

# Informationstechnologien im digitalen Produkt

**Educational Material**

**Author(s):**

Zwicker, E.; Montau, R.

**Publication date:**

2006

**Permanent link:**

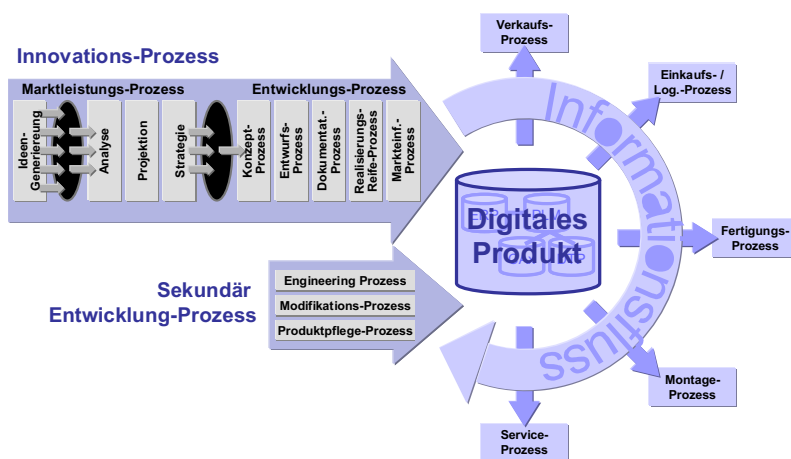
<https://doi.org/10.3929/ethz-a-005173604>

**Rights / license:**

In Copyright - Non-Commercial Use Permitted

# Informationstechnologien im Digitalen Produkt

Autoren: Dr. E. Zwicker, Dr. R. Montau  
Sommersemester 2006





## Informationstechnologien im Digitalen Produkt

<b>Das Digitale Produkt</b> .....	<b>1</b>
Überblick .....	3
Das Digitale Produkt .....	6
Grundlagen der rechnerintegrierten Produktentwicklung .....	10
Rechnerintegration auf Basis der Modelltransformation .....	12
Rechnerintegrierte Produktentwicklung auf Basis der Modellintegration .....	15
Einordnung des integrierten Produktmodells in das Digitale Produkt .....	20
Zusammenfassung .....	22
<b>Einführung in die PLM-Technologie</b> .....	<b>1</b>
Überblick .....	3
Historische Entwicklung und Problemstellung .....	4
Rationalisierungsdruck durch die Verschärfung der Marktsituation ..	6
Charakterisierung des PLM-Ansatzes .....	12
Anwendungsfunktionen von PLM-Systemen .....	14
Informationshandhabung mit PLM-Systemen .....	17
Systemarchitektur von PLM-Systemen .....	20
Zusammenfassung .....	24
<b>Datenbanktechnologie im Digitalen Produkt</b> .....	<b>1</b>
Überblick .....	3
Modellbildung von Datenbankapplikationen .....	8
Konzeptueller Datenbankentwurf .....	9
Semantische Informationsmodellierung .....	10
Relationenmodell als operationales Modell .....	12
Abbildung vom ERM auf das Relationenmodell .....	15
Anfragesprache SQL .....	18
Triggerfunktionalität von SQL .....	23
Transaktionsverwaltung von Datenbanksystemen .....	24
Datensicherung in PLM-Systemen .....	28
Verteilung in PLM-Systemen .....	35
Anforderungen an die PLM-Verteilungsfunktionalität .....	35
Verteilungskonzepte in PLM-Systemen .....	38
Zusammenfassung .....	42
<b>Objektmanagement</b> .....	<b>1</b>
Überblick .....	3
PLM-Objektmodell .....	5
Abstraktionsebenen der PLM-Informationsmodellierung .....	9
Beziehungen zwischen Objekten .....	11
Objektrelationen (Zuordnungen) .....	12
Objektstrukturen .....	19
Produktstrukturen .....	21
Generische Konfigurationsmechanismen .....	27
Status-Konfiguration .....	28
Effectivity-Konfiguration .....	29
Variantenbildung .....	36
Konzepte der Variantendefinition .....	36
Benummerung von Produktvarianten .....	38
Ableitung auftragsspezifischer Strukturausprägungen .....	39
Methoden von PLM-Systemen zur Variantendefinition .....	41
Autorisierung .....	43
Grundkonzepte der Autorisierung .....	43
Zusammenfassung .....	54



<b>Objektklassifikation</b> .....	<b>1</b>
Überblick .....	3
Nutzgrößen der Wiederholteilverwendung .....	4
Betriebliche Anwendungsgebiete der Objektklassifikation .....	6
Begriffe, Merkmale und Eigenschaften .....	8
Gliederung und Beschreibung .....	13
Methoden der Gruppentechnik .....	15
Begriffssysteme .....	17
Klassifikationsschlüssel .....	22
Sachmerkmalelisten .....	30
Verbindung BS/SML .....	37
Merkmalbasierte Gliederung und Beschreibung .....	41
Generische Objektsysteme .....	44
Clusteranalyse .....	48
Faktorenanalyse .....	52
Fuzzy-Sets .....	53
Fourierdeskriptoren .....	55
Zusammenfassung .....	57
<b>Objektidentifikation mit Sachnummernsystemen</b> .....	<b>1</b>
Überblick .....	3
Konzepte der Nummerungstechnik .....	5
Aufgaben einer Nummer .....	5
Aufbauprinzipien .....	9
Wertebereichsbildung .....	11
Kombinationsprinzipien .....	12
Sachnummerierung .....	15
Anwendungsbezogene Klassifizierungsmerkmale von Sachnummern .....	15
Charakterisierung von Sachnummernsystemen .....	20
Beispielhafte Analyse eines Sachnummernsystems .....	21
Objektidentifikation in PLM-Systemen .....	25
Zusammenfassung .....	29
<b>CAx/PLM-Integration</b> .....	<b>1</b>
Überblick .....	3
Check In/Check Out-Mechanismus .....	4
Expliziter und impliziter Check Out .....	5
Export-Funktionalität beim Check Out .....	6
Check Out-Sperren versus Prozess-Sperren .....	7
Concurrent Engineering .....	8
Einkapselung von Fremdapplikationen .....	9
„Tiefe“ Integration von Autorenssystemen .....	10
Integration von E-CAD mit PLM/M-CAD .....	23
Visualisierung und Rendering .....	25
Generierung neutraler Formate (Rendering) .....	25
Viewing und Markup .....	26
Prozessbezug von PLM-Applikationen .....	29
Zusammenfassung .....	32
<b>Workflow Management</b> .....	<b>1</b>
Überblick .....	3
Freigabewesen .....	4
Änderungswesen .....	8
Methoden zum Kopieren von Objekten zur Revisionierung .....	18
Zusammenfassung .....	21

<b>Schnittstellen im Digitalen Produkt .....</b>	<b>1</b>
Überblick .....	3
Motivation und Zielsetzung .....	6
<b>CAx-Schnittstellen der rechnerintegrierten Produktentwicklung .....</b>	<b>8</b>
Interne CAx-Softwareschnittstellen .....	9
Externe CAx-Softwareschnittstellen .....	11
Externe CAx-Softwareschnittstellen zum Austausch von Produktdaten .....	12
Schnittstellen zum Austausch von Produktdaten .....	13
Schnittstellen zum Austausch von Prozessdaten .....	18
<b>Übersicht über ausgewählte Produktdatenschnittstellen .....</b>	<b>22</b>
IGES .....	23
VDA-FS .....	25
DXF .....	27
<b>Die ISO 10303 „STEP“ .....</b>	<b>28</b>
Zielsetzung .....	28
Architektur und Aufbau .....	31
Die Integrierten Ressourcen (integrated resources) .....	35
Die Anwendungsmodelle (application resources) .....	44
Die Anwendungsprotokolle (application protocols) .....	47
Die Datenschnittstelle SDAI .....	58
Ausblick und Kritik .....	59
<b>Integration von Norm- und Zukaufteile .....</b>	<b>62</b>
CAD-Teilebibliotheken .....	64
Beschreibungsverfahren .....	65
Neutrale Parameterrepräsentation auf Basis der DIN V 4001 .....	68
VDA-Programmschnittstelle nach DIN V 66304 VDAPS .....	71
<b>Die ISO 13584 „Parts Library“ .....</b>	<b>73</b>
Aufbau und Entwicklung der ISO 13584 .....	76
Konzeptuelles Modell einer Teilebibliothek gemäss ISO 13584 .....	78
Das Architektur Referenzmodell .....	80
Das Strukturreferenzmodell .....	81
Das Information Semantic Reference Model .....	92
Ausblick .....	94
<b>Zusammenfassung .....</b>	<b>95</b>

<b>Enterprise Application Integration (EAI)</b> .....	<b>1</b>
Überblick .....	3
Datenübergabe an ERP .....	6
Prinzipielle Aufgabenteilung PLM und ERP .....	6
Informationsfluss PLM – ERP .....	6
<b>Systemtechnologie – Motivation</b> .....	<b>13</b>
Physikalische Verbindungen zwischen zwei Datenendgeräten – High-Low, Ja-Nein und deren Übertragung .....	14
Physikalische Verbindungen zwischen zwei Datenendgeräten – Digitale Informationsübertragung .....	15
Physikalische Verbindungen zwischen zwei Datenendgeräten – Synchrone und asynchrone Übertragung .....	16
Physikalische Verbindungen zwischen zwei Datenendgeräten – Bussysteme .....	18
Physikalische Verbindungen zwischen zwei Datenendgeräten – Geschwindigkeit .....	19
Physikalische Verbindungen zwischen zwei Datenendgeräten – Modulation und Demodulation (Modem) .....	20
Physikalische Verbindungen zwischen zwei Datenendgeräten – Integrated Services Digital Network (ISDN) .....	21
Kommunikationsprotokolle .....	22
Applikationen .....	29
Verteilte Systeme – LAN/WAN .....	34
Verteilte Systeme – Arten von Netzwerken .....	35
Verteilte Systeme – Client/Server .....	37
Verteilte Systeme – Fenstersysteme .....	38
Verteilte Systeme – NFS und NIS .....	39
Verteilte Systeme – Datenbanksysteme in einer verteilten Umgebung .....	39
WWW .....	41
Systemschnittstellen .....	45
Industrielle Nutzung dieser Technologien – Geschwindigkeit .....	48
Industrielle Nutzung dieser Technologien – Kommunikation innerhalb eines verteilten PLM-Systems .....	49
Industrielle Nutzung dieser Technologien – Kommunikation zwischen weltweit verteilten Systemen .....	50
Industrielle Nutzung dieser Technologien – Kosten .....	52
Industrielle Nutzung dieser Technologien – Sicherheit .....	53
<b>Zusammenfassung</b> .....	<b>54</b>

# Das Digitale Produkt

Autor: Dr. E. Zwicker



# 1 Überblick

Die Trends der Gegenwart können wie folgt zusammengefasst und kategorisiert werden:

- *Globalisierung* der Märkte und Firmen:  
Unternehmen fusionieren in einer zunehmenden Dynamik – Ergebnis sind global verteilte Unternehmensstrukturen mit entsprechend global verteilten Entwicklungs-/Produktionsstandorten. Der Zuliefermarkt und Absatzmarkt ist einerseits weit über die lokale Umgebung hinaus gewachsen – andererseits kooperieren Unternehmen mit Zulieferanten und Kunden immer enger. Schon heute ist beispielsweise für viele Schweizer Unternehmen als Absatzmarkt Schweiz nur noch von geringer Bedeutung.
- Moderne *Informationstechnologien* und wachsende *Informationsvernetzung*:  
Das Internet bzw. die Internettechnologie im allgemeinen ermöglicht den zeitlich und örtlich unabhängigen Zugriff auf Daten, Informationen und Wissen. Global verteiltes Wissen wird transparent und die nutzenorientierte Bereitstellung von Produktdaten für Kunden und Zulieferanten stellt eine neue Herausforderung dar. Die heute vor allem lokal zur Anwendung kommenden Softwaretools wandeln sich aus diesem Kontext heraus zu strategischen, unternehmensübergreifenden Werkzeugen.  
CAx-, PLM- oder ERP-Systeme verbinden sowohl einzelne Geschäftsbereiche eines Unternehmen als auch global verteilte Firmen durch eine breite Funktionalität und Datenpipelines. Die Bereichswände bzw. Grenzen innerhalb eines Unternehmen sowie nach aussen werden dadurch in Frage gestellt.
- *Innovation* erhält eine höhere Priorität:  
Reine Prozessoptimierungen und evolutionäre Produktverbesserungen sind ausgereizt und bieten keine grossen Differenzierungssprünge am Markt. Dies wird von vielen Unternehmen heute erkannt und die Suche nach revolutionären Produktideen gehört wieder an die obere Stelle der strategischen Prioritäten.

Was sind die Erfolgspotentiale zukünftiger Innovationen im Überblick:

- Branchenfokussierung  
Die Globalisierung der Märkte erfordert, dass die Produkte nicht mehr eine diffuse Breite von potentiellen Kunden ansprechen, sondern auf eine ganze Branche (branchenspezifisch) abzielen. Diese Einengung des Kundenprofils bereitet Mühe, ist

aber der Schlüssel zur Marktführung und wird durch die nächsten zwei Punkte erweitert.

- Erweiterung der Systemgrenze eines Produktes  
Die Systemgrenze des Verkaufsobjektes wird massiv vergrößert und umfasst nebst dem Produkt als Einzelobjekt weitere Leistungen am Markt über den gesamten Lebenszyklus. Z.B. sind Kundenberatung, Schulung, Wartung oder Rücknahmen nicht mehr notwendige Nebenprodukte, sondern gezielte, integrale Leistungen und Differenzierungen des Produktes selber und werden entsprechend ausgeweitet. Den zugehörigen Produktdaten und deren Nutzung innerhalb auf das Produkt und den Markt abgestimmten Unternehmensprozessen kommt damit eine zentrale Bedeutung zu.
- Vollständige, modulare Produkte  
Standardprodukte können vielfach die Kundenbedürfnisse nicht optimal befriedigen und sind auch einfach zu kopieren. Um aber nicht einen aufwendigen Sondermaschinenbau zu betreiben, ist es notwendig, schon in der Entwicklung ein modulares, vollständiges Konzept zu erarbeiten, welches in der definierten Kombinatorik bestmöglich alle Bedürfnisse des Kundensegmentes abdeckt – und die Ziele des Unternehmens nach Standardisierung und optimalem Herstellungsprozess gleichbedeutend sicherstellt.
- Integrierte mechatronische Systeme  
Die Integration von Mechanik, Elektronik und Software zu sogenannten integrierten mechatronischen Systemen gewinnt von Tag zu Tag an Bedeutung. Damit wird die Entwicklung modularer Produkte zu einer interdisziplinären Aufgabe von Mechanik, Elektronik und Software, innerhalb welcher in sich vollständig definierte Module definiert werden, die entsprechend der Kundenanforderungen zu Gesamtlösungen integriert werden können.  
Damit einhergehend kommt der Integration der jeweils spezialisierten Entwicklungssysteme aus den einzelnen Bereichen (Mechanik: M-CAD, Elektronik: E-CAD, Software: CASE-Tools) eine entscheidende Bedeutung zu.
- Produktdatenbereitstellung  
Produktdaten rücken in das Zentrum des Unternehmens und werden zur Informationsplattform aller Unternehmensprozesse. Das Entwicklungsteam beschränkt sich nicht ausschliesslich auf die Produktdatenerstellung innerhalb der Entwicklung bzw. für die Prozesskette Entwicklung-Produktion, sondern beschreibt

das Produkt umfassend mit erweiterten Informationen, welche vom Verkauf, Kunden, Service, von der Montage und der Fertigung genutzt werden können. Kunden greifen z.B. direkt auf bereit gestellte Informationen zu, integrieren diese und verwenden sie direkt innerhalb ihrer Produktentwicklung (bzw. Anwendung).

- Unternehmensbereiche rücken zusammen.  
Die integrierte Produktdatenhaltung innerhalb eines Unternehmens oder in der Zusammenarbeit mit Kunden und Lieferanten ermöglicht eine Parallelisierung und ein effizienteres Durchführen der heutigen seriellen Abläufe.

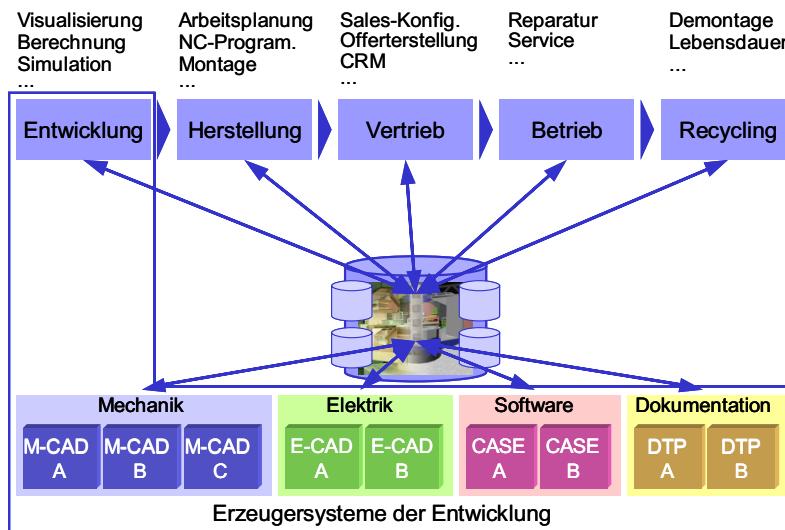


Bild (B001dprZ) Datenerzeugung und Datennutzung basierend auf dem Digitalen Produkt



## 2 Das Digitale Produkt

Das Konzept des Digitalen Produktes zielt auf eine definierte, konsistente Daten- und Informationsstruktur. Alle Daten, die das Produkt realitätsnah beschreiben und von Unternehmensprozessen genutzt werden, sind in dedizierten Datenbanken mit festgelegter Datenhoheit abgelegt und miteinander verknüpft.

Die Produktdaten werden innerhalb der Primärentwicklung erzeugt und während des gesamten Produktlebenszyklus ergänzt, modifiziert und gepflegt. Die Erzeugung erfolgt mit dem Ziel der optimalen Nutzung in den nachgelagerten Unternehmensprozessen: Herstellung, Vertrieb und Marketing, Betrieb und Service, Entsorgung und Recycling. Damit profitieren von diesen Daten die Unternehmensprozesse Verkauf, Kunde, Service, Produktion und Logistik, denn sie können jederzeit auf die Informationen des Digitalen Produktes zugreifen und effizient Nutzen daraus ziehen.

Neue Bedürfnisse, Schwachstellen, Erfahrungen fließen permanent in den Innovationsprozess zurück und werden durch die Sekundärentwicklung in das Digitale Produkt integriert oder verbessern dieses.

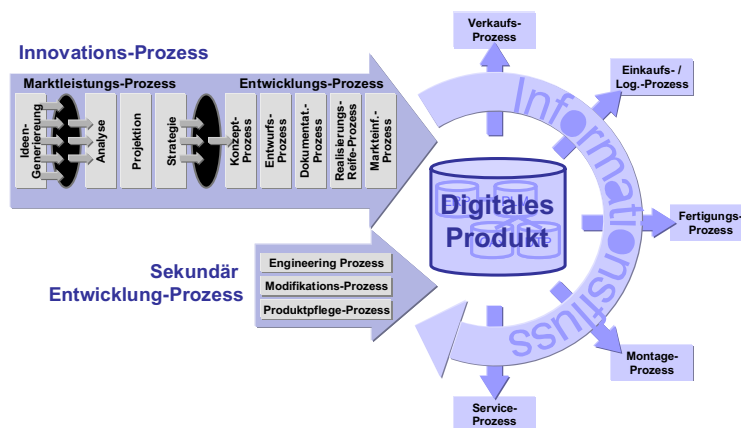


Bild (B002dprZ) Grundkonzepte des Digitalen Produkts

Das Digitale Produkt unterstützt den Innovationsprozess:

- durch die marktgerechte Strukturierung und Modularisierung der Produkte,
- durch frühzeitige Verifizierung und Optimierung der Produkte durch CAx- und Visualisierungstools,

- durch strukturierte Ablage der entstehenden Produktdaten, so dass diese direkt in den nachgelagerten Unternehmensprozessen genutzt werden können.

Die Daten des Digitalen Produkts lassen sich direkt in den nachgelagerten Unternehmensprozessen nutzen und fliessen in den Innovationsprozess zurück (Sekundärentwicklung). Im Rahmen einer Sekundärentwicklung sollen kundenspezifische Anpassungen schnell realisiert, Produkte verbessert/optimiert und Informationen im Digitalen Produkt laufend integriert werden.

Am Zentrum für Produkt-Entwicklung ist das Digitale Produkt wie folgt definiert: Das Digitale Produkt ist die Gesamtheit der Produktdaten, welche in der Primärentwicklung erzeugt, konsistent verwaltet und über den Lebenszyklus laufend ergänzt werden und das reale Produkt hinreichend genau repräsentieren, um von *Unternehmensprozessen* mittels *Diensten* genutzt zu werden.

*Unternehmensprozesse* sind wertschöpfende Handlungen des Unternehmens unter Einbezug aller Partner, welche das Digitale Produkt unter Verwendung von *Diensten* nutzen, modifizieren und ergänzen.

*Dienste* sind spezifische Anwendungen, welche es Unternehmensprozessen ermöglichen, individuell oder in Teams, im Digitalen Produkt zu navigieren, zu interagieren, dieses zu visualisieren und zu modifizieren. Für die Realisierung der Dienste kommen moderne kommerziell erhältliche oder eigens implementierte IT-Tools zur Anwendung.

Massgeblich beeinflusst bzw. definiert wird das Digitale Produkt durch

- die Produktplattform eines Unternehmens „P“,
- definierte Unternehmensprozesse im und ausserhalb des Unternehmens „U“,
- leistungsfähige, moderne IT-Tools „T“ die zur Anwendung kommen.

Nur das Gleichgewicht des Dreieckes Produktplattform – Unternehmensprozesse – IT-Tools ermöglicht innerhalb des Digitalen Produktes die erwünschte nachhaltige Effizienz- und Qualitätssteigerung.

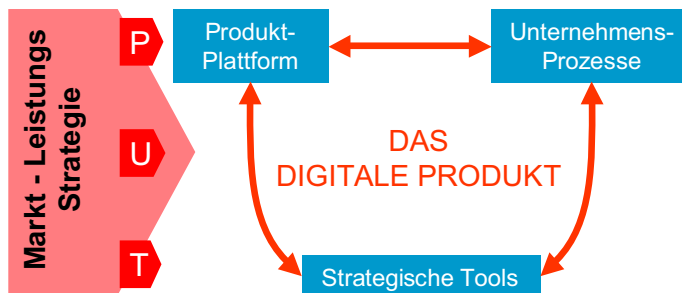


Bild (B003dprZ) Das P-U-T Prinzip

Eine *Produktplattform* besteht aus einer definierten Menge an Modulen, welche basierend auf definierten Regeln zu kunden- bzw. auftragspezifischen Produkten konfiguriert werden können. Ein Modul ist in verschiedenen Produkte-Plattformen einsetzbar und hat einen eigenen Lebenszyklus (z. B. primärer Entwicklungsprozess). Innerhalb des Produktkonzeptes werden die einzelnen Module, deren Eigenschaften und die verschiedenen Kombinations- und Variationsmöglichkeiten zu einer Gesamtlösung festgelegt.

Diese wird einer oftmals sehr aufwendigen interdisziplinären Arbeit für eine neue Produktplattform innerhalb des Innovationsprozess oder bestehende Produktlinie innerhalb eines Reengineering festgelegt. Dazu bewährt sich die Methodik der Produkte-Konfigurationsmatrix, innerhalb welcher die auftragsneutrale Produktstruktur (mit den einzelnen Sichten, Funktionen, Modulen, Baugruppen) definiert und mit der Kundensicht des Produktes (Kundenmerkmale zur Auswahl der benötigten Produktkonfiguration) in Beziehung gebracht wird.

Die Produktplattform umfasst damit:

- die Eigenschaften der einzelnen Module, bestehend aus der Modul-Parametrik, d.h. alle variablen Grössen (Masse, Gestalt, Struktur, Werkstoff etc.)
- die allgemeine, auftragsneutrale Produktstruktur bestehend aus den einzelnen Modulen mit den verschiedenen Regeln zur Ableitung bzw. Erstellung der kunden- bzw. auftragspezifischen Produktstruktur
- sowie die jeweils zugeordneten Informationsobjekte (Geometriemodell, Offerttexte, Betriebs- Serviceanleitungen, Preise, Präsentationsobjekte etc.),

Entsprechend dem Produktkonzept sowie der strategischen Unternehmensausrichtung im Hinblick auf Zulieferanten und Kunden werden die *Unternehmensprozesse* definiert, welche die Daten des Digitalen

Produktes im gesamten Produktlebenszyklus innerhalb oder ausserhalb des Unternehmens nutzen.

Zur Realisierung der innerhalb der Unternehmensprozesse benötigten Dienste werden die oben erwähnten IT-Tools als die für ein Unternehmen *strategischen Tools* benötigt. Hierfür kommen so weit als möglich kommerziell erhältliche Software-Tools zur Anwendung, die entsprechend den Anforderungen resultierend aus dem Produktkonzept sowie den Unternehmensprozessen angepasst und falls notwendig durch spezielle Implementierungen ergänzt werden. Die einzelnen Software-Tools gilt es als Subsysteme innerhalb eines funktionell zu definieren und untereinander abzugrenzen. Die funktionelle Abgrenzung erfolgt auf Basis der in den einzelnen Subsystemen jeweils benötigten Informationen des Digitalen Produktes sowie den Unternehmensprozessen die diese Informationen nutzen.

Von Bedeutung sind in diesem Zusammenhang:

- CAx-Tools zur Datenerzeugung (CAD/CAE) des Digitalen Produktes;
- PLM-Tools zur Datenverwaltung innerhalb des Digitalen Produktes
- CAx-, ERP- und Viewing-Tools zur Datennutzung

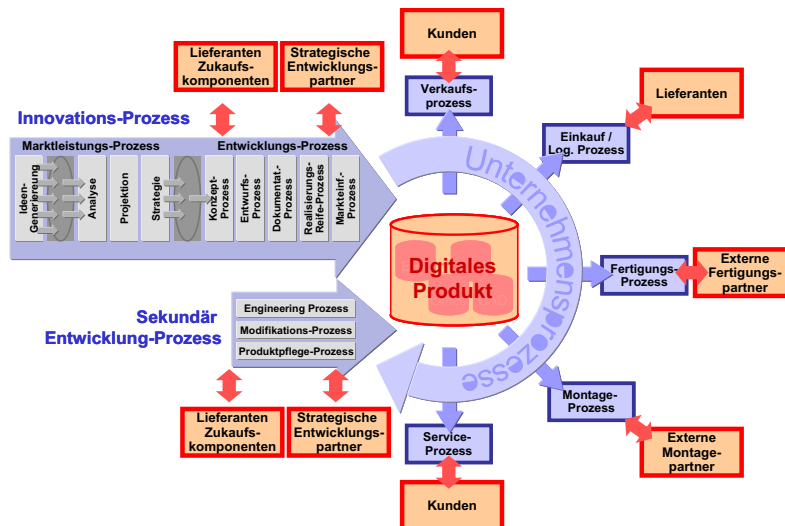


Bild (B004dprZ) Anwendung von CAx-/PLM-/ERP-Tools im Digitalen Produkt

### 3 Grundlagen der rechnerintegrierten Produktentwicklung

Der Begriff rechnerintegrierte Produktentwicklung steht für die technologische und organisatorische Integration von Informationstechnologien in ein Gesamtkonzept zur durchgängigen Informationsverarbeitung innerhalb aller Funktionsbereiche der Produktentwicklung.

Bei der *konventionellen Produktentwicklung* (vgl. Bild B005dpdZ Spalten 1 und 2) werden die Produktdaten (Zeichnungen, Anforderungslisten, Arbeitspläne etc.) manuell bzw. unterstützt durch Computersysteme erstellt und zur Weitergabe an nachgelagerte Unternehmensbereiche auf Papier ausgedruckt. Dort müssen diese Papierdokumente zunächst interpretiert werden, um hierauf aufbauend weitere in diesem Bereich benötigte Unterlagen erstellen zu können.

Im Gegensatz dazu basiert *die rechnerintegrierte Produktentwicklung* ausschliesslich auf der digitalen Bereitstellung, Weiterleitung und Verarbeitung von Produktdaten. Unter dem Begriff Produktdaten werden alle Informationen zusammengefasst, welche innerhalb des gesamten Produktlebenszyklus anfallen, d.h. von der Produktplanung bis zur Produktentsorgung.

Die Informationsverarbeitung in den einzelnen Phasen der Produktentwicklung ist dabei durch den Einsatz von EDV-Systemen geprägt, die jeweils die bereichsspezifischen Tätigkeiten unterstützen.

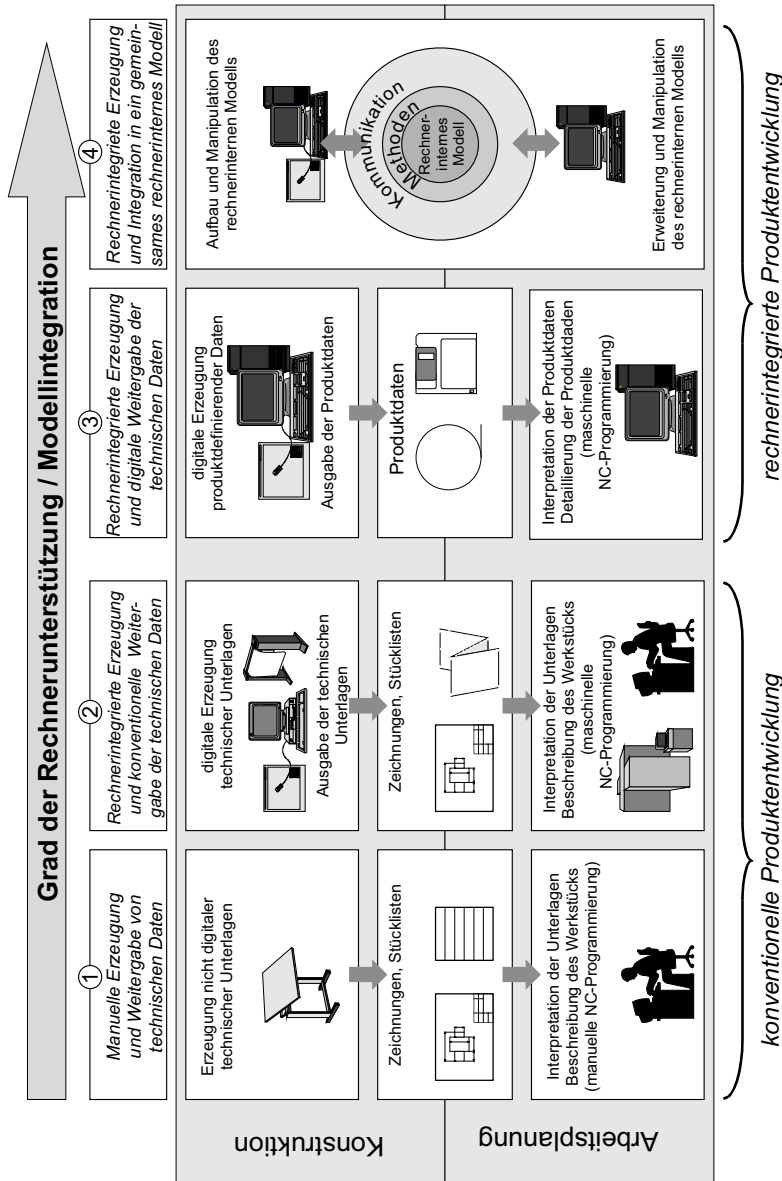


Bild (B005dprZ) Integrationstiefen von DV-Systemen in der Produktentwicklung (Beispielausschnitt Konstruktion und Arbeitsvorbereitung) in Anlehnung an [GAP-93]

Im Rahmen der rechnerintegrierten Produktentwicklung werden zwei wesentliche Arten zu unterscheiden:

- Die Rechnerintegration auf Basis der der Modelltransformation (Spalten 3 in B005dprZ) und
- Die Rechnerintegration auf Basis der Modellintegration (Spalte 4 in Bild B005dprZ).

Die beiden Ansätze soll nachfolgend erläutert werden.

### 3.1 Rechnerintegration auf Basis der Modelltransformation

Im Laufe der Zeit sind in den einzelnen Funktionsbereichen der Produktentwicklung CAx-Systeme entstanden, die für die Ausführung der bereichsbezogenen Aufgaben ausgelegt, entwickelt und implementiert wurden. Man spricht in diesem Zusammenhang von den CAx-Technologien. Dazu gehören hinsichtlich der technischen Prozesskette CAD (Computer Aided Design), CAE (Computer Aided Engineering), CAP (Computer Aided Planning), CAM (Computer Aided Manufacturing) sowie CAQ (Computer Aided Quality Assurance).

Diese werden heute in der Regel als Insellösungen eingesetzt, d.h. sie sind nicht in einen bereichsübergreifenden Verbund eingebettet (vgl. Bild B005dpdZ Spalte 3). Entsprechend der jeweiligen Zielsetzung bzw. Funktionalität werden die Produktdaten innerhalb dieser spezialisierten CAx-Systeme auf unterschiedliche Datenstrukturen abgebildet. Grundlage für diese Datenstrukturen ist jeweils ein *Produktmodell*.

Unter einem Produktmodell versteht man ein semantisches Informationsmodell, das Informationen über ein Produkt abbilden kann. Der abbildbare Informationsgehalt soll dabei das reale Produkt hinreichend genau repräsentieren, um Rückschlüsse auf die realen Produkteigenschaften ziehen zu können.

In den verschiedenen Funktionsbereichen der Produktentwicklung kommen dabei jeweils verschiedene auf die jeweilige Aufgabe spezialisierte Produktmodelle zur Anwendung (z. B. Geometriemodelle, Berechnungsmodelle, Fertigungsmodelle etc.).

Aus der Sicht der Modelltheorie können diese Produktmodelle durch eine Reihe von Merkmalen bestimmt bzw. charakterisiert werden:

- *Abbildungsmerkmal*: Ein Modell ist immer eine Abbildung eines natürlichen oder technischen Objektes
- *Verkürzungsmerkmal*: Modelle erfassen im allgemeinen nicht alle Eigenschaften des durch sie repräsentierten Originals, sondern nur solche, die den jeweiligen Modellentwicklern und/oder Modellbenutzern relevant erschienen.
- *Pragmatisches Merkmal*: Modelle sind ihren Originalen nicht per se eindeutig zugeordnet. Sie erfüllen eine Ersetzungsfunktion für bestimmte Personen, zu gewissen Zeitpunkten und unter Beschränkung auf bestimmte gedankliche oder tatsächliche Operationen.

Im Verlauf des Produktentwicklungsprozesses werden die Arbeitsergebnisse eines Funktionsbereiches als Eingangsinformation an den nachgelagerten Funktionsbereich weitergereicht bzw. von CAx-Sys-

tem zu CAx-System übergeben und innerhalb eines solchem die benötigten Informationen bzw. Arbeitsdokumente erstellt. Für den Produktdatenaustausch zwischen den einzelnen Funktionsbereichen bzw. CAx-Systemen werden daher spezielle Schnittstellen benötigt, welche eine Transformation zwischen den unterschiedlichen Repräsentationen ermöglichen. Man spricht in diesem Zusammenhang von einer Produktmodelltransformation.

Diesen sogenannten Daten-Schnittstellen kommt innerhalb der rechnerintegrierten Produktentwicklung eine wichtige Bedeutung zu, da die Qualität des geforderten Informationsaustausches direkt von der Leistungsfähigkeit dieser Schnittstellen abhängig ist.

Charakteristisch für die rechnerintegrierte Produktentwicklung basierend auf einer Modelltransformation ist, dass in den einzelnen betrieblichen Funktionsbereichen die jeweils benötigten Detaillierungen des Produktes erzeugt und zur Weiterverarbeitung an die nachgelagerten Bereiche übergeben werden. Dabei existiert kein direkter Bezug zwischen den erzeugten Detaillierungen. Deren Konsistenz untereinander ist nicht auf Modellebene sichergestellt und muss durch organisatorische Massnahmen sichergestellt werden (z.B. Freigabe- und Änderungswesen).

Der Einsatz von Verwaltungssystemen zielt in diesem Zusammenhang auf die Sicherstellung der geforderten Konsistenz ab durch die Definition und Sicherstellung festgelegter Prozesse und Kontrollmechanismen (z. B. Autorisierung etc.).



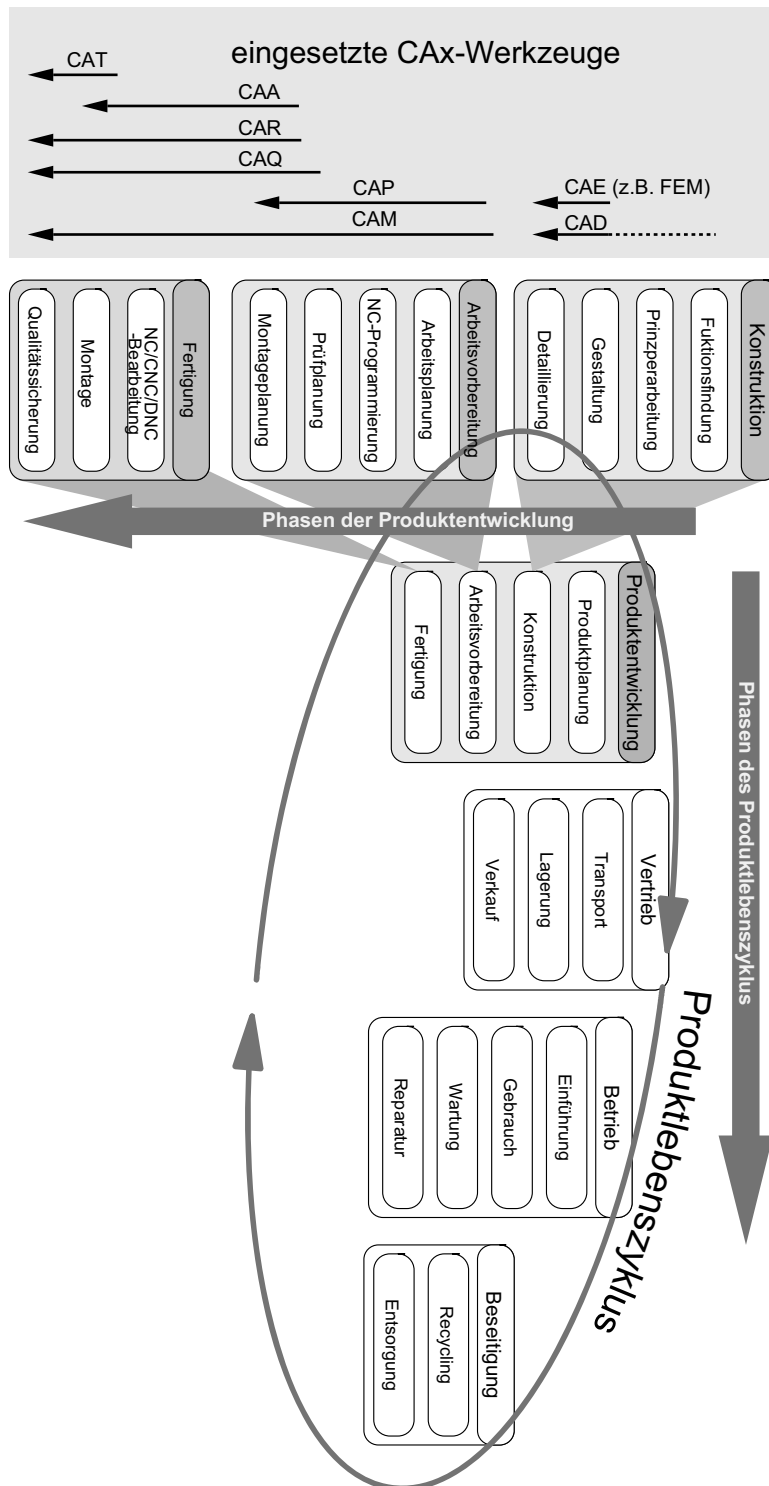


Bild (B006dprZ) Einsatz spezialisierter CAx-Tools im Produktlebenszyklus

### 3.2 Rechnerintegrierte Produktentwicklung auf Basis der Modellintegration

Aus der oben erläuterten Problematik heraus resultiert die Forderung, dass alle Funktionsbereiche auf einem gemeinsamen „digitalen Abbild“ des zu entwickelnden Produktes arbeiten, dem sogenannten rechnerinternen Modell. Auf Basis einer Datenintegration sollen damit die bestehenden Insellösungen zu einem Gesamtsystem integriert werden, wobei die Konsistenz der einzelnen Produktsichten auf bereits Modellebene sichergestellt wird.

Werden alle charakteristischen Merkmale eines Produktes im Verlauf des Produktlebenszyklusses in ein Produktmodell integriert, spricht man von einem *integrierten Produktmodell*.

Das integrierte Produktmodell beinhaltet die Abbildung aller relevanten Produktmerkmale, die in den einzelnen Produktlebensphasen entstehen, auf der Basis einer einheitlichen, allgemeinen, lebensphasenübergreifenden und redundanzfreien Grundstruktur.

Das integrierte Produktmodell gliedert sich in eine Anzahl von Partialmodellen. Die Partialmodelle stellen Spezialisierungen der gemeinsamen Grundstruktur zur Unterscheidung einzelner Lebenslaufphasen dar. Alle Partialmodelle erhalten durch die gemeinsame Grundstruktur eine konsistente Basis. Im Unterschied zu den einzelnen jeweils voneinander unabhängigen Produktmodellen erfolgt die Spezialisierung der Partialmodelle für einzelne Produktlebensphasen nach einer einheitlichen und für alle Partialmodelle gültigen Methode. Diese stellt die Kohärenz aller Partialmodelle im integrierten Produktmodell sicher.

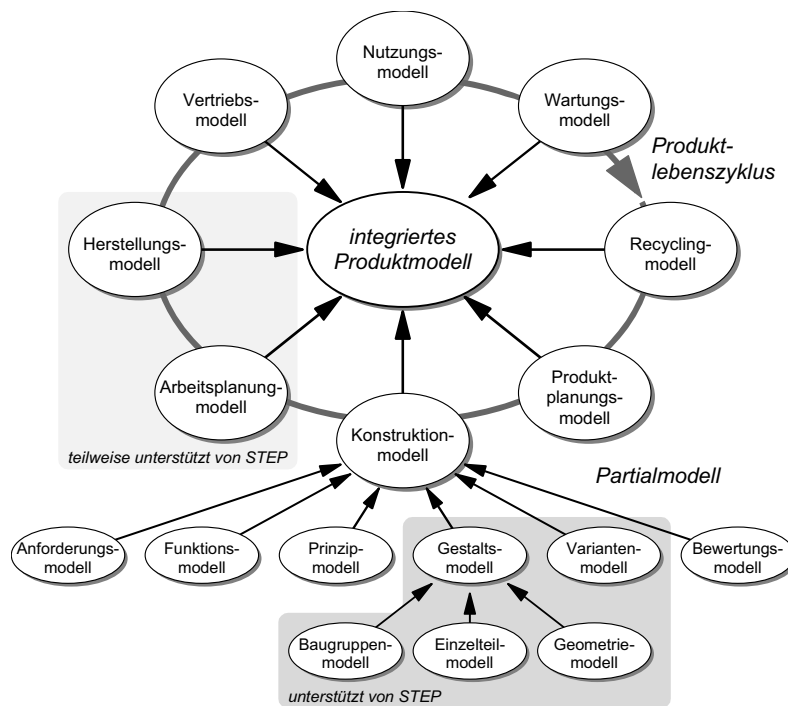


Bild (B007dprZ) Partialmodelle des integrierten Produktmodells

Das integrierte Produktmodell unterscheidet prinzipiell drei Klassen von Produktdaten:

- Die *Produktdefinition* beinhaltet alle administrativen und organisatorischen Daten, wie z.B. die Sachnummer, die Version, den Status, die Zugriffsrechte etc.. Man spricht in diesem Zusammenhang auch von den Produktmetadaten, welchen hinsichtlich des Produktdatenmanagements eine wichtige Bedeutung zukommt.
- Die *Produktrepräsentation* beschreibt die verschiedenen Aspekte eines Produktes, wie zum Beispiel die Produktstruktur, die Gestalt oder die Toleranzen. Dies geschieht innerhalb einzelner Partialmodelle des integrierten Produktmodells, welche im Hinblick auf den jeweiligen Anwendungsaspekt entwickelt wurden. Derzeit steht in diesem Zusammenhang vor allem die Repräsentation der geometrischen Gestalt im Mittelpunkt. Zukünftig sollen weitere Aspekte eines Produktes, wie zum Beispiel die Anforderungen, die Funktion oder die Wirkprinzipien definiert werden können. Dies ist jedoch noch Gegenstand der Forschung. Man erhofft sich davon, schon von Anbeginn einer Produktentwicklung CAX-Tools einsetzen zu können.

Änderungen an einer Produktrepräsentation resultieren in eine Änderung des Produktes selbst und haben eine direkte Auswirkung auf die hiervon betroffenen weiteren Partialmodelle, d.h. die Änderungen sind in diesen sofort sichtbar. Man spricht in diesem Zusammenhang von einer *direkten Assoziativität*.

- Bei *Produktpräsentationen* handelt es sich um verschiedene abgeleitete Sichten des Produktes, wie zum Beispiel die Technische Zeichnung, eine Stückliste oder das VRML-Modell eines Produktes.

Änderungen in einer Produktpräsentation haben keine Auswirkungen auf das Produkt, d.h. sie sind nicht in anderen Präsentationen sichtbar. Dagegen wirken sich Änderungen an einer Produktrepräsentation auf die abgeleiteten Präsentationen aus. Charakteristisch für Produktpräsentationen ist somit die sogenannte *indirekte Assoziativität*.

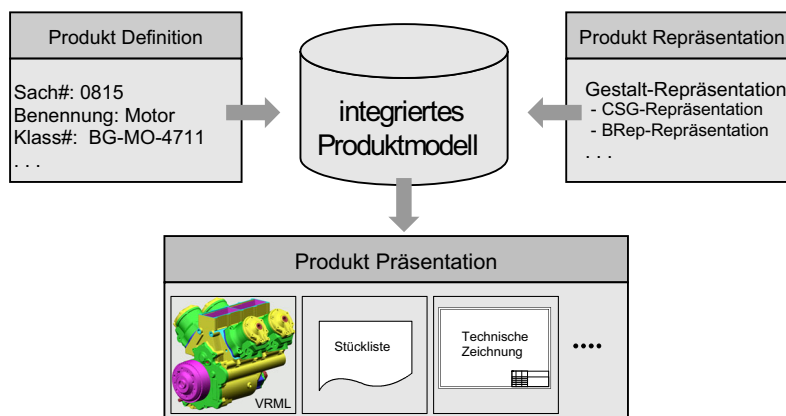


Bild (B008dprZ) Produktdefinition, -repräsentation und -präsentation

Die einzelnen, hinsichtlich des jeweiligen Produkaspektes spezialisierten CAx-Systeme (z. B. CAD für die Gestaltsmodellierung) dienen dabei als Erzeugersysteme, mittels welchen das integrierte Produktmodell mit den einzelnen Partialmodellen schrittweise aufgebaut und detailliert wird. Innerhalb des gesamten Produktlebenszyklus wird dieses integrierte Produktmodell Schritt für Schritt aufgebaut. Der wesentliche Unterschied zu dem oben beschriebenen Ansatz der Modelltransformation besteht darin, dass alle in die Produktentwicklung involvierten Unternehmensbereiche auf demselben rechnerinternen Modell arbeiten und dieses schrittweise um weitere Informationsinhalte ergänzen. Die benötigten Arbeitsdokumente wie

z. B. Stücklisten, Technische Zeichnungen, Montagepläne etc., werden als Sichten (Präsentationen) von diesem rechnerinternen Modell.

Die organisatorische Koordination der Erzeugersysteme bzw. des integrierten Produktmodells (Abläufe, Zugriffsrechte, Versionen etc.) ist Aufgabe der Verwaltungssysteme, welche entsprechend auf den produktdefinierenden Daten arbeiten. Als Verwaltungssysteme sind beispielsweise PLM- oder PPS-Systeme zu nennen.

PLM-Systeme zielen dabei auf Funktionen ab, welche vorwiegend in den frühen Phasen des Produktlebenszyklus (Freigabe-, Änderungs-wesen, Zugriffskontrolle, Revisionierung, Kommunikation etc.) benötigt werden und übernehmen teilweise Funktionen der Erzeugersysteme. Beispielsweise können Produktstrukturen ebenfalls mit PLM-Systemen aufgebaut bzw. manipuliert werden. Dies geschieht dadurch, dass ein Ober-Unterordnungsverhältnis der produktdefinierenden Daten einzelner Produkte definiert wird. Die geometrische Modellierung der PLM-seitig festgelegten Produktstruktur muss in einem CAD-System erfolgen.

Auf diese Weise kann entsprechend einer Top-Down-Vorgehensweise (vom Gesamtprodukt zum Einzelteil) die Produktstruktur auf oberster Ebene zu Beginn einer Produktentwicklung definiert werden. Mit CAD-Systemen wird diese im folgenden detailliert sowie einzelne Baugruppen im Bottom-Up-Verfahren (vom Einzelteil zur Baugruppe) erstellt und in die Struktur integriert.

Die oftmals vorhandenen Überlappungen der Funktionalität einzelner Systeme und das damit verbundene mehrfach parallele Erzeugen von Produktdaten gleichen Inhalts aber verschiedener Repräsentation (Formate) verursacht damit keine Probleme mehr. Vielmehr können den Anwendern in den einzelnen Unternehmensbereichen Funktionen entsprechend ihrer Bedürfnisse bereitgestellt werden.

Die Produktdatenarchivierung bezweckt den Export definierter Teilmengen des Produktmodells auf speziell hierfür ausgelegte Archivierungsformate (Datenkomprimierung etc.), Klassifizierungsmerkmale und Zugriffsschnittstellen. Für die Produktdatenarchivierung kommen spezielle Backup-Tools sowie Utilities von PLM-Systemen zum Einsatz.

Der Austausch von Produktdaten erfolgt mittels speziellen Import/Export-Monitoren. Innerhalb dieser kann entsprechend der vorliegenden Modellinhalte der geeignete Schnittstellenprozessor ausgewählt sowie direkt die Übertragungsqualität kontrolliert werden. Heutzutage ist diese Funktionalität in der Regel Bestandteil von Erzeuger- und Verwaltungssystemen. Diese besitzen in der Regel Schnittstellenprozessoren für die von ihnen behandelten Modellinhalte.

Zum heutigen Zeitpunkt haben CAX-Systemanbieter spezifische Produktmodellformate definiert und ihre einzelnen CAX-Komponenten darauf basierend integriert. Auf diese Weise können die Vorteile eines integrierten Produktmodells genutzt werden.

Derzeit beschränkt sich dies auf die CAD-, CAM-, FEM- sowie Verwaltungsfunktionen und damit vornehmlich auf die Produktentwicklung. Daten aus Fremdsystemen können nicht direkt in ein Produktmodell integriert werden, sondern müssen über entsprechende Schnittstellen importiert werden.

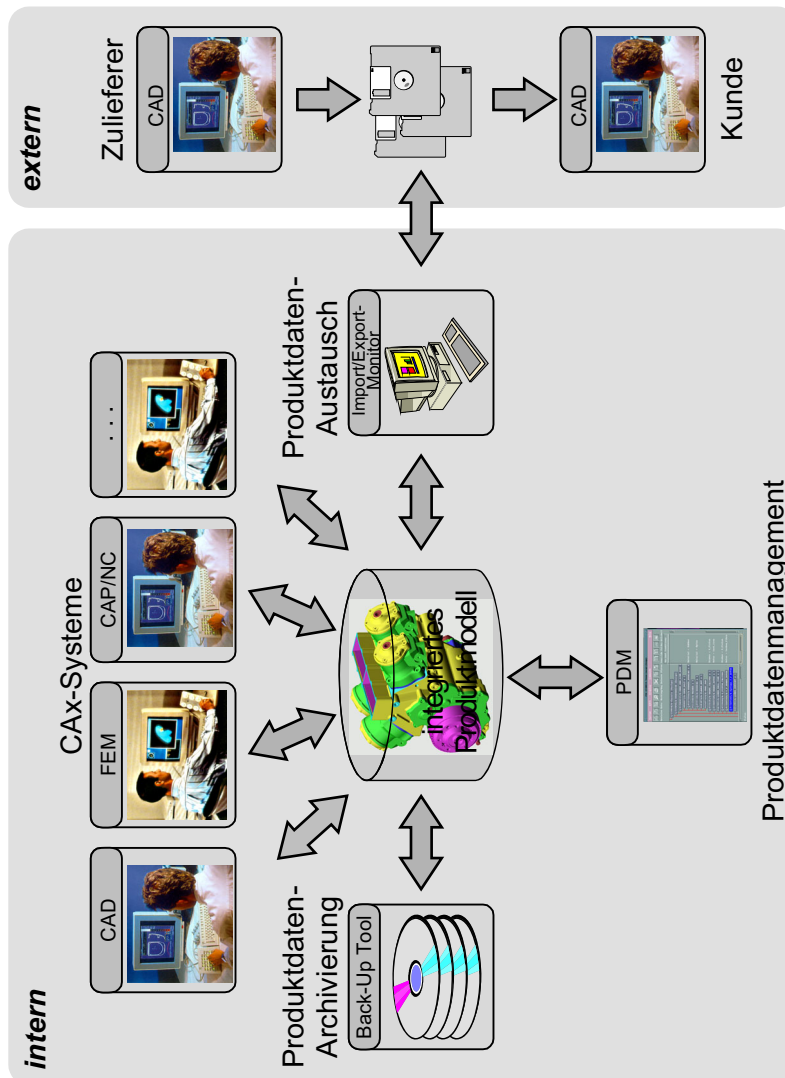


Bild (B009dprZ) Das integrierte Produktmodell

Von grosser Bedeutung sind deshalb Standardisierungsbemühungen für das integrierte Produktmodell. Die Zielsetzung besteht darin, dieses integrierte Produktmodell zu standardisieren und diese Spezifikation in Form von genormten Dokumenten allen CAx-, PLM- sowie PPS-Systemanbietern zugänglich zu machen. Damit müssen die Erzeuger- Management- oder Archivierungssysteme nicht von einem Anbieter stammen, sondern es können alle jene Systeme verwendet werden, welche dieses genormte Produktmodellformat unterstützen.

Die Entwicklung und Standardisierung des integrierten Produktmodells erfolgt derzeit innerhalb der ISO unter der Bezeichnung „ISO 10303 Product Data Representation and Exchange“ und wurde bekannt unter dem Namen STEP (Standard for the Exchange of Product Model Data).

Auch wenn zur Zeit lediglich Programme zur Transformation (Prozessoren genannt) systemspezifischer Datenstrukturen in das integrierte Produktmodell und umgekehrt implementiert werden, ist diese Entwicklung für den zukünftigen Einsatz moderner Informationstechnologien in der Produktentwicklung von grosser Bedeutung. Die heute vorherrschende starke Trennung von CAx-Tools, PLM (Produktdatenmanagement) und PPS (Produktionsplanung und -steuerung) wird sich mit der fortschreitenden Entwicklung, Standardisierung und der Implementierung durch die zahlreichen Anbieter zunehmend auflösen.

### 3.3 Einordnung des integrierten Produktmodells in das Digitale Produkt

Der Fokus der oben beschriebenen Produktdatenintegration im allgemeinen sowie des integrierten Produktmodells im speziellen liegt innerhalb der industriellen Anwendung sowie den kommerziell erhältlichen CAx-/PLM Tools vor allem in der Integration der verschiedenen innerhalb der Entwicklung und Herstellung zur Anwendung kommenden CAx-Systeme. Die Produktdatenintegration bzw. das integrierte Produktmodell ist zentraler Bestandteil des Digitalen Produktes. Innerhalb des Digitalen Produktes werden für die Integration der weiteren Produktlebensphasen und vor allem in Hinblick auf die darin zur Anwendung kommenden Unternehmensprozesse um weitere Informationsinhalte erweitert.

