen der Anwender sind. Für die verschiedenen Typen besteht darüber hinaus die Möglichkeit eigene Elemente zu definieren, die eine ähnliche Grundfunktionalität erfüllen. Das Produktreferenzmodell ist jederzeit offen für Erweiterungen neuer Getriebeobjekte. Die Produktstruktur bestehend aus Bauteilen, Baugruppen und funktionalen Beziehungen ist die Basis für die Anbindung weiterer relevanter Daten.

6.1.3 Geometrie und Topologie

Das Produktreferenzmodell kann alle gängigen Geometriemodellierungen beschreiben, die auch im AP214 enthalten sind, und deckt den datentechnischen Inhalt üblicher CAD-Systeme vollständig ab. Dazu gehören Kanten-, Flächen- und Volumenmodelle. Verschiedene Geometrien können das gleiche Objekt für unterschiedliche Zwecke aus anderer Sicht beschreiben. So ist es beispielsweise möglich, für ein Getriebe eine CAD-Geometrie in Form eines

Volumenmodells mit allen Fasen, Keilwellen und anderen Detaillierungen zu definieren und parallel dazu ein einfaches datenreduziertes Ersatzmodell als CSG¹-Geometrie für die Berechnungsprogramme bereitzustellen, die in der Regel solche CAD-Details nicht benötigen und verarbeiten können.

6.1.4 Produkteigenschaften und Getriebedaten

Neben der Beschreibung der Produktstruktur, der Geometrie und der Gestalt des Produkts können weitere Informationen abgebildet werden. Dazu zählen neben Materialangaben, Oberflächeneigenschaften oder speziellen Kennwerten auch Angaben über die verschiedenen Berechnungsverfahren und weitere produktbeschreibende Dokumente. Als Quelle für diese zusätzlichen Daten dienen Lehrbücher (z.B. [B4][N2][N3]) und der datentechnische Inhalt der bestehenden Berechnungs- und Simulationsprogramme [A14][D3][G6][S3][S20][S21]. Für die sich daraus ergebende Datenmenge ist deren strukturierte Abbildung im Produktreferenzmodell notwendig. Folgende Informationen spielen eine besonders wichtige Rolle:

- **Materialangaben:** Im Produktreferenzmodell zählen zu den Materialien die Werk- und Schmierstoffe, die aus Sicht der Datenhandhabung gleich zu werten sind. Es können einzelne Materialkennwerte für entsprechende Bauteile definiert werden. Des Weiteren sind Materialien als eigene Objekte definiert. Durch eine Zuordnung relevanter Materialkennwerte entsteht eine Materialdatenbank. Für die Zuteilung der Materialdaten an ein Bauteil reicht in diesem Fall die Angabe des Materials als Grundwerkstoff für das Bauteil aus. Die Attribute des Materials werden aufgrund der Vererbung mit übergeben. Durch diese zwei Definitionsmöglichkeiten der Materialeigenschaften eines Objektes bleibt das Produktmodell sehr flexibel. Auf die gleiche Weise ist die Beschreibung von Schmierstoffen und deren Zuordnung zu Objekten möglich.
- **Oberflächeneigenschaften:** Oberflächenangaben und Kennwerte der Oberflächenbeschaffenheit werden durch das Produktreferenzmodell erfasst. Diese Informationen werden über die Gestalt den entsprechenden Bauteilen zugeordnet, um für verschiedene Bereiche eines Bauteils unterschiedliche Oberflächeneigenschaften definieren zu können. Die Härteangaben werden nach unterschiedlichen Messverfahren unterschieden.
- **Getriebe-Kennwerte:** Die Getriebe-Kennwerte umfassen alle Ein- und Ausgabegrößen der wichtigen Berechnungs- und Simulationsprogramme, stellen den größten Teil des Produktreferenzmodells dar und unterteilen sich in:
 - **Anwendungsdaten**, die insbesondere die für die Berechnungen wichtigen Informationen über Belastungen im Betrieb und damit auftretende Kräfte, Momente, Verschiebungen und Verformungen und daraus resultierende Lebensdauern, Wirkungsgrade und Sicherheiten enthalten. Zu ihrer besseren Untergliederung sind folgende Oberklassen für die Anwendungsdaten definiert: Servicedaten, Antriebsdaten, Kräfte und Momente, Pressungen und Druckverteilungen, Verformungen und Durchbiegungen, temperatur-

¹ CSG steht für „Constructive Solid Geometry“ und beschreibt Volumenmodelle, die sich aus einfachen geometrischen Körpern durch Boolesche Verknüpfungsvorgänge ergeben.

spezifische Daten, Sicherheitsangaben, Wirkungsgrad und Verluste, Verschleiß und Umgebungsdaten.

- **Parameter**, die folgenden Klassen zuzuordnen sind: Eingriffsdaten, geometriebezogene Kennwerte, Tragfähigkeitsdaten, Steifigkeiten, Dynamikdaten und Zahnrad-Herstellungsdaten.
- **Berechnungsverfahren**: Im Bereich der Getriebeentwicklung gibt es zur Berechnung ein und desselben Kennwertes, z.B. der Sicherheit, eine große Anzahl parallel zu wertender Berechnungsverfahren, die in unterschiedlichen Normungsausschüssen oder Anwendungsgremien entstanden sind. Um eine Unterscheidung gleicher Werte nach unterschiedlichen Berechnungsverfahren vorzunehmen, können für die definierten Produktdaten und Kennwerte, sofern sie für Berechnungen relevant sind oder diesen entstammen, die verwendeten Berechnungsverfahren explizit angegeben werden. In der Anwendung kann so beispielsweise ein direkter Vergleich unterschiedlicher Berechnungsverfahren erzielt werden.

6.1.5 Externe Dokumente

Das Produktreferenzmodell erlaubt die Hinterlegung von Informationen über jede Art von externen Datenquellen im Sinne eines Dokuments. Damit sind insbesondere die Ein-, Ausgabe- und Grafikdateien der Berechnungs- und Simulationsprogramme gemeint, auf die aus dem Produktmodell heraus referenziert werden kann.

6.1.6 Verknüpfung von Produktstruktur und Produkteigenschaften

Die Gruppierung der Kennwerte in Kapitel 6.1.4 orientiert sich nicht an den relevanten Bauteilen oder Bauteilstrukturen, sondern ist nach funktionalen Gesichtspunkten gestaltet. Dadurch können die Daten flexibler zugeordnet werden. Durch die eindeutige Bezeichnung der Kennwerte werden Falschinterpretationen ausgeschlossen und Fehlzuweisungen minimiert. Die verschiedenen Produkteigenschaften und Kennwerte können folgenden Objekten zugeordnet werden:

- Bauteilen (inklusive Werkzeug)
- Baugruppen
- Bauteilbeziehungen
- Baugruppenbeziehungen
- Materialien (Werkstoffe, Schmierstoffe)

Die Produktstruktur ermöglicht zusammen mit der Geometrie, den externen Dokumenten und den verknüpften Produkteigenschaften eine vollständige Abbildung der Produktdaten und stellt den Umfang des Produktmodells für Getriebe dar.

6.2 Das AP214 als Grundlage für das Produktmodell für Getriebe

Das AP214 [I5] enthält alle nötigen Konstrukte, um die in Kapitel 6.1 beschriebenen Daten abzubilden. Darüber hinaus sind im AP214 umfassende Informationen definiert, die im Umfang des hier beschriebenen Datenmodells für Getriebe durch dessen Beschränkung auf den Konstruktionsprozess nicht explizit benötigt werden. Zur Fokussierung der für Getriebe notwendigen Daten wird das Produktmodell für Getriebe als eigenständiges Modell definiert. Durch eine Abbildungsvorschrift wird das getriebespezifische Produktreferenzmodell unter Verwendung der AP214-ARM-Konstrukte (Kapitel 3.6.1.3) beschrieben. Dabei werden bestehende Konstrukte des AP214 adaptiert und gleichzeitig geprüft, welche Vorgehensweise zur Erstellung eines STEP-konformen Produktreferenzmodells (Kapitel 4.4.3) angewendet werden kann. Es stellte sich heraus, dass eine Projektion des Produktmodells für Getriebe auf das AP214 möglich ist, was eine hundertprozentige Kompatibilität zum AP214 sicherstellt, ohne dieses zu verändern.

Die Suche nach Lösungsmöglichkeiten zur Projektion des Produktreferenzmodells für Getriebe auf das AP214 hat zu zwei unterschiedlichen Ansätzen für die Abbildung geführt:

- Verwendung der **Kinematikstruktur** des AP214
- Verwendung der **Produktstruktur** des AP214

6.2.1 Kinematikstruktur des AP214 als Kern des Produktreferenzmodells

Die Konstrukte zur Abbildung kinematischer Modelle innerhalb des AP214 erlauben die Modellierung einer einfachen Struktur, die neben den Bauteilen oder Baugruppen und deren Zusammenhängen auch Informationen über deren kinematische Beziehungen enthält. Hier ist neben den Definitionen von drehbar verbundenen Achsen, festverbundenen Elementen oder Gelenkverbindungen auch die Definition einer Zahnradpaarung vorgesehen. Daher scheint der kinematische Mechanismus des AP214 auf den ersten Blick geeignet, um die Produktstruktur des Datenmodells für Getriebe darauf abzubilden. Elemente des kinematischen Mechanismus sind einfache Körper, die über kinematische Paarbeziehungen zu einer Struktur zusammengefasst werden. Die Paarbeziehungen unterscheiden sich durch die zulässigen Freiheitsgrade, die zwischen den verbundenen Elementen bestehen. Durch die Benennung der Beziehungen und ihrer Elemente mit getriebespezifischen Bezeichnungen (Welle, Zahnrad, Lager, ...) lässt sich damit bereits eine Getriebestruktur definieren. Da das Produktmodell für Getriebe nicht den gesamten Produktlebenszyklus abbilden soll, sondern sich auf die Bereiche der Konstruktion und Berechnung beschränkt, ist diese Modellierungsart als zentrale Produktstruktur ausreichend. Voraussetzung ist allerdings, dass entsprechende Produktdaten wie Dokumente, Geometrieinformationen und Materialeigenschaften mit der Struktur verknüpft werden können. Diese Anbindungen sind jedoch in einer derart direkten Form im AP214 nicht vorgesehen.

Um die entsprechenden direkten Anbindungsmöglichkeiten dennoch zu erzeugen, müssten sowohl auf der ARM- als auch auf der AIM-Ebene Änderungen durchgeführt werden. Die Änderungen auf der ARM-Ebene wären vertretbar, da die ARM-Ebene nicht als Grundlage für die Implementierung dient. Auf der ARM-Ebene kann nach Bild 6-1 die benötigte Bezie-

hung durch die Erweiterung vorhandener Zuordnungsklassen erfolgen, die ähnliche Verknüpfungen zwischen den Objekten herstellen.

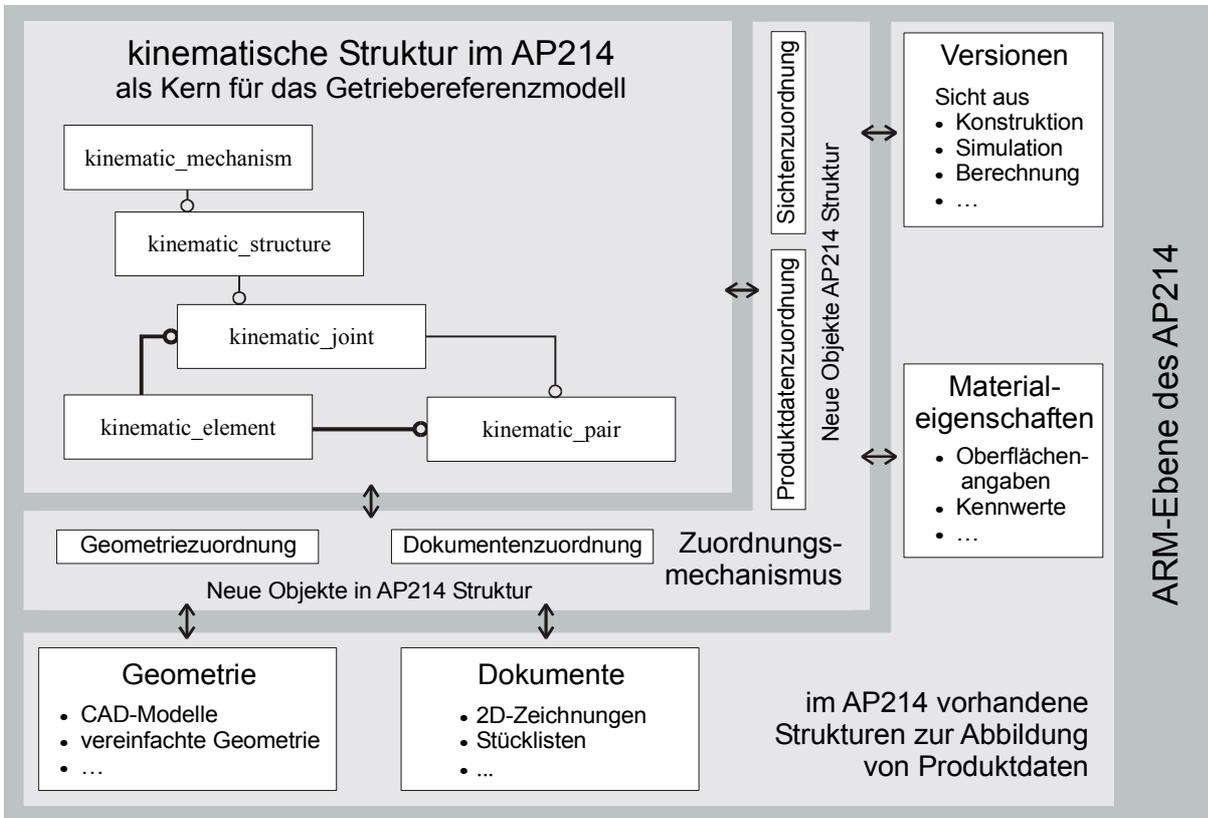


Bild 6-1: Erweiterung der ARM-Ebene für die Zuordnung der Produkteigenschaften zum kinematischen Modell

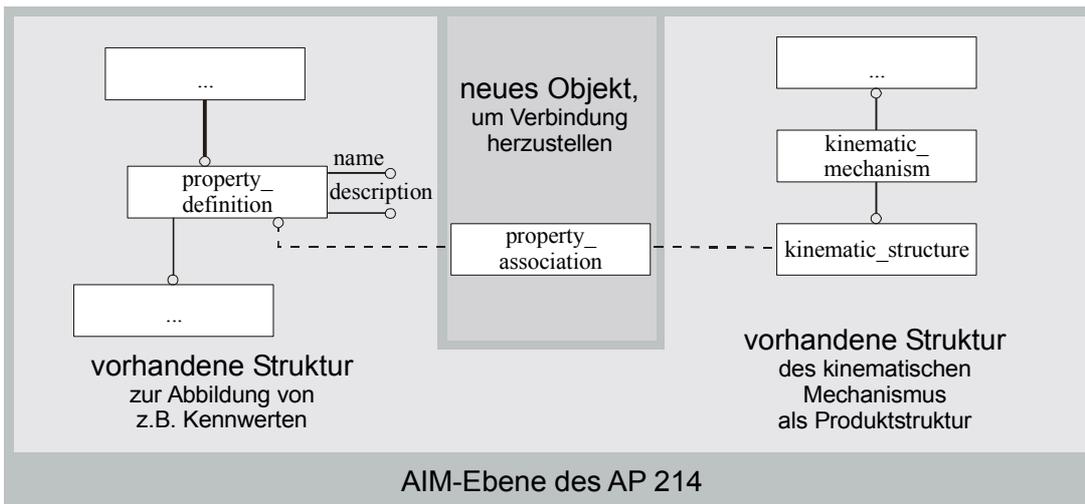


Bild 6-2: Benötigte Erweiterung der AIM-Ebene durch zusätzliche Elemente und Beziehungen

Die Veränderung auf der AIM-Ebene ist jedoch nicht problemlos durchführbar. Zur Herstellung der benötigten Verknüpfungen müssen entsprechende Zwischenelemente definiert werden, die über ihre Attribute die normalerweise unabhängigen Objekte in Beziehung setzen (Bild 6-2). Die neu erzeugten Elemente auf der AIM-Ebene führen dazu, dass AP214-kompatible Rechnerwerkzeuge die neuen, unbekanntenen Elemente und ihre Verknüpfungen nicht verarbeiten können und unter Umständen andere Daten nicht mehr interpretierbar sind. Eine Kompatibilität zum AP214 ist dann nicht gegeben, was gegen die Kinematikstruktur des AP214 als Kern für das Produktreferenzmodell für Getriebe spricht.

6.2.2 Produktstruktur des AP214 als Kern des Produktreferenzmodells

Die Verwendung der vorhandenen Konstrukte aus Kapitel 3.6.2 zur Beschreibung der Produktstruktur und der Bauteile im AP214 ermöglicht die Anbindung aller benötigten Daten sowie die Definition von Versionen und Sichten. Daher können diese als Kern des Produktmodells für Getriebe angewendet werden. Funktionelle Beziehungen zwischen den Bauteilen und Baugruppen müssen jedoch explizit definiert werden. Dies kann durch vorhandene Elemente des AP214 ohne Änderungen auf AIM-Ebene erfolgen.

6.2.3 Bewertung und Auswahl

Die Veränderung der AIM Ebene bei Verwendung der Kinematikstruktur des AP214 bedeutet zwangsläufig eine Änderung der entsprechenden Basismodelle des AP214. Damit geht der eigentliche Vorteil einer systemneutralen Schnittstelle durch die Abbildung des Produktreferenzmodells für Getriebe auf das AP214 verloren.

Dem gegenüber entstehen bei Verwendung der Produktstruktur des AP214 keine Einschränkungen oder Notwendigkeiten, die implementierungsnahen Basismodelle zu verändern. Die Vorteile bei der Vorgehensweise aus Kapitel 4.4.3 bleiben damit voll erhalten.

Somit wird das Produktreferenzmodell auf Basis einer Abbildung auf die AP214-Konstrukte zur Modellierung der Produktstruktur realisiert.

6.3 Realisierung des Produktmodells für Getriebe

Das Produktreferenzmodell für Getriebe ist ein eigenständiges Produktreferenzmodell und ist aus Kompatibilitätsgründen vollständig auf das AP214 projizierbar. Nach DYLA et al. [D6] soll die Dokumentation als Quasi-Normdokument in Anlehnung an STEP-Normdokumente erfolgen und eine Erweiterung des AP214 zur Anwendung in der Getriebeentwicklung darstellen. Kapitel 7.4 zeigt die Integration des Produktreferenzmodells in das rechnerbasierte Programmsystem. In Kapitel 8 wird dessen Anwendung an einem Beispiel gezeigt.

6.3.1 Struktur des Produktreferenzmodells

Die Struktur des Produktreferenzmodells für Getriebe orientiert sich so weit wie möglich an der objektorientierten Struktur des AP214 ARM. Dadurch wird die spätere Abbildung des Produktreferenzmodells auf Konstrukte des AP214 erleichtert. Durch die Projektion auf das AP214 stehen alle weiteren Datenstrukturen des AP214 weiterhin zur Verfügung. Eine Kompatibilität mit beliebiger STEP AP214-konformer Software ist somit gewährleistet.

6.3.1.1 Produktstruktur, Versionen, Sichten

Die Produktstruktur bildet den Kern des Produktmodells und ist vereinfacht in Bild 6-3 dargestellt. Sie beinhaltet die Definition der Bauteile mit zugehörigen Versionen und entsprechenden Sichten. Die Verknüpfung zwischen verschiedenen Bauteilen erfolgt in der Ebene der Sichten, da diese alle Produktdaten für den entsprechenden Anwendungsfall bündeln.

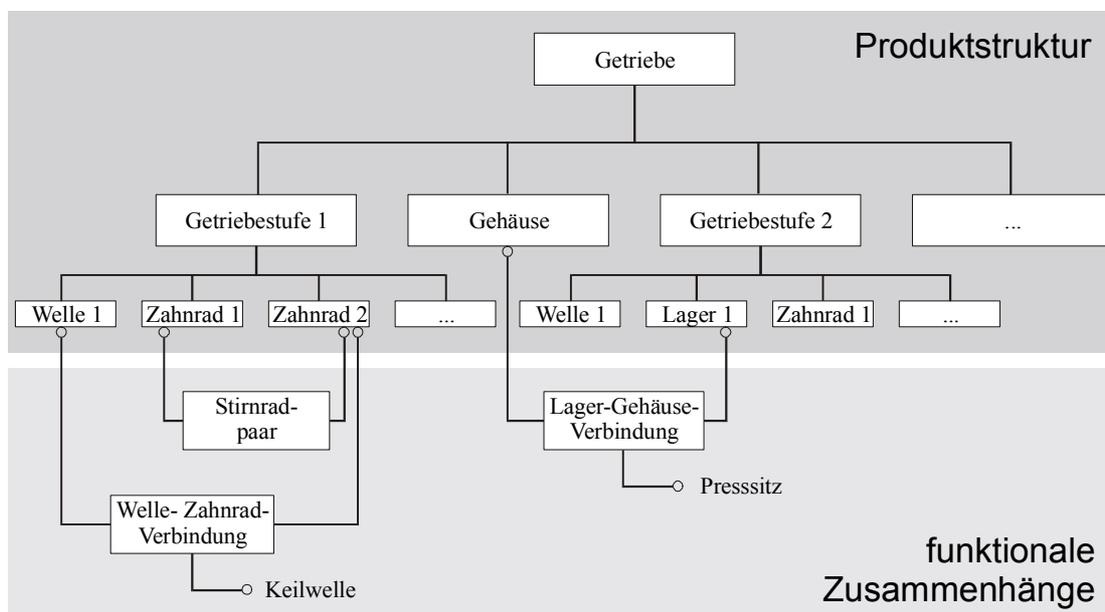


Bild 6-3: Produktstruktur in der Sichtenebene und übergreifende Funktionsbeziehungen zwischen den Bauteilen

Dadurch können Produktkonfigurationen erstellt werden, die nur für bestimmte Zwecke nötige Daten enthalten, beispielsweise eine Produktkonfiguration, die keine speziellen CAD-Daten enthält, sondern nur eine Ersatzgeometrie für Berechnungszwecke. Für die Definition einer Sicht werden zwei Typen zur Verfügung gestellt: Eine Sicht für die Repräsentation des Bauteils als Einzelteil (*Single_part_definition*) und eine Sicht für die Repräsentation des Bauteils als Baugruppe (*Assembly_definition*).

Die Definition der einzelnen getriebetypischen Maschinenelemente erfolgt auf der Bauteilebene, da sich die Typen durch eine Versionierung und verschiedene Sichten nicht ändern. Eine Baugruppe entsteht durch die Definition eines Bauteils als Baugruppe und ihrer Verknüpfung mit anderen Bauteilen oder Baugruppen über ein Element, das den Typ der Einbaubeziehung angibt. Dadurch ist es möglich nach hierarchischen Baugruppen oder normalen Baugruppen zu unterscheiden. Um eine funktionale Beziehung zwischen zwei Bauteilen herzustellen werden ihre entsprechenden Sichten über ein weiteres Element miteinander verbunden und der Typ der Verbindung dadurch festgelegt. Die Abbildungsvorschrift stellt sicher, dass funktionale Beziehungen auch nur zwischen passenden Bauteilen hergestellt werden können. Beispielsweise kann eine Welle-Zahnrad Beziehung nur ein Bauteil vom Typ Welle und eines vom Typ Zahnrad miteinander verknüpfen. Bild 6-3 zeigt die zwei Arten der Beziehungsmodellierung zwischen Bauteilen (Welle-Zahnrad-Verbindung) und zwischen Bauteil und Baugruppe (Lager-Gehäuse-Verbindung).

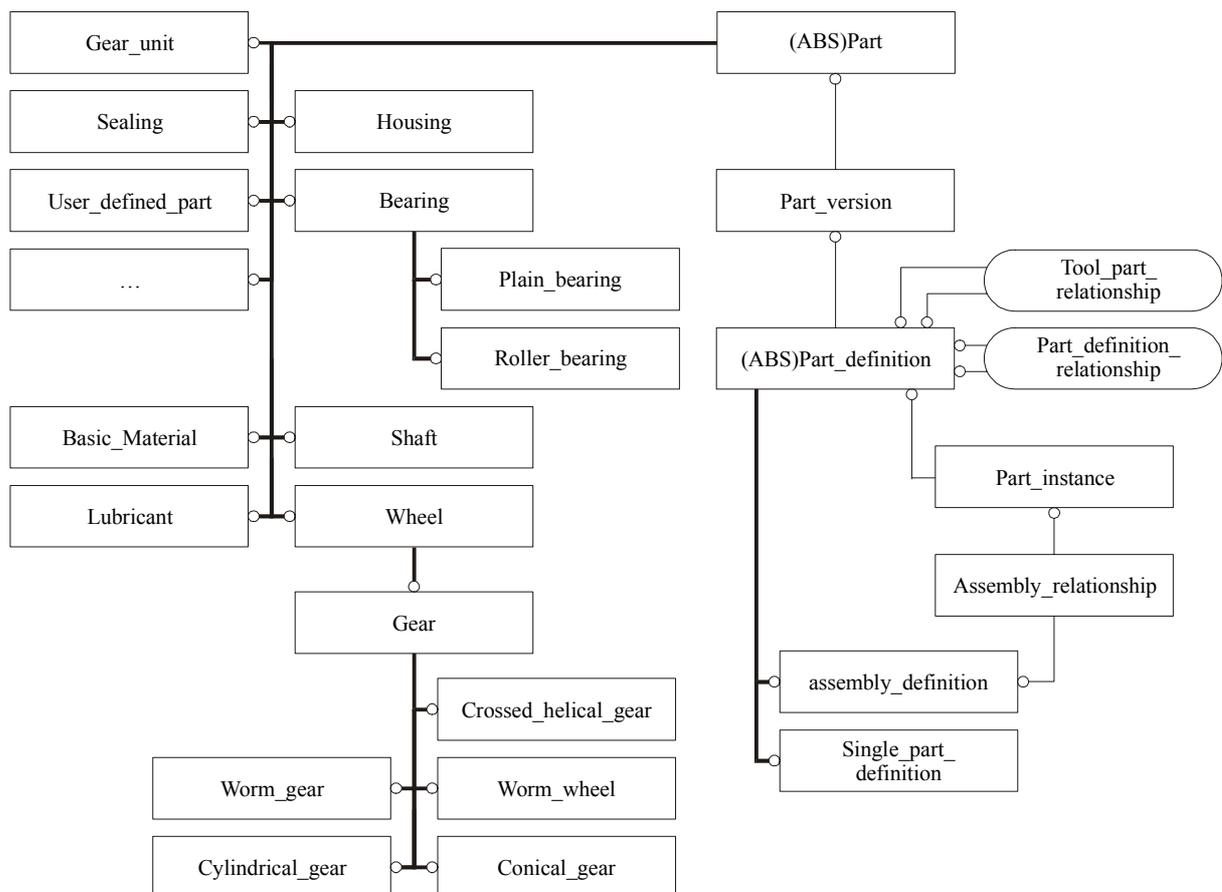


Bild 6-4: Elemente und ihre Beziehungen für die Definition der Produktstruktur

Für Bauteile und funktionale Beziehungen sind gleichartige Typen, die sich in ihrer Spezialisierung unterscheiden, unter abstrakten Oberklassen zusammengefasst. Von der abstrakten Oberklasse *Part* leiten sich alle Maschinenelemente ab, die ein typisches Bauteil eines Getriebes repräsentieren (Bild 6-4). Von der abstrakten Oberklasse *Part_definition_relationship* leiten sich alle Objekte ab, die eine Beziehung zwischen zwei Bauteilen herstellen. Die Beziehung zwischen den Objekten wird nur über die abstrakte Oberklasse hergestellt. Ein Sonderfall ist die Klasse *Tool_part_relationship*, die eine funktionale Beziehung zwischen einem Zahnrad und dem Herstellwerkzeug definiert. Sie wird nicht von der Oberklasse *Part_definition_relationship* abgeleitet, sondern stellt ein eigenständiges Objekt dar.