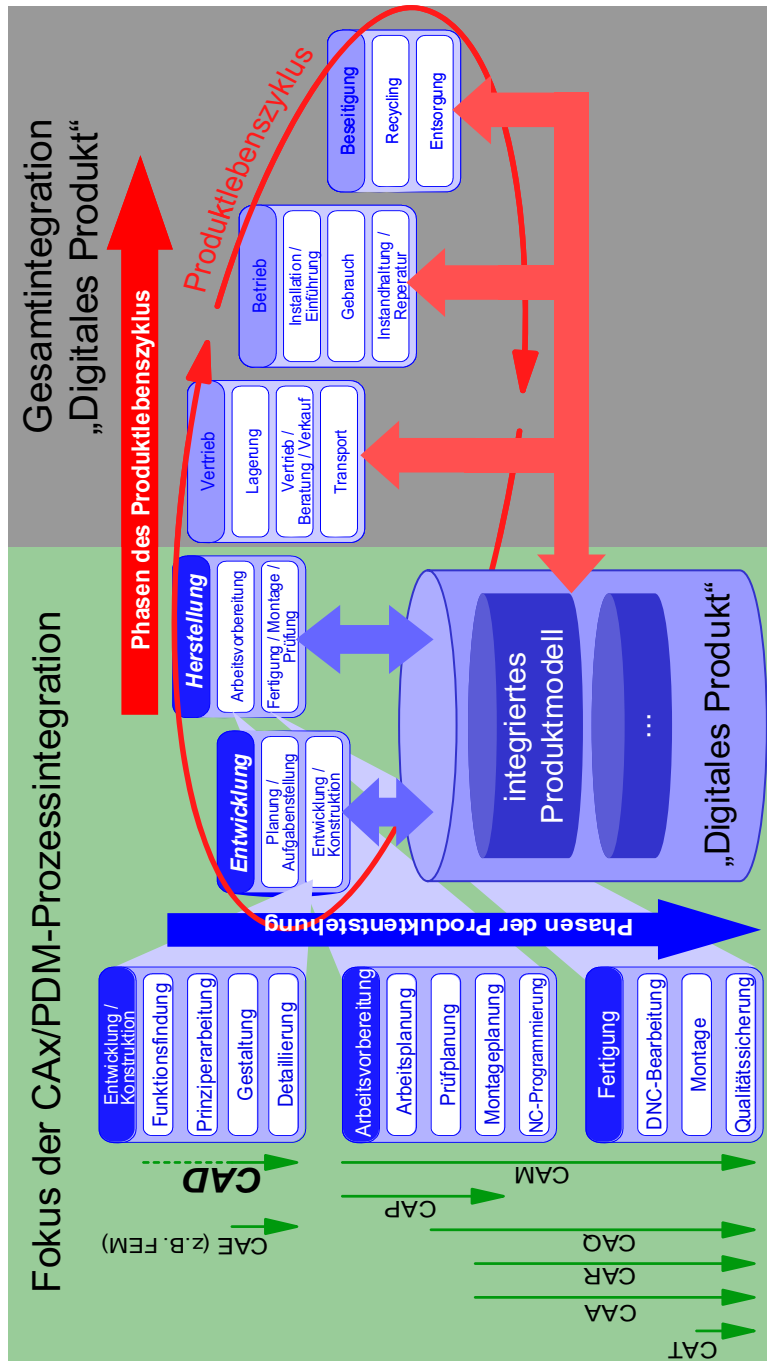


Bild (B010dprZ) Einordnung des integrierte Produktmodells in das Digitale Produkt



Beispiel aus der Praxis

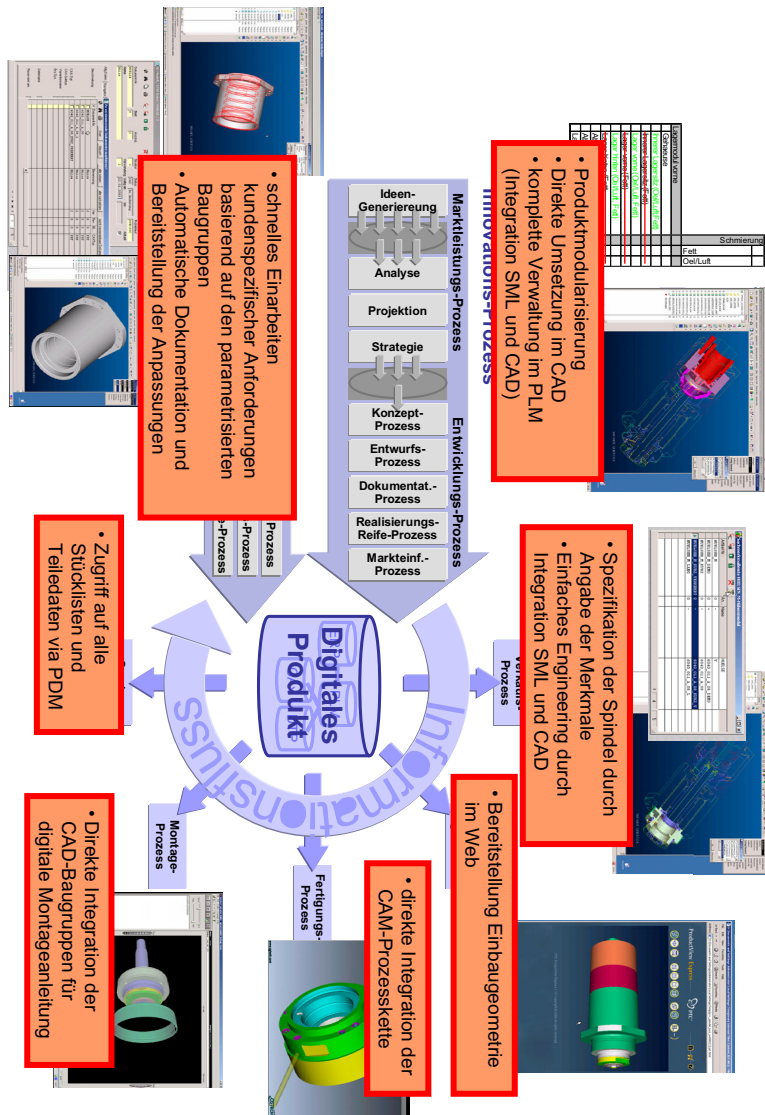


Bild (B011dprZ) Beispiel aus der Praxis

## 4 Zusammenfassung

Das *Digitale Produkt* ist definiert durch die Summe der Daten, welche in der Entwicklung erzeugt und über den gesamten Lebenszyklus angepasst und ergänzt werden. Die Nutzung dieser Daten erfolgt mittels spezifischer, IT-basierter Anwendungsdienste im Rahmen von wertschöpfenden Unternehmensprozessen.

Grundlage für das Konzept des Digitalen Produktes ist das Vorhandensein einer Produktplattform. Sie setzt sich aus einer Menge von Modulen zusammen, welche zu kunden- oder auftragsspezifischen Produkten konfiguriert werden können. Unternehmensprozesse sind wertschöpfende Tätigkeiten eines Unternehmens, welche gezielt die Informationen rund um das Digitale Produkt nutzen. Diese Nutzung erfolgt mittel IT-Tools. Produktplattform, Unternehmensprozesse und IT-Tools müssen aufeinander abgestimmt sein, um einen maximalen Nutzen aus dem Konzept des Digitalen Produktes zu ziehen.

*Rechnerintegrierte Produktentwicklung* ist die Bereitstellung und Verarbeitung von Produktdaten in digitaler Form. Produktdaten sind Informationen über ein Produkt. Sie fallen über den gesamten Lebenszyklus an.

Der Trend der Digitalisierung hält an. Die Bearbeitung und Aufbereitung von Information in eine Form, in der sie sich gezielt, schnell und standortunabhängig nutzen lässt, schafft Wettbewerbsvorteile.

### Verständnisfrage 1

Was sind die massgeblichen Trend der Gegenwart innerhalb einem modernen Industrieunternehmen

### Verständnisfrage 2

Worin liegen die zukünftigen Erfolgspotentiale für ein Unternehmen?

### Verständnisfrage 3

Was versteht man unter dem Digitalen Produkt – wie ist dieses definiert?

### Verständnisfrage 4

Was versteht man unter PUT – was ist daran bedeutend für ein Industrieunternehmen?

#### Verständnisfrage 5

Was versteht man unter der Produktplattform, den Unternehmensprozessen und den strategischen Tools innerhalb des Digitalen Produktes?

#### Verständnisfrage 6

Was versteht man im allgemeinen unter der rechnerintegrierten Produktentwicklung?

#### Verständnisfrage 7

Was für Integrationsgrade werden unterschieden?

#### Verständnisfrage 8

Was versteht man unter einer Modelltransformation, wodurch ist diese Charakterisiert?

#### Verständnisfrage 9

Worin liegen die grundsätzlichen Probleme der Modelltransformation?

#### Verständnisfrage 10

Wodurch ist die Modellintegration charakterisiert?

#### Verständnisfrage 11

Was versteht man unter einem integrierten Produktmodell?

#### Verständnisfrage 12

Welche drei Klassen von Produktmodellldaten werden unterschieden (Beispiele)?

#### Verständnisfrage 13

Was versteht man unter dem Begriff direkte Assoziativität

#### Verständnisfrage 14

Wo kommt diese innerhalb des integrierten Produktmodells zu tragen?

#### Verständnisfrage 15

Was versteht man unter dem Begriff indirekte Assoziativität?

**Verständnisfrage 16**

Worin besteht der Unterschied zur direkten Assoziativität?

**Verständnisfrage 17**

Warum ist es sinnvoll das integrierte Produktmodell zu standardisieren?

**Verständnisfrage 18**

Worin liegen die Rationalisierungspotentiale?

**Verständnisfrage 19**

Was sind die wesentlichen Anforderungen an das integrierte Produktmodell?

**Verständnisfrage 20**

Was ist notwendige Voraussetzung für die Definition des integrierten Produktmodells?

**Verständnisfrage 21**

Warum gliedert sich das integrierte Produktmodell in einzelne Partialmodelle?

**Verständnisfrage 22**

In welcher Organisation und unter welchem Namen wird das integrierte Produktmodell definiert und standardisiert?

**Relevante Cases**

- [Mikron Agno](#)

**Publikationsverzeichnis – Literatur**

- [EHS-91] *Eigner, M.; Hiller, C.; Schindewolf, S.; Schmich, M.*: Engineering Database: strategische Komponente in CIM-Konzepten. München: Hanser Verlag 1991
- [GAP-93] *Grabowski, H.; Anderl, R.; Polly, A.*: Integriertes Produktmodell. Berlin: Beuth verlag 1993
- [PaB-93] *Pahl, G.; Beitz, W.*: *Konstruktionslehre: Methoden und Anwendung*. 3. Aufl., Berlin: Springer Verlag 1993
- [Wie-89] *Wiendahl, H.-P.*: Betriebsorganisation für Ingenieure. 3. Auflage, München: Hanser Verlag 1989



# Einführung in die PLM-Technologie

Autor: Dr. R. Montau



# 1 Überblick

„Information darf nicht als Rohstoff sondern muss als Werkstoff vorliegen“. Diese Forderung verdeutlicht die zentrale Bedeutung der *Information*, die zunehmend in immer mehr Bereiche unserer heutigen Leistungs- und Technologiesgesellschaft vordringt. Für die Produktentwicklung ergibt sich aus den rasanten Weiterentwicklungen der Informationstechnologien heute in zweifacher Hinsicht ein Handlungsbedarf:

- Einerseits wird es mit der heutigen Rechnerunterstützung durch CAx-Technologien immer einfacher, unüberschaubare Datenmengen in nahezu endlos scheinender Variationsvielfalt zu erzeugen, ohne jedoch eine adäquate Verwaltung für eine spätere Nutzung sicherzustellen → Verwaltungsaspekt).
- Andererseits erfordert der zunehmende Wettbewerbsdruck im Markt immer neue und wirksamere Rationalisierungsmassnahmen zur Produktivitätssteigerung, welche nur durch den effizienten Einsatz modernster Informationstechnologien erreicht werden können → Prozessoptimierungsaspekt).

In der industriellen Praxis sind derzeit weder der Verwaltungsaspekt noch der Prozessoptimierungsaspekt zufriedenstellend gelöst. „Fast alle Unternehmen kämpfen heute mit einem Datenüberfluss und leiden gleichzeitig unter Informationsmangel!“

Nachfolgend sollen die Ursachen dieser Diskrepanzen sowie die Notwendigkeit und Lösungsansätze zur Abhilfe anhand folgender Aspekte aufgezeigt werden:

- historische Entwicklung,
- Rationalisierungsdruck durch Verschärfung der Marktsituation,
- Produkt Lifecycle Management - Ansatz.



## 2 Historische Entwicklung und Problemstellung

Charakteristisch für die Produktentwicklung ist heute nach wie vor die Aufteilung technischer Funktionen in kleinere, weniger komplexe Unterfunktionen und deren arbeitsteilige Bearbeitung durch eine Menge von Mitarbeitern. Die Ursache und Notwendigkeit hierfür liegt in der starken Spezialisierung und der zunehmenden Komplexität technischer Aufgabenbereiche begründet. Die tayloristischen Prinzipien der Arbeitsteilung und des Spezialistentums ermöglichen zwar eine deutlich höhere Effizienz und Wirtschaftlichkeit, haben jedoch auch eine hochgradig funktional untergliederte Aufbauorganisation zur Folge (vgl. Bild B001plmZ).

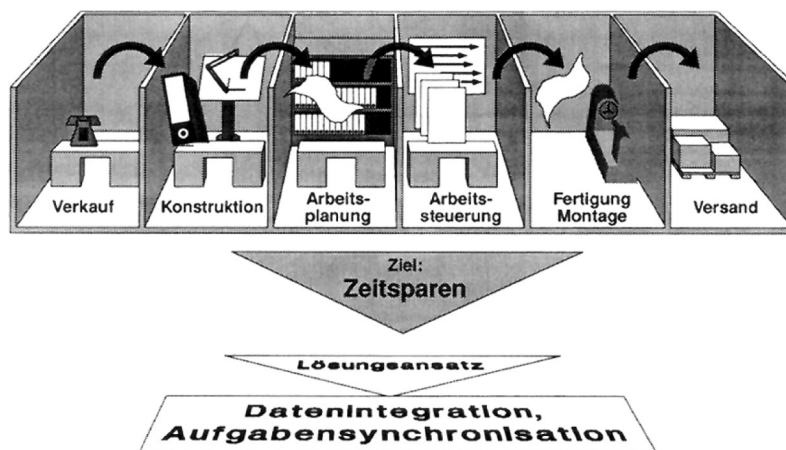


Bild (B001plmZ) Sequentieller Informationsfluss bei konventioneller Auftragsabwicklung [[VDI-91]]

Um die Tätigkeiten innerhalb der einzelnen Bereiche zu unterstützen, wurden jeweils spezifische Computersysteme entwickelt, die nach und nach auf die vorliegenden Bedürfnisse optimiert wurden (z.B. CAD-Systeme in der Konstruktion). Aufgrund dieser historisch getrennten Entwicklung sogenannter Insellösungen ist deren Zusammenspiel im unternehmerischen Gesamtzusammenhang leider keinesfalls optimal. Aufgrund der strikten Bereichstrennung pflegt jeder Funktionsbereich seinen eigenen Datenbestand und verwendet eine Kopie der Ergebnisse des vorhergehenden Bereichs als Eingabedaten. Nachteilig hieran ist nicht nur die wiederholte Grunddatengenerierung wegen möglicher Eingabefehler und Redundanzen, sondern viel-

mehr der Zwang zu einer rein sequentiellen Arbeitsfolge und einem ebensolchen Informationsfluss.

Zeiteinsparungen im Produktentstehungsprozess erfordern jedoch eine Parallelisierung von Tätigkeiten, die nur ausgehend von einer *Datenintegration* möglich ist. Ohne Datenintegration ist die anfangs geforderte Verfügbarkeit von Information nicht zu gewährleisten, da im Produktentwicklungsprozess vielfältige Fragestellungen mit oftmals bereichsübergreifendem Charakter auftauchen, die mit herkömmlichen Insellösungen nicht hinreichend beantwortet werden können.



Bild (B002plmZ) Alltägliche Fragestellungen im Rahmen einer Produktentwicklung

### 3 Rationalisierungsdruck durch die Verschärfung der Marktsituation

In der heutigen Zeit sind Unternehmen wie oben bereits beschrieben zahlreichen Einflussfaktoren unterworfen, die eine Beseitigung der vorstehend erläuterten archaischen Verhältnisse unabdingbar machen, um im Markt bestehen zu können, wie etwa:

- zunehmender in- und ausländischer Wettbewerb mit immer grösser werdendem Termin- und Kostendruck,
- Verkürzung der Produktlebenszyklen,
- Wandel vom Anbieter- zum Kundenmarkt, was eine grössere Flexibilität erfordert,
- zunehmende Komplexität der Produkte und deren Dokumentation,
- höhere Qualitätsanforderungen durch schärfere Produkthaftungsrichtlinien (ISO/EN/DIN 9000 bis 9004),
- Zunahme standortübergreifender Unternehmen und verteilter Produktentwicklungen sowie Globalisierung der Märkte.

Ursache und Wirkung der verschiedenen Einflussfaktoren sollen nachfolgend jeweils kurz erläutert werden.

Analysen nach den Ursachen für Gewinnschmälerungen haben ergeben, dass die Produkthanlaufzeit (time to market) der wesentlichste Wettbewerbsfaktor ist (vgl. Bild B003plmZ). Da die produktdefinierenden Bereiche (Konstruktion, Arbeitsvorbereitung) etwa die Hälfte einer Gesamtdurchlaufzeit in Anspruch nehmen, ist eine Beschleunigung bzw. Optimierung dieser Tätigkeiten von besonderem Interesse. Oftmals ist die Lieferzeit sogar der allein ausschlaggebende Faktor zur Auftragserteilung.

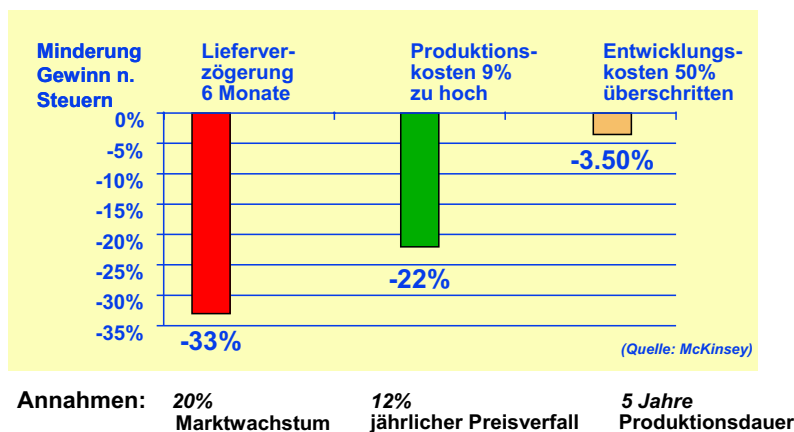


Bild (B003plmZ) Ursachenanalyse für Gewinnschmälerungen

Hinzu kommt, dass die Konstruktion gegenüber den übrigen produktionstechnischen Bereichen auch hinsichtlich der Kosten eine Schlüsselrolle einnimmt (vgl. Bild B004plmZ). Veranschaulicht wird dies durch die Diskrepanz zwischen der Kostenfestlegung und deren Verursachung. In der Konstruktion und Entwicklung werden bereits über 80% der späteren Produktkosten festgelegt, da die Produkteigenschaften hier fast vollständig definiert werden. Demgegenüber fällt auf, dass die verursachten Kosten lediglich etwa 10% betragen, was aufgrund der grossen Bedeutung dieses Bereichs erstaunlich ist.

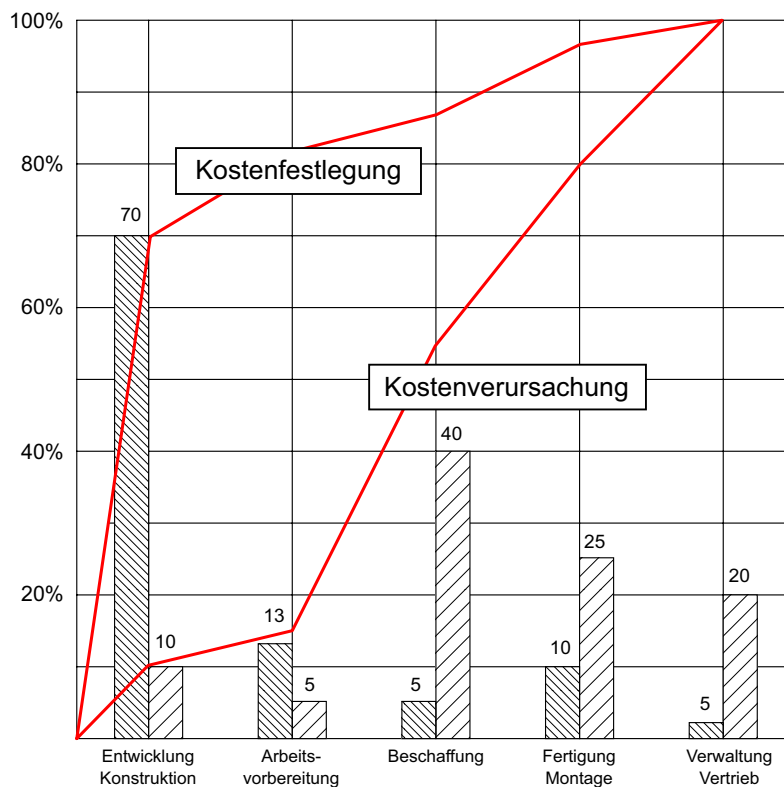


Bild (B004plmZ) Diskrepanz zwischen Kostenfestlegung und -verursachung [VDI2234, VDI-91]

Aufgrund der herausragenden Bedeutung des Konstruktionsbereichs muss die dort vorherrschende Aufgabenverteilung einer genaueren Beachtung unterzogen werden. Da die Häufigkeit von Neukonstruktionen lediglich bei etwa 25% liegt [PaB-93], können in der überwiegenden Mehrzahl der Fälle bereits existierende Konstruktionslösungen wiederverwendet werden. Somit müssen nur bei Neukonstruktionen sämtliche Konstruktionsphasen durchlaufen werden, bei allen übrigen

Konstruktionsarten können erhebliche Zeiteinsparungen erzielt werden, falls dem Konstrukteur zweckmässige Suchfunktionalität und Variantentechniken bereitgestellt werden. Die effiziente Unterstützung dieser Prozesse und insbesondere die Nutzung des grossen Wiederverwendungspotentials durch den zweckmässigen Einsatz von Informationstechnologien sind entscheidend zur Erreichung der heute geforderten Flexibilität in immer kürzer werdender Zeit.

		Konstruktionsarten			
		Neu-konstruktion (25%)	Anpassungs-konstruktion (55%)	Varianten-konstruktion (15%)	Konstruktion mit festem Prinzip (5%)
Konstruktionsphasen	Aufgabenklärung	■	■	■	■
	Funktionsfindung				
	Prinzip-erarbeitung		■		
	Gestaltung			■	
	Detaillierung		■		■

Bild (B005plmZ) Zuordnung von Konstruktionsarten zu Konstruktionsphasen basierend auf [VDI2210]

Im Hinblick auf die Produktlebenszyklen ist bereits seit mehreren Jahren eine stetige Verringerung zu beobachten. Da die Lebensdauer eines Produktes am Markt immer kürzer wird, verringert sich auch der für innovative Produktentwicklungen zur Verfügung stehende Zeitraum. Infolgedessen müssen Produktentwicklungen heute immer schneller und effizienter vorangetrieben werden, was durch die Zunahme kurzfristiger Trendbewegungen noch verstärkt wird.

Der oben beschriebene Wandel vom Anbieter- zum Kundenmarkt hat zur Folge, dass Produkte immer weniger als einheitliche Lösungen angeboten werden können und ein Anbieter immer häufiger auf Sonderwünsche von Kunden eingehen muss. Daraus folgt ein Zwang zu grösserer Flexibilität, die sich insbesondere durch Variantentechniken erreichen lässt, wofür jedoch eine leistungsfähige Systemunterstützung erforderlich ist.

Eine Konsequenz des Technologiefortschritts ist die zunehmende Komplexität der Produkte und deren Dokumentation sowie der zugehörigen Entwicklungsprozesse. In Anlehnung an den bereits proklamierten Verwaltungsaspekt muss hierfür naturgemäss ebenso ein

grösserer Aufwand betrieben werden bzw. leistungsfähigere Verwaltungsmethoden zur Anwendung kommen.

In juristischer Hinsicht hat sich die Produkthaftung seit einigen Jahren grundlegend geändert. Durch die Einführung der EG-Richtlinie 85/374 und verschiedener nationaler Normen wurde die bislang angewendete vertragliche Haftung gegenüber dem Käufer in eine deliktische Haftung umgewandelt, die verschuldensunabhängig auch gegenüber Nichtvertragspartnern Gültigkeit besitzt. Die Beweislast liegt nun ausschliesslich beim Hersteller, der den Konstruktionsstand und sämtliche Änderungsstände des ausgelieferten Produkts mindestens für 10 Jahre (in manchen Branchen 30-50 Jahre) rückverfolgbar halten muss, um nötigenfalls Entlastungsbeweise führen zu können (vgl. Bild B006plmZ). Dies schliesst die eindeutige Zuordnung zu sämtlichen Zeichnungen, Spezifikationen oder anderen Dokumenten des Produktes mit ein. In den Qualitätsrichtlinien ISO/EN/DIN 9000 bis 9004 finden sich diesbezüglich umfassende Vorgaben, die systemtechnisch in geeigneter Weise unterstützt werden müssen.

**Produkthaftung:** Haftungsrecht infolge von Mängeln eines Produktes für den Hersteller, dessen Zulieferer sowie den nationalen und internationalen Handel.

**Seit Einführung der EG-Produkthaftungsrichtlinie 85/374:**

wird nicht nur **vertraglich** gegenüber dem Käufer, sondern auch **deliktisch** gegenüber Nichtvertragspartnern gehaftet,

wird nicht nur **verschuldensabhängig** sondern auch **verschuldensunabhängig** gehaftet, wodurch die Beweislast zugunsten des Verbrauchers auf den Hersteller, den Zulieferer und den Handel übergeht,

müssen der Konstruktionsstand und die dazugehörigen Änderungszustände **bis mindestens 10 Jahre** nach Auslieferung eines Produktes noch **zurückverfolgbar** sein, um Entlastungsbeweise führen zu können.

Bild (B006plmZ) Verschärfung der Produkthaftung

Ein wesentlicher Aspekt der vergangenen Jahre ist die erwähnte Globalisierung der Märkte, die Zunahme verteilter Unternehmensstrukturen und der Trend zu sogenannten virtuellen Fabriken, worunter dynamische Zusammenschlüsse verschiedener Unternehmen entlang einer Produktentwicklungsprozesskette zu verstehen sind. Immer häufiger ergibt sich heute insbesondere in High-Tech-Bereichen (Luft- und Raumfahrtindustrie, Automobilbau etc.) die Notwendigkeit, dass örtlich getrennte Standorte Produktdaten digital miteinander austauschen, um Zeit zu sparen (vgl. in [Mar-97] digitale Übermittlung von

3D-Produktmodellldaten anstelle Technischer Zeichnungen, um von einem Fremdunternehmen schnellstmöglich NC-Werkzeugverfahrwege für die eigenerstellten CAD-Modelle ableiten zu lassen). Bild B007plmZ verdeutlicht diese Thematik anhand eines fiktiven Szenarios mit zwei kooperierenden Entwicklungsstandorten, einem Zulieferer und einem Kunden auf jeweils unterschiedlichen Kontinenten. In bezug auf die Netzwerkverbindungen muss dabei beachtet werden, dass nicht immer für alle Standorte Direktverbindungen über ein WAN (Wide Area Network) möglich sind (vgl. Zulieferer in Bild B007plmZ). Zumindest kann jedoch nicht von einer 100%-igen Verfügbarkeit des WAN's ausgegangen werden, wodurch verteilte Informationssysteme eine anspruchsvolle technische Herausforderung darstellen.

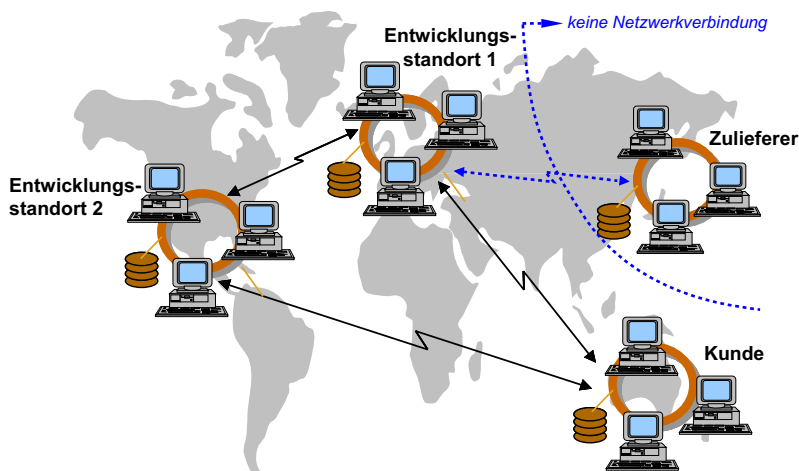


Bild (B007plmZ) Beispiel einer standortübergreifenden Produktentwicklung

Eine zunehmende Bedeutung kommt in diesem Zusammenhang der direkten Integration von Kunden und Lieferanten zu. Viele Kunden sind nicht mehr mit reinen Massenerzeugnissen zufrieden, sondern fordern eine kundenspezifische Anpassung der Produkte und eine schnellere Entwicklung innovativer Produkte, massgeschneidert auf die jeweils vorliegenden Anforderungen und Bedürfnisse. In der selben Weise wird von Lieferanten gefordert, nicht nur das Produkt als solches zu liefern, sondern vielmehr auch alle benötigten Produktdaten zur Verfügung zu stellen, um diese innerhalb den eigenen Prozessketten direkt weiterverwenden zu können.

Die Anbieter – Kunden – Beziehungen sind dabei in der Regel von kurzer Dauer. Dementsprechend muss die geforderte Zusammenarbeit und der notwendige Datenaustausch innert kürzester Frist aufgebaut

und durch entsprechende Tools unterstützt werden. Eine zentrale Bedeutung kommt in diesem Zusammenhang dem Internet zu.

Um diesen Anforderungen gerecht zu werden, sollen unter dem Stichwort Produkt Lifecycle Management „PLM“ verschiedene Daten- und Anwendungsfunktionen zu einem Gesamtsystem zusammengeführt werden. Das bedeutet, dass die verschiedenen innerhalb eines Unternehmens zur Anwendung kommenden Applikationen wie zum Beispiel das Enterprise Resource Planning (ERP), das Customer Relationship Management (CRM), das Produkt Lifecycle Management (PLM) oder auch das CAD untereinander verbunden und durch Web-Browser basierende, standardisierte Benutzeroberflächen zugänglich gemacht werden. Zielsetzung ist es nicht mehr, Produktdaten an alle Personen übertragen zu müssen, die aktiv am Produktlebenszyklus beteiligt sind (früheres Push-Prinzip der Dokumentverteilung). Vielmehr soll standortunabhängig über das Internet auf einfache Art und Weise auf das digitale Produkt zugegriffen werden können (Pull Prinzip, immer aktueller Datenbestand).

Die vorstehend diskutierten Einflussfaktoren verdeutlichen die unabdingbare Notwendigkeit und strategische Bedeutung für den Einsatz moderner Informationstechnologien in der Produktentwicklung. Seit einigen Jahren werden diesbezüglich Produktdatenmanagement-Systeme (PDM) und neuerdings insbesondere Produkt Lifecycle Management Systeme (PLM) viel diskutiert, die den gesamten Produktlebenslauf umfassen, ausgehend von der Planung bis hin zur Produktentsorgung und somit einen ganzheitlichen Abbildungs- und Dokumentationsaspekt verfolgen.



## 4 Charakterisierung des PLM-Ansatzes

Betrachtet man den Informationsfluss zwischen den verschiedenen Bereichen eines Unternehmens, so ergibt sich eine Grobeinteilung in zwei unterschiedliche Informations- oder Prozessketten.

- Einerseits umfasst eine Produktentwicklung eine Reihe technischer Funktionen, die insbesondere durch verschiedene CAx-Systeme unterstützt werden, zwischen denen ein produktbezogener Informationsfluss vorliegt. Zielsetzung und Ergebnis hierbei ist eine digitale Produktdefinition.
- Andererseits sind in betriebswirtschaftlich-logistischer Hinsicht diverse Funktionen zur Produktionsplanung und -steuerung erforderlich, die durch ein Enterprise Resource Planning (ERP) System unterstützt werden und einen auftragsbezogenen Informationsfluss zwischen den einzelnen ERP-Modulen ergeben. Zielsetzung hierbei ist die Optimierung des Materialflusses und benötigter Ressourcen zur deterministischen Erzeugung von physikalischen Kopien einer Produktdefinition.

Für die Zusammenführung betrieblicher Datenbestände und Applikationen wurde unter dem Schlagwort *Computer Integrated Manufacturing (CIM)* über viele Jahre mit grossem Aufwand die Vision einer integrierten Gesamtlösung verfolgt. Mittlerweile hat sich herausgestellt, dass dieser Ansatz nicht praktikabel ist. Aufgrund der Komplexität und den unterschiedlichen, teilweise sogar gegenläufigen Anforderungen der Applikationen kann eine zentrale Datenbank mit einem unternehmensweiten Datenmodell keine zweckmässige Lösung darstellen.

Aus diesen Gründen werden heute fast ausschliesslich sogenannte *Repository*-Konzepte [BeD-94] über Integrationsdatenbanken diskutiert, die eine grössere Flexibilität besitzen. Im Gegensatz zur Vollintegration wird hierbei zwischen datenerzeugenden Systemen (z.B. CAD) und Integrationsdatenbank(en) unterschieden. Hierzu zählen auch die sogenannten *Produkt Lifecycle Management-Systeme* (PLM-Systeme), die sich bzgl. zwei Problemebenen eingesetzt werden können:

- Integration des technischen Innovationsprozesses zur digitalen Produktdefinition inkl. CAx als PLM-Backbone,
- bereichsübergreifende Verknüpfung mit der betriebswirtschaftlich-logistischen Prozesskette (PLM und ERP).

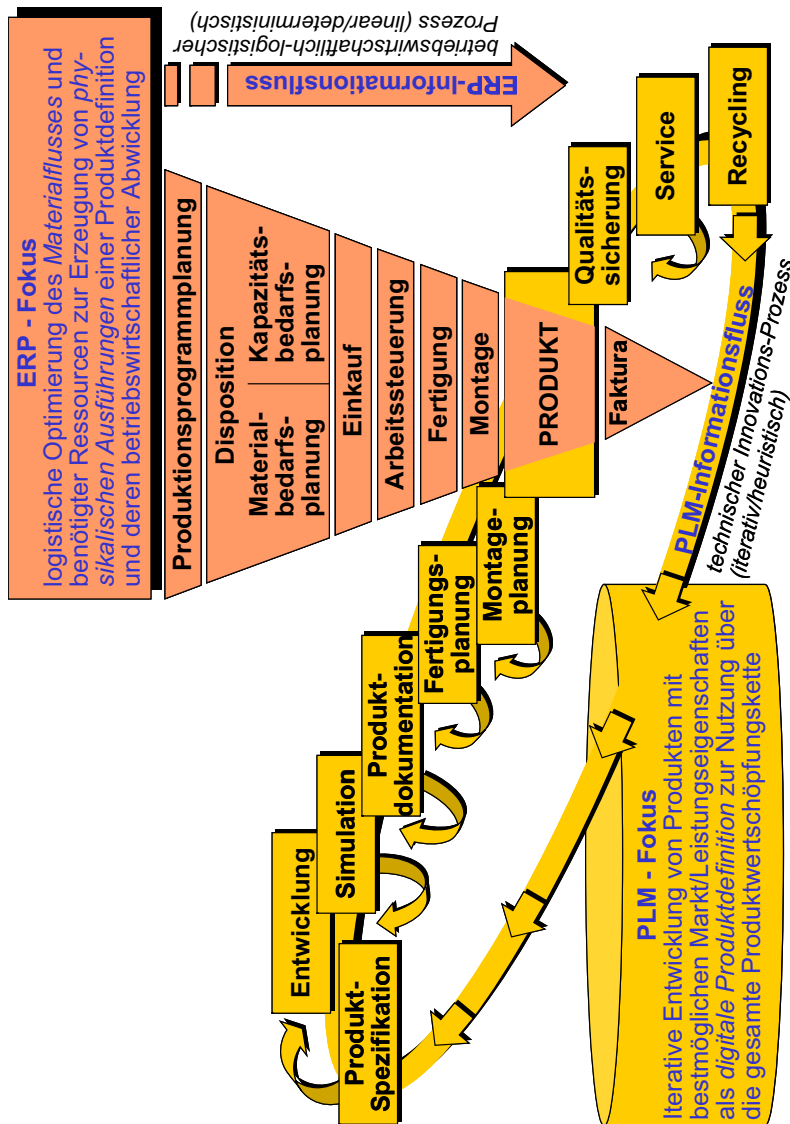


Bild (B008plmZ) Produkt Lifecycle Management-Ansatz

Primär ermöglichen PLM-Systeme eine integrative Unterstützung des Objekt- (statische Komponente) und Prozessmanagements (dynamische Komponente) entlang der technischen Prozesskette. Hinsichtlich der bereichsübergreifenden Verknüpfung zu ERP wird unter Betonung der Autonomie nur eine lose Kopplung angestrebt. Die PLM-Datenbank wird dadurch zur Integrationsdrehscheibe und Backbone für sämtliche technischen Produktdaten, wie z.B. CAD-Modelle, Stücklisten, Anforderungslisten, NC-Programme, technische Berichte, Projektdokumentationen etc...

Die Zielsetzungen des PLM-Einsatzes lassen sich wie folgt beschreiben:

- Reduktion der Produkthanlaufzeiten (time to market),
- Reduktion der verursachten und festgelegten Kosten,
- Steigerung der Flexibilität für Kundenwünsche,
- Absicherung gegen Produkthaftungsansprüche durch gewissenhafte Dokumentation,
- Unterstützung des Qualitätsmanagements nach DIN/EN/ISO 9000 bis 9004,
- bereichsübergreifende Zusammenführung betrieblicher Applikationen hinsichtlich Datenaustausch und -konsistenz,
- Vereinheitlichung des Produktdatenmanagements.

#### 4.1 Anwendungsfunktionen von PLM-Systemen

Hierfür stehen dem PLM-Anwender eine ganze Reihe von Funktionen zur Verfügung. In Anlehnung an die Darstellung in Bild B009plmZ soll im folgenden ein kurzer Überblick über die Funktionalität in den Bereichen Objekt-, Struktur- und Prozess-Management gegeben werden.

Im Rahmen des *Objektmanagements* können Objekte und deren Beziehungen zueinander abgebildet werden. Hierdurch lässt sich beispielsweise der Bezug zwischen einer Stückliste und einem CAD-Modell festhalten. Entsprechend der herkömmlichen Vorgehensweise mit Papierdokumenten, können zudem Projektmappen definiert werden, die zur Gruppierung verschiedener zusammengehörender Objekte dienen. Im Gegensatz zur gewöhnlichen Papiermappe, die zu einem Zeitpunkt entweder immer nur an einem Ort sein kann oder aber in mehrfacher Kopie existiert, lassen sich digitale Projektmappen (Folder) mehrfach referenzieren, ohne irgendwelche physische Duplikate zu erfordern. Somit kann ein Folder definiert werden, der für mehrere Mitarbeiter eines Entwicklungsteams verfügbar ist, um z.B. eine Anforderungsliste, einen Projektplan, Artikelstammdaten oder CAD-Modelle zu bearbeiten, wobei sämtliche Benutzer immer auf denselben aktuellen und konsistenten Datenbestand zugreifen.

Ein jeder Entwicklungsprozess ist von zahlreichen Entwicklungsschritten und Änderungen gezeichnet. Um hierbei die bestehenden Datenzustände dauerhaft festhalten zu können, anstelle diese zu überschreiben, wird mit Hilfe der *Revisionierung* eine Kopie des Objektes erstellt, auf welcher weitergearbeitet wird. Der alte Zustand kann somit jederzeit wieder abgerufen werden, was z.B. von der ISO9001 gefordert wird.



Bild (B009plmZ) PLM-Anwendungsfunktionen

Charakteristisch für PLM ist die Fähigkeit zur Handhabung von Fremdobjekten durch *die Integration externer Applikationen*. Diese können direkt aus PLM gestartet werden und speichern ihre Dateien jeweils in die PLM-Datenverwaltung in Form eines speziellen Container-Objekts (häufig Dataset genannt).

Weiterhin stehen umfangreiche Mechanismen zur Verfügung, um Produktdaten entsprechend den Produkthaftungsrichtlinien langfristig zu archivieren. Zur Absicherung gegen allfällige Systemausfälle stehen umfangreiche Backup- und Recovery-Möglichkeiten zur Verfügung.

Daneben dient die *Autorisierung* dem Definieren von Zugriffsprivilegien (Read, Write, Delete etc.), um Objekte gegenüber zugreifenden Benutzern, Gruppen oder Rollen (betriebliche Funktionen) entsprechend zu schützen.

Das *Strukturmanagement* kann als Pendant zur herkömmlichen Stücklistenverwaltung angesehen werden. Hiermit lassen sich auf sehr leistungsfähige und flexible Weise Objektstrukturen definieren, wobei insbesondere Baugruppenhierarchien, d.h. Produktstrukturen eine zentrale Bedeutung zukommt.

Zur Steigerung der Flexibilität dienen *Varianten* in Produktstrukturen, um dem Kunden ein Produkt in mehreren Ausführungsformen anbieten zu können, worunter dieser dann auswählen kann. Neben einer grösseren Flexibilität auf der Kundenseite ergeben sich hieraus auch Vereinfachungen für das innerbetriebliche Änderungswesen.

Mittels *Konfiguration* können Produktstrukturen auf flexible Weise abgefragt werden, was einen wesentlichen Unterschied zu statischen Papierdokumenten darstellt. Beispielsweise lässt sich die aktuell gültige, d.h. freigegebene Produktstruktur abfragen oder die zu einem bestimmten früheren Datum gültige etc..

Das *Prozessmanagement* stellt die dynamische Komponente im PLM dar und ermöglicht das Abbilden betrieblicher Arbeitsabläufe auf einen formalen, systemtechnisch gesteuerten Prozess. Von herausragender Bedeutung sind hierbei Freigabeprozesse sowie Änderungsprozesse. Eine Freigabe dient der Überprüfung und letztendlich Sperrung eines Objektes gegen weitere Schreibzugriffe, bevor es allgemein verfügbar gemacht wird. Nachdem ein Objekt erfolgreich einen Prozess durchlaufen hat, wird diesem zur Kennzeichnung ein bestimmter Status (z.B. Freigegeben) zugewiesen. Änderungsprozesse sind demgegenüber immer Folge eines aufgetretenen Problems und dienen der Änderung bereits freigegebener Objekte auf eine neue Revision. Eine *Klassifizierung* dient dem Ordnen von Objekten, unabhängig von deren Verwendung in einer oder mehreren Produktstrukturen. Klassifizierungsstrukturen basieren auf der Zusammenfassung ähnlicher Objekte zu Gruppen, welche durch eine zweckmässige Strukturierung effiziente Suchmechanismen ermöglichen, um das bereits erwähnte grosse Wiederholpotential zu nutzen.

Eine der wichtigsten Funktionen von PLM ist die *Zugriffskontrolle*, welche die Zugriffe der einzelnen Anwender kontrolliert und steuert.

Mittels Check-In/Check-Out Mechanismen wird dabei sichergestellt, dass zu einem bestimmten Zeitpunkt immer nur ein Anwender Schreibrecht auf ein Objekt besitzt, wodurch ein gegenseitiges Überschreiben von Datenbeständen verhindert wird.

Anhand der *Mailing*-Funktion können PLM-Benutzer miteinander kommunizieren und sich digital gegenseitig informieren sowie sich Objekte der PLM-Datenbank in Form von Referenzen zusenden (z. B. einen Projekt-Folder). Über Verteilerlisten lassen sich auch automatisierte Mitteilungen an Benutzergruppen realisieren (z. B. für Änderungsnotizen).

Mit der *Visualisierungs*-Funktion können PLM-Benutzer miteinander kommunizieren und sich digital gegenseitig informieren sowie sich Objekte der PLM-Datenbank in Form von Referenzen zusenden (z. B. einen Projekt-Folder). Über Verteilerlisten lassen sich auch automatisierte Mitteilungen an Benutzergruppen realisieren (z. B. für Änderungsnotizen).

Anhand der *Collaboration*-Funktionalität können PLM-Benutzer miteinander kommunizieren und sich digital gegenseitig informieren (z. B. via Markups) sowie sich Objekte der PLM-Datenbank in Form von Referenzen zusenden (z. B. einen Projekt-Folder). Über Verteilerlisten lassen sich mittels Notification auch automatisierte Mitteilungen an Benutzergruppen realisieren (z. B. für Änderungsnotizen).

## 4.2 Informationshandhabung mit PLM-Systemen

Mit Hilfe der oben erläuterten Anwendungsfunktionen lassen sich verschiedenste Produktinformationen über sämtliche Produktlebenslaufphasen gemeinsam in einheitlicher Form verwalten. Das PLM-System stellt hierbei die Integrationsdrehscheibe dar für alphanumerische Produktinformationen und graphische Produktvisualisierung bzw. -entwicklung. Jede Art von Produktinformation kann mittels PLM-Technologie konsistent abgelegt und verfügbar gemacht werden (vgl. Bild B010plmZ).

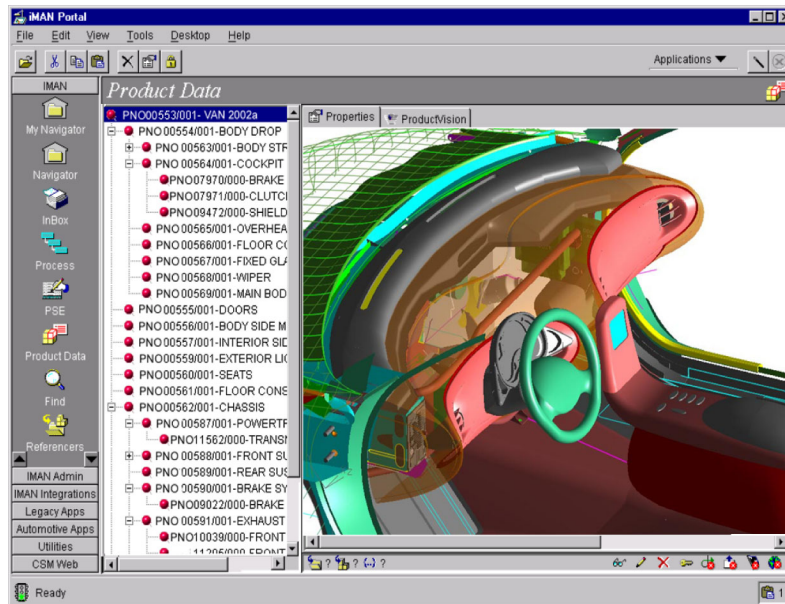


Bild (B010plmZ) Beispiel für die Darstellung alphanumerischer Produktstrukturinformationen in PLM zusammen mit der geometrischen Visualisierung in CAD

Von grundlegender Bedeutung für das Verständnis der PLM-Technologie ist die Unterscheidung von PLM-Objekten gegenüber Fremdobjekten externer Applikationen. PLM-Objekte werden im Datenbanksystem, d. h. in der Regel relational verwaltet und erlauben die uneingeschränkte Anwendung von PLM-Anwendungsfunktionen (z. B. Autorisierung, Revisionierung, Strukturierung), wo immer dies zweckmässig ist. Fremdobjekte werden hingegen von externen Fremdapplikationen (CAx oder DTP) erzeugt, die lediglich in PLM „eingekapselt“ sind als herkömmliche Dateien (engl. Files) in einem geschützten Betriebssystembereich (genannt Vault) und damit ausserhalb des eigentlichen Datenbanksystems verwaltet werden. Zur Einkapselung dieser Fremdobjekte in das PLM-Objektmodell werden spezielle Container-Objekte verwendet, häufig Dataset oder auch Document genannt. Diese Container-Objekte enthalten Referenzen (Zeiger) auf ein oder evtl. auch mehrere physische Files. Desweiteren besitzen Container-Objekte ein ganz besonderes Verhalten, da sie je nach Typ mit einer bestimmten Fremdapplikation verknüpft sind und es dadurch erlauben, referenzierte physische Files ausgehend vom PLM-System direkt in die Fremdapplikation zu laden (z.B. durch Doppelklicken). Container-Objekte (Datasets) können somit als die PLM-seitige Repräsentation von Files externer Fremdapplikationen (CAx/DTP)

angesehen werden, um diese mit den herkömmlichen PLM-Anwendungsfunktionen verwalten zu können.

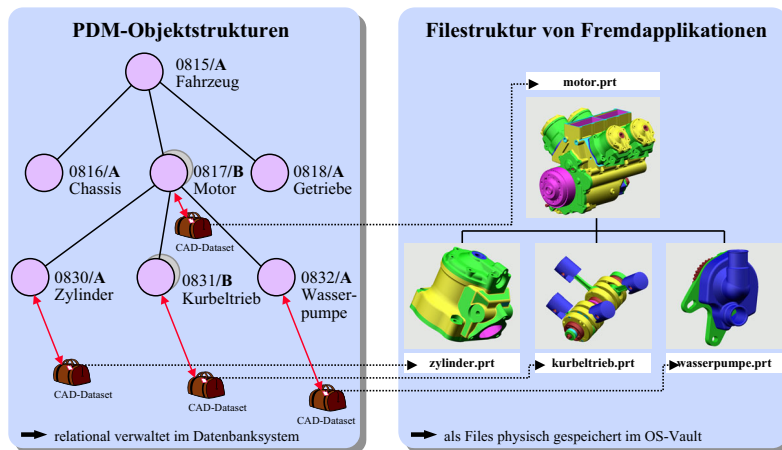


Bild (B011plmZ) Integration von Filestrukturen externer CAx-Applikationen in PLM

Das oben erläuterte Container-Prinzip wird durch das Anwendungsbeispiel in Bild B011plmZ veranschaulicht. Linkerhand ist die Produktstruktur eines Fahrzeuges dargestellt, die im Zuge der Entwicklung zunehmend weiter detailliert wird. Für den Motor und dessen Bestandteile sollen bereits CAD-Modelle existieren, die in Bild B011plmZ rechterhand abgebildet sind. Ein jedes CAD-Modell (z. B. zylinder.prt) wird PLM-seitig durch ein Container-Objekt (z.B. CAD-Dataset Zylinder) repräsentiert, welches dem entsprechenden Bauteilobjekt (z. B. 0830/A Zylinder) zugeordnet ist. Ausgehend von einem Bauteil in der PLM-Objektverwaltung können somit zunächst sämtliche zugeordneten Container-Objekte abgefragt werden (CAD-Datasets, Word-Datasets etc.). Bei Interesse kann eines ausgewählt und -vergleichbar mit dem Öffnen eines Koffers - das referenzierte physische File in die jeweilige Erzeugerapplikation geladen und damit geöffnet werden.

Solange Fremdapplikationen mit atomaren Files arbeiten, ist die Einkapselung in PLM mittels dem Container-Prinzip einfach und flexibel zu bewerkstelligen. Um ein Vielfaches komplizierter wird die Sachlage hingegen, falls eine Fremdapplikation vernetzte Filestrukturen verwendet und diese im PLM-System interpretiert werden müssen. Aus diesem Grunde stellt insbesondere die in CAD-Systemen seit einigen Jahren zunehmend unterstützte Assembly-Modellierung eine schwierige Herausforderung für die PLM-Systemwelt dar. Hierfür muss



ein bedeutend grösserer Integrationsaufwand betrieben werden, um eine hinreichende Interoperabilität von CAD-Assembly-Strukturen mit PLM-Produktstrukturen zu erreichen. Idealerweise wäre im Falle einer „tiefen“ CAD/PLM-Integration der Baugruppenaufbau des Motors im vorliegenden Beispiel von Bild B011plmZ sowohl in der PLM-Produktstruktur als auch in der CAD-Assembly-Struktur gleichermassen konsistent zugreifbar und bidirektional modifizierbar.

### 4.3 Systemarchitektur von PLM-Systemen

In systemtechnischer Hinsicht basieren PLM-Applikationen in höchstem Masse auf generischen Methoden, die eine grosse Flexibilität zur Anpassung an verschiedenste unternehmensspezifische Aufbau- und Ablauforganisationen ermöglichen. Voraussetzung hierfür ist zunächst eine IT-Standards basierende, modulare Systemarchitektur, die sich stark vereinfacht gemäss Bild B012plmZ darstellen lässt.

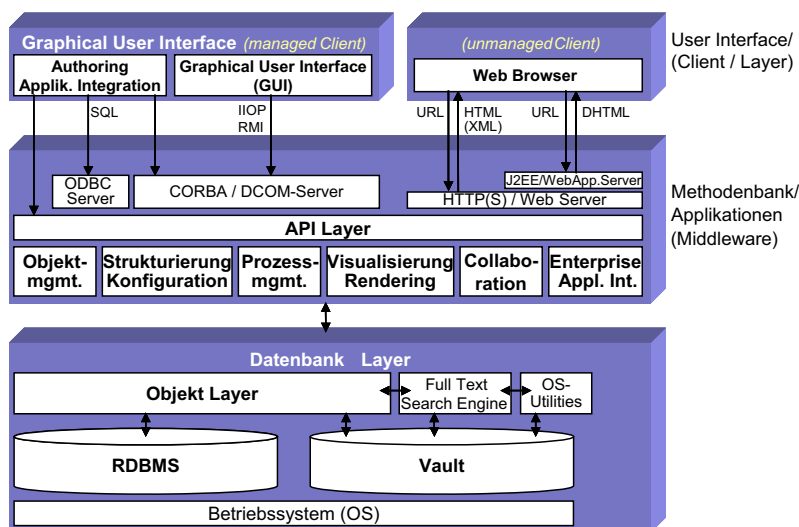


Bild (B012plmZ) Systemarchitektur von PLM-Applikationen

Der Benutzer kommuniziert mit dem System in der Regel über ein grafisch interaktives Benutzer-Interface, welches eine Client-seitige Installation voraussetzt (Managed Client). Für das User Interface werden heute komfortable Oberflächengeneratoren eingesetzt, die ein grafisches Entwerfen der Oberfläche erlauben, woraus der Programmcode in einem Software-Standard (wie z. B. Java), grösstenteils automatisch

erzeugt werden kann. In der Regel ist das graphische Benutzer-Interface plattformübergreifend, d. h. sowohl auf PCs (z.B. Windows NT/2000) als auch auf verschiedenen Unix-Implementierungen verfügbar.

Darüber hinaus bieten heute fast alle PLM-Applikationen Web-basierende Clients an (Unmanaged Client), die jedoch üblicherweise eine eingeschränkte Funktionalität aufweisen (Untermenge der GUI-Funktionalität). Hierbei ist zu unterscheiden zwischen einfacheren HTML-basierten Browser-Lösungen und leistungsfähigeren DHTML-Lösungen, die auf einer J2EE (Java 2 Enterprise Edition) Architektur aufbauen und mittels einem zusätzlichen Web Application Server (z. B. Tomcat oder Bea WebLogic), serverseitig Programmlogik in sogenannten Servlets ausführen können. Letztere haben den Vorteil einer gegenüber Applets besseren Performance und ermöglichen eine bessere Interaktivität und Funktionalität als reine HTML-Seiten (z. B. Drag & Drop).

Im Falle eingebundener Fremdapplikationen wird selbstverständlich deren Benutzeroberfläche verwendet, die gegebenenfalls kleine Modifikationen oder Erweiterungen erfährt.

Die eigentliche Funktionalität der PLM-Applikation ist in den einzelnen Modulen der Methodenbank hinterlegt, welche häufig als modulare Middleware-Implementierung über einen Serverprozess angesprochen wird, der grundsätzlich sowohl Client- als auch Serverseitig lauffähig sein kann. Haben die ursprünglichen PLM-Systeme vorwiegend die Module Objektmanagement, Strukturierung/Konfiguration und Prozessmanagement unterstützt, so bieten heute moderne PLM-Systeme zusätzlich Funktionalität für die 2D/3D Visualisierung und das Rendern von CAX- und/oder DTP-Datenstrukturen, zur Collaboration (Conferencing, Diskussionsforen, Notifications) und zur Integration mit weiteren Enterprise Applications (generische Triggerung weiterer autonomer Datenbanken).

Hinsichtlich der Datenverwaltung werden demgegenüber in der überwiegenden Mehrzahl der kommerziell verfügbaren Systeme zwei weitere Komponenten benötigt. Seit die Konzepte der Objektorientierung (OO) Einzug in die Software-Entwicklung gehalten haben, wird diese Technologie nach Möglichkeit für die Methodenbank zur Verfügung gestellt, um deren Vorteile nutzen zu können (Kapselung von Struktur und Verhalten, Vererbungsmechanismen etc.). Seitens der Datenbanktechnologie wird dies bis heute jedoch nicht hinreichend unterstützt, so dass für den produktiven Einsatz in der Industrie auf bewährte (relationale) Datenbanksysteme (RDBMS) zurückgegriffen werden muss. Aufgabe des Objekt Layers ist daher die Transformation der Objektstrukturen in Datenbankinstanzen. Das Datenbanksystem

übernimmt schlussendlich die Verwaltungsfunktion für sämtliche Instanzen. Die zweite Komponente stellt der sogenannte Vault (Tresor) dar, welcher die Files von CAD- und Office-Applikationen verwaltet. Zunehmend unterstützen PLM-Systeme auch Full Text Retrieval, wofür häufig Internet-Search Engines eingesetzt werden.

Von besonderer Bedeutung für den Benutzer und die Systemfunktionalität ist das auf der Ebene des Objekt-Layers definierte Objektmodell. Das Systemdesign des Objektmodells bestimmt massgebend die Art und Weise, wie Produktinformationen in einem PLM-System abgelegt werden können. Wenngleich die verfügbaren PLM-Systeme alleamt die vorgängig vorgestellten Anwendungsfunktionen (mehr oder weniger gut) unterstützen, so unterscheiden sie sich doch stark bezüglich dem „wie“ eine bestimmte Funktion erfüllt wird. PLM-Objektmodelle haben daher einen sehr systemspezifischen Charakter. Ein Beispiel für ein Objektmodell ist in Bild B013plmZ dargestellt.

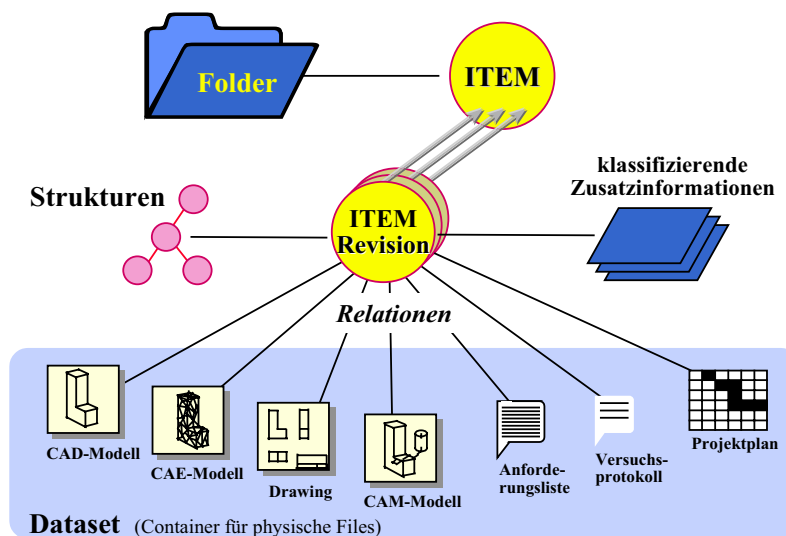


Bild (B013plmZ) PLM-Objektmanagement (beispielhaft)

In Anlehnung an herkömmliche Papierdossiers oder Ordner können zusammengehörende Objekte in Foldern gruppiert werden. Folder können dabei jede Art von Objekten beinhalten, wobei vorwiegend jedoch die „Hauptobjekte“ der Anwendungswelt, d. h. Bauteilstammdaten oder Dokumente zu finden sind. Diese Hauptobjekte werden hier Items genannt und haben als wesentlichste Eigenschaft die Fähigkeit zur Revisionierung. Anhand von Revisionen können mehrere

Änderungs- oder auch Entwicklungszustände festgehalten werden, um jederzeit wieder auf frühere Datenbestände zurückgreifen zu können. Item-Revisionen (z. B. Bauteil- oder Dokumentrevisionen) spielen hierbei eine zentrale Rolle, da diesen eine ganze Reihe weiterer Informationen in Form von eigenständigen Objekten zugeordnet sind. Einerseits können ausgehend von Item-Revisionen Strukturbeziehungen gebildet werden, deren wichtigste Vertreter Produktstrukturen sind (herkömmlich als Stücklisten bezeichnet). Andererseits können Item-Revisionen klassifizierende Zusatzinformationen zugeordnet werden in Form von Formularen oder Sachmerkmaleisteneinträgen.

Neben den bisherigen „reinen“ PLM-Objekttypen können mittels dem bereits diskutierten Container-Prinzip auch Fremdobjekte, d.h. Files externer Applikationen abgelegt werden. Für die verschiedenen Fremdapplikationen werden jeweils spezifische Dataset-Typen definiert (z. B. CAD-Modell, Word-File etc.), welche über Relationen einer Item-Revision zugeordnet werden können.

In diesem Objektmodell nehmen somit Item-Revisionen eine zentrale Stellung ein. Dies liegt an ihrer Fähigkeit zur Revisionierung begründet, was sie von sämtlichen anderen Objekten unterscheidet. Von Revision zu Revision kann daher jeweils entschieden werden, ob ein zugeordnetes Objekt (z. B. ein Word-Dataset) weiterhin zugeordnet werden soll oder ob ein solches massgebliche Änderungen erhält und infolgedessen eine physische Kopie neu zugeordnet wird.

## 5 Zusammenfassung

Der Rationalisierungsdruck in den Unternehmungen nimmt infolge andauernder Verschärfung der Marktsituation stetig zu. Die Produktlebenszyklen werden kürzer, der Termin- und Kostendruck wächst und die Ansprüche der Kunden gegenüber den Herstellern bezüglich Individualisierung der ursprünglich als Massenprodukt konzipierten Produkte nehmen zu. Dies zwingt die Unternehmen zur Entwicklung effizienterer, flexibler Prozesse. Darunter fällt auch das Datenmanagement. Moderne Informationssysteme gelangen zum Einsatz. In einem *Produkt Lifecycle Management (PLM) System* innerhalb eines Unternehmens werden Datenbestände und Anwendungsfunktionen des Digitalen Produktes untereinander verbunden und durch Web-Browser basierende, standardisierte Benutzeroberflächen zugänglich gemacht. Es handelt sich dabei also nicht um eine zentrale Datenbank mit einem unternehmensweiten Datenmodell. Aufgrund der Komplexität und der Verschiedenartigkeit der Daten und Anwendungen wäre dies keine zweckmässige Lösung. Vielmehr bildet ein PLM-System eine Integrationsdrehseibe und Backbone für alle technischen Produktdaten. Der Umstand, dass solche Systeme vor allem in der Entwicklung Anwendung finden, liegt darin begründet, dass der wesentlichste Erfolgsfaktor (Time to Market) in diesem Bereich bestimmt wird und die Kosten zwar mehrheitlich in der Beschaffung, Produktion und im Vertrieb verursacht, aber zum grössten Teil in der Entwicklung festgelegt werden.

### Verständnisfrage 1

Was sind die massgeblichen Trend der Gegenwart innerhalb einem modernen Industrieunternehmen?

### Verständnisfrage 2

Worin liegen die zukünftigen Erfolgspotentiale für ein Unternehmen?

### Verständnisfrage 3

Was versteht man unter dem Digitalen Produkt – wie ist dieses definiert?

### Verständnisfrage 4

Was versteht man unter PUT – was ist daran bedeutend für ein Industrieunternehmen?

#### Verständnisfrage 5

Was versteht man unter der Produktplattform, den Unternehmensprozessen und den strategischen Tools innerhalb des Digitalen Produktes?

#### Verständnisfrage 6

Welche beiden Hauptaspekte fordern durch den Fortschritt der Informationstechnologien eine Einführung von PLM?

#### Verständnisfrage 7

Was sind die Probleme bei konventioneller Auftragsabwicklung?

#### Verständnisfrage 8

Welche marktbezogenen Einflussfaktoren erfordern den Einsatz der PLM-Technologie?

#### Verständnisfrage 9

Was ist bezüglich dem unternehmerischen Gewinn heute der wichtigste Faktor?

#### Verständnisfrage 10

Was versteht man unter Produkthaftung?

#### Verständnisfrage 11

Was versteht man unter integrierter Produktmodellierung?

#### Verständnisfrage 12

Welche Prozessketten und Informationsflüsse können im unternehmerischen Gesamtzusammenhang unterschieden werden?

#### Verständnisfrage 13

Was versteht man im allgemeinen unter der rechnerintegrierten Produktentwicklung?

#### Verständnisfrage 14

Was für Integrationsgrade werden unterschieden?

**Verständnisfrage 15**

Was versteht man unter einer Modelltransformation, wodurch ist diese Charakterisiert?

**Verständnisfrage 16**

Worin liegen die grundsätzlichen Probleme der Modelltransformation?

**Verständnisfrage 17**

Wodurch ist die Modellintegration charakterisiert?

**Verständnisfrage 18**

Was versteht man unter einem integrierten Produktmodell?

**Verständnisfrage 19**

Welche drei Klassen von Produktmodelldaten werden unterschieden (Beispiele)?

**Verständnisfrage 20**

Was versteht man unter dem Begriff direkte Assoziativität?

**Verständnisfrage 21**

Wo kommt diese innerhalb des integrierten Produktmodells zu tragen?

**Verständnisfrage 22**

Was versteht man unter dem Begriff indirekte Assoziativität?

**Verständnisfrage 23**

Worin besteht der Unterschied zur direkten Assoziativität?

**Verständnisfrage 24**

Warum ist es sinnvoll das integrierte Produktmodell zu standardisieren?

**Verständnisfrage 25**

Worin liegen die Rationalisierungspotentiale?

**Verständnisfrage 26**

Was sind die wesentlichen Anforderungen an das integrierte Produktmodell?

**Verständnisfrage 27**

Was ist notwendige Voraussetzung für die Definition des integrierten Produktmodells?

**Verständnisfrage 28**

Warum gliedert sich das integrierte Produktmodell in einzelne Partialmodelle?

**Verständnisfrage 29**

In welcher Organisation und unter welchem Namen wird das integrierte Produktmodell definiert und standardisiert?

**Verständnisfrage 30**

Was ist der Unterschied von CIM und PLM?

**Verständnisfrage 31**

Was versteht man unter einem Repository?

**Verständnisfrage 32**

Welche Hauptzielsetzungen werden mit dem Einsatz von PLM verfolgt?

**Verständnisfrage 33**

Welche Anwendungsmodule unterscheidet man in PLM?

**Verständnisfrage 34**

Was sind die wichtigsten Anwendungsfunktionen von PLM-Systemen?

**Verständnisfrage 35**

Was speichert man in PLM-Systemen?

**Verständnisfrage 36**

Wie werden CAD-Daten in PLM gespeichert?



**Verständnisfrage 37**

Was ist ein Vault?

**Verständnisfrage 38**

Wodurch unterscheidet sich die Einbindung von DTP- gegenüber CAX-Applikationen im PLM?

**Verständnisfrage 39**

Welche Systemebenen kann man bei einem PLM-System unterscheiden?

**Verständnisfrage 40**

Welche Rolle spielt das Objektmodell für ein PLM-System?

**Verständnisfrage 41**

Welche positiven Rationalisierungseffekte lassen sich mit einem PLM-Projekt realisieren?

**Relevante Cases**

- [Studer](#)

**Publikationsverzeichnis – Literatur**

- [BeD-94] *Bernstein, P. A.; Dayal, U.*: An Overview of Repository Technology. Proc. 20th Int. Conf. on Very Large Databases (VLDB), Santiago (Chile) 1994, p. 705-713
- [Mar-97] *Martin, J.*: Part design comes together on the Net. Mechanical Engineering (US), June 1997, p. 76-78
- [EHS-91] *Eigner, M.; Hiller, C.; Schindewolf, S.; Schmich, M.*: Engineering Database: strategische Komponente in CIM-Konzepten. München: Hanser Verlag 1991
- [GAP-93] *Grabowski, H.; Anderl, R.; Polly, A.*: Integriertes Produktmodell. Berlin: Beuth verlag 1993
- [PaB-93] *Pahl G.; Beitz, W.*: Konstruktionslehre: Methoden und Anwendung. 3. Aufl., Berlin: Springer Verlag 1993
- [Sei-85] *Seiler, W.*: Technische Modellierungs- und Kommunikationsverfahren für das Konzipieren und Gestalten auf Basis der Modell-Integration. Fortschritt-Berichte VDI, Reihe 10: Nr. 49, Düsseldorf: VDI-Verlag 1985

- [VDI2210] *VDI-Richtlinie 2210: Analyse des Konstruktionsprozesses im Hinblick auf den EDV-Einsatz.* Düsseldorf: VDI-Verlag, November 1975
- [VDI-91] *VDI-Gemeinschaftsausschuss CIM, VDI-ADB: Auftragsabwicklung. Band 3: Rechnerintegrierte Konstruktion und Produktion,* Düsseldorf: VDI-Verlag 1991
- [Wie-89] *Wiendahl, H.-P.: Betriebsorganisation für Ingenieure.* 3. Auflage, München: Hanser Verlag 1989



# Datenbanktechnologie im Digitalen Produkt

Autor: Dr. R. Montau



# 1 Überblick

Die Bedeutung der Datenbankthematik lässt sich veranschaulichen an der heutigen Lebensdauer von Computer-Komponenten: Hardware wird nach ca. 2–3 Jahren als veraltet angesehen, Software spätestens nach 5–8 Jahren. Datenbestände müssen dagegen oftmals über 40 Jahre aufbewahrt und nutzbar gehalten werden.

Insbesondere in der Investitionsgüterindustrie und speziell bei Hochsicherheits-Produkten, wie z. B. bei Kernkraftwerken oder im Flugzeugbau, kommt der Langzeitarchivierung eine entscheidende Bedeutung zu. Erwähnt sei an dieser Stelle, dass zahlreiche Schweizer Unternehmen ihre Technischen Zeichnungen auch heute noch im Gesteinsmassiv der Alpen lagern. Die Entwicklungsdokumente beinhalten schliesslich das Know-how eines Unternehmens und erfordern somit ein Höchstmass an Sorgfalt und Sicherheit.

Die Datenbanktechnologie basiert auf dem Prinzip der Integration, das sich von einer jeden Kopplung wesentlich unterscheidet (vgl. Bild B001dbtZ).

Im Falle einer *Kopplung* sind die beteiligten Systeme nach wie vor autonom und der Informationsaustausch zwischen den Insellösungen erfolgt über einen Datentransfer, der entweder manuell durch den Menschen (durch erneutes Eintippen) oder anhand von Kopplungsprozessoren (Übertragungsprogrammen) erfolgt. Letztere unterscheidet man in spezifische und standardisierte Kopplungen, je nach dem, ob die Schnittstelle einer Norm entspricht, wodurch der Verknüpfungsaufwand entscheidend verringert wird. Um Änderungen im Datenbestand eines Systems auch an anderer Stelle bekannt zu machen, ist in jedem Falle ein Dateitransfer erforderlich.

Demgegenüber liegt einer *Integration* eine einheitliche Datenbasis zugrunde, worauf sämtliche Applikationen über dieselbe einheitliche Schnittstelle (Anfragesprache) zugreifen. Änderungen sind daher immer sofort für sämtliche Applikationen konsistent (widerspruchsfrei) verfügbar, was für zahlreiche Anwendungsfälle von grossem Vorteil ist.

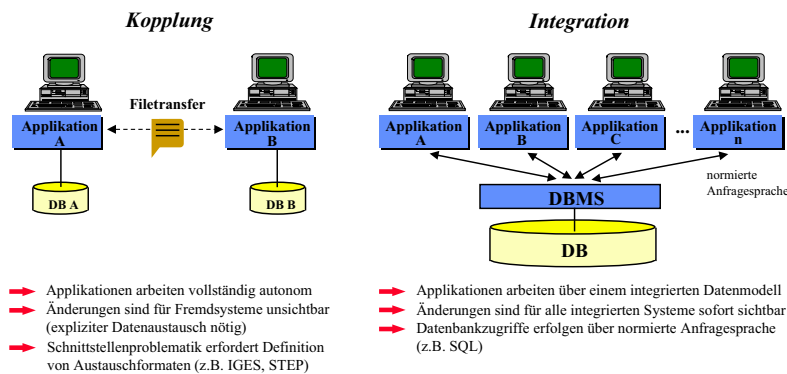


Bild (B001dbtZ) Kopplung versus Integration

Die auf dem Prinzip der Integration basierenden Datenbanksysteme bringen dabei eine ganze Reihe vorteilhafter Eigenschaften bzw. Funktionalität mit sich, die oftmals als Selbstverständlichkeit angesehen werden, wodurch deren Ursprung in Vergessenheit gerät. Um die Funktionalität und Bedeutung dieser Technologie für das Informationsmanagement hinreichend zu verdeutlichen und diese gegenüber Minimalst-Datenbanklösungen im PC-Umfeld abzugrenzen, seien nachfolgend die wichtigsten der mit der Verwendung eines Datenbanksystems einhergehenden Vorzüge erwähnt (vgl. [LoS-87, Vos-95, EIN-94, Zeh-89]):

Datenintegration	Vereinheitlichung der Datenverwaltung verschiedener Applikationssysteme in einem logisch zentralen Datenbestand
Datenunabhängigkeit	physische und logische Unabhängigkeit der Zugriffsstrategie der Methodenprogramme von der internen Datenverwaltung,
Datenkonsistenz	physische und logische Widerspruchsfreiheit der gespeicherten Informationselemente,
Datenintegrität	Gewährleistung einer korrekten und vollständigen Abbildung der Benutzersicht auf die Daten,
Redundanzfreiheit	Vermeidung überflüssiger Informationselemente sowohl explizit (jedes Element ist nur einmal physisch und logisch gespeichert) als auch implizit (kein Element kann aus einem anderen abgeleitet werden),
Transaktionsverwaltung	Eine Transaktion besteht aus einer Folge von Datenbankoperationen, die eine Datenbank von einem konsistenten Zustand in einen wiederum konsistenten Zustand überführt.

Tabelle (T001dbtZ) Vorzüge eines Datenbanksystems

Für die Integration von Applikationen auf der Basis eines Datenbanksystems sind einige grundlegende Aspekte von zentraler Bedeutung und stellen eine weitgehend unabdingbare Grundvoraussetzung für eine erfolgreiche PLM-Datenbanklösung dar.

- *Modellbildung von Datenbankapplikationen:*  
Um mehrere Applikationen durch eine gemeinsame Datenbank integrieren zu können, muss vorgängig ein übergreifendes Datenmodell erstellt werden, das sämtliche Informationselemente umfasst. Ausgehend von eigenständigen Applikationen mit jeweils spezifischen Datenmodellen müssen diese zu einem einheitlichen Modell zusammengefasst werden. Im Falle von sich „überschneidenden“ Informationsmengen müssen die sich semantisch entsprechenden Elemente zu einem einzigen Datenmodellelement vereinheitlicht werden (Schemaintegration). Hierbei ist zunächst konzeptuell ein integriertes Datenmodell zu entwerfen, welches schlussendlich einer systemtechnischen Realisierung bedarf.



- *Normierung der Anfragesprache:*  
Dieser Punkt ist vergleichbar mit dem Problem der internationalen Völkerverständigung, wofür eine allseits bekannte Sprache benötigt wird (z. B. Englisch). Es muss auch für den Datenbankzugriff verschiedener Applikationen eine global gültige und normierte Anfragesprache definiert werden. Hierdurch können sämtliche Applikationen in derselben Art und Weise auf den gemeinsamen Datenbestand zugreifen, um Datenobjekte einfügen, modifizieren oder auch nur lesen zu können.
- *Transaktionsverwaltung für konkurrierende Mehrbenutzerzugriffe:*  
Durch die Integration von Applikationen stellt sich ein Mehrbenutzerbetrieb ein, wobei die einfachste und sicherste Lösung wäre, gleichzeitig immer nur einem Benutzer Zugriff zu gewähren und damit sämtliche Anfragen zu serialisieren. Das ist jedoch keinesfalls tragbar, da erst durch die Parallelisierung der erwünschte Hauptnutzen ermöglicht wird (Zeiteinsparungen für die Produktentwicklung im Team; vgl. weltweite Bank- und Flugreservations-Datenbanken). Schliesslich würde es keinen Sinn ergeben, wenn eine sehr kleine bzw. kurze Transaktion (z. B. Speichervorgang eines CAD-Modelles) auf die Beendigung einer sehr langwierigen Transaktion (z. B. FEM-Berechnung) warten müsste, die schlussendlich vielleicht sogar abgebrochen wird. Aus diesen Gründen besitzen Datenbanksysteme eine sogenannte *Transaktionsverwaltung*, die für eine konsistente Abarbeitung paralleler Zugriffe Sorge trägt und somit eine ganz wesentliche Voraussetzung für die erfolgreiche Anwendung dieser Technologie darstellt.
- *Datensicherung gegen Systemfehler:*  
Aufgrund der grossen Bedeutung der Datenbestände hinsichtlich langfristiger Verfügbarkeit für juristische Gewährleistungsfälle zum einen und aufgrund der grossen Abhängigkeit heutiger Entwicklungsteams von einer möglichst uneingeschränkten Datenverfügbarkeit zum anderen müssen Vorkehrungen für allfällige Systemfehler getroffen werden. Um negative Auswirkungen durch Systemfehler jeder erdenklichen Art (Systemversagen, Stromausfall, Feuer, Wasserschäden etc.) möglichst minimieren zu können, sind vorbeugend geeignete Massnahmen erforderlich.
- *Verteilung von Datenbanken:*  
Der Integration von Applikationen sind natürliche Grenzen gesetzt, da einerseits heutige Computersysteme nicht in der

Lage sind, Datenbankzugriffe für beliebig grosse Benutzerzahlen (über 1 000 oder gar 10 000 Benutzer) mit befriedigender Performance abzuwickeln. Andererseits ermöglichen die heute verfügbaren Netzwerkverbindungen keinen dauerhaften weltumspannenden Direktzugriff, der zugleich laufzeiteffizient, kostengünstig und ausfallsicher ist. Darüber hinaus ist eine Vollintegration auch nicht in jedem Falle zweckmässig, wenn man beispielsweise den Datenaustausch mit Zulieferfirmen oder auch Kunden sowie temporär aquirierten Sub-Unternehmen betrachtet, wobei einer der Standorte z. B. in technologisch nur mässig erschlossenen Gebieten gelegen sein kann. Für derartige Szenarien wird daher eine Möglichkeit benötigt, um autonome Datenbanken eigenständig betreiben zu können und zugleich eine gewisse Interoperabilität zu unterstützen.

Die obigen Aspekte sollen aufgrund ihrer grundlegenden Bedeutung für Datenbankapplikationen im Rahmen der folgenden fünf Hauptabschnitte (Kap. Modellbildung von Datenbankapplikationen bis Verteilungskonzepte in PLM-Systemen) detaillierter betrachtet werden.

## 2 Modellbildung von Datenbankapplikationen

Grundlage einer jeden Beschreibung der realen Welt durch ein Modell ist die Abstraktion sämtlicher informationeller Zusammenhänge auf interessierende relevante Sachverhalte, wobei nach dem ANSI/SPARC-Komitee [ANS-75] drei Abstraktionsebenen zu unterscheiden sind, die über der physischen Datenverwaltung eines Betriebssystems angeordnet sind (vgl. Bild B002dbtZ).

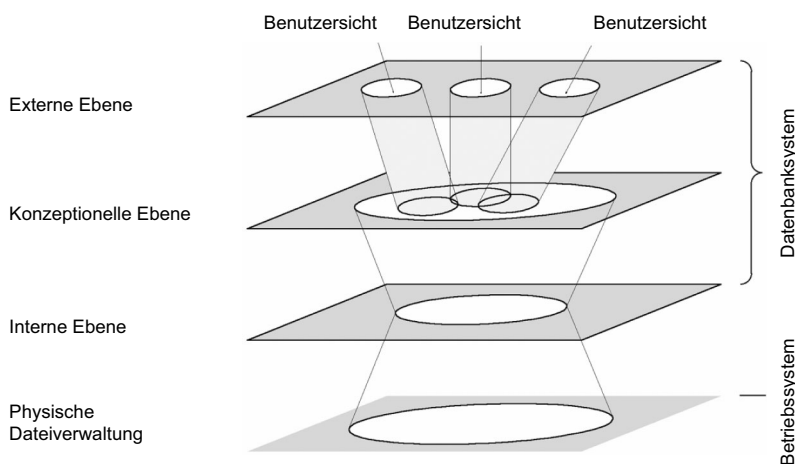


Bild (B002dbtZ) 3-Ebenenkonzept nach dem ANSI/X3/SPARC-Architekturmodell

Von besonderer Bedeutung ist hierbei die *konzeptuelle Ebene*, welche die logisch formale Gesamtsicht sämtlicher in der Datenbank enthaltener Informationen repräsentiert. Implementierungsbedingte Restriktionen müssen hierbei nicht berücksichtigt werden, da keine physische Realisierung auf irgendeinem Speichermedium erforderlich ist.

Auf der *internen Ebene* wird die physische Speicherungsform für die verwendeten Datenstrukturen festgelegt, die zur programmtechnischen Realisierung nötig ist. Auf dieser Festlegung basiert auch die Abbildung vom logischen in den physischen Adressraum anhand bestehender Betriebssystem-Funktionalität.

Im Gegensatz hierzu können auf der *externen Ebene* verschiedene individuelle Sichten (Views) auf das konzeptuelle Modell definiert werden, die nur einen beliebigen Ausschnitt der Gesamtsicht beinhalten, sei es, weil ein Benutzer sich nur für einen Teil der Informationen interessiert oder weil er nur für diesen Zugriffsberechtigung besitzen soll.

## 2.1 Konzeptueller Datenbankentwurf

Für die Modellbildung insbesondere auf der konzeptuellen Ebene des ANSI/SPARC-Architekturmodells sind zwei wesentliche Aspekte zu unterscheiden, einerseits der

- Modellierungsaspekt zur Beschreibung realer Sachverhalte in einem *semantischen Informationsmodell* sowie andererseits der
- Implementierungsaspekt zur systemtechnischen Realisierung anhand eines *operationalen Modells*.

Semantisches und operationales Modell haben einen engen Bezug zueinander, da sie im Entwurfsprozess direkt aufeinanderfolgen und das Letztere aus dem Ersteren abgeleitet wird.

Ausgangspunkt gemäss Bild B003dbtZ ist eine Informationsanalyse der realen Welt, woraus ein semantisches Modell in einer Datenmodellierungssprache (vorwiegend in grafischer Notation) erarbeitet wird, das letztlich auf ein implementierbares, operationales Modell für ein Datenbankschema abgebildet wird.

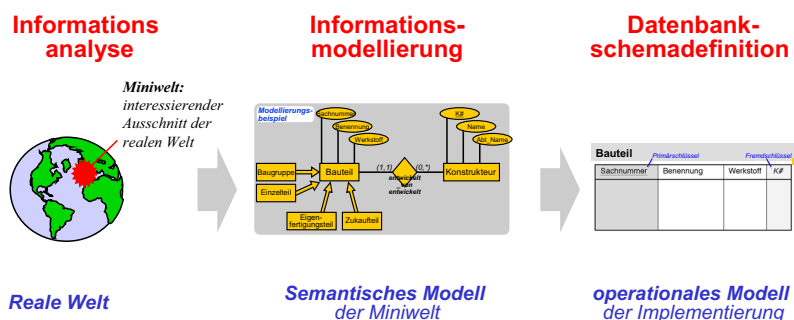


Bild (B003dbtZ) Vorgehensweise beim konzeptuellen Datenbankentwurf

Zielsetzung der semantischen Informationsmodellierung ist die möglichst realitätsgetreue Abbildung realer Sachverhalte der Diskurswelt (interessierender Ausschnitt der realen Welt) auf ein Informationsmodell mit grösstmöglicher Semantik (Bedeutung).

Veranschaulicht werden soll dies am Beispiel der Objekte Bauteil und Konstrukteur, deren Definition sich aus den gemeinsamen Eigenschaften konkreter Bauteile und Konstrukteure (vgl. Mengendiagramm in Bild B004dbtZ) ableitet. Neben den eigentlichen Merkmalen eines Bauteils (Sachnummer, Werkstoff, Benennung) besteht eine (1:n)-Beziehung „entwickelt\_von“ zu Konstrukteur. Die Kardinalität (1:n) beschreibt die minimale und maximale Anzahl, die ein Element diese Beziehung eingehen kann (ein Bauteil wird immer genau von einem

Konstrukteur entwickelt. Ein Konstrukteur kann dagegen mehrere Bauteile entwickelt haben).

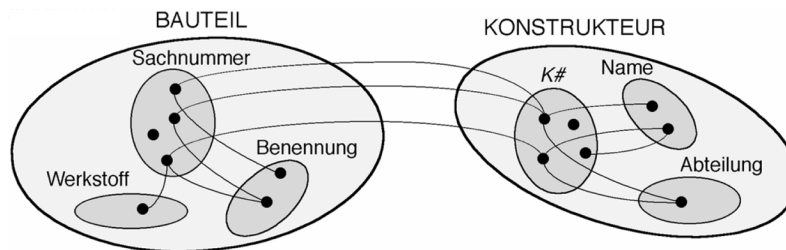


Bild (B004dbtZ) Mengendiagramm des Modellierungsbeispiels

## 2.2 Semantische Informationsmodellierung

Ausgehend von einer eingehenden Informationsanalyse (z. B. in Form eines Mengendiagramms) wird ein semantisches Informationsmodell entworfen, wofür mehrere semantische Datenmodellierungssprachen zur Verfügung stehen. Hier wird das Entity Relationship Model (kurz ERM) nach Chen verwendet [Che-76], welches sich durch vielfältige semantische Modellierungskonstrukte auszeichnet sowie seine Übersichtlichkeit, Einfachheit und weite Verbreitung. Hiermit lassen sich produktionstechnische Randbedingungen und Gesetzmässigkeiten sehr effizient und elegant abbilden.

Objekte der realen Welt werden im ER-Modell als Entities aufgefasst und mittels Entity Types beschrieben (z. B. ein Konstrukteur). Beziehungen zwischen Entities werden in Form von Relationships abgebildet. Z. B. kann durch die Beziehung entwickelt\_von, für ein Bauteil derjenige Konstrukteur zugeordnet werden, der dieses entwickelt. Mit Hilfe von sogenannten Kardinalitäten kann angegeben werden, mit wieviel Entities einer anderen Klasse minimal und maximal eine „Beziehungspartnerschaft“ eingegangen werden kann. Von links nach rechts betrachtet kann aus Bild B005dbtZ beispielsweise abgeleitet werden, dass für ein Bauteil mindestens 1 und maximal 1 (1,1) verantwortlicher Konstrukteur über die entwickelt\_von-Relationship angegeben werden darf. Im Gegensatz dazu kann von rechts nach links betrachtet ein Konstrukteur minimal kein und maximal unendlich viele (0;\*) Bauteile entwickeln. Über eine Spezialisierung kann ein Entity Type in Subtypes untergliedert werden, z.B. Bauteile in Eigenfertigungsteile und Zukaufteile oder in Baugruppen und Einzelteile.

Sprachlich benennbare konkrete Objekte werden als Attribute abgebildet, denen Werte zugewiesen werden können (z. B. Name = ‚Müller‘). Attribute, welche eine identifizierende Funktion besitzen, sind sogenannte Primärschlüssel und werden unterstrichen dargestellt.

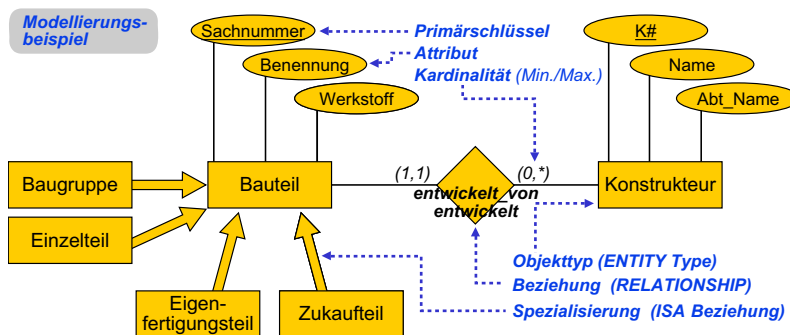


Bild (B005dbtZ) Beispiel eines semantischen Informationsmodells in NIAM

Als operationale Modelle kommen traditionellerweise das hierarchische Datenmodell, das Netzwerkdatenmodell und insbesondere das Relationenmodell zur Anwendung. Alle drei Modelle sind als kommerziell eingeführte Datenbanktechnologien verfügbar und aufgrund ihres guten Laufzeitverhaltens bei der geometrischen Modellierung komplexer Bauteilstrukturen besitzen Netzwerkmodelle insbesondere im CAD-Bereich eine weite Verbreitung. Das hierarchische und das Netzwerkmodell weisen jedoch keine exakte Trennung der geforderten drei Abstraktionsebenen auf (vgl. Bild B002dbtZ) und sind hinsichtlich Datenabhängigkeit und Redundanzfreiheit ungenügend. Charakteristisch ist weiterhin das *prozedurale* Navigieren in den Datenstrukturen, wobei jeweils angegeben werden muss, *wie* auf die Daten zugegriffen werden soll.

Das Relationenmodell ermöglicht eine Abkehr von diesen Prinzipien, weshalb es einen bedeutenden Meilenstein in der Datenbankforschung darstellt und die übrigen Modelle im Bereich der Informationssysteme fast völlig verdrängt hat.

## 2.3 Relationenmodell als operationales Modell

Das von Codd vorgeschlagene Relationenmodell [Cod-70] ermöglicht eine strikte Trennung der logischen und physischen Ebene und besitzt dadurch erstmals eine vollständige Datenunabhängigkeit. Auf verschiedene Strukturierungsmechanismen wird verzichtet, da sämtliche Informationen auf ein einziges Abstraktionskonzept zurückgeführt werden, das als Relation auf einer mathematisch exakten Definition basiert. Ausgehend von  $n$  Wertebereichen (Domänen) einer Folge von Attributen  $(A_1, \dots, A_n)$  ist eine Relation  $R$  als Teilmenge des kartesischen Produkts dieser Domänen definiert:

$$R \subseteq \text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n).$$

Eine Relation ist damit als eine Menge von Tupeln aufzufassen. Die genormte relationale Anfragesprache Sprache SQL (Structured Query Language) [ANS-86] ermöglicht nunmehr einen *deskriptiven* Zugriff, wobei im Gegensatz zur obigen Vorgehensweise nur noch angegeben werden muss, *welche* Daten abgefragt werden sollen.

Anhand einer Reihe feststehender Abbildungsregeln, die erst nachfolgend näher diskutiert werden sollen (wie z.B. semantischer Objekttyp, d.h. ENTITY ergibt relationale Tabelle; semantisches Attribut aus dem ERM ergibt ein relationales Attribut etc.), kann ein semantisches Informationsmodell auf ein relationales Modellschema abgebildet werden. Hierfür existieren kommerziell erhältliche Datenbankmodellierungs-Tools, die eine solche Transformation, wie z.B. von ERM auf das Relationenmodell, automatisch ausführen. Das Ergebnis für das vorliegende Beispiel ist in Bild B006dbtZ dargestellt.

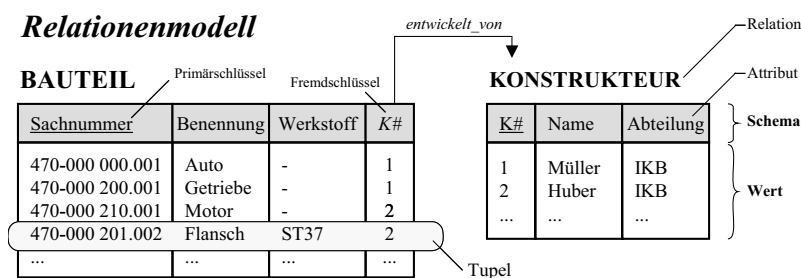


Bild (B006dbtZ) Anwendungsbeispiel im Relationenmodell

Eine Relation besteht immer aus zwei wesentlichen Bestandteilen, dem Schema (Tabellenaufbau) und den Werteausprägungen (Tabelleninhalt). Das Schema ergibt sich aus der Abstraktion der Informationsanalyse im Mengendiagramm auf verallgemeinerte Objekttypen des

semantischen Modells. Die in der realen Welt existierenden konkreten Bauteile und Konstrukteure, welche im Mengendiagramm durch Punkte dargestellt werden, ergeben die Werte bezüglich der relationalen Domänen (Attributwertebereiche). Eine Objekt*instanz* wird hierbei durch ein Tupel repräsentiert.

Von besonderer Bedeutung für eine Relation sind deren Schlüssel. Der *Primärschlüssel* ist eine, evtl. unter mehreren Schlüsselkandidaten ausgewählte minimale Kombination von Attributen, die ein Tupelobjekt eindeutig identifizieren. *Fremdschlüssel* müssen dagegen nicht eindeutig sein, erfordern jedoch ein gleichwertiges Partnertupel im Primärschlüssel einer anderen Relation. Hervorzuheben ist, dass diese beiden Integritätsbedingungen Bestandteil des Relationenmodells und damit modellinhärent sind (Tabelle T002dbtZ). Im Falle der Verwendung eines relationalen Datenbanksystems werden diese modellinhärenten Integritätsbedingungen systemseitig sichergestellt, woraus grosser Nutzen gezogen werden kann.

Entity Integrity	Primärschlüsselwerte sind nie mit Nullwerten belegt (undefiniert) und identifizieren ein Tupel eindeutig.
Referential Integrity	Fremdschlüsselwerte sind entweder undefiniert (null) oder es existiert ein entsprechendes Partnertupel in einer anderen Relation, wo die referenzierte Domäne als Primärschlüssel definiert sein muss.

Tabelle (T002dbtZ) Modellinhärente Integritätsbedingungen des Relationenmodells

Im vorliegenden Modellierungsbeispiel in Bild B005dbtZ wird z. B. die Eindeutigkeit der Sachnummer durch die Primärschlüsseldefinition gewährleistet und die (1:n)-Kardinalität der Beziehung zu Konstrukteur durch die Fremdschlüsselbedingung erzwungen, da immer genau nur einer angegeben werden kann. Durch die Abbildung von Funktionalabhängigkeiten der realen Welt auf modellinhärente Integritätsbedingungen des relationalen Datenbanksystems werden diese systemseitig sichergestellt und sowohl der Entwickler der Applikation als auch deren Benutzer müssen sich hierum nicht kümmern.

Dieser Nutzen stellt sich jedoch nicht automatisch ein, sondern ist von der Art der Modellierung abhängig. Trotz der Integritätsbedingungen in Tabelle T002dbtZ können im Falle eines schlechten Modellentwurfs (vgl. Beispiel in Bild B007dbtZ) durchaus grössere Probleme auftreten, die allgemein als Anomalien bekannt sind:



Einfüge-Anomalie	Ein neuer Tupel eintrag mit zumindest teilweise noch unbekanntem Attributwerten (Nullwerten) ist nicht möglich, wird aber gewünscht
Lösch-Anomalie	Durch das Löschen eines Tupels gehen ungewollt weitere Informationen verloren, die noch benötigt werden.
Änderungs-Anomalie	Eine Informationsänderung bedingt eine Änderung mehrerer Tupel, die gleichzeitig erfolgen muss, um Inkonsistenzen bei Mehrbenutzerbetrieb zu verhindern.

Tabelle T003dbtZ) Anomalien

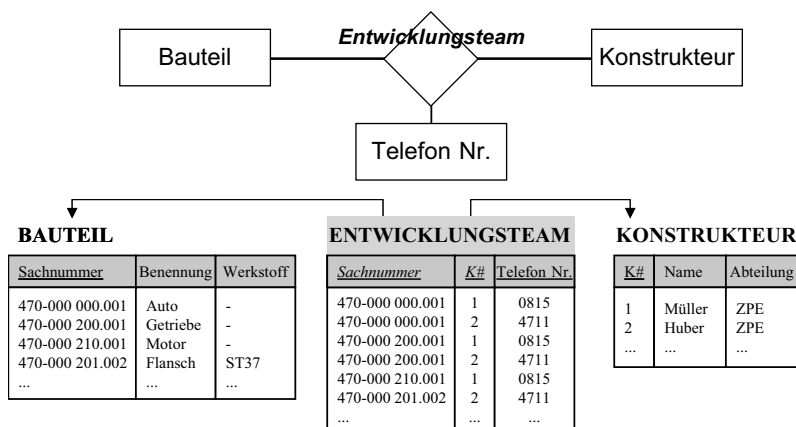


Bild (B007dbtZ) Beispiel einer schlechten (!) Modellierung

Dieses Beispiel verdeutlicht die grosse Tragweite einer zweckmässigen Datenbankmodellierung. Im Anschluss an die bisherigen sehr beispielhaften Erläuterungen zum Datenbankmodellentwurf soll nachfolgend vertieft auf das Vorgehen zur Abbildung semantischer NIAM-Modelle auf relationale Datenbankschemata eingegangen werden.

## 2.4 Abbildung vom ERM auf das Relationenmodell

Basierend auf dem erarbeiteten Grundverständnis für die semantische Informationsmodellierung kann der Abbildungsprozess vom semantischen ER-Modell auf das operationale Relationenmodell detaillierter betrachtet werden. Für sämtliche ER-Modellkonzepte lassen sich äquivalente Konstrukte im Relationenmodell angeben, wobei einerseits Fallunterscheidungen getroffen werden müssen und andererseits ein gewisser Informationsverlust entstehen kann.

NIAM	Relationenmodell
Objektyp bzw. Entity	Relation
Merkmalsfunktion bzw. Attribut	Attribut
Beziehung bzw. Relationship	<p>3 Fälle:</p> <ul style="list-style-type: none"> <li>• <i>binär-hierarchische Beziehung</i>: Fremdschlüsseldefinition</li> <li>• <i>allgemeine Beziehung</i> (<i>n-är und/oder n:m-Kardinalität</i>): Relation mit den Schlüsselattributen der beteiligten Objekttypen</li> <li>• <i>rekursive Beziehung</i>: Relation mit dem Schlüsselattribut des beteiligten Objekttyps bezeichnet mittels Rollennamen</li> </ul>
Spezialisierung bzw. Subtype	<p>keine direkte Abbildung, 2 Möglichkeiten:</p> <p>vertikale Partitionierung:</p> <ul style="list-style-type: none"> <li>• Die Relation des allgemeinen Objekttyps beinhaltet sämtliche Objektinstanzen (Tupel); die Relation des speziellen Objekttyps beinhaltet nur die speziellen Objektinstanzen, d.h. eine Untermenge des Ersteren.</li> <li>• Das Schema des speziellen Objekttyps beinhaltet nur die speziellen Attribute.</li> </ul> <p>horizontale Partitionierung:</p> <ul style="list-style-type: none"> <li>• Die Relation des allgemeinen Objekttyps beinhaltet nur allgemeine Objektinstanzen (Tupel); die Relation des speziellen Objekttyps beinhaltet nur spezielle Objektinstanzen (Tupel), d.h. die Objektmengen sind disjunkt.</li> <li>• Das Schema des speziellen Objekttyps beinhaltet sowohl die allgemeinen als auch die speziellen Attribute.</li> </ul>
Max. Kardinalität bzw. IDENTIFIER	Nicht direkt abbildbar, bei Objektbeziehungen ggf. Fremdschlüsseldefinition
Min. Kardinalität bzw. Existenz	NOT NULL-Bedingung

Tabelle (T005dbtZ) Abbildung von ER-Konzepten auf das Relationenmodell

Insbesondere für die Abbildung von Beziehungen sind Fallunterscheidungen erforderlich, da sich je nach Anzahl der beteiligten Objekttypen (binäre Beziehung zwischen 2 Objekttypen oder n-äre Beziehung zwischen 3 und mehr Objekttypen) und der Kardinalität (1:n bzw. n:m) unterschiedliche relationale Modellschemata ergeben. Zur Verdeutlichung sind für die wichtigsten repräsentativen Beziehungsfälle in Bild B008dbtZ einige beispielhafte Abbildungen dargestellt.

### Abbildung des ERM auf das Relationenmodell

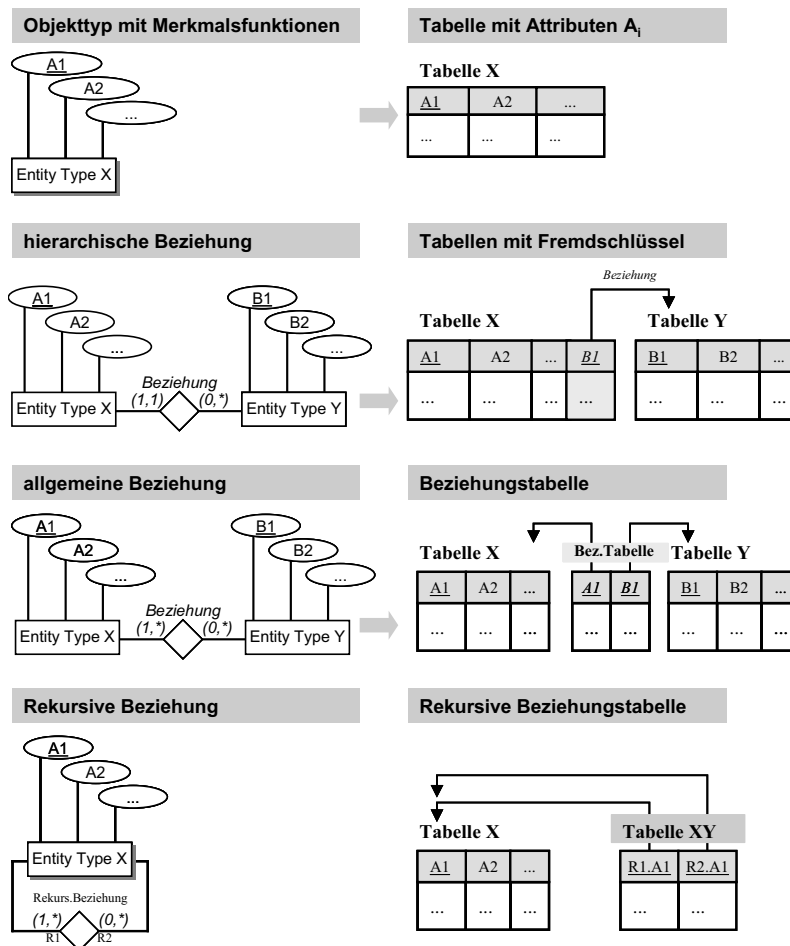


Bild (B008dbtZ) Abbildung beispielhafter ERM-Beziehungen auf das Relationenmodell

Des weiteren können Spezialisierungen (Subtype-Konzept im ERM) nicht direkt in das Relationenmodell überführt werden, weshalb zwischen zwei Varianten gewählt werden kann, der vertikalen und der horizontalen Partitionierung [Sche-90a].

Naheliegender ist in der Regel eine vertikale Partitionierung, wobei die Spezialisierung auf eine Fremdschlüsseldefinition im Datenbankmodellschema abgebildet wird und in der allgemeineren Tabelle (X) sämtliche Tupel gespeichert werden. Vorteilhaft hieran ist der direkte Zugriff auf die Gesamtheit aller Tupel, ohne eine tabellenübergreifende Join-Abfrage zu erfordern (Bild B009dbtZ).

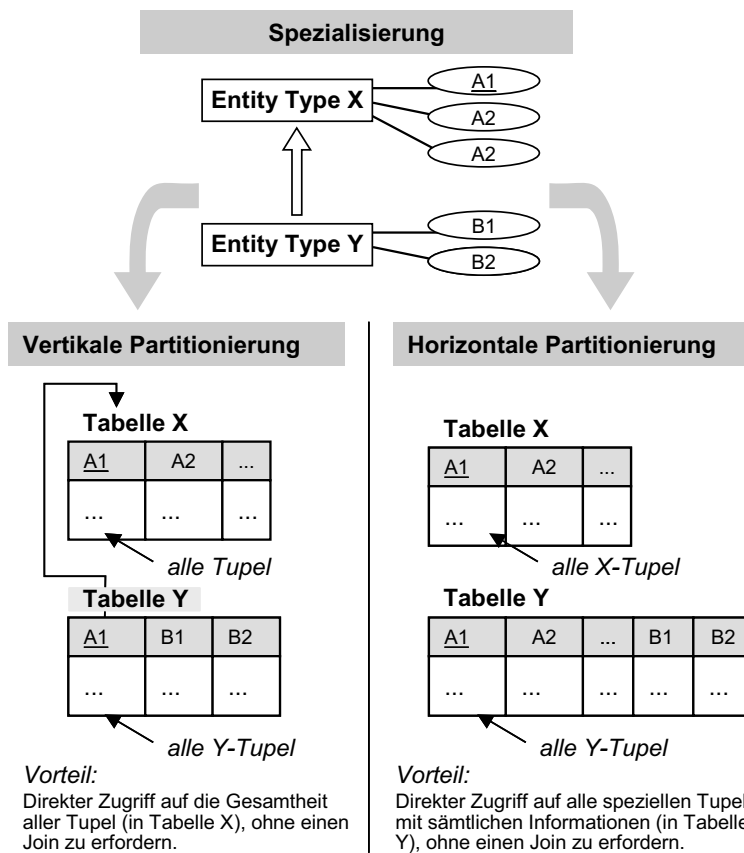


Bild (B009dbtZ) Abbildung einer Spezialisierung vom ERM in das Relationenmodell

Bei der horizontalen Partitionierung werden zwei unabhängige Tabellen definiert und sämtliche allgemeinen Attribute auch in das Schema der speziellen Tabelle (Y) übernommen. Somit werden die speziellen Tupel ausschliesslich in der speziellen Tabelle beschrieben und gespeichert, woraus sich der Vorteil des direkten Zugriffs auf spezielle Tupel ergibt.

### 3 Anfragesprache SQL

Um dem operationalen Aspekt zur systemtechnischen Umsetzung und Nutzung der vorgängig diskutierten Datenmodelle Rechnung tragen zu können, wird ein Mechanismus zum Schreiben und Lesen von Objektinstanzen benötigt, eine sogenannte Anfragesprache. Charakteristisch für semantische Datenmodelle ist, dass sie zwar eine reichhaltige semantische Ausdrucksfähigkeit besitzen, aber leider über keinerlei Anfragesprache verfügen. „Semantische Datenbanksysteme“ existieren somit nicht.

Seit einigen Jahren sind zunehmend Entwicklungen im Bereich objektorientierter Datenbanksysteme (OODBMS) zu verzeichnen [ABD-89, Dit-92, Dit-95, Nor-93], die ein semantisches Datenmodell unterstützen und zudem eine Anfragesprache besitzen. Objektorientierte Datenbanksysteme können daher als systemtechnische Realisierung semantischer Datenmodelle angesehen werden. Bis heute weisen OODBMS jedoch noch eine Reihe von Unzulänglichkeiten auf (z.B. Concurrency Control), weshalb sie bislang für den produktiven Einsatz in PLM-Systemen nur sehr selten eingesetzt werden. Als operationales Modell wird daher fast ausschliesslich das Relationenmodell verwendet.

Die genormte relationale Anfragesprache Sprache SQL (Structured Query Language) [ANS-86] hat wesentlichen Anteil am grossen Erfolg des Relationenmodells. Im Gegensatz zur Vorgehensweise herkömmlicher 3GL-Programmiersprachen (C, Pascal, Fortran etc), die ein *prozedurales* Navigieren in den Datenstrukturen unter Berücksichtigung physischer Speicherstrukturen erfordern (Programmierer muss sagen *wie* auf Daten zugegriffen werden soll), benötigt SQL nur noch *deskriptive* Angaben (Programmierer muss lediglich sagen, *was* für Daten abgefragt werden sollen). Das Grundgerüst für Datenbankzugriffe beruht dabei auf der folgenden Syntax:

SELECT	Spaltenattribute
FROM	Tabellenname
WHERE	Auswahlbedingung

Tabelle (T008dbtZ) Standard Anfragesyntax

Aufgrund der vollständigen Datenunabhängigkeit des Relationenmodells, sind hierbei keinerlei Kenntnisse über den physischen Speicherungsort der Objektdaten erforderlich, d.h. der Anwendungsprogram-

mierer braucht sich darum nicht zu kümmern. Da sich der physische Speicherort im Programmcode nicht widerspiegelt, muss derselbe bei Änderungen der physischen Speicherorganisation auch nicht modifiziert werden.

In funktioneller Hinsicht besteht eine Datenbankanfragesprache aus den folgenden beiden Haupt-komponenten:

- Datendefinition (Data Definition Language, DDL),
- Datenmanipulation(Data Manipulation Language, DML).

Die *Datendefinition* ermöglicht den Aufbau eines Datenbankschemas entsprechend einem vorgängig festgelegten Datenmodellentwurf. Hieraus folgt eine exakte Definition der Datenstrukturen, die in einer Datenbank abgelegt werden können.

Im Anschluss hieran können einzelne Objektinstanzen (Tupel) mit Hilfe der *Datenmanipulation* in die Datenbank eingegeben, d.h. erzeugt werden sowie gegebenenfalls wieder gelesen, modifiziert oder auch gelöscht werden. Grundfunktionen der Datenmanipulation sind somit das Erzeugen, Lesen, Modifizieren und Löschen von Objekten (engl. Insert, Select, Update, Delete).

Zur exemplarischen Verdeutlichung sind in Tabelle T005dbtZ einige repräsentative Beispiele für SQL-Statements angegeben, die sich auf das Modellschema in Bild B006dbtZ beziehen.

Aktion	SQL-Statement
Tabelle Konstrukteur erzeugen	CREATE TABLE zpedb.Konstrukteur( K# INTEGER NOT NULL, Name VARCHAR(30) NOT NULL, Abteilung VARCHAR(20) NOT NULL, CONSTRAINT PK_K# PRIMARY KEY (K#));
Tabelle Bauteil erzeugen	CREATE TABLE zpedb.Bauteil( Sachnummer VARCHAR(15) NOT NULL, Benennung VARCHAR(25) NOT NULL, Werkstoff VARCHAR(20), K# INTEGER, CONSTRAINT PK_Sachnummer PRIMARY KEY(Sachnummer);
Fremdschlüssel definieren von Bauteil auf Konstrukteur	ALTER TABLE zpedb.Bauteil ADD CONSTRAINT FK_Konstrukteur FOREIGN KEY (K#) REFERENCES zpedb.Konstrukteur (K#) ON DELETE RESTRICT);
Tupel in Konstrukteur einfügen	INSERT INTO zpedb.Konstrukteur (K#, Name, Abteilung) VALUES (1, 'Müller', 'ZPE'); INSERT INTO zpedb.Konstrukteur (K#, Name, Abteilung) VALUES (2, 'Huber', 'ZPE');

Tabelle (T005dbtZ) Beispiele für SQL-Statements

Aktion	SQL-Statement
Tupel in Bauteil einfügen	<pre>INSERT INTO zpedb.Bauteil (Sachnummer, Benennung, Werkstoff, K#) VALUES (,470-000 000.001', ,Sportwagen', ,-', 1); INSERT INTO zpedb.Bauteil (Sachnummer, Benennung, Werkstoff, K#) VALUES (,470-000 200.001', ,Schaltgetriebe', ,-', 1); INSERT INTO zpedb.Bauteil (Sachnummer, Benennung, Werkstoff, K#) VALUES (,470-000 210.001', ,Motor', ,-', 2); INSERT INTO zpedb.Bauteil (Sachnummer, Benennung, Werkstoff, K#) VALUES (,470-000 201.002', ,Getriebe- flansch', ,ST37', 2);</pre>
Tupel von Bauteil anzeigen	<pre>SELECT * FROM zpedb.Bauteil;</pre>
Konstrukteure der Abteilung ZPE anzeigen	<pre>SELECT NAME FROM zpedb. Konstrukteur WHERE Abtei- lung = ,ZPE';</pre>
Bauteil-Tupel anzeigen, die von Konstrukteur Müller stammen (Join-Abfrage)	<pre>SELECT (Sachnummer, Benennung, Werkstoff, K.Name) FROM zpedb.Bauteil B, zpedb.Konstrukteur K WHERE B.K# = K.K# AND Name = ,Müller';</pre>
Tupel löschen	<pre>DELETE FROM zpedb.Bauteil WHERE Benennung = ,Motor';</pre>

Tabelle (T005dbtZ) Beispiele für SQL-Statements

Die in Tabelle T005dbtZ angeführten SQL-Statements können entweder *interaktiv* abgesetzt oder aber in eine Wirtsprache (z. B. C oder Pascal) eingebettet werden („*embedded SQL*“). Für Applikationsentwicklungen ist dies zwingend erforderlich, da SQL als rein deskriptive Datenbankanfragesprache über keine prozeduralen Programmierkonstrukte verfügt (if... then..., repeat... until... etc.). Ein solches C-Programm mit embedded SQL-Statements (Endung \*.csql) muss dann zunächst durch Precompilieren in herkömmlichen C-Code übersetzt werden, welcher daraufhin wie ein gewöhnliches Programm noch kompiliert und gelinkt werden muss.

Aufgrund der bereits mehrfach erwähnten Datenunabhängigkeit benötigen SQL-Statements und damit auch embedded SQL-Methodenprogramme keinerlei Anpassung bei einer Reorganisation der physischen Speicherungsstruktur einer Datenbank (z. B. Hinzufügen/ Wechsel einer Disk hat keinerlei Einfluss auf SQL-Statements, da diese keine Zugriffspfade beinhalten, vgl. Tab. 3.3).

Das Konzept des *dynamischen SQL* ermöglicht eine effiziente Programmentwicklung sowie eine grösstmögliche Flexibilität für die Anwendung. Zum Zeitpunkt der Programmentwicklung müssen hierbei die Argumente für die SQL-Statements (Spaltennamen, Tabellennamen, Auswahlbedingungen etc.) noch nicht bekannt sein und

angegeben („hart codiert“) werden, sondern können - ähnlich einer Programmvariablen - erst zur Laufzeit spezifiziert werden.

Als Beispiel sei die in Tabelle T005dbtZ angeführte Abfrage aller Konstrukteure der Abteilung ‚ZPE‘ genannt, wo mit dynamischem SQL der Wert ‚ZPE‘ nicht als Wert ins Programm codiert werden muss, sondern eine Variable definiert wird. Dieser Variable kann dann zur Laufzeit durch Benutzereingabe der Wert ‚ZPE‘ zugewiesen werden oder auch ein abweichender Wert, wie z.B. ‚ETH‘, wodurch das einmal programmierte SQL-Statement für mehrere verschiedene Abfragen nutzbar wird.

In vielen Fällen erfordern Benutzerzugriffe tabellenübergreifende bzw. tabellenverknüpfende Datenbankabfragen, die als Verbundoperationen (engl. Join) bezeichnet werden und häufig mittels dynamischem SQL implementiert werden. Hierbei müssen zwei Arten von Join-Operationen unterschieden werden, die in der Regel erheblich voneinander abweichende Ergebnisse liefern:

kartesisches Produkt	Tabellen werden ungeachtet von Attributwerten miteinander „multipliziert“, wodurch sich die resultierende Tupelzahl aus dem Produkt der Tupel sämtlicher beteiligter Tabellen ergibt
natürlicher Verbund	Tabellen werden über Wertegleichheit von Partnertupeln miteinander „verbunden“, wodurch sich die resultierende Tupelzahl immer kleiner oder höchstens gleich der höchsten Tupelzahl der beteiligten Tabellen ergibt.

Tabelle (T006dbtZ) Zwei Arten von Join-Operationen

Bild B010dbtZ und Bild B011dbtZ verdeutlichen den prinzipiellen Unterschied beider Join-Arten anhand dem Anwendungsbeispiel in Bild B006dbtZ. In der überwiegenden Mehrzahl der Fälle wird der Grund für einen Join in einer Fremdschlüsseldefinition begründet sein, die aufgrund ihrer Semantik fast immer eine Abfrage mittels natürlichem Verbund bedingt.



**Tabellen-Join als kartesisches Produkt** *(jeder mit jedem)*

**SELECT** Sachnummer, Benennung, Werkstoff, Name, Abteilung  
**FROM** Bauteil, Konstrukteur  
**WHERE** Sachnummer **LIKE** '470'

**Ergebnis:**

Sachnummer	Benennung	Werkstoff	Name	Abteilung
470-000 000.001	Auto	-	Müller	ZPE
470-000 000.001	Auto	-	Hub	ZPE
470-000 200.001	Getriebe	-	Müller	ZPE
470-000 200.001	Getriebe	-	Hub	ZPE
470-000 210.001	Motor	-	Müller	ZPE
470-000 210.001	Motor	-	Müller	ZPE
470-000 201.002	Getriebeflansch	ST37	Müller	ZPE
470-000 201.002	Getriebeflansch	ST37	Müller	ZPE
...	...	...	...	...

Bild (B010dbtZ) Tabellen-Join als kartesisches Produkt

**Tabellen-Join als natürlicher Verbund**  
*(nur Partnertupel mit entsprechendem Referenzwert)*

**Abfrage:** **SELECT** Sachnummer, Benennung, Werkstoff, Name, Abteilung  
**FROM** Bauteil, Konstrukteur  
**WHERE** Sachnummer **LIKE** '470' **AND** Bauteil.K#=Konstrukteur.K#

**Ergebnis:**

Sachnummer	Benennung	Werkstoff	Name	Abteilung
470-000 000.001	Auto	-	Müller	ZPE
470-000 200.001	Getriebe	-	Müller	ZPE
470-000 210.001	Motor	-	Hub	ZPE
470-000 201.002	Getriebeflansch	ST37	Hub	ZPE
...	...	...	...	...