6.3.1.2 Geometrie und Topologie

Die Definition der Geometrie und Topologie eines Bauteils besteht aus zwei Bereichen. Für das Bauteil wird eine Gesamtgestalt erzeugt, die wiederum einen oder mehrere Gestaltabschnitte enthalten kann. Die Gestalt ist unabhängig von einer speziellen Geometriedarstellung. Für die einzelnen Gestaltabschnitte wird dann eine entsprechende Geometrie in Form des gewünschten Modells beschrieben (Volumenmodell, Flächenmodell...). Eine Welle lässt sich so beispielsweise der Gestalt nach in verschiedene Wellenabschnitte unterteilen. Die Geometriemodellierung für die Welle erfolgt dann entweder als Gesamtgeometrie oder als Summe der Einzelgeometrien der Abschnitte.

Im Rahmen der Definition des Produktreferenzmodells für Getriebe werden nur die Gestaltdefinition und die nötigen Anknüpfungspunkte für die Geometriemodellierung (*Geometric_model*) explizit beschrieben (Bild 6-5). Die eigentliche Geometriemodellierung ist aufgrund der Anwendung im CAD-Bereich im AP214 Stand der Technik. Hier kann für die Abbildung des Produktreferenzmodells das AP214 in vollem Leistungsumfang in Anspruch genommen werden.

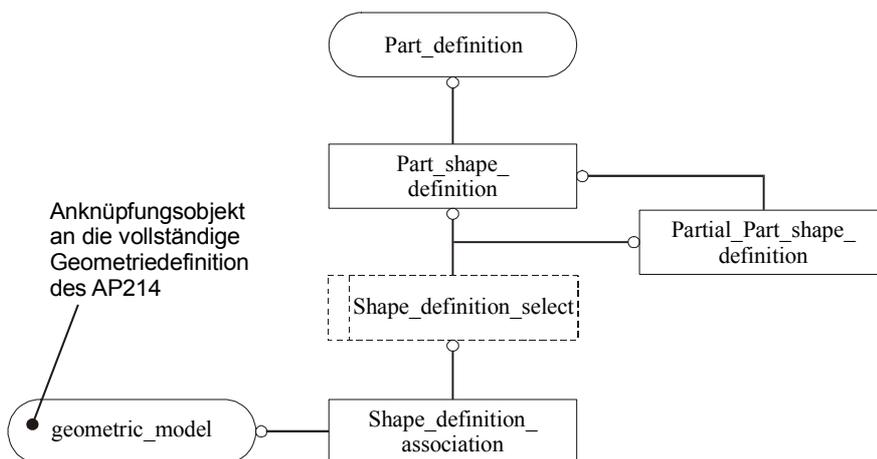


Bild 6-5: Elemente und ihre Beziehungen für die Geometrie- und Gestaltdefinition

6.3.1.3 Kennwerte, Betriebs- und Anwendungsdaten

Die Struktur für die in Kapitel 6.1.4 beschriebene Definition von Kennwerten sowie Betriebs- und Anwendungsdaten orientiert sich weitestgehend an der vorhandenen Struktur im AP214 (Bild 6-6). Die Daten werden durch drei zusammengehörige Elemente definiert:

- Ein *Property* ist eine abstrakte Oberklasse und beinhaltet Typen für die Klassifizierung nach Anwendungsdaten und Kennwerten, die wiederum in weitere Typen unterteilt sind.
- Ein *Property_value* ist ein Werteausdruck, der sich aus der Bezeichnung, dem Wert (numerischer Wert, Zeichenfolge, ...) und einer optionalen Einheit zusammensetzt.
- Das *Property_value_definition* Objekt dient zur Zusammenfassung der Klassifizierung und des Werteausdrucks.

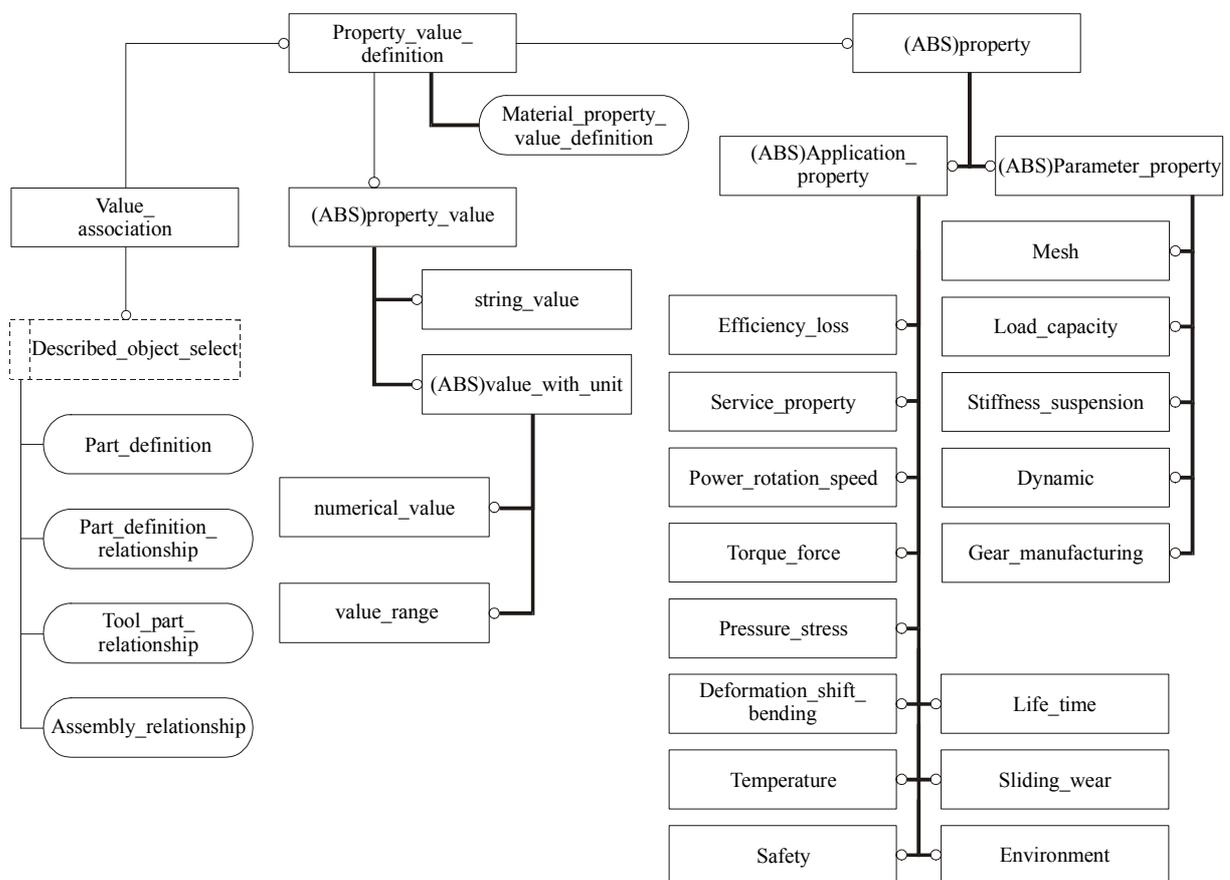


Bild 6-6: Elemente und ihre Beziehungen für die Definition von Properties

Durch die hohe Zahl der aus den Berechnungs- und Simulationsprogrammen stammenden möglichen Bezeichnungen eines *Property_value* werden diese Daten nicht durch entsprechende unterschiedliche Typen im Produktmodell beschrieben, wie es beispielsweise für die Bauteile ausgeführt worden ist. Erst in der Abbildungsvorschrift des Produktreferenzmodells auf das AP214 ARM werden, abhängig von der groben Klassifizierung über das *Property* Element, die möglichen Bezeichnungen des *Property_value* aus Kennwerttabellen übernom-

men. Die Zuordnung zu einem Bauteil oder zu einer Baugruppe, einer Bauteil- oder Baugruppenbeziehung, einer funktionalen Beziehung und einem Werkzeug, die durch den definierten Kennwert beschrieben werden, erfolgt über ein weiteres Element (*Value_association*), das Kennwert und Objekt in Beziehung zueinander setzt. Bild 6-7 zeigt vereinfacht die Vorgehensweise bei der Verwendung einer Kennwerttabelle zur Definition und Instanzenbildung von *Properties*.

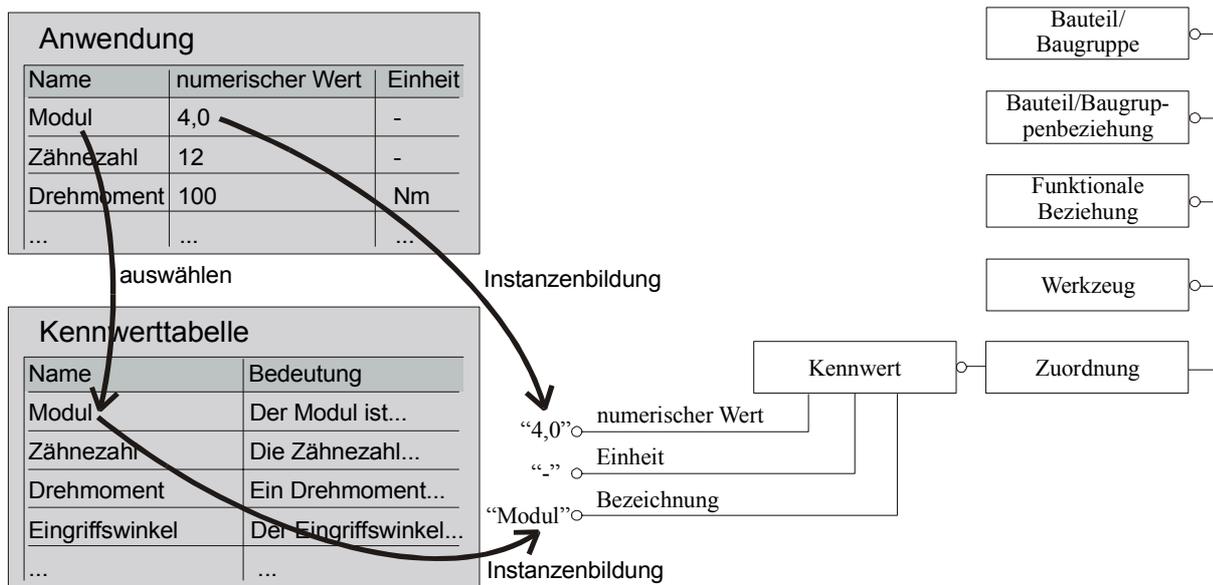


Bild 6-7: Definition, Zuordnung und Instanzenbildung von *Properties* unter Verwendung einer Kennwerttabelle

6.3.1.4 Materialangaben und Oberflächeneigenschaften

Die Besonderheit bei der Definition der Materialangaben besteht in zwei Möglichkeiten der Beschreibung. Materialkennwerte lassen sich analog der Struktur für Kennwerte, mit teilweise speziellen, materialbezogenen Unterklassen definieren und Objekten zuordnen. Zudem ist es möglich, über eine *Material-Definition* alle Materialkennwerte zusammenzufassen. Diese *Material-Definition* kann dann einem Objekt, das ein konkretes Material (*Material*, *Lubricant*) repräsentiert, zugeordnet werden. Durch Beziehungsdefinitionen zwischen dem Material und den entsprechenden Bauteilen werden alle Kennwerte des Materials für das Bauteil übernommen. Bild 6-8 zeigt die definierten Strukturen.

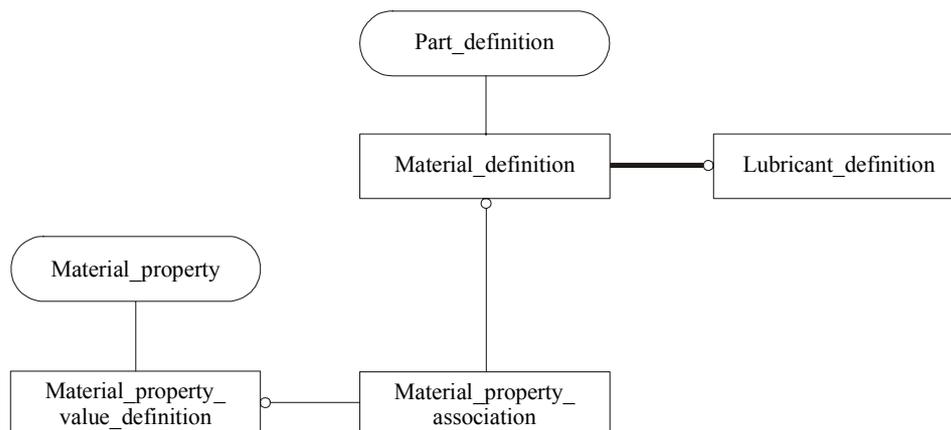


Bild 6-8: Elemente und Beziehungen für die Definition von Materialangaben

Für die Definition von Oberflächenangaben wurden die nötigen Strukturen aus dem AP214 übernommen (Bild 6-9). Oberflächenangaben werden immer einer Gestaltdarstellung zugeordnet (vergleiche *Part_shape_definition* oder *Partial_part_shape_definition* in Bild 6-5). Da ein Bauteil in mehrere Gestaltbereiche unterteilt werden kann, können für ein Bauteil verschiedene Bereiche mit unterschiedlichen Oberflächen definiert werden.

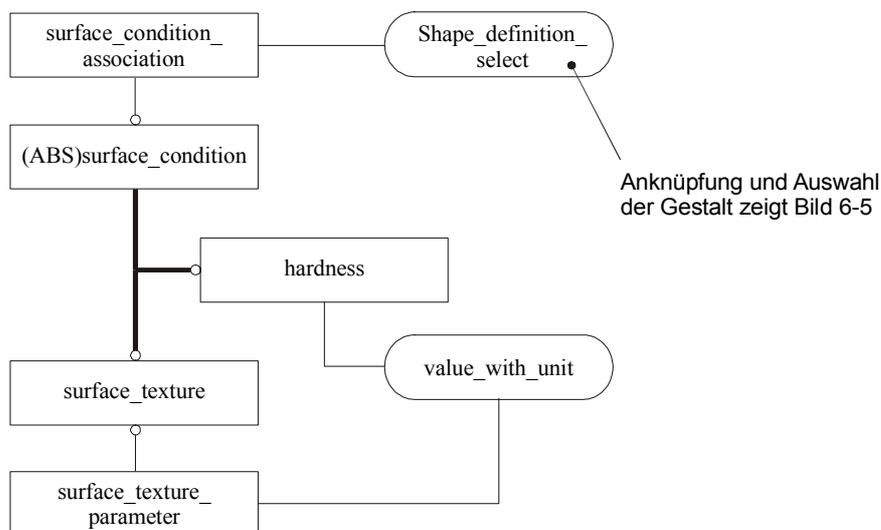


Bild 6-9: Elemente und Beziehungen für die Definition von Oberflächeneigenschaften

6.3.1.5 Verknüpfung von Dokumenten

Die in Bild 6-10 dargestellte Anbindung von Dokumenten an bestehende Strukturen des Produktmodells erfolgt analog zu der Vorgehensweise im AP214. Die Verknüpfung zwischen den Dokumenten und den korrespondierenden Objekten wird über die Klasse *Document_assignment* hergestellt. Die Anknüpfungspunkte für die Dokumentdefinition im AP214 sind die Klassen *Document*, *Document_version*, *Document_representation* und *Document_file*.

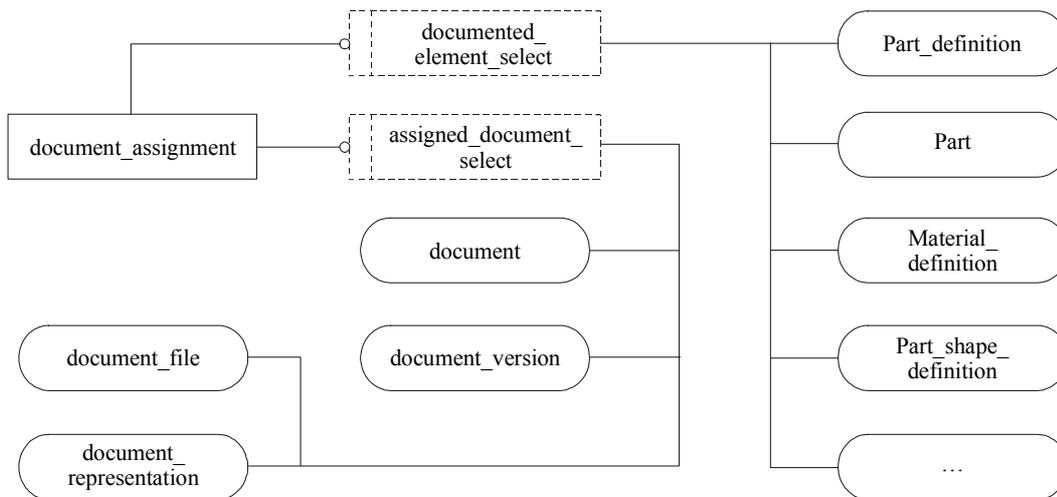


Bild 6-10: Elemente und Beziehungen für die Definition von Dokumenten

Dokumente können als einfaches einzelnes Dokument vorliegen (digitale Datei, Zeichnung, ...) oder es wird eine Repräsentation eines Dokuments erzeugt, die sich aus entsprechenden einzelnen Dokumenten zusammensetzt.

Da die Anbindung von Dokumenten keine getriebespezifische Ausprägung des Produktreferenzmodells ist, wird hier nur die Anbindungsmöglichkeit abgebildet. Die eigentlichen Darstellungsmöglichkeiten eines Dokuments werden im AP214 explizit definiert. Bild 6-10 zeigt die dafür vorgesehenen Strukturen.

6.3.2 Aus dem AP214 übernommene Anwendungsobjekte

Das Produktreferenzmodell für Getriebe enthält aus Kompatibilitätsgründen keine neuen Datenklassen, die in ihrer grundlegenden semantischen Bedeutung nicht bereits im AP214 vorhanden sind. Sofern die notwendige AP214-Struktur für die Datenbeschreibung nicht sinnvoll im Kontext der Anwendung für Getriebe vereinfacht dargestellt werden konnte, wurde die AP214-Struktur exakt übernommen. Dadurch gestaltet sich für diese Objekte des Produktreferenzmodells die Beschreibung unter Verwendung der AP214-ARM sehr einfach, da eine eins zu eins Abbildung vorgenommen wird. Das bedeutet, dass beispielsweise ein Objekt des Produktreferenzmodells vom Typ *Assembly_definition* auch im AP214 das Objekt vom Typ *Assembly_definition* ist. Beide Objekte verfügen über die gleichen Attribute und Beziehungen zu anderen Objekten. Der einzige Unterschied liegt teilweise lediglich in dem Weglassen von nicht relevanten Unterklassen, um das Produktreferenzmodell für Getriebe zu vereinfachen. Folgende Objekte wurden aus dem AP214 übernommen:

- *Application_context*
- *Assembly_definition*
- *Centre_of_mass* und *Moments_of_inertia*
- *Document*, *Document_version*, *Document_representation*, *Document_file* und *Document_assignment*
- *Material_property*
- *Material_property_association*
- *Property*
- *Property_value* mit seinen Unterklassen
- *Surface_condition* mit seinen Unterklassen

Teilweise gibt es zudem Strukturen, die nur durch die Benennung der enthaltenen Objekte abweichen, um ihre Zuordnung zu neu definierten Objekten hervorzuheben. Bei den entsprechenden Elementen wird im Prinzip auch eine eins-zu-eins-Abbildung auf das AP214-ARM durchgeführt, der einzige Unterschied liegt in der unterschiedlichen Benennung. Die Beziehungen und die Bedeutung der Attribute sind in beiden Modellen identisch. Dies trifft auf folgende Objekte des Produktreferenzmodells für Getriebe zu. Das jeweils äquivalente AP214 Objekt ist in Klammern angegeben.

- *Geometry_property* (*General_shape_dependent_property*)
- *Lubricant_definition* (*Material*)
- *Material_definition* (*Material*)
- *Part_definition* (*Design_discipline_item_definition*)
- *Part_instance* (*Item_instance*)
- *Part_version* (*Item_version*)
- *Part_shape_definition* (*Item_shape*)
- *Partial_part_shape_definition* (*Shape_element*)
- *Property_value_definition* (*Property_value_representation*)
- *Shape_definition_association* (*Shape_description_association*)
- *Shape_property* (*Shape_dependent_property*)
- *Single_part_definition* (*Design_discipline_item_definition*)

6.3.3 Im Produktreferenzmodell für Getriebe neu definierte Anwendungsobjekte

Neu definierte Anwendungsobjekte entstehen durch eine Zusammenführung mehrerer vorhandener AP214-Objekte in ein einzelnes Objekt oder durch die Festlegung bestimmter Attributausprägungen der ARM-Objekte bei der Abbildung auf das AP214. Grund für diese Vereinfachung und Zusammenfassung des AP214 auf der Ebene des Produktreferenzmodells ist die Erstellung eines möglichst einfachen Produktreferenzmodells für Getriebe mit einer

die Erstellung eines möglichst einfachen Produktreferenzmodells für Getriebe mit einer größtmöglichen Übersichtlichkeit. In der Implementierungs- beziehungsweise Datenaustauschebene tritt diese Vereinfachung aufgrund des zweifachen *Mappings* nicht mehr auf, was eine hundertprozentige Kompatibilität zum AP214 AIM sicherstellt.

Bei der Zusammenführung werden für die Abbildung auf das AP214 auch mehrere Objekte des AP214 benötigt, um die Funktionalität des neu definierten Anwendungsobjektes herzustellen. Dieser Fall tritt auf, wenn beispielsweise Oberklassen für bestimmte Attribute oder für die Verknüpfung zu einem anderen Element gebraucht werden, ihre weiteren Funktionalitäten in dem spezifischen Fall jedoch nicht interessieren.

Bei der Festlegung bestimmter Attributausprägungen werden diese für die Abbildung auf das AP214 in die benötigten Objekte mit übernommen. Dadurch bekommen die Objekte des AP214 die entsprechende Bedeutung, die das Objekt des Produktreferenzmodells ausgezeichnet hat.

Folgende Objekte wurden für das Produktreferenzmodell neu definiert. In Klammern ist die Methode angegeben, mit der die neu definierten Objekte aus bestehenden Objekten des AP214 hervorgegangen sind.

- *Application_property* mit allen Unterklassen (vordefinierte Attributausprägungen)
- *Assembly_relationship* (Zusammenfassung mehrerer AP214 Objekte)
- *Parameter_property* mit allen Unterklassen (vordefinierte Attributausprägungen)
- *Part* mit allen Unterklassen (vordefinierte Attributausprägungen)
- *Part_definition_relationship* mit allen Unterklassen (vordefinierte Attributausprägungen)
- Alle Unterklassen des *Property*-Objekts (vordefinierte Attributausprägungen)
- *Tool_part_relationship* (Zusammenfassung mehrerer AP214 Objekte)
- *Value_association* (Zusammenfassung mehrerer AP214 Objekte)

6.3.4 Mechanismus der Abbildung des Produktreferenzmodells auf das AP214

Bei der Definition eines realen Getriebes mit Hilfe des AP214 benötigt der Anwender viel Kenntnis der dort definierten Strukturen und ihrer Funktionen. Durch den hohen Abstraktionsgrad besteht die Gefahr der unterschiedlichen Interpretation der AP214-Funktionen auf ARM-Ebene in Bezug auf die Modellierung der spezifischen Getriebedaten durch verschiedene Anwender. Dies führt unter Umständen zu unterschiedlichen Darstellungsformen des Getriebes im AP214, die parallel beständig und AP214-konform sein können. Bild 6-11 zeigt beispielhaft den Vorgang vom realen Getriebe zur implementierungsnahen Darstellung auf der AIM-Ebene des AP214, der auf zwei unterschiedlichen Wegen zum gleichen Ziel eines AP214-kompatiblen Produktreferenzmodells führt. Mit Hilfe des Produktreferenzmodells für Getriebe wird dieses Problem umgangen. Durch die einfache Darstellungsform aller getriebe-spezifischen Elemente und Daten im Produktreferenzmodell wird eine eindeutige Abbildung eines realen Getriebes auf das Produktreferenzmodell ermöglicht. Der Anwender muss auf dieser Ebene kein Wissen über das AP214 besitzen, da das Produktreferenzmodell für Getrie-

be ein eigenständiges Datenmodell ist. Durch den festgelegten Abbildungsmechanismus des Produktreferenzmodells für Getriebe auf das AP214 können auf AP214 ARM-Ebene keine unterschiedlichen Modellierungen entstehen.

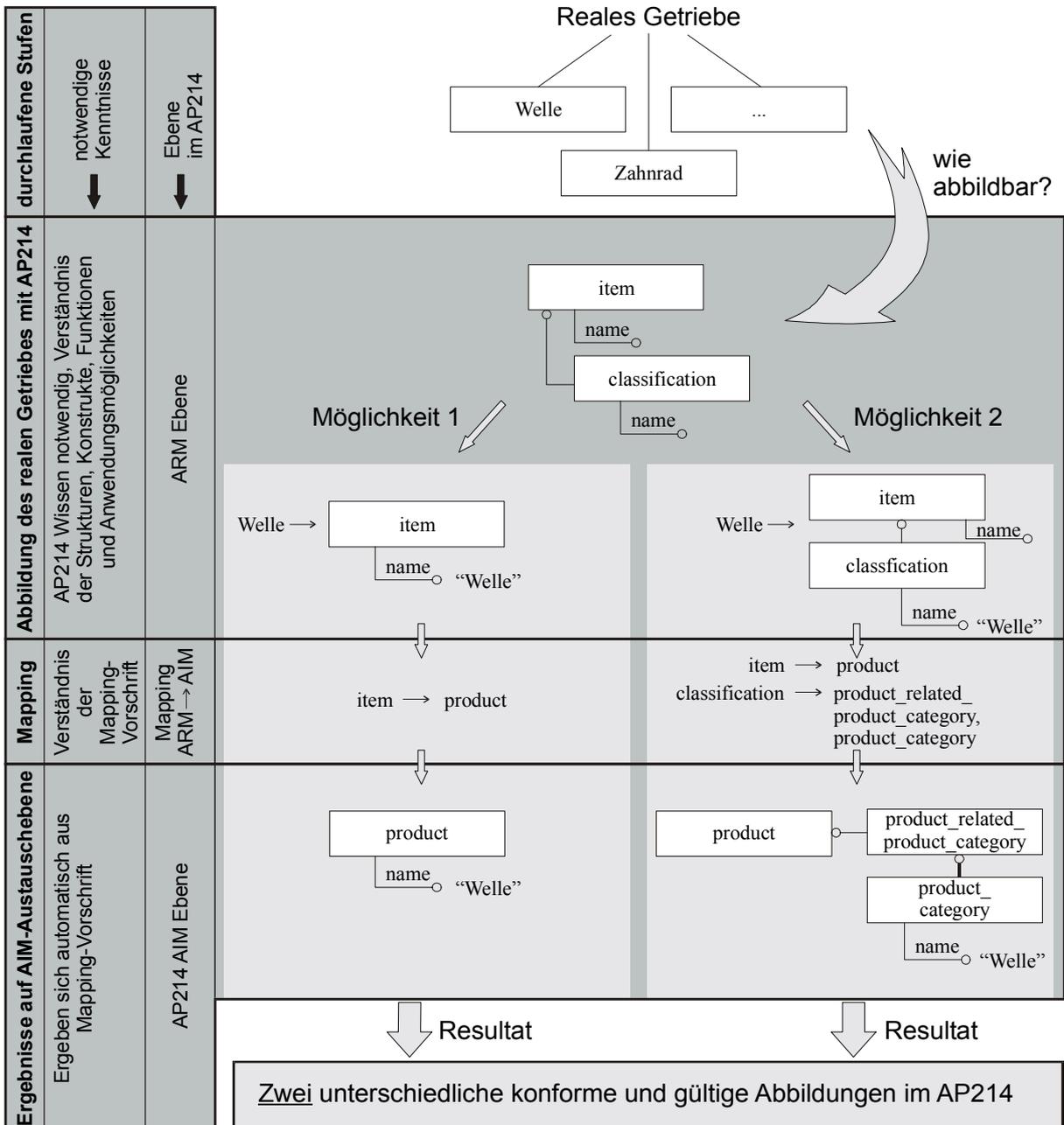


Bild 6-11: Unterschiedliche ARM-Interpretation einer Getriebemodellierung im AP214

Bild 6-12 zeigt beispielhaft den Vorgang vom realen Getriebe zur implementierungsnahen Darstellung auf der AIM-Ebene des AP214 bei Verwendung des Produktreferenzmodells für Getriebe. Im Endeffekt entsteht eine eindeutige, AP214-konforme Abbildung des Getriebes.