Bild (B011dbtZ) Tabellen-Join als natürlicher Verbund

### 3.1 Triggerfunktionalität von SQL

Die modellinhärenten Integritätsbedingungen des Relationenmodells in Tabelle T002dbtZ werden wie bereits erwähnt vom Datenbanksystem automatisch sichergestellt. Für den Fall, dass eine der Integritätsbedingungen verletzt werden sollte, muss das Datenbanksystem geeignete Gegenmassnahmen treffen.

Eine Verletzung der Eindeutigkeit von Primärschlüsselwerten (Entity Integrity) muss von Beginn an unterbunden werden, so dass Insert- oder Update-Versuche mit derartigen Konsequenzen prinzipiell immer fehlschlagen müssen.

Sollte jedoch die Referential Integrity verletzt werden (durch Löschen oder Modifizieren eines referenzierten Parent-Tupels), kann von seiten der Anwendung ein unterschiedliches Verhalten wünschenswert sein. Anhand sogenannter Trigger lässt sich dies für Fremdschlüsseldefinitionen definieren, wobei gemäss Bild B012dbtZ drei Möglichkeiten bestehen.

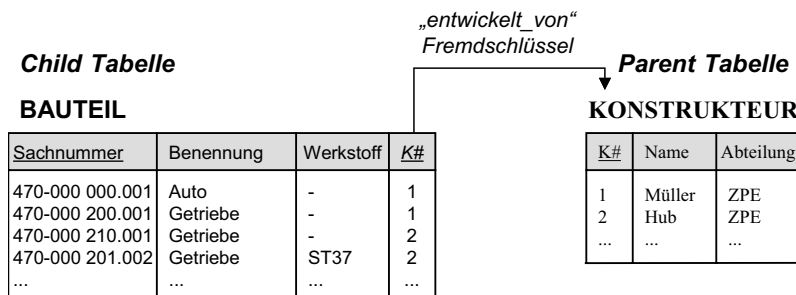


Bild (B012dbtZ) Triggermöglichkeiten einer Fremdschlüsseldefinition

Als Default für Fremdschlüsseldefinitionen verwenden Datenbanksysteme in der Regel den Trigger ON DELETE RESTRICT, der sich für sämtliche Beziehungen anbietet. Für Spezialisierungen (is\_a-Beziehungen) ist dagegen häufig ON DELETE CASCADE vorzuziehen.

Vorteilhaft an solchen Triggerdefinitionen sind nicht nur die Vereinfachungen für den Applikationsprogrammierer, sondern auch die daraus resultierende grössere Integrität des Datenbankschemas.

## 4 Transaktionsverwaltung von Datenbanksystemen

Eine Transaktion ist eine Folge von Datenbankoperationen, die eine Datenbank von einem konsistenten Zustand in einen wiederum konsistenten Zustand überführt anhand folgender Eigenschaften (ACID-Prinzipien):

- *Atomarität (atomicity)*  
Eine Transaktion wird entweder vollständig oder überhaupt nicht ausgeführt („alles oder nichts“).
- *Konsistenz (consistency)*  
Sämtliche bestehenden Konsistenzbedingungen, die während dem Verlauf einer Transaktion durchaus verletzt sein dürfen, müssen am Ende der Transaktion wieder erfüllt sein.
- *Isolation (isolation)*  
Korrektheit im Parallelbetrieb durch die Isolation von Informationselementen einer Transaktion gegenüber weiteren Zugriffen (Einbenutzersicht, Serialisierung).
- *Persistenz (durability)*  
Garantie für die Dauerhaftigkeit von Änderungen abgeschlossener Transaktionen auch bei Systemausfällen.

In funktioneller Hinsicht besteht eine Transaktionsverwaltung (vgl. [Wei-88]) aus folgenden Bereichen:

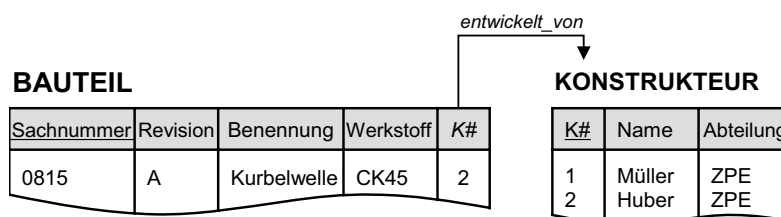
- *Mehrbenutzerkontrolle (Concurrency Control)*  
Synchronisation konkurrierender Informationszugriffe verschiedener Benutzer, die gleichzeitig Operationen über denselben Daten ausführen (z. B. durch das Zwei-Phasen-Sperrprotokoll 2PC).
- *Datensicherung (Recovery)*  
Wiederherstellung eines konsistenten Datenbankzustandes nach einem Systemausfall aufgrund von Hardware-, Software- oder Bedienungsfehlern (Logging-Prinzip).

Beide Bereiche sind von grundlegender Bedeutung für die PLM-Technologie, weshalb zumindest ein Verständnis für diese Problemstellung unerlässlich ist. Nachfolgend wird zunächst auf die Mehrbenutzerkontrolle in Datenbanksystemen eingegangen und im Anschluss hieran können die Aspekte der Datensicherung diskutiert werden.

Charakteristisch für den Mehrbenutzerbetrieb sind gleichzeitig eingehende Benutzeraufträge, die sich gegenseitig beeinflussen können. Als einführendes Beispiel soll wiederum der bekannte Datenbankentwurf mit den Tabellen Bauteil und Konstrukteur dienen, wobei lediglich die Bauteiltabelle um ein zusätzliches Attribut (Revision) erweitert

wurde (vgl. Bild B013dbtZ). Das Revisionsattribut soll rein klassifizierend sämtliche Modifikationen an einem Bauteiltupel dokumentieren und nach der Initialisierung auf den Wert ‚A‘ bei jedem Update programmtechnisch auf den nächsten Buchstaben im Alphabet inkrementiert werden (B, C, D etc.).

Bezüglich der bereits existierenden Kurbelwelle (Anfangsrevision A) sollen parallel zwei Transaktionen verschiedener Konstrukteure durchgeführt werden. In der Transaktion I ändert Konstrukteur Huber den Werkstoff von CK45 auf 16MnCr5, wodurch der Revisionszähler automatisch auf B gesetzt wird. In der Transaktion II führt Konstrukteur Müller eine Umbenummerung durch, um die vorläufig vergebene Sachnummer 0815 auf die endgültige Sachnummer abzuändern. Hierbei wird ebenfalls der Revisionszähler erhöht und zudem der Eigentümer gewechselt.



Transaktion I	Transaktion II
<p><b>Inhalt:</b> Konstrukteur Huber ändert den Werkstoff der Kurbelwelle von CK45 auf 16MnCr5. Der Revisionszähler wird automatisch um 1 auf ‚B‘ erhöht.</p>	<p><b>Inhalt:</b> Konstrukteur Müller führt eine Umbenummerung durch, um der unter der vorläufigen Sachnummer 0815 abgelegten Kurbelwelle die endgültige Sachnummer 470-000 210.002 zuzuweisen. Der rein klassifizierende Revisionszähler wird automatisch um 1 erhöht. Aufgrund der Neuvergabe der Sachnummer wird zudem automatisch der Eigentümer gewechselt.</p>

Bild (B013dbtZ) Anwendungsbeispiel für Transaktionskonflikt (Lost Update-Problematik)

Transaktion I	Transaktion II
<p>Ablauf:</p> <ol style="list-style-type: none"> <li>1) Begin Of Transaction (T<sup>I</sup>)</li> <li>2) Read (Kurbelwellentupel) (Ergebnis: ‚0815‘, ‚A‘, ‚Kurbelwelle‘, ‚CK45‘, 2)</li> <li>3)</li> <li>4)</li> <li>5) Update (Werkstoff, Revision) (Ergebnis im RAM: ‚0815‘, ‚B‘, ‚Kurbelwelle‘, ‚16MnCr5‘, 2)</li> <li>6)</li> <li>7) Write (Kurbelwellentupel)</li> <li>8) End Of Transaction (T<sup>I</sup>)</li> <li>9)</li> <li>10)</li> </ol>	<p>Ablauf:</p> <ol style="list-style-type: none"> <li>Begin Of Transaction (T<sup>II</sup>)</li> <li>Read (Kurbelwellentupel) Ergebnis: ‚0815‘, ‚A‘, ‚Kurbelwelle‘, ‚CK45‘, 2)</li> <li>Read (aktuelle K#) Ergebnis: 1)</li> <li>Update (Sachnummer, Revision, K#) (Ergebnis im RAM: ‚470-000 210.002‘, ‚B‘, ‚Kurbelwelle‘, ‚CK45‘, 1)</li> <li>Write (Kurbelwellentupel)</li> <li>End Of Transaction (T<sup>II</sup>) (Gesamtergebnis in DB: ‚470-000 210.002‘, ‚B‘, ‚Kurbelwelle‘, ‚CK45‘, 1)</li> </ol>
<p><b>Resultat:</b> Transaktion erfolgreich beendet. Änderungen (von Werkstoff und Revision) gehen jedoch verloren.</p>	<p><b>Resultat:</b> Transaktion erfolgreich beendet. Sämtliche Änderungen (Sachnummer, K#) sind unter dem „nächsten“ Revisionsindex ‚B‘ abgelegt.</p>

Bild (B013dbtZ) Anwendungsbeispiel für Transaktionskonflikt (Lost Update-Problematik)

Als Ergebnis einer korrekten Abwicklung zunächst der einen und daraufhin der anderen Transaktion wäre zu erwarten, dass die Kurbelwelle einen neuen Werkstoff, eine neue Sachnummer, einen Revisionszähler ‚C‘ sowie einen neuen Eigentümer aufweist. Entsprechend dem in Bild B013dbtZ skizzierten Ablauf werden die beiden Transaktionen jedoch parallel durchgeführt und entgegen der obigen Erwartung besitzt die Kurbelwelle nach wie vor den alten Werkstoff (CK45) und führt als Revisionszähler ein ‚B‘. Als Folge der Parallelbearbeitung sind also die Änderungen der Transaktion I verloren gegangen, was in der Literatur als sogenanntes Lost Update-Problem bekannt ist (vgl. [Vos-95]). Eine verallgemeinerte Darstellung der Problematik zusammen mit zwei weiteren charakteristischen Problemfällen des Mehrbenutzerbetriebes zeigt Bild B014dbtZ.

<b>Lost Update</b>			<b>Phantom (unrepeatable Read)</b>			<b>Dirty Read</b>		
$t_i$	Zeitpunkt	$t_{ij}$	$t_i$	Zeitpunkt	$t_{ij}$	$t_i$	Zeitpunkt	$t_{ij}$
BOT ( $t_i$ )	1		BOT ( $t_i$ )	1		BOT ( $t_i$ )	1	
	2		sum:=0	2		r(x)	2	
	3		r(x)	3		u(x)	3	
	4	BOT ( $t_{ij}$ )	sum:=sum+x	4		w(x)	4	
u(x)	5		r(y)	5			5	BOT ( $t_{ij}$ )
	6		sum:=sum+y	6			6	r(x)
w(x)	7			7	BOT ( $t_{ij}$ )		7	u(x)
EOT ( $t_i$ )	8			8	r(x)	abort	8	
	9	w(x)		9	x:=x		9	w(x)
	10	EOT ( $t_{ij}$ )		10	w(x)		10	EOT ( $t_{ij}$ )
				11	r(z)			
				12	z:=z+10			
				13	w(z)			
				14	commit			
				15	EOT ( $t_{ij}$ )			
				16	r(z)			
				17	sum:=sum+z			
				18	EOT ( $t_i$ )			

↳ der von  $t_i$  durchgeführte Update geht verloren, weil er von  $t_{ij}$  überschrieben wird.

↳ die von  $t_i$  ermittelte Summe ist falsch, da bereits ein neuer z-Wert gelesen wird (obwohl die Summe konstant bleibt und der DB-Inhalt korrekt ist).

↳ der von  $t_{ij}$  gelesene Wert ist aufgrund des nachfolgenden Abort von  $t_i$  inkonsistent, da er eigentlich nie in der DB existiert hat!

Bild (B014dbtZ) Typische Probleme im Mehrbenutzerbetrieb (Concurrency Control); BOT = Begin Of Transaction; EOT = End Of Transaction; r = read; u = update; w = write

Die Transaktionsverwaltung von DBMS-Systemen stellt mittels Sperrmechanismen (z. B. 2 Phasen Sperrprotokoll) sicher, dass die interne Abarbeitung von Lese- und Schreib-Aktionen korrekt durchgeführt wird und immer wieder in einen konsistenten DB-Zustand mündet, so dass derartige Problemfälle ausgeschlossen werden können.

Wichtig für die PLM-Technologie ist die Transaktionsverwaltung aber nicht nur für die reine Software-Entwicklung, sondern ebenso für das unternehmensspezifische Customize, für datenbanksystemspezifische Unterschiede oder auch für die Analyse von Systemfehlern. Darüber hinaus stellt die Transaktion auch die grundlegende Basiseinheit für die nachfolgend erläuterten Aspekte der Datensicherung dar.

## 5 Datensicherung in PLM-Systemen

Sicherheit und Verfügbarkeit des Datenbestandes sind ein wichtiger Aspekt im Produktdatenmanagement, da schlussendlich sämtliche Produktinformationen im PLM-System hinterlegt werden und ein jeder Benutzer diese für seine tägliche Arbeit benötigt. Eine absolute Sicherheit kann es auch für PLM-Applikationen nicht geben und im Falle eines Versagens können weittragende Folgen entstehen, wenn beispielsweise ein Entwicklungsteam von mehreren hundert Ingenieuren nicht mehr arbeiten kann, Daten verliert und bis zur Behebung eines Fehlerfalles mehrere Tage zurückgeworfen wird. Aufgrund der heute üblichen, strengen Projektterminierung und der grossen Abhängigkeit von einer PLM-Applikation ist daher ein sogenanntes „fail-safe“-Design unabdingbar, um im Versagensfall auf einen früheren bekannten und konsistenten Stand der PLM-Datenbank zurückkehren zu können. Die Grösse des Datenverlustes, welche (in Arbeitszeit der Benutzer gemessen, wie z. B. 1 Tag) eintreten kann, ist dabei vom geleisteten Aufwand abhängig.

Voraussetzung für das Zurückkehren auf einen früheren Stand ist das periodische Durchführen eines sogenannten Backups zu festgelegten Zeitpunkten (z. B. täglich, alle drei Tage, wöchentlich). Unter einem Backup ist hierbei das Kopieren sämtlicher Sekundärspeicherdaten (Magnetplatte/Disk) auf einen Offline-Speicher (Magnetband, DAT-Tape) zu verstehen, welcher sinnvollerweise an einem „sicheren“, örtlich getrennten Standort gelagert werden sollte.

Um eine zweckmässige Backup-Strategie für den bestimmungsgemässen Einsatz im Fehlerfall ausarbeiten zu können, müssen vorweg sämtliche möglichen Problemszenarien in einer PLM-Architektur analysiert werden. Bild B015dbtZ zeigt eine stark vereinfachte Darstellung, wobei insbesondere zwischen einer *Hauptspeicherebene* (i.d.R. verteilt auf drei Maschinen), einer *Sekundärspeicherebene* mit zwei Hauptbestandteilen und einer *Archivebene* für die Offline-Sicherung unterschieden wird.

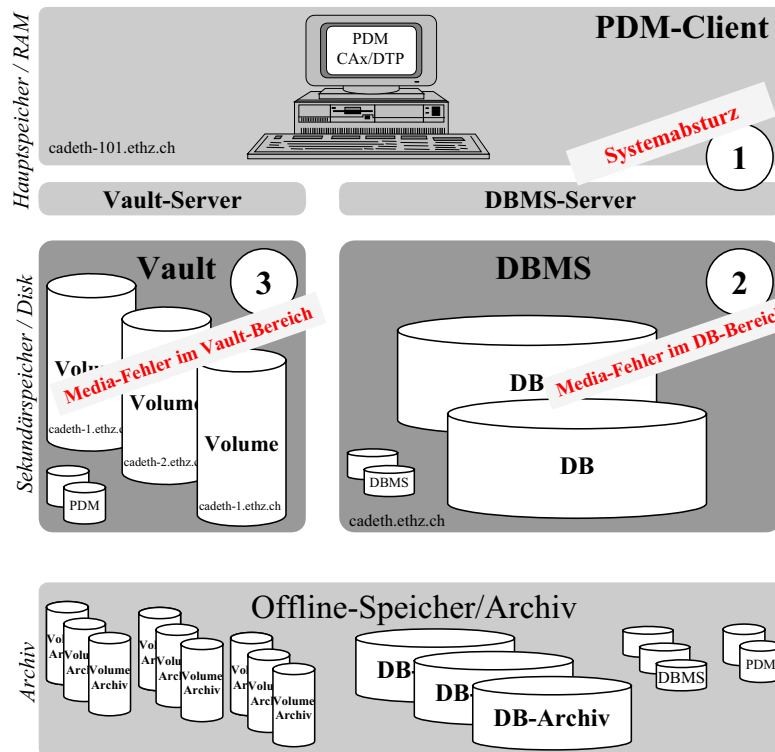


Bild (B015dbtZ) Analyse der Fehlermöglichkeiten in PLM-Applikationen

Demzufolge ist als *einfachster Fehlerfall* ein Systemabsturz anzunehmen, der überwiegend den PLM-Client betreffen wird, welcher zumeist vom Benutzer selbst wieder aufgestartet werden kann. Sollte jedoch das Datenbanksystem (DBMS = DataBase Management System) oder gar das Betriebssystem (OS = Operating System) versagen, muss ein Systemadministrator einspringen. Als Konsequenz des Systemausfalles ist die aktuelle Benutzeraktion natürlich fehlgeschlagen und die Hauptspeicherinhalte sind verloren. Die folgenden Betrachtungen beschränken sich hierbei auf Systemfehler, die zumindest den PLM-Client betreffen und damit direkte Auswirkungen auf die PLM-Datenbank und die übrigen Benutzer haben.

Nicht betrachtet werden Fehler eingekapselter CAx- oder DTP-Applikationen, da hierbei schlimmstenfalls ein einzelnes oder auch mehrere Files korruptiert werden, die bekannt sind und vom Systemadministrator nötigenfalls einzeln vom Backup zurückgeladen werden können, ohne die übrigen Benutzer zu beeinträchtigen.

Als *zweite Fehlerklasse* sind Media-Fehler im Datenbankbereich anzuführen (z.B. Head Crash oder irrtümlicherweise gelöschte Files). Hierbei handelt es sich um ein ernstes Problem, welches ein Zurückkla-



den des letzten Backups erfordert und je nach gewählter Backup-Strategie bereits mit unwiederbringlichen Datenverlusten einer bestimmten Zeitdauer verbunden sein kann.

Die *dritte Fehlerklasse* und damit der schlimmste Problemfall überhaupt sind Media-Fehler im Volume-Bereich, wo die Files eingekapselter Fremdapplikationen (CAx, DTP) gespeichert sind. Hier muss ebenfalls der letzte Backup zurückgeladen werden, womit aufgrund fehlender Datenbankfunktionalität jedoch in der Regel zwingend ein Rücksetzen auf den Stand des Backup-Datums verbunden ist, was unwiederbringliche Datenverluste bedeutet.

Tabelle T007dbtZ gibt einen zusammenfassenden Überblick über die Fehlerklassen von PLM-Applikationen. Zusätzlich zu den bereits erwähnten Fehlerklassen eins bis drei sind auch Transaktionsfehler angeführt, die durch einen Deadlock, Applikationsfehler (z.B. Divisions durch 0) oder Benutzerfehler entstehen. Obwohl Transaktionsfehler sehr häufig auftreten, stellen sie für den Endanwender einer Datenbank-Applikation kein Problem dar. Da sie als „innerer“ Fehler von der Transaktionsverwaltung erkannt werden, können sie durch einen Rollback (UNDO) behoben werden, ohne einen manuellen Eingriff zu erfordern.

Fehlerklassen von PLM-Applikationen

Fehlerklasse	Ursache	Häufigkeit	Massnahme
0. Transaktionsfehler	Deadlock, Applikations oder Benutzerfehler	stündlich	Transaction Abort <ul style="list-style-type: none"> <li>Zurücksetzen (Rollback) einer Transaction (mittels UNDO).</li> </ul>
1. Systemfehler	Absturz PLM ,DBMS, OS sowie Hardware-Problem oder Stromausfall	wöchentlich	Crash Recovery <ul style="list-style-type: none"> <li>Zurücksetzen (Rollback) aller nicht abgeschlossenen Transaktionen (mittels UNDO).</li> <li>Wiederholen aller abgeschlossenen Transaktionen (mittels REDO).</li> </ul>

Tabelle (T007dbtZ) Übersicht über Fehlerklassen von PLM-Applikationen

Fehlerklassen von PLM-Applikationen

Fehlerklasse	Ursache	Häufigkeit	Massnahme
2. Media-Fehler im DB-Bereich	Speicherfehler durch z.B. Head Crash, irrtümlich gelöschte Files, korruptes Filesystem oder höhere Gewalt durch Feuer, Wasserschaden etc.	jährlich	Media Recovery (nur DB-seitig) <ul style="list-style-type: none"> <li>• Zurückladen des zuletzt gesicherten DB-Zustandes mittels Archiv-DB (vom Band).</li> <li>• Wiederholen aller abgeschlossenen Transaktionen (REDO) zwischen Backup-Datum und Fehlerzeitpunkt durch „Vorwärtsrollen“ mittels Log-Dateien (auf Disk) und ggf. Archiv-Log (vom Band).</li> <li>• *Bei richtiger Backup-Strategie KEIN Datenverlust!</li> </ul>
3. Media-Fehler im Volume-Bereich	Speicherfehler durch z.B. Head Crash, irrtümlich gelöschte Files, korruptes Filesystem oder höhere Gewalt durch Feuer, Wasserschaden etc.	jährlich	Media Recovery (gesamthaft) <ul style="list-style-type: none"> <li>• Zurückladen des zuletzt gesicherten DB-Zustandes mittels Archiv-DB (vom Band).</li> <li>• Zurückladen des zuletzt gesicherten Volume-Zustandes mittels Archiv-Volume (vom Band).</li> <li>• *Da zwingend auf den Stand des letzten Volume-Backups zurückgegangen werden muss, ist kein „Vorwärtsrollen“ möglich. Ein gewisser Datenverlust (i.d.R. 1 Tag) ist unumgänglich!</li> </ul>

Tabelle (T007dbtZ) Übersicht über Fehlerklassen von PLM-Applikationen

Das Pendant eines Backups im Fehlerfall wird als Recovery bezeichnet. Unter *Recovery* versteht man das Wiederherstellen eines konsistenten Datenbankzustandes nach dem Eintreten eines Fehlers. Die Grundlage hierfür ist Redundanz in Form eines Backups der Datenbank sowie der Protokollierung sämtlicher Vorgänge in Log-Dateien. Dadurch wird gewährleistet, dass bereits abgeschlossene Transaktionen dauerhaft gespeichert bleiben und dass nicht beendete Transaktionen keine (dauerhaften) Spuren in der Datenbank hinterlassen. Bild B016dbtZ verdeutlicht das Prinzip der Protokollierung in Datenbanksystemen mittels Log-Dateien.

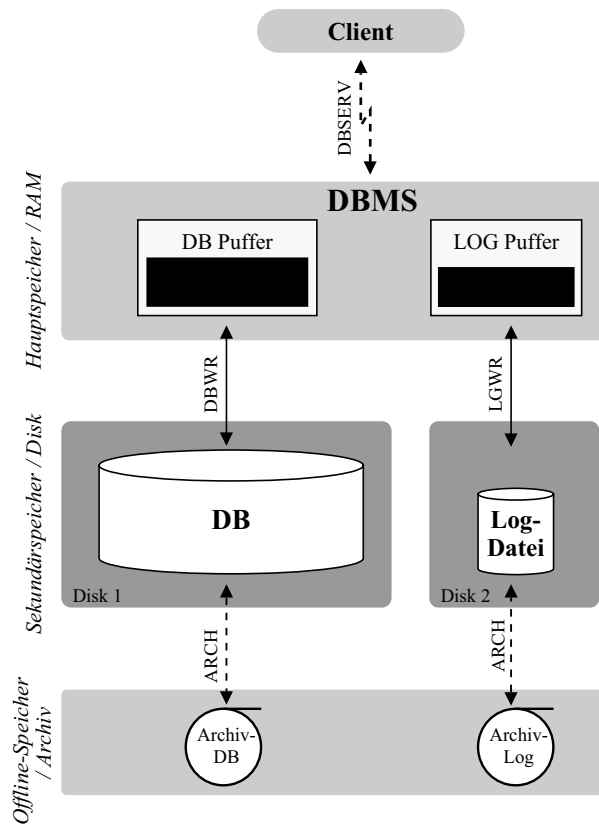


Bild (B016dbtZ) Protokollierung in Datenbanksystemen als Grundlage der Recovery

Im Hauptspeicher werden in einem sogenannten DB-Puffer möglichst sämtliche benötigten Datenbankinhalte geführt, um Operationen direkt im Hauptspeicher ausführen zu können. Dadurch werden die erheblich langsameren Sekundärspeicherzugriffe (Disk-I/O) reduziert und schlussendlich die Performance erheblich gesteigert. Da das Zurückschreiben auf Platte hierbei asynchron zu unbekanntem Zeitpunkt erfolgt (Prozess: DBWR), resultiert daraus eine gewisse Unsicherheit über den Datenbankzustand auf dem SekundärspeichermEDIUM. Einerseits kann je nach Systemauslastung eine abgeschlossene Transaktion vom DB-Puffer im Hauptspeicher noch nicht auf die Platte zurückgeschrieben sein, andererseits kann die Notwendigkeit bestehen, eine noch nicht beendete Transaktion bereits auf Platte herauszuschreiben. Um dies hinsichtlich der obigen Zielsetzung handhaben zu können, wird für jede Operation im DB-Puffer ein zugehöriger Log-Satz im Log-Puffer abgelegt. Dieser beinhaltet sowohl UNDO- als auch REDO-Informationen, um jederzeit ein Rücksetzen oder Wiederholen der Transaktionen gewährleisten zu können. Im Gegensatz zum

Transfer zwischen DB-Puffer und DB ist die Übertragung vom Log-Puffer zur Log-Datei (Prozess: LGWR) „sicherer“ geregelt gemäss folgenden zwei Regeln:

- Atomaritätsregel:  
Bevor eine Veränderung (einzelne Aktion(en) einer Transaktion) in die persistente Datenbank (DB auf Platte) eingebracht wird, muss die zugehörige UNDO-Information in die Log-Datei (DB-Log auf Platte) gezwungen werden („WAL“-Prinzip: Write Ahead Log).
- Persistenzregel:  
Vor dem Commit einer Transaktion müssen alle zugehörigen REDO-Informationen in die Log-Datei (DB-Log auf Platte) gezwungen werden.

Sowohl von der Datenbank als auch von den Log-Dateien wird periodisch ein Backup auf einen Offline-Speicher gezogen (Prozess: ARCH), woraus die sogenannte Archiv-DB und das Archiv-Log entstehen.

Im Falle eines *Systemfehlers* muss grundsätzlich mit folgender Ausgangslage gerechnet werden:

- Sämtliche Hauptspeichereinhalte sind flüchtig und gehen verloren (DB-Puffer, Log-Puffer).
- Die auf Sekundärspeicher befindliche Datenbank ist in einem inkonsistenten Zustand.

Abhilfe schafft hier die sogenannte Crash Recovery, wodurch die Datenbank (DB) mittels der Log-Datei(en) einem globalen UNDO/REDO unterzogen wird. Resultat ist ein wiederum konsistenter Zustand und im Falle des Betriebes im sogenannten Archiv-Modus keinerlei Datenverlust.

Im Falle eines *Media-Fehlers* der Datenbank muss diese mittels der Archiv-Dublikate (Archiv-DB und Archiv-Log) wieder hergestellt werden. In der Regel werden zudem noch die Log-Dateien verfügbar sein, welche idealerweise auf separaten und gespiegelten Medien vorliegen und infolgedessen ebenfalls als „sicher“ anzusehen sind. Mittels der sogenannten Media Recovery-Funktionalität von Datenbanksystemen kann auch in diesem Fehlerfalle wieder ein konsistenter Datenbankzustand hergestellt werden - ohne Datenverluste, falls der Archiv-Modus betrieben wird.

Für den Backup eines PLM-Systems kommt zur DB-Thematik die Volume-Problematik hinzu, wofür entweder zwei separate Backup Prozeduren (eine für die Datenbank und eine für das Volume) durchgeführt werden oder es ist eine Utility erforderlich, die für beide Komponenten einen Backup erstellt. In jedem Fall muss unbedingt beachtet werden, dass Volume und Datenbank korrespondieren müs-

sen, d.h. deren Backup zur gleichen Zeit erfolgen muss. Dies impliziert, dass vom Beginn des Backups an Volume und/oder Datenbank bis zum Ende des Backups (welcher für Volume und Datenbank allgemein verschiedene Dauern haben wird) keine Veränderungen durch Schreiboperationen erfolgen.

Im Fehlerfall müssen Volume und Datenbank nach einem Recovery-Prozess ebenfalls wieder auf einem identischen Aktualitätsstand sein. Hervorzuheben ist hierfür die Diskrepanz von Volume und Datenbank bezüglich ihrer Recovery-Funktionalität:

- Volume-Recovery:  
„Point-in-time“-Charakteristik, d.h. im Fehlerfall kann immer nur auf das Datum des letzten Backups zurückgegangen werden, was in der Regel immer mit einem Datenverlust verbunden ist.
- Datenbank-Recovery:  
„Point-of-failure“-Charakteristik, d.h. im Fehlerfall kann der Datenbankzustand wieder bis zum Zeitpunkt des Fehlers rekonstruiert werden, wodurch Datenverluste in der Regel ausgeschlossen werden können.

Da Volume und Datenbank normalerweise auf unterschiedlichen Maschinen, zumindest jedoch auf unterschiedlichen Disks gehalten werden, ist die Wahrscheinlichkeit relativ gross, dass nur eine der beiden Komponenten ausfällt. Ein gleichzeitiger Media-Ausfall beider Komponenten kann z.B. in einem Katastrophenfall ebenfalls eintreten.

Der mit einem Systemfehler einhergehende Datenverlust ist von vielerlei Faktoren abhängig und umgekehrt proportional zu den Vorsorge-Massnahmen und insbesondere dem Backup-Aufwand. Die Recovery-Funktionalität von DB-Systemen mit ihrer „Point-of-failure“-Charakteristik ist jedoch ein ganz entscheidender Vorteil und zugleich eine zwingende Voraussetzung, um Business-kritische Prozesse auf sogenannte „Mission critical“ IT-Lösungen abbilden zu können, deren Verfügbarkeit garantiert werden muss.

## 6 Verteilung in PLM-Systemen

Globale Unternehmen, virtuelle Fabriken und eine Reihe weiterer aktueller Schlagwörter verdeutlichen den Trend zur verteilten Produktentwicklung über mehrere Standorte hinweg, die mittels moderner Informationstechnologien zusammenarbeiten und koordiniert werden. Immer häufiger finden sich Beispiele für standortübergreifende Produktentwicklungsprozesse, in welchen örtlich getrennte Projektteams gemeinsam ein rechnerinternes Produktmodell erstellen, obwohl sie eventuell über grosse Entfernungen auseinander liegen oder gar auf unterschiedlichen Kontinenten beheimatet sind. Seinen Anfang nahm diese Entwicklung in der örtlichen Trennung von Entwicklungs- und Produktionsstandorten, um letztere in Billiglohnländer auslagern zu können. Mittlerweile finden sich zunehmend auch innerhalb der Entwicklung örtliche Aufsplittungen oder auch Zusammenschlüsse, wo einzelne Prozessschritte (z. B. Produktkonstruktion mittels CAD und CAM-Herstellprozessdefinition) an verschiedenen Standorten erfolgen und rein digital koordiniert werden [Mar-97]. Anhand verteilter PLM-Systeme können hierbei die Rationalisierungsmöglichkeiten durch örtliche Trennungen bzw. Zusammenschlüsse genutzt werden, ohne die rechnerinternen Produktinformationen - und damit das Know-how - aus der Hand zu geben. Da dies eine interessante Chance darstellt, um Industriestandorte, wie beispielsweise die Schweiz, trotz hohem Lohnniveau langfristig zu sichern, sollen nachfolgend die Konzepte der Verteilung in PLM-Systemen vorgestellt werden, wofür zunächst die diesbezüglichen Anforderungen zu klären sind.

### 6.1 Anforderungen an die PLM-Verteilungsfunktionalität

Ein realistisches Anwendungsszenario einer verteilten Systemarchitektur besteht aus mehreren Standorten, die herkömmlich über ein lokales Netzwerk (lokal Area Network, LAN) oder ein globales (Global Area Network, WAN) verbunden sind sowie Standorte, die über keinen direkten Netzwerkanschluss verfügen (z. B. ein Zulieferer). Problematisch für die über ein WAN zugreifenden Benutzer sind hierbei Zuverlässigkeit (d.h. Verfügbarkeit) und Performance des WAN. Desweiteren können Laufzeitprobleme entstehen, falls eine grosse Zahl von Benutzern (mehrere Tausend) hochgradig interaktiv auf einen zentralen Datenbank-Server zugreifen.

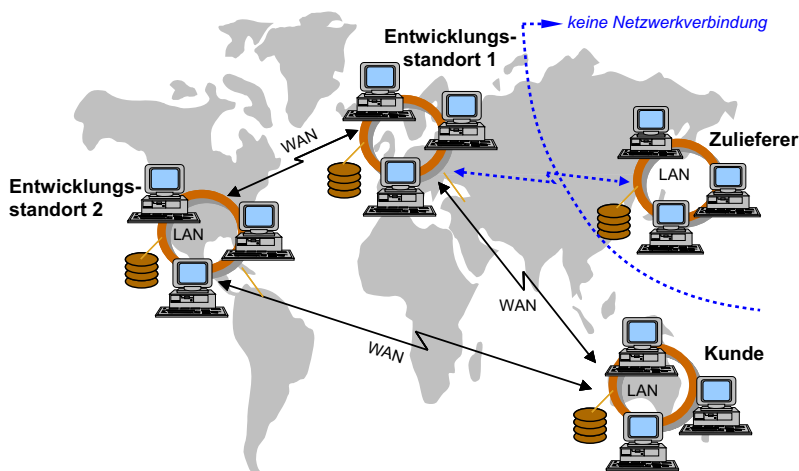


Bild (B017dbtZ) Herkömmliches Anwendungsszenario ohne Verteilungsfunktionalität

Für die Unterstützung derartig verteilter Anwendungsszenarien durch mehrere autonome PLM-Systeme, die über eine geeignete Verteilungsfunktionalität verfügen, müssen folgende Gesichtspunkte beachtet werden:

- *Anbindungsarten:*  
Benutzer können entweder direkt über ein LAN zugreifen, über WAN angebunden sein oder einem Standort ohne Netzwerkverbindung, d.h. einer separaten Installation angehören.
- *Datenklassen:*  
Es müssen sowohl lokale Daten (die nur innerhalb eines Standortes interessieren) als auch globale Daten (die an mindestens zwei Standorten verfügbar sein müssen) unterstützt werden.
- *Abstraktionsebenen:*  
Es müssen sowohl relational verwaltete Metadaten als auch eingekapselte Files externer CAx-/DTP-Applikationen unterstützt werden.
- *Aufbauorganisation/Benutzerdefinition:*  
Die einzelnen Standorte haben in der Regel unterschiedliche Gruppen-, Rollen- und Benutzerdefinitionen, wodurch beispielsweise unterschiedliche Benutzer an verschiedenen Standorten identische User ID's haben können.
- *Ablauforganisation:*  
Die einzelnen Standorte haben in der Regel abweichende Freigabe- und Änderungsabläufe sowie Statusdefinitionen.

- *Customize:*  
PLM-Systeme haben eine offene und flexible Systemarchitektur und werden allgemein für jeden Standort spezifisch konfiguriert.
- *lokale Autonomie:*  
Falls die Netzwerkverbindung zu einem oder mehreren Standorten unterbrochen ist, muss die lokale Funktionsfähigkeit erhalten bleiben.
- *Ortstransparenz:*  
In einer verteilten PLM-Systemumgebung müssen sich Benutzer nicht darum kümmern, wo interessierende Objektdaten physisch gespeichert sind.
- *Gleichberechtigung:*  
Die Standorte sollen gleichberechtigt sein, damit an jeder Stelle in eigener Regie Objektdaten erzeugt und modifiziert werden können.
- *Objektidentität:*  
Die Objekte müssen systemintern zwingend eine standortübergreifend unterschiedliche Identifikation haben. Darüber hinaus muss auch die Eindeutigkeit der benutzerseitigen Identifikation mittels der Sachnummerung gewährleistet werden.

Auch von seiten der Datenbanksystemanbieter existieren Konzepte zur Unterstützung verteilter Datenbanken, was von PLM-Applikationen genutzt werden könnte. Aus verschiedenen Gründen ist die von Datenbanksystemen angebotene Verteilungsfunktionalität jedoch ungenügend für verteilte PLM-Architekturen:

- Sämtliche datenbankseitige Verteilungsfunktionalität (Zwei-Phasen-Commit-Protokoll 2PC, synchrone Replikation, DB-Trigger) setzen eine jederzeit vollumfänglich funktionsfähige Netzwerkverbindung zu sämtlichen Standorten voraus.
- Keine Unterstützung für die Handhabung externer Files eingekapselter CAx-/DTP-Applikationen, wofür 3GL-Funktionsaufrufe in DB-Triggern erforderlich wären.
- DB-Trigger bieten wenig Flexibilität bezüglich Customize (kein dynamisches SQL, keine Generierung zur Laufzeit).
- Die Verteilungsfunktionalität von Datenbanksystemen ist herstellerspezifisch, weshalb ein PLM-System, das mehrere Datenbanksysteme unterstützt, jeweils spezifische Schnittstellen benötigen würde.

Folglich ist hierfür eine Lösung auf PLM-Systemebene erforderlich, wofür im folgenden die grundlegenden Verteilungskonzepte vorgestellt werden.



## 6.2 Verteilungskonzepte in PLM-Systemen

Das meistgenutzte und zweckmässigste Verteilungskonzept basiert auf systemseitig kontrollierter Datenredundanz mit Hilfe von Replikationsmechanismen. Hierbei werden von einem Objekt eine oder mehrere Read-Only-Kopien für weitere Standorte erstellt. Das Ursprungsobjekt verbleibt an einem Standort (Site) und wird hierdurch lediglich zum „global bekannten“ Objekt, da sämtliche Hoheitsrechte zur Modifikation nur an dieser einen Stelle verbleiben. Das nach wie vor mit Schreibrecht ausgestattete Ursprungs-Objekt wird häufig als „Master“ bezeichnet, da dieses immer den aktuellen Stand widerspiegelt. Demgegenüber sind die Read Only-Kopien als ein „Schnappschuss“ des Master Objektes anzusehen, der zu einem bestimmten Zeitpunkt erstellt wurde und nicht zwingend den gegenwärtigen Zustand des Masters wiedergeben muss. Wird ein Master Objekt nach einer Replikation modifiziert, so hat dies auf Read Only-Kopien zunächst keinen Einfluss. Read Only-Replikat sind infolgedessen potentiell immer veraltet und müssen durch einen asynchronen Aktualisierungsvorgang wieder mit ihrem Master synchronisiert werden. Kernprinzip der Verteilung ist somit die Replikation von Master Objekten in Form von Read Only-Kopien.

Um eine möglichst grosse Unabhängigkeit von der Verfügbarkeit eines Netzwerkzugriffs zu jedem der eingebundenen Standorte zu erreichen und um die Antwortzeiten globaler Datenabfragen sowie unnötige Redundanzen zu minimieren, wird der Replikationsprozess oftmals in mehrere Schritte und Datenbanken unterteilt:

- *Publizieren:*  
Bevor ein zunächst rein lokales Objekt von einem weiteren Standort gesehen werden kann, muss dieses publiziert werden. Publizieren bedeutet hierbei das Ablegen einiger wichtiger reduzierter Objektinformationen (Sachnummer, Name, Site etc.) in einem global bekannten Publikations-Server, der eine Art Objektverzeichnis darstellt, vergleichbar mit einer Litfassäule.
- *Globale Datenabfrage:*  
Daraufhin kann von einem anderen Standort eine globale Datenabfrage (remote query) an den Publikationsserver gestellt werden, der als Ergebnis unter anderem die obigen reduzierten Objektinformationen zurückgibt (sog. Publication Record, ggf. ebenfalls bereits veraltet).
- *Objektimport:*  
Falls Interesse an einem publizierten Objekt besteht, kann dieses in die eigene PLM-Datenbank importiert werden. Hierfür

wird die Anfrage mittels der Standort-Information im Publication Record direkt an das PLM-System weitergeleitet, wo das Master Objekt liegt und vom dortigen Import-/Export-Server eine aktuelle Read Only-Kopie exportiert wird, die abschliessend lokal importiert werden kann.

- *Synchronisation*:  
Gegebenenfalls kann bzw. muss die Read Only-Kopie an neue Änderungszustände des Master Objektes aktualisiert werden, was entweder manuell vom Benutzer per re-import oder systemseitig, d.h. asynchron und periodisch erfolgt.

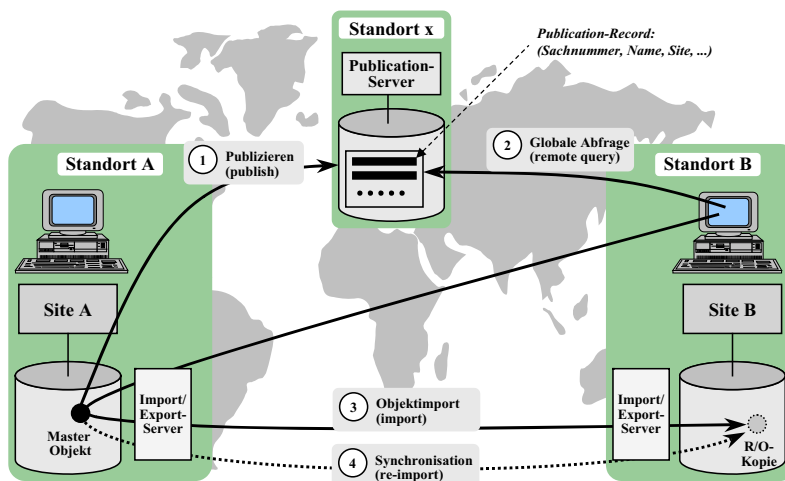


Bild (B018dbtZ) Prinzip der Verteilung in PLM-Applikationen

Das Grundprinzip der Verteilung wird durch Bild B018dbtZ verdeutlicht. Einer detaillierten Betrachtung bedürfen insbesondere noch die Vorgänge

- Synchronisation/Aktualisierung,
- lokales Löschen und
- globales Löschen.

Eine *Synchronisation/Aktualisierung* sollte von seiten des Bedürfnisträgers, d.h. der Site mit der Read Only-Kopie gewählt werden können. Veranlassen sollte einen Aktualisierungsprozess jedoch der Eigentümer, d.h. die Site mit dem Master Objekt, da dort Modifikationen am einfachsten beobachtet werden können und dadurch unnötiger Netzbetrieb vermieden wird. Unter Wahrung der Datenhoheit kann zunächst ein Master Objekt nach Belieben lokal modifiziert werden. Die Aktualisierung der Read Only-Kopien im Netzwerk muss daher in

jedem Falle asynchron erfolgen, einerseits aus Performance-Gründen, zum anderen können benötigte Netzwerkverbindungen temporär gestört sein. In der Regel werden Aktualisierungen periodisch (z. B. täglich, alle 3 Tage) oder in Abhängigkeit bestimmter Ereignisse (z. B. Freigabe) vorgenommen, was im Hintergrund abläuft und dem Benutzer verborgen bleibt. Anstelle der modifizierten Objektdaten, welche unter Umständen sehr umfangreich sein können, ist auch das bloße Informieren mittels einer automatisierten E-Mail möglich, anhand welcher ein Halter einer Read Only-Kopie entscheiden kann, ob er seine Kopie aktualisieren will oder nicht. Voraussetzung für jede Aktualisierungs- bzw. Synchronisationsform ausgehend vom Master Objekt ist jedoch das dortige Speichern derjenigen Sites, welche eine Read Only-Kopie „gezogen“ haben („exported to“-Information).

Unter *lokalem Löschen* versteht man das Löschen einer Read Only-Kopie. Hierbei muss das Löschen (ebenfalls asynchron) an die Site mit der Master Kopie zurückgemeldet werden, damit diese ihre „exported to“-Informationen aktualisieren kann. Um den Standort der Master Kopie herausfinden zu können, muss daher für jede Read Only-Kopie immer auch festgehalten werden, woher ein Objekt importiert wurde bzw. wer das Master Objekt besitzt („*owing site*“-Information).

*Globales Löschen* betrifft dagegen das Master Objekt und erfordert eine gewisse Vorarbeit, da ein Master Objekt global bekannt ist und somit eine globale Ressource darstellt. Vorgängig muss daher einerseits mittels Unpublish der Publication Record vom Publikations-Sever entfernt werden, wodurch auch die beim Master Objekt gespeicherte „published to“-Information gelöscht wird. Andererseits müssen sämtliche Halter von Read Only-Kopien im Netzwerk informiert und aufgefordert werden, ihre Read Only-Kopien lokal zu löschen. Erst nachdem dies vollumfänglich durchgeführt wurde und sämtliche globalen Referenzen (die beim Master Objekt als „exported to“-Informationen gespeichert waren) gelöscht sind, kann das Master Objekt als nun wiederum rein lokale Ressource global gelöscht werden.

Selbstverständlich kann darüber hinaus auch das Eigentümerrecht an eine andere Site wechseln. Hierbei wird das Master Objekt importiert und an der ursprünglichen Owning site verbleibt lediglich eine Read Only-Kopie. Hinsichtlich Autorisierung und Prozessmanagement stellen Produktentwicklungen bzw. Produktstrukturen mit verteilten Master Objekten den höchsten Komplexitätsgrad dar.

Unter den kommerziell verfügbaren PLM-Systemen wird zunehmend Verteilungsfunktionalität in mehr oder weniger grosser Übereinstimmung mit den vorstehenden Konzepten angeboten. Basis einer jeden Verteilung ist die Replikation von Read Only-Kopien. Falls dabei

kein Publikations-Server zum Einsatz kommt, muss dies nicht unbedingt eine Abkehr vom Prinzip des Publizierens bedeuten, sondern vereinfacht dieses auf eine rein lokale Aktion und verkompliziert dagegen globale Abfragen, die dann über das gesamte Netzwerk (d.h. zu mehreren Sites) erfolgen müssen. Eine Variation des Verteilungskonzeptes bedeutet somit in der Regel lediglich einen etwas anders gelagerten Kompromiss mit leicht verschobenen Vor- und Nachteilen.

Ein ernstes Problem für verteilte PLM-Datenbanken stellt die Datensicherung dar, wobei insbesondere die vorgängig bereits diskutierten Media-Fehler zu beachten sind. Denkbar wäre das Importieren eines Master Objektes kurz vor einem Fehlerfall, was das Zurückspielen eines Backups erfordert. Hiernach wäre das Master Objekt, welches am Ursprungsort nur noch als Read Only-Kopie existiert, nicht mehr vorhanden. Datenverluste können bei verteilten Systemen daher besonders negative Folgen haben und müssen mit einer sehr leistungsfähigen Backup- und Recovery-Strategie soweit als möglich vermieden werden. Die grosse Bedeutung sämtlicher Datenbankfragen für das Produktdatenmanagement wird hierdurch eindrucksvoll unterstrichen.

## 7 Zusammenfassung

Datenbanksysteme arbeiten nach dem Prinzip der *Integration*:

- Applikationen arbeiten über einem integrierten Datenmodell.
- Änderungen sind für alle integrierten Systeme sofort sichtbar.
- Datenbankzugriffe erfolgen über normierte Anfragesprachen.

Der *konzeptuelle Entwurf einer Datenbank* erfolgt in drei Schritten. Zunächst findet eine *Informationsanalyse* des Ausschnittes der realen Welt statt, der für die Modellierung relevant ist. Daraus lässt sich ein semantisches Modell in einer Datenmodellierungssprache erarbeiten (*Informationsmodellierung*). Es handelt sich dabei vorwiegend um grafische Notationen. Das Entity Relationship Model (ERM) bildet Objekte der realen Welt als Entities ab. Beziehungen zwischen Entities werden in Form von Relationships modelliert. Am Schluss erfolgt die *Definition eines Datenbankschemas*, das letztlich auf ein implementierbares, operationales Modell abgebildet wird. Eine Möglichkeit bietet das weit verbreitete Relationenmodell.

Eine *Abfragesprache* dient dem Schreiben und Lesen von Information in Datenbanken. Die genormte relationale Abfragesprache SQL (Structured Query Language) hat wesentlichen Anteil am grossen Erfolg des Relationenmodells.

Eine *Transaktion* ist eine Folge von Datenbankoperationen, die eine Datenbank von einem konsistenten Zustand in einen nächsten konsistenten Zustand überführt. Dabei spielen Mehrbenutzerkontrolle (Concurrency Control) und Datensicherung (Recovery) eine wichtige Rolle. Sie sind für die Implementierung einer Datenbank im Rahmen eines PLM-Systems von grundlegender Bedeutung.

### Verständnisfrage 1

Wodurch unterscheiden sich Kopplung und Integration?

### Verständnisfrage 2

Wofür braucht man Datenbanksysteme bzw. was sind deren Vorteile?

### Verständnisfrage 3

Welche Ebenen unterscheidet man in der Datenbankarchitektur?

### Verständnisfrage 4

Wozu benötigt man ein Datenmodell?

**Verständnisfrage 5**

Welche Aspekte sind auf der konzeptuellen Ebene zu unterscheiden?

**Verständnisfrage 6**

Auf welchem Modell beruhen heutige Datenbanksysteme?

**Verständnisfrage 7**

Welche Integritätsbedingungen werden durch heutige Datenbanksysteme sichergestellt?

**Verständnisfrage 8**

Was kann bei schlechter Datenmodellierung auftreten?

**Verständnisfrage 9**

Welche Modellierungskonzepte besitzen semantische Datenmodelle?

**Verständnisfrage 10**

Weshalb ist im Datenbankentwurfsprozess eine Modellabbildung erforderlich?

**Verständnisfrage 11**

Besteht ein Informationsgefälle bei der Modellabbildung?

**Verständnisfrage 12**

Wer hat das Relationenmodell erfunden?

**Verständnisfrage 13**

Was ist im Relationenmodell das Pendant zu einer Beziehung in einem semantischen Modell und was muss bei der Abbildung beachtet werden?

**Verständnisfrage 14**

Wie wird eine Spezialisierung auf das Relationenmodell abgebildet?

**Verständnisfrage 15**

Welche Grundfunktionen muss eine Datenbankanfragesprache unterstützen? Aus welchen Hauptkomponenten besteht eine Datenbankanfragesprache?

**Verständnisfrage 16**

Was versteht man unter deskriptiv?

**Verständnisfrage 17**

Was ist der Unterschied zwischen Delete und Drop?

**Verständnisfrage 18**

Was versteht man unter embedded SQL?

**Verständnisfrage 19**

Was versteht man unter dynamischem SQL?

**Verständnisfrage 20**

Welche Verknüpfungsoperationen über mehrere Tabellen sind zu unterscheiden?

**Verständnisfrage 21**

Welche Programmiersprache wird für Datenbankzugriffe benötigt?

**Verständnisfrage 22**

Welche Programmiersprache wird für semantische Datenmodelle verwendet?

**Verständnisfrage 23**

Was versteht man unter einem Trigger und welche Triggermöglichkeiten werden unterstützt?

**Verständnisfrage 24**

Welche Triggerung empfiehlt sich für Beziehungen, welche für Spezialisierungen?

**Verständnisfrage 25**

Welche Eigenschaften haben Transaktionen?

**Verständnisfrage 26**

Woraus besteht eine Transaktionsverwaltung?

**Verständnisfrage 27**

Welche typischen Probleme treten im Mehrbenutzerbetrieb auf?

**Verständnisfrage 28**

Woraus besteht eine Transaktion?

**Verständnisfrage 29**

Weshalb werden Schutzmechanismen benötigt?

**Verständnisfrage 30**

Was versteht man unter Datensicherung?

**Verständnisfrage 31**

Wozu benötigt man ein Backup?

**Verständnisfrage 32**

Mit welchen Fehlermöglichkeiten bzw. -klassen muss man bei PLM-Applikationen rechnen?

**Verständnisfrage 33**

In welchem Bereich können Media-Fehler die schlimmsten Folgen ergeben?

**Verständnisfrage 34**

Was versteht man unter Recovery?

**Verständnisfrage 35**

Was versteht man unter dem "WAL"-Prinzip?

**Verständnisfrage 36**

Mit welcher Ausgangslage muss bei einem Systemfehler ausgegangen werden?

**Verständnisfrage 37**

Wo sind UNDO/REDO-Informationen hinterlegt?



**Verständnisfrage 38**

Wodurch unterscheiden sich Crash Recovery und Media Recovery?

**Verständnisfrage 39**

Was ist der entscheidende Unterschied zwischen Volume- und Datenbank-Recovery und wie kann deren Charakteristik beschrieben werden?

**Verständnisfrage 40**

Aus welchen Gründen bestehen Grenzen für die Integration von Applikationen?

**Verständnisfrage 41**

Wodurch unterscheiden sich zentrale und verteilte Applikationen?

**Verständnisfrage 42**

Welche Anforderungen stellen sich an verteilte PLM-Applikationen?

**Verständnisfrage 43**

Können Datenbanksysteme allein die PLM-Verteilungsproblematik lösen?

**Verständnisfrage 44**

Auf welchem Konzept basieren heutige PLM-Verteilungskonzepte?

**Verständnisfrage 45**

Welche systemübergreifende Aktivitäten unterscheidet man?

**Verständnisfrage 46**

Wodurch unterscheiden sich globales und lokales Löschen?

**Verständnisfrage 47**

Was kann bei verteilten Systemen grosse Probleme verursachen?

## Publikationsverzeichnis – Literatur

- [ABD-89] *Atkinson, M.; Bancilhon, F.; DeWitt, D.; Dittrich, K.; Maier, D.; Zdonik, S.*: The Object-Oriented Database System Manifesto. Proc. 1th Int. Conf. on Deductive and Object-Oriented Databases (DOOD'89), Kyoto (Japan) 1989, p. 223-240
- [ANS-75] *ANSI/X3/SPARC*: Interim Report from the Study Group on Database Management Systems. FDT-Bulletin of ACM-SIGMOD, Vol. 7 (1975) No. 2, p. 3-140
- [Cod-70] *Codd, E.F.*: A Relational Model of Data for Large Shared Data Banks. Communications of the ACM, Vol. 13 (1970) No. 6, p. 377-387
- [Che-76] *Chen, P.P.-S.*: The Entity-Relationship Model - Toward a Unified View of Data. ACM Trans. on Database Systems, Vol. 1 (1976) No. 1, p. 9-36
- [Dit-92] *Dittrich, K.*: Objektorientierte Datenbanksysteme: Konzepte, Nutzen, und Positionierung. In: Wirtschaftsinformatik in Forschung und Praxis, S. 177-200, München: Hanser Verlag 1992
- [Dit-95] *Dittrich, K.*: Objektorientierte Datenbanken als Basistechnologie für EDM-Systeme. 4. int. Ploenzke EDM-Kongress, Karlsruhe, 2.-4. Mai 1995
- [EIN-94] *Elmasri, R.; Navathe, S. B.*: Fundamentals of Database Systems. 2nd ed., Redwood City (CA): Benjamin/Cummings 1994
- [LoS-87] *Lockemann, P. C.; Schmidt, J. W.*: Datenbank-Handbuch. Berlin: Springer Verlag 1987
- [Mar-97] *Martin, J.*: Part design comes together on the Net. Mechanical Engineering ( US), June 1997, p. 76-78
- [Nor-93] *Norrie, M. C.*: Objectoriented Databases. Scriptum zur Vorlesung 37-375 an der ETH Zürich, WS 93/94, IS-DBS/ETH Zürich 1993
- [Sche-90a] *Schek, H.-J.*: Informationssysteme. Scriptum zur Vorlesung 37-301 an der ETH Zürich, WS 90/91, IS-DBS/ETH Zürich 1990
- [Vos-95] *Vossen, G.*: Datenmodelle, Datenbanksprachen und Datenbank-Management-Systeme. 2. Aufl., Bonn: Addison-Wesley 1994
- [Wei-88] *Weikum, G.*: Transaktionen in Datenbanksystemen. Bonn: Addison-Wesley 1988
- [Zeh-89] *Zehnder, C. A.*: Informationssysteme und Datenbanken. 5. Aufl., Stuttgart: Teubner 1989



# Objektmanagement

Autor: Dr. R. Montau



# 1 Überblick

Aufgrund ihrer zentralen Stellung im betrieblichen Informationsmanagement (vgl. Abschnitt 1.4, Bild B012obmZ) und ihrer Rolle als Integrationsdrehscheibe für sämtliche technischen CAx-Applikationen müssen PLM-Systeme eine ganze Reihe verschiedenartiger Anwendungsobjekte berücksichtigen, wie z. B.:

- Bauteilstammdaten,
- Produktstrukturinformationen (Stücklisten),
- CAD-Modelle,
- Technische Zeichnungen,
- Anforderungslisten,
- NC-Programme,
- FEM-Analysen,
- Visualisierungsdaten (Virtual Reality, Animationen etc.),
- Technische Berichte,
- Klassifikationen,
- Projektdokumentationen,
- Marketing Dokumente etc..

Ein jedes dieser Anwendungsobjekte hat eigene, mehr oder weniger spezifische Anforderungen an ein zweckmässiges Objektmanagement, das nicht nur bezüglich dieser Objektvielfalt Leistungsfähigkeit und Flexibilität benötigt, sondern darüber hinaus auch generisch an unternehmensspezifische Gegebenheiten anpassungsfähig sein muss.