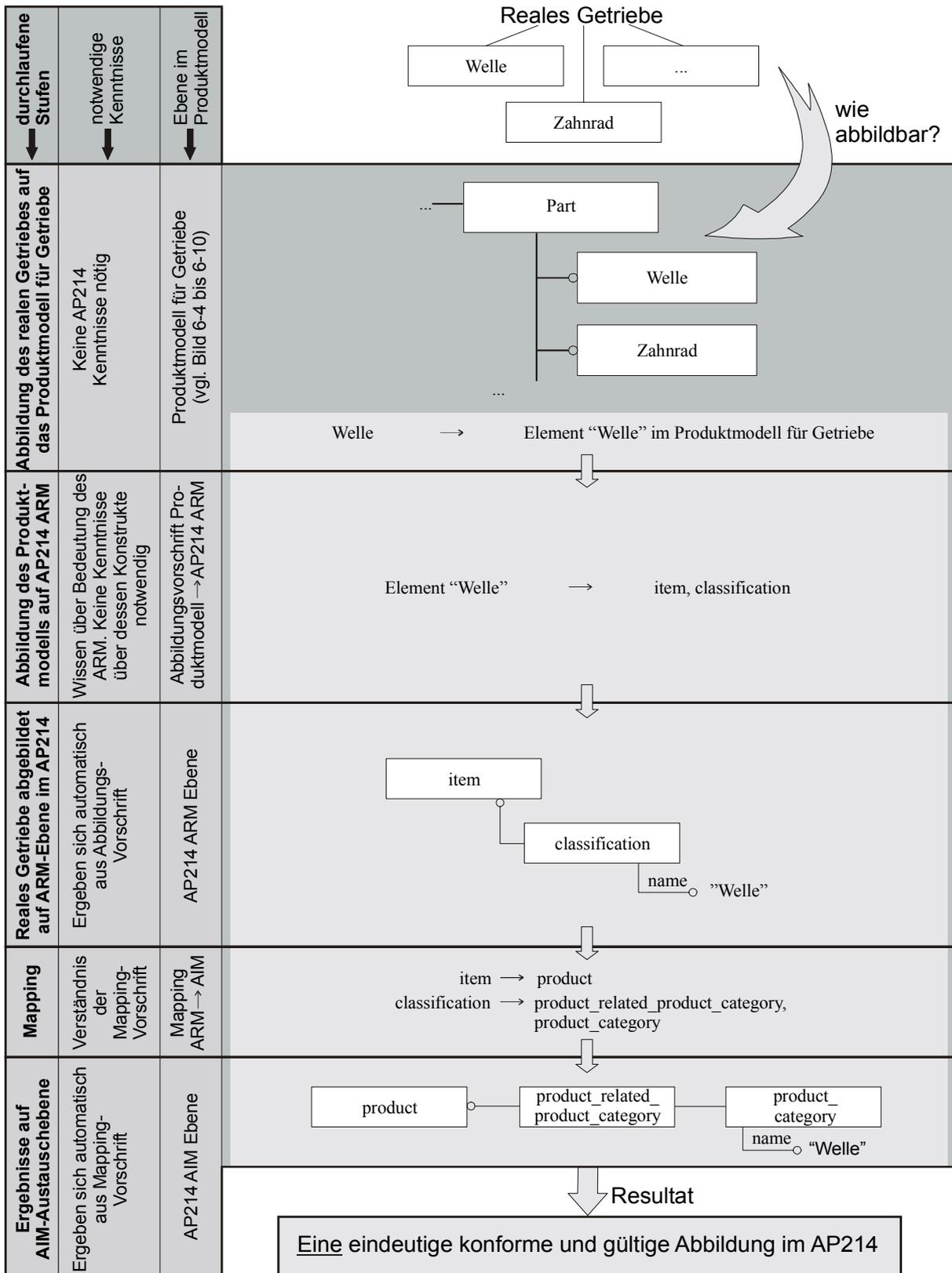


Bild 6-12: Verwendung des Produktreferenzmodells für eine eindeutige Getriebemodellierung mit dem Ergebnis eines AP214-konformen Produktmodells

7 Übertragung des rechnerbasierten Programmsystems zur Anwendung in der Getriebeentwicklung

Die Anwendung des in Kapitel 5 allgemein beschriebenen, rechnerbasierten Programmsystems zur Produktentwicklung wird exemplarisch am Produktentwicklungszyklus in der Getriebeentwicklung dargestellt. Grundlage für die Produktmodellierung der Datenstrukturen in der Getriebeentwicklung stellt das für Getriebe entwickelte STEP-Produktreferenzmodell aus Kapitel 6 dar. Um die Funktionsweise des rechnerbasierten Programmsystems während der Entwicklung eines Getriebes nachvollziehen zu können, müssen die dabei verwendeten Programme, ihre Wirkungsweisen und ihre Zusammenhänge untereinander im Spannungsfeld des gestalterischen und berechnenden/simulierenden Entwicklungsprozesses verstanden werden.

Der Entwicklungsprozess mit einem rechnerbasierten Programmsystem nach Kapitel 5.1 wird für die Getriebeentwicklung vertieft und darauf basierend ein Prozessmodell definiert, welches die Grundlage für die Anpassung des rechnerbasierten Programmsystems an die Getriebeentwicklung darstellt.

7.1 Prozessmodell Getriebeentwicklung

Für Getriebe stellt sich der Entwicklungsprozess wie in Bild 7-1 dar.

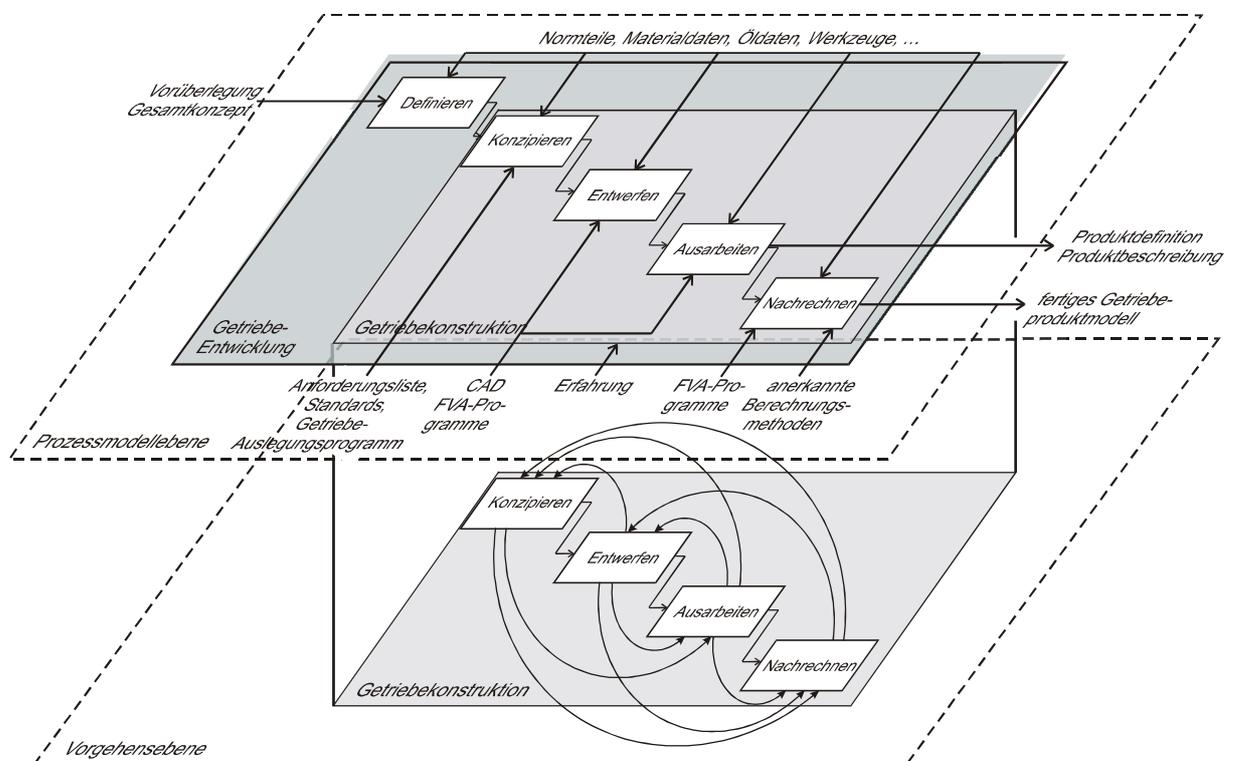


Bild 7-1: Prozessmodell und Vorgehensebene des Getriebeentwicklungsprozesses

Zu den wichtigsten Bestandteilen des Produktentwicklungsprozesses von Getrieben gehören das Vorausberechnen zur Festlegung der groben Getriebedaten (**Definieren und Konzipieren**), die gestalterischen Tätigkeiten zur Definition der Produktgeometrie (**Entwerfen und Ausarbeiten**) und die Nachweisrechnung zur Bestätigung von Produkteigenschaften aus zuvor festgelegten Merkmalen (**Nachrechnen**). Das Vorgehen bei der Getriebeentwicklung besteht in der Vorgehensebene aus dem zyklischen Wiederholen einzelner Entwicklungsphasen und dient der **Optimierung** von Merkmalen unter systematischer Rückschlussziehung. Diesen Zusammenhang zeigt die parallel zum Entwicklungsprozess laufende Vorgehensebene in Bild 7-1.

Das grundsätzliche Vorgehen bei der Getriebeentwicklung ist in vielen Firmen der Getriebeindustrie ähnlich. Die hierfür eingesetzten Rechnerwerkzeuge unterscheiden sich jedoch. Aus diesem Grund wurde besonderes Augenmerk darauf gelegt, beliebige Programme in den Arbeitsplatz integrieren zu können und das Prozessmodell allgemeingültig zu gestalten.

7.2 Verwendete Programme zur Gestaltung und Berechnung von Getrieben

In das rechnerbasierte Programmsystem sind auf Berechnungsseite bestehende und in der Industrie eingeführte Programme aus der Antriebstechnik, auf Geometrieseite unveränderte und nicht angepasste CAD-Systeme eingebunden. Die Berechnungs- und Simulationsprogramme werden über ihre nativen Ein- und Ausgabedateien angesprochen, die CAD-Systeme ausschließlich über deren STEP-Datenschnittstellen.

7.2.1 Auslegungsprogramme

Die frühen Phasen der Getriebeentwicklung sind geprägt von einfachen Überschlagsrechnungen, die unter Einbezug von Erfahrung zu einer groben Getriebeauslegung führen. Dieser bislang nicht unterstützte Bereich der rechnergestützten Produktentwicklung ist durch das im Rahmen der vorliegenden Arbeit entwickelte interaktive Getriebe-Auslegungsprogramm ergänzt worden. Unter „interaktiv“ wird nach Dateneingabe eine direkte Reaktion des Programms mit einer sofortigen Ergebnisausgabe verstanden. Die Berechnung basiert auf hinterlegten und erweiterbaren Formeln, die mit Erfahrungskennwerten bekannter Getriebe gepaart werden. Das Programm besteht aus einfachen Ein- und Ausgabefenstern und bietet dem Benutzer zudem Wertevorschläge, die sich von ausgeführten Getrieben herleiten lassen. Die Einsatzmöglichkeiten des Getriebe-Auslegungsprogramms werden ausführlich in Kapitel 8.3.2 beschrieben.

7.2.2 CAD-Systeme

Zur Festlegung der Geometrie kommen neben der Erzeugung einer Ersatzgeometrie in den Nachrechnungs- und Simulationsprogrammen aus Kapitel 7.2.3 ausschließlich 3D-CAD-Systeme zum Einsatz, die eine uneingeschränkte Detaillierung des Geometriemodells zulassen.

sen. Basisformat zur Datenübertragung des CAD-Geometriemodells ist die STEP-Norm. Die Anpassung des Geometriemodells an die Ersatzgeometrie aus Berechnung und Simulation und die Eingliederung in das STEP-Produktmodell übernehmen programminterne Konverter des rechnerbasierten Programmsystems.

Zur direkten Erzeugung von Fertigungszeichnungen oder standardisierten Produkt-Datenblättern, die nicht automatisch aus den 3D-CAD-Systemen abgeleitet werden können, kommen 2D-CAD-Systeme zum Einsatz, die über eine 2D-STEP-AP201- oder DXF-Schnittstelle (Kapitel 3.2.4) verfügen.

7.2.3 Nachrechnungs- und Simulationsprogramme

Die wichtigsten industriell angewendeten Berechnungs-/Simulationsprogramme der FVA mit ihren korrespondierenden Datenbanken sind in Bild 7-2 dargestellt. Die in Bild 7-2 mit einem Stern (*) versehenen Programme wurden nicht an der FZG entwickelt und sollen beispielhaft den Umfang der verfügbaren Programme für die Getriebeentwicklung ausdrücken. Eine Integration dieser Programme in das rechnerbasierte Programmsystem wäre leicht möglich.

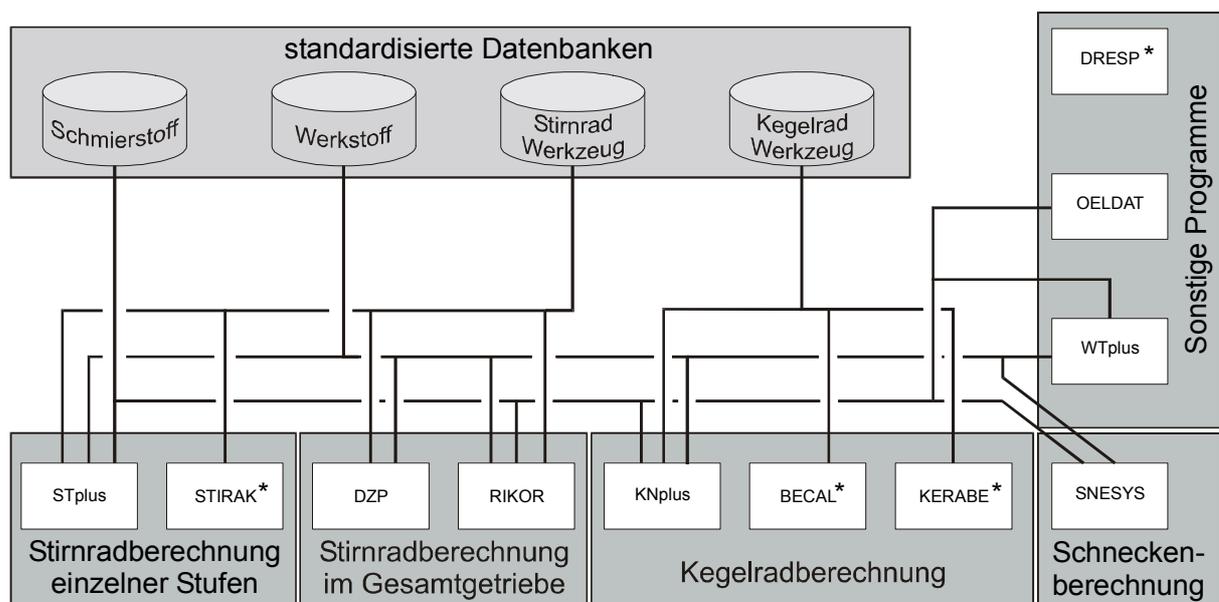


Bild 7-2: Mögliche Verknüpfungen bestehender FVA-Programme und Datenbanken

Zur beispielhaften Anwendung und Integration in das rechnerbasierte Programmsystem kommen neben den standardisierten Datenbanken die folgenden Programme aus Bild 7-2, die an der FZG entwickelt worden sind:

- **STplus** [H16][S20] wird zur Berechnung der Geometrie und Tragfähigkeit von Evolverstirnpaarungen und der Paarung Stirnrad-Werkzeug verwendet. Die Leistungsfähigkeit von STplus umfasst bei Stirnradstufen die Geometrieberechnung und den Tragfähigkeitsnachweis nach unterschiedlichen Methoden aus ISO, DIN, AGMA und diversen Anwendungsgremien. Neben der Berechnung einzelner Stirnräder und Zahnkupplungen ist auch die Berechnung des Werkzeugs für ein Stirnrad im Programm enthalten. Zum weite-

ren Leistungsumfang gehören die Berechnung und grafische Ausgabe der exakten Zahngeometrie eines Zahnrads.

- Das Kegelradnormprogramm **KNplus** [A14] dient zur Berechnung der wesentlichen geometrischen Größen und zur Nachrechnung der Tragfähigkeit von Kegelrädern mit Okto-idenzahnform. Für die Berechnung der Tragfähigkeit und des Wirkungsgrades stehen verschiedene Methoden nach ISO, DIN, AGMA und diversen Anwendungsgremien zur Verfügung.
- **RIKOR** [S3] steht für Ritzel-Korrekturprogramm und berechnet die Verformungen der Getriebeelemente bei Belastung. Bei Durchbiegung und Verdrillung von Wellen und Zahnrädern ändert sich die Lastverteilung auf der Zahnflanke. Dies wiederum bewirkt eine ungleichmäßige Verformung der Zähne. Mit RIKOR kann eine Flankenkorrektur berechnet werden, welche die obigen Erscheinungen ausgleicht. Trotz lastbedingter Verbiegungen und Verdrillungen herrschen dadurch wieder optimale Berührungszustände für einen bestimmten Lastfall.
- Das Programm **WTplus** [D3] dient der Erfassung des Wärmehaushalts eines Getriebes. An Elementen sind Stirnradgetriebe, Kegelradgetriebe, Hypoidgetriebe, Schraubradgetriebe und Schneckengetriebe inklusive der zugehörigen Wälz-, Gleitlager und Radialwellendichtringe enthalten. Zu jedem Getriebeelement lässt sich die Verlustleistung bestimmen. Eine über das gesamte Getriebe aufgestellte Wärmebilanz ergibt die mittlere Öltemperatur, die sich während eines stationären Betriebes einstellt. Nicht berücksichtigt werden hingegen die auftretenden Wärmeverteilungen und die Temperaturen der einzelnen Bauteile. Wie RIKOR beinhaltet auch WTplus die Geometriedaten eines kompletten Getriebes.
- Das Schneckenprogrammssystem **SNESYS** [S21] beinhaltet die Berechnung der Verzahnungsgeometrie, der Tragfähigkeit, des Tragbilds, der Berührverhältnisse und der Lastverteilung von Schneckengetrieben. Des weiteren wird die Selbstbremsung von Schneckengetrieben berechnet. SNESYS stützt sich auf gängige ISO- und DIN-Normen der Schneckengetriebeberechnung.
- Das Programm **DZP** [G6] bezeichnet das „Dynamische Zahnkräfte Programm“ der FVA und ermittelt die inneren dynamischen Zusatzkräfte von gerad- und schrägverzahnten Stirnradgetrieben. Mittels des Programms DZP kann das Schwingungsverhalten von ein- und mehrstufigen Stirnradgetrieben mit An- und Abtriebsstrang näher beschrieben werden. Insbesondere kann für jeweils eine Verzahnung des betrachteten Getriebes die parametrische Schwingungsanregung durch die periodische Verzahnungssteifigkeit sowie der Einfluss von Verzahnungsfehlern und Verzahnungskorrekturen näher untersucht werden.

Allen diesen Berechnungsprogrammen gemeinsam ist deren programmtechnischer Aufbau gemäß Bild 7-3 und eine ergonomisch ähnlich gestaltete Benutzungsoberfläche, die eine einfache Bedienung der Programme zulässt. Zudem greifen alle Programme auf die standardisierten Werkstoff- und Schmierstoffdatenbanken [S22] zu, die somit nur einmal im Unternehmen vorhanden sein müssen. Eine Integration anderer Programme zur Simulation und Berechnung von Getrieben oder einzelnen Maschinenelementen ist jederzeit möglich (Kapitel 7.4).

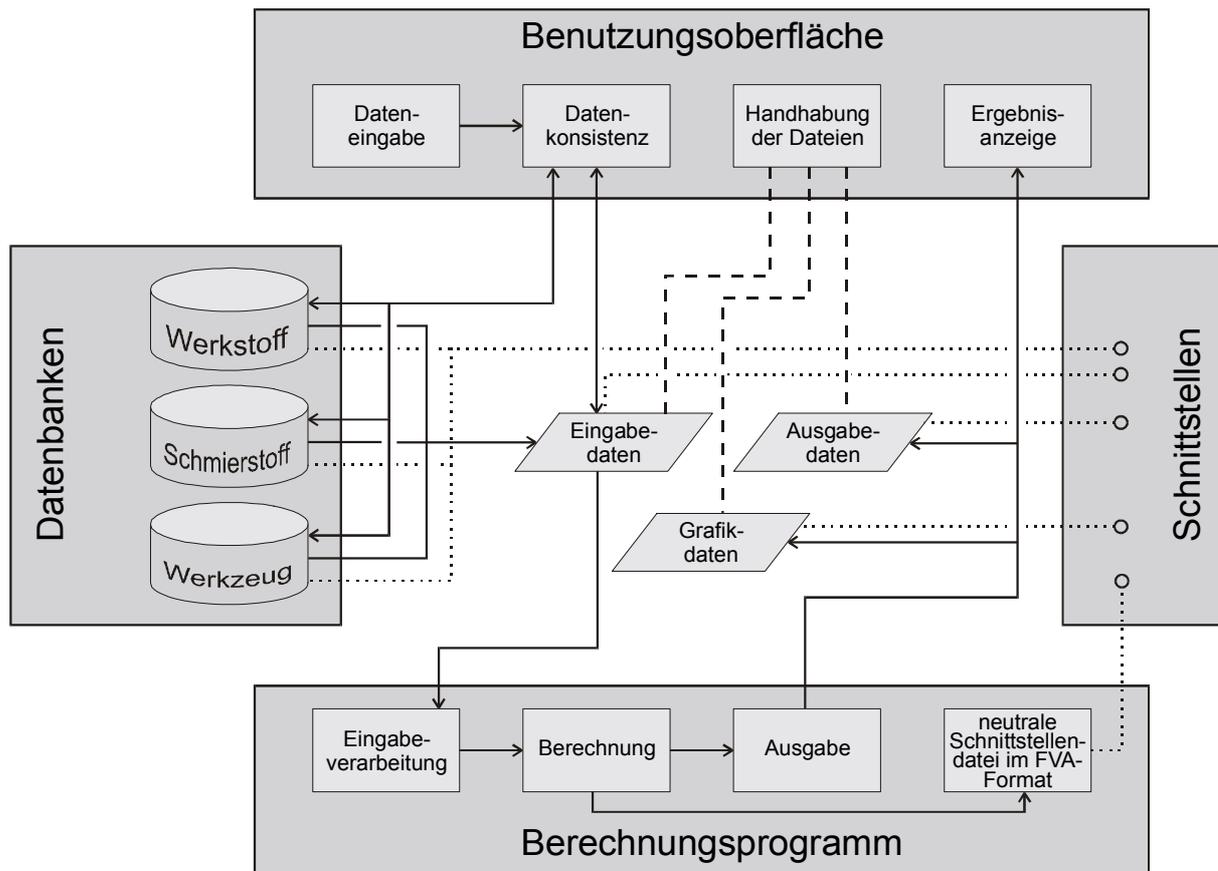


Bild 7-3: Programmtechnischer Aufbau von FVA-Programmen am Beispiel STplus (in Anlehnung an [H16])

7.3 Anpassung der Benutzeroberfläche

Die Benutzeroberfläche stellt die Schnittstelle zwischen Mensch und rechnerbasiertem Programmsystem dar, ist einheitlich aufgebaut und orientiert sich an gängigen Windows-Programmen, deren Leistungsfähigkeit sie in den Anpassungsmöglichkeiten durch den Anwender übernimmt. So können die Größe und Anordnung der Bitmaps bestimmt und an die integrierten Programme angepasst werden. Damit eine datentechnische Kompatibilität gewährleistet werden kann, müssen die Änderungen in einem Konfigurationsmenü vorgenommen und in einer Konfigurationsdatei gespeichert werden. Die in Kapitel 5.3.1.4 vorgestellten Visualisierungsfunktionen für Produkt- und Prozessdaten können nicht geändert werden, da sie für alle Produkte gleich gelten. Es können lediglich spezifische Bezeichnungen an die Anwendung angepasst werden.

Die gesamte Benutzeroberfläche ist auf verschiedene Betriebssysteme übertragbar und somit in allen Rechnerarchitekturen verwendbar.

7.4 Integration externer Programme und des Produktreferenzmodells für Getriebe in das rechnerbasierte Programmsystem

In Kapitel 6 wurde das Produktreferenzmodell für Getriebe vorgestellt. Es enthält alle Strukturen und Maschinenelemente, die während der Getriebeentwicklung beliebiger Getriebe auftreten können. Zudem werden im Produktreferenzmodell für Getriebe alle in der Praxis verwendeten Merkmale, Kennwerte und Eigenschaftsvariablen erfasst und den Maschinenelementen zugeteilt. Die im Produktreferenzmodell für Getriebe enthaltenen Merkmale und Eigenschaften decken alle in den Berechnungs- und Simulationsprogrammen (Kapitel 7.2.3) vorkommenden Werte ab.

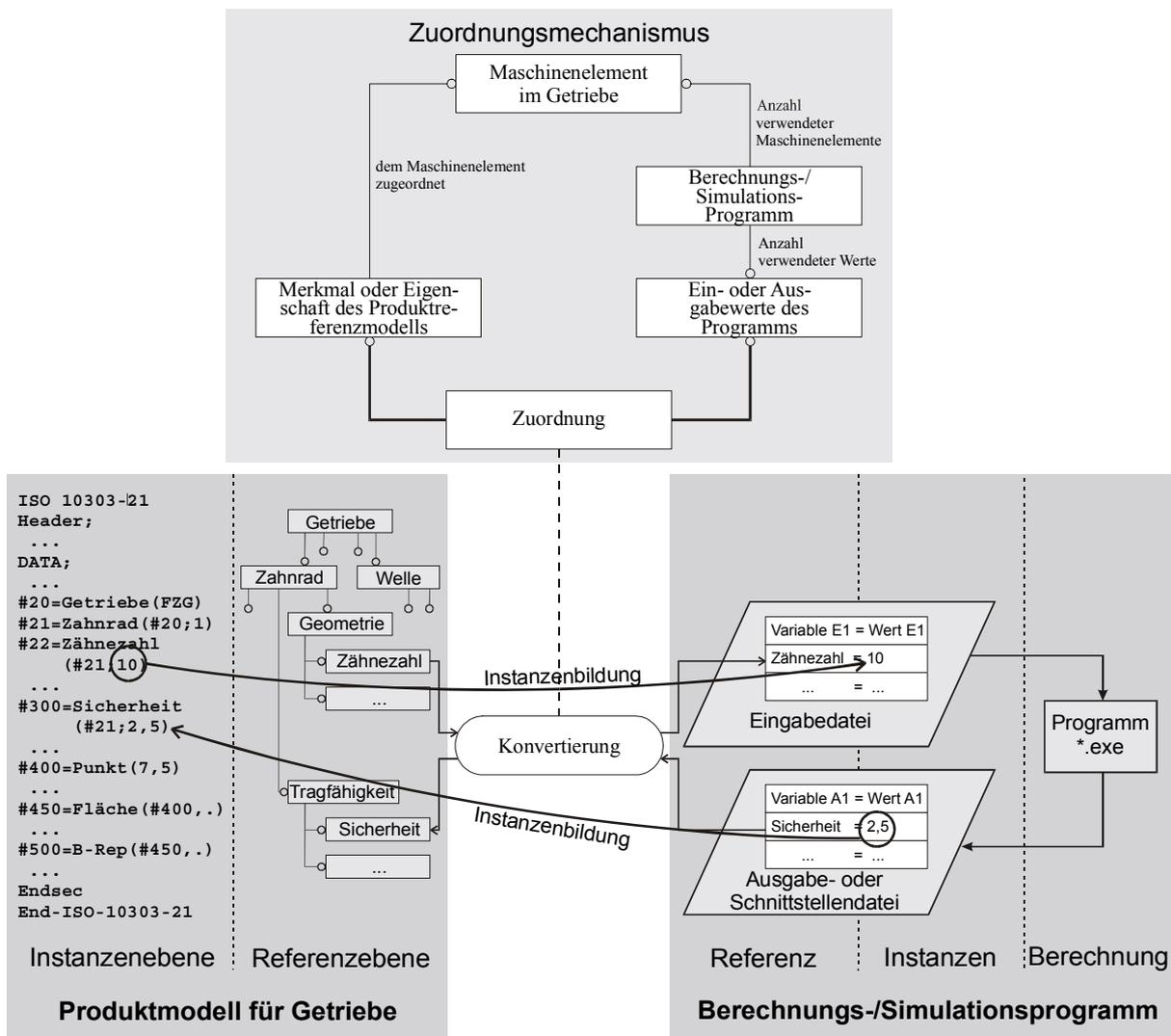


Bild 7-4: Zuordnungsmechanismus der Ein- und Ausgabegrößen von Berechnungsprogrammen zu den Merkmalen und Eigenschaften des Produktreferenzmodells für Getriebe (dargestellt in EXPRESS-G) und Zusammenhang zwischen Referenz- und Instanzenebene der Produktabbildung

Sollen diese oder beliebige andere Berechnungs- und Simulationsprogramme in das rechnerbasierte Programmsystem eingebunden werden, geschieht dies über deren Ein- und Ausgabe-

dateien und den dort auftretenden Ein- und Ausgabegrößen, die mit ihren synonymen Bezeichnungen im Produktreferenzmodell für Getriebe assoziiert werden müssen. Dies bildet die Basis für die Erstellung der entsprechenden Konverter. Den Zuordnungsmechanismus zeigt Bild 7-4.

Der Konverter übernimmt die Aufgabe, den Ein- oder Ausgabewert des Berechnungs-/Simulationsprogramms dem Merkmal oder der Eigenschaft des Produktreferenzmodells zuzuordnen. Dabei werden unterschiedliche Einheiten angeglichen und die Herkunft und die Genauigkeit des Wertes spezifiziert, der z.B. im Entwurf überschlägig und bei der Nachrechnung exakt sein kann.

Das Produktreferenzmodell für Getriebe wird bei der Anwendung in einer Getriebekonstruktion und der dabei durchgeführten Instanzenbildung zu einem Produktmodell. Bild 7-4 zeigt die Zusammenhänge zwischen der Referenz- und der Instanzenebene sowohl für das STEP-Produktmodell als auch für die internen Modelle der Berechnungs- und Simulationsprogramme. Die Produktreferenzebene des STEP-Produktmodells stellt sich als objektorientierte Struktur dar und beschreibt gemäß Kapitel 6.1 alle möglichen Maschinenelemente in beliebigen Getrieben und alle denkbaren Eigenschaften der Getriebe, die diesen zugeordnet werden können. Wird nun eine Instanzenbildung durchgeführt, so ist das Ergebnis eine physikalische, systemneutrale Austauschdatei nach [14].

Durch einen Integrationsvorgang der externen Anwendungsprogramme mit ihren zugehörigen Programmpfaden werden diese in das rechnerbasierte Programmsystem eingebunden. Das rechnerbasierte Programmsystem übernimmt für sie Programmstart, Programmende, Daten-, Dateibesorgung, Archivierung der Daten und Dateien und steuert die externe Anwendung. Der Ablauf des externen Programms erfolgt autonom. Besonders wichtig bei der Einbindung externer Programme ist die Anpassung der Konverter des rechnerbasierten Programmsystems an die von den externen Programmen verwendeten Daten- und Dateiformate.

Bei den Berechnungs- und Simulationsprogrammen ist die Referenzebene nicht so explizit ausgeprägt, nicht von der Instanzenebene getrennt und beide Ebenen in nur einer Datei abgebildet. Unter Referenz wird dort der Variablenname verstanden, der in den Ein- und Ausgabedateien oder programmintern verwendet wird. Jeder Referenz ist in der Ein- oder Ausgabedatei eine konkrete Instanz zugeordnet. Bei einem Programmstart werden aus der Eingabedatei gleichzeitig Referenzvariable und Wert eingelesen und nach Programmdurchlauf zu einer Ausgabedatei ähnlichen Aufbaus verarbeitet.

Die Zuordnung zwischen dem Produktmodell für Getriebe und den Berechnungs-/Simulationsprogrammen kann, muss aber nicht bei jeder Datenübertragung neu getroffen werden. Für Programme, die als externe Anwendung fest in das rechnerbasierte Programmsystem integriert sind, erfolgt die Zuordnung nur ein einziges Mal in der Benutzungsoberfläche. Ab diesem Zeitpunkt ist eine vollständige redundanzfreie Kommunikation zwischen Produktmodell für Getriebe und Berechnungs-/Simulationsprogramm möglich.

7.5 Anpassung und Eingliederung des rechnerbasierten Programmsystems für Getriebe in vorhandene Strukturen

Kapitel 7.3 und 7.4 haben gezeigt, wie das rechnerbasierte Programmsystem um Berechnungs-/Simulationsprogramme erweitert und wie dessen Oberfläche an die Belange der Anwendung angepasst werden kann.

Eine Betrachtung des in Bild 7-5 beispielhaft dargestellten informationstechnischen Umfelds zeigt, dass noch viel mehr Programme und Systeme in enger Kommunikation mit dem Produkt und dessen Daten stehen.

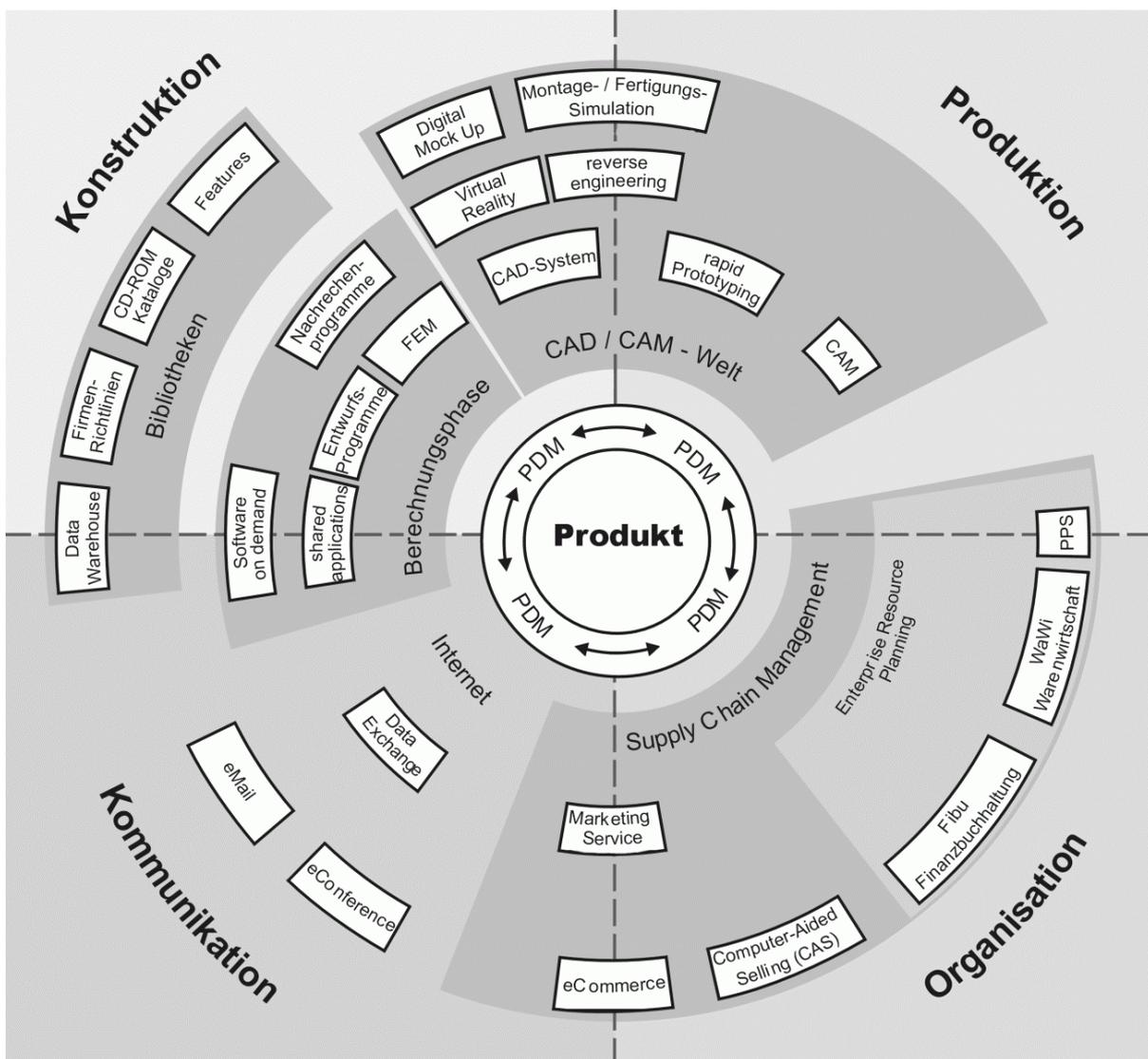


Bild 7-5: Beispiel für ein informationstechnisches Umfeld mit allen Programmen und Systemen, die sich um das im Zentrum befindliche Produkt anordnen

Um das sich im Zentrum befindliche Produkt scharen sich die unterschiedlichsten Anwendungen aus Konstruktion, Produktion, Organisation und Kommunikation. Das rechnerbasierte

Programmsystem ist im Stande, mit allen diesen Anwendungen zu kommunizieren, seine im STEP-Produktmodell befindlichen Daten und die Daten und Dateien des Prozesses, die in der Datenbank für Prozesshistorie gespeichert sind, weiterzugeben und umgekehrt Daten und Dateien anderer Programme aufzunehmen. Dem direkt das Produkt umkreisenden Produkt-Datenmanagement (PDM) werden dann die Daten des rechnerbasierten Programmsystems zur Verfügung gestellt. Dies funktioniert über den in Bild 5-19 dargestellten Mechanismus über die externe Datenschnittstelle.

8 Anwendung und Funktion des rechnerbasierten Programmsystems an einem Beispielgetriebe

Nach Kapitel 5 ist das rechnerbasierte Programmsystem für unterschiedliche Produkte in bestehenden informationstechnischen Umgebungen einsetzbar. Um seine Funktionsweise besser beschreiben zu können, zeigte Kapitel 7 die Anpassung an den Entwicklungsprozess für Getriebe. In der Praxis bieten die in Kapitel 7.2 vorgestellten und in der Antriebstechnik bewährten Programme eine intensive Unterstützung in den einzelnen Entwicklungsphasen an. Nachteil dieser Programme ist ihre Fokussierung auf bestimmte Maschinenelemente, spezifische Probleme jeweiliger Entwicklungsschritte und ihre Beschränktheit bei der Weitergabe oder Austauschbarkeit gewonnener Daten und Dateien. Genau hier setzt das rechnerbasierte Programmsystem an. Es verknüpft die einzelnen Programme und Systeme im Sinne des Entwicklungsprozesses und organisiert alle dabei entstehenden Daten und Dateien. Zusätzlich übernimmt es die Kurz- und Langzeitarchivierung, was einer Aufzeichnung des gesamten Entwicklungsprozesses und der Ablage aller dort verwendeten Daten und Dateien entspricht. Die Basis zur Ablage der Produktdaten ist das STEP-Produktmodell, welches nach jedem Entwicklungsschritt aufgezeichnet wird.

In diesem Kapitel wird ein Getriebe mit Hilfe des rechnerbasierten Programmsystems entwickelt. Das verwendete Getriebe ist ein Referenzgetriebe, welches für Testläufe von Berechnungs- und Simulationsprogrammen [W14] genutzt wird. Ziel dieses Kapitels ist es nicht, das Getriebe zu optimieren oder neue Berechnungs- oder Simulationsergebnisse zu liefern, sondern die Methode des Entwickelns des Getriebes mit der Unterstützung des rechnerbasierten Programmsystems aufzuzeigen. Dabei können allerdings insbesondere im Entwurf abweichende Berechnungsergebnisse erzielt werden, da sich das neu erstellte Getriebe-Auslegungsprogramm auf eine Wissensbasis ausgeführter Standardgetriebe stützt.

8.1 Vorstellung des Beispielgetriebes

Als Beispielgetriebe dient das 1986 an der FZG entwickelte virtuelle FVA-Getriebe [R8][W14], nachfolgend nur als „Getriebe“ bezeichnet.

Das Getriebe hat eine Antriebsleistung von 345 kW, eine Antriebsdrehzahl von 3000 min⁻¹ und ein sich ergebendes Antriebsdrehmoment von 1100 Nm. Die Gesamtübersetzung der ursprünglichen Auslegung von ca. 13 erfolgt über eine Kegelradstufe und je einer schräg- und geradzahnten Stirnradstufe. Daraus ergibt sich eine Abtriebsdrehzahl von 228.5 min⁻¹ bei einem Abtriebsdrehmoment von 14440 Nm. Bild 8-1 zeigt den Aufbau des ursprünglichen Getriebes im Schnitt.

Ziel der Auslegung, geometrischen Modellierung und Nachrechnung/Simulation mit dem rechnerbasierten Programmsystem ist es, die Abtriebsdaten bei Einhaltung der Antriebsdaten mit einer Toleranz von $\pm 3\%$ zu treffen. Weiterhin muss der 90° Winkel zwischen An- und Abtriebswelle erhalten bleiben. Für die anderen Getriebewerte kann das ursprüngliche Getriebe eine Orientierung geben. Beispielsweise scheint die Verwendung einer Kegelradstufe

mit einem dahinter geschalteten, 2-stufigen Stirnradgetriebe eine durchaus sinnvolle Anordnung zur Erreichung der Gesamtübersetzung zu sein.

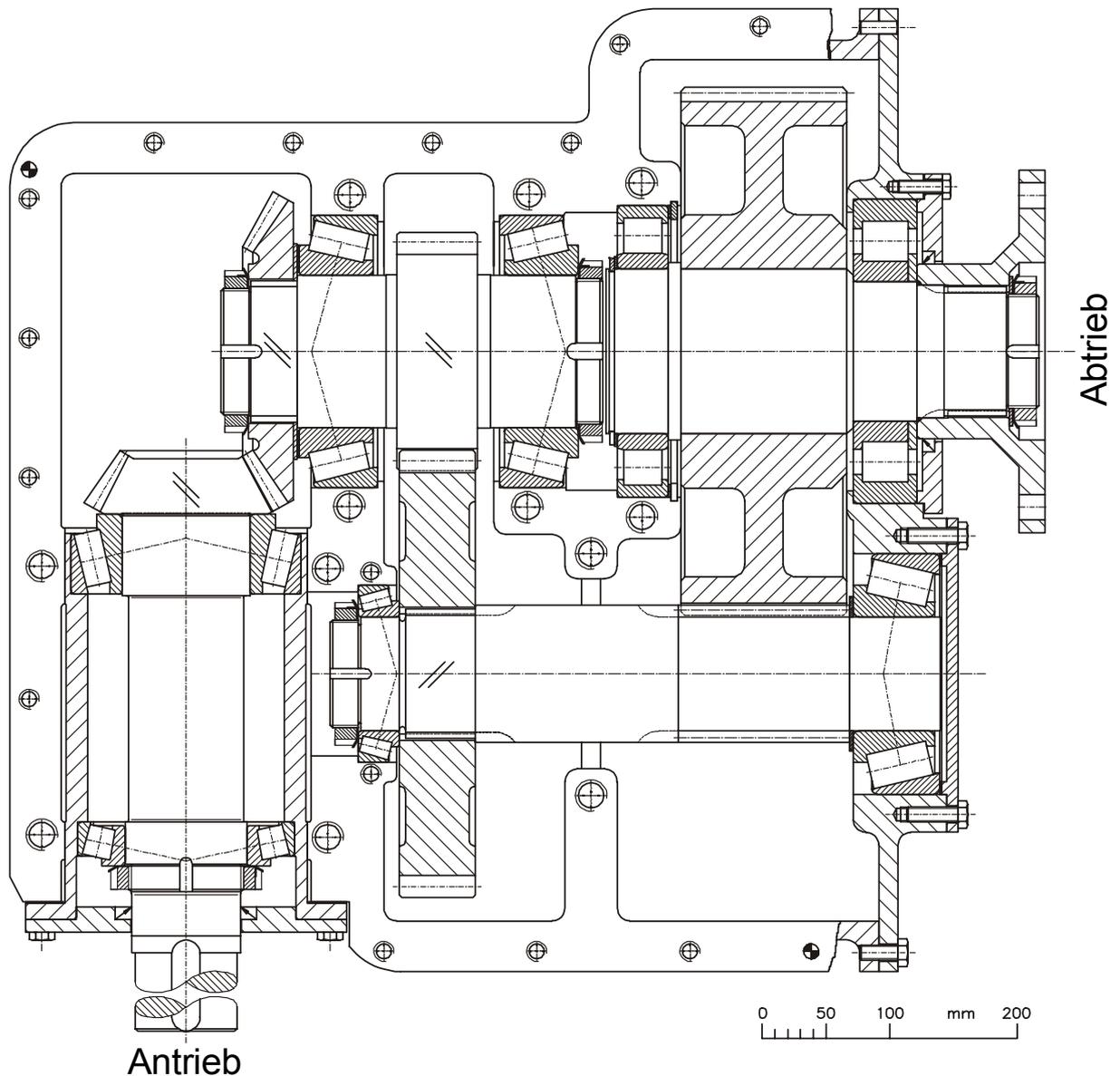


Bild 8-1: Ursprüngliches virtuelles FVA-Getriebe [R8][W14]

Aufgrund der Vielfältigkeit an Maschinenelementen und seiner komplexen Produktstruktur eignet sich das Getriebe sehr gut, um neue Methoden der digitalen Getriebeentwicklung anhand des rechnerbasierten Programmsystems von den frühen Phasen bis hin zu Fertigungszeichnungen vorzustellen. Darauf soll in Kapitel 8 der Schwerpunkt gesetzt werden, die erzielten wertebehafteten Ergebnisse der Berechnungen und Simulationen sind für diese Arbeit von untergeordnetem Interesse.

8.2 Initialisierung des rechnerbasierten Programmsystems

Zu Beginn des Getriebeentwicklungsprozesses mit Hilfe des rechnerbasierten Programmsystems wird ein neues Projekt angelegt. Mit Hilfe eines Eingabefensters wird ein Projektname definiert und die Programmpfade angegeben, unter welchen die Datenbank für die Prozesshistorie und alle bei der Produktentwicklung entstehenden Daten und Dateien abgelegt werden. Die Ordnung und den Aufbau innerhalb der Datenbank für Prozesshistorie (Bild 5-8) übernimmt das rechnerbasierte Programmsystem. Nun kann mit der eigentlichen Getriebekonstruktion begonnen werden. Für eine erste Auslegung des Getriebes kann zum Beispiel das interaktive Getriebe-Auslegungsprogramm aus der grafischen Benutzungsoberfläche heraus gestartet werden. Die grobe Dimensionierung des Getriebes umfasst die Aufteilung der Gesamtübersetzung auf einzelne Getriebestufen, die Festlegung der Getriebeart je Stufe und eine Grobdimensionierung typischer Getriebekennwerte.

8.3 Getriebeentwurf

Der Beschreibung der traditionellen Vorgehensweise in der Getriebeentwicklung folgen die Funktionsweise des interaktiven Getriebe-Auslegungsprogramms und das dabei entstehende Zusammenspiel der Programmkomponenten des rechnerbasierten Programmsystems.

8.3.1 Traditionelle Vorgehensweise

Das vorgestellte Getriebe wird im Folgenden mit dem rechnerbasierten Programmsystem entwickelt. Bei der Getriebeauslegung, dem ersten Schritt der Getriebekonstruktion, sind die in Kapitel 8.1 genannten Kennwerte einzuhalten und als vorgegebene Aufgabenstellung zu betrachten.

Die konventionelle Vorgehensweise bei der Getriebeauslegung gestaltet sich nach NIEMANN&WINTER [N2] wie folgt:

- Wahl von Getriebeart sowie Anschluss an An- und Abtrieb.
- Aufteilen der Gesamtübersetzung auf die Getriebestufen.
- Bestimmung der vorläufigen Hauptabmessungen der einzelnen Getriebestufen durch Überschlagsrechnungen, die auf Erfahrungsangaben aus dem jeweiligen Anwendungsgebiet basieren.
- Vorläufiges Festlegen der charakteristischen Verzahnungsdaten sowie des Werkstoffs und der Wärmebehandlung der Zahnräder.

Für die Auslegung eines Getriebes ist Erfahrung notwendig. Bei der Aufteilung der Gesamtübersetzung auf die einzelnen Getriebestufen und der damit verbundenen Bestimmung der

vorläufigen Hauptabmessungen sind verschiedene Kennwerte (z.B. K^*-1 , U-Faktor²) sinnvoll anzunehmen. Richtlinien dieser Kennwerte können einschlägiger Fachliteratur (z.B. [B4][N2]) entnommen werden. Die dort angegebenen Werte sind sehr allgemein gehalten und präzisieren sich erst mit dem durch Erfahrung gewonnenen Wissen. Die Richtigkeit der Annahmen bestätigt sich erst bei der Nachweisrechnung gegen Ende des Getriebeentwicklungsprozesses. Wurden Kennwerte zu ungenau angenommen, muss die Getriebeauslegung korrigiert werden. Dies ist mit einem Kosten- und Zeitaufwand verbunden, da der Konstruktionsprozess nochmals durchlaufen werden muss.

8.3.2 Getriebeauslegung mit dem interaktiven Getriebe-Auslegungsprogramm

Das rechnerbasierte Programmsystem bietet zur Unterstützung des in seinem ganzen Umfang komplexen Getriebeauslegungsprozesses ein interaktives Getriebe-Auslegungsprogramm an, dessen Möglichkeiten innerhalb der Getriebeauslegung in Bild 8-2 dargestellt sind.

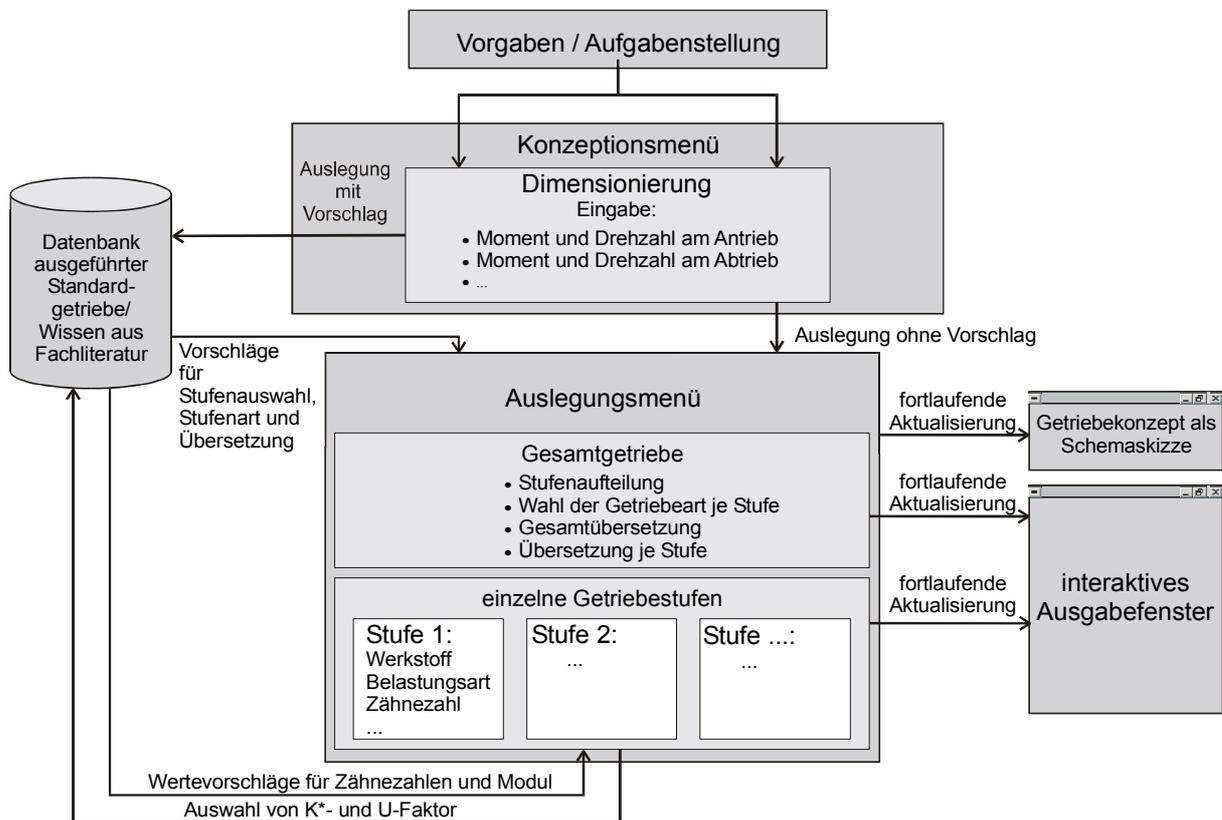


Bild 8-2: Überblick über die Funktionsweise des interaktiven Getriebe-Auslegungsprogramms

¹ Der K^* -Faktor [N/mm^2] ist nach NIEMANN&WINTER [N2] ein vereinfachter Tragfähigkeitskennwert für die Hertzsche Pressung.

² Der U-Faktor [N/mm^2] ist nach NIEMANN&WINTER [N2] ein vereinfachter Ausdruck für die Zahnfußspannung und dient zur überschlägigen Berechnung der Bruchsicherheit einer Verzahnung.

Die Eingabe der Kennwerte der Aufgabenstellung wird im Bild 8-3 gezeigten Konzeptionsmenü des interaktiven Getriebe-Auslegungsprogramms vorgenommen. Dort wird eine überbestimmte Dateneingabe verhindert, indem entsprechende Eingabefelder gesperrt werden und dafür das jeweils berechnete Ergebnis angezeigt wird. Im Folgenden kann ausgewählt werden, ob eine Auslegung des gesamten Getriebes und der Stufenaufteilung mit oder ohne Vorschläge durch das Getriebe-Auslegungsprogramm ausgeführt werden soll.

	Eingabe	berechnet
Eingangsleistung [kW]	345	
Eingangsdrehmoment [Nm]		1098.17
Eingangsdrehzahl [1/min]	3000	
Gesamtübersetzung []	13	
Abtriebsdrehmoment [Nm]		14201.58
Abtriebsdrehzahl [1/min]		231.98
Lage Antrieb/Abtrieb [°]	90	

Buttons: Auslegung ohne Vorschlag, **Auslegung mit Vorschlag**, Löschen, Abbrechen

Bild 8-3: Konzeptionsmenü des Getriebe-Auslegungsprogramms (Screenshot)

Bei Auswahl von „**Auslegung mit Vorschlag**“ springt das Getriebe-Auslegungsprogramm in das in Bild 8-4 gezeigte Auslegungsmenü und es wird automatisch eine sinnvolle Stufenanzahl gewählt und die empfohlenen Stufenarten vorgegeben sowie die anderen bekannten Felder ausgefüllt. Das Getriebe-Auslegungsprogramm beruft sich hierfür auf eine Datenbank ausgeführter Standardgetriebe und hinterlegte Fachliteratur [N2] und nimmt selbstständig sinnvolle Werte als grobe Richtlinie für die Auslegung an, die durch den Anwender im Anschluss auch verändert werden können.