In diesem Zusammenhang sind im vorliegenden Kapitel folgende zentrale Aspekte zu diskutieren:

- *PLM-Objektmodell:*  
Zunächst werden eine Reihe allgemeiner Basisobjektclassen benötigt, anhand derer die obengenannten realen Objekte auf Systeminstanzen abgebildet werden können, die möglichst ähnliche Eigenschaften und auch Verhalten wie die realen Objekte besitzen. Wichtig für den Anwender ist deren detaillierte Kenntnis, um je nach Aufgabenstellung die bestgeeignete Objektclassen auswählen zu können.
- *Beziehungen zwischen Objekten:*  
Objekte sind zwar eigenständig, gehen jedoch oftmals eine Reihe von Beziehungen mit anderen Objekten ein und interagieren miteinander hierüber. Aufbauend auf einer eingehenden Analyse der möglichen Beziehungsarten, sind die jeweiligen Unterschiede, Einsatzbereiche sowie Vor- und Nachteile zu klären, um anwendungsseitige Abhängigkeiten auf geeignete Objektrelationen oder -strukturen abbilden zu können.

- *Generische Konfigurationsmechanismen:*  
Aufgrund ihrer grossen Bedeutung bestehen für Objektstrukturen eine ganze Reihe generischer Konfigurationsmechanismen, die im Gegensatz zu herkömmlichen Stücklisten erheblich flexiblere Abfragen und eine bedeutend grössere Funktionalität ermöglichen.
- *Variantenbildung:*  
Mittels Variantenbildung können insbesondere im PLM und CAx-Bereich erhebliche Rationalisierungserfolge erzielt werden. Um einen effizienten Einsatz der Variantenbildung in Produktstrukturen zu ermöglichen, ist jedoch ein detailliertes Verständnis der Grundkonzepte sowie deren zweckmässige Abbildung auf Systemkonstrukte erforderlich.
- *Autorisierung:*  
Um leistungsfähige und zugleich flexible Methoden für den Zugriffsschutz im Objektmanagement bereitstellen zu können, benötigen PLM-Systeme ein geeignetes Autorisierungsmechanismus.

## 2 PLM-Objektmodell

Von seiten eines PLM-Systems werden verschiedene Objektklassen zur Verfügung gestellt, die jeweils zweckmässige Mechanismen für ein oder mehrere der obengenannten Anwendungsprojekte bereitstellen. Zur Veranschaulichung sind in Bild B001obmZ die in der Standardkonfiguration eines modernen objektorientierten PLM-Systems angebotenen Objektklassen dargestellt. Diese Objektklassen sind vergleichbar mit einem Werkzeugkasten, woraus der Benutzer für ein jedes *Anwendungsobjekt* (z. B. Bauteilstammdaten) die bestgeeignete *Objektklasse* (z. B. Item) auszuwählen hat.

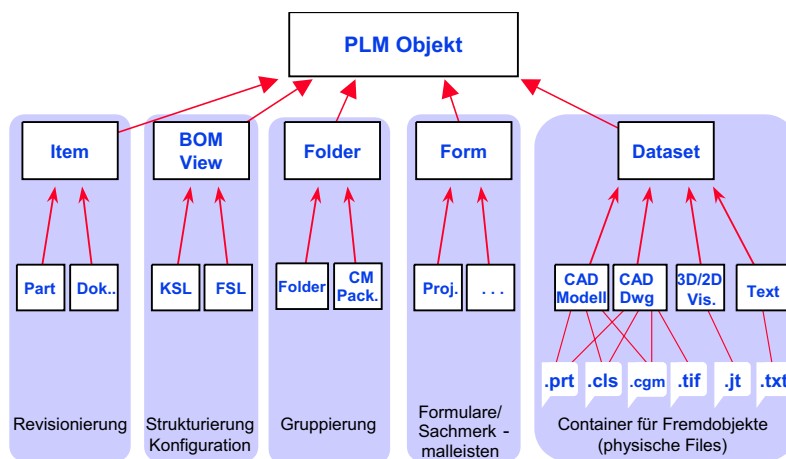


Bild (B001obmZ) Objektklassen von PLM-Systemen (beispielhaft und stark vereinfacht)

Das Objektmodell ist von systemspezifischer Natur und kann bei einer Betrachtung verschiedener PLM-Systeme mehr oder weniger variieren. Darüber hinaus muss jedes PLM-System auf unternehmensspezifische Anforderungen angepasst werden, wofür nötigenfalls auch eine Erweiterung des Standardobjektmodells erforderlich sein kann. Aus diesem Grunde soll hier nur auf die allgemeingültigen Konzepte eingegangen werden, damit die nachfolgenden Anwendungsfunktionen verständlich diskutiert werden können. Grundsätzlich erwähnt sei an dieser Stelle, dass hier von einer objektorientierten PLM-Systemarchitektur ausgegangen wird, die dem Anwender durch den Objekt Layer (vgl. Bild B011obmZ) ein Objektmodell ähnlich wie in Bild B001okI zur Verfügung stellt und er dadurch die Vorteile der Objektorientierung nutzen kann (Kapselung von Struktur und Verhalten, Vererbungsmechanismen etc.).



Das wichtigste Anwendungsobjekt für die PLM-Technologie überhaupt stellen die Bauteilstammdaten dar. Diese werden im Zuge einer Produktentwicklung oftmals mehrfach geändert und müssen daher revisionierbar sein, um verschiedene Änderungs- bzw. Revisionszustände auf das System abbilden zu können. Wie in Abschnitt 1.2 bezüglich der Produkthaftung bereits erläutert, müssen frühere Revisionszustände jederzeit rückverfolgbar sein, um beispielsweise den Datenbankzustand eines früheren Datums (z. B. zum 1. Okt. 2001) wieder nachbilden zu können.

Hierfür wird im Objektmodell in Bild B001obmZ das Konzept des *Items* angeboten. Da obige Anforderungen aber nicht nur für Bauteilstammdaten, sondern auch für Dokumente, wie z.B. Anforderungslisten, gelten können, werden Items weiter unterschieden in Bauteile (Parts), Dokumente usw. Bei dieser Untergruppierung innerhalb einer Klasse (häufig Typenbildung genannt) wird jeweils der meist benötigte Typ als sogenannter *primary type* definiert (hier: Part), wodurch dieser als Default-Einstellung verwendet wird, falls keine Typ-Angabe erfolgt.

Bauteile sind entweder Baugruppen oder Einzelteile und erstere bestehen immer aus mindestens zwei Unterteilen bzw. Komponenten, die wiederum Unterstrukturen aufweisen können. Diese Produktstrukturinformationen, die im herkömmlichen Sprachgebrauch als Stücklisten bezeichnet werden, sind eng verknüpft mit den oben erläuterten Bauteilstammdaten.

Im Objektmodell von Bild B001obmZ steht hierfür das Konzept der *BOM View* (Bill Of Material = BOM) zur Verfügung, das nur zusammen mit einem Ober-Bauteil einen Sinn ergibt. Produktstrukturen (BOM Views) erfordern neben der Rückverfolgbarkeit zudem generische Konfigurationsmechanismen, worauf nachfolgend detaillierter eingegangen wird.

Bei der konventionellen Projektbearbeitung wird häufig eine Art Projektmappe eingesetzt, womit temporär oder auch dauerhaft verschiedene zusammengehörige Dokumente gruppiert werden. Seit Einführung der Apple/Macintosh-Betriebssystemoberfläche wird zunehmend versucht, den Bildschirmarbeitsplatz (Desktop) mehr und mehr an einen realen Schreibtisch anzunähern (Schubladen, Hängemappen, Papierkorb), um so für den Benutzer die Funktionalität moderner Applikationen mit gewohnten Arbeitsweisen zu verbinden.

Zur Gruppierung von Objekten wird im Objektmodell von Bild B001obmZ das *Folder*-Konzept angeboten, welches sowohl nur temporär als auch zur langfristigen Projektdokumentation und somit -organisation genutzt werden kann. Da Objektgruppierungen insbesondere im Änderungswesen benötigt werden, bauen Systemmodule,

wie z.B. das Change Management (CM), auf dem Folderkonzept auf und verwenden sogenannte CM Packages, um die zu einer Änderung zugehörigen Informationsobjekte zu gruppieren.

In der Regel sind für betriebliche Objekte verschiedene typspezifische Attributwerte von Interesse. Dies gilt ebenso für betriebliche Vorgänge, die häufig spezielle Formulare erfordern (z. B. Antragsformular für Änderungen).

Hierfür wird im Objektmodell das Konzept der *Form* angeboten, welches in vielfältiger Weise genutzt werden kann und meist bereits in der Standardkonfiguration eine Vielzahl an speziellen Formtypen besitzt. Zur Dokumentation von Projektinformationen kann beispielsweise eine Projekt-Form zur Verfügung gestellt werden.

Eine darüber hinausgehende Funktionalität bieten *Sachmerkmalleisten*, die weniger für eigenständige Formulare eingesetzt werden, sondern vielmehr für weitere Attribute von z. B. speziellen Items (z. B. Durchmesser und Länge einer Schraube). Wesentliches Merkmal von Sachmerkmalleisten gegenüber Forms ist deren Fähigkeit zur dynamischen Erzeugung während der Laufzeit und die enge Zuordnung zu bzw. Abhängigkeit von einem Stammobjekt.

Sämtliche bisherigen Konzepte beziehen sich auf Informationen, die in der Regel direkt im PLM-System entstehen und nicht zwingend irgendeiner Fremdapplikation bedürfen. Demgegenüber sind eine Vielzahl von Anwendungsobjekten zu unterscheiden, die in einer externen Applikation (CAx, DTP/Office-Pakete) erstellt und lediglich als Fremdobjekte in der PLM-Datenbank verwaltet werden (z. B. CAD-Modelle, Techn. Zeichnungen, techn. Berichte etc.).

Diese Container-Funktionalität für Fremdobjekte übernimmt das *Dataset*-Konzept im Objektmodell. Aufgrund der Verschiedenheit möglicher Fremdapplikationen ist die Unterscheidung mehrerer Dataset-Typen von wesentlicher Bedeutung. Im einfachen Fall einer Text-Applikation muss lediglich eine physische Datei berücksichtigt werden, auf die über eine Referenz (häufig "Named Reference" in Abhängigkeit der Dateiendung, z. B. Text für .doc) zugegriffen wird. Im Falle von CAx-Applikationen (z. B. Unigraphics) kann ausgehend von einem Dataset-Objekt jedoch eine ganze Reihe untereinander verketteter Dateien vorliegen, die das Produktmodell der technischen CAx-Prozesskette repräsentieren (vgl. Bild B006obmZ). Da CAD-seitig ebenfalls gewisse Informationen bezüglich Bauteilstammdaten und Produktstrukturen entstehen bzw. benötigt werden (vgl. Zeichnungskopf, CAD-Assembly-Struktur), die insbesondere in Item- und BOM View-Objekten gespeichert sind, ist für die Einbindung solcher komplexer Dateistrukturen eine "tiefe Integration" erforderlich.

Die vorstehend erläuterten Objektklassen des Objektmodells in Bild B001obmZ sind mit mehr oder weniger stark abweichender Bezeichnung, Funktionalität und Implementierung in den meisten der kommerziell verfügbaren PLM-Systeme vorhanden. Ohne äquivalente Konzepte für Stammdaten (hier: Items), Produktstrukturen (hier: BOM Views) und Container-Objekte zur Einbindung externer Applikationen (hier: Dataset) kann eine PLM-Lösung ihre Aufgaben nicht erfüllen.

Hiermit kann die bislang als simpler Datenkübel symbolisierte PLM-Datenbank detaillierter dargestellt werden (vgl. Bild B002obmZ). Zu unterscheiden sind einerseits

- die im relationalen Datenbanksystem (z.B. Oracle) in Tabellenform hinterlegten PLM-Objekte sowie andererseits
- die in einem geschützten Speicherbereich (Electronic Vault) des Filesystems auf Betriebssystemebene verwalteten Dateien der eingebundenen Fremd-Applikationen.

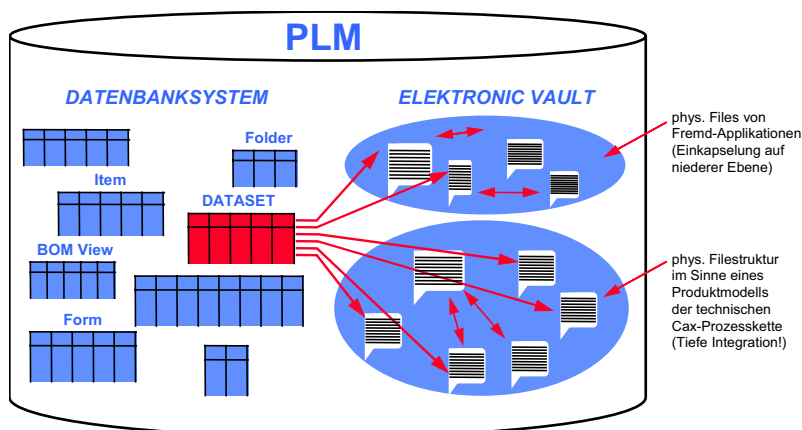


Bild (B002obmZ) Inhalt der PLM-Datenbank

Die Unterteilung in relational verwaltete PLM-Objekte und lediglich auf Betriebssystemebene zugeordnete physische Dateien ist ein charakteristisches Merkmal des PLM-Ansatzes und zugleich von entscheidender Bedeutung für die Interoperabilität mit dem Erzeugersystem.

## 2.1 Abstraktionsebenen der PLM-Informationsmodellierung

PLM-Systeme zielen auf eine integrative Unterstützung der technischen CAX-Prozesskette ab, wobei deren Datenbestände als Fremdoobjekte „mitverwaltet“ werden, ohne dass ein integriertes Datenmodell existiert. Bild B003obmZ verdeutlicht das Prinzip der Einkapselung beliebiger Fremdapplikationen, welches sich durch das Ablegen einer Datei in einem Container oder geschlossenen Sack veranschaulichen lässt, dem weitere PLM-relevante Informationen (z. B. Zeichnungsnummer) als Etiketten hinzugefügt werden können. Diese in der Regel relational verwalteten Informationen werden häufig auch als „Meta“-Daten bezeichnet, da sie gegenüber den spezifischen, im Sack verborgenen Datenstrukturen eine höhere Abstraktionsebene darstellen.

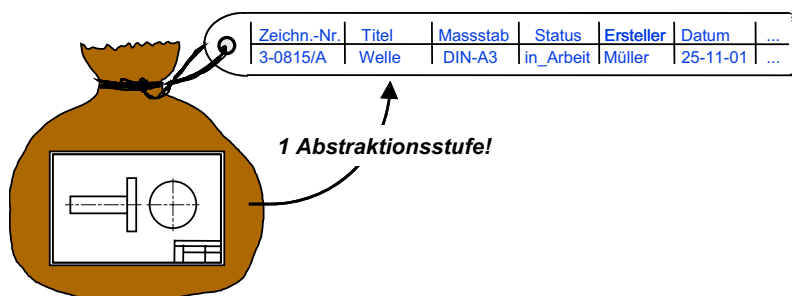


Bild (B003obmZ) Prinzip der Einkapselung von Fremdobjekten und deren Verwaltung anhand von Meta-Daten

Die Interoperabilität insbesondere mit CAX-Applikationen ist jedoch von ganz entscheidender Bedeutung für den effizienten Einsatz der PLM-Technologie, da sich CAX-Systeme zunehmend in Richtung integrierte Produktmodellierer entwickeln und Prozessketten immer durchgängiger abbilden können. Da beispielsweise Produktstrukturinformationen sowohl CAX- als auch PLM-seitig geführt werden, ergeben sich zwei wesentliche Anforderungen an die Interoperabilität:

- Assoziativität  
Informationsbeziehungen auf CAX-Datenmodell-ebene (z. B. vom CAD-Geometriemodell zum NC-Programm) müssen auf abstrahierter PLM-Datenmodellebene abbildbar sein.  
Änderungen globaler Objekte (z. B. CAD-Assembly Modell, PLM-Produktstruktur) müssen jeweils in sämtlichen Systemen automatisch und konsistent nachgeführt werden.

- Funktionalität  
Die Unterordnung der CAx-Erzeugersysteme unter die Kontrolle eines PLM-Systems darf nicht zum Verlust vorhandener CAx-Funktionalität führen.

Bild B004obmZ verdeutlicht die unterschiedlichen Abstraktionsebenen des Produktmodells auf CAx-Datenmodellebene (welches sich aus verschiedenen Partialmodellen zusammensetzt) gegenüber der PLM-Modellebene. Sämtliche Informationsbezüge zwischen den verschiedenen CAx-Applikationsobjekten müssen im diesbezüglich abstrahierten PLM-Modell abgebildet werden. Produktdatenänderungen auf einer der beiden Ebenen müssen auf der jeweils anderen konsistent nachgeführt werden. Dies gilt auch für die Änderung von Beziehungen zwischen verschiedenen CAx-Partialmodellen.

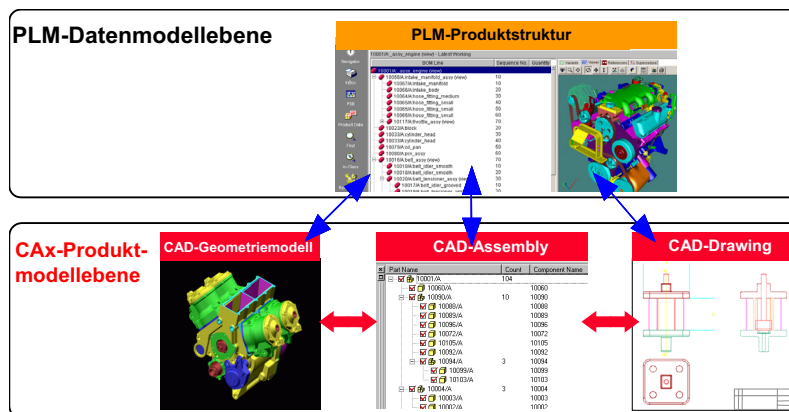


Bild (B004obmZ) Abstraktionsebenen von CAx und PLM

### 3 Beziehungen zwischen Objekten

Charakteristisch für die Objekte einer Produktentwicklung sind die vielfältigen Beziehungen zwischen den Instanzen der einzelnen Objektklassen, worunter anwendungsseitig z. B. Bauteilstammdaten, CAD-Modelle, Technische Zeichnungen etc. zu verstehen sind. Grundsätzlich sind hierbei zwei Arten von Beziehungen zu unterscheiden:

- *Zuordnung* mehrheitlich verschiedenartiger Objekte im Sinne von Objektverknüpfungen bzw. Objektrelationen,
- *Strukturierung* von ausschliesslich gleichartigen Objekten im Sinne von Objekthierarchien.

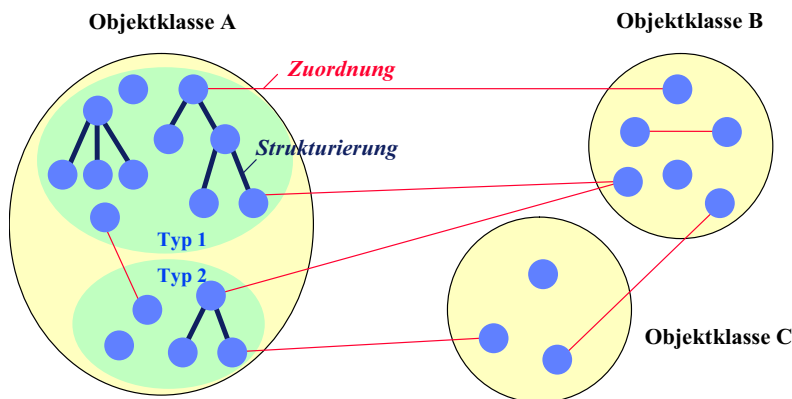


Bild (B005obmZ) Unterscheidung von Objektbeziehungen

Objektrelationen stellen somit eine „flache“ Zuordnung von Objekten dar, um einen Bezug zwischen zusammengehörenden Objekten auszudrücken (z. B. CAD-Modell beschreibt Bauteil). Objektstrukturen werden demgegenüber auf artverwandte Objekte angewendet, um ein hierarchisches Ordnungsverhältnis aufzubauen (z. B. Baugruppe 1 besteht aus Einzelteilen und Unterbaugruppen, Einzelteil wird verwendet in Baugruppe 1 und Baugruppe 2).

Nachfolgend sollen die charakteristischen Merkmale beider Beziehungsarten vorgestellt werden.

### 3.1 Objektrelationen (Zuordnungen)

In der betrieblichen Praxis finden sich Objektrelationen in kaum überschaubarer Vielzahl. Üblich ist die Zusammenfassung interessierender Informationen einer Produktentwicklung in einer Projektmappe, die gegebenenfalls in Form mehrerer Kopien zwischen den involvierten Stellen hin- und hergereicht wird. Beispiele solch interessierender Informationen sind Bauteilstammdaten, Produktstrukturen, CAD-Modelle usw., zwischen welchen jeweils unterschiedliche Beziehungen existieren (vgl. Bild B006obmZ).

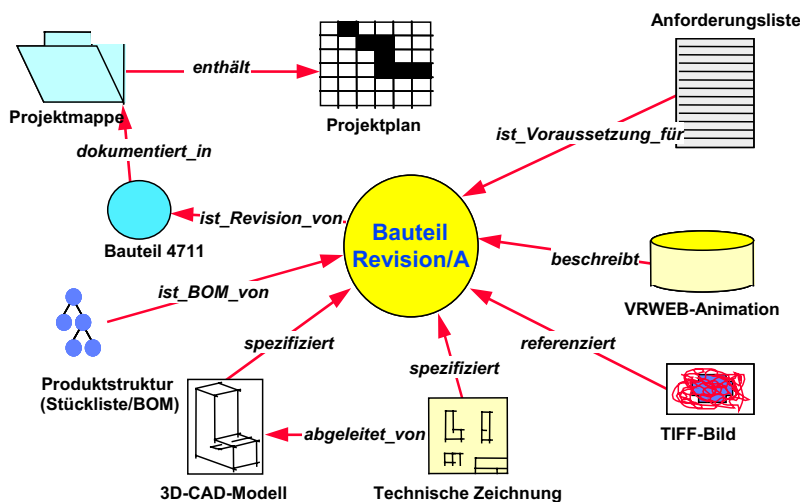


Bild (B006obmZ) Beziehungen zwischen Anwendungsobjekten

Aufgrund der Verschiedenheit der Beziehungen zwischen Anwendungsobjekten (bzw. deren Typs), werden in einer Produktentwicklung unterschiedliche Zugriffsmechanismen und Prozessabläufe für die einzelnen in Relation stehenden Objekte benötigt. Hierbei kann es sinnvoll sein, ein Abhängigkeitsverhältnis auszudrücken, was vorwiegend für Objektrelationen bezüglich Bauteilstammdaten (Items) zweckmässig ist, wie z. B.:

- Bevor ein Bauteil freigegeben werden kann, müssen vorgängig die Anforderungsliste oder die untergeordneten Baugruppen freigegeben sein (Relationstyp: Requirement).
- Falls eine Bauteilrevision freigegeben wird, sollen auch deren Produktstruktur, das zugehörige CAD-Modell und die hiervon abgeleitete Technische Zeichnung freigegeben werden (Relationstyp: Specification).

- Manche Objekte sind zwar eigenständig und sollen separat zu einem späteren Zeitpunkt freigegeben werden, sind aber von anderen Objekten abgeleitet worden (z. B. NC-Werkzeugverfahrwege vom CAD-Modell; Relationstyp: Manifestation).
- Oftmals besteht irgendein Zusammenhang zwischen Objekten, welcher allgemeiner Art ist, aber keinerlei Veränderungen hervorruft und daher beliebig auch für bereits freigegebene Objekte anwendbar sein soll (z. B. Produktentwicklung wird begründet in Marktanalyse; Relationstyp: Reference).
- Darüber hinaus sind gewisse Objekte (z.B. Revisionen, Produktstrukturen, spezifische Attributwerte nur im Zusammenhang mit anderen Objekten (Bauteilen) sinnvoll (weak objects, vgl. Bild B001obmZ) und sollen daher beim Löschen der Letzteren mitgelöscht werden.

Für Zwangsbedingungen dieser Art müssen PLM-Systeme äquivalente Mechanismen anbieten, womit die gewünschten Effekte erzielt werden können. Zur beispielhaften Verdeutlichung ist in Bild B007obmZ eine exemplarische Anwendung dieser Relationstypen dargestellt. Die selektierte Bauteilrevision (Item 4711/A) ist bereits freigegeben (Release Status: Released), womit automatisch sämtliche Objekte mit freigegeben wurden, welche diesem mit einer Bom View- oder Specification-Relation zugeordnet sind.

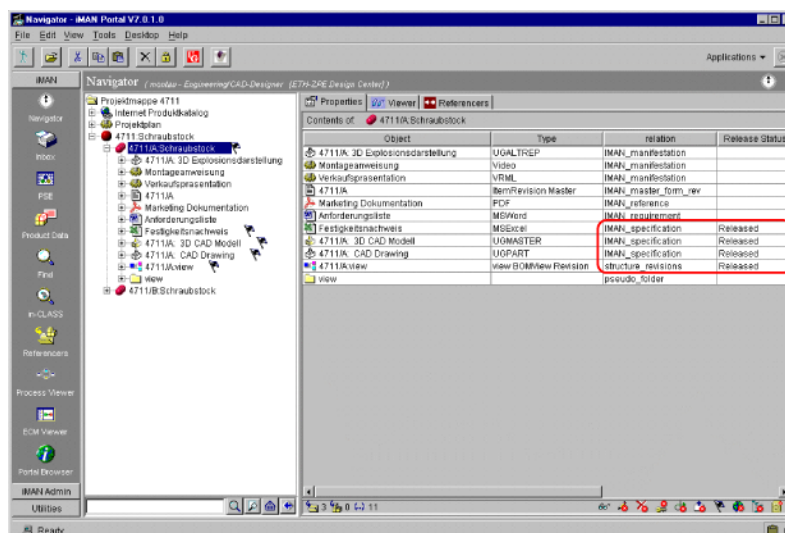


Bild (B007obmZ) Beispielhafte Anwendung von Relationstypen (Systembeispiel Team-center)



Für den Anwender ist hierbei wichtig, dass sämtliche Zwangsbedingungen, die anwendungsseitig benötigt werden, systemseitig auf entsprechende Relationstypen abgebildet werden können. Zudem müssen für die zweckmässige Auswahl des jeweils bestgeeigneten Relationstyps einige wesentliche Unterscheidungsmerkmale bekannt sein, wie deren beispielhafte Gegenüberstellung in Tabelle T001obmZ verdeutlicht.

Relationstyp	Anwendung	Schreibrecht	Mit-Freigabe	Mit-Löschung	Bemerkungen
BOMView BOMViewRev.	Produktstrukturierung (Stückliste/ Bill of Mat.)	X	X	X	immer nur bzgl. 1 Item/ItemRev.
Master_Form	Zuordnung spezieller Attribute	X	X	X	Mit-Freigabe durch Rule Handler erzwingbar
Revision	Revisionierung	X	/	X	systemkontrolliert
Specification	Zuordnung spezifizierender Daten	X	X	-	Mit-Freigabe durch Rule Handler erzwingbar
Requirement	Zuordnung von Anforderungen, Normen	X	-	-	vorgängige Freigabe durch Rule Handler erzwingbar
Manifestation	Zuordnung abgeleiteter Objekte	-	-	-	beliebig anwendbar auch nach Freigabe
Reference	Zuordnung beliebiger Objekte	-	-	-	beliebig anwendbar auch nach Freigabe

Tabelle (T001obmZ) Charakterisierung von Relationstypen

Der Aufbau solcher Relationen darf natürlich in keinster Weise zu Datenredundanzen führen, obwohl ein Objekt selbstverständlich mit beliebig vielen anderen in Relation stehen kann und dadurch an mehreren Stellen in einem Arbeitsfenster aufgeführt sein könnte. Hierfür kommen sogenannte Objektreferenzen zum Einsatz, die eine Voraussetzung für die Anwendung des Folder-Konzeptes darstellen.

### 3.1.1 Objektreferenzen und Folder-Konzept

Objektreferenzen sind eine Art Pointer (Zeiger) auf ein Objekt, wobei zwei Ebenen zu unterscheiden sind (vgl. Bild B008obmZ):

- *Referenzebene* mit mehreren gegebenenfalls strukturierten Objektreferenzen,
- *Objektebene* als physikalische Repräsentation auf einem Speichermedium.

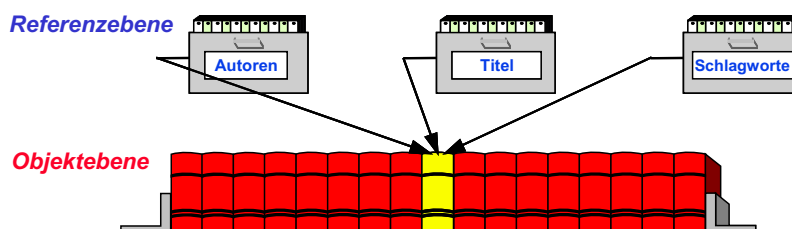


Bild (B008obmZ) Beispielhafte Verdeutlichung von Objektreferenzen

Veranschaulichen lässt sich dies anhand dem Beispiel einer Bibliothek, die mehrere Karteien, sortiert nach Autoren, Titeln oder Schlagworten unterhält. Über Einträge in verschiedenen Karteien kann effizient auf ein gesuchtes Buch zugegriffen werden, ohne den gesamten Buchbestand durchsuchen zu müssen. Äquivalent hierzu ist in der Datenbanktechnologie das Anlegen eines Indexes für ein häufig benötigtes Attribut.

Obwohl dieser Vergleich noch so einleuchtend sein mag, bereitet der Umgang mit Objektreferenzen in einem PLM-System immer wieder Schwierigkeiten. Es sei daher ausdrücklich darauf hingewiesen, dass Operationen auf der Referenzebene (häufig mittels Cut&Paste realisiert) völlig unabhängig sind von solchen auf der Objektebene (Create bzw. Delete). Das Entfernen einer Objektreferenz (mittels Cut) löscht daher nur die Relation zwischen Objekten, aber nicht die Objekte selbst. Ebenso führt das Kopieren einer Objektreferenz (mittels Copy&Paste) lediglich zur Instanziierung einer Relation und eben nicht (!) zu einer physikalischen Kopie des Objektes selbst.

In Anlehnung an die externe Ebene des Datenbankarchitekturmodells ermöglicht die Referenzebene somit den Aufbau unterschiedlicher Benutzersichten auf dieselben Objektdaten.

Hierdurch können problemlos Mehrfachzuordnungen instanziiert werden, die beispielsweise für die Zuordnung von Objekten zu Földern zweckmässig sein kann. In den gegebenenfalls strukturierten Fol-

dern, die z. B. eine Projektmappe repräsentieren, werden lediglich Objektreferenzen geführt. Ein 3D CAD-Modell, welches z. B. in zwei verschiedenen Foldern enthalten ist, bleibt in der Datenbank dennoch nur einmal gespeichert (Bild B009obmZ).

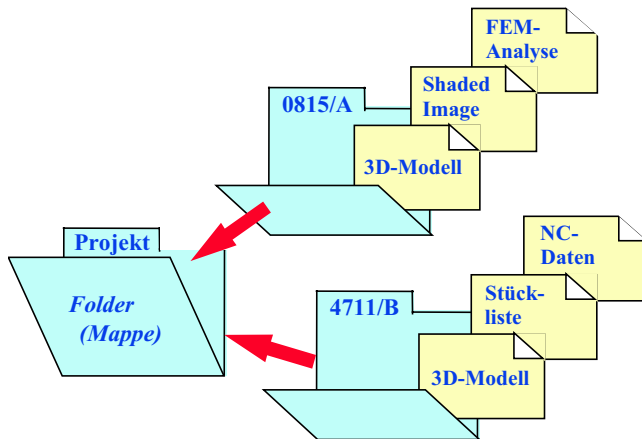


Bild (B009obmZ) Objektgruppierung durch das Folder-Konzept

Erwähnt sei an dieser Stelle, dass die Anwendung von Foldern keine Notwendigkeit ist, sondern nur eine Arbeitserleichterung darstellt, um ähnlich zur gewohnten konventionellen Arbeitsweise mit Mappen in Papierform effizient arbeiten zu können. Objekte, die in keinerlei Folderstruktur geführt werden oder Objekte, die mit ihren Foldern aus dem Arbeitsbereich eines Benutzers mittels Cut entfernt wurden, können selbstverständlich jederzeit anhand von Datenbankabfragen wiedergefunden werden. Das Ergebnis einer Datenbanksuche anhand gewünschter Selektionskriterien kann ausserdem zum systemgenerierten Aufbau von Ordnungsstrukturen genutzt werden, die dann in Foldern beliebig weiterverwendet werden können.

### 3.1.2 Referenz- versus Objekt-basierende Systemphilosophie

Von den heute kommerziell verfügbaren PLM-Systemen unterstützt die überwiegende Mehrzahl das Folder-Konzept auf der Basis von Objektreferenzen. Hinsichtlich deren Implementierung und Anwendungsweise durch den Benutzer bestehen jedoch grosse Unterschiede, wobei insbesondere zwei grundsätzliche Systemphilosophien zu unterscheiden sind (Bild B010obmZ):

- Referenz-basierend,
- Objekt-basierend.

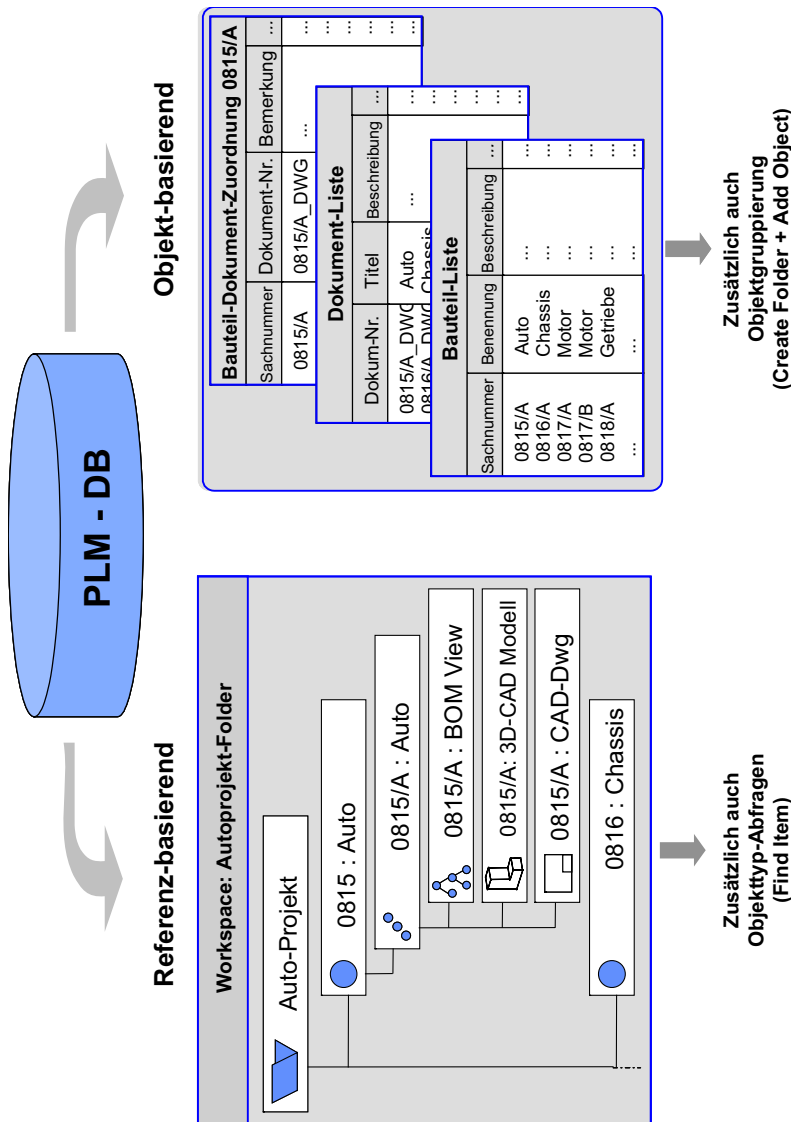


Bild (B010obmZ) Gegenüberstellung von Referenz- und Objekt-basierender Arbeitsweise

Bei der *Referenz-basierenden Systemphilosophie* arbeitet der Benutzer in erster Linie mit einer Darstellung, welche Objekte unterschiedlichen Typs in einer gegenseitigen Zuordnung präsentiert, entsprechend den in der Datenbank hinterlegten Objektbeziehungen. Zuordnungen können hier oftmals durch simples Doppelklicken abgefragt und im selben Window dargestellt werden. Das Erzeugen und Löschen von Zuordnungen erfolgt in der Regel mit Cut, Copy und Paste, was insbesondere PC-Benutzern (Macintosh, Windows, NT etc.) entgegenkommt, die hauptsächlich in Office-Applikationen arbeiten und nur

gelegentlich PLM-Funktionalität benötigen. Jeder Benutzer kann hierbei unterschiedliche und mehr oder weniger viele Objekte (d.h. Objektpreferenzen (!), z. B. auf Projekt-Folder) in seinen persönlichen Arbeitsbereich (häufig Workspace genannt) holen und wird dadurch vorwiegend nur mit der für ihn relevanten Objektmenge konfrontiert. Abfragen über den gesamten Datenbestand (z. B. Find-Query für alle Bauteile) sind jedoch ebenfalls möglich und des öfteren erforderlich.

Bei der *Objekt-basierenden Systemphilosophie* arbeiten sämtliche Benutzer in erster Linie direkt mit den Objekten eines bestimmten Typs, z. B. mit einer Bauteil-Auflistung in einem Window. Solange keine Einschränkung für die interessierenden Bauteile erfolgt (z. B. nur Sachnummern mit "470-\*"), arbeiten verschiedene Benutzer mit ein und derselben Objektmenge in einer Gesamtsicht auf den Datenbestand (z. B. alle Bauteile). Ausgehend von einer spezifischen Bauteilinstanz (z. B. Auto 0815/A) können auch Zuordnungen zu Objekten eines anderen Typs abgefragt werden, die dann häufig in separaten Windows dargestellt werden. Einträge in diesen Zuordnungs-Windows entsprechen den Verknüpfungslinien bei Referenz-basierenden Systemen und sind ebenso als Referenzen auf z. B. Bauteil- oder Dokument-Objekte aufzufassen. In einer Objekttypauflistung befindet man sich dagegen ausschliesslich auf der Objektebene und wird daher mit der Unterscheidung von Referenz- und Objektebene (vgl. Bild B009obmZ) nicht konfrontiert. Aufgrund ihres einfachen Aufbaus werden PLM-Systeme mit objekt-basierender Philosophie daher insbesondere von ungeübten Benutzern als leichter verständlich empfunden. Um jedoch Objekte z. B. im Rahmen eines Entwicklungsprojektes gruppieren zu können, ist jeweils ein weiterer Schritt erforderlich, indem zunächst ein Folder erzeugt und dann die einzelnen Objekte hinzugefügt werden müssen. Für die produktive Projektarbeit in einem Entwicklungsteam wird dies häufig bereits als zu umständlich empfunden, weshalb bei dieser Systemphilosophie vorwiegend alle Benutzer in derselben Gesamtsicht auf den Datenbestand und damit näher an der eigentlichen Speicherungsstruktur arbeiten.

Bei Objektstrukturen als Vertreter der zweiten Art von Beziehungen zwischen Objekten stehen demgegenüber andere Aspekte im Vordergrund.

## 3.2 Objektstrukturen

Objektstrukturen dienen der hierarchischen Ordnung gleichartiger Anwendungsobjekte im Sinne einer Über- bzw. Unterordnung und werden herkömmlich daher als Ordnungssysteme bezeichnet. Demzufolge handelt es sich hierbei in Abhängigkeit der betrachteten Anwendungsobjekte entweder um:

- Bauteil-,
- Dokument- oder auch um
- Projekt-Strukturen.

Von besonderer Bedeutung ist hierbei die Strukturierung von Bauteilen (Artikeln), auf die im folgenden beispielhaft näher eingegangen wird. Die prinzipiellen Überlegungen gelten jedoch ebenso für die beiden anderen genannten Objektarten.

Nach dem begrifflichen Ordnungsgrad lassen sich hierbei drei grundsätzliche Aufbauprinzipien unterscheiden [[DIN 2331]]:

- hierarchisch,
- ahierarchisch und
- teilhierarchisch.

Für die folgenden Betrachtung sollen nur hierarchische Ordnungssysteme bzw. Strukturen interessieren.

Anhand der Art der vorliegenden Beziehungen unterscheidet man bei hierarchischem Aufbauprinzip die folgenden beiden wesentlichen Ordnungssysteme bzw. Objektstrukturen (vgl. Bild B011obmZ):

- Bestandsysteme mit partitiven Strukturbeziehungen  
Hierbei wird die Zusammensetzung eines Objektes aus untergeordneten Bestandteilen (Sub-Objekten) beschrieben. Für Bauteile ergibt sich hieraus die sogenannte Produktstruktur, deren partitive Auswertung von oben nach unten (besteht aus) herkömmlich als Stückliste und umgekehrt (wird verwendet in) als Verwendungsnachweis bezeichnet wird.
- Abstraktionssysteme mit generischen Strukturbeziehungen  
Hierbei stellt ein untergeordnetes Strukturobjekt (Einzelobjekt oder Objektgruppe) eine Spezialisierung bzw. Konkretisierung des übergeordneten Strukturobjekts (Objektgruppe) dar, da es lediglich bezüglich zusätzlicher Merkmale detaillierter beschrieben ist, wodurch sich ein Klassifikationssystem ergibt.

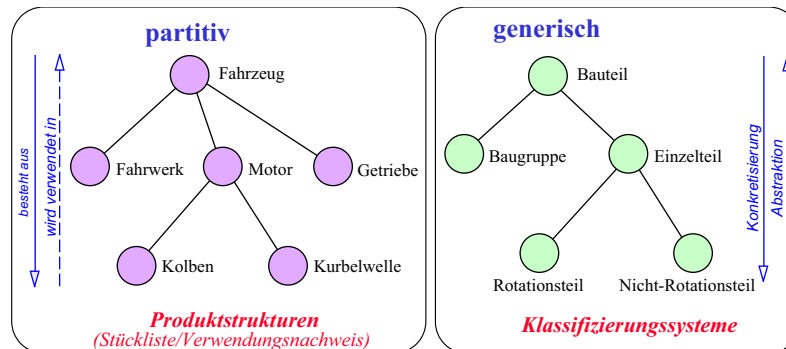


Bild (B011obmZ) Arten von Objektstrukturen; Produktstrukturen: dienen der strukturellen Zusammenfassung von Objekten zu Produkten; Klassifizierungssysteme: dienen der produktübergreifenden Ordnung des gesamten Objektspektrums in Objektgruppen

Klassifizierungssysteme mit generischen Strukturbeziehungen werden für die folgenden Ausführungen nicht benötigt und sollen daher erst in einem späteren Abschnitt (Vorlesungsabschnitt 6) separat behandelt werden.

### 3.3 Produktstrukturen

Die Produktstruktur (häufig auch als Erzeugnisgliederung bezeichnet) ist eine der wichtigsten Informationen über ein Produkt und damit auch Organisationsgrundlage für die Produktentwicklung, -herstellung und zur Steuerung eines Auftragsdurchlaufes. Aufgrund der verschiedenen Anforderungen der einzelnen Unternehmensbereiche ergibt sich hierfür eine komplexe Problemstellung, die in Bild B012obmZ am Beispiel eines Automobils verdeutlicht wird.

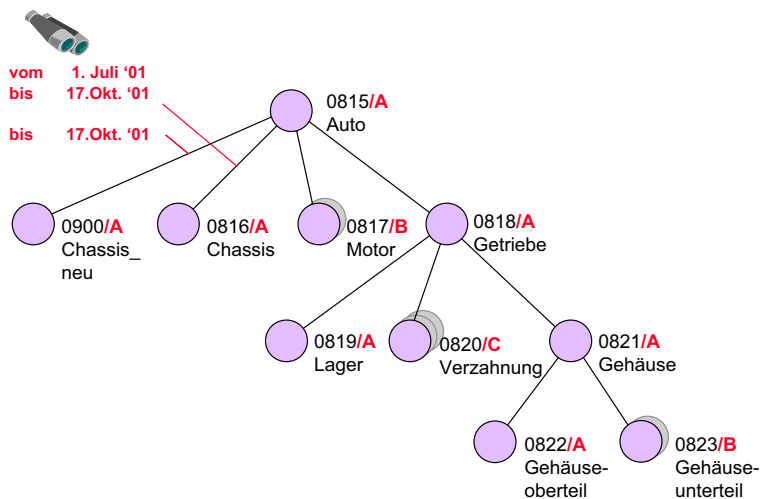


Bild (B012obmZ) Beispiel einer Produktstruktur und deren Konsistenzproblematik

Folgende Konsistenzproblematik kann auftreten:

- mehrere Darstellungsformen
- unterschiedliche Struktursichten
- unterschiedliche Aktualitätszustände
- generische Konfiguration
- Variantenbildung

Hinsichtlich der Konsistenzproblematik sind zunächst *verschiedene Darstellungsformen* zu betrachten, wobei nach [[DIN199/2, VSM-91]] unterschieden wird in:

- Mengen-,
- Struktur- und
- Baukastenstückliste.



MENGEN-STÜCKLISTE 0815/A: Auto				
Pos.-Nr.	Sachnummer	Benennung	Menge	...
10	0816/A	Chassis	1	...
20	0817/B	Motor	1	...
30	0819/A	Lager	1	...
40	0820/C	Verzahnung	1	...
50	0822/A	Gehäuseoberteil	1	...
60	0823/B	Gehäuseunterteil	1	...

STRUKTUR-STÜCKLISTE 0815/A: Auto				
Pos.-Nr.	Sachnummer	Benennung	Menge	...
10	0816/A	Chassis	1	...
20	0817/B	Motor	1	...
30	0818/A	Getriebe	1	...
40	0819/A	Lager	1	...
50	0820/C	Verzahnung	1	...
60	0821/A	Gehäuse	1	...
70	0822/A	Gehäuseoberteil	1	...
80	0823/B	Gehäuseunterteil	1	...

BAUKASTEN-STÜCKLISTE 0815/A: Auto				
Pos.-Nr.	Sachnummer	Benennung	Menge	...
10	0816/A	Chassis	1	...
20	0817/B	Motor	1	...
30	0818/A	Getriebe	1	...

BAUKASTEN-STÜCKLISTE 0818/A: Getriebe				
Pos.-Nr.	Sachnummer	Benennung	Menge	...
10	0819/A	Lager	1	...
20	0820/C	Verzahnung	1	...
30	0821/A	Gehäuse	1	...

BAUKASTEN-STÜCKLISTE 0821/A: Gehäuse				
Pos.-Nr.	Sachnummer	Benennung	Menge	...
10	0822/A	Gehäuseoberteil	1	...
20	0823/B	Gehäuseunterteil	1	...

Bild (B013obmZ) Beispielhafte Gegenüberstellung von Stücklisten unterschiedlicher Darstellungsformen

Mengenstücklisten enthalten keine Strukturinformationen sondern bestehen lediglich aus einer Aufzählung der Einzelteile eines Produktes mit akkumulierten Mengenangaben („Blätter“ des Strukturbaumes). Strukturstücklisten enthalten sämtliche Baugruppen und Einzelteile eines Produktes, wobei Wiederholteile und -gruppen mehrfach aufgeführt sind. Baukastenstücklisten beschreiben die Produktstruktur jeweils nur für die nächsttiefere Strukturebene, weshalb für eine mehrstufige Struktur eine Aufteilung in mehrere Baukastenstücklisten erforderlich ist.

Bei Verwendung eines PLM-Systems stellen diese Darstellungsformen aufgrund der zumindest relationalen Datenverwaltung lediglich unterschiedliche Abfragen derselben Produktstruktur dar. Insbesondere bei objektorientierten Systemen, welche die Produktstruktur als ein eigenständiges, in der Regel einstufiges Objekt abbilden (vgl. BOM View in Bild B001obmZ), besteht eine grosse Ähnlichkeit zwischen der Baukastenstückliste und der Speicherungsform der Produktstruktur. Sämtliche weiteren Darstellungsformen können hieraus abgeleitet werden, weshalb für die Ablauforganisation (z. B. Freigabewesen) nur die gespeicherte Produktstruktur interessiert und nicht mehr die verschiedenen Darstellungsformen.

Da eine Produktstruktur sowohl in zahlreichen Unternehmensbereichen (Konstruktion, Fertigung etc.) als auch teilweise vom Kunden benötigt wird, ergeben sich *unterschiedliche Struktursichten* (z.B. Konstruktions- (KSL), Fertigungs- (FSL), Montage- (MSL), Ersatzteil-Stücklisten), die bezüglich mehrerer Gesichtspunkte variieren können:

- unterschiedliche Strukturen  
(In einer funktionell strukturierten KSL könnte für ein Auto z. B. eine Hauptbaugruppe Bremsen definiert werden, worin der Hauptbremszylinder, alle vier Radbremszylinder und sämtliche Leitungs- und Verbindungselemente aufgeführt sind. Im Gegensatz hierzu könnte in einer fertigungstechnisch strukturierten FSL oder MSL, die den Zusammenbau beschreibt, jeder Radbremszylinder in einer separaten Gruppe (z. B. Einzelradaufhängung hinten rechts) aufgeführt sein (vgl. Bild B014obmZ).
- unterschiedlich aufgeführte Komponenten  
(Eine MSL enthält zusätzliche Positionen für z.B. Schmierstoffe wie etwa 5 L Öl).
- unterschiedliche beschreibende Merkmale  
(Eine FSL enthält zusätzlich die Spalte „Gewicht“).

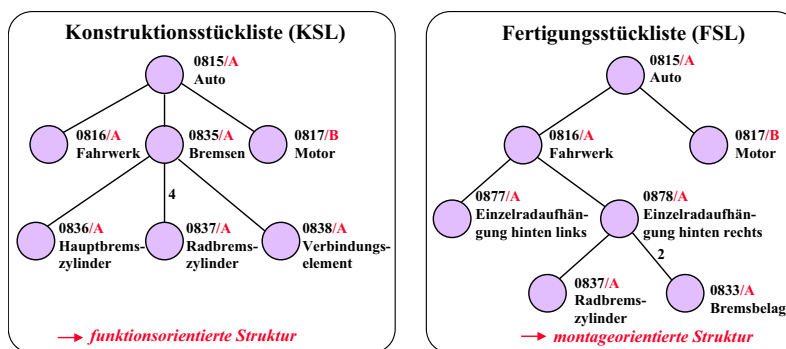


Bild (B014obmZ) Unterscheidung von Produktstruktursichten (BOMView types)

Für eine Baugruppe können mehrere produktbezogene Struktur-  
schichten (BOM View-Types) existieren, die sich unterscheiden bzgl.:

- Strukturierung
- Komponenten
- Merkmale

Nach [DIN199/2] ist die Unterscheidung in Konstruktionsstückliste  
(KSL) und Fertigungsstückliste (FSL) folgendermassen definiert:

KSL	Eine Konstruktionsstückliste KSL wird im Konstruktionsbereich im Zusammenhang mit den zugehörigen Zeichnungen erstellt und ist in der Regel auftrags- und fertigungsneutral.
FSL	Eine Fertigungsstückliste trägt in ihrem Aufbau und Inhalt den Gesichtspunkten der Fertigung Rechnung und entsteht in der Regel durch Aufbereitung und/oder Ergänzung aus der KSL.

Tabelle (T002obmZ)

Unterscheidung Konstruktionsstückliste (KSL) und Fertigungsstückliste (FSL)

Ein PLM-System sollte für ein Bauteilobjekt mehrere unterschiedliche Struktursichten unterstützen, damit derartige Anforderungen erfüllt werden können. Solange irgend möglich sollte jedoch versucht werden, möglichst nur eine einzige Struktursicht zu verwenden, da dies bedeutend einfacher und übersichtlicher in der Handhabung ist (vgl. [EKK-87]).

In sämtlichen Unternehmen, die mehrere Prozessschritte einer Produktentwicklung bearbeiten, wird es sich nicht vermeiden lassen, *unterschiedliche Aktualitätszustände* z. B. in der Konstruktion und Fertigung zu führen. Nachdem eine Änderung in der Konstruktion abgeschlossen ist, besteht ein neuer Konstruktionsstand (z. B. Revision B), der erst zu einem späteren Zeitpunkt für die Fertigung effektiv bzw. gültig wird, welche solange noch mit dem alten Fertigungsstand (z. B. Revision A) produziert.

PLM-Systeme bieten hierfür in der Regel mehrere Mechanismen an (mehrere Freigabestufen, Status-/Reifegrad-Konzepte, Effectivity bezüglich Datum oder Serien-Nr.).

Produktstrukturen in PLM-Systemen unterscheiden sich von herkömmlichen (rein statischen) Stücklisten insbesondere durch das Konzept der *generischen Konfiguration*. In der Regel werden Strukturbeziehungen über den Objekten (z. B. Bauteil) und nicht bezüglich deren Revisionen (z. B. Bauteil-Revision) definiert, wodurch Produktstrukturen dynamisch immer die neueste Revision eines Unterteils referenzieren. In Bild B012obmZ könnte die Produktstruktur des Autos beispielsweise ursprünglich mit einem Motor im Revisionszustand A definiert (und auch freigegeben) worden sein und sobald für diesen eine neue Revision B eingeführt wird, ist diese auch für die Produktstruktur des übergeordneten Autos gültig.

Einige PLM-Systeme unterstützen in einer Produktstruktur sowohl eine dynamische bzw. generische oder auch unpräzise (imprecise) genannte Strukturdefinition als auch eine feste (statische oder auch

präzise bzw. precise genannte) Definition sowie deren Kombination. Die Unterschiede dynamischer und statischer Strukturdefinition werden durch Bild B015obmZ verdeutlicht.

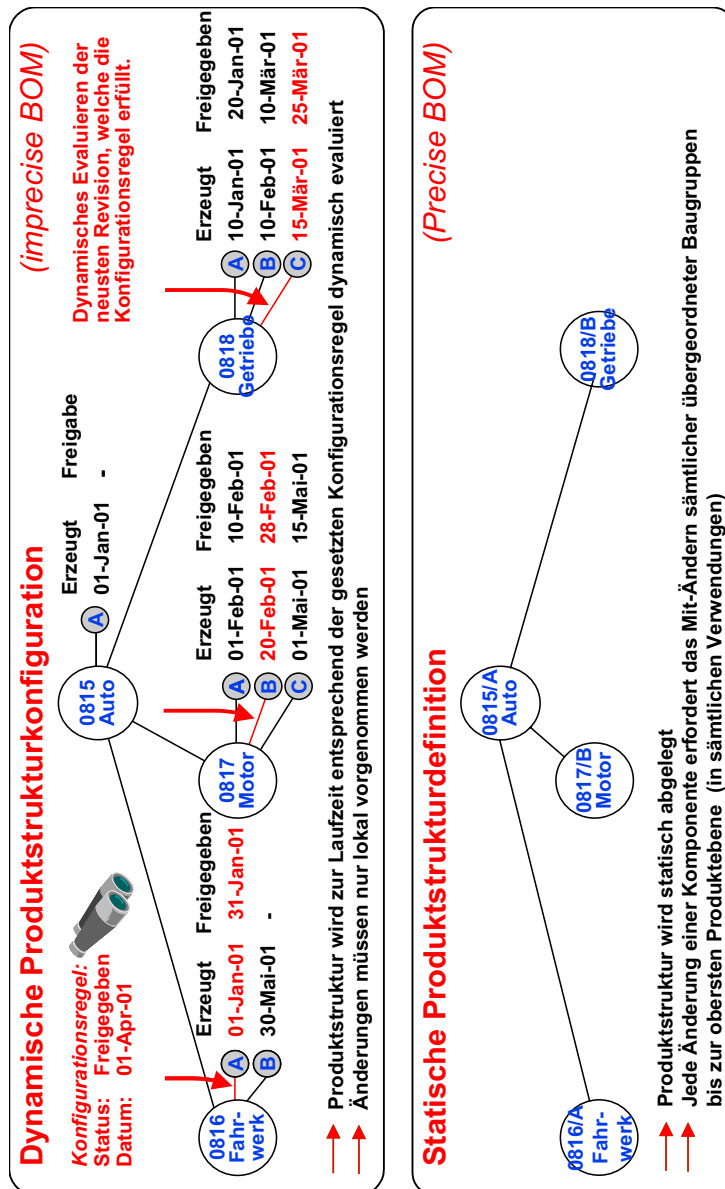


Bild (B015obmZ) Dynamische versus statische Produktstrukturdefinition

Dynamisch definierte Strukturbeziehungen sind über Objekten definiert (z. B. Bauteil 0817: Motor) und können bezüglich der konkreten Objektrevision (z. B. A, B oder C für Bauteil 0817: Motor) flexibel

abgefragt werden. In Abhängigkeit der vorgegebenen Konfigurationsregel (z. B. zeige die freigegebene Produktstruktur vom 1. April '01) werden die konkreten Objektrevisionen (z. B. Bauteilrevision 0817/B: Motor) erst zur Laufzeit dynamisch bestimmt. Als gültig konfiguriert wird hierbei jeweils die neueste Revision, welche eine vorgegebene Konfigurationsregel noch erfüllt. Entscheidender Vorteil der dynamischen Strukturdefinition sind aber nicht nur die flexiblen Abfrage- bzw. Konfigurationsmöglichkeiten, sondern insbesondere die einfache Einführung von Änderungen, d.h. neuer Revisionen. Um eine neue Revision (z. B. Revision C des Motors) in die Produktstruktur des Autos einzuführen, ist ausser deren "lokaler" Freigabe keine weitere Aktion erforderlich.