Wird hingegen eine „**Auslegung ohne Vorschlag**“ ausgewählt, weil das Getriebekonzept im Kopf des Entwicklers vorhanden ist oder ein bestehendes Getriebe neu entwickelt werden soll, müssen im Auslegungsmenü die Stufenaufteilung und Getriebearten je Stufe selbst ausgewählt werden.

Beiden Vorgehensweisen gemeinsam ist die Anwahl eines „**Einzelmenüs**“, wodurch in einem weiteren Fenster (Bild 8-5) die Auslegung einer einzelnen Getriebestufe gestartet wird. Dort werden Werkstoff, Belastungsart und Lagerungsart sowie die Mindest-Sicherheiten gegen Zahnbruch und Grübchen vorgegeben. Nach Eingabe des optionalen Achsabstands und des Schrägungswinkels macht das Getriebe-Auslegungsprogramm Wertevorschläge für die

Zahnbreite, die Zähnezah und den Modul. Hierfür wird über die o.g. K*- und U-Faktoren auf die Datenbank ausgeführter Standardgetriebe zurückgegriffen.



Bild 8-4: Eingabefenster Auslegungsmenü gesamtes Getriebe (Screenshot)

Bei der Auslegung werden fortlaufend die Schemaskizze und das interaktive Ausgabefenster aktualisiert. Die Schemaskizze visualisiert die funktionale Anordnung der Getriebebestandteile und zeigt den Momentenfluss durch das Getriebe auf. Das interaktive Ausgabefenster enthält die Ergebnisse nach Überschlagsrechnungen mit Hilfe der Eingabewerte. Aufgrund der zeitgleichen Wertausgabe bei einer einzelnen Werteeingabe kann auf ein bestimmtes Ergebnis durch Variation der Eingabegrößen hingearbeitet werden. Dadurch wird der Zusammenhang der Getriebewerte sehr gut deutlich und zudem ein Lerneffekt erzielt.

Interaktives Getriebe-Auslegungsprogramm

Auslegung einzelne Stufe

Werkstoff: 16 MnCr5E

Belastungsart: schwellend, leichte Stoesse

Lagerung auf der Welle: aussermitig

Mindestsicherheiten: S_F 2.5, S_H 2.0

Achsabstand [mm]: a 254 (optional)

Schrägungswinkel [°]: beta 11

	berechnet	gewählt
Breite [mm]	b 62.735	b 65.8
Zähnezahl	z_1 39.237	z_1 41
	z_1HF 43.123	
Normalmodul [mm]	m_n 3.869	m_n 4

Buttons: Übernehmen, Löschen, Abbrechen

Bild 8-5: Eingabefenster Auslegungs Menü Stirnradstufe (Screenshot)

8.3.3 Diskussion der mit dem Getriebe-Auslegungsprogramm erzielten Ergebnisse

Das interaktive Getriebe-Auslegungsprogramm schlägt für die Anordnung des Beispielgetriebes nach Eingabe der genannten Vorgaben aufgrund der 90°-Lage von An- und Abtrieb eine Kegelradstufe und zum Erreichen der gewünschten Übersetzung zwei Stirnradstufen vor, was auch dem Aufbau des ursprünglichen Getriebes nach Bild 8-1 entspricht.

Tabelle 8-1 zeigt einen Ausschnitt der Verzahnungsdaten des mit dem Getriebe-Auslegungsprogramm berechneten Getriebes im Vergleich zur ursprünglichen Auslegung. Dabei fällt auf, dass bei der überschlägigen Auslegung für die beiden Stirnradstufen von der Ausgangsversion des Getriebes abweichende Verzahnungsgrößen ausgewählt worden sind. Das liegt in der Tatsache begründet, dass sich das Getriebe-Auslegungsprogramm auf Standardgetriebe stützt, aus denen Wertevorschläge für die Zähnezahlen und den Modul hervorgehen. Wird der Achsabstand nicht festgelegt und die ursprünglichen Daten für die Zähnezahlen, den Modul und den Schrägungswinkel eingegeben, schlägt das Getriebe-Auslegungsprogramm unterschiedliche und stark abweichende Achsabstände für die beiden Stirnradstufen vor. Da jedoch die Vorgabe eines gleichen Achsabstands für beide Stirnradstufen eingehalten werden soll, wurde der Achsabstand für beide Stirnradstufen gleich vorgegeben. Dies hat zur Folge, dass automatisch Wertevorschläge für die Zähnezahlen, den Modul und den Schrägungswinkel generiert werden, die sich wiederum auf Standardgetriebe stützen und für einen gleichen

Achsabstand von der ursprünglichen Version des Getriebes abweichen. Im Folgenden wird mit den Daten des Getriebe-Auslegungsprogramms weitergearbeitet, was in Bezug auf die Gesamtübersetzung und Abtriebsmoment und –drehzahl keine nennenswerten Auswirkungen hat.

	Vergleich der Ergebnisse der Kegelradstufe		Vergleich der Ergebnisse der Stirnradstufe 1		Vergleich der Ergebnisse der Stirnradstufe 2	
	Getriebe-Auslegungsprogramm	ursprüngliche Auslegung	Getriebe-Auslegungsprogramm	ursprüngliche Auslegung	Getriebe-Auslegungsprogramm	ursprüngliche Auslegung
Zähnezahl Ritzel z_1	20	20	41	20	17	25
Zähnezahl Rad z_2	33	33	83	39	66	102
Normalmodul m	5	5	4	8	6	4
Schrägungswinkel β	35°	35°	11°	21,7°	0°	0°
Übersetzung i	1,65	1,65	2,02	1,95	3,88	4,08
Achsabstand a	-	-	254 mm	254 mm	254 mm	254 mm

Tabelle 8-1: Auszug von mit dem Getriebe-Auslegungsprogramm berechneten Verzahnungsdaten des Getriebes im Vergleich zur ursprünglichen Auslegung

8.3.4 Datenweitergabe nach Beendigung des interaktiven Getriebe-Auslegungsprogramms

Wird das interaktive Getriebe-Auslegungsprogramm beendet, werden alle Eingabegrößen in eine native Datei gespeichert, die dem Format typischer Eingabedateien von FVA-Programmen entspricht. Durch die entsprechenden Anwendungsfunktionen „Konvertierung“ und „Datenrückführung“ werden die Daten im Prozessmanagement-System konvertiert, in das STEP-Produktmodell eingegliedert und dort gespeichert (Bild 5-18). Die native Datei selbst sowie das STEP-Produktmodell werden in der Datenbank für Prozesshistorie zur Archivierung abgelegt. Ist die Auslegung beendet, kann für die nun folgende Gestaltung und Berechnung des Getriebes auf das STEP-Produktmodell aufgebaut werden. Es müssen keine Daten manuell transferiert oder neu eingegeben werden. Dadurch wird Zeit gespart, die Qualität des Prozesses erhöht und Fehler vermieden.

Nach der Eingliederung der in der Getriebeauslegung entstandenen Daten in das STEP-Produktmodell kann mit Hilfe der Basisfunktionen des rechnerbasierten Programmsystems ein Überblick über den momentanen Entwicklungsstand des Getriebes dargestellt werden. Die Anordnung des Getriebes ist nun festgelegt. Die Zahnradpaarungen und die damit verbunde-

nen einzelnen Übersetzungen sind ausgewählt. Für die charakteristischen Verzahnungsgrößen der einzelnen Zahnräder sowie deren Werkstoffe sind Vorschläge getroffen worden.

Ein mögliches Vorgehen bei der Entwicklung des Getriebes von der Konzipierung bis hin zur Nutzung des fertigen Produktmodells für Fertigungszeichnungen oder separate externe Anwendungen zeigt Bild 8-6.

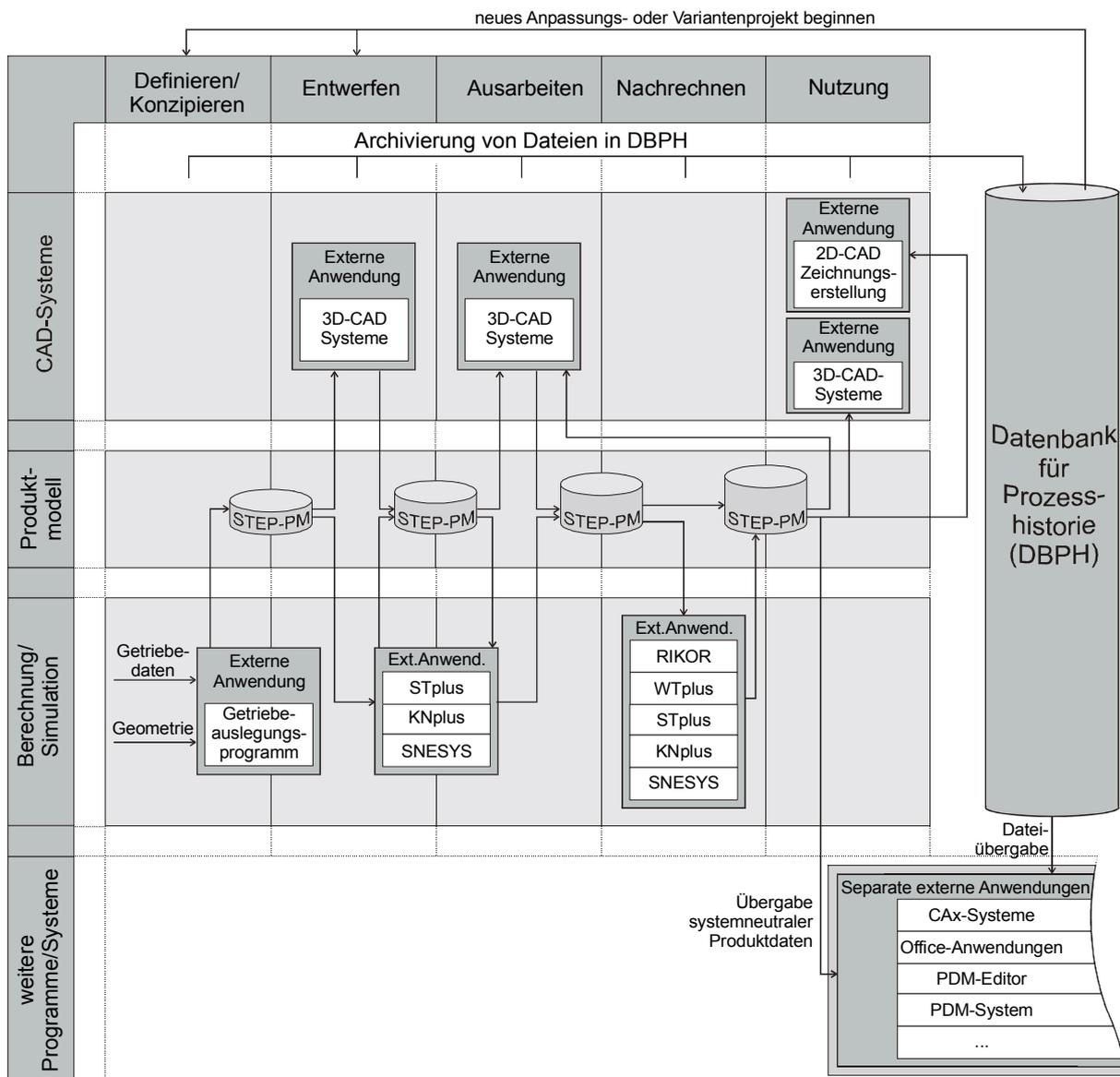


Bild 8-6: Vorgehensweise und eingesetzte Methoden und Programme bei der Entwicklung des Getriebes

Zur Ausarbeitung des Getriebes wird nachfolgend die Anwendung der 3D-CAD-Systeme und Berechnungs- und Simulationsprogramme gezeigt. Die Ausarbeitung erfolgt nicht nacheinander, sondern in einem Kreislauf zwischen Geometriemodellierung (Kapitel 8.4) und Berechnung/Simulation (Kapitel 8.5). Folgende Tätigkeiten sind beispielsweise am Getriebe durchzuführen:

- Gestaltung der Wellen und Welle-Nabe-Verbindungen
- Nachrechnung, Simulation und Optimierung einzelner Zahnräder, Zahnradpaarungen, Wellen und des gesamten Getriebes nach verschiedenen Gesichtspunkten
- Auswahl der Lagerungsart und der einzelnen Lagertypen
- Werkzeugauswahl zur Zahnradherstellung
- Entwurf des Gehäuses mit Gehäuseteilungen und sinnvollen Montagevorkehrungen
- Detaillierung des Getriebes im 3D-CAD-System
- Schmierstoff-, Dichtungsauswahl und Festlegung des Ölhaushalts
- Werkstoffwahl weiterer Getriebeelemente

8.4 Geometriemodellierung

Die Werkzeuge zur Geometriemodellierung sind 3D-CAD-Systeme. Mit ihrer Hilfe ist es möglich, nach der Getriebeauslegung weitere Bauteile zu entwerfen und bestehende zu detaillieren, um sie dann in einem Kreislauf von Gestaltung und Berechnung weiter zu optimieren.

Der Start der zur Geometriemodellierung benötigten 3D-CAD-Systeme erfolgt als externe Anwendung aus der grafischen Benutzungsoberfläche des rechnerbasierten Programmsystems heraus. Das weitere Vorgehen entspricht dem in Kapitel 5.3.2 beschriebenen externen Datenfluss mit einem systemneutralen Datenformat. Durch den Startbefehl der externen Anwendung wird die Anwendungsfunktion „Dateibesorgung“ aktiviert. Nach der Kopie des aktuellen STEP-Produktmodells wird dies gesperrt und die Kopie an das Prozessmanagement-System übergeben. Dies übernimmt die Beschaffung der vom CAD-System benötigten systemneutralen Daten und stellt diese in einer STEP-Datei zur Verfügung, welche unter Verwendung der Anwendungsfunktion „Programm starten“ dem externen 3D-CAD-System übergeben wird. Während der Nutzung eines 3D-CAD-Systems wartet das rechnerbasierte Programmsystem auf das Ende der externen Anwendung. Die Datenablage bei Programmende erfolgt in umgekehrter Reihenfolge und endet mit der Freigabe des aktualisierten STEP-Produktmodells. Die Historie der Produktentwicklung wird durch die Ablage des ursprünglichen STEP-Produktmodells in der Datenbank für die Prozesshistorie sichergestellt. Die Eingliederungspunkte der CAD-Systeme im rechnerbasierten Entwicklungsprozess sind in Bild 8-6 dargestellt.

Es kann auch eine Geometriemodellierung oder –anwendung in den Berechnungs- und Simulationsprogrammen erfolgen (Kapitel 8.5). In diesem Fall wird von „Ersatzgeometrie“ gesprochen, da es sich zumeist um idealisierte Geometriemodelle handelt, die keine CAD-Detaillierungen enthalten.

8.5 Berechnung und Simulation

Im Kreislauf von Geometriemodellierung und Berechnung/Simulation dient die Nachweisrechnung der Bestätigung einer ausreichenden Sicherheit, einer Festigkeitsabsicherung und einer Garantie der Produktfunktion der einzelnen Bauteile und des gesamten Getriebes. Weiterhin wird eine Optimierung hinsichtlich Gewicht, einer optimalen Materialausnutzung und einer wirtschaftlichen Herstellbarkeit und Montage angestrebt. Hierzu werden die in der Antriebstechnik anerkannten Nachrechnungsprogramme aus Kapitel 7.2.3 verwendet. Je nach Programm können einzelne Bauteile oder das gesamte Getriebe nachgerechnet werden.

Die in das rechnerbasierte Programmsystem integrierten Berechnungsprogramme verwenden ein natives Datenformat und nicht wie die 3D-CAD-Systeme das systemneutrale STEP-Datenformat. Ihre Nutzung erfordert daher eine Konvertierung vorhandener Daten oder Dateien in ein natives, für die Berechnungsprogramme nutzbares Datenformat. Bild 5-18 zeigt diesen Vorgang. Hierbei werden vom Prozessmanagement-System des rechnerbasierten Programmsystems die benötigten Daten aus dem STEP-Produktmodell selektiert und in eine entsprechende Eingabedatei konvertiert. Falls detaillierte 3D-CAD-Geometrie im STEP-Produktmodell vorhanden ist, wird diese in eine Ersatzgeometrie umgewandelt, die von den Berechnungsprogrammen verstanden wird. Während des Datenaustausches zwischen den einzelnen Programmen und dabei eventuell benötigten Konvertierungen wird ein enormer Zeitgewinn durch den automatischen Datenaustausch und der Konvertierung der Datenformate erreicht und es werden zudem Datenredundanzen vermieden.

In der grafischen Benutzungsoberfläche des rechnerbasierten Programmsystems werden vor dem Start eines externen Berechnungsprogramms die Bauteile, Baugruppen oder das komplette Getriebe ausgewählt, die im jeweiligen Berechnungsprogramm verarbeitet werden sollen. Das Prozessmanagement-System übernimmt dann die Beschaffung der benötigten Daten aus dem STEP-Produktmodell oder der Datenbank für die Prozesshistorie. Bei der Datenbeschaffung ist der Unterschied zu den 3D-CAD-Systemen der Geometriemodellierung ein von den Berechnungsprogrammen gefordertes natives Datenformat (Bild 5-18). Dies bedarf einer zusätzlichen Konvertierung der Daten aus dem STEP-Format durch das Prozessmanagement-System. Da die Berechnungs- und Simulationsprogramme nicht alle Daten aus der Geometriemodellierung benötigen, bildet ein Konverter im Prozessmanagement-System ein Ersatzgeometriemodell für die Berechnungs- und Simulationsprogramme. Die Datenablage nach Beendigung der Berechnung erfolgt durch die Anwendungsfunktionen „Konvertieren“ und „Datenrückführung“ im Prozessmanagement-System (siehe auch Kapitel 5.3.2). Eine Archivierung erfolgt, wie auch bei der Verwendung des interaktiven Getriebe-Auslegungsprogramms, durch Ablage der nativen Datei nach Beendigung des externen Programms und des STEP-Produktmodells, vor der Eingliederung der konvertierten Daten, in der Datenbank für die Prozesshistorie.

In Bild 8-6 sind symbolisch die Anknüpfungspunkte der Berechnungs- und Simulationsprogramme in den letzten Stadien des Grobentwurfs und einer beginnenden Ausarbeitung sowie innerhalb der Nachrechnung angegeben. Zu erkennen ist, dass der Ablauf nicht prozedural verläuft, sondern durch Rücksprünge und Kreisläufe charakterisiert ist.

8.6 Visualisierung aller Daten und der Produktstruktur

Das datentechnische Resultat während und jeweils nach Entwurf, Ausarbeitung und Nachweisrechnung ist das STEP-Produktmodell, welches alle bisherigen Daten des Getriebes beinhaltet. Wie in Kapitel 6 beschrieben ist das objektorientiert aufgebaute Produktmodell im Stande, alle anfallenden Getriebedaten in einer maschinenlesbaren Form aufzunehmen. Da im Vergleich zu früheren, vom Menschen lesbaren Ausgabedateien von Berechnungs- und Simulationsprogrammen ein „Lesen“ der Datenbank nicht so einfach möglich ist, wird softwaretechnisch der Inhalt des STEP-Produktmodells und auch der Datenbank für Prozesshistorie in eine dem Menschen gut verständliche Form umgesetzt.

Durch die Nutzung verschiedener externer Programme mit verschiedenen Sichten auf Bauteile und Baugruppen ist es schwierig, die Zusammenhänge der Daten und der Bauteilstruktur zu erfassen und abzubilden. Dies ist für die Kreisläufe der Nachweisrechnung und CAD-Modellierung der Bauteile, Baugruppen oder des gesamten Getriebes und der damit verbundenen Dateneingabe notwendig. Um den Überblick über die Produktstruktur mit Hierarchien, den Dokumenten und den anderen Zusammenhängen des Produkts zu fördern, bietet das rechnerbasierte Programmsystem verschiedene Basisfunktionen an:

- Das **Karteikartensystem** vereint alle geometrischen und nichtgeometrischen Daten des Getriebes, eines Bauteils oder einer Baugruppe in Form von Text.
- Der **3D-Strukturbaum** ermöglicht die Darstellung der zweidimensionalen hierarchischen Getriebebauproduktstruktur. Mit Hilfe der dritten Dimension werden gleichzeitig die mehrfachen Beziehungen und funktionale Verknüpfungen zwischen den einzelnen Bauteilen aufgezeigt, wie sie beispielsweise in Bild 6-3 dargestellt sind.
- Der **Dateiviewer** ermöglicht es alle Dateiformate, die während der Entwicklung entstanden sind, in ihrem ursprünglichen Format zu betrachten. Ein Dateiviewer ist auch Standard innerhalb der Berechnungsprogramme.
- Das **LRZ-Grafiksystem** [W9] dient zur Ansicht der grafischen Ausgabedateien, die beispielsweise Diagramme und Verläufe enthalten. Alle integrierten Berechnungs- und Simulationsprogramme der FVA nutzen für die grafische Ausgabe das LRZ-Grafiksystem.
- Die **2D-Strukturbäume** für Dokumente und Bauteile ermöglichen einen Überblick über alle Bauteile des Getriebes, ihre Versionen und Sichten und den zugehörigen Dokumenten.
- Die **Tabelle der Prozesshistorie** ermöglicht eine lückenlose Nachvollziehbarkeit des Getriebeentwicklungsprozesses und der dabei gegangenen Schritte. Die Archivierung der Daten ermöglicht es, zu jedem Zeitpunkt des Getriebeentwicklungsprozesses zurückzukehren und die dabei erstellten Dokumente neu zu nutzen.

8.7 Weiterverarbeitung und Übergabe von Geometriemodellen und Produktdaten

Das gesamte oder Ausschnitte aus dem STEP-Produktmodell und der Dateien aus der Datenbank für die Prozesshistorie können isoliert und zur Weiterverarbeitung an andere Anwendungen übergeben werden. Bild 8-6 zeigt unter „Nutzung“ die Möglichkeiten zur Weiterverarbeitung der Daten mit den integrierten externen Anwendungen auf und stellt den Mechanismus zur Anknüpfung von separaten externen Anwendungen vor.

8.7.1 Nutzung mit integrierten externen Anwendungen

Eine Nutzung der Produkt- und Prozessdaten mit den integrierten externen Anwendungen wurde bereits umfassend beschrieben und ist auch nach einem Projektabschluss weiterhin möglich.

Ein weiterer Schwerpunkt der integrierten externen Anwendungen sind 2D-CAD-Systeme, die für die automatisierte Erstellung von 2D-Zeichnungen verwendet werden können. Dies ist insbesondere für wiederkehrende Datenblätter oder technische Zeichnungen interessant, die ähnliche geometrische Objekte mit ihren nichtgeometrischen Produktdaten vereinen.

Mit einem entsprechenden Konverter zur Zeichnungserstellung und der in Bild 8-7 gezeigten Vorgehensweise ist die automatische Erzeugung solcher Datenblätter und Zeichnungen leicht möglich. Hierfür ist zuerst eine Vorlagedatei mit Platzhaltern für nichtgeometrische Daten zu entwerfen. Im konkreten Anwendungsfall ist eine Zeichnung mit integriertem Datenblatt gemeint, welches alle zur Fertigung notwendigen Daten eines Zahnrads enthält. Das darzustellende Zahnrad wird in der grafischen Benutzungsoberfläche ausgewählt und unter Verwendung einer produktbezogenen Basisfunktion im rechnerbasierten Programmsystem separiert. Nach Auswahl der systemneutralen Vorlagedatei im DXF- oder STEP AP201-Format (Kapitel 3.2.4) werden zuerst die nichtgeometrischen Daten aus dem STEP-Produktmodell beschafft. Das Prozessmanagement-System selektiert mit Hilfe eines Konverters alle benötigten Daten und gliedert sie in den dafür vorgesehenen Platzhalter der Vorlagedatei ein. Ein weiterer Konverter übernimmt nun die Beschaffung der Zahnradgeometrie. Hiermit erfolgt die Fertigstellung der 2D-CAD-Zeichnung durch den Konverter unter Verwendung von 2D-Zeichnungsbefehlen, die in einer DXF/STEP AP 201-Bibliothek vorliegen und im Zuge der Erzeugung der Fertigungszeichnung des Zahnrads angesprochen werden. Die erstellte, systemneutrale 2D-Zeichnungsdatei kann von beliebigen 2D-CAD-Systemen mit DXF- oder STEP-Schnittstelle eingelesen, dargestellt und weiterverarbeitet werden.

- **Weiterverarbeitung von Daten in 3D-CAX-Systemen:** Volumenmodelle des Beispielgetriebes können für weitere 3D-CAX-Anwendungen im STEP-AP214-Format von jedem Zeitpunkt der Entwicklung isoliert und an beliebige Programme übergeben werden.
- **Nutzung der Produktdaten in CAE- und PDM-Systemen:** Das STEP-Produktmodell und die darin enthaltenen Daten können an beliebige, das STEP-Format verwendende CAE- und PDM-Systeme übergeben werden.
- **Dokumentation in Office-Anwendungen:** Mit der Weitergabe des systemneutralen STEP-Produktmodells an Office-Anwendungen mit DXF- oder STEP-Schnittstelle kann der Anwender zusätzliche Dokumentationsprogramme nutzen.
- **Verwendung der Daten im PDM-Editor:** Mit ihm können Daten während des gesamten Getriebeentwicklungsprozesses manuell angelegt, beeinflusst und gelöscht werden. Der PDM-Editor lässt sich ebenfalls als externe Anwendung aus der grafischen Benutzeroberfläche des rechnerbasierten Programmsystems starten und ist für manuelle Eingriffe in das STEP-Produktmodell vorgesehen, da er nur Änderungen am Produktmodell zulässt, die mit der STEP-Norm [I4] kompatibel sind, was mit der Anwendung eines gewöhnlichen Editors nicht gegeben ist.

8.8 Ergonomische Unterstützung im rechnerbasierten Programmsystem

Das rechnerbasierte Programmsystem begleitet den Anwender während der gesamten Entwicklung und stellt ihm größtmögliche Freiräume bei Gestaltung und Berechnung im rechnerbasierten Getriebe-Entwicklungsprozess zur Verfügung. Neben der Datenverwaltung und Archivierung der Daten bietet das rechnerbasierte Programmsystem eine Unterstützung durch Grafiken, Modelle und arbeitsbegleitende Hilfestellungen innerhalb der Basisfunktionen an. Deren Darstellung erfolgt durch die grafische Benutzeroberfläche.

Der Aufbau der grafischen Benutzeroberfläche nach Bild 5-6 ist in ihrer Grundstruktur einfach gehalten. So wird eine gute Übersicht gewährleistet. Nach der in Kapitel 7.3 beschriebenen Anpassung der Benutzeroberfläche an die Getriebeentwicklung ergibt sich Bild 8-8. Durch eine klare Trennung von Basis- und Anwendungsfunktionen und der Möglichkeit, alle Basisfunktionen parallel zu nutzen, wird ein klares Navigieren während der Entwicklung und der Nutzung des rechnerbasierten Programmsystems ermöglicht. Die Anordnung der externen Anwendungen folgt dem Getriebeentwicklungsprozess und führt somit den Benutzer. Zusätzlich fördert die grafische Aufbereitung der Oberfläche mit klar gestalteten Abbildungen ihrer Funktionen die Übersicht und Schnelligkeit im Umgang mit dem rechnerbasierten Programmsystem bei der Entwicklung von Getrieben.

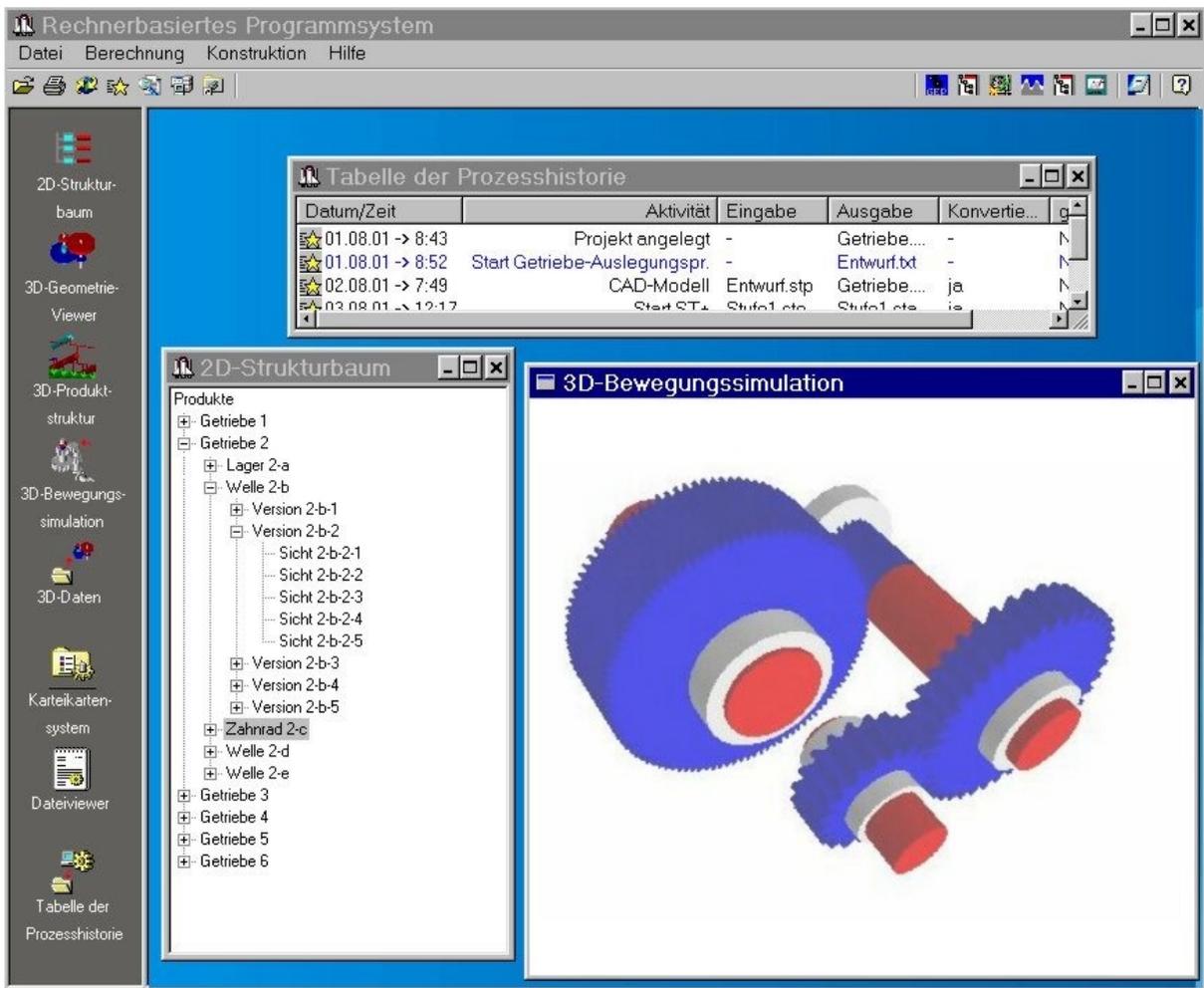


Bild 8-8: Grafische Benutzeroberfläche (Screenshot)

Die eigenständigen nativen Dateien der Berechnungsprogramme sind durch die Archivierung des rechnerbasierten Programmsystems in der Datenbank für die Prozesshistorie abgelegt und können nach der Beschaffung durch das Prozessmanagement-System in einem Viewer betrachtet werden. Neben der Ansicht der Ein-, Ausgabe- und Grafikdateien mit entsprechenden Viewern kann der Inhalt des STEP-Produktmodells und der Datenbank für Prozesshistorie innerhalb des rechnerbasierten Programmsystems dargestellt werden. Dies erfolgt durch die produktmodell- und prozessbezogenen Basisfunktionen (Bild 5-13). Sie ermöglichen einen Überblick über die Ergebnisse der Entwicklung und jedes einzelnen Schrittes während des Getriebe-Entwicklungsprozesses. Die Basisfunktionen führen jede für sich eine Aufteilung und Ordnung der Daten und somit eine geordnete Darstellung der Ergebnisse des Entwicklungsprozesses durch. Eine Neuerung ist das nach Kapitel 5.3.1.4 beschriebene Zusammenspiel dieser Basisfunktionen (Bild 5-16). So werden selbst sehr komplexe Getriebeentwicklungen transparent dargestellt, Zusammenhänge entschlüsselt und Schwachstellen aufgedeckt. Dadurch wird der Entwicklungsprozess qualitativ verbessert, beschleunigt und es werden die Entwicklungskosten gesenkt.

9 Einordnung und Positionierung der rechnerbasierten Produktentwicklung

9.1 Einordnung und Weiterverwendung der Ergebnisse und entwickelten Methoden

Die wissenschaftlichen Erkenntnisse und entwickelten Methoden der vorliegenden Arbeit tragen zur Klärung der in der Einleitung (Kapitel 1) genannten, offenen Fragen der Initiatoren BEITZ et al. [B3] und des Koordinators MERTENS [M8] des DFG-Schwerpunktprogramms „Innovative rechnerunterstützte Konstruktionsprozesse: Integration von Berechnung und Gestaltung“ [B3] bei. Dies wird im Folgenden unter einer verknüpfenden Beantwortung der Leitfragen der Einleitung zu den entsprechenden Themen diskutiert:

- **Systemneutrale Datenschnittstellen:** Die Bedeutung der systemneutralen Schnittstellen ist zur Überwindung von Programm-, System- und Umgebungsgrenzen sehr hoch. Die vielen verfügbaren Schnittstellenformate unterscheiden sich in deren Leistungsfähigkeit. Derzeit bietet ausschließlich die einzige genormte Schnittstelle STEP an, umfassende Produktdaten zu modellieren und systemneutral zu übertragen. Bislang noch ungelöst ist allerdings die Abbildung von Parametrik und der CAD-Formelemente (Features). In der vorliegenden Arbeit stellt STEP das Medium für den systemneutralen Datenaustausch und die Grundlage für das in das rechnerbasierte Programmsystem integrierte Produktmodell dar.
- **Produktmodell:** Ein Produktmodell ist die Voraussetzung für eine qualitativ hochwertige und objektorientierte Strukturierung und Verwaltung der anfallenden Daten im Entwicklungsprozess. Die Erstellung eines Produktreferenzmodells ist die Basis für die Produktmodellierung. Wird ein systemneutrales Produktmodell angestrebt, bietet STEP hierfür die derzeit leistungsfähigsten Grundkonstrukte an. Das Arbeiten mit der Norm stellt sich aufgrund dessen Umfang als sehr komplex dar. Da die STEP-Norm universell einsetzbar ist, bietet sie viele Möglichkeiten an, darauf aufbauend eigene Produktmodelle zu entwickeln. Mit STEP steht ein sehr leistungsfähiges Werkzeug zur Verfügung, doch ist nicht jeder mögliche Weg zum Produktmodell für die eigene Anwendungsmodellierung sinnvoll. Deshalb stellte die vorliegende Arbeit drei Möglichkeiten vor, eigene Produktmodelle mit STEP zu verwirklichen. Hierbei hat sich herausgestellt, dass eine Projizierung des eigenen Produktmodells auf das STEP-ARM am sinnvollsten ist, da die Kompatibilität erhalten bleibt und dass die Erstellung eines neuen STEP-Anwendungsprotokolls nur dann durchgeführt werden soll, wenn definitiv nachgewiesen ist, dass der derzeitige Leistungsstand von STEP eine Einbindung des eigenen Produktmodells nicht zulässt. Für Zahnradgetriebe ist eine Projizierung gut möglich. Dies wurde an dem Produktreferenzmodell für Getriebe verifiziert, welches sich durch eine 100%ige Kompatibilität auf der STEP-Austauschebene auszeichnet. Das Produktmodell ist statisch an einem bestimmten Zeitpunkt zu betrachten. Kommt die Zeitkomponente hinzu, tritt das Produktmodell in enge Wechselbeziehung mit dem Prozessmodell, was im entwickelten rechnerbasierten Programmsystem über ein Prozessmanagement-System gesteuert wird.

- **Prozessmanagement:** PDM-Systeme sollten in einem ersten Ansatz die architektonische Basis für das rechnerbasierte Programmsystem bilden, da sie zur zeitlichen Handhabung von Produktdaten und Dokumenten bestimmt sind. Dabei hat sich herausgestellt, dass PDM-Systeme entweder aufgrund ihres Ursprungs aus den CAD-Strukturbäumen relativ primitiv sind und deshalb ihre Systemarchitektur nur bedingt für die komplexen Zusammenhänge der rechnerbasierten Produktentwicklung geeignet ist, oder dass PDM-Systeme so komplex zu handhaben sind, dass sie nicht standardmäßig in verschiedenen Unternehmen eingesetzt werden können. Es wären dann umfangreiche und kostenintensive Anpassungen in den Unternehmen notwendig. Dies schränkt die Neutralität von PDM-Systemen ein. Bei der architektonischen Umsetzung des rechnerbasierten Programmsystems wurde deshalb auf eine universelle Anwendung für unterschiedliche Produkte und eine leichte Integration in das Unternehmensumfeld geachtet. Die Transparenz der Prozesse und Entscheidungen, die im rechnerbasierten Programmsystem durch das Prozessmanagement-System aufgezeichnet werden, sind gerade in Hinblick auf Dokumentation, Wiederholkonstruktionen und zur Langzeitarchivierung von Vorteil.
- **Konstruktionsleitsystem:** Die Analyse ausgeführter rechnerintegrierter-, rechnerunterstützender- und programmverbindender Systeme hat ergeben, dass diese einerseits bestehende CAD-Programme um Funktionen der Berechnung/Simulation erweitern oder andererseits eine Kopplung über Schnittstellen zwischen CAD und Berechnung/Simulation vornehmen. Sind bestehende Programme und Systeme mit geeigneten Schnittstellen zu integrieren und sollen diese eigenständig bleiben, ist der zweite Weg von Vorteil und wurde deswegen auch für das rechnerbasierte Programmsystem ausgewählt. Die Integration, Ansteuerung und Verwaltung verschiedener Programme und Systeme über das rechnerbasierte Programmsystem bedarf zudem ergonomischer Methoden innerhalb der einheitlichen grafischen Benutzungsoberfläche, um für den Anwender alle Daten gut aufzubereiten. Innerhalb der Oberfläche wurde auf ein Karteikartensystem und auf zwei- und dreidimensionale Produktstruktur- und Geometrievierwer zurückgegriffen, die eine möglichst große Transparenz auf die Daten im STEP-Produktmodell und die Dateien und Prozessinformationen in der Datenbank für die Prozesshistorie ermöglichen.
- **Frühe Phasen:** Für die Grobauslegung von Zahnradgetrieben wurde ein interaktives Getriebe-Auslegungsprogramm entwickelt, welches dem Konstrukteur in den frühen Phasen eine ganzheitliche Formung und Erfassung des Getriebes ermöglicht. Da zu diesem Zeitpunkt nur bedingt Daten zur Verfügung stehen, verwendet das Getriebe-Auslegungsprogramm Erfahrungswerte ausgeführter Getriebe und deckt deshalb zunächst nur den Bereich der Standardgetriebe ab. Mit der Eingabe weiterer ausgeführter Getriebe wächst der Wissensstand des Getriebe-Auslegungsprogramms für wiederkehrende Getriebe und ermöglicht so dessen Einsatz in verschiedenen Unternehmen zur Auslegung unterschiedlichster Getriebe. Die Methode der Ergebnisberechnung unterscheidet sich von den bestehenden FVA-Programmen, da ein interaktives Ausgabefenster eine stetige Aktualisierung jedes einzelnen Ausgabewertes bei einer Dateneingabe vornimmt. Durch diese interaktive Vorgehensweise wird eine schnelle und effektive Getriebeauslegung ermöglicht.

9.2 Positionierung des rechnerbasierten Programmsystems in das Umfeld der Gestaltung und Berechnung

Bild 9-1 zeigt alle im Umfeld von Berechnung und Gestaltung eingesetzten Programme und Systeme, deren Zusammenhänge und darauf anwendbare Methoden sowie die Ausprägung des Produkts in den verschiedenen Stadien. Die in der vorliegenden Arbeit behandelten Komponenten und Themen sind entsprechend gekennzeichnet. Die anderen Bereiche, Methoden und denkbare Einsatzgebiete werden im Folgenden kurz vorgestellt, Anschlussmöglichkeiten dargelegt und vor Allem eine Abgrenzung vorgenommen. Dadurch wird die Positionierung des rechnerbasierten Programmsystems deutlich und begründet.

- **Beschränkung auf bestimmte Anwendungssoftware:** In der vorliegenden Arbeit wurden in der Praxis bewährte Programm- und Systemkopplungen nicht neu aufgelegt. Aus diesem Grund wurde auf die Kopplung zu Werkzeugmaschinen [A3] oder deren Simulation (CAM) [M1][P5] und auf die Verknüpfung zu Finite-Element-Programmen (FEM) [L10][R6][W5][N1][W7] beziehungsweise Mehrkörper-Simulationssystemen (MKS) nicht näher eingegangen. Die Verwendung dieser Systeme entspricht dem Stand der Technik [M8] und schließt sich direkt an die Modellierung im 3D-CAD System an. Sind die Systeme nicht direkt in das CAD-System integriert, so erfolgt der Datenaustausch über Datenschnittstellen [B10]. Das STEP-Produktmodell eignet sich demnach auch, direkt in CAM-, FEM- oder MKS-Systeme eingelesen werden zu können, sofern die Programme über geeignete, systemneutrale Schnittstellen verfügen [H6]. Bei diesem Vorgang wird nur die Geometrie übertragen, die für FEM- und MKS-Berechnungen und Fertigungssimulationen ausreichend ist. FEM-Definitionen sind auch in den STEP-Basismodellen enthalten, werden aber noch in keinem STEP-Anwendungsprotokoll verwendet. Der über die Herstellsimulation hinausgehende Anschluss an virtuelle Fabriken [B12] und Montageanlagen [F2][J3][L1][R2] wurde nicht behandelt und ist der entsprechenden Literatur zu entnehmen.
- **Systemunabhängigkeit im CAD-Bereich:** Die geometrische Datenübertragung von, beziehungsweise zum CAD-System in einer systemneutralen Schnittstelle kann nur realisiert werden, wenn deren Leistungsfähigkeit in die CAD-Systeme projiziert wird und die CAD-Funktionalitäten nicht über diesen Leistungsbereich hinaus verwendet werden. Daraus lassen sich die folgenden Einschränkungen bei der CAD-Modellierung definieren: Es gibt keine systemneutrale Schnittstelle, mit der parametrische Beziehungen [K17][V5] in der Bauteilgeometrie übertragen werden können. Bei STEP beginnen dazu die Arbeiten. Ebenso lassen sich keine Features [H1][H2][H17] systemneutral übertragen. Eine VDI-Richtlinie [V11][W3] über Feature-Technologie [K12][V8] und der Ansatz zur Erstellung von standardisierten Feature-Bibliotheken [K3][M7] finden noch keine verbreitete Anwendung.

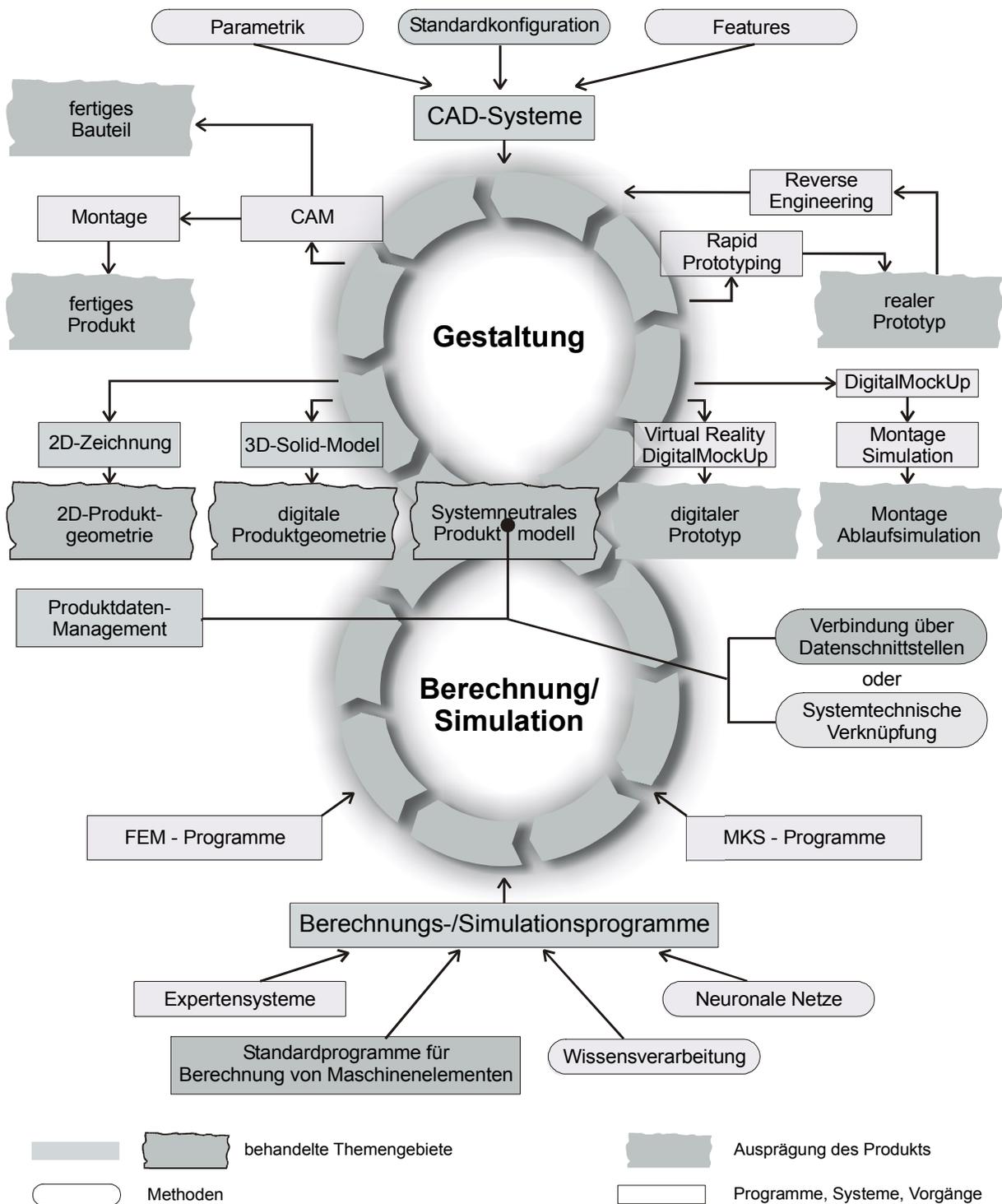


Bild 9-1: Programme und Systeme zur Gestaltung und Berechnung von Produkten im Umfeld der rechnerbasierten Produktentwicklung und vom rechnerbasierten Programmsystem abgedeckte Bereiche

- **Keine automatisierte Entscheidungsfindung, kein Wissensmanagement und Expertensystem, keine Anwendung neuronaler Netze:** Das rechnerbasierte Programmsystem stellt kein automatisiertes System dar, welches dem Konstrukteur Entscheidungen abnimmt oder gar Wissen speichert. Im Gegensatz zu dem FVA-Programmsystem [W15] enthält es keine auf Entscheidungswissen beruhenden Komponenten, welche Vorgänge aufgrund einer automatisierten Rückschlusskette auslösen. RUDZEWSKI [R8] beschreibt ein hierfür notwendiges Bedingungs- und Ereignisnetz zur Bestimmung der Berechnungsreihenfolge der Einzelprogramme für die Getriebeberechnung. Einen Ansatz für das wissensbasierte Konstruieren von Getrieben beschreibt ROSER [R7]. Das erstellte rechnerbasierte Programmsystem enthält keine wissensbasierten Komponenten wie [L9][S11][T7] und ist nicht zur Ablage von Expertenwissen hin ausgelegt. Die Möglichkeit mittels neuronaler Netze in den frühen Phasen Überschlagsrechnungen durchzuführen wurde ebenfalls nicht berücksichtigt. KRAUSE [K10] beschreibt damit die Auslegung einer Stirnradstufe, ZENNER&MARQUARDT [Z1] gehen auf die Bestimmung von Bauteilwöhlerlinien mit neuronalen Netzen ein.
- **Integration von Gestaltung und Berechnung nicht auf Basis einer systemtechnischen Verknüpfung:** GAUSEMEIER et al. [G3] unterteilen die Verknüpfung von CAE-Werkzeugen in eine enge und indirekte und in eine lose Kopplung der Anwendungen. Die lose Kopplung wurde nicht berücksichtigt, da sie sich auf der System- und nicht der Datenaustauschebene abspielt. Um den Daten- bzw. Informationsaustausch zu realisieren, gibt es nach PETERS [P1] die Möglichkeit der Datenkopplung, d.h. Programme tauschen Datensätze untereinander aus, und der Nachrichtenkopplung, d.h. Programme unterrichten sich, um Aktionen auszulösen. Die vorliegende Arbeit stützt sich ausschließlich auf die Kopplung über Schnittstellendateien. Mechanismen zur Nachrichtenkopplung [S24] (Remote-Procedure-Calls RPC, Common Object Request Broker Architecture CORBA [P1][P5], Object Linking&Embedding OLE [S9], Distributed Common Object Model DCOM [G3]) wurden nicht behandelt.
- **Schwerpunkt auf bestimmte Konstruktionsphasen:** Die Beherrschung der konstruktionsphasen-übergreifenden rechnerbasierten Produktentwicklung in Bezug auf einen hochwertig qualitativen Datenaustausch und eine Datenspeicherung in einem Produktmodell steht im Vordergrund der vorliegenden Arbeit. Es wurde beispielhaft aufgezeigt, wie insbesondere die Verknüpfung zwischen Gestaltung und Berechnung im zeitlich durchgängigen Entwicklungsprozess realisiert werden kann. Hierbei wurden nur die Konstruktionsphasen Entwurf, Gestaltung und Nachrechnung/Simulation mit einbezogen, da in diesen vorrangig komplexe rechnerbasierte Vorgänge stattfinden. Ziel war es nicht, allen Phasen aus gängigen Vorgehensweisen zur Produktentwicklung [E1][V13] rechnerbasierte Hilfsmittel anzubieten. In den frühesten Phasen der Produktentwicklung wird auf die Veröffentlichungen [F4][L6][M5][S7][S15] verwiesen. Auf der zeitlichen Schiene der Produktentwicklung beginnt die vorliegende Arbeit bei der Grobauslegung von Getrieben. Hierzu wurde ein eigens entwickeltes Getriebe-Auslegungsprogramm beschrieben. In den späten Phasen der Produktentwicklung endet das rechnerbasierte Programmsystem neben einem universell verwendbaren STEP-Produktmodell mit der Ausgabe von Fertigungszeichnungen. Die Montage [C6][F2][L1], Fertigung [M9][A3] und deren Simulation als Digital MockUp [G1][P3] oder als Rechnersimulation für die Herstellung von Zahnrädern [H7] wurde weiter nicht behandelt.

- **Programmoberfläche nicht im Vordergrund:** Die Akzeptanz von Anwendersoftware wird größtenteils durch die Programmoberfläche bestimmt, welche die Schnittstelle zwischen Mensch und Maschine darstellt. Zur Erstellung von Programmoberflächen existieren eine Vielzahl von rechnerunterstützten und optischen Hilfsmitteln, die in beliebiger Art kombiniert werden können. Ein geeignetes Oberflächenkonzept bedeutet eine ausgiebige Projektierungsphase, stellt aber keine Innovation mehr dar. Es wurde deshalb auf eine detailgetreue Vorgehensweise zur Auswahl des Oberflächenprogrammierungssystems, der softwaretechnischen Umsetzung und auf den Weg zur Umsetzung der Anwender- und Systemanforderungen in eine ergonomische Oberfläche nicht weiter eingegangen.
- **Wegfall von Benutzer- und Projektmanagement-Funktionen:** Auf folgende typische Funktionen [V12] von PDM-Systemen wurde nicht eingegangen, da deren Verständnis und Umsetzung den Stand der Technik repräsentieren und keine Innovation mehr darstellen:
 - **Verzicht auf eine Benutzerverwaltung:** Die Funktionalität des Programmsystems wird nur an einem einzigen Benutzer aufgezeigt. Es entfällt die Umsetzung einer Benutzerverwaltung, von Zugriffsrechten der jeweiligen Benutzer und das daran gekoppelte Bearbeitungs- und Freigabemanagement.
 - **Keine Integration eines zeitlichen Projektmanagements:** Bei der Projektierung dienen Projektmanagementsysteme dazu, vor der Produktentwicklung den zeitlichen Ablauf anzugeben, Termine festzulegen, Zuständigkeiten zu definieren und Kapazitäten zu verteilen.
 - **Firmenübergreifende und verteilte Produktentwicklung:** Alle Funktionen des Programmsystems sind auf den Betrieb in einem einzigen Unternehmen ausgerichtet. Bis auf die Schnittstellen verfügt das Programmsystem über keine Methoden, die eine firmenübergreifende Kommunikation von Rechnersystemen zulässt.
 - **Keine Integration von Freigabe-, und Genehmigungsabläufen,** keine Berücksichtigung von **Reifegraden.**

10 Zusammenfassung und Ausblick

10.1 Zusammenfassung

Im Rahmen dieser Arbeit wurde ein Modell für die rechnerbasierte Produktentwicklung vorgestellt. Hierzu wurde ein rechnerbasiertes Programmsystem für allgemeine Produkte erstellt und seine Anwendbarkeit an einer integrierten Produktentwicklung am Beispiel der Zahnradgetriebe nachgewiesen.

Ziele der Untersuchungen, die zu dem rechnerbasierten Programmsystem führten, war die systemneutrale, bidirektionale und Konstruktionsphasen-übergreifende Verknüpfung von Gestaltung und Berechnung und eine zentrale Dokumentation, Speicherung und Verwaltung der Produkt-, Prozessdaten und digitalen Dokumente. Letztendlich sollte die rechnerbasierte Produktentwicklung einen Beitrag für einen innovativen Produktentwicklungsprozess liefern, um die Entwicklungszeit und damit die Kosten zu senken und gleichzeitig die Qualität der Prozesse und des Produkts zu erhöhen.

Die Grundbestandteile des entwickelten rechnerbasierten Programmsystems sind Basis- und Anwendungsfunktionen zur Steuerung des Systems und für die Visualisierung, ein Prozessmanagement-System zur kontrollierten Realisierung der Abläufe, eine daran angeschlossene Datenbank für die Prozesshistorie zur Archivierung der Daten, Dateien und Prozesse, eine externe Datenschnittstelle zur Kommunikation mit separaten externen Anwendungen sowie eine grafische Benutzungsoberfläche. In das rechnerbasierte Programmsystem sind externe Anwendungen aus allen Entwicklungsphasen eines Produkts über deren Schnittstellen integrierbar. Die Grundlage zur Speicherung und Verwaltung der Produktdaten stellt ein eigenständiges systemneutrales Produktmodell dar.

Die Entwicklung des Produktmodells für Getriebe wurde zur Erreichung einer Systemneutralität in Anlehnung an das STEP AP214 durchgeführt. Dabei hat sich gezeigt, dass mehrere Wege für die Erstellung des Produktmodells für Getriebe und seiner Abbildung auf das AP214 möglich sind. Die ausgewählte und durchgeführte Projizierung des Produktmodells für Getriebe auf das STEP AP214 sichert eine Kompatibilität mit dem derzeit leistungsfähigsten Standard für systemneutrale Produktmodellierung, ohne die STEP-Norm zu erweitern oder zu verändern.