Demgegenüber sind *statische Strukturbeziehungen* direkt über den konkreten Objektrevisionen definiert (z. B. 0815/A Auto besteht aus 0817/B Motor) und können infolgedessen weder flexibel abgefragt werden noch in irgendeiner Weise Änderungen berücksichtigen. Das Setzen einer Konfigurationsregel hat daher keinen Einfluss auf eine statisch definierte Produktstruktur. Im Fall der Kombination von statischen und dynamischen Beziehungen innerhalb einer Produktstruktur bedeutet dies, dass das Setzen einer Konfigurationsregel nur die dynamischen Strukturbeziehungen beeinflusst. Für Änderungen hat dies erhebliche Konsequenzen, da die Freigabe einer neuen Revision eines Bauteils (z. B. Revision C des Motors 0817) keinerlei Einfluss auf dessen Verwendung in einer Produktstruktur hat. Um die neue Bauteilrevision 0817/C in eine statisch definierte Produktstruktur einzuführen (die bereits freigegeben und daher ohne Schreibrecht ist), muss zudem auch die übergeordnete Baugruppe (hier: 0815/A Auto) revidiert werden. Erst durch eine neue Revision 0815/B des übergeordneten Autos könnte infolgedessen die neue Motorrevision eingeführt werden. Im allgemeinen bedeutet dies für statische Produktstrukturdefinitionen, dass jede Änderung einer einzelnen Komponente über sämtliche Strukturebenen bis zur obersten Baugruppe „durchschlägt“, was deren Revisionsindex sehr schnell ins Uferlose ansteigen lassen kann! Trotz dieser gravierenden Nachteile werden in Branchen mit hohen Sicherheitsauflagen (z. B. in der Luft- und Raumfahrtindustrie) statische Produktstrukturen nach wie vor angewendet.

Zu empfehlen sind heute jedoch ausnahmslos dynamische Produktstrukturdefinitionen, deren Vorteile weitgehende Rationalisierungen ermöglichen. Im folgenden soll daher detaillierter auf die generischen Konfigurationsmechanismen für dynamische Produktstrukturen eingegangen werden.

## 4 Generische Konfigurationsmechanismen

Für die Konfiguration dynamischer Produktstrukturen sind verschiedenartige Mechanismen zu unterscheiden. Einerseits kann eine *ereignisabhängige* Sicht auf eine Produktstruktur abgefragt werden. Hierbei wird eine Revision nur dann konfiguriert (ist gültig), wenn bereits ein bestimmtes Ereignis (z. B. Freigabe) eingetroffen ist (Anwendung: zeige die freigegebene Produktstruktur). Zum anderen mag der Zustand einer Produktstruktur zu einem bestimmten (früheren oder späteren) Zeitpunkt oder auch zu einem bestimmten Fertigungslos interessieren. Hierfür kann eine Produktstruktur entweder *zeitabhängig* (ab/bis zu einem Datum) oder *produktionsbezogen* (ab/bis zu einer Serien-Nr.) abgefragt werden, was als Effectivity-Funktionalität bezeichnet wird (Anwendung: zeige den Stand der Produktstruktur am 1. Okt. 2001; zeige die Produktstruktur ab Serien-Nr. 12345). Darüber hinaus kann eine Produktstruktur auch mehrere Varianten beinhalten, die sich über *variantenbezogene* Sichten abfragen lassen. Bild B016obmZ zeigt die verschiedenartigen Konfigurationsmechanismen, welche in den folgenden Abschnitten näher betrachtet werden.

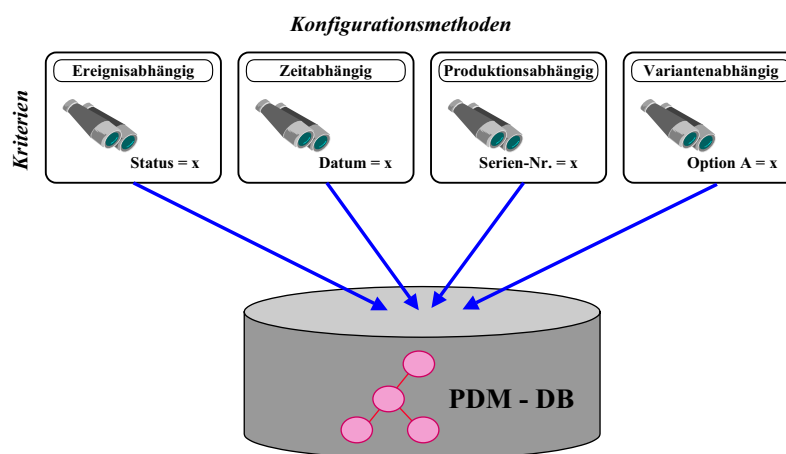


Bild (B016obmZ) Zusammenstellung generischer Konfigurationsmechanismen

## 4.1 Status-Konfiguration

Kennzeichen für das Eintreten eines Ereignisses ist die Existenz einer entsprechenden Statusinformation. Mit der Freigabe erhält eine Bauteilrevision beispielsweise den Status „Freigegeben“. Eine Produktstruktur kann somit entweder bezüglich dem neuesten Zustand abgefragt werden oder unter der Prämisse, dass ein bestimmter Status existiert. Falls eine derartige Statusvorgabe in einer Konfigurationsregel erfolgt, können für die Komponenten in einer Produktstruktur (bzw. deren Revisionen) drei Fälle eintreten:

- die neueste Revision besitzt bereits den geforderten Status  
→ die neueste Revision wird konfiguriert,
- nur eine ältere Revision besitzt den geforderten Status, die neueste jedoch nicht  
→ diese ältere Revision wird konfiguriert,
- keine Revision besitzt den geforderten Status  
→ es wird keine Revision konfiguriert.

Auf eine einmal in der PLM-Datenbank hinterlegte Produktstruktur lassen sich somit verschiedene Sichten erzeugen. Bild B017obmZ zeigt ein Beispiel für eine Konfiguration mittels Statusvorgabe, wobei für den Motor 0817 bei gesetzter Konfigurationsregel (Status = „Freigegeben“) der obige Fall 2 eintritt.

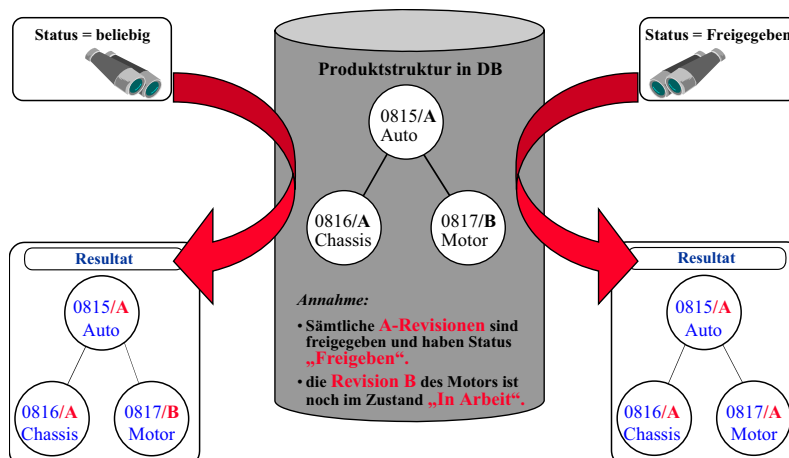


Bild (B017obmZ) Konfiguration mittels Statusvorgabe

Anhand mehrerer Stati (z. B. „in Produktion“) können auch die geforderten unterschiedlichen Aktualitätszustände einer Produktstruktur abgebildet werden. Zur Abfrage des aktuellen Konstruktionsstandes

muss der Status „Freigegeben“ existieren. Der Fertigungszustand kann durch die Konfigurationsregel Status = „in Produktion“ abgefragt werden.



## 4.2 Effectivity-Konfiguration

Unter dem Begriff Effectivity-Konfiguration werden sämtliche zeit- und produktionsabhängigen Datenbanksichten bezüglich Datum und Serien-Nr. zusammengefasst. Für die Datum- und Serien-Nr. Werte muss jeweils sorgfältig unterschieden werden, ob es sich um die gesetzte Effectivity-Regel handelt, die abgefragt werden soll (Betrachtungs-Effectivity oder Effectivity View), oder ob es sich um eine einschränkende Effectivity-Definition in Form von Objektattributen handelt.

### 4.2.1 Zeitabhängige Datenbanksichten

Zunächst soll lediglich die Abfrage unterschiedlicher Sichten auf die Datenbank betrachtet werden (vgl. Bild B018obmZ). Angenommen, zum Zeitpunkt X wird die Datenbank abgefragt, so kann entweder lediglich der aktuelle Zustand oder aber ein anderes Datum zur Betrachtung gewählt werden. Wird der Betrachtungszeitpunkt in die Vergangenheit gelegt, dann werden die Veränderungen derjenigen Benutzertransaktionen unsichtbar, deren Ende erst nach dem Betrachtungszeitpunkt liegt (T4 für Y2, T2 für Y1). Würde im Rahmen von T4 beispielsweise ein Bauteil 4711/A Flansch erzeugt, dann ist dieses zum Betrachtungszeitpunkt  $X = Y3$  selbstverständlich sichtbar, zum Zeitpunkt Y2 dagegen jedoch noch nicht. Abfragen in die Zukunft (Y4) können natürlich das Ende von aktuell laufenden Transaktionen (T5) sowie deren Ergebnis nicht vorhersehen. Zweckmässig sind Abfragen in die Zukunft daher nur, um gegebenenfalls bestehende Effectivity-Einschränkungen auswerten zu können.

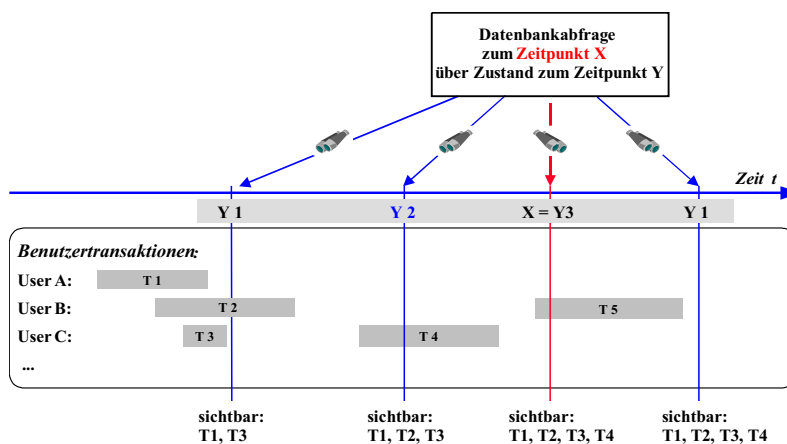


Bild (B018obmZ) Abfragesichten auf die PLM-Datenbank

#### 4.2.2 Zeitabhängige Datenbanksichten auf Effectivity-Definitionen

Unter einer Effectivity-Definition versteht man eine Einschränkung der Gültigkeit bzw. Verwendbarkeit freigegebener Objekte (für Start und/oder Ende) vorwiegend in der Produktionssteuerung. Die hauptsächliche Zielsetzung besteht darin, ein in der Konstruktion zum Zeitpunkt X freigegebenes Objekt für die Fertigung erst zu einem späteren Zeitpunkt ( $X+\Delta t$ ) einzuführen. Neben der Einschränkung des Start-Datums kann jedoch ebenso ein Ende-Datum, eine Start Serien-Nr. oder eine Ende Serien-Nr. sowie beliebige Kombinationen derselben angegeben werden. Da eine Effectivity-Definition immer nur zur Einschränkung dienen kann, müssen angegebene Start und Ende Daten sinnvollerweise jeweils nach dem Freigabezeitpunkt liegen. Weiterhin sind natürlich nur Ende-Restriktionen sinnvoll, die nach einer Start-Restriktion liegen (Ende-Datum nach Start-Datum, Ende Serien-Nr. grösser als Start Serien-Nr.) sowie lediglich positive Serien-Nr. Werte grösser 0.

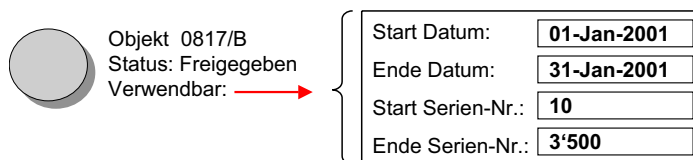


Bild (B019obmZ) Effectivity-Konzept

Das Effectivity-Konzept findet in folgenden Bereichen Anwendung:

- Gleichzeitige Einführung verschiedener Objekte in eine Produktstruktur (Start Datum)
- Verzögerte bzw. kontrollierte Einsteuerung von entwicklungsseitig bereits freigegebenen Objekten in den Produktionsablauf (Start Datum)
- Kontrollierte Aussteuerung von Objekten aus dem Produktionsablauf aufgrund z. B. veränderter Sicherheitsbestimmungen, Verfalldaten usw. (Ende Datum)
- Losabhängige Einsteuerung von entwicklungsseitig bereits freigegebenen Objekten in den Produktionsablauf, wie z. B. bei Prototypserien (Start Serien-Nr.)
- Losabhängige Aussteuerung von Objekten aus dem Produktionsablauf, z. B. aus Lagerbestandsgründen oder Lieferantenwechsel (Ende Serien-Nr.)

- Kontrollierte Zuordnung auftragsspezifischer Objekte zu einzelnen Produktionslosen (Start Serien-Nr., Ende Serien-Nr. ggf. kombiniert mit Start Datum und/oder Ende Datum)

Ferner ist zu beachten:

- Das Konzept ist nur sinnvoll für freigegebene Objekte bzw. Zeiträume nach einer Freigabe.
- Das Schreibrecht auf Effectivity-Informationen muss auch nach der Freigabe vorhanden sein und erfordert daher einen speziellen Autorisierungsmechanismus (z. B. über Rollenkonzept).

Sobald für ein Objekt eine Effectivity-Definition besteht, beeinflusst dies dessen Konfiguration in Abhängigkeit des Betrachtungs-Datums und/oder dessen Serien-Nr.. Verdeutlicht sei dies am Beispiel des Motors 0817, dessen neue Revision B mit einer Effectivity-Definition freigegeben wird, wodurch deren Gültigkeitsbereich eingeschränkt wird (vgl. Bild B020obmZ für Datum-Effectivity sowie Bild B021obmZ für Serien-Nr.-Effectivity). Für den Fall der Definition einer Datum-Effectivity wird beispielsweise auch nach dem Freigabezeitpunkt auf eine Konfigurationsregel bezüglich Status und Datum die ältere A-Revision konfiguriert, falls das Betrachtungsdatum entweder vor einem Effectivity-Start oder nach einem Effectivity Ende Datum gesetzt wird. Falls bis zum Ende des definierten Gültigkeitsbereichs (Effectivity Ende) keine neue Revision C eingeführt oder aber das Effectivity Ende aufgehoben wird, wird automatisch wieder die Vorgänger-Revision A konfiguriert. Die Effectivity Konfiguration mittels Serien-Nr. verhält sich diesbezüglich völlig äquivalent (vgl. Bild B020obmZ und Bild B021obmZ).

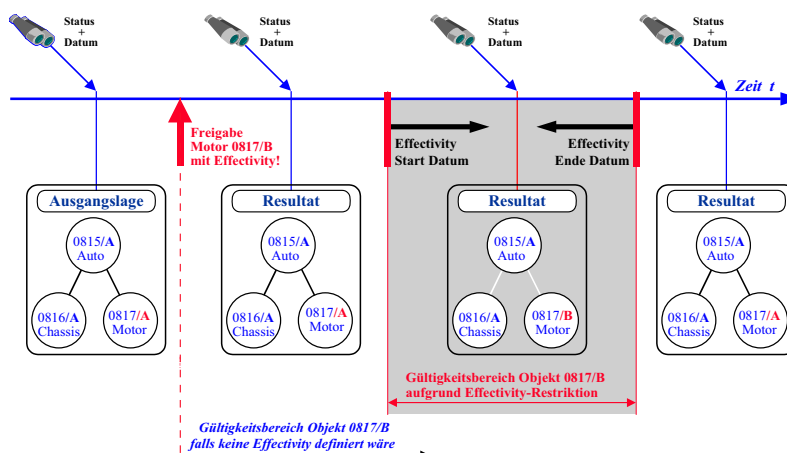


Bild (B020obmZ) Konfiguration mittels Effectivity (Datum)

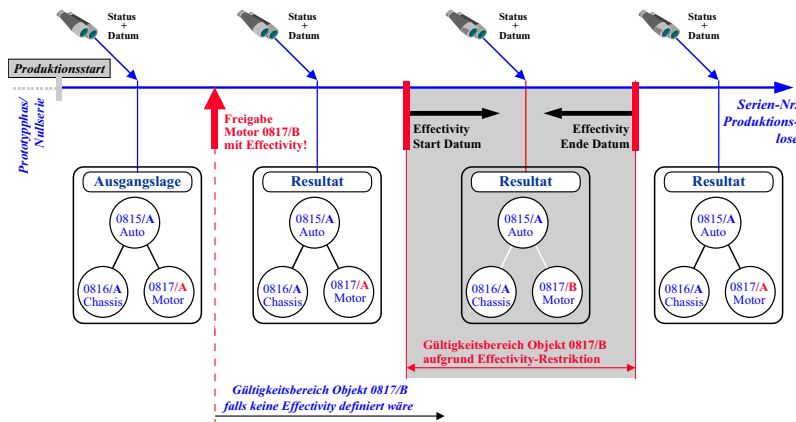


Bild (B021obmZ) Konfiguration mittels Effectivity (Serien-Nr.)

Etwas komplizierter wird die Sachlage bei einer Kombination von Effectivity-Definitionen bezüglich Datum und Serien-Nr.. Da die Zusammenhänge von Serien-Nr. und Datum im allgemeinen nicht ausreichend bekannt sind und zudem häufig geändert werden müssten, empfiehlt sich eine separate Betrachtung beider Effectivity-Kriterien anhand zweier Achsen (vgl. Bild B022obmZ). Der Gültigkeitsbereich bezüglich beider Kriterien lässt sich hierdurch als Fläche darstellen, die bei einer kombinierten Konfigurationsregel (bezüglich Status, Datum und Serien-Nr.) „getroffen“ werden muss, um eine Revision (hier: 0817/B Motor) zu konfigurieren. Je nach vorliegendem Informationsbedarf des Benutzers kann eine Abfragesicht auch nur bezüglich einem Effectivity-Kriterium formuliert werden (z. B. Status und Serien-Nr. oder Status und Datum) wobei dann dieses allein für die Konfiguration evaluiert wird.

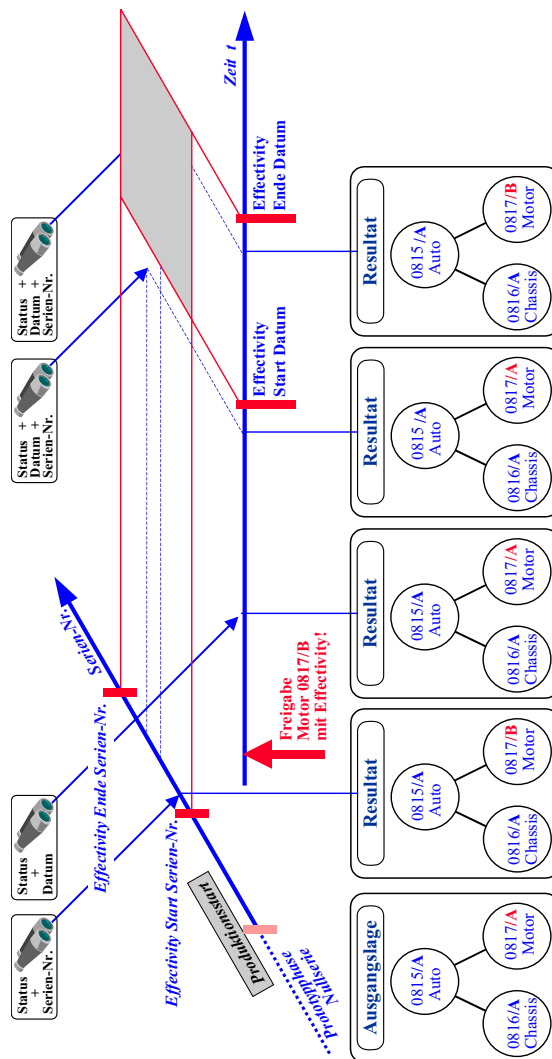


Bild (B022obmZ) Konfiguration mittels Effectivity (Datum kombiniert mit Serien-Nr.)

Bei den bisherigen Betrachtungen stand die Anwendung und Funktionalität der Effectivity-Konfiguration im Vordergrund, wobei lediglich eine beispielhafte Produktstruktur diskutiert wurde. Unter der Prämisse einer Mehrfachverwendung von Objekten und Effectivity-Definitionen in neueren Produktstrukturen soll nunmehr die Auswirkung der Effectivity auf verschiedene Verwendungen interessieren. Entscheidend hierfür ist, wie bzw. an welcher Stelle die Effectivity-Definition erfolgt (vgl. Bild B023obmZ):

- für die Objekt-Revision(Revision Effectivity) oder
- für die Strukturbeziehung(Occurrence Effectivity).

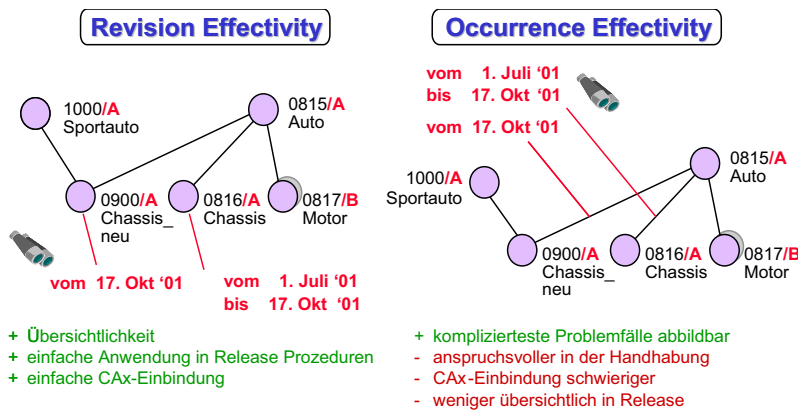


Bild (B023obmZ) Methoden zur Definition von Effectivity

Falls die *Effectivity* für die Objekt-Revision definiert wird, erhält sie dadurch globalen Charakter und somit Gültigkeit für sämtliche Verwendungen, d.h. Oberbaugruppen. Im linken Beispiel in B023obmZ beeinflusst das Effectivity-Datum infolgedessen sowohl das Auto 0815/A als auch das Sportauto 1000/A.

Demgegenüber hat eine *Effectivity-Definition für eine Strukturbeziehung* nur lokalen Charakter und somit nur Gültigkeit für die betreffende Oberbaugruppe. Das rechte Beispiel in Bild Bild B023obmZ zeigt, dass die Occurrence-Effectivity bezüglich dem Auto 0815/A keinerlei Einfluss auf die Verwendung des neuen Chassis 0900/A im Sportauto 1000/A hat.

Leistungsfähige PLM-Systeme unterstützen beide Methoden, so dass der Benutzer die jeweils bestgeeignete für seinen Anwendungsfall auswählen kann. Aufgrund der besseren Übersichtlichkeit ist nach Möglichkeit - soweit es zweckmässig ist - die Revision-Effectivity zu bevorzugen. Nur in begründeten Ausnahmefällen (wo die Revision-Effectivity nicht mehr ausreicht), sollte die Occurrence-Effectivity angewendet werden, da diese schwieriger in der Handhabung ist, insbesondere im Zusammenhang mit Release-Prozeduren und CAx-Einbindungen.

Abschliessend zur Effectivity-Thematik sei darauf hingewiesen, dass dies eine Domäne von PPS-Systemen ist. Falls in einem Unternehmen in Anlehnung an Bild B012obmZ ein PPS-System existiert sowie ggf. eine PLM/PPS-Kopplung realisiert ist, kommt die Effectivity erst im PPS-System zu tragen. Innerhalb des PLM-Systems handelt es sich daher nur um eine sogenannte „geplante Effectivity“, die anschliessend an ein PPS-System übergeben wird. Insgesamt wird aus den bislang diskutierten Konfigurationsmechanismen jedoch die um ein

Vielfaches grössere Funktionalität von PLM-Produktstrukturen gegenüber herkömmlichen Stücklisten deutlich.

Eine spezielle Form der generischen Konfiguration stellt die *Variantenbildung* dar, der anwendungsseitig eine eigene Semantik zukommt, weshalb diese gesondert zu betrachten ist.

## 5 Variantenbildung

Die Variantenbildung ist ein sehr häufig eingesetztes Rationalisierungsmittel, wodurch eine grössere Flexibilität für verschiedene Kundenbedürfnisse erreicht wird und infolge der verringerten Datenredundanz erhebliche Verbesserungen für den Änderungsdienst, die Disposition und andere Produktionsaufgaben erzielt werden. Unter Varianten sind nach [[DIN199/2]] Objekte ähnlicher Form und/oder Funktion mit einem in der Regel hohen Anteil identischer Baugruppen oder Einzelteile zu verstehen. Anstatt für *jede* mögliche Variante eine vollständige Produktstruktur anzulegen, wird durch *eine* gemeinsame Variantenstruktur (Variantenstückliste) die gesamte Vielfalt eines sogenannten Produkttyps zusammenfassend beschrieben. Eine Variante stellt hierbei eine konkrete Ausführung eines Produkttyps dar. Demzufolge müssen bei der Variantenbildung zwei grundlegende Aspekte unterschieden werden:

- *Definition* einer Variantenstruktur für einen Produkttyp (Definition sämtlicher möglicher Kombinationen),
- *Ableiten* einer konkreten Variantenausprägung (Auswahlentscheidung bezüglich sämtlicher variabler Positionen).



## 5.1 Konzepte der Variantendefinition

Um einen Überblick über die Variantenbildung zu geben wird nachfolgend zunächst auf den ersten Aspekt der Variantenstrukturdefinition eingegangen. Bild B023obmZ verdeutlicht die anwendungsseitigen Variationsmöglichkeiten für Produktstrukturen, woraus sich drei grundlegende *Variationskonzepte* ableiten lassen:

- *Teilevariation*  
Für ein Unterteil sind verschiedene Ausprägungen möglich (die als eigenständige Objekte existieren und durch einen Teile-Typ verallgemeinert werden).
- *Strukturvariation*  
Für ein Oberteil sind verschiedene Strukturbeziehungen zu Unterteilen möglich, die in ihrer Kombination als Strukturvarianten zusammengefasst werden.
- *Mengenvariation*  
Für eine Strukturbeziehung sind verschiedene Mengenwerte zulässig.

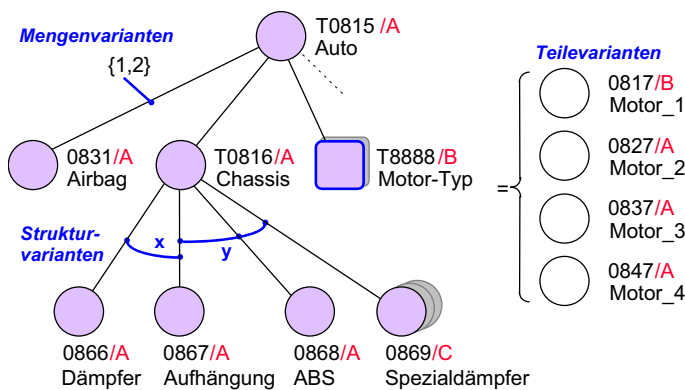


Bild (B024obmZ) Variationskonzepte von Produktstrukturen; Verwendungsregeln:  
Motor\_1 z.B. nur zulässig für Fahrwerkstruktur y

Hinsichtlich der *Auswahlmöglichkeit* sind für jede Variationsmethode darüber hinaus folgende Arten zu unterscheiden:

Festvarianten	Definitivteile, die immer enthalten sein müssen (sog. Gleichteile),
Mussvarianten	Alternativteile, die sich gegenseitig ausschliessen und wovon eines enthalten sein muss,

Tabelle (T003obmZ) Arten von Variationsmethoden hinsichtlich der Auswahlmöglichkeit

Kannvarianten | Fakultativteile, die unabhängig von anderen Teilen zusätzlich enthalten sein können.

Tabelle (T003obmZ) Arten von Variationsmethoden hinsichtlich der Auswahlmöglichkeit

## 5.2 Benummerung von Produktvarianten

Durch die Kombination von Variationskonzepten und Auswahlmöglichkeiten lassen sich komplexe Variantenstrukturen definieren. In Abhängigkeit der Variantenvielfalt bestehen zwei Methoden für die Benummerung der einzelnen Varianten eines Produkttyps:

- vollständig benummerte Produktvarianten (Katalogvarianten),
- unvollständig benummerte Produktvarianten (Kundenvarianten).

Methode	vollständige Variantenbenummerung	unvollständige Variantenbenummerung
Merkmal	Sämtliche Varianten sind als eigenständige Objekte definiert und besitzen daher eine Sachnummer.	Sämtliche möglichen Varianten werden unter einer Typennummer zusammengestellt, ohne eine eigene Objektdefinition und damit Sachnummer zu besitzen.
Vorgehen	Die Gleichteile der Produktvarianten werden zu einer Gleichteilegruppe zusammengefasst. Gleich- und Variantenteilen werden Sachnummern und Teilstammsätze zugewiesen, wodurch die Produktvarianten eindeutig festgelegt sind.	Im Auftragsfall dienen die Kundenauftragsnummer und die Typennummer als Sachnummer. Die auftragsabhängige Produktvariante wird in Abhängigkeit der Kundenspezifikationen im Rahmen der zugelassenen Varianten für die Produktstruktur bestimmt.
Anwendung	Programmfertiger mit begrenzter Variantenzahl	Auftragsfertiger mit beliebig grosser Variantenvielfalt
Bezeichnung	Katalogvarianten	Kundenvarianten

Tabelle (T004obmZ) Methoden der Variantenbenummerung

Im Falle weniger Varianten, die zudem häufig benötigt werden, bietet sich eine vollständige Benummerung an. Besteht demgegenüber eine unbegrenzte Variantenvielfalt mit zum Teil nur selten oder voraussichtlich nie benötigten Kombinationen, so bietet sich die unvollständige Benummerung der Produktvarianten an. Häufig finden sich auch Mischformen beider Definitionsarten.

### 5.3 Ableitung auftragspezifischer Strukturausprägungen

Ausgehend von einer solchen Variantenstrukturdefinition kann eine auftragspezifische Strukturausprägung abgeleitet werden (vgl. Bild B025obmZ). Hierbei erfolgt zunächst eine Auswahl bezüglich sämtlicher Katalog- und/oder Kundenvarianten, die ein Kunde entsprechend seinen Anforderungen durchführt. Je nach Produktbranche und Fertigungsart kann zusätzlich eine Nach-Konfiguration stattfinden, wobei der Kunde ganz spezifische Zusatzwünsche äussert, die über die Variantenstrukturdefinition hinausgehen und gegebenenfalls völlige Neukonstruktionen von Komponenten erfordert. Ergebnis ist abschliessend eine Auftragsstückliste, die sich zusammensetzt aus:

- Gleichteilen,
- Varianten
  - Katalogvarianten und/oder
  - Kundenvarianten,
- kundenspezifischen Zusatzkomponenten.

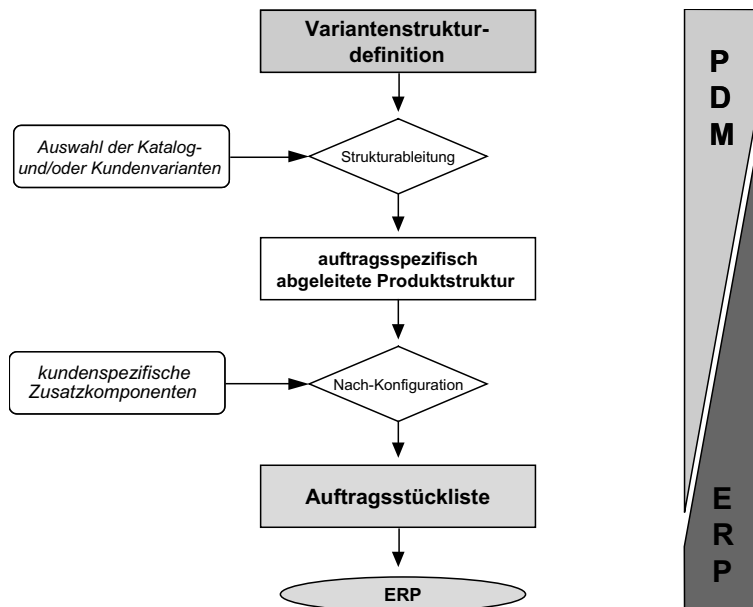


Bild (B025obmZ) Prozess zur Konfiguration einer Auftragsstückliste

Diese Funktionalität zur Ableitung spezifischer Auftragsstücklisten kann sowohl in einem PLM-System als auch in einem ERP-System realisiert werden. Kopplungsansätze basieren dabei entweder auf der

(generischen) Variantenstrukturdefinition oder auf der abgeleiteten (konkreten) Auftragsstückliste, die als Schnittstelle definiert sein können und zwischen den Systemen ausgetauscht werden.

Zur Verdeutlichung ist in Bild B026obmZ ein Beispiel einer Auftragsstückliste dargestellt. Das bestellte Auto ist in dieser Kombination bislang noch nicht aufgetreten (unvollständige Variantenbenummerung) und muss daher rein auftragspezifisch benummert werden. Hierfür wird eine Kombination aus der Auftragsnummer Ax und der Typennummer T0815/A verwendet. Bezüglich der definierten Varianten sind für den Airbag die Menge 2, der Motor\_3 und die Strukturvariante y des Chassis ausgewählt. Da diese Strukturvariante sehr häufig vorkommt, besitzt sie eine eigene Sachnummer 0873/A, unabhängig von diesem Auftrag (vollständige Benummerung). Als Beispiel eines rein kundenspezifischen Zusatzwunsches, der bislang noch nie aufgetreten war, ist hier ein Internet-Anschluss gewählt, der im Rahmen einer Nach-Konfiguration vom Kunden zusätzlich gewünscht wird. Der grundsätzliche Unterschied zwischen einer *Variantendefinition* und einer *Variantenausprägung* bzw. Auftragsstückliste wird durch den Vergleich von Bild B024obmZ und Bild B026obmZ veranschaulicht.

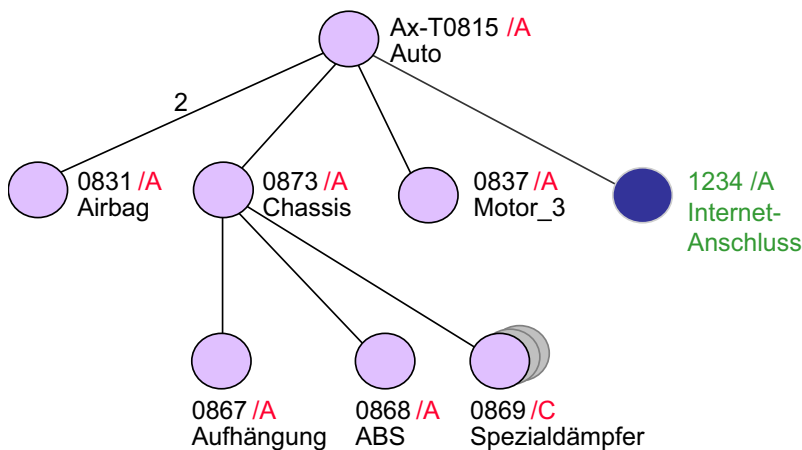


Bild (B026obmZ) Beispiel einer spezifischen Auftragsstückliste

## 5.4 Methoden von PLM-Systemen zur Variantendefinition

Abschliessend zur Variantenthematik muss noch darauf hingewiesen werden, dass PLM-Systeme niemals gleichzeitig alle der vorstehend diskutierten „anwendungsseitigen“ Variationskonzepte zur Variantendefinition (vgl. Bild B024obmZ) unterstützen. In der Regel wird entweder nur eine eingeschränkte Strukturvariation (bezüglich einer einzelnen Strukturbeziehung) oder die Teilvariation als systemseitiges Variantenmodell verwendet [Mon-97]. Um eine Variantenstruktur mit ähnlicher Komplexität wie in Bild B024obmZ zu modellieren, steht dem Anwender infolgedessen nur ein Variationskonzept zur Verfügung (Strukturvariation *oder* Teilvariation). Nötigenfalls müssen infolgedessen fehlende Variationskonzepte mittels dem jeweils vorhandenen Konzept nachgebildet werden, obwohl dessen Semantik eigentlich nicht zutreffend ist.

Zur Verdeutlichung ist das Variantenbeispiel von Bild B024obmZ nachfolgend sowohl auf eine reine Strukturvariation, als auch auf eine reine Teilevariation abgebildet, wodurch deren Gegensätzlichkeit zum Ausdruck kommt. Bei der Strukturvariation, die beispielsweise im Teamcenter System eingesetzt wird, geht man von immer gültigen Teilekomponenten aus und versieht lediglich die Strukturbeziehungen mit Bedingungen, wodurch Strukturen entweder gültig oder ungültig sein können (vgl. Bild B027obmZ).

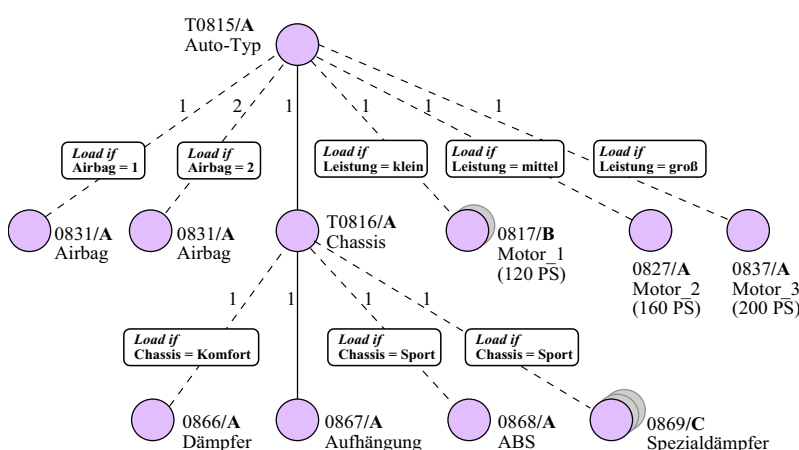


Bild (B027obmZ) Variantendefinition mittels Strukturbedingungen (Strukturvariation)

Optionen sind global definiert, so dass deren Konfiguration Auswirkungen auf mehrere Bereiche einer Variantendefinition haben kann.

Option	Values
Leistung	{klein, middle, gross}
Airbag	{1, 2}
Fahrverhalten	{Komfort, Sport}

Tabelle (T005obmZ) Optionen und Values

## 6 Autorisierung

Die Autorisierung ist von zentraler Bedeutung im Product Lifecycle Management, da im Gegensatz zu den herkömmlichen Insellösungen nun eine integrierte Datenhaltung vorliegt, wo ohne Zugriffskontrolle grundsätzlich jeder auf sämtliche Daten – im positiven wie im negativen Sinne – zugreifen könnte. Infolgedessen wird ein Konzept zum Datenschutz benötigt, wofür PLM-Applikationen in der Regel leistungsfähige Autorisierungsmechanismen anbieten. Diesbezüglich werden in den nachfolgenden Betrachtungen zunächst die

- Grundkonzepte der Autorisierung und dann die
- Methoden zur Autorisierung in heutigen PLM-Applikationen diskutiert.



## 6.1 Grundkonzepte der Autorisierung

Von grundlegender Bedeutung für den Mechanismus der Autorisierung ist der Zeitpunkt des Zugriffs im Lebenslauf eines Objekts. Diesbezüglich sind drei Hauptphasen zu unterscheiden, wobei die Autorisierung jeweils auf unterschiedliche Weise geregelt wird (vgl. Bild B028obmZ). Während der ersten Phase „Working“ ist ein Objekt quasi in Arbeit und die Zugriffsrechte können bei Bedarf beliebig verändert werden, um gegebenenfalls einem anderen User, einer anderen Rolle, Gruppe usw. die benötigten Privilegien einzuräumen. Sobald sich ein Objekt jedoch in einem formalen Prozess befindet, sind sämtliche Zugriffsrechte in der Prozessdefinition festgelegt und werden ausschliesslich hierüber kontrolliert, d.h. sie können i.d.R. vom Benutzer – auch dem Eigentümer (Owning User) – nicht beeinflusst werden! Nach einem erfolgreichen Prozessdurchlauf erhält ein Objekt in der Regel einen Status zugewiesen, womit auch eine feste Definition von Zugriffsrechten verbunden ist. Zum Schutz von z. B. freigegebenen Objekten dürfen diese Privilegien (i.d.R. nur noch World: Read) nicht mehr geändert werden. Die Unterscheidung der in Bild B028obmZ dargestellten drei Hauptphasen ist somit von grundlegender Bedeutung für die Autorisierung.

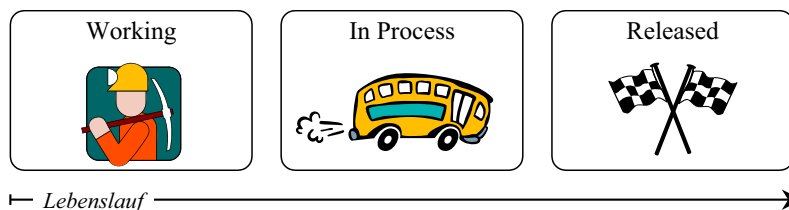


Bild (B028obmZ) Unterscheidung dreier Hauptphasen für die Autorisierung

Ausgangspunkt eines jeden Autorisierungsentscheides ist ein Benutzer, der eine Funktion auf ein Objekt anwenden möchte. Für die Autorisierung stehen daher drei wesentliche Zugriffsaspekte im Vordergrund, die zusammen ein Zugriffstripel bilden (Bild B029obmZ):

- Accessor(User im Kontext der Sitzung),
- Anwendungsfunktion (z.B. Read oder Kopieren auf neue Revision),
- Objektprivilegien (Menge definierter Zugriffsrechte).

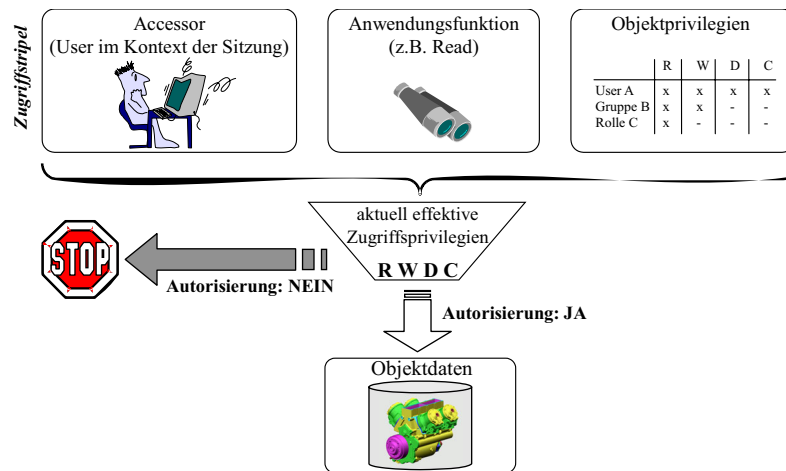


Bild (B029obmZ) Aspekte der Autorisierung

## Accessor

Unter dem Begriff Accessor ist der „Benutzer im Kontext seiner Sitzung“ zu verstehen, d.h. die Kombination aus User, aktueller Gruppe und Rolle. Der Benutzer alleine ist zumindest in allgemeiner Hinsicht nicht ausreichend, da ein Benutzer häufig mehrere Rollen annehmen kann.

Die Unterscheidung von Gruppen und Rollen und deren zweckmässige Anwendung ist hierbei von grundlegender Bedeutung:

- Gruppen
  - repräsentieren ein Entwicklungsteam bzw. eine organisatorische Unternehmenseinheit,
  - üblicherweise werden Zugriffsrechte allgemein auf Gruppenmitglieder beschränkt.
- Rollen
  - repräsentieren funktionale Verantwortlichkeiten,
  - üblicherweise werden Zugriffsrechte für spezifische Informationsaspekte anhand von Rollen definiert (Konstrukteure haben Schreibzugriff auf CAD-Datasets, NC-Programmierer haben Zugriff auf CAM-Datasets),
  - Zugriffsrechte für Rollen können sowohl abhängig von der Gruppe als auch global vergeben werden (ggf. gibt es nur einen NC-Programmierer für alle Gruppen).

In allgemeiner Hinsicht können von der funktionellen Verantwortlichkeit her drei grundsätzliche Benutzerkategorien, d.h. charakteristische Rollen unterschieden werden, die zur Wahrnehmung ihrer Aufgaben entsprechende Privilegien in der Datenbank benötigen:

- Systemadministratoren
  - unterhalten das System,
- Projektleiter
  - bestimmen die Zugriffsrechte und
- Sachbearbeiter
  - erzeugen Daten (CAD, CAM, etc.).

Objektdaten werden daher vornehmlich von Sachbearbeitern erzeugt, wobei insbesondere der erzeugende User und dessen aktuelle Gruppenzugehörigkeit interessiert. Ein Objekt „gehört“ daher immer einem User (Owning User) und einer Gruppe (Owning Group).

Gegenüber diesen beiden primären Hauptmerkmalen eines Objekts in punkto Zugriff muss der bereits erwähnte Kontext der Sitzung ermittelt werden. Hierbei kann eine mehr oder weniger feine Untergliederung von Accessors getroffen werden, die idealerweise von einem PLM-System unterstützt werden sollte (vgl. Tab. 1.3). Da für einen Zugreifenden gleichzeitig mehrere Accessors zutreffend sein können, muss in der Regel eine Priorisierung erfolgen, wobei immer vom Owning User absteigend bis zum allgemeinsten Zugriffsprofil World gegliedert wird.

Nr.	Accessor	Bemerkung
1	Owning User	Eigentümer des Objekts
2	User	spezifischer User (z. B. Müller)
3	Role in Owning Group	spezifische Rolle in der Eigentümer-Gruppe (z. B. CAD-Designer)
4	Role in Group	spezifische Rolle
5	Owning Group	Gruppe, der das Objekt gehört
6	Administrator	Systemadministrator
7	Group	spezifische Gruppe
8	World	Menge aller Benutzer

Tabelle (T006obmZ) Beispielhafte Rangordnung der Accessors

### Anwendungsfunktion

Der zweite wesentliche Zugriffsaspekt ist die Anwendungsfunktion, die ein Benutzer ausführen will. Hierfür können eine oder mehrere Zugriffsprivilegien von einem oder mehreren Objekten benötigt werden. Im einfachen Fall eines Read-Zugriffs auf ein atomares Objekt wird lediglich eine Read-Erlaubnis auf dasselbe benötigt. Für den

etwas komplexeren Fall einer Revisionierung wird je nach systemseitiger Modellbildung eine Kopiererlaubnis auf die betreffende Objektrevision und eine Schreiberlaubnis für das eine Sachnummer repräsentierende Objekt benötigt. Unabhängig davon, ob eine Funktion ein oder mehrere Zugriffsprivilegien für ein oder mehrere Objekte benötigt, ist zu beachten, dass hier zunächst von „geforderten Privilegien“ (!) auszugehen ist.

### Objektprivilegien

Der dritte Zugriffsaspekt betrifft die Objekte mit ihren Privilegien selbst. Als kleinstes Granulat sind hier die Einzelprivilegien Read, Write etc. zu unterscheiden, wobei gesamthaft eine ganze Liste möglichst flexibel erweiterbarer Privilegien zu unterscheiden ist (vgl. Tabelle T006obmZ)

Privileg		Bemerkung
Read	R	Recht zum Lesen
Write	W	Recht zum Schreiben
Delete	D	Recht zum Löschen
Change	C	Recht zum Ändern der Rechte
Print	P	Recht zum Printen
Copy	c	Recht zum Kopieren
Archive	A	Recht zum Archivieren
Import	I	Recht zum Importieren
Export	E	Recht zum Exportieren
...		...

Tabelle (T007obmZ) Beispiele von Privilegien

Im Hinblick auf die Unterstützung dieser Privilegien und der Art und Weise ihrer Definition bestehen einige Unterschiede bei den heutigen PLM-Applikationen, die im Gesamtzusammenhang der verfügbaren Autorisierungsfunktionalität zu vergleichen sind.

#### 6.1.1 Methoden der Autorisierung

Aufbauend auf dem vorstehend dargelegten Grundverständnis werden im folgenden die in marktüblichen PLM-Applikationen verfügba-

ren Autorisierungskonzepte vorgestellt und analysiert. Hierbei sind drei wesentliche Methoden zu unterscheiden:

- objektbasierende 3 Stufen-Autorisierung,
- objektbasierende ACL-Autorisierung,
- regelbasierende ACL-Autorisierung.

### Objektbasierende 3 Stufen-Autorisierung

Abgesehen vom primitiven Schutz transportabler Speichermedien (DAT-Tape, Disketten ect.) durch einen binären Schalter auf Read Only oder Writable stellt die 3 Stufen-Autorisierung die einfachste Methode für einen Zugriffsschutz dar, mit der eine PLM-Applikation betrieben werden kann. In Anlehnung an die Unix-Zugriffsdefinition werden nur drei Accessor-Typen unterstützt:

- Eigentümer (Owning User),
- Gruppe (Owning Group),
- World (Menge aller Benutzer).

		<i>Privilegien</i>	
		<b>R</b>	<b>W</b>
<i>Accessors</i>	Owner	Y	Y
	Group	Y	Y
	World	Y	

Bild (B030obmZ) Beispiel einer 3 Stufen-Autorisierung

Hinsichtlich der Privilegien werden häufig nur Read und Write unterschieden, wodurch Zugriffsrechte nur sehr rudimentär definiert werden können. Erwähnt sei an dieser Stelle, dass oftmals parallel eine objektunabhängige Autorisierung auf Anwendungsfunktionen (z. B. Archive-Funktion) angeboten wird, wodurch die übrigen Objektprivilegien zumindest teilweise nachgebildet werden können. Hauptnachteil der 3 Stufen-Autorisierung ist jedoch die Beschränktheit auf die drei allgemeinen Accessors, ohne jede Möglichkeit, spezifische Zugriffsrechte für andere Gruppen (d.h. Projektteams oder Abteilungen) zu vergeben. Da nur allgemein die Privilegien für die Owning Group und eben nicht für eine spezielle Gruppe (z. B. Arbeitsvorbereitung) vergeben werden kann, spricht man hier auch von einer Pseudo-Gruppe

und einer Pseudo-Autorisierung. Anwendung findet diese Autorisierungsmethode beispielsweise im PLM-System Axalant.

### Objektbasierende ACL-Autorisierung

Deutlich leistungsfähiger präsentiert sich die Methode der objektspezifischen ACL-Autorisierung. Hierbei wird eine sogenannte Access Control List (ACL) eingeführt, die aus mehreren Einträgen besteht, wo jeweils für einen Accessor dessen Privilegien aufgeführt sind (vgl. Bild B031obmZ). Da die ACL-Liste um weitere Accessors beliebig erweiterbar ist, besteht eine erheblich grössere Flexibilität gegenüber der obenerläuterten 3 Stufen-Autorisierung. Die Privilegien sind in der Regel zwar vielfältiger, aber nicht unbedingt erweiterbar. Angewendet wird diese objektbasierende ACL-Methode in der Mehrzahl der modernen objektorientierten PLM-Systeme, wobei als Beispiel der HP/WorkManager genannt sei.

		<i>Privilegien</i>				
		<b>R</b>	<b>W</b>	<b>D</b>	<b>C</b>	<b>c</b>
<i>Accessors</i>	Owner	Y	Y	Y	Y	Y
	Owning Group	Y	Y			
	Administrator	Y	Y	Y	Y	Y
	Engineering/CAD-Designer	Y	Y			Y
	World	Y				Y
	User Müller	Y	Y		Y	Y

Bild (B031obmZ) Beispiel einer objektbasierenden ACL-Autorisierung

### Regelbasierende ACL-Autorisierung

Einen völlig neuartigen Ansatz stellt die regelbasierende ACL-Autorisierung dar, die sich von den beiden oben erläuterten Methoden deutlich abhebt und die Autorisierungsfunktionalität insbesondere für die Working-Phase erheblich vergrössert. Bislang lag die Verantwortung für die Autorisierung allein beim Benutzer, der sich damit jedoch vielfach gar nicht beschäftigen will oder sogar kann. Insbesondere unerfahrene Benutzer sollten nicht nur vor unbeabsichtigten Schreibzugriffen geschützt werden, sondern auch vor ungewollten Änderungen der Autorisierung. Weiterhin muss einem Unternehmen an einer möglichst einheitlichen, transparenten und strategiekonformen Autorisierung gelegen sein. Diesbezüglich ist nicht nur ein möglichst leistungsfähiger Autorisierungsmechanismus gefordert, der mit

herkömmlichen ACLs bereits zur Verfügung stünde, sondern ebenso ein möglichst flexibler mit einfacher Handhabung. Um beispielsweise in einem laufendem Entwicklungsprojekt einem weiteren Projektteam Zugriffsrechte einzuräumen, müsste für sämtliche bereits erarbeiteten Objektdaten ein ACL-Eintrag hinzugefügt werden (ggf. mehrere hundert oder mehr). Mit einer rein objektbasierenden Autorisierungsdefinition besteht daher kaum ausreichende Flexibilität zur Erweiterungen der Zugriffsrechte auf bestehende Objektdaten.

Demgegenüber präsentiert sich die regelbasierende ACL-Autorisierung um ein Vielfaches leistungsfähiger, wobei zwei Basiskonzepte Anwendung finden:

- Regelbaum (Rule Tree)  
Hierarchie von Regeln zur Klassifizierung von Objekten anhand von Attributen und Verknüpfung mit ACL-Definitionen (Syntax: Bedingung {Wert} → Named ACL).
- Global definierte ACL-Objekte (Named ACL's)  
Eigenständige Objektdefinition von ACLs die vollständig getrennt und unabhängig von den Objektdaten erfolgt und daher beliebig referenziert werden kann (Syntax: Accessor: Privilege {Name} - {Yes/No/Not Set}).

Der Regelbaum dient dem Aufbau einer Klassifizierungsstruktur, um Objekte für eine typbezogene und damit instanzunabhängige Definition von Zugriffsprivilegien sinnfällig zu gliedern. In erster Linie müssen dabei die Hauptphasen der Autorisierung Berücksichtigung finden (Working, In Job, Released). Insbesondere für die Working-Phase ist eine detaillierte Untergliederung zweckmässig anhand dem vorliegenden Objekttyp (3D CAD-Modell, CAD-Drawing, Word-Text, etc.) und der Angabe von Accessor-Restriktionen (Owning User, Owning Group etc.). Tabelle T007obmZ gibt einen beispielhaften Überblick über die am häufigsten benötigten Regeln.

Rule	Bemerkung
Has Status	besitzt Status (z.B. Released)
In Job	befindet sich in einem formalen Workflow-Prozess (z.B. Freigabeprozess)
Has Class	ist von Objektklasse (z.B. 3D-Modell)
Has Name	besitzt Benennung
Owning User/Group/Site	gehört Eigentümer, Gruppe, Standort
Is local	gehört dem hiesigen Standort
Has Object ACL	hat zusätzliche Objekt-ACL
Has Bypass	Umgehen der Autorisierung durch den Systemadministrator
...	...

Tabelle (T008obmZ) Beispiele von Regeln

Für den Fall, dass die Bedingung einer Regel erfüllt ist, kann auf eine global definierte Named ACL verwiesen werden, die eine Reihe von Accessor-Privilegiendefinitionen beinhaltet. Ein einzelnes Privileg kann hierbei nicht nur erlaubt oder verweigert, sondern auch „nicht gesetzt“ werden (not set). Dies hat zur Folge, dass eine Aussage über ein gewisses Privileg zu einem späteren Zeitpunkt erfolgen kann, wodurch die Positionierung und Rangfolge von Privilegiendefinitionen im Regelbaum einen grossen Stellenwert erhält.



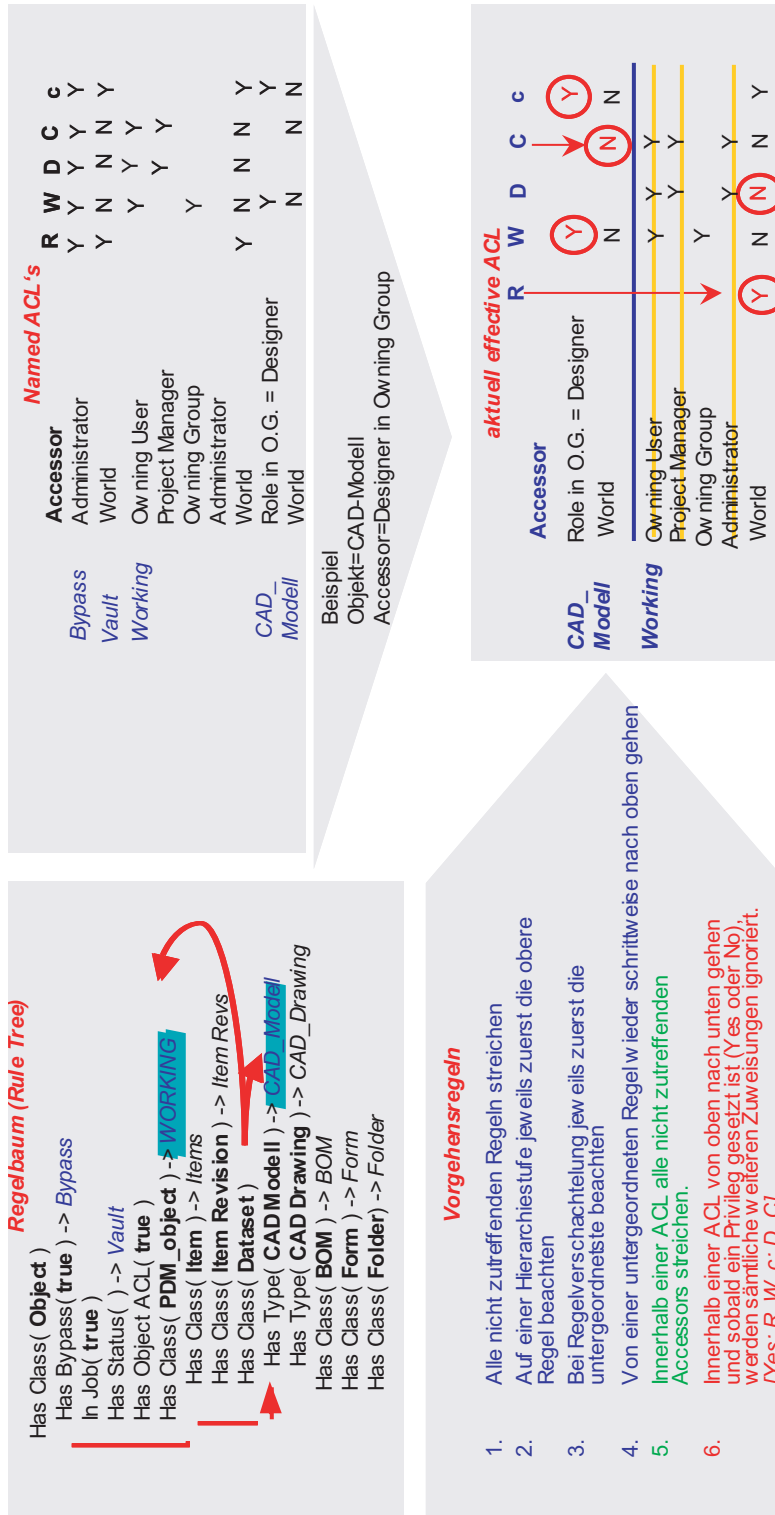


Bild (B032obmZ) Beispiel einer regelbasierten ACL-Autorisierung

Zur Veranschaulichung ist in Bild B032obmZ ein Beispiel zur Ermittlung einer regelbasierenden Autorisierung dargestellt. Als Objekt sei ein „CAD-Modell“ angenommen, welches sich noch in der Working-Phase befindet. Deshalb sind für den Accessor „Designer in Owning Group“ lediglich die ACLs Working und CAD\_Modell relevant. Gemäss den zugrundeliegenden Vorgehensregeln sind aus den Einträgen dieser Named ACLs die nicht zutreffenden Accessor-Fälle herauszustreichen, woraus sich die aktuell effektive ACL ergibt. Unter Beachtung der Präferenz innerhalb von ACLs können hieraus die gewährten Zugriffsprivilegien ermittelt werden (erlaubt: Read, Write, Copy).

Hieraus ergeben sich grosse Vorteile gegenüber sämtlichen herkömmlichen, objektbasierenden Autorisierungen, die sich wie folgt zusammenfassen lassen (vgl. Gegenüberstellung in Bild B033obmZ):

- grosse Flexibilität zur Änderung von Zugriffsprivilegien auch für den bereits existierenden Datenbestand,
- effiziente Methode zur Vereinheitlichung von Zugriffsdefinitionen gemäss einer vorgegebenen Unternehmensstrategie,
- Entlastung herkömmlicher Benutzer, da die Verantwortung für die Autorisierungsdefinition auf den Systemadministrator übergeht,
- bessere Laufzeit-Performance, da Privilegienänderungen fast ausschliesslich durch den Regelbaum implizit abgehandelt werden können (Regelbaum und globale ACLs sind Hauptspeicherresident) und hierfür nur noch in Ausnahmefällen Datenbankzugriffe nötig sind (bei zusätzlichen Objekt-ACLs).

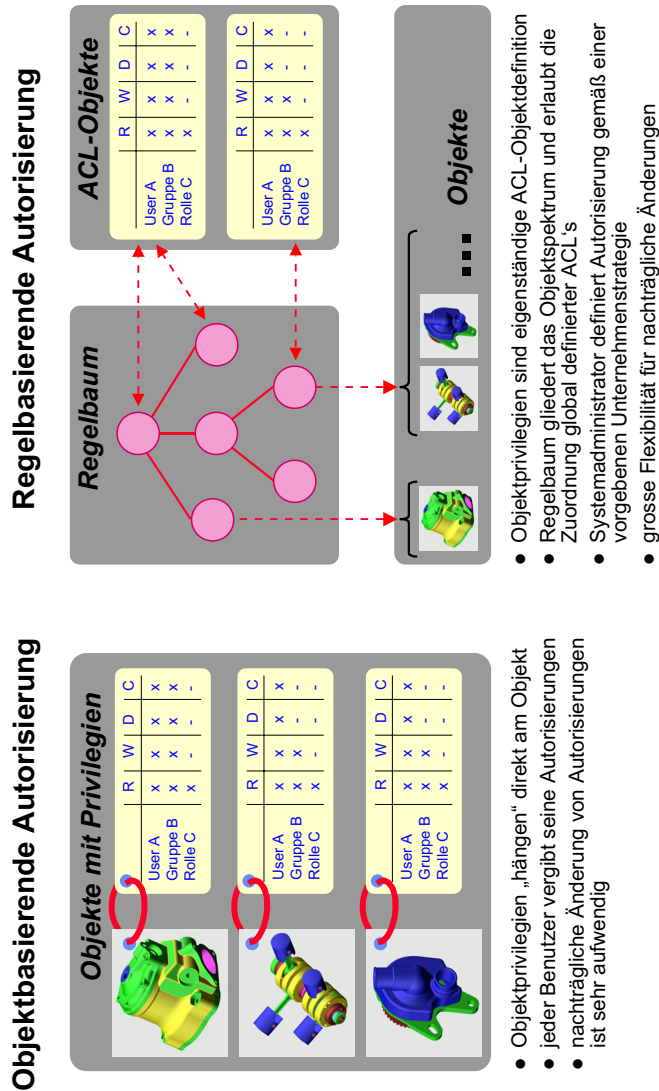


Bild (B033obmZ) Gegenüberstellung von objekt- und regelbasierender Autorisierung

Als nachteilig zu erwähnen ist die erheblich grössere Komplexität des regelbasierenden Autorisierungsmechanismus, die der Systemadministrator zu bewältigen hat. Zukünftig werden regelbasierende Autorisierungen aufgrund ihrer grossen Vorteile jedoch vermehrt Einzug in die PLM-Technologie halten, insbesondere bei den modernen objektorientierten Systemen. Anwendung findet diese Autorisierungsmethode beispielsweise im PLM-System „Teamcenter“.

## 7 Zusammenfassung

PLM-Objekte müssen eine ganze Reihe verschiedenartiger *Anwendungsobjekte* berücksichtigen. Dies sind u.a Bauteilstammdaten, Stücklisten, CAD-Modelle und viele mehr.

Ein *PLM-Objektmodell* ist eine allgemeine Basisobjektklasse, anhand derer die vorher genannten Objekte auf Systeminstanzen abgebildet werden können, die möglichst ähnliche Eigenschaften und Verhalten wie die realen Objekte besitzen. Diese Objektklassen sind vergleichbar mit einem Werkzeugkasten, woraus der Benutzer für ein jedes Anwendungsobjekt die bestgeeignetste Objektklasse auszuwählen hat.

Zwischen Instanzen einzelner Objektklassen können folgende Beziehungen auftreten:

- Zuordnung im Sinne von Objektverknüpfungen,
- Strukturierung gleichartiger Objekte im Sinne von Objekthierarchien.

Die *Konfiguration* dynamischer Produktstrukturen kann mittels verschiedener Mechanismen vollzogen werden. Bei der ereignisorientierten Sicht wird eine Revision nur dann konfiguriert, wenn bereits ein bestimmtes Ereignis eingetroffen ist. Die Produktstruktur kann auch zeitabhängig oder produktionsbezogen abgefragt werden, wenn der Zustand einer Produktstruktur zu einem bestimmten Zeitpunkt bzw. zu einem bestimmten Fertigungslos interessieren. Varianten lassen sich über variantenbezogene Sichten abfragen.

Die Bildung von *Varianten* ist ein Rationalisierungsmittel. Flexibilität für verschieden Kundenbedürfnisse wird vergrößert, Datenredundanz reduziert. Anstatt für jede mögliche Variante eine vollständige Produktstruktur anzulegen, wird durch eine gemeinsame Variantenstruktur (Variantenstückliste) die gesamte Vielfalt eines sogenannten Produkttyps zusammenfassend beschrieben.

Im Product Lifecycle Management liegt eine integrierte Datenhaltung vor. Der Zugriff ist grundsätzlich auf alle Daten ohne Kontrolle möglich. Ein Konzept zum Datenschutz und zur Datensicherung ist nötig. Autorisierungsmechanismen schaffen Abhilfe.

**Verständisfrage 1**

Welche Objektklassen muss ein PLM-System mindestens unterstützen und welche Eigenschaften besitzen diese?

**Verständisfrage 2**

Aus welchen Bestandteilen setzt sich die PLM-Datenbank zusammen?

**Verständisfrage 3**

Welche Abstraktionsebenen unterscheidet man im PLM?

**Verständisfrage 4**

Was versteht man unter Meta-Daten?

**Verständisfrage 5**

Welche Arten von Beziehungen unterscheidet man im PLM?

**Verständisfrage 6**

Wozu können unterschiedliche Beziehungsarten genutzt werden?

**Verständisfrage 7**

Was versteht man unter einer Revision?

**Verständisfrage 8**

Mit welchem Konzept kann man Datenredundanz vermeiden?

**Verständisfrage 9**

Wodurch unterscheiden sich Objekt- und Referenzebene?

**Verständisfrage 10**

Welches Gruppierungskonzept gibt es für Produktdaten?

**Verständisfrage 11**

Welche Systemphilosophien unterscheidet man im PLM?

**Verständisfrage 12**

Für welche Anwendungsobjekte werden üblicherweise Objektstrukturen benötigt?

**Verständisfrage 13**

Welche Arten von Objektstrukturen unterscheidet man?

**Verständisfrage 14**

Was ist das PLM-Äquivalent zu herkömmlichen Stücklisten und worin liegt der Unterschied?

**Verständisfrage 15**

Welche Konsistenzprobleme sind bei Produktstrukturen zu beachten?

**Verständisfrage 16**

Welche Darstellungsformen können eine Produktstruktur repräsentieren?

**Verständisfrage 17**

Wodurch unterscheiden sich Konstruktions- und Fertigungsstücklisten?

**Verständisfrage 18**

Wodurch unterscheiden sich dynamische und statische Produktstrukturen?

**Verständisfrage 19**

Welche generischen Konfigurationsmechanismen gibt es?

**Verständisfrage 20**

Was versteht man unter einer Datenbanksicht?

**Verständisfrage 21**

Was versteht man unter Effectivity und wozu braucht man diese?

**Verständisfrage 22**

Welche Effectivity-Konfigurationen unterscheidet man?

**Verständisfrage 23**

Bezüglich welchen PLM-Objektklassen kann man Effectivity einsetzen?

**Verständisfrage 24**

Wozu dienen Produktvarianten?

**Verständisfrage 25**

Welche grundlegenden Aspekte sind bei der Variantenbildung zu unterscheiden?

**Verständisfrage 26**

Welche anwendungsseitigen Variationskonzepte bestehen für Produktstrukturen?

**Verständisfrage 27**

Welches sind die grundsätzlichen Auswahlmöglichkeiten bei Variantenstrukturen?

**Verständisfrage 28**

Welche Benummerungsarten von Varianten sind zu unterscheiden?

**Verständisfrage 29**

Was versteht man unter Katalogvarianten?

**Verständisfrage 30**

Welche wesentlichen Schritte kennzeichnen den Prozess bis hin zur Variantenstückliste?

**Verständisfrage 31**

Mit welchen systemseitigen Variantenmethoden unterstützen PLM-Systeme die Variantendefinition?

**Verständisfrage 32**

Welche Modellbildungen muss man aufgrund der Forderung nach Flexibilität bei PLM-Applikationen unterscheiden?

**Verständisfrage 33**

Welche Datenarten sind bei der Stücklistenproblematik grundsätzlich zu unterscheiden?

**Verständisfrage 34**

Welches sind die wesentlichen Aspekte einer Autorisierung?

**Verständisfrage 35**

Welche Hauptphasen unterscheidet man im Lebenslauf bezüglich der Autorisierung?

**Verständisfrage 36**

Was versteht man unter einem Accessor?

**Verständisfrage 37**

Was versteht man unter Privilegien?

**Verständisfrage 38**

Was ist der Unterschied zwischen Gruppen und Rollen?

**Verständisfrage 39**

Welche wesentlichen Methoden der Autorisierung sind zu unterscheiden?

**Verständisfrage 40**

Was versteht man unter einer ACL?

**Verständisfrage 41**

Wozu dient ein Regelbaum im Zusammenhang einer Autorisierung?

**Verständisfrage 42**

Wodurch unterscheiden sich objektbasierende und regelbasierende Autorisierungen?



## Publikationsverzeichnis – Literatur

- [DIN199/1-5] *DIN 199 Teile 1 bis 5: Begriffe im Zeichnungs- und Stücklistenwesen*. Berlin: Beuth Verlag  
Teil 1: Zeichnungen, 1984  
Teil 2: Stücklisten, 1977  
Teil 3: Stücklisten-Verarbeitung,  
Begriffe in Schlüsselssystemen, 1978  
Teil 4: Änderungen, 1981  
Teil 5: Stücklisten-Verarbeitung,  
Stücklistenauflösung, 1981
- [DIN2331] *DIN 2331: Begriffssysteme und ihre Darstellung*. Berlin: Beuth Verlag 1980
- [EHS-91] *Eigner, M.; Hiller, C.; Schindewolf, S.; Schmich, M.: Engineering Database: strategische Komponente in CIM-Konzepten*. München: Hanser Verlag 1991
- [EKK-87] *Eversheim, W.; Konz, H.-J.; Kosmas, I.: Produktionsgerechte Produktstrukturen - Montageorientierte Gliederung komplexer Produkte nutzt der gesamten Auftragsabwicklung*. VDI-Z 129 (1987) Nr. 3, S. 38-42
- [GeW-70] *Gerlach, H.-H.; Wiendahl, H.-P.: Das Speichern und Verarbeiten von Variantenstücklisten mittels elektronischer Datenverarbeitungsanlagen*. wt-Z. industrielle Fertigung 60 (1970) Nr. 8, S. 461-464
- [Mon-97] *Montau, R.: Concepts of Variant Configuration*. Internal ER-Doc., Unigraphics Solutions, Zürich 1997
- [Mon-00] *Montau, R.: Regelbasierende Autorisierung: PDM-Technologie zur Durchsetzung von Unternehmensstrategien*. SMM Nr. 43-2000, S. 24-30
- [RBP-93] *Rumbaugh, J.; Blaha, M.; Premerlani, W.; Eddy, F.; Lorenzen, W.: Objektorientiertes Modellieren und Entwerfen*. München: Hanser Verlag 1993
- [VDI-72] *VDI-Fachgruppe Betriebstechnik (ADB): Elektronische Datenverarbeitung bei der Produktionsplanung und -steuerung III: Informations- und Stücklistenwesen*. VDI-Taschenbücher T 28, Düsseldorf: VDI-Verlag 1972
- [Wie-89] *Wiendahl, H.-P.: Betriebsorganisation für Ingenieure*. 3. Auflage, München: Hanser Verlag 1989

# Objektklassifikation

Autor: Dr. E. Zwicker



# 1 Überblick

Seit der Einführung moderner Informationstechnologien ist ein explosionsartig ansteigendes Datenvolumen zu verzeichnen. Um diese Datenflut handhaben zu können sind Möglichkeiten für einen effizienten Zugriff auf Informationen notwendig.

Die Objektklassifikation ist eine notwendige Voraussetzung für die Strukturierung vorhandener Informationen bzw. für den effizienten Informationszugriff. Sie erhält damit eine entscheidende Bedeutung hinsichtlich des betrieblichen Daten- bzw. Informationsmanagements. Fehlt dieser Zugriff oder wird dieser nur ungenügend unterstützt, werden oftmals Informationen bzw. Objekte neu erzeugt, obwohl diese im Unternehmen bereits vorhanden sind. Benötigt werden somit effiziente Zugriffssysteme um beispielsweise die Vorzüge einer *Wiederholteilverwendung* nutzen zu können.

Entsprechend den unterschiedlichen Anforderungen und Bedürfnissen in verschiedenen Funktionsbereichen eines Unternehmens, innerhalb welchen eine Objektklassifizierung zum Einsatz kommt, ergeben sich verschiedene *betrieblichen Anwendungsgebiete* der Objektklassifikation.

Die *Gruppentechnik* stellt verschiedene Verfahren zur Gruppenbildung und Objektbeschreibung zur Verfügung. Ziel ist es, in den verschiedenen Bereichen einer Produktentwicklung sowie den nachgelagerten Unternehmensprozessen (Einkauf/Logistik, Fertigung/Montage, Verkauf, Service etc.) Objekte definierter Ähnlichkeit systematisch handhaben zu können.

Um Elemente zu Gruppen zusammenfassen zu können, müssen zunächst die wesentlichen Merkmale erfasst und strukturiert, gespeichert und verwaltet, wiedergefunden und zueinander in Relation gebracht werden. Voraussetzung hierfür ist die Beschreibung aller zu erfassenden Objekte durch *Merkmale bzw. Eigenschaften*.

## 2 Nutzgrössen der Wiederholteilverwendung

Eine Informationserzeugung ist nicht nur mit einem einmaligen Erzeugungsaufwand, sondern auch mit einem Verwaltungsaufwand und damit Kosten verbunden, welche die Erzeugungskosten bei weitem übersteigen. Die benötigten Ressourcen zur Speicherung dieser Informationen verursachen in diesem Zusammenhang ebenfalls einen vergleichsweise geringen Aufwand.

Die grössten Folgekosten verursachen Neukonstruktionen. In der Literatur wird der Verwaltungsaufwand für die Einführung eines Neuteils mit ca. 1000.- sFr., die spätere Pflege mit ca. 200.- bis 600.- sFr. im Jahr angegeben. Neben diesem reinen Verwaltungsaufwand fallen zudem weitere Aufwendungen innerhalb der nachgelagerten Funktions- und Organisationseinheiten eines Unternehmens an (Vorrichtungsbau, Fertigungs- und Prüfmittel, Ersatzteilwesen etc.). Der Objektklassifikation und der Ähnlichkeitsuche kommt infolgedessen eine wichtige Bedeutung zu.

Als wesentliche Nutzgrössen einer durch eine Objektklassifikation ermöglichte Wiederholteilverwendung sind die folgenden zu nennen [Bra-86]:

- Verringerung des Verwaltungsaufwands,
- Verkürzung der Durchlaufzeiten sowie der Liefer- bzw. Anlauftermine,
- effektivere Kosten- und Terminplanung,
- Verwertung von vorhandenen Know-how,
- Reduzierung des Fehler- und Qualitätsrisikos,
- günstigere Einkaufskonditionen,
- Senkung der Stückkosten (Eigenfertigung),
- Senkung des Investitionsumfangs für Fertigungs- und Prüfmittel,
- Reduzierung von Umlaufkapital und des Verschrottungsrisikos,
- Erleichterung beim Kundendienstwesen.

Jede unnötige Neukonstruktion ist demzufolge als Fehlinvestition zu betrachten, die in der Summe zu erheblichen Gewinneinbussen eines Unternehmens führt!

Benötigt werden somit Methoden und Werkzeuge, die einen effizienten und eindeutigen Zugriff auf vorhandene Objektinformationen ermöglichen.

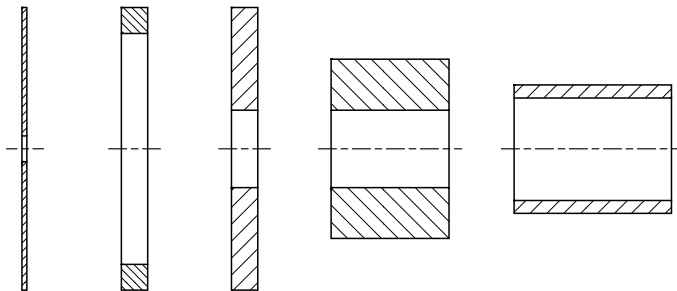
Aufgrund der Vielfältigkeit von Benennungen, deren verschiedener Interpretationsmöglichkeiten sowie unterschiedlicher Anwendungen,

sind diese als alleiniger Zugriffsmechanismus nur sehr bedingt geeignet.

Aus Bild B001oklZ wird diese Problematik ersichtlich. Geometrisch betrachtet unterscheiden sich die Teile nur durch die Durchmesser/Längenverhältnisse  $L/D$  und  $L/d$ , bzw. dem Durchmesser Verhältnis  $D/d$ . Für bestimmte  $L/D$ ,  $L/d$ ,  $D/d$ - Bereiche sind jedoch unterschiedliche Benennungen im Gebrauch. Oftmals werden somit geometrisch sehr ähnliche Teile unterschiedlich bezeichnet.

Gestalt: Hohlzylinder

Parameter: Aussendurchmesser  $D$ , Innendurchmesser  $d$ , Länge  $L$



Blende

Ring

Scheibe

Buchse

Hülse

Bild (B001oklZ) Uneindeutigkeit von Benennungen

Benötigt werden somit Klassifikationssysteme,

- die sich durch eindeutige Zugriffsmechanismen auszeichnen,
- die eine Definition anwendungsspezifischer Zugriffsmechanismen erlauben (ein Konstrukteur wird beispielsweise ein Bauteil auf andere Weise suchen wollen als ein Mitarbeiter aus der Arbeitsvorbereitung),
- welche flexibel erweiterbar sind,
- welche die Klassierung bzw. den Zugriff auf das gesamte Objektspektrum erlauben,
- und in PDM-Systeme integriert werden können.

Methodische Grundlagen und Konzepte für den Aufbau von Klassifikationssysteme bietet die Gruppentechnik, welche im folgenden detailliert beschrieben werden.

### 3 Betriebliche Anwendungsgebiete der Objektklassifikation

Unter dem Begriff *Klassifikation* versteht man innerhalb der Produktentwicklung das Zusammenfassen von Objekten anhand konstruktiver, funktioneller oder fertigungstechnischer Eigenschaften zu Objektgruppen definierter Ähnlichkeit. Objekte solcher Gruppen bilden vereinheitlichte Gruppen, die gemeinsam (weiter-) behandelt werden können.

Hinsichtlich der Art, des Umfangs sowie der Behandlung der zu klassifizierenden Objekte bzw. Objektgruppen werden grundsätzlich zwei Methoden zur Objektklassifikation unterschieden:

- *Klassifikation auf Basis einer bekannten Ähnlichkeit (dynamische Klassifikation):*

Die Zielsetzung besteht darin, eine definierte Menge technischer Objekte in Gruppen zu strukturieren, um diese Objektgruppen innerhalb einer definierten Phase des Produktlebenszyklus gemeinsam behandeln zu können. Beispiel hierfür ist die Bildung von Teilefamilien innerhalb der Prozessplanung flexibler Fertigungssysteme (FFS). Charakteristisch ist der temporäre Charakter dieser Klassifikation im Produktlebenslauf. Entsprechend der jeweiligen Anwendung (zum Beispiel zur Variantenbildung innerhalb der Konstruktion, Investitionsplanung von Produktionsanlagen, Teilefamilienfertigung in der Fertigung etc.) wird für eine Teilmenge (z.B. alle Fertigungsteile) des gesamten Objektspektrum auf Basis anwendungsspezifischer Merkmale (z.B. die Länge und der Durchmesser von rotationsymmetrischen Fertigungsteile) eine Objektklassifikation definierter Strukturierung und Granularität aufgebaut.

Die dynamische Klassifikation weist folgende spezifische Charakteristika auf:

- Die einzelnen Objekte werden basierend auf speziell definierten Eigenschaften zu Gruppen zusammengefasst.
- Entsprechend der geforderten Granularität wird basierend auf diesen Gruppen eine temporäre Klassifikation erstellt.
- Gemäss einer sich ändernden Problemstellungen werden die dafür geeigneten Eigenschaften definiert.
- Aufgrund der sich wiederholenden Klassifikation ist eine möglichst weitgehende Automatisierung der Klassifizierung notwendig.

- Es wird nicht der gesamte Teilstamm eines Unternehmens, sondern lediglich Teilmengen klassifiziert.
- *Klassifikation auf Basis einer erwarteten Ähnlichkeit (statische Klassifikation):*

Zielsetzung ist die langfristige Klassifikation des gesamten Objektspektrums. Dies geschieht im Hinblick auf einen eindeutigen und effizienten, möglichst verwendungsunabhängigen Zugriff auf Objekte bzw. Objektinformationen. Um auf Objekte bzw. Objektgruppen zugreifen zu können, muss die Zielvorstellung hinsichtlich eines benötigten Objektes auf eine erwartete Ähnlichkeit mit den klassierten Objekten hin überprüft werden. Man spricht in diesem Fall von einer erwarteten Ähnlichkeit. Statische Klassifikationssysteme bilden die notwendige Basis von Ähnlichkeitssuchsystemen.

Statische Klassifikationssysteme werden im Gegensatz zur dynamischen Klassifikation einmal unter Berücksichtigung der jeweiligen Gesamtzielsetzung definiert und in der Folge beibehalten.

Die statische Klassifikation weist folgende spezifische Charakteristika auf:

- Die Klassifikation zielt auf die Suche nach vorhandenen Objekten ab.
- Statische Klassifikationen werden einmalig erstellt und in der Folge sukzessive ausgebaut bzw. angepasst.
- Das gesamte Objektspektrum wird klassifiziert.



## 4 Begriffe, Merkmale und Eigenschaften

Von zentraler Bedeutung innerhalb der Gruppentechnik sind Merkmale, deren Ausprägungen (auch als Merkmalsausprägungen bezeichnet) sowie Eigenschaften. Mittels den letzteren werden Objekte beschrieben und basierend auf dieser Beschreibung zu Objektgruppen definierter Ähnlichkeit zusammengefasst bzw. gegliedert. Objektgruppen können als abstrakte Objekte (z.B. doppelwirkende Pneumatikzylinder) aufgefasst und in der selben Weise wie konkrete Objekte (z. B. der doppelwirkende Pneumatikzylinder DSNP-60.80 der Fa. FESTO) behandelt werden.

*Merkmale* dienen zum Beschreiben und Unterscheiden von Objekten [DIN4000-1]. Es handelt sich dabei um nicht veränderbare Kennzeichen einer Menge von Objekten, die zu deren Charakterisierung verschiedene Ausprägungen (Werte) annehmen.

Die Charakteristika der Beschreibung von Objekten oder Objektgruppen wird demzufolge durch das der Beschreibung zugrundeliegende Merkmal definiert. Die sorgfältige und eindeutige Definition der zu verwendenden Merkmale beeinflusst die Qualität der daraus resultierenden Objektklassifikation massgeblich.

Innerhalb der Merkmalsdefinition umfasst folgende Informationen:

- die eindeutige Definition hinsichtlich der vorgesehenen Anwendung (als Semantik eines Merkmals bezeichnet),
- die jeweilige Skalierungsmethode,
- die Ausprägungsdomäne sowie gegebenenfalls
- die Masseinheit.

Die *Semantik* eines Merkmals beinhaltet im wesentlichen die exakte Definition wie dieses Merkmal anzuwenden ist. Beispielsweise kann ein doppelwirkender Pneumatikzylinder durch das Merkmal „Hub H“ beschrieben werden. Die Definition dieses Merkmal muss beinhalten, was unter dem Hub H des zu beschreibenden abstrakten Gegenstandes zu verstehen ist - in diesem Fall die Strecke zwischen der eingefahrenen und ausgefahrenen Position der Kolbenstange des Pneumatikzylinders.

Oftmals wird die Semantik eines Merkmals in einer Skizze verdeutlicht. In diesen werden die verschiedenen Merkmale anhand eines Komplexteiles verdeutlicht.

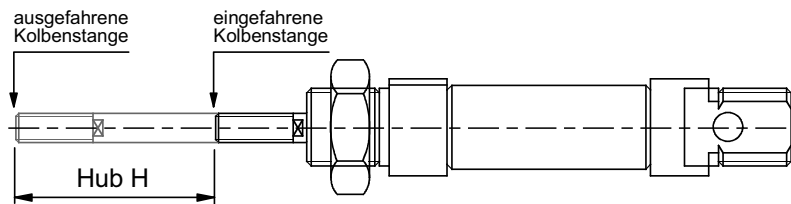


Bild (B002okIz) Das Merkmal „Hub“ eines Pneumatikzylinders

Hinsichtlich der Semantik von Merkmalen wird weiterhin zwischen expliziten und impliziten Merkmalen unterschieden.

- *Implizite Merkmale* sind nicht quantifizierbar und weisen als Ausprägung Begriffe oder Bezeichnungen auf. In diesem Fall sind die Ausprägungen oftmals interpretierbar. Eine eindeutige Aussage ist wiederum erst in Verbindung mit dem zugehörigen Merkmal zu treffen. Beispielsweise könnte ein Objekt bezüglich dem Merkmal „Fertigungsverfahren“ die Ausprägung „Drehteil“ aufweisen. Alleinstehend „impliziert“ diese Ausprägung, dass es sich um ein rotationssymmetrisches Objekt handelt. Die eindeutige Semantik ist erst mit der Eigenschaft „Fertigungsverfahren Drehteil“ gegeben. Die entsprechende Ausprägung kann entweder verschlüsselt (in diesem Fall handelt es sich um eine nominale Skalierung) oder einem Objekt direkt als Ausprägung bezüglich eines Merkmals zugewiesen werden.
- *Explizite Merkmale* können quantifiziert werden. In Abhängigkeit der Skalierungsmethode wird zwischen nominalen, ordinalen sowie kardinalen Merkmalen unterschieden. Explizite Merkmale sind nicht interpretierbar.

Die *Skalierung* des Merkmals bestimmt den Informationsgehalt der auf diesem Merkmal basierenden Eigenschaft massgeblich (vgl. Bild B003okIz).

Die Ausprägungen *nominaler* Merkmale beschreiben lediglich verschiedene Zustände des Objekts durch kategoriale Urteile, ohne eine Rangfolge zu enthalten. Eine Sonderform stellen die binären Merkmale dar, die lediglich zwei Ausprägungen annehmen können. Nominal skalierte Merkmale bzw. Eigenschaften weisen den geringsten Informationsinhalt auf. Wird für ein explizites Merkmal nominal skaliert kommt es aufgrund der hierfür notwendigen Bereichsbildung zu einem Informationsverlust.

Eigenschaft		Aussage	Ausprägungsdomäne	Beispiel
Skalierungsart				
Kardinal	Metrisch	Quantifizierung fest definierter Nullpunkt	0 1 2 3 4 5 →	Werkstücklänge [mm]
	Intervall	ohne natürlichen Nullpunkt	← -2 -1 0 +1 +2 →	Temperatur [°C]
		fest definierter Null- und Endpunkt	0 2 4 6 8 10 12	Windstärke [Bft]
Ordinal		Reihenfolge	$0 < 1 < 2 \dots$	0 : max. 2 Bohrungen 1 : 3 - 5 Bohrungen 2 : min. 6 Bohrungen
Nominal		Identifikation	$0, 1, 2, \dots, n$	0 : Rotationsteil 1 : prismatisches Teil 2 : Blechteil
Binär			$0, 1$	0 : Nut vorhanden 1 : keine Nut vorhanden

Informationsgehalt

Bild (B003okIz) Skalierungsarten [HiT-71]

Wird beispielsweise die Länge eines Werkstückes mittels eines nominalen Merkmals beschrieben (vgl. Bild B004okIz) kann im folgenden lediglich die Aussage getroffen werden, dass das beschriebene Werkstück eine Länge im Bereich 20 - 29 mm ausweist.

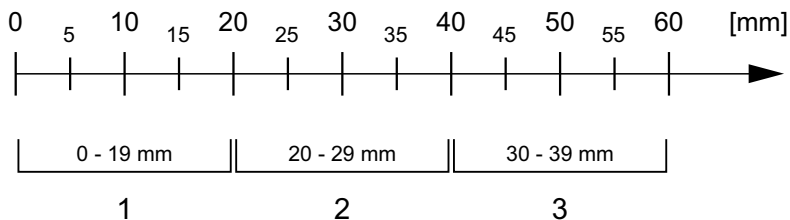


Bild (B004okIz) Informationsverlust einer nominalen Skalierung

*Ordinale* Merkmale erlauben im Gegensatz dazu, eine Ordnungsrelation abzubilden, d.h. es kann zusätzlich eine Rangfolge der Merkmalsausprägungen angegeben werden.

*Kardinale* Merkmale können beliebige Werte einer Skala mit definierter Masseinheit annehmen. Zu unterscheiden ist hierbei zwischen Skalen mit fest definierten Nullpunkt (metrische Skalierung), solchen die ein beliebiges Intervall ohne definierten Nullpunkt abbilden und Skalen mit fest definierten Null- und Endpunkt (Intervall Skalierung).

Die *Ausprägungsdomäne* eines Merkmals definiert den zugrundeliegenden Datentyp (z.B. Ganzzahl, Kommazahl etc.) sowie den mög-

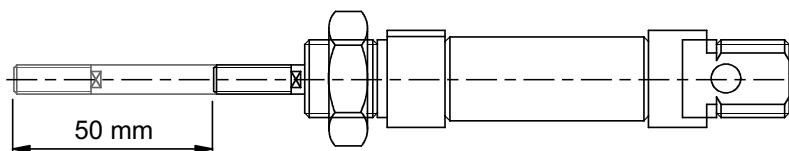
lichen Wertebereich (z.B. Anzahl der Stellen) bzw. vorgesehene oder natürlich-physikalische Grenzen (z.B. den absoluten Nullpunkt bei der Temperatur). Die Wahl bzw. Definition der Domäne eines Merkmals ist von der gewählten Skalierungsart abhängig.

Die *Masseinheit* legt fest, ob und falls vorhanden welche Masseinheit zur Anwendung kommen soll.

Beispielsweise würde im Falle des rotationssymmetrischen Zylinders für das Merkmal „Länge des Zylinders“ eine metrische Skalierung, als Domäne eine 4-stellige Kommazahl mit 2 Stellen hinter dem Komma sowie als Masseinheit die SI-Einheit [mm] definiert.

Über die jeweilige *Ausprägung* (z.B. „50“ im Falle des obigen Beispiels) können konkrete Gegenstände hinsichtlich des zugrunde liegenden Merkmals beschrieben und untereinander unterschieden werden. Eine Ausprägung für sich allein (z.B. 50) hat keine Aussagekraft, da Ausprägungen von dem zugehörigen Merkmal semantisch abhängig sind.

Die *Eigenschaft* eines Objektes oder einer Objektgruppe ist durch die Verknüpfung von Merkmal und Ausprägung definiert. Eine Eigenschaft des Zylinders ist beispielsweise „Hub des Pneumatikzylinders  $H = 50 \text{ mm}$ “.



$$\text{Merkmal: Hub [mm] / Ausprägung: 50}$$

$$\text{Eigenschaft = Merkmal + Ausprägung:}$$

$$\text{Hub = 50 mm}$$

Bild (B005okI2) Zusammenhang Merkmal - Ausprägung - Eigenschaft

Durch *Begriffe* werden Mengen individueller Gegenstände zu einem gedachten, allgemeinen Gegenstand zusammengefasst und durch diesen gedanklich und sprachlich verfügbar gemacht [DIN2330]. Unter dem Begriff „Pneumatikzylinder“ werden beispielsweise alle Arten von Pneumatikzylinder unabhängig von den verschiedenen möglichen Ausführungsvarianten (einfachwirkende Zylinder, doppeltwirkende Zylinder, Kompaktzylinder etc.) zusammengefasst und

gedanklich greifbar. Mittels Merkmale können Begriffen weitere Informationen zugeordnet werden, die zur Unterscheidung bzw. Einschränkung dienen [DIN2330]. Der Begriff „Pneumatikzylinder Hub 12mm / Kolben  $\varnothing$  50mm“ beschränkt sich beispielsweise auf Zylinder mit einem definierten Hub (50mm) und Kolben  $\varnothing$  (12mm).

## 5 Gliederung und Beschreibung

Merkmale werden sowohl für die Gliederung als auch für die Beschreibung von Objekten verwendet:

Die *Gliederung* einer Menge von Objekten zielt auf das Zusammenfassen von Objekten zu Gruppen einer definierten Ähnlichkeit. Diese können auf unterschiedlichen Abstraktionsebenen definiert werden. Durch *Unterscheidungsmerkmale* sowie deren möglichen Ausprägungen werden die Gruppen untereinander abgegrenzt. Hierfür ist die Reduktion der Merkmalsausprägungen auf den für eine Gruppe definierten Ausprägungsbereich (vgl. Bild B006okIZ) erforderlich. Die einer Gruppe zugewiesenen Objekte werden untereinander nicht unterschieden und weisen alle die der Gruppe zugewiesenen Eigenschaften auf. Vorteilhaft sind die verschiedenen Abstraktionsebenen einer Gliederung, die sich vortrefflich für den Zugriff auf einzelne Gruppen eignen. Nachteilig ist der sich aufgrund der Reduktion exakter Merkmalsausprägungen ergebende Informationsverlust (vgl. Bild B006okIZ).

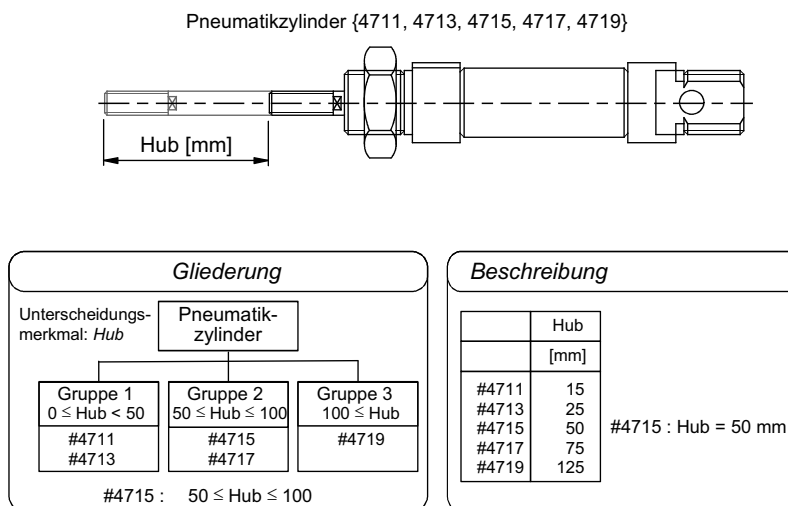


Bild (B006okIZ) Gliederung und Beschreibung von Pneumatikzylinder

Bei der *Beschreibung* werden einzelne Objekte basierend auf einer Anzahl von Merkmalen durch die jeweils zutreffenden Ausprägungen ohne Informationsverlust beschrieben. Aufgrund den nicht vorhandenen Abstraktionsebenen muss im Fall eines Zugriffs auf benötigte Objekte die gesamte Objektmenge durchsucht werden.

In der Regel werden diese beiden Konzepte miteinander kombiniert, wodurch die Vorteile einer Gliederung (Zugriff auf verschiedenen Abstraktionsebenen) mit denen der Beschreibung (kein Informationsverlust) miteinander vereinigt werden.

## 6 Methoden der Gruppentechnik

Der Gruppentechnik stehen verschiedene Verfahren für die Gruppenbildung zur Verfügung (vgl. Tabelle T001okIZ), die auf unterschiedliche Weise genutzt und auch miteinander kombiniert werden können.

Ausgangspunkt sind die Klassifikationsschlüssel welche in den 60er und 70er Jahren in grosser Zahl im Hinblick auf die Bildung von Teilefamilien zur Unterstützung der Fertigungsplanung entwickelt wurden und mit dem Aufkommen leistungsfähiger Rechnersysteme an Bedeutung verloren haben. Sachmerkmale spielen im Zusammenhang von systemunabhängigen Normteillbibliotheken bzw. Produktkatalogen eine wichtige Rolle und stehen als kommerziell erhältliche Produkte zur Verfügung.

Begriffssysteme stellen unabhängig von der beabsichtigten Anwendung eine Systematik zur Gliederung von Objekten zur Verfügung. Dies basiert darauf, dass Begriffe untereinander in Beziehung gebracht werden. Innerhalb der DIN 2331 sind die verschiedenen Möglichkeiten diese sogenannten Begriffsbeziehungen zu definieren beschrieben.

Generische Objektsysteme bauen auf den Paradigma der Objektorientierung auf. Sie entstammen der objektorientierten Analyse und finden mit der wachsenden Verbreitung von CAx-Systemen welche auf objektorientierten Konzepte aufbauen zunehmend Anwendung.

Die Clusteranalyse bzw. deren Optimierung auf Basis von Fuzzy-Sets entstammt der Fertigungsplanung und eignet sich besonders für generische Klassifikationen.

Fourierdeskriptoren sind für die Beschreibung geometrischer Informationen besonders geeignet. Sie sind lediglich Gegenstand der Forschung, stellen aber einen interessanten Aspekt dar, da das gesuchte Teil durch eine Handskizze beschrieben werden kann.

Die Wahl des geeigneten Verfahrens wird massgeblich durch den jeweiligen Anwendungsbereich beeinflusst.



Verfahren	Methode	Klassifikation	Hauptanwendungsbereich
Ordnungssysteme	Begriffssysteme (BS)	statisch	Gliederung von Objekten wie z.B. Folder innerhalb von PDM-Systemen
	Klassifikationsschlüssel(KS)	statisch	Fertigungsplanung, Teilefamilienbildung, Nummerungssysteme
	Sachmerkmalleisten (SML)	statisch	Normteil-/Wiederholteilbibliotheken, Teileverwaltung in zahlreichen CAx- PDM- und PPS-Systemen
	Verbindung BS/SML	statisch	Normteil-/Wiederholteilbibliotheken Teileverwaltung in zahlreichen CAx-, PDM- und PPS-Systemen
	Merkmalbasierte Gliederung und Beschreibung	statisch	Teilebibliotheken (ISO 13584)
	generische Objektsysteme	statisch / (dynamisch)	CAD-Normteilbibliotheken
multivariate Statistik	Clusteranalyse	dynamisch	Fertigungsplanung, Teilefamilienbildung
	Faktoren-Analyse	dynamisch	Fertigungsplanung, Teilefamilienbildung
	Fuzzy-Sets	dynamisch	Fertigungsplanung, Teilefamilienbildung
mathematische Analytik	Fourier-Deskriptoren	dynamisch	Ähnlichteilssuche

Tabelle (T001okIz) Verfahren und Methoden der Gruppentechnik

## 6.1 Begriffssysteme

Begriffssysteme dienen der Gliederung von Objektmengen. Begriffssysteme sind anwendungsunabhängig und werden oft mit anderen Methoden der Gruppentechnik vor allem beim Aufbau von statischen Klassifikationssystemen verwendet. Generische Begriffssysteme finden teilweise in Klassifikationsschlüsseln sowie bei generischen Objektsystemen eine direkte Verwendung.

### 6.1.1 Funktionsprinzipien von Begriffssysteme

Entscheidend für den Informationsgehalt solcher Begriffssysteme ist die Art der Beziehung zwischen den einzelnen Begriffen. Nach DIN 2331 und DIN 2330 unterscheidet man hierbei 3 Arten:

- hierarchische Beziehungen  
Es liegt ein Über- und Unterordnungsverhältnis vor. Eine hierarchische Beziehung besteht immer zwischen einem Oberbegriff und mehreren Unterbegriffen.  
Der Oberbegriff wird durch Eigenschaften charakterisiert, die auch für alle Unterbegriffe zutreffen.  
Auf semantischer Ebene unterscheidet man dabei zweierlei verschiedene Arten hierarchischer Beziehungen:
  - Abstraktionsbeziehungen (generische Beziehung)  
Der Begriffsinhalt des engeren Begriffs (Unterbegriff) schliesst den Begriffsinhalt des weiteren Begriffs (Oberbegriff) ein, wobei sich der Unterbegriff in mindestens einer zusätzlichen Eigenschaft vom Oberbegriff unterscheidet (vgl. Bild B007okI2). Generische Begriffssysteme liegen somit vor, wenn die Beziehung zwischen einem abstrakten und einem konkreten Begriff durch einen Konkretisierungsschritt bzw. einem Abstraktionsschritt gekennzeichnet ist. Mittels Abstraktionsbeziehungen können Objektklassifikationen aufgebaut werden.
  - Bestandsbeziehungen (partitive Beziehung)  
Bei der Bestandsbeziehung steht der einem Ganzen entsprechende übergeordnete Begriff mit den untergeordneten Begriffen in einer Beziehung, die sich durch gedankliches Zerlegen in die Bestandteile des Ganzen ergibt.  
Ein partitives Begriffssystem stellt damit immer ein Ganzes und die Teile, aus denen dieses Ganze besteht, dar. Eine

Strukturstückliste stellt z.B. eine hierarchisch partitive Begriffsbeziehung dar.

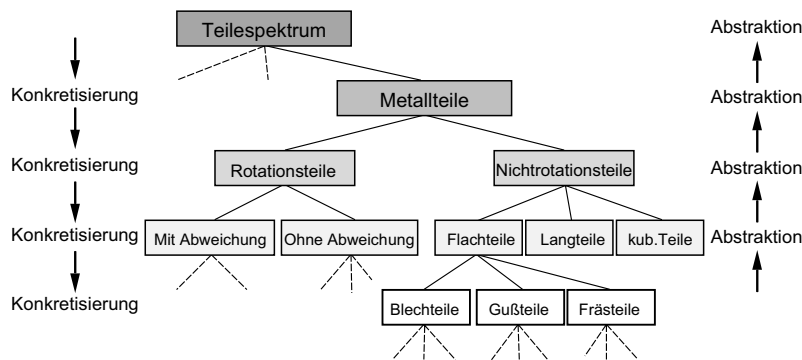


Bild (B007oklZ) Hierarchisch-generische Begriffsbeziehung Objektklassifikation

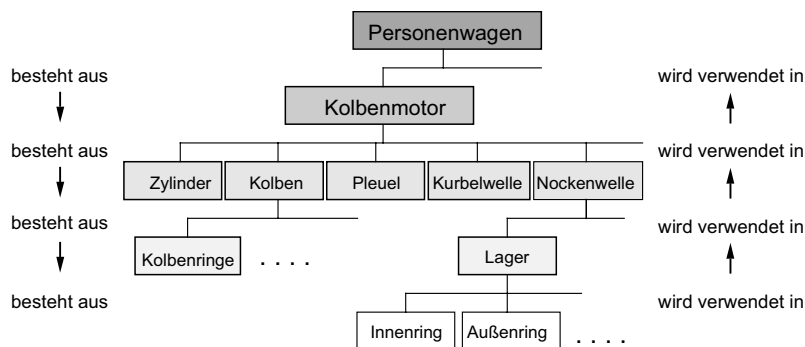


Bild (B008oklZ) hierarchisch-partitive Begriffsbeziehungen: Stückliste

Auf syntaktischer Ebene unterscheidet man zwischen:

- monohierarchischen Begriffsbeziehungen:  
Diese sind dadurch charakterisiert, dass ein Unterbegriff genau einem Oberbegriff zugeordnet sein muss.
- polyhierarchischen Begriffsbeziehungen:  
Im Gegensatz zu den monohierarchischen Begriffsbeziehungen kann ein Unterbegriff mittels einer polyhierarchischen Beziehung einem oder mehreren Oberbegriffen zugeordnet werden. Das daraus resultierende polyhierarchische Begriffssystem stellt ein gerichtetes semantisches Netz dar. Liegt zwischen den einzelnen Begriffen kein Über-/Unterordnungsverhältnis vor, handelt es sich um ein semantisches Netz.

- Ahierarchische Begriffsbeziehungen  
Ahierarchische Begriffsbeziehungen stellen einen Sonderfall dar. Hier liegt lediglich eine unstrukturierte, d.h. voneinander unabhängige Zusammenstellung von Informationen vor, die z.B. alphabetisch nach den beschreibenden Begriffen in Tabellenform geordnet sind.  
Beispiel für ein ahierarchisches Ordnungssystem ist ein Telefonbuch, welches eine alphabetisch geordnete Liste von Namen, Vornamen und den zugehörigen Telefonnummern enthält.
- Teilhierarchische Begriffsbeziehungen  
Sind in einem Begriffssystem sowohl hierarchische als auch ahierarchische Beziehungen enthalten, spricht man von teilhierarchischen Begriffsbeziehungen. Ein Thesaurus stellt zum Beispiel ein teilhierarchisches Ordnungssystem dar. Unter einem Thesaurus versteht man eine alphabetische Sammlung von Begriffen (z.B. eines Fachgebietes). Dieser Begriff wird in einem hierarchischen Ordnungssystem dann weiter aufgeschlüsselt.

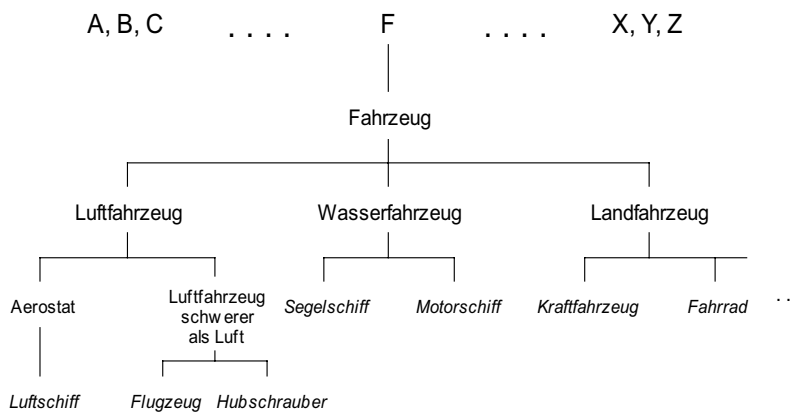


Bild (B009okIz) Beispiel eines Thesaurus [DIN 1463]

Hinsichtlich dieser Begriffssysteme existieren eine Vielzahl sogenannter Klassifizierungs- oder Ordnungsschlüssel, die in der industriellen Praxis mehr oder weniger Verbreitung gefunden haben.

### 6.1.2 Datenmodelltechnische Umsetzung

Hinsichtlich der Abbildung auf ein semantisches Datenmodell bestehen zwei verschiedene Möglichkeiten:

- explizite Modellierung der Ober- bzw. Untergruppen:  
Bei der expliziten Modellierung eines Begriffssystems wird jeder Begriff auf einen Objekttyp abgebildet. Die Abbildung des Über-/Unterordnungsverhältnisses erfolgt durch die Spezialisierung des jeweiligen Oberbegriffs durch den zugeordneten Unterbegriff.

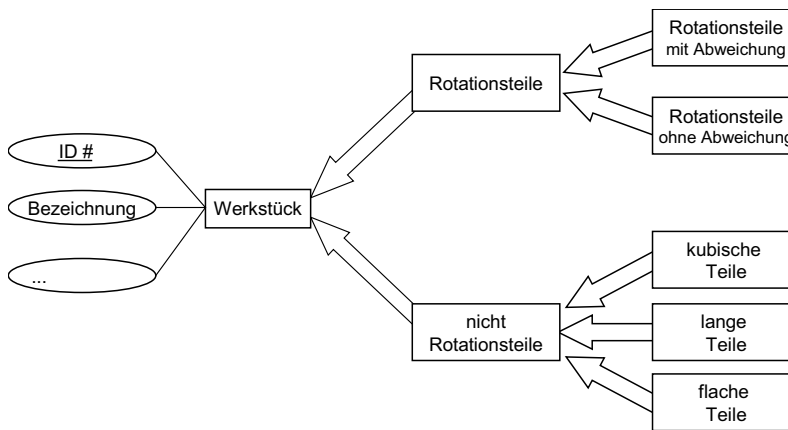


Bild (B010okIz) Explizites Modell eines monohierarch generischen Begriffssystems

Der Vorteil einer expliziten Modellierung liegt darin, dass die gesamte Semantik des Begriffssystems auf das Modell abgebildet wird und diese folglich durch modellinhärente Integritätsbedingungen sichergestellt werden kann. Als Voraussetzung müssen sowohl die Syntax (monohierarchisch, polyhierarchisch etc.) als auch die Semantik beim Modellentwurf definiert sein und können zu einem späteren Zeitpunkt nur in einem begrenzten Masse geändert werden.

- implizite Modellierung auf Basis eines Metaschemas:  
Innerhalb des Metaschemas wird lediglich die Syntax des Begriffssystems auf das Datenmodell abgebildet während die Semantik des Begriffssystems auf Instanzebene definiert wird. Entsprechend können keine modellinheränten Integritätsbedingungen zur Sicherstellung der semantischen Integrität genutzt werden. Der Vorteil liegt in der grossen Flexibilität, weshalb diese Art der Modellierung in der Regel in PDM-Systemen

Anwendung findet. Aufgrund der geforderten hohen Flexibilität von Klassifikationssystemen innerhalb von PDM-Systemen wird ausserdem ein Teil der Strukturdefinition (vornehmlich die Festlegung auf eine mono- oder polyhierarchische Struktur) des Begriffsystems ebenfalls auf Instanzebene definiert (z.B. durch eine Defaulteinstellung). D.h. das Metaschema erlaubt die Abbildung von poly- und monohierarchischen Begriffssystemen.

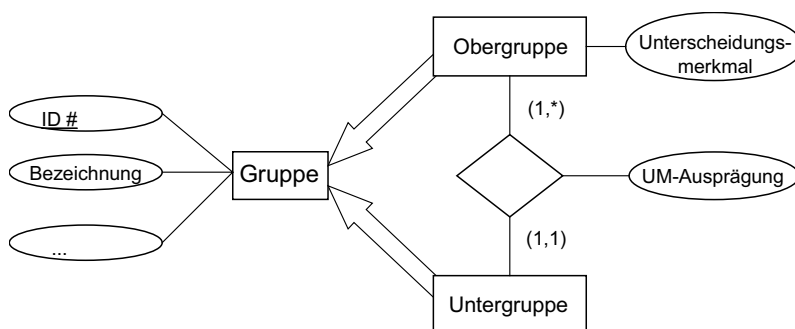


Bild (B0110klZ) Metamodell eines Begriffsystems

Die explizite Modellierung kann mit der Metamodellierung kombiniert werden. Der sich nicht ändernde Teil eines Begriffsystems (z.B. die Unterscheidung zwischen Zukauf- und Eigenfertigungsteilen) erfolgt mittels einer expliziten Modellierung auf den oberen Stufen des Begriffstems. Der darunter liegende Bereich wird auf ein Metamodell abgebildet und kann im folgenden geändert und erweitert werden.

### 6.1.3 Praktische Anwendung von Begriffssystemen

Begriffssysteme kommen in vielfältiger Weise sowohl im alltäglichen Leben (z.B. Lexika, Schlagwortverzeichnissen etc.) als auch in der Produktentwicklung zur Anwendung. In den meisten Fällen werden diese entsprechend dem jeweiligen Anwendungsgebiete mit weiteren Methoden der Gruppentechnik kombiniert. Begriffssysteme sind nur sehr eingeschränkt zu automatisieren, weshalb sie vorwiegend bei statischen Klassifikationssystemen verwendet werden.

Innerhalb von PDM-Systemen werden Begriffssysteme genutzt um Ordnungsbäume aufzubauen. Wird zum Beispiel im PDM-System eine Folderstruktur aufgebaut, so handelt es sich dabei um ein hierarchisches Begriffssystem.

Häufigste Verwendung finden Begriffssysteme in Verbindung mit Sachmerkmaleisten, weshalb diese Anwendung in Abschnitt Verbindung BS/SML getrennt beschrieben wird.

## 6.2 Klassifikationsschlüssel

Klassifikationsschlüssel stellen den Ausgangspunkt der Gruppentechnik dar. Sie wurden bis zum Aufkommen leistungsfähiger Rechnersysteme bzw. PDM-Systemen in kaum überschaubarer Vielzahl definiert und innerhalb der Teilefamilienfertigung, Ähnlichteilsuche und Nummerungstechnik verwendet. Heute beschränkt sich ihre Anwendung vorwiegend auf die Nummerungstechnik.

### 6.2.1 Funktionsprinzipien von Klassifikationsschlüssel

Zielsetzung von Klassifikationsschlüsseln ist die Unterteilung einer Menge von Teilen in Klassen, die bezüglich des betrachteten Unterscheidungsmerkmals möglichst heterogen sind. Innerhalb einer Klasse sollten die Teile dagegen möglichst homogen sein. Werden die jeweiligen Klassen systematisch durchnummeriert, erhält man einen Klassifikationsschlüssel. Jedes Objekt erhält durch diese Verschlüsselung eine für seine Merkmalsausprägung charakteristische Nummer. Alle Objekte einer Gruppe besitzen somit die gleiche Klassifikationsnummer und stimmen dadurch auch in den Merkmalsausprägungen überein.

Die wichtigsten Ordnungsgesichtspunkte, welche auch in den gängigen Klassifikationsschlüsseln verwendet sind entsprechend der jeweiligen Zielsetzung oder Anforderungen:

- Form  
Die meisten der heute bekannten und verwendeten Ordnungsschlüssel gehen entweder von der Endform aus oder von einer Zwischenform als Ergebnis eines bestimmten Fertigungsverfahrens. Die Formbeschreibung wird in der Regel aufgeteilt in die das Werkstück charakterisierende Hauptform und zusätzliche Formdetails, den Nebenformelementen.
- Funktion  
Für Baugruppen ist eine Klassifikation nach der Funktion die sinnvolle Vorgehensweise. In den meisten Fällen wird die Funktion durch einen charakterisierenden Namen oder eine kurze Beschreibung ausgedrückt. Eine Verschlüsselung nach der Funktion setzt demnach eindeutige Funktionsbegriffe voraus.
- Fertigungsanforderungen  
Die Fertigungsanforderungen eines Objektes ergeben sich aus der gewünschten Endform, dem Werkstoff, der geforderten Genauigkeit und der Oberflächenbeschaffenheit.



Sie beziehen sich dabei auf alle Stufen des Fertigungsprozesses und den damit verbundenen speziellen Anforderungen, wie z.B. bezüglich der Werkzeuge oder der Aufspannungsart.

Bei der Mehrzahl der bekanntgewordenen Teile-Klassifikationsschlüssel wird der offensichtliche Zusammenhang zwischen der Form eines Teils und dem Fertigungsprozess benutzt. Zur Beschreibung der die Fertigungsanforderungen charakterisierenden technologischen Merkmale werden also die konstruktiven Merkmale der Form und umgekehrt herangezogen.

- Anwendungsbereich

Der überwiegende Teil der veröffentlichten Systeme wurde zur Ordnung eines, gegenüber der unbegrenzten Vielfalt sämtlicher konstruierter Objekte, begrenzten Teilespektrums entwickelt. Man beschränkte sich dabei entweder in der zur Herstellung angewandten Technologie, in der Teileauswahl (Anwendungsbereich) oder in beiderlei Hinsicht.

Es wird im allgemeinen zwischen verzweigten, hierarchischen und unverzweigten d.h. parallelen Systemen unterschieden.

Innerhalb den verzweigten Systeme ist die Bedeutung jeder Schlüsselstelle mit Ausnahme der ersten vom Inhalt der vorigen abhängig. Die hierarchische Ordnung des Teilstamms ist durch eine Baumstruktur darstellbar. Jede Gabelung des Baumes entspricht einer Gruppe ähnlicher Objekte, wobei die tieferen Gruppen durch Hinzufügen einer weiteren Schlüsselstelle entstehen, also jeweils Teilegruppen der höheren Gruppe sind.

Unverzweigte, parallele Systeme sind übersichtlich und einfach in der Handhabung. Hier ist im Gegensatz zu den hierarchischen Systemen jeder Schlüsselstelle eine immer gleiche Bedeutung zugeordnet.

Die meisten Klassifikationsschlüssel weisen eine Mischform beider Systeme auf. Die ersten Schlüsselstellen haben dabei meist einen hierarchischen Aufbau. Den dadurch definierten Gruppen werden in den nachfolgenden Stellen allgemeingültige Merkmale, wie zum Beispiel der Werkstoff oder die Bearbeitungsqualität durch eine parallele Verschlüsselung zugeordnet (vgl. Opitz-Schlüssel).

In der Vergangenheit wurde eine grosse Anzahl an Ordnungsschlüssel zur Rationalisierung in der Konstruktion und Fertigung entwickelt. Tabelle T002okIZ gibt einen Überblick über die zur Zeit in der einschlägigen Fachliteratur beschriebenen Ordnungsschlüssel. Hervorgehoben sind dabei jene Schlüssel, welche weltweit eine besondere Bedeutung oder Verbreitung aufweisen.

Die meisten Systeme dienen zum formorientierten Ordnen von Einzelteilen. Baugruppen werden bisher im allgemeinen nicht durch Ordnungsschlüssel klassifiziert.

Die Mehrheit der Schlüssel sind für eine allgemeine Anwendung konzipiert, das heisst sie sind nicht auf ein spezielles Teilespektrum zugeschnitten. Die Anzahl der Stellen reicht dabei von 4 (VUOSO- und VUSTE-Schlüssel) bis hin zu über 20 Stellen (z.B. das ZAFO- und KK-3-System mit jeweils 21 Stellen). Durch eine hohe Anzahl an Stellen wird versucht, die zu ordnenden Teile möglichst exakt zu beschreiben, was jedoch einen aufwendigen und langwierigen Klassifizierungsaufwand mit sich bringt.

Ein Teil der Schlüssel ist speziell für bestimmte Fertigungsverfahren entwickelt worden (z.B. der Pacyna-Schlüssel zur Ordnung von gusstechnischen Teilen) oder bevorzugen bestimmte Teilegruppen (z.B. der VUSTE-Schlüssel für nicht rotationssymmetrische Bauteile). Viele dieser Klassifikationsschlüssel sind Ende 60er bzw. Anfang 70er Jahre für die Arbeitsvorbereitung zur Strukturierung des Teilestammes entstanden.

Name	Land	Art	Verwendung	Stellen	Struktur
Opitz	D	Form	Allg. (Span. Bearbeitung)	9	1-5 H
PGM	Schwede	Form	Allg. (Span. Bearbeitung)	6	1-6 H
DERO	D	Form	Allg. (Rotationsteile)	18	TH
IAMA	Jug.	Form	Allg. (Span. Bearbeitung)	8	1-2 H
Brisch	UK	Form	Allg.	var.	H
Parts Analog	USA	Form	Allg.	var.	H
CODE	USA	Form	Allg.	8	1-6 H
A. Chalmers	USA	Form	Allg.	8	H
KCI	Japan	Form	Allg. (Rotationsteile)	5	H
VUOSO	CZ	Form	Allg. (Span. Bearbeitung)	4	1-3 H
MICLASS	Niederl.	Form	Allg.	12	
DCLASS	USA	Form	Allg.	8	TH
VUSTE	CZ	Form	Allg. (Nichtrotationsteile)	4	1-2 H
Gildemeister	D	Form	Allg.	10	TH
ZAFO	D	Form	Allg.	21	1-5 H
Niitmash	UDSSR	Form	Allg.	15	TH
VPT I	UDSSR	Form	Allg.	var.	TH
HFU	China	Form/ Funkt.	Allg.	16	1-3 H
Pittler	D	Form/ Funkt.	Allg.	9	TH
KK-3	Japan	Form/ Funkt.	Allg.	21	TH
Stuttgart	D	Form/ Funkt.	Allg.	10	H
Toyoda	Japan	Form/ Funkt.	Allg.	10	TH
Spieß	D	Form/ Funkt.	Umformtechnik	3	H

Tabelle (T002okIZ) die weltweit bekanntesten Klassifikationsschlüssel  
(die bedeutendsten sind hervorgehoben)

Name	Land	Art	Verwendung	Stellen	Struktur
Auerswald	DDR	Form/ Funkt.	Umformtechnik	4	H
Gurevich	UDSSR	Form/ Funkt.	Umformtechnik	9	TH
Knight	UK	Form/ Funkt.	Umformtechnik	4	TH
Salford	UK	Form/ Funkt.	BleCHbearbeitung	6	TH
Puschmann	D	Form/ Funkt.	BleCHbearbeitung	3	TH
WZL	D	Form/ Funkt.	Stahlbau	9	TH
Pacyna	D	Form/ Funkt.	Giesstechnik	30	TH
Malek	CZ	Form/ Funkt.	Giesstechnik	12	TH
Law	UK	Form/ Funkt.	Giesstechnik	8	TH

H: hierarchische Klassifikation

TH: teilhierarchische Klassifikation

Tabelle (T002okIz) die weltweit bekanntesten Klassifikationsschlüssel  
(die bedeutendsten sind hervorgehoben)

Das bekannteste Klassifizierungssystem aus dieser Zeit wurde am Laboratorium für Werkzeugmaschinen und Betriebslehre an der Technischen Universität Aachen für Maschinenbau-Einzelteile entwickelt [Opi-66]. Der sogenannte Opitz-Schlüssel wird in seiner Originalform oder in mehr oder weniger starken Abwandlungen (z.B. das PGM- und IAMA-System) in In- und Ausland angewendet.

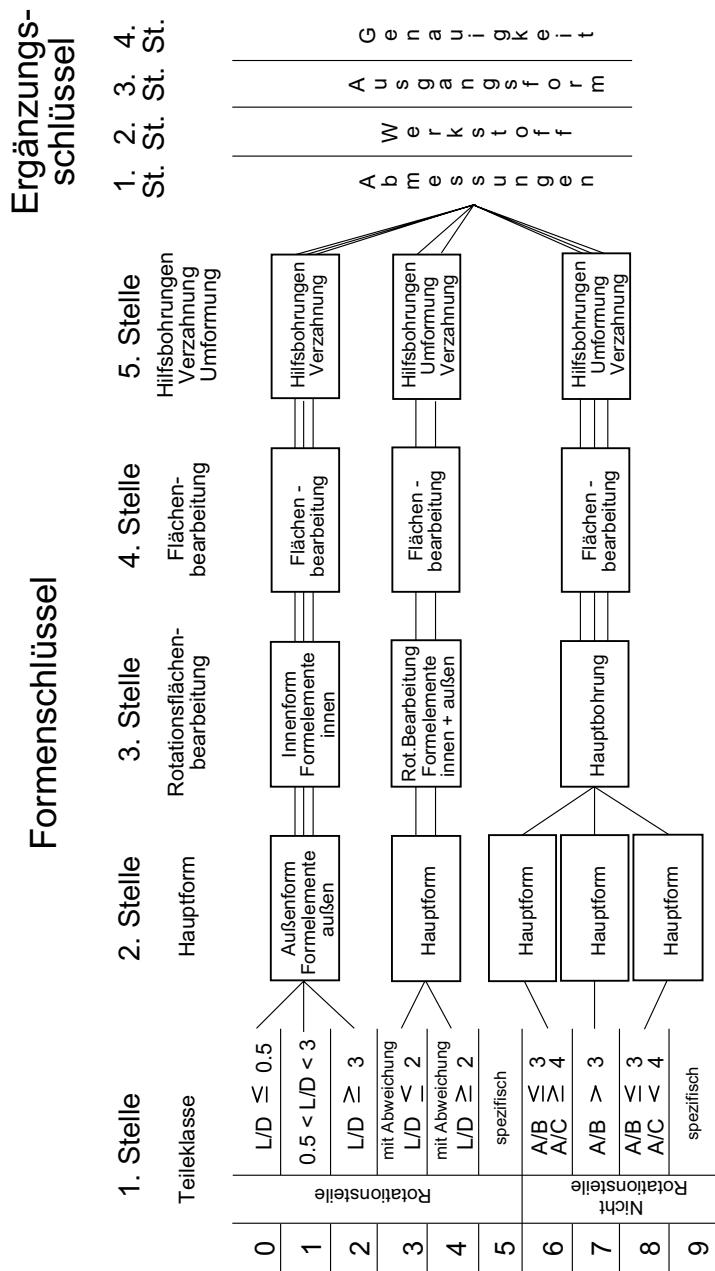
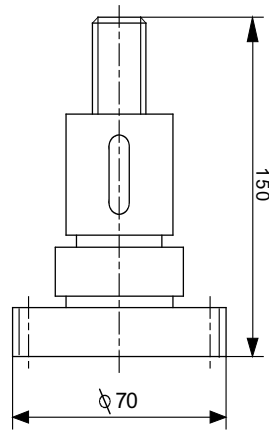


Bild (B012oklZ) Klassifikationsschlüssel nach Opitz [Opi-66]

Er baut im wesentlichen auf einer Formenbeschreibung durch einen fünfstelligen hierarchischen Formenschlüssel auf, ist produktunabhängig und besonders für mechanisch bearbeitete Einzelteile des Maschinenbaus geeignet. Bestandteil ist zudem ein vierstelliger paralleler Ergänzungsschlüssel in welchem Informationen über die Aussenform des Rohlings, dem Werkstoff, die Abmessungen des Fertigteils und

der Genauigkeit dem klassifizierten Teil bei Bedarf hinzugefügt werden können.



Teilkategorie	Teilkategorie		Außenform Formelemente		Innenform Formelemente		Flächenbearbeitung		Hilfsbohrungen und Verzahnung		
	0	1	0	1	0	1	0	1	0	1	
0	L/D $\leq 0.5$	0.5 < L/D < 3	glatt ohne Formelemente	ohne Formelemente	ohne Bohrung ohne Durchbruch	ohne Formelemente	ohne Flächenbearbeitung	0	ohne Bohrung ohne Durchbruch	0	ohne Verzahnung
1			oder glatt	Gewinde	Gewinde	Gewinde	ebene u./o. in einer Richtung gekrümmte Flächen, außen in einem Teilungspunkt	1	Hilfsbohrung	1	Hilfsbohrung
2	L/D $\geq 3$	mit Abweichung L/D $\leq 2$	Funktions-einsch	ohne Formelemente	Funktions-einsch	ohne Formelemente	Nur u./o. Schlitz	2	axial ohne Teilung	2	axial ohne Teilung
3			ohne		glatt o. eins. stgd	ohne	Vielkeil (Polygon) außen	3	axial mit Teilung	3	axial mit Teilung
4	mit Abweichung L/D $\geq 2$		mehrfach steigend	Gewinde	mehrfach steigend	Gewinde	Vielkeil (Polygon) innen	4	radial ohne Teilung	4	radial ohne Teilung
5			Funktions-einsch	Funktions-einsch	ebene Fläche u./o. Nut u./o. Vielkeil außen	ebene Fläche u./o. Nut u./o. Vielkeil außen	5	axial u./o. radial mit Teilung u./o. sonstige Richtungen	5	axial u./o. radial mit Teilung u./o. sonstige Richtungen	
6	AB $\leq 3$	AC $\leq 4$	Funktionskonus	Bewegungsgewinde	Funktionskonus	Bewegungsgewinde	6	Keilverzahnung	6	Keilverzahnung	
7	AB > 3	AC > 4	sonstige	sonstige	sonstige	sonstige	7	sonstige	7	sonstige	
8	AB $\leq 3$	AC $\leq 4$					8		8		
9	spezifisch						9		9		

Resultierender Klassifikationsschlüssel : 1 3 0 3 2

Bild (B013okIz) Klassierungsvorgang mit dem Opitz-Schlüssel (ohne Ergänzungsschlüssel)

Nachteilig wirkt sich die sehr hohe Informationsverdichtung und der damit verbundene Informationsverlust aus, der sich aus der Reduktion realer Merkmalsausprägungen, d.h. exakter Werte, auf einen abgegrenzten Wertebereich ergibt (vgl. Abschnitt Begriffe, Merkmale und Eigenschaften). Je geringer die Anzahl an Schlüsselstellen ist, desto grösser ist die Heterogenität innerhalb einer Klasse. Im Laufe der Zeit kann es daher zu einer Konzentration von Objekten in nur wenigen

Gruppen kommen, während eine grosse Anzahl anderer Gruppen nur wenige Objekte enthält. Die Auswertungsmöglichkeiten werden stark eingeschränkt, da innerhalb dieser umfangreichen Gruppen das Ergebnis nicht weiter differenziert werden kann und somit ein grosses Potential zur Werkstückwiederverwendung verloren geht. Darüber hinaus ist das Ordnen der Elemente einer Gruppe nach dem Grad der Ähnlichkeit zu einem gesuchten Teil nicht möglich. Innerhalb einer Gruppe haben alle Teile die gleiche Wertigkeit, da keine (oder lediglich durch die gruppenspezifischen Merkmale eine implizite) Teilebeschreibung vorliegt.

Durch die begrenzte Länge und der fixen Merkmalsfestlegung von seiten der Nummerungstechnik sind Klassifikationsschlüssel zudem in sich starr. Diese Restriktionen schränken die Änderungs- und Erweiterungsmöglichkeiten und damit die häufig geforderte Flexibilität stark ein.

### 6.2.2 Anwendung von Klassifikationsschlüssel

Klassifikationsschlüssel finden aufgrund der oben beschriebenen Nachteile und Probleme sowie durch das Aufkommen leistungsfähiger PDM-Systemen kaum noch praktische Anwendung zur Ähnlichkeitsuche und zur Teilefamilienbildung.

Innerhalb der Nummerungstechnik finden Klassifikationsschlüssel dagegen eine breite Anwendung. Zielsetzung ist die Verschlüsselung von Merkmalen bzw. Eigenschaften und die Integration in ein Nummerungssystem. Es handelt sich dabei jeweils um unternehmensspezifische Klassifikationsschlüssel.

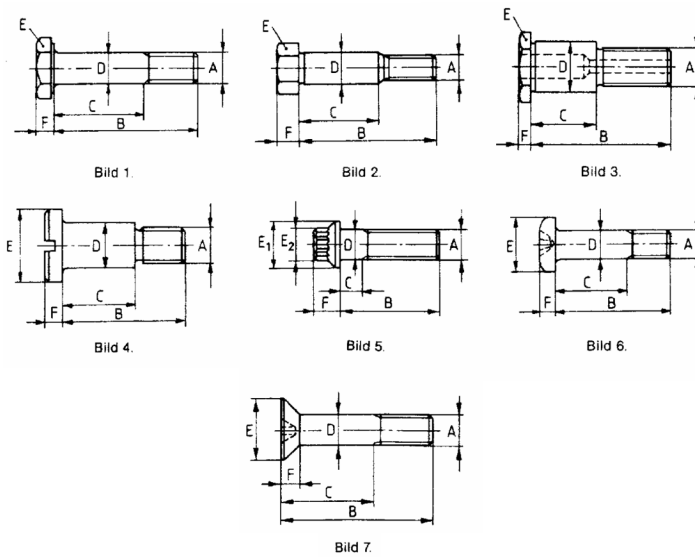
### 6.3 Sachmerkmaleisten

Sachmerkmaleisten (SML) sind auf eine einstufige Ordnung beschränkt. Abgesehen von der durch das Einordnen der entsprechenden Elemente in eine Sachmerkmaleiste resultierenden einstufigen Ordnung, unterliegen sie keiner hierarchischen Ordnung. In DIN 4000 ist eine Sachmerkmaleiste wie folgt definiert:

„Eine Sachmerkmaleiste ist die Zusammenstellung und Anordnung der für eine Gegenstandsgruppe relevanten Sachmerkmale“.

Grundlage der Sachmerkmaleiste ist die Einteilung von Objekten in Gruppen, so dass zwischen den jeweiligen Objekten einer SML eine Ähnlichkeitsbeziehung besteht. Eine SML kann für unterschiedliche Gegenstandsgruppen eine verschiedene Anzahl an Objekten beinhalten. Der Aufbau von Sachmerkmaleisten wird innerhalb der DIN 4000 festgelegt (Bild B014okIz).

#### 3.2 Sachmerkmal-Leiste Nr 2 für Paßschrauben



Sachmerkmal-Leiste DIN 4000 - 2 - 2									
Kennbuchstabe	A	B	C	D	E	F	G	H	J
Sachmerkmal-Benennung	Gewinde	Länge	Schaftlänge	Schaftdurchmesser und zul. Abweichung	Kopfdurchmesser und/oder Schlüsselweite E <sub>1</sub> , E <sub>2</sub>	Kopfhöhe	Bestellzusätze nach Norm	Werkstoff	Oberfläche und/oder Schutzart
Referenzhinweis									
Einheit	–	mm	mm	mm	mm	mm	–	–	–

Bild (B014okIz) DIN Sachmerkmaleiste nach DIN 4000 [DIN 4000]



Die einzelnen Objekte werden in einer Sachmerkmaliste durch ein allgemeingültiges Komplettteil mittels Merkmalen beschrieben. Dabei wird nach DIN 4000 zwischen Sachmerkmalen und Relationsmerkmalen unterschieden:

- *Sachmerkmale* sind Merkmale, die Gegenstände unabhängig vom Umfeld (z.B. Herkunft oder Verwendung) beschreiben. Die Änderung eines Sachmerkmals ergibt einen anderen Gegenstand. Beispiel für ein Sachmerkmal ist die Schlüsselweite einer Schraube. Ändert man die Schlüsselweite dieser Schraube ergibt dies eine andere Schraube.
- *Relationsmerkmale* sind Merkmale, die eine Beziehung von Gegenständen zu ihrem Umfeld kennzeichnet. Im Gegensatz zu Sachmerkmalen ergibt die Änderung einer Ausprägung eines solchen Merkmals keinen anderen Gegenstand. Die Herstellungskosten, Bestellmenge oder Prüfmethode sind Beispiele für Relationsmerkmale.

Die jeweiligen Sachmerkmale können sich in Anzahl und Art von Sachmerkmalisten unterscheiden. Sachmerkmalisten sollen dabei jedoch so angelegt sein, dass sie eine zweckmässige Anzahl ähnlicher Gegenstände einschliesst [DIN 4000].

Art und Anzahl der Merkmale einer Sachmerkmaliste sind zunächst beliebig wählbar. Lediglich der prinzipielle Aufbau von Sachmerkmalisten ist festgelegt. Für ausgewählte Normteile sind die einzelnen Sachmerkmale sowie deren Bezeichnung und Kennung innerhalb der DIN 4000 Teil 2 definiert worden (ein Beispiel zeigt Bild B014okIZ).

Zwei Elemente sind im Sinne der Sachmerkmaliste dann identisch, wenn alle Sachmerkmalausprägungen der Sachmerkmaliste übereinstimmen. Aufgrund der expliziten Teilebeschreibung kann, im Gegensatz zu den Klassifizierungssystemen, innerhalb einer Sachmerkmaliste eine Rangordnung der Elemente nach Grad der Ähnlichkeit zum gesuchten Objekt erstellt werden. Hierdurch kann anhand der jeweiligen Merkmalsausprägungen zu einem gesuchten Teil ein identisches Teil oder das jeweils Ähnlichste ermittelt werden. Im Gegensatz zu Klassifikationsschlüsseln eignen sich Sachmerkmalisten zur expliziten Teilebeschreibung ohne jeglichen Informationsverlust.

Um das Sachmerkmalistenkonzept der DIN 4000 innerhalb der rechnerintegrierten Produktentwicklung nutzen zu können, wurden aufbauend auf der DIN 4000 CAD-spezifische Ergänzungen vorgenommen. Dies führte zur Definition sogenannter CAD-gerechter

Sachmerkmale welche in der DIN V 4001 beschrieben sind. Auf die DIN 4001 soll in Abschnitt 13\_detailliert eingegangen werden.

Das Sachmerkmalleistenkonzept der DIN 4000 bzw. DIN 4001 wurde schliesslich von der internationalen Normungsbestrebung im Bereich Repräsentation und Austausch von Teilebibliotheken aufgegriffen (ISO 13584) und hinsichtlich der Funktionalität sowie der datenmodelltechnischen Abbildung massgeblich erweitert. Innerhalb Abschnitt 13 werden die zugrunde liegenden Konzepte sowie die datenmodelltechnische Abbildung detailliert beschrieben.

### 6.3.1 Anwendung von Sachmerkmalleisten

In einer Sachmerkmalleiste werden somit alle möglichen Instanzen eines abstrahierten Bauteils definiert. Identifiziert werden diese über die jeweiligen Ausprägungen zu den einzelnen Sachmerkmalen. Wird ein Element einer Sachmerkmalleiste innerhalb einer Konstruktion verwendet, kann dieses entsprechend instanziiert und im folgenden unabhängig vom jeweiligen Grad der Automatisierung bzw. Rechnerunterstützung über die eindeutige Angabe der zugehörigen Sachmerkmalleiste (z.B. DIN 911) in Verbindung mit den identifizierenden Parametern dieser Sachmerkmalleiste (z.B. M4x50) identifiziert werden (also DIN 911 M5x50). Das Sachmerkmalleistenkonzept auf Basis der DIN 4000 wurde ursprünglich zur Handhabung häufig verwendeter Normteile (z.B. Schrauben, Stifte etc.) bzw. genormter Gestaltungselemente definiert (z.B. Freistiche, Einstiche etc.). Von zentraler Bedeutung ist in diesem Zusammenhang die Repräsentation der innerhalb einer Norm definierten identifizierenden Parameter eines Bauteils (z.B. Durchmesser, und Länge einer Schraube) sowie die Zuordnung zu dem entsprechenden Komplettteil.

Die einfache und übersichtliche Struktur und nicht zuletzt die Tatsache, dass die implementierungstechnische Umsetzung nicht Bestandteil der Norm ist, haben dazu beitragen, dass das Sachmerkmalleistenkonzept von vielen CAx, PDM- sowie PPS-Systemen aufgegriffen wurde und heute eine breite Anwendung findet.

Als wichtigste Einsatzgebiete sind die folgenden zu nennen:

- Innerhalb der meisten PDM-Systeme kommen Sachmerkmalleisten in Kombination mit einem übergeordneten Begriffssystem zur systematischen Strukturierung und Beschreibung von Teilstämmen zum Einsatz.
- PPS-Systeme verwenden Sachmerkmalleisten in derselben Weise wie PDM-Systeme. Oftmals werden die definierten Sachmerkmale innerhalb des Systems in weiteren Anwendungen wie zum Beispiel der Produktkonfiguration verwendet.

- CAD-Normteilkpakete verwenden Sachmerkmalleisten in der originären Weise zur Handhabung und Darstellung der benötigten Parametertabellen. Die vom Anwender ausgewählte Sachmerkmalkombination wird dem parameterisierten CAD-Modell zur Instanzierung des benötigten Bauteils übergeben.
- Digitale Teilekataloge, wie zum Beispiel die immer mehr grössere Verbreitung findenden Online-Teilekataloge, basieren in der Regel auf dem Sachmerkmalleistenkonzept. Anstelle des innerhalb der PDM- oder PPS-Systeme verwendeten übergeordneten Begriffsystems werden diese in die jeweilige Katalogstruktur eingebunden.

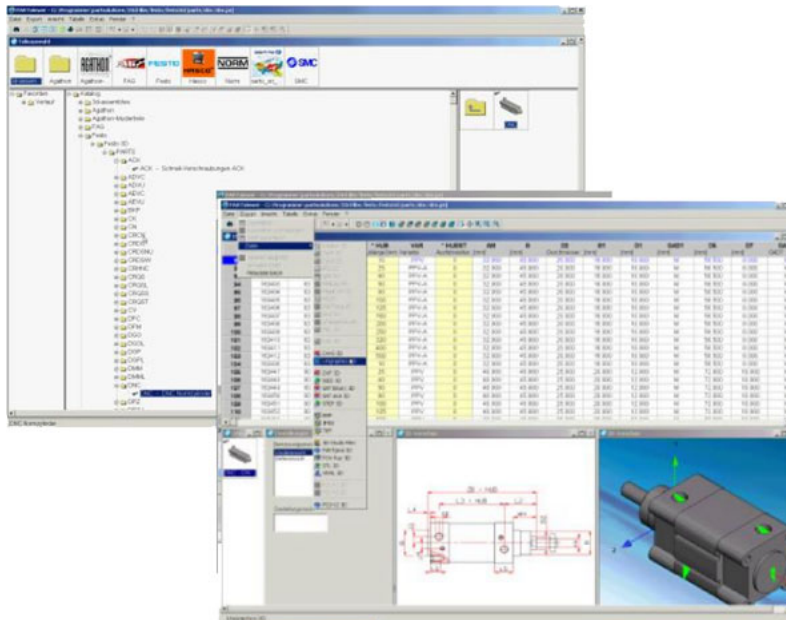


Bild (B0150kIZ) Beispiel einer CAD Teilebibliothek

Entsprechend der jeweiligen Zielsetzung bzw. dem Anwendungsbereich existieren grosse Unterschiede hinsichtlich dem Verhalten, der Integration sowie der Implementierung des Sachmerkmalleistenkonzeptes.

Innerhalb von CAD-Normteilkpaketen wird das ursprüngliche Sachmerkmalleistenkonzept verwendet. Die Datenmodelltechnische Abbildung ist *objektmengenzentriert*. Um Inhalte von Sachmerkmalleisten einfach austauschen bzw. neue Inhalte integrieren zu können basiert die datentechnische Modellierung auf einer zweistufigen Abstraktion. Zielsetzung ist die Trennung von Sachmerkmalleistendefinition und

Sachmerkmaleisteninhalt. Die Sachmerkmaleistendefinition umfasst deren Identifizierung, die Definition der Sachmerkmale sowie die Kennzeichnung der identifizierenden Merkmale. Der Sachmerkmaleisteninhalt wird der Definition spaltenweise zugeordnet (vgl. Bild B016okIZ).

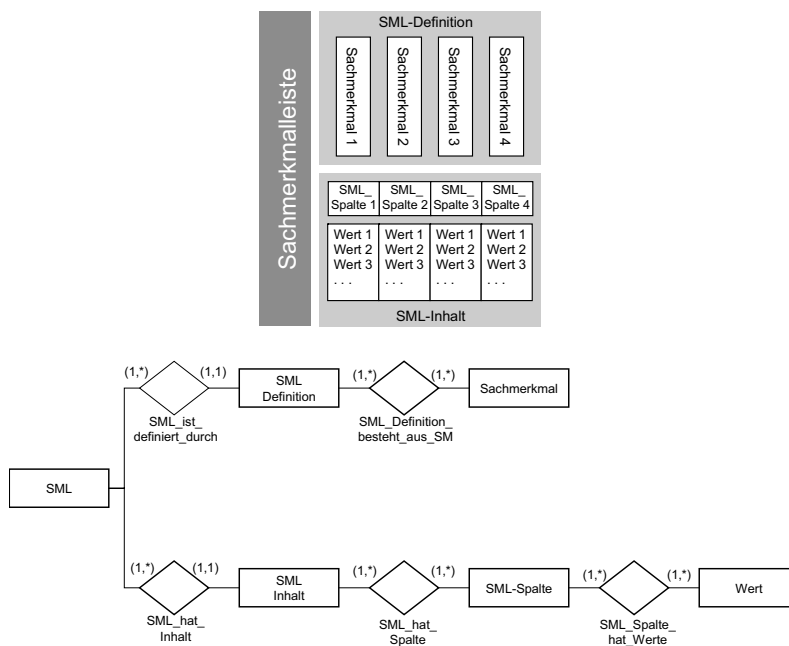


Bild (B016okIZ) Objektmengenorientierte SML-Modellierung auf Basis einer zweistufigen Abstraktion

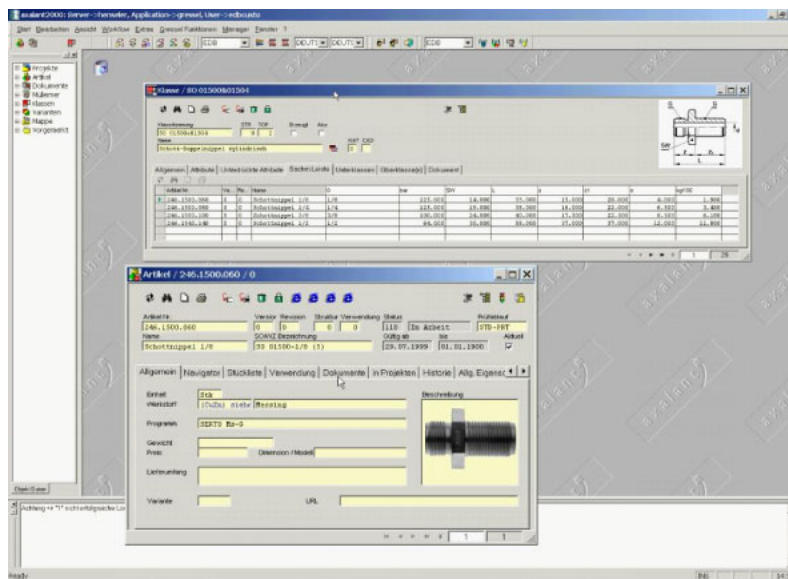


Bild (B017okIz) Beispielhafte SML eines PDM-Systems

Im Gegensatz zum „ursprünglichen“ Sachmerkmaleistenkonzept dienen Sachmerkmaleisten in PDM-Systemen nicht zur Beschreibung aller prinzipiell möglichen Instanzen einer Klasse sondern zur Beschreibung existierender PDM-Objekt (vgl. Bild B019okIz). Diese werden primär über die jeweilige eindeutige Identnummer identifiziert und mittels Sachmerkmalen innerhalb einer Sachmerkmaleiste beschrieben.

Demgemäss ist die datentechnische Modellierung des Sachmerkmaleistenkonzeptes innerhalb von PDM-Systemen *objektidentitätszentriert*. Dabei finden zwei grundsätzlich verschiedene Konzepte Anwendung zur Datenmodelltechnischen Abbildung von Sachmerkmaleisten:

- Die vollständige Abstraktion logischer Sachmerkmaleisten und deren Abbildung auf ein Metaschema zeichnet sich durch die hohe Flexibilität aus. Charakteristisch ist, dass die Einordnung eines Objektes in eine Sachmerkmaleiste indirekt über die Ausprägungen des Objektes bezüglich der Sachmerkmale in einer Sachmerkmaleiste erfolgt. Sollen Sachmerkmale global für mehrere Sachmerkmaleisten definiert werden, ist folglich eine disjunkte Objekt/SML-Zuordnung nicht mehr möglich. Notwendig hierfür ist ein zusätzliches Konstrukt zur Abbildung der SML-Zugehörigkeit eines Objektes.

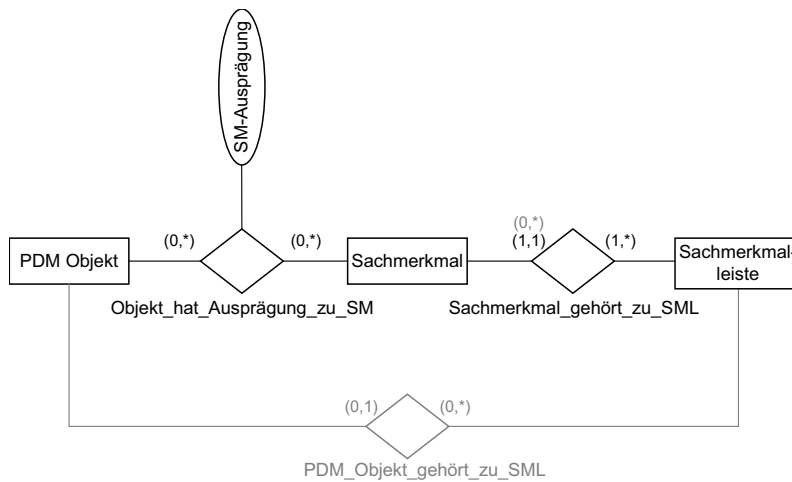


Bild (B0180klZ) vollständige Abstraktion des SML-Konzeptes und Abbildung auf ein Metamodell

- Die teilweise Abstraktion logischer Sachmerkmalleisten trennt im Gegensatz zur vollständigen Abstraktion zwischen einer abstrahierten Definition der Sachmerkmalleiste bzw. deren Sachmerkmale, sowie dem konkret modellierten Inhalt einer Sachmerkmalleiste bzw. den darin jeweils beschriebenen Bauteilen. Nachteilig sind zum einem die in einem relationalen Datenbanksystem entsprechend hohe Anzahl dynamisch anzulegender Tabellen. Desweiteren muss die Integrität (Anzahl und Reihenfolge der Attribute sowie der Datentyp) von Sachmerkmalleistendefinition und -Inhalt über das Anwendungsprogramm sichergestellt werden. Von Vorteil ist, dass gemäss der jeweiligen Definition die Angabe eines Sachmerkmals erzwungen werden kann (not null-Definition). Die Zuordnung Objekt/SML erfolgt bei dieser Modellierungsmethode direkt über die explizite Modellierung des entsprechenden Sachmerkmalleisteninhalts. Damit können prinzipiell Sachmerkmale global definiert werden. Allerdings kann eine Abfrage über global definierte Sachmerkmale nicht alleinig auf Basis einer Datenbankabfrage erfolgen. Die Tabellen müssen einzeln abgefragt und das Gesamtergebnis über das Anwendungsprogramm ausgewertet werden.

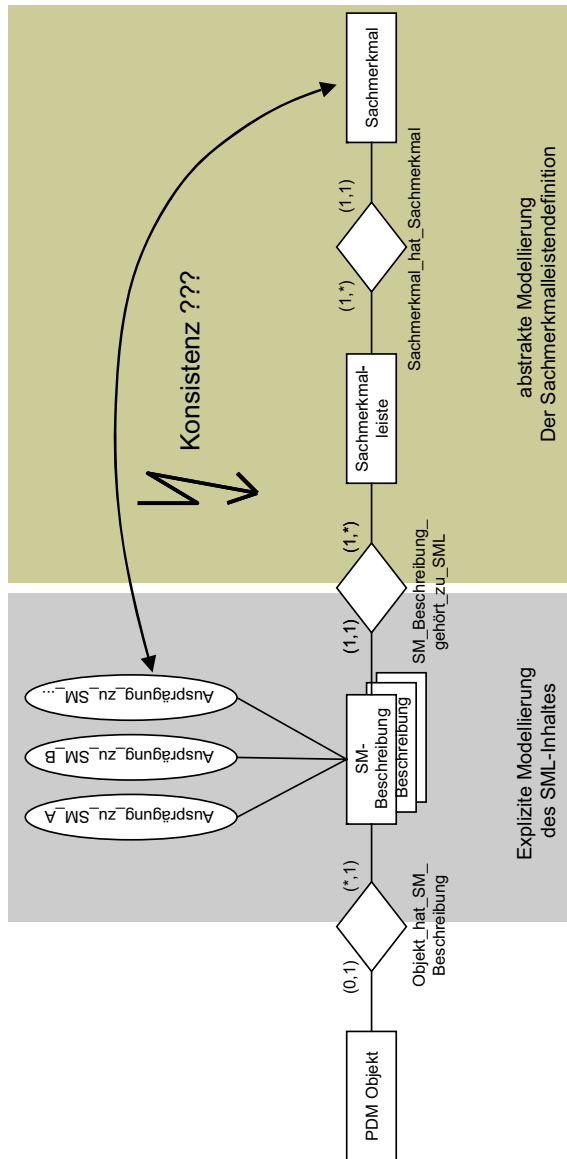


Bild (B019okIz) zweistufige Abstraktion des Sachmerkmaleistenkonzeptes

## 6.4 Verbindung BS / SML

Im Gegensatz zu Klassifikationsschlüsseln und Begriffssystemen, wo die Teilebeschreibung allein auf der Gruppenzuordnung sowie der Gruppenbeschreibung basiert und daher mit einem Informationsverlust verbunden ist, bleibt bei einer expliziten Objektbeschreibung einer SML der Informationsgehalt voll erhalten. Andererseits sind Sachmerkmale zur Wiederholteilsuche jedoch nur sehr beschränkt verwendbar, da aufgrund der nur einstufigen Gliederungstiefe zunächst jene Sachmerkmale ermittelt werden muss welche das gesuchte Teil enthält.



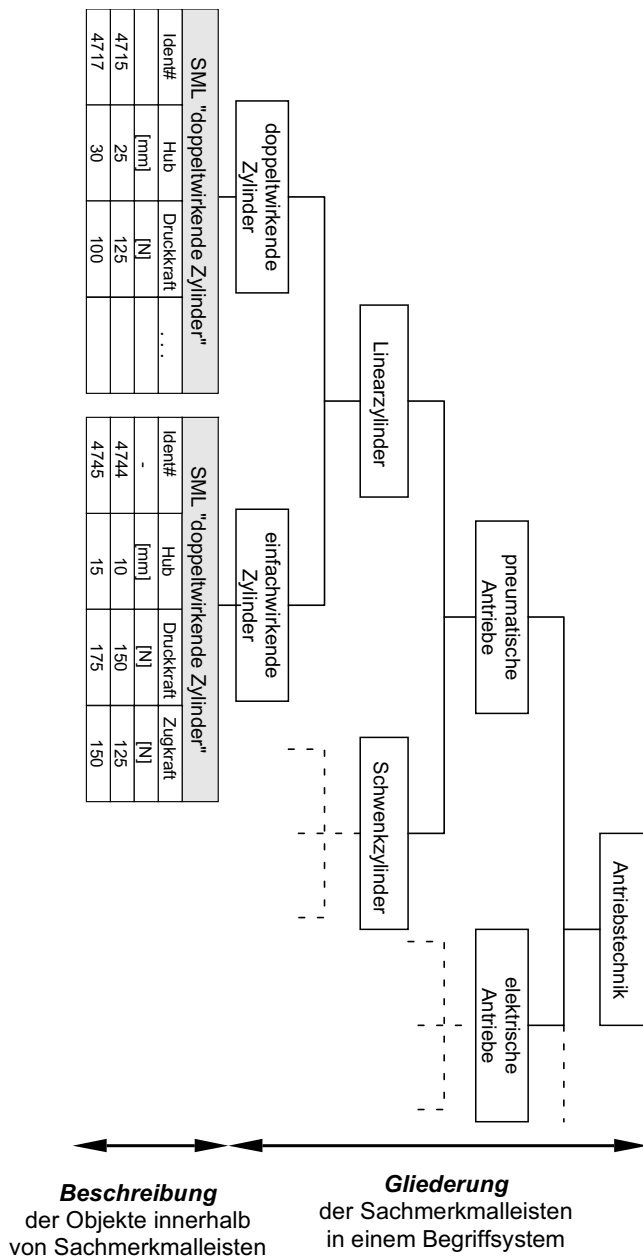


Bild (B020okIz) Gliederung und Beschreibung des Teilstamms auf Basis der Kombination Klassifikation und SML

Eine Kombination von Sachmerkmaleiste und Begriffssystem verbindet die Vorzüge der exakten Teilebeschreibung innerhalb einer Sachmerkmaleiste mit denen von Begriffssystemen zur Gliederung von Objekten bzw. Objektgruppen. Dabei wird das Teilespektrum durch eine hierarchisch-generische Klassifikation soweit unterteilt, bis jede Teileklasse der untersten Stufe durch eine Sachmerkmaleiste zufriedens-

tellend ohne Informationsverlust beschrieben werden kann (vgl. Bild (B020okIz).

Die Sachmerkmaleisten müssen infolgedessen zusätzlich „geordnet“ werden, um sie untereinander differenzieren und somit innerhalb einer Ähnlichteilsuche verwenden zu können.

#### 6.4.1 Anwendung der Kombination Begriffssystem-Sachmerkmaleiste

Die Kombination Sachmerkmaleiste – Begriffssystem findet ausschliesslich innerhalb der statischen Klassifikation Anwendung. Aufgrund der oben erwähnten Vorteile werden Sachmerkmaleisten praktisch immer mit einem Begriffssystem kombiniert.

PDM-Systeme bieten oftmals ein getrenntes Klassifizierungsmodul an, mit welchem derartige Klassifikationsstrukturen durch den Anwender definiert werden können. Um den Anwendern dabei eine grösstmögliche Flexibilität hinsichtlich der Definition benötigter Klassifikationssysteme zu bieten basieren die Klassifizierungsmodule auf der oben beschriebenen Metamodellierung. Die gewünschte Einschränkung auf ein monohierarchisches Begriffssystem muss folglich durch den Benutzer selbst sichergestellt werden.

Leistungsfähige Systeme erlauben ausserdem die Definition globaler Sachmerkmale, welche für eine beliebige Anzahl an Untergruppen Gültigkeit besitzen (vgl. Bild B021okIz).

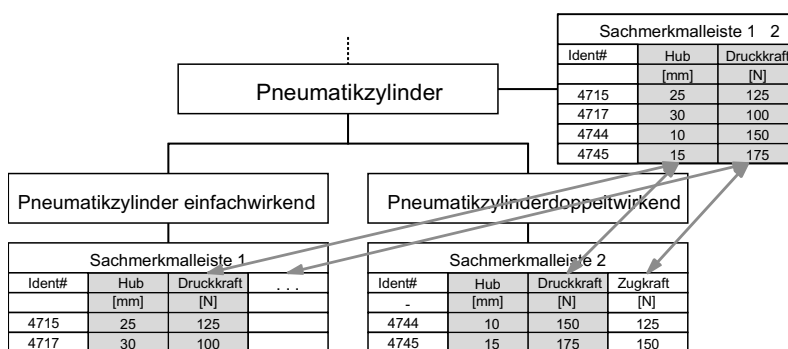


Bild (B021okIz) globale Definition und Abfrage von Sachmerkmalen

Die Definition globaler Sachmerkmale, die für eine beliebige Anzahl an Untergruppen Gültigkeit besitzen, erlaubt einen Sachmerkmaleisten- bzw. gruppenübergreifenden Zugriff auf Objekte. Entsprechend können diese auf der jeweiligen Obergruppenstufe angezeigt und

abgefragt werden. Zwingende Voraussetzung ist, dass diese globalen Sachmerkmale für alle Untergruppen über welche diese definiert sind, dieselbe Semantik aufweisen.

Da beispielsweise die Sachmerkmale „Hub“ und „Druckkraft“ innerhalb der beiden Sachmerkmalleisten „Pneumatikzylinder einfachwirkend“ und „Pneumatikzylinder doppelwirkend“ definiert sind sowie die selbe Semantik aufweisen, ist es möglich diese in der übergeordneten Gruppe „Pneumatikzylinder“ zur Auswahl anzubieten (vgl. Bild B021okIZ). Das Sachmerkmal „Zug-Kraft“ darf nicht nach oben gezogen werden, da dies lediglich in einer der beiden Sachmerkmalleisten (Pneumatikzylinder doppelwirkend) definiert ist.

Das Sachmerkmal „Hub“ dürfte beispielsweise nicht global für die Sachmerkmalleisten „Kolbenmotor“ und „Pneumatikzylinder“ definiert werden, da der „Hub“ im Kontext eines Pneumatikzylinder eine andere Semantik aufweist als im Kontext eines Kolbenmotors.

Um Entscheiden zu können, ob aus semantischen Gesichtspunkten ein Sachmerkmal global über eine Anzahl von Sachmerkmalleisten definiert werden darf, kann auf folgende Kriterien zurückgegriffen werden:

- Austauschbarkeitskriterium  
Zwei nicht identische Objekte welche zu verschiedenen Sachmerkmalleisten gehören, müssen unter bestimmten Umständen austauschbar sein und sollen falls sie ausgetauscht werden hinsichtlich des global definierten Merkmals identisch sein. Die beiden Pneumatikzylinder 4715 (einfachwirkend) und 4744 (doppelwirkend) sind hinsichtlich des Merkmals „Hub“ austauschbar. Das Austauschbarkeitskriterium ist damit erfüllt.
- Homogenitätskriterium  
Soll ein Merkmal global über eine Anzahl von Sachmerkmalleisten definiert werden, muss dieses das selbe Verhalten in Bezug auf automatische oder nicht automatische Prozesse oder Methoden, welche auf eine Liste von Objekten angewendet werden aufweisen. Das Merkmal „Druckkraft“ erfüllt dieses Kriterium hinsichtlich der beiden Sachmerkmalleisten aus, da sich diese jeweils entsprechend der gleichen physikalischen Gesetzmässigkeit  $F = \Delta p A$  errechnet.

## 6.5 Merkmalbasierte Gliederung und Beschreibung

Die merkmalsbasierte Gliederung und Beschreibung erweitert den Ansatz der Kombination Begriffssystem / Sachmerkmalleiste dadurch, dass die einzelnen Gruppen bzw. Obergruppen wie die Objekte einer Sachmerkmalleiste ebenfalls mittels Merkmalen bzw. durch die entsprechenden Ausprägungen zu den Merkmalen beschrieben werden. Damit kann der merkmalsbasierte Zugriff nicht nur auf die Objekte einer Sachmerkmalleiste, sondern auch auf die Objektgruppen einer Obergruppe angewendet werden.

Dieser Typ von Ordnungssystem weist zwei Bestandteile auf:

- Ein fest definiertes generisches Begriffssystem bestehend aus einer Anzahl von Gruppen bzw. Obergruppen, in denen auf unterster Ebene die einzelnen Objekte in Sachmerkmalleisten beschrieben werden, dient zur Gliederung und Beschreibung des Objektspektrums. Sachmerkmale können, bei Sicherstellung der gleichen Semantik, wie oben beschrieben auf Obergruppenebene definiert und abgefragt werden.
- Eine zusätzliche Beschreibung der einzelnen Gruppen bzw. Obergruppen erfolgt durch eine Liste von Eigenschaften. Hierfür werden auf der jeweils übergeordneten Obergruppe Gruppenmerkmale definiert. Die einzelnen Gruppen einer Obergruppe können entsprechend den Ausprägungen zu den Gruppenmerkmalen unterschieden werden. Eine Obergruppe kann somit als eine Sachmerkmalleiste betrachtet werden. Der Unterschied besteht darin, dass innerhalb dieser nicht Objekte, sondern Objektgruppen beschrieben werden. Bei diesen Objektgruppen kann es sich ebenfalls um Obergruppen handeln, die weitere Objektgruppen beinhalten. Die aus der Kombination Merkmal/Ausprägung resultierende Eigenschaft ist für alle weiteren Untergruppen sowie für die einzelnen Objekte der Sachmerkmalleiste auf unterster Stufe gültig. Diese Gruppenmerkmale können entsprechend den globalen Sachmerkmalen innerhalb übergeordneter Obergruppen definiert werden, wenn diese innerhalb aller untergeordneten Gruppen die gleiche Semantik aufweisen.

Die strikte Trennung von Gliederung und Beschreibung wird dadurch aufgelöst. Der Vorteil liegt darin, dass aus der vorgegebenen starren Hierarchie, bestehend aus den einzelnen Produktgruppen, flexibel einzelne Teiläste bzw. Elemente entsprechend den jeweiligen Merkmalsausprägungen ausgeschnitten und diese effizient durchsucht werden können. Auf diese Weise werden die Vorteile einer merkmals-

basierten Beschreibung sowohl innerhalb der Sachmerkmalelisten und den darin beschriebenen Objekte als auch in der übergeordneten Struktur genutzt.

Dies ist vor allem unter dem Gesichtspunkt von grosser Bedeutung, dass es oftmals nahezu unmöglich ist, für ein Objektspektrum eine Gliederung zu definieren, die allen Anforderungen Rechnung trägt.

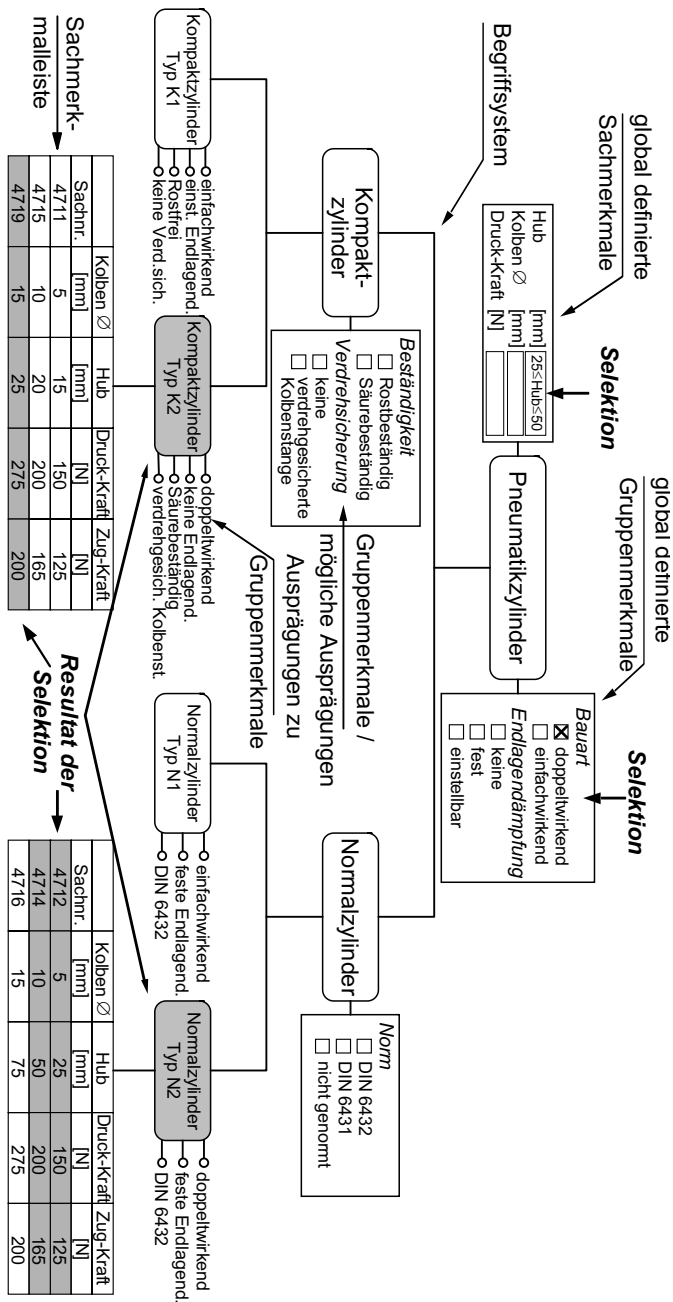


Bild (B022okIz) Prinzip der Merkmalbasierten Suche

Anwendungsbereiche der merkmalsbasierten Gliederung und Beschreibung: Die merkmalsbasierte Gliederung und Beschreibung findet innerhalb der ISO 13584 „Parts Library“ bzw. den auf diesem Standard basierenden Klassifikationssystemen bzw. Teilebibliotheken Anwendung. Zukünftig wird diese Kombination mit Sicherheit nicht zuletzt wegen den mächtigen Abfragemöglichkeiten sowie der vorliegenden Standardisierung in der Praxis Anwendung finden.

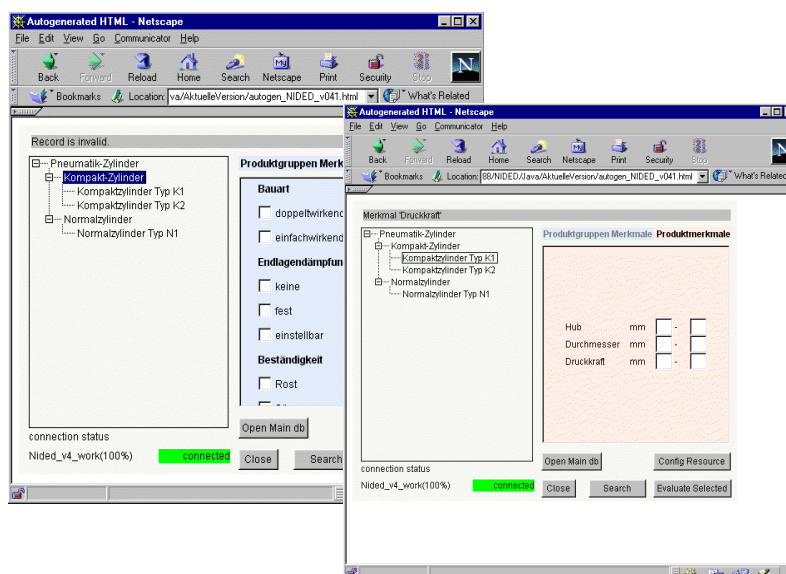


Bild (B023okIz) Merkmalsbasierte Gliederung und Beschreibung in einem JAVA-basierten Klassifikationstool für das Internet.

## 6.6 Generische Objektsysteme

Generische Objektsysteme sind durch Objekte sowie deren Objekteigenschaften charakterisiert. Sie bauen auf dem Paradigma der Objektorientierung auf (Objektidentität, Kapselung von Struktur und Verhalten, Klassifikation bzw. Klassenbildung, Vererbung sowie der Polymorphismus).

Bei den Objekteigenschaften wird zwischen globalen Eigenschaften, die für bestimmte Domänen oder für das gesamte System gelten sowie lokalen Eigenschaften, welche lediglich für ein einzelnes Objekt definiert sind, unterschieden.

Die Definition und Beschreibung der Eigenschaften erfolgt in sogenannten Data Dictionaries. Innerhalb diesen wird die abzubildende Ontologie definiert. Beim Erzeugen neuer bzw. dem Umklassieren vorhandener Objekte wird auf dieses Dictionary zugegriffen. Es dient somit der semantischen Konsistenzsicherung generischer Objektsysteme.

Innerhalb der generischen Objektsysteme wird unterschieden zwischen:

- objektorientierten generischen Objektsystemen,
- objektzentrierten generischen Objektsystemen sowie
- hybriden generischen Objektsystemen.

### 6.6.1 Objektorientierte Systeme

Der objektorientierte Ansatz basiert direkt auf dem Paradigma der Objektorientierung. Die individuellen Objekte werden entsprechend ihren Eigenschaften zu Klassen zusammengefasst, welche hierarchisch strukturiert werden können. Die Eigenschaften der Superklassen werden an deren Subklassen vererbt, welche diese um zusätzliche subclassenspezifische Eigenschaften erweitert. Als Nachteile des objektorientierten Ansatzes sind folgende zu nennen:

- Es ist kein expliziter Mechanismus vorhanden, um zielgerichtet Klassen mit allen zutreffenden globalen Eigenschaften definieren zu können. Es ist lediglich eine Klassendefinition basierend auf den notwendigen Eigenschaften einer Klasse möglich.
- In objektorientierten Datenmodellen kann ein Objekt nur als eine Instanz einer vordefinierten Klasse mit einer festen Anzahl an Eigenschaften erzeugt werden.
- Es ist nicht möglich, ein Objekt zu erzeugen und dieses erst zu einem späteren Zeitpunkt einer Klasse zuzuordnen. Statt dessen muss ein Objekt stets als Instanz einer dedizierten Klasse definiert werden.

- Existierende Klassen können, nachdem Objekte instanziiert worden sind, nicht mehr geändert werden.

### 6.6.2 Objektzentrierte Systeme

Im Gegensatz zu dem hierarchischen Klassensystem des objektorientierten Ansatzes weist der objektzentrierte Ansatz keine Strukturierung der Objekte auf. Jedes Objekt kann durch das objekteigene Verhalten und die Struktur charakterisiert werden sowie eine Anzahl von Eigenschaften aufweisen. Damit bestehen keine Restriktionen hinsichtlich der Definition neuer Objekte. Der Nachteil liegt in der fehlenden Strukturierung der Objekte, was sich negativ auf den zielgerichteten Zugriff auf gesuchte Objekte auswirkt.

### 6.6.3 Hybride Systeme

Der Hybridansatz kombiniert den objektorientierten mit den objektzentrierten Ansatz. Dabei wird das Konzept der hierarchischen Klassenstruktur sowie die Vererbung vom objektorientierten Ansatz übernommen und dadurch erweitert, dass in diese Struktur Objekte entsprechend dem objektzentrierten Ansatz eingebracht werden können.

Die globale Eigenschaften sind Bestandteil der Klassen bzw. der Klassenhierarchie; lokale Eigenschaften sind entsprechend dem objektzentrierten Ansatz für einzelne Objekte definiert.

### 6.6.4 Funktionsprinzipien hybrid-generischer Objektsysteme

Hybrid generische Objektsysteme sind eng verwandt mit der merkmalsbasierten Gliederung und Beschreibung. Unterschiedlich ist im wesentlichen die Art und Weise wie das Begriffssystem aufgebaut wird.

Wie erwähnt, können Merkmale und Eigenschaften gleichermaßen auf Objekte und Objektgruppen angewendet werden. Objektgruppen werden über die zugeordneten Objektgruppeneigenschaften, Objekte über die zugeordneten Objekteigenschaften voneinander unterschieden. Ist für eine Objektgruppe eine Objektgruppeneigenschaft definiert, weisen alle Objekte dieser Objektgruppe diese Eigenschaft auf. Der Objektgruppe 1 aus Bild B024okI2 ist beispielsweise die Objektgruppeneigenschaft Werkstoff = CK45 zugewiesen. Folglich müssen alle Objekte dieser Gruppe aus dem Werkstoff CK45 gefertigt sein.

Um die Objekte einer Objektgruppe unterscheiden zu können, werden einer Objektgruppe Objektgruppenmerkmale zugeordnet.



Einzelne Objekte einer Objektgruppe können entsprechend der Ausprägungen zu diesen Objektgruppenmerkmalen identifiziert werden. Die Wellen der Objektgruppe 2 werden beispielsweise entsprechend der Ausprägung zu den Merkmalen L und D unterschieden (vgl. Bild B024okIZ).

Weisen alle Objekte einer Objektgruppe bezüglich eines Objektgruppenmerkmals dieselbe Ausprägung auf, kann dieses Merkmal zusammen mit der Ausprägung der Objektgruppe als Objektgruppeneigenschaft zugewiesen werden. Da alle Objekte der Objektgruppe 1 den selben Durchmesser D und den selben Durchmesser d aufweisen kann der Objektgruppe 1 die Objektgruppeneigenschaften Durchmesser  $D = 15 \text{ mm