Die Besonderheiten des rechnerbasierten Programmsystems zeigen sich am besten an dessen beispielhafter Anwendung in der Getriebeentwicklung. Es wurde gezeigt, wie der gesamte Entwicklungsprozess eines Getriebes mit dem rechnerbasierten Programmsystem schnell, durchgängig und mit einer hohen Qualität im virtuellen Produkt und den Prozessen durchgeführt werden kann. Für die frühen Phasen wurde ein Getriebe-Auslegungsprogramm entwickelt, welches das Wissen ausgeführter Getriebe dazu verwendet, Getriebewerte abzuschätzen und so eine frühzeitige überschlägige Berechnung zu ermöglichen. Dadurch und aufgrund einer interaktiven Arbeitsweise wird ein schneller Getriebeentwurf möglich. Das weitere Vorgehen bei der Getriebeentwicklung zeichnete sich durch einen Kreislauf zwischen Gestaltung und Berechnung aus. Aufgrund der bidirektionalen Verknüpfung der jeweils eingesetzten Programme und Systeme, der sorgfältigen Dokumentation des Vorgehens und der

Archivierung aller Daten und Dateien ergibt sich eine hohe Qualität des Prozesses, Datenredundanzen werden vermieden und der Anwender wird von routinemäßigen Datenaustauschvorgängen entlastet. Eine jeweils zentrale Speicherung der Produktdaten in dem STEP-Produktmodell für Getriebe und der Dateien und Prozessdaten in der Datenbank für Prozesshistorie sichert eine größtmögliche Transparenz der Produkt- und Prozessinformationen, einen wiederholten Einstieg in beliebige Projektstadien für die Entwicklung von Varianten und erlaubt eine Langzeitarchivierung. Zwei- und dreidimensionale Visualisierungstechniken gestatten in allen Projektstadien eine Übersicht über die Produktstruktur, die Geometrieausprägung mit Bewegungssimulation, den Zusammenhang zwischen Dokumenten, Bauteilen und den jeweils erzielten Ergebnissen. Wie das Produktmodell und die angeschlossenen Programme und Systeme lässt sich auch das rechnerbasierte Programmsystem in verschiedenen informationstechnischen Umgebungen betreiben und erlaubt aufgrund definierter Schnittstellen eine größtmögliche Offenheit mit seiner Umgebung.

Die vorliegende Arbeit deckt den von BEITZ et al. [B3] und MERTENS [M8] in der Einleitung (Kapitel 1) anhand der offenen Fragestellungen des DFG-Schwerpunktprogramms definierten notwendigen Handlungsbedarf ab und stellt mit der rechnerbasierten Produktentwicklung neue Methoden zum durchgängigen Rechnereinsatz innerhalb der Konstruktion vor.

10.2 Ausblick

Die vorliegenden Ergebnisse, die mit der Erstellung des rechnerbasierten Programmsystems erzielt worden sind, bieten eine gute Grundlage für rechnerbasierte Entwicklungsprozesse.

Das systemneutrale Produktmodell und die Datenbank für Prozesshistorie mit dem daran hängenden Prozessmanagement-System stellen die Basis zur Abbildung beliebiger Produkt- und Prozessdaten, Dateien und Dokumente dar. Weiteres Ziel kann es deshalb sein, alle produkt- und prozessverarbeitenden Systeme, die über den bisherigen Funktionsumfang des rechnerbasierten Programmsystems hinaus gehen, mit ihren entstehenden Daten und Dateien zu integrieren. Dies können beispielsweise PDM- und ERP-Systeme, CAM-Anwendungen, MKS- und FEM-Programme und -Systeme sein.

Eine Erweiterung des rechnerbasierten Programmsystems in verschiedene Richtungen ist zudem denkbar. Zum Beispiel könnte ein systemneutrales Norm-, Gleich- und Wiederholteilesystem entwickelt werden, welches auf Basis des systemneutralen Produktmodells CAD-System-unabhängige Bauteile zur Verfügung stellt, sich um unternehmensspezifische Bauteile erweitern lässt und eine Standarddatenbank verwendeter Bauteile im Unternehmen darstellen könnte. Eine andere Anwendung könnte die systemübergreifende Kommunikation zwischen Zulieferer und OEM über das rechnerbasierte Programmsystem sein. Für diesen Fall müsste das Prozessmanagement-System um Internet-basierte Austauschmechanismen für die Daten und Dokumente erweitert werden, ein separater Austausch des STEP-Produktmodells und somit der gesamten Produktdaten ist auch jetzt schon möglich. Als weitere Option könnte das rechnerbasierte Programmsystem um typische PDM-Funktionen wie Benutzermanagement, Reifegrad, Freigabestatus und Projektmanagement ergänzt werden.

Die im STEP-Produktmodell geforderte Systemneutralität während der Datenübertragung zwischen Gestaltung und Berechnung ergibt sich als kleinste gemeinsame Teilmenge der beteiligten Programme und Systeme und der Leistungsfähigkeit von STEP. Dies schränkt die Möglichkeiten ein. Wünschenswert wäre deshalb eine genormte Weiterentwicklung von STEP zur systemneutralen Abbildung von CAD-Formelementen („Features“) und einer Parametrik. Voraussetzung hierfür müsste sein, dass sich die CAD-Systeme an diesen Stellen annähern und gleichermaßen Features und Parametrik verwalten und somit auch systemneutral weitergeben könnten. Dies könnte Gegenstand weiterer Forschungsarbeiten sein.

Das in der Einleitung (Kapitel 1) zitierte Wort von G. Niemann „Ein Mann der konstruieren will, der schaue erst mal und denke“ [N3] wird auch und gerade in der rechnerbasierten Produktentwicklung von steigender Bedeutung sein, da einerseits immer mehr Routinetätigkeiten vom Rechner übernommen werden, dadurch Vorgänge schneller ablaufen und dem Anwender mehr Zeit für intelligentere Aufgaben lassen und andererseits die Rechnertechnologie im Bereich der Berechnung und Simulation den Anwender in komplexen Vorgängen unterstützt, die dennoch seine größte Aufmerksamkeit zur Steuerung der Systeme und Bewertung der Ergebnisse fordern, ihn also auch zum intensiven „Denken“ anregen.

11 Literatur

- [A1] ABELN, O.: CAD-Referenzmodell zur arbeitsgerechten Gestaltung zukünftiger computergestützter Konstruktionsarbeit. Stuttgart: Teubner 1995
- [A2] ABRAMOVICI, M.: EDM/PDM-Einführungsstrategien – Erfahrungen und Perspektiven. In: Beschleunigung der Produktentwicklung durch EDM/PDM- und Feature-Technologie. Düsseldorf: VDI-Verlag 1999, S. 209-226. (VDI-Berichte 1497).
- [A3] ADOLF, H.; DAMMER, M.; JUNGNICKEL, G.; WEIßENBERGER, M.: Integration von Gestaltung und Berechnung bei der Produktionsmaschinen-Entwicklung. Konstruktion (1999) 10, S. 19-25.
- [A4] AMBROSY, S.: Methoden und Werkzeuge für die integrierte Produktentwicklung. Aachen: Shaker 1997. (Konstruktionstechnik München, Band 26). Zugl. München: TU, Diss. 1996
- [A5] AMFT, M.; DYLA, A.; HÖHN, B.-R.; LINDEMANN, U.: Bidirektionale Integration von Berechnung und Gestaltung. Konstruktion (2001) 10, S. 77-81.
- [A6] ANDERL, R.: CAD-Schnittstellen. München: Hanser 1992
- [A7] ANDERL, R.; BEUTHEL, R.: Moderner CAX-Datenaustausch in mittelständischen Unternehmen – eine Situationsanalyse anhand deutscher Schmieden. In: Produkte entwickeln im realen Umfeld. Düsseldorf: VDI-Verlag 2000, S. 285-297. (VDI-Berichte 1569).
- [A8] ANDERL, R.; BEUTHEL, R.; GOGOLL, A.; AWISZUS, B.: Die Schmiedeprozesskette – Analyse ihrer Abbildungsmöglichkeiten in der ISO 10303. Produktdaten Journal (2000) 2, S. 33-37.
- [A9] ANDERL, R.; LINDEMANN, U.; THOMSON, B.; GAUL, H.-D.; GIERHARDT, H.; OTT, T.: Investigation of distributed Product Design and Development Processes. In: Lindemann, U.; Birkhofer, H.; Meerkamm, H.; Vajna, S. (Eds.): Proc. Of the 12th Intern. Conference on Engineering Design 1999, Vol. 2, München, 24.-26.08.1999. S. 1675-1678. (Schriftenreihe WDK 26)
- [A10] ANDERL, R.; PHILIPP, M.: Konstruktionswissenschaft und Produktdatentechnologie. Konstruktion 51 (1999) 3, S. 20-24.
- [A11] ANDERL, R.; PÜTTER, C.: COMET – a Modelling Environment for co-Operative Product Information Modelling. In: Lindemann, U.; Birkhofer, H.; Meerkamm, H.; Vajna, S. (Eds.): Proc. of the 12th Intern. Conference on Engineering Design 1999, Vol. 2, München, 24.-26.08.1999. S. 649-654. (Schriftenreihe WDK 26)
- [A12] ANDERL, R.; TRIPPNER, D.: Standard for the Exchange of Product Model Data. Stuttgart: Teubner 2000
- [A13] ANDERL, R.; WASMER, A.: Chancen und Risiken der Produktdatentechnologie. In: CAD'94 – Produktmodellierung und Prozessmodellierung als Grundlage neuer CAD-Systeme. Paderborn: Hanser 1994, S. 57-72. (Heinz Nixdorf Institut 1994).

-
- [A14] ANNAST, R.: FVA-Forschungsvorhaben Nr. 49A: FVA Kegelradnormprogramm KNplus. Benutzeranleitung, Programmdokumentation. FVA-Heft 233 (2001) – Firmenschrift
- [B1] BALD, D.; RÜHRNSCHOPF, G.: Configuration Management mit PDM/EDM in den Prozessketten der Produktentwicklung. In: Engineering Data Management (EDM). München: Herbert Utz 1998, S. 97-120 (Seminarberichte IWB).
- [B2] BEERMANN, S.; KISSLING, U.: Entwicklung der Berechnungsprogramme für den Maschinenbau. Antriebstechnik 40 (2001) 7, S. 47-51.
- [B3] BEITZ, W.; KOLLMANN, F.; LECHNER, G.: Antrag auf Einrichtung eines neuen DFG Schwerpunktprogramms SPP 732, "Innovative rechnerunterstützte Konstruktionsprozesse: Integration von Gestaltung und Berechnung". Fachreferat II D 1. Bonn: DFG, 1995.
- [B4] BEITZ, W.; KÜTTNER, K.-H.: Dubbel – Taschenbuch für den Maschinenbau. 18. Auflage. Berlin: Springer 1995
- [B5] BENDER, K.: Offene Systemarchitektur zur Unterstützung integrierter Engineering-Prozesse. In: Lindemann, U. (Hrsg.): Arbeits- und Ergebnisbericht Sonderforschungsbereich 336 – Montageautomatisierung durch Integration von Konstruktion und Planung. München: 2000, S. 77-96
- [B6] BENDER, K.: Einführungsstrategien für Engineering-Werkzeuge zur integrierten Produktentwicklung. In: Lindemann, U. (Hrsg.): Arbeits- und Ergebnisbericht Transferbereich 2 – Montageautomatisierung durch Integration von Konstruktion und Planung. München: 2000, S. 79-128
- [B7] BENDER, K.; HEINZL, J.; LINDEMANN, U.; MILBERG, J.; PFEIFFER, F.; REICHWALD, R.; REINHART, G.: Kolloquiumsband Transferbereich 2 – Montageautomatisierung durch Integration von Konstruktion und Planung. München: 1998
- [B8] BENDER, K.; KARCHER, F.; FISCHER, F.: KEIM – Kontinuierliches Engineering-Informationssystemmanagement im Product Life Cycle. Industrie & Management (2001) 6, S. 58-62.
- [B9] BOPP, R.; SCHAEDE, J.; SCHWITAL, P.: Prozessorientiertes 3D-Produktdatenmodell (POP) zur Entwicklung virtueller Druckmaschinen. In: Produkte entwickeln im realen Umfeld. Düsseldorf: VDI-Verlag 2000, S. 67-85. (VDI-Berichte 1569).
- [B10] BOSHOFF, F.: Integration von FEM-Berechnungen in den CAD-gestützten Konstruktionsprozeß durch bidirektionalen automatischen Geometrieaustausch. Aachen: Shaker 1997 (Berichte aus der Produktionstechnik Bd. 97, 19). Zugl. Aachen: RWTH, Diss. 1997
- [B11] BÖTTNER, D.; KRÜGER, K.; DAUM, B.; KRETZSCHMAR, O.: STEP-Einsatz in kleinen und mittleren Unternehmen. CAD-CAM Report 18 (1999) 9, S. 86-93.
- [B12] BRANDNER, S.: Integriertes Produktdaten- und Prozessmanagement in virtuellen Fabriken. München: TU, Diss. 1999

- [C1] CHONG, T.; LEE, J.; AHN, J.; LYU, S.: Representing Design Information of Gear Drives in XML. In: Proc. of the JSME Intern. Conference on Motion and Power Transmissions, Fukuoka, Japan, 15.-17.11.2001, S. 662-667
- [C2] CHONG, T.; PARK, S.: A Study on Integration of Gear Design Data using XML. In: Proc. of the JSME Intern. Conference on Motion and Power Transmissions, Fukuoka, Japan, 15.-17.11.2001, S. 657-661
- [C3] CHRIST, M.: Rechnersoftware für die integrierte Gestaltung und Berechnung von Planetengetrieben. (Schriftenreihe des Institutes für Konstruktionstechnik, Nr. 99.1). Zugl. Bochum: Ruhr-Univ., Diss. 1999
- [C4] COLLIN, H.: Management von Produkt-Informationen in kleinen und mittelständischen Unternehmen. (Produktentwicklung, Band 43). Zugl. München: TU, Diss. 2001
- [C5] COOPER, C.: Issues of Gear Design Using 3D Solid Modeling Systems. Gear Technology (1999) 1/2, S. 38-43.
- [C6] CUIPER, R.: Durchgängige rechnergestützte Planung und Steuerung von automatisierten Montagevorgängen. München: TU, Diss. 2000
- [D1] DIETZ, P.; TAWIL, M.; TRENKE, D.: 3D-Produktmodellierung in der Entwicklungskette – Aufwand und Nutzen. In: Effiziente 3D-Produktmodellierung – Fortschritte und Fallstricke. Düsseldorf: VDI-Verlag 2001, S. 113-130. (VDI-Berichte 1614)
- [D2] DOBRE, G.; MILOIU, G.; MIRICA, R.-F.: The product development based on experience – application of flexible development on gearboxes. In: Culley, S.; Duffy, A.; McMahon, C.; Wallace, K. (Eds.): Proc. Of the 13th Intern. Conference on Engineering Design 2001, Vol. “Design Applications in Industry and Education”, Glasgow, 21.-23.08.2001. Glasgow: I Mech E 2001, S. 163-170. (Schriftenreihe WDK 28)
- [D3] DOLESCHEL, A.: FVA-Forschungsvorhaben Nr. 69/3: FVA-Programm WTplus. Benutzeranleitung, Programmdokumentation. FVA-Heft 625 (2001) – Firmenschrift
- [D4] DONGES, C.; KRASSEL, M.; ANDERL, R.: MechaSTEP – STEP-Datenmodelle zur Abbildung mechatronischer Systeme. Produktdaten Journal (1999) 1, S. 30-34.
- [D5] DYLA, A.; HÖHN, B.-R.: Integrated development of gear units – product- and workflow modelling. In: Culley, S.; Duffy, A.; McMahon, C.; Wallace, K. (Eds.): Proc. Of the 13th Intern. Conference on Engineering Design 2001, Vol. “Design Research – Theories, Methodologies, and Product Modelling”, Glasgow, 21.-23.08.2001. Glasgow: I Mech E 2001, S. 709-716. (Schriftenreihe WDK 28)
- [D6] DYLA, A.; HÖHN, B.-R.; ANDERL, R.; PFEIFER, U.: Neutrales Format für die Produktmodellierung von Getrieben. Produktdaten Journal (2001) 1, S. 25-29.
- [D7] DYLA, A.; HÖHN, B.-R.; STEINGRÖVER, K.: Integrierte Entwicklung von Getrieben: Praxisanwendung und Visionen. In: Produkte entwickeln im realen Umfeld. Düsseldorf: VDI-Verlag 2000, S.115-135. (VDI-Berichte 1569).

- [E1] EHRENSPIEL, K.: Integrierte Produktentwicklung: Methoden für Prozessorganisation, Produktentwicklung und Konstruktion. München: Hanser 1995.
- [E2] EHRENSPIEL, K.; AMBROSY, S.; AßMANN, G.: Integrierter Konstruktionsarbeitsplatz. ZWF 90 (1995) 9, S. 410-413.
- [E3] EHRENSPIEL, K.; KIEWERT, A.; LINDEMANN, U.: Kostengünstig Entwickeln und Konstruieren. 3. Aufl. Berlin: Springer 2000
- [E4] EIGNER, M.: EDM/PDM und verteilte Kommunikation. CAD-CAM Report 18 (1999) 1, S. 42-47.
- [E5] EIGNER, M.; ZAGEL, M.: PDM-Technologie als Schlüssel zur virtuellen Produktentstehung. In: Beschleunigung der Produktentwicklung durch EDM/PDM- und Feature-Technologie. Düsseldorf: VDI-Verlag 1999, S. 103-119. (VDI-Berichte 1497).
- [E6] EVERSHEIM, W.; BAUMANN, M.; MARCZINSKI, G.; LEBER, M.: STEP als Integrationskern für die Produktdatengenerierung. VDI-Z 135 (1993) 7, S. 63-66.
- [F1] FELDHUSEN, J.: Systemkonzept für die durchgängige und flexible Rechnerunterstützung des Konstruktionsprozesses. Berlin: KT 1989. (Konstruktionstechnik 16). Zugl. Berlin: TUB, Diss. 1989
- [F2] FELDMANN, C.: Eine Methode für die integrierte rechnergestützte Montageplanung. München: TU, Diss. 1996
- [F3] FISCHER, F.; GREINDL, A.; PUFFALDT, J.; RECH, N.; VAJNA, S.: Engineering – Workflow. CAD-CAM Report 19 (2000) 9, S. 46-48.
- [F4] FRANKENBERGER, E.: Computer-supported systematic design and knowledge management in the early design phase. In: Culley, S.; Duffy, A.; McMahon, C.; Wallace, K. (Eds.): Proc. Of the 13th Intern. Conference on Engineering Design 2001, Vol. “Design Management – Process and Information Issues”, Glasgow, 21.-23.08.2001. Glasgow: I Mech E 2001, S. 115-122. (Schriftenreihe WDK 28)
- [G1] GAIGL, H.: Automatische Design Verifikation als Grundlage des Digital Mock-Up. In: Virtuelle Produktion – Prozeß- und Produktsimulation. München: Herbert Utz 1999, S. 65-70 (Seminarberichte IWB).
- [G2] GAUSEMEIER, J.: Kurzberichte über die Untersuchung. Neue Wege zur Produktentwicklung. 2. Aufl. Paderborn: (Berliner Kreis, Produktion 2000) 1998
- [G3] GAUSEMEIER, J.; GRASMANN, M.; KESPOHL, H.: Verfahren zur Integration von Gestaltungs- und Berechnungssystemen. In: Verkürzte Entwicklungsprozesse durch Integration von Gestaltung und Berechnung. Düsseldorf: VDI-Verlag 1999, S. 71-87. (VDI-Berichte 1487).
- [G4] GAUSEMEIER, J.; GRASMANN, M.; PUSCH, R.; SCHNEIDER, W.: Integrierte Produkt- und Prozessmodellierung – ein Ansatz zur schnelleren Einführung von Engineering Daten Managementsystemen. In: Neue Generation von CAD/CAM-Systemen. Düsseldorf: VDI-Verlag 1997, S. 263-276. (VDI-Berichte 1357).

- [G5] GAUSEMEIER, J.; LÜCKEL, J.: Entwicklungsumgebungen Mechatronik. Paderborn: Bonifatius 2000. (HNI Verlagsschriftreihe Band 80)
- [G6] GEISER, H.-J.: FVA-Forschungsvorhaben Nr. 338/2: FVA-Programm DZP, Version 4.0. Benutzeranleitung, Programmdokumentation. FVA-Heft 635 (2001) – Firmenschrift
- [G7] GERMER, C.; HANSEN, U.; FRANKE, H.-J.: Die rechnerunterstützte Modellierung dreidimensionaler Mikrokomponenten. In: Effiziente 3D-Produktmodellierung – Fortschritte und Fallstricke. Düsseldorf: VDI-Verlag 2001, S. 349-360. (VDI-Berichte 1614)
- [G8] GRABOWSKI, H.: CAD- und CAx-Software. Konstruktion (1999) 10, S. 3.
- [G9] GRABOWSKI, H.; ANDERL, R.; POLLY, A.: Integriertes Produktmodell. Berlin: Beuth 1993.
- [G10] GRABOWSKI, H.; ANDERL, R.; SCHILLI, B.; SCHMITT, M.: STEP – Entwicklung einer Schnittstelle zum Produktdatenaustausch. VDI-Z 131 (1989) 9, S. 68-76.
- [G11] GRABOWSKI, H.; ERB, J.; POLLY, A.: STEP – Grundlage der Produktdatentechnologie Aufbau und Entwicklungsmethodik. CIM Management 10 (1994) 4, S. 45-51.
- [G12] GRABOWSKI, H.; GEIGER, K.: Neue Wege zur Produktentwicklung. Stuttgart: Raabe 1997.
- [G13] GRAVEL, G.; POMMER, A.: Einheitliches Format beim Austausch von Verzahnungsdaten. Antriebstechnik (1999) 2, S. 39-41.
- [H1] HAASIS, S.: Integrierte CAD-Anwendungen. Berlin: Springer 1995.
- [H2] HAASIS, S.; FRANCK, D.; ROMMEL, B.; WEYRICH, M.: Features – Rückgrat zur Integration der Produkt-, Prozess- und Ressourcengestaltung. Konstruktion (2000) 4, S. 26-27.
- [H3] HAASIS, S.; MISCHKOLIN, F.; ZÜFLE, J.: CAD-gestützte Konstruktion von Getrieben. VDI-Z 136 (1994) 10, S. 58-64.
- [H4] HAHN, A.: Integrationsumgebung für verteilte objektorientierte Ingenieursysteme. Paderborn: Univ.-GH, Diss. 1998
- [H5] HELPENSTEIN, H.: STEP als Verbindung zwischen Systemen. Konstruktion 45 (1993), S. 316-320.
- [H6] HELPENSTEIN, H.: STEP-Schnittstelle für Finite-Elemente-Daten. VDI-Z 138 (1996) 3, S. 42-44.
- [H7] HEMMELMANN, J.; BEULKER, K.; WECK, M.: Potential of Softwaretools for the Simulation of Gear Generation Processes. In: Proc. of the JSME Intern. Conference on Motion and Power Transmissions, Fukuoka, Japan, 15.-17.11.2001, S. 350-354

-
- [H8] HERBST, A.; MALLE, B.: Perspektiven für die Archivierung von CAD-Dateien in einer STEP-Umgebung. In: CAD'94 – Produktmodellierung und Prozessmodellierung als Grundlage neuer CAD-Systeme. Paderborn: Hanser 1994, S. 583-600. (Heinz Nixdorf Institut 1994).
- [H9] HÖHN, B.-R.; STEINGRÖVER, K.; DYLA, A.: Rechnerunterstützte Getriebeentwicklung. Konstruktion 51 (1999) 1/2, S. 17-20.
- [H10] HÖHN, B.-R.; STEINGRÖVER, K.; DYLA, A.: Computer Aided Gear Development. In: Proc. of DMMI'99, 4th Int. Conference, Podcetrtek, Slovenia, 20.-21.09.1999, S. 540-551
- [H11] HÖHN, B.-R.; STEINGRÖVER, K.; DYLA, A.: Rechnerbasierte Produktentwicklung. Konstruktion (2000) 1/2, S. 36-40.
- [H12] HÖHN B.-R.; STEINGRÖVER, K.; DYLA, A.: Systemneutrale CAE-Methoden für die innovative Produktentwicklung. CAD-CAM Report 19 (2000) 3, S. 34-40.
- [H13] HÖHN, B.-R.; STEINGRÖVER, K.; DYLA, A.: Computer aided product development. In: Proc. of International Design Conference – Design 2000, Dubrovnik, 23.-26.5.2000, S. 307-312.
- [H14] HÖHN, B.-R.; STEINGRÖVER, K.; DYLA, A.: Product Model based Workplace for the Integrated Development of Gear Units. In: Proc. of ProSTEP Science Days 2000, Stuttgart, 13-14.9.2000, S. 64-73.
- [H15] HÖHN, B.-R.; STEINGRÖVER, K.; JAROS, M.: Entwicklung einer Datenschnittstelle auf Basis des Produktdatenmodells „STEP“. In: FVA-Forschungsreport 2001 anlässlich FVA-Informationstagung, Würzburg, 27./28.11.2001, S. 406.1-406.12
- [H16] HÖHN, B.-R.; STEINGRÖVER, K.; MATTEN, D.: Das FVA-Stirnradprogramm STplus: Ein neues EDV-Programm zur Berechnung von Evolventen-Stirnrädern. Antriebstechnik 37 (1998) 3, S. 49-53.
- [H17] HORN, T.: Entwicklung eines Verfahrens zur interaktiven Definition und Erkennung benutzerdefinierter Features. Essen: Univ., Diss. 1997
- [I1] ISO-Nom 10303-1: Industrielle Automatisierungssysteme und Integration – Produktdatendarstellung und -austausch – Teil 1: Überblick und grundlegende Prinzipien. Berlin: Beuth 1994.
- [I2] ISO-Norm 10303-11 (Technical Corrigendum 1): Industrielle Automatisierungssysteme und Integration – Produktdatendarstellung und -austausch – Teil 11: Beschreibungsmethoden: Handbuch der Modellierungssprache EXPRESS. Berlin: Beuth 1999.
- [I3] ISO-Norm 10303-201: Industrielle Automatisierungssysteme und Integration – Produktdatendarstellung und -austausch – Teil 201: Anwendungsprotokoll: Explizite Zeichnungsdarstellung. Berlin: Beuth 2001.
- [I4] ISO-Norm 10303-21 (Technical Corrigendum 1): Industrielle Automatisierungssysteme und Integration – Produktdatendarstellung und -austausch – Teil 21: Implementierungsmethoden: Klartext-Kodierung der Austauschstruktur. Berlin: Beuth 1996.

- [I5] ISO-Norm 10303-214: Industrielle Automatisierungssysteme und Integration – Produktdatendarstellung und -austausch – Teil 214: Anwendungsprotokoll: Datenmodelle für die Prozesskette Mechanik in der Automobilindustrie. Berlin: Beuth 2001.
- [I6] ISO-Norm 10303-22: Industrial automation systems and integration – Product data representation and exchange – Part 22: Implementation methods: Standard data access interface. Berlin: Beuth 1998.
- [J1] JÄGER, K.-W.: Schnittstellen bei CAD/CAE-Systemen. Düsseldorf: VDI-Verlag 1991
- [J2] JANDELEIT, M.; ZIRKEL, M.; STROHMEIER, K.; VAJNA, S.: Optimierung der Apparatekonstruktion durch integrierte Rechnerunterstützung. In: Produkte entwickeln im realen Umfeld. Düsseldorf: VDI-Verlag 2000, S. 99-113. (VDI-Berichte 1569).
- [J3] JONAS, C.: Konzept einer durchgängigen rechnergestützten Planung von Montageanlagen. München: Utz, 2000. (Forschungsberichte IWB, Band 145). Zugl. München: TU, Diss. 2000
- [K1] KAISER, A.: STEP, IGES und VDAFS im Vergleich. CAD-CAM Report 17 (1998) 10, S. 34-45.
- [K2] KARCHER, A.; FISCHER, F.; VIERTLBÖCK, M.: EDM/PDM-Systeme als Rückgrat der Integrierten Produktentwicklung – ein modulares Einführungs- und Integrationskonzept. In: Beschleunigung der Produktentwicklung durch EDM/PDM- und Feature-Technologie. Düsseldorf: VDI-Verlag 1999, S. 451-470. (VDI-Berichte 1497).
- [K3] KASAN, R.-D.; MEERKAMM, H.; SANDER, S.: Ein Konzept zur Erstellung von Featurebibliotheken. In: Beschleunigung der Produktentwicklung durch EDM/PDM- und Feature-Technologie. Düsseldorf: VDI-Verlag 1999, S. 393-410. (VDI-Berichte 1497).
- [K4] KERTSCHER, U.; LEIDICH, E.: CAM Gears – Online Calculation and Modelling. In: Lindemann, U.; Birkhofer, H.; Meerkamm, H.; Vajna, S. (Eds.): Proc. Of the 12th Intern. Conference on Engineering Design 1999, Vol. 2, München, 24.-26.08.1999. S. 1809-1812. (Schriftenreihe WDK 26)
- [K5] KIEPE, O.: Integration von CAD/CAM, EDM/PDM und EDI durch ENGDAT. CAD-CAM Report 18 (1999) 10, S. 70-73.
- [K6] KISSLING, U.; ZIEBEIL, P.: Neue Wege für die integrierte Gestaltung und Berechnung. In: Prozessketten für die virtuelle Produktentwicklung in verteilter Umgebung. Düsseldorf: VDI-Verlag 1999, S. 379-394. (VDI-Berichte 1435).
- [K7] KLOSE, J.; GITTER, J.; MEERKAMM, H.; STORATH, E.: Perspektiven der Konstruktionsunterstützung durch das CAD-Referenzmodell. In: CAD'94 – Produktmodellierung und Prozessmodellierung als Grundlage neuer CAD-Systeme. Paderborn: Hanser 1994, S. 333-352. (Heinz Nixdorf Institut 1994).

-
- [K8] KLOSE, J.; RÖMER, S.; STEGER, W.; ZETZSCHE, T.: Methode zur Integration von Berechnungsverfahren in Konstruktionssysteme. In: Neue Generation von CAD/CAM-Systemen. Düsseldorf: VDI-Verlag 1997, S. 143-159. (VDI-Berichte 1357).
- [K9] KOCH, M.; MEERKAMM, H.: Ein Konstruktionsassistenzsystem für komplexe Produkte auf Basis eines hybriden Produktmodells. In: Effiziente 3D-Produktmodellierung – Fortschritte und Fallstricke. Düsseldorf: VDI-Verlag 2001, S. 217-231. (VDI-Berichte 1614)
- [K10] KRAUSE, F.-L.: Auf dem Weg zur virtuellen Produktentwicklung. In: Neue Generation von CAD/CAM-Systemen. Düsseldorf: VDI-Verlag 1997, S. 17-33. (VDI-Berichte 1357).
- [K11] KRAUSE, F.-L.: Erfolgreiche Produktentwicklung – eine Frage der Software. Konstruktion 51 (1999) 3, S. 3
- [K12] KRAUSE, F.-L.; BAUMANN, R.; MARTINI, K.: Normung, Datenaustausch und ausgewählte Anwendungsbeispiele im Bereich Feature-Technologie. Düsseldorf: VDI-Verlag 1999, S. 393-410 (VDI-Berichte 1497)
- [K13] KRAUSE, F.-L.; JANSEN, H.; KAUFMANN, U.; ZIEBEIL, P.: iViP – Perspektiven für die integrierte, virtuelle Produktentstehung. Produktdaten Journal (2000) 1, S. 31-33.
- [K14] KRAUSE, F.-L.; KAHLERT, T.; MENG, L.: Die Gesetzgebung zwingt die Unternehmen zu handeln. EDM-Report (1999) 2, S. 68-73.
- [K15] KREBS, T.: Integration elektromechanischer CA-Anwendungssysteme über einem STEP-Produktmodell. Bamberg: Meisenbach 1997. (Fertigungstechnik – Erlangen, Band 59). Zugl. Erlangen: Univ., Diss. 1995
- [K16] KRETSCHMER, O.: STEP im Automobilbau – Ergebnisse eines Pilotprojekts. Düsseldorf: VDI-Verlag 1996, S. 369-380 (VDI-Berichte 1289)
- [K17] KUNHENN, J.; BUGERT, T.; GÖTZELT, U.; ENDERS, L.; SCHÖN, A.; VAJNA, S.: Parametrik im Produktentstehungsprozeß – Möglichkeiten und Risiken. In: Neue Generation von CAD/CAM-Systemen. Düsseldorf: VDI-Verlag 1997, S. 533-547. (VDI-Berichte 1357).
- [K18] KUNTZ, P.: Rechnerunterstützte Synthese und Variantenkonstruktion von Planetengetrieben. Braunschweig: TU, Diss. 1995.
- [L1] LINDEMANN, U.: Erarbeitung der Elemente und Struktur für einen integrierten Entwicklungsprozess von Produkt und Montageanlage. In: Lindemann, U. (Hrsg.): Arbeits- und Ergebnisbericht Sonderforschungsbereich 336 – Montageautomatisierung durch Integration von Konstruktion und Planung. München: 2000, S. 17-36
- [L2] LINDEMANN, U.: Methodische Unterstützung der integrierten Produktentwicklung. In: Lindemann, U. (Hrsg.): Arbeits- und Ergebnisbericht Transferbereich 2 – Montageautomatisierung durch Integration von Konstruktion und Planung. München: 2000, S. 13-78

- [L3] LINDEMANN, U.: Methoden in der Produktentwicklung. *Konstruktion* (2001) 1/2, S. 3.
- [L4] LINDEMANN, U.; AMBROSY, S.; ABMANN, G.: Generische Produktmodellierung zur Adaption von Anwendungssystemen. In: *Neue Generation von CAD/CAM-Systemen*. Düsseldorf: VDI-Verlag 1996, S. 457-468. (VDI-Berichte 1289).
- [L5] LINDEMANN, U.; AMFT, M.: Rechnergestützte Integration von Gestaltung und Berechnung im Konstruktionsprozess. *Konstruktion* (2000) 10, S. 78-82.
- [L6] LINDEMANN, U.; AMFT, M.; ASSMANN, G.; WULF, J.; BIRKHOFFER, H.; WALLMEIER, S.: Rechnerunterstützung für frühe Phasen der Entwicklung. *F&M* 106 (1998) 3, S. 123-127
- [L7] LÖFFEL, C.; GOLBACH, H.: Auf dem Weg zum virtuellen Produkt durch integrierte Berechnungsmethoden. In: *Verkürzte Entwicklungsprozesse durch Integration von Gestaltung und Berechnung*. Düsseldorf: VDI-Verlag 1999, S. 123-138. (VDI-Berichte 1487)
- [L8] LÖFFEL, C.; GOLBACH, H.: Auf dem Weg zum virtuellen Produkt mit integrierten Berechnungsmethoden. *Konstruktion* (2000) 1/2, S. 24-25.
- [L9] LORENZ, J.: Rechnergestütztes Informationsmanagementsystem zur Optimierung klebtechnischer Konstruktionen. München: TU, Diss. 1998
- [L10] LÖW, H.: Automatisierung der Berechnung und Konstruktion des Stahlbaus von Fahrzeugkranen auf der Basis von FE- und CAD-Methoden. München: TU, Diss. 1993
- [L11] LÜHRSEN, H. : Die Entwicklung von Datenbanken für das Produktmodell der ISO-Norm STEP. Erlangen: Univ., Diss. 1996
- [M1] MACHNER, B.: ProSTEP – Implementierungen für den CAD/CAM-Datenaustausch und die Datenverwaltung. In: *CAD'94 – Produktmodellierung und Prozessmodellierung als Grundlage neuer CAD-Systeme*. Paderborn: Hanser 1994, S. 555-565. (Heinz Nixdorf Institut 1994).
- [M2] MACHNER, B.: STEP bringt auch EDM einen großen Schritt weiter. *EDM-Report* (1996) 1, S. 57-61.
- [M3] MARCA, D.; GOWAN, L.: *SADT – Structured Analysis and Design Technique*. New York: McGraw-Hill 1988.
- [M4] MEERKAMM, H.: Integrierte Produktentwicklung. *Konstruktion* 50 (1998) 9, S. 3
- [M5] MEERKAMM, H.; HEYNEN, C.; SCHWEIGER, W.: Integration von Berechnungen in den frühen Entwurfprozeß. In *Verkürzte Entwicklungsprozesse durch Integration von Gestaltung und Berechnung*. Düsseldorf: VDI-Verlag 1999, S. 103-122. (VDI-Berichte 1487)
- [M6] MEERKAMM, H.; LÖFFEL, C.; SCHWEIGER, W.: Integration von Berechnungswerkzeugen in den Konstruktionsprozeß – ein ganzheitlicher Ansatz auf Basis des Konstruktionssystems mfk. *Konstruktion* 49 (1997), S. 26-32

-
- [M7] MEERKAMM, H.; SANDER, S.; KASAN, R.-D.: Ein Konzept zur Erstellung von Feature-Bibliotheken. *Konstruktion* (2000) 4, S. 22-23.
- [M8] MERTENS, H.: DFG-Schwerpunktprogramm „Innovative, rechnerunterstützte Konstruktionsprozesse – Integration von Gestaltung und Berechnung“. In: *Verkürzte Entwicklungsprozesse durch Integration von Gestaltung und Berechnung*. Düsseldorf: VDI-Verlag 1999, S. 13-30. (VDI-Berichte 1487)
- [M9] MERTINS, K.; SÜSSENGUTH, W.; JOCHEM, R.: *Modellierungsmethoden für rechnerintegrierte Produktionsprozesse*. München: Hanser 1994
- [M10] MOHRMANN, J.; SPECK, H.-J.: Das Produktmodell als Integrationsplattform für Prozessketten. In *CAD'94 – Produktmodellierung und Prozessmodellierung als Grundlage neuer CAD-Systeme*. Paderborn: Hanser 1994, S. 93-110. (Heinz Nixdorf Institut 1994).
- [N1] NAAS, J.; STOFFELS, H.; TROSKA, A.: Integrierte Berechnung zur Optimierung des Verzahnungsgeräusches eines Pkw-Handschaltgetriebes. *ATZ Automobiltechnische Zeitschrift* 103 (2001) 1, S. 16-23.
- [N2] NIEMANN, G.; WINTER, H.: *Getriebe allgemein, Zahnradgetriebe – Grundlagen, Stirnradgetriebe*. 2. Aufl. Berlin: Springer 1989. (Maschinenelemente, Bd. 2)
- [N3] NIEMANN, G.; WINTER, H.; HÖHN, B.-R.: *Konstruktion und Berechnung von Verbindungen, Lagern, Wellen*. 3. Aufl. Berlin: Springer 2001. (Maschinenelemente, Bd. 1)
- [N4] NIKOL, G.: Viewer – wohin geht die Entwicklung? *EDM-Report* (1999) 3, S. 34-39.
- [N5] NIKOL, G.: Viewer sind mehr als Visualisierungswerkzeuge. *EDM-Report* (2000) 4, S. 34-39.
- [P1] PETERS, M.: *Kommunikationssystem rechnerunterstützter Konstruktionswerkzeuge*. Braunschweig: TU, Diss. 1997
- [P2] POLLY, A.: *Methodische Entwicklung und Integration von Produktmodellen*. Aachen: Shaker 1997. (Forschungsberichte aus dem Institut für Rechneranwendung in Planung und Konstruktion der Univ. Karlsruhe, Band 4/1996). Zugl. Karlsruhe: Univ., Diss. 1996
- [P3] von PRAUN, S.: Digital Mock-up. *VDI-Z Special C-Techniken* (1998) 3, S. 42-44.
- [P4] PREDKI, W.; CHRIST, M.: CAD-Anforderungen in der Getriebekonstruktion. *CAD-CAM Report* 17 (1997) 10, S. 122-129.
- [P5] PRITSCHOW, G.; SPUR, G.; WECK, M.: *Schnittstellen im CAD/CAM-Bereich*. München: Hanser 1997
- [R1] RADTKE, M.: *Konzept zur Gestaltung prozeß- und integrationsgerechter Produktmodelle*. Kaiserslautern: ZBT-Abteilung Univ. Kaiserslautern 1995. (Produktionstechnische Berichte, Band 19). Zugl. Kaiserslautern: Univ., Diss. 1995

-
- [R2] REINHART, G.: Methoden und Werkzeuge zur Konfiguration und Steuerung integrierter Entwicklungsprozesse von Produkt und Montageanlage. In: Lindemann, U. (Hrsg.): Arbeits- und Ergebnisbericht Sonderforschungsbereich 336 – Montageautomatisierung durch Integration von Konstruktion und Planung. München: 2000, S. 37-56
- [R3] REN, Z.: Computer aided gear design. Proceedings Vol.1, 4th World Congress on Gearing and Power Transmission, Paris (1999) 3. S. 355-360
- [R4] REPETZKI, S.: Entwicklungsschritte zum ingenieurgerechten Konstruktionsarbeitsplatz. (Berichte aus der Produktionstechnik), Zugl. Aachen: RWTH, Diss. 1997
- [R5] RIEST, K.: Eine Konstruktionsumgebung für integriertes Gestalten und Berechnen am Beispiel von Kalandern. Düsseldorf: VDI-Verlag 1999. (Fortschritts-Berichte der VDI-Zeitschriften Reihe 3, Nr. 599)
- [R6] RIEST, K.; MEWES, D.: CAD-FEM Coupling using a STEP Database – Designing Machines in Polymer Processing Industry. In: Lindemann, U.; Birkhofer, H.; Meerkamm, H.; Vajna, S. (Eds.): Proc. of the 12th Intern. Conference on Engineering Design 1999, Vol. 2, München, 24.-26.08.1999, S. 1789-1792. (Schriftenreihe WDK 26)
- [R7] ROSER, T.: Wissensbasiertes Konstruieren am Beispiel von Getrieben. Stuttgart: - Univ., Diss. 1992
- [R8] RUDZEWSKI, S.: Systemtechnische Verknüpfung eingeführter Getriebeberechnungsprogramme. München: TU, Diss. 1995
- [R9] RULAND, D.: Die Bedeutung von Referenzinformationsmodellen für CIM Realisierungen. *ZwF* (1991), S. 56-59
- [R10] RYBAK, H.: STEP – eine Herausforderung. *CAD-CAM Report* (1997) 10, S. 49-66.
- [S1] SARFERT, J.: Berechnung und Simulation in der Wälzlagertechnik. In: FVA-Forschungsreport 2001 anlässlich FVA-Informationstagung, Würzburg, 27./28.11.2001, S. 11.1-11.14
- [S2] SCHENCK, D.; WILSON, P.: Information Modeling the EXPRESS Way. New York: Oxford Univ. Press 1994
- [S3] SCHINAGL, S.: FVA-Forschungsvorhaben Nr. 30/4: Ritzelkorrektur, Programmbeschreibung (RIKOR G). FVA-Heft 481 (2000) – Firmenschrift
- [S4] SCHMIDT, U.; LOZINSKI, V.: Implementierungsstrategien für integrierte Produktentwicklungsprozesse. EDM-Report (1998) 3, S. 20-23.
- [S5] SCHÖTTNER, J.: Produktdatenmanagement in der Fertigungsindustrie. München: Hanser 1999
- [S6] SCHRÖDER, F.: Ein datenbasiertes Anwendersystem zur Berechnung von Gittermast-Fahrzeugkranen. München: TU, Diss. 2000
- [S7] SCHULZ, A.: Systeme zur Rechnerunterstützung des funktionsorientierten Grobentwurfs. Braunschweig: TU, Diss. 1996

-
- [S8] SENDLER, U.: 3-D CAD. Berlin: Springer (1995)
- [S9] SENDLER, U.: CAD & Office Integration. OLE für Design und Modellierung – Eine neue Technologie für CA-Software. Berlin: Springer (1995)
- [S10] SENDLER, U.: ACIS und STEP. Konstruktionspraxis 8 (1997) 5, S. 108-110.
- [S11] SIGL, I.: Computergestütztes Informationsmanagement für das Fertigungssystem Kleben. München: TU, Diss. 1997
- [S12] SPUR, G.; KRAUSE, F.-L.: CAD-Technik. München: Hanser 1984.
- [S13] SPUR, G.; KRAUSE, F.-L.: Das virtuelle Produkt. München: Hanser 1997
- [S14] SPUR, G.; ZIEBEIL, P.: Integrierte Gestaltung und Berechnung durch rechnerunterstützte Navigation in Lösungsräumen. Konstruktion 49 (1997), S. 27-30.
- [S15] STABEROH, U.; HÜBER, N.; SELIGER, J.: 3D-CAD-Einsatz im Spannungsfeld der frühen Produktentwicklungsphase. In: Effiziente 3D-Produktmodellierung – Fortschritte und Fallstricke. Düsseldorf: VDI-Verlag 2001, S. 61-78. (VDI-Berichte 1614)
- [S16] STACHOWIAK, H.: Allgemeine Modelltheorie. Wien: Springer 1973
- [S17] STAUB, G.: Komponentenbasierte Entwicklung und Implementierung von Produktmodellen als Basis für interoperable Ingenieursanwendungen. Aachen: Shaker 2000. (Forschungsberichte aus dem Institut für Rechneranwendung in Planung und Konstruktion der Univ. Karlsruhe, Band 4/2000). Zugl. Karlsruhe: Univ., Diss. 1999
- [S18] STEGER, W.: Integration von Berechnungsmethoden in rechnerunterstützte Konstruktionssysteme. Dresden: TU, Diss. 1998
- [S19] STEINBRINK, O.; KUHN, G.; WARTZACK, S.; SCHWEIGER, W.; MEERKAMM, H.: Rechnerunterstützung im Wechselfeld zwischen Konstruktion und Berechnung. Konstruktion 51 (1999) 3, S. 13-19.
- [S20] STEINGRÖVER, K.: FVA-Forschungsvorhaben Nr. 241: FVA-Stirnradprogramm STplus. Benutzeranleitung, Programmdokumentation. FVA-Heft 477 (2000) – Firmenschrift
- [S21] STEINGRÖVER, K.: FVA-Forschungsvorhaben Nr. 320: Schneckenprogramm-system. Benutzeranleitung, Programmdokumentation. FVA-Heft 556 (2000) – Firmenschrift
- [S22] STEINGRÖVER, K.: Programmierrichtlinie für Berechnungsprogramme der FVA, Forschungsvereinigung Antriebstechnik e.V.. FVA-Merkblatt 0/11 (2001) – Firmenschrift
- [S23] STEINMEIER, E.: Realisierung eines systemtechnischen Produktmodells – Einsatz in der PKW-Entwicklung. Aachen: Shaker 1999. (Konstruktionstechnik München, Band 28). Zugl. München: TU, Diss. 1998

- [S24] SWIENCZEK, B.; ARNOLD, F.; KILB, T.; JANOCZA, A.; SARTIONO, R.: Online-Kopplung von CAX-Systemen für die virtuelle Produktentwicklung – ein Vergleich mit dem dateibasierten Datenaustausch. In: Prozessketten für die virtuelle Produktentwicklung in verteilter Umgebung. Düsseldorf: VDI-Verlag 1999, S. 219-238. (VDI-Berichte 1435).
- [T1] THEIßEN, J.: Kopplung von Gestaltung und Berechnung während und nach der Entwicklungsphase einer Industrie-Getriebebaureihe. In: Verkürzte Entwicklungsprozesse durch Integration von Gestaltung und Berechnung. Düsseldorf: VDI-Verlag 1999, S. 139-158. (VDI-Berichte 1487)
- [T2] THEIßEN, J.: Kopplung von Gestaltung und Berechnung während und nach der Entwicklungsphase einer Industrie-Getriebebaureihe. Konstruktion (2000) 1/2, S. 21-23.
- [T3] TÖNSHOFF, H.; WOELK, P.-O.; ZAHN, G.: Datenmodell zum systemneutralen Informationsaustausch auf der Basis Technischer Elemente. Konstruktion (2000) 4, S. 28-29.
- [T4] TÖNSHOFF, H.; ZAHN, G.; WOELK, P.-O.: Datenmodell zum systemneutralen Informationsaustausch auf der Basis Technischer Elemente. In: Beschleunigung der Produktentwicklung durch EDM/PDM- und Feature-Technologie. Düsseldorf: VDI-Verlag 1999, S. 431-449. (VDI-Berichte 1497).
- [T5] TRAUTHEIM, A.: Die Produktdatennorm STEP und der Nutzen für PDM. In: Engineering Data Management (EDM). München: Herbert Utz 1998, S. 135-160 (Seminarberichte IWB).
- [T6] TRAUTHEIM, A.: Produktdatenmanagement (PDM) und der Einsatz von STEP. Schmiede-Journal (2001) 3, S. 32.
- [T7] TROPSCHUH, P.: Rechnerunterstützung für das Projektieren mit Hilfe eines wissensbasierten Systems. München: Hanser 1989. (Konstruktionstechnik München, Band 1). Zugl. München: TU, Diss. 1988
- [V1] VAJNA, S.: Entwicklungsstand von CAD/CAM-Systemen. VDI-Z Special C-Techniken (1997) 10, S. 16-21.
- [V2] VAJNA, S.: Die neue Richtlinie VDI 2219: Praxiserprobte Hinweise zu Einführungsstrategien und Wirtschaftlichkeit von EDM/PDM-Systemen. In: Beschleunigung der Produktentwicklung durch EDM/PDM- und Feature-Technologie. Düsseldorf: VDI-Verlag 1999, S. 25-42. (VDI-Berichte 1497).
- [V3] VAJNA, S.: Die neue Richtlinie VDI 2209 – Praxiserprobte Hinweise zur 3D-Produktmodellierung. In: Effiziente 3D-Produktmodellierung – Fortschritte und Fallstricke. Düsseldorf: VDI-Verlag 2001, S. 3-22. (VDI-Berichte 1614)
- [V4] VAJNA, S.; BOGÁR, R.; BERCESEY, T.: Anwendungsneutrale Metrik für CAD/CA-Systeme. CAD-CAM Report 18 (1999) 6, S. 38-47.
- [V5] VAJNA, S.; MUTH, M.; SANDER, R.; OBINGER, F.: Einsatz der Parametrik in der Produktentwicklung. VDI-Z Special C-Techniken (1998) 3, S. 38-42.

-
- [V6] VAJNA, S.; WEBER, C.; SCHLINGENSIEPEN, J.; SCHLOTTMANN, D.: CAD/CAM für Ingenieure. Braunschweig: Vieweg 1994
- [V7] VAJNA, S.; WEGNER, B.: Stand und Tendenzen der rechnerunterstützten Modellierung. CAD-CAM Report 15 (1996) 2, S. 70-80.
- [V8] VAJNA S.; WEGNER, B.: Support the product life cycle by a computer aided, integrated, feature based modelling system. In: Riitahuhta, A. (Ed.): Proc. of the 11th Intern. Conference on Engineering Design ICED 1997, Tampere (Finnland), 19.-21.08.1997. Tampere: University of Technology 1997, S. 301-306.
- [V9] VDA-Empfehlung 4950: Schnittstellen für CAD/CAM-Daten. Frankfurt a. M. VDA-Verlag 1996.
- [V10] VDA-Empfehlung 4951: Datenübertragung von CAD/CAM-Daten, ENGDAT / ENGPART. Frankfurt a. M.: VDA-Verlag 2000.
- [V11] VDI-Richtlinie 2218 (Entwurf): Feature-Technologie. Düsseldorf: VDI-Verlag 1999
- [V12] VDI-Richtlinie 2219 (Entwurf): Datenverarbeitung in der Konstruktion – Einführung und Wirtschaftlichkeit von EDM/PDM-Systemen. Düsseldorf: VDI-Verlag 1999
- [V13] VDI-Richtlinie 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte. Düsseldorf: VDI-Verlag 1993
- [V14] VOGEL, M.; BUNTE, P.: PRO/ENGINEER und PRO/MECHANICA. Konstruieren, Berechnen und Optimieren. München: Hanser 2001
- [W1] WARTZACK, S.; MEERKAMM, H.: Durchgängige Rechnerunterstützung in der Produktentwicklung durch den Einsatz von semantisch hochwertigen Features. In: Beschleunigung der Produktentwicklung durch EDM/PDM- und Feature-Technologie. Düsseldorf: VDI-Verlag 1999, S. 369-391. (VDI-Berichte 1497).
- [W2] WASMER, A.: Methodische Vorgehensweise beim Entwurf von Mappings zwischen Produktmodellen. Darmstadt: TU, Diss. 1998
- [W3] WEBER, C.; KRAUSE, F.-L.: Features mit System – die neue Richtlinie VDI 2218. In: Beschleunigung der Produktentwicklung durch EDM/PDM- und Feature-Technologie. Düsseldorf: VDI-Verlag 1999, S. 43-75. (VDI-Berichte 1497).
- [W4] WEBER, C.; MUTH, M.; BÄR, T.; WERNER, H.: Intelligente Kopplung von Gestaltung und Berechnung. In: Verkürzte Entwicklungsprozesse durch Integration von Gestaltung und Berechnung. Düsseldorf: VDI-Verlag 1999, S. 47-70. (VDI-Berichte 1487)
- [W5] WECK, M. BOSHOFF, F.: Integration von CAD und FEM durch bidirektionalen automatischen Geometriaustausch. Konstruktion 48 (1996), S. 299-304.
- [W6] WECK, M.; DAMMER, M.; CARMER, E.: Integration von Gestaltung und Berechnung – mit geeignetem Informationsmanagement zum integrierten Konstruktionsarbeitsplatz. In: Neue Generation von CAD/CAM-Systemen. Düsseldorf: VDI-Verlag 1997, S. 105-124. (VDI-Berichte 1357).

-
- [W7] WECK, M.; HURASKY-SCHÖNWERTH, O.; SCHÄFER, J.: Berechnung von Stirnradpaarungen mit Mehrfacheingriffen mit Hilfe der Finite-Element-Methode. In: FVA-Forschungsreport 2001 anlässlich FVA-Informationstagung, Würzburg, 27./28.11.2001, S. 377.1-377.14
- [W8] WEHLITZ, P.: Nutzenorientierte Einführung eines Produktdatenmanagement-Systems. München: TU, Diss. 2000
- [W9] WEIDNER, K.: Das LRZ-Graphiksystem Benutzer Manual - Teil I Grundsoftware. LRZ-Benutzerschrift Nr. 7708/14 (1996) - Firmenschrift
- [W10] WELLNIAK, R.: Das Produktmodell im rechnerintegrierten Arbeitsplatz. München: Hanser 1995. (Konstruktionstechnik München, Band 17). Zugl. München: TU, Diss. 1994
- [W11] WELP, E.; BRAUN, P.: Semantische und systemtechnische Kopplung kommerzieller CAD- und Berechnungssysteme durch einen objektorientierten Integrationsprozessor. In: Neue Generation von CAD/CAM-Systemen. Düsseldorf: VDI-Verlag 1997, S. 125-142. (VDI-Berichte 1357).
- [W12] WENZEL, B.: Die Zukunft von EXPRESS. In: CAD'94 – Produktmodellierung und Prozessmodellierung als Grundlage neuer CAD-Systeme. Paderborn: Hanser 1994, S. 655-658. (Heinz Nixdorf Institut 1994).
- [W13] WESER, G.: Intelligente CAD-Bauteile auf Basis der Berechnung. Konstruktion (2001) 6, S. 44-45.
- [W14] WINTER, H.; PLACZEK, T.; BENKLER, K.: Konstruktion und Berechnung von Getrieben mit FVA-Programmen – dargestellt am Beispiel des virtuellen FVA-Getriebes. Antriebstechnik 27 (1988) 7, S. 35-40.
- [W15] WINTER, H.; PLACZEK, T.; BENKLER, K.: Programmsystem zur Auslegung und Berechnung von Stirnradgetrieben. Antriebstechnik 29 (1990) 9; S. 75-81.
- [W16] WIRTZ, J.: Ein Referenzmodell zur integrationsgerechten Konzeption von Produktdatenmanagement. München: TU, Diss. 2001
- [W17] WÖLFLE, F.: Virtuelle Berechnungskompetenzzentren als Dienstleister zur Integration von Gestaltung und Berechnung. Düsseldorf: VDI-Verlag 1998. (Fortschritt-Berichte der VDI-Zeitschriften Reihe 20, Nr. 274)
- [Z1] ZENNER, H.; MARQUARDT, C.: Ermittlung von Bauteilwöhlerlinien mittels künstlicher neuronaler Netze. In: FVA-Forschungsreport 2001 anlässlich FVA-Informationstagung, Würzburg, 27./28.11.2001, S. 15.1-15.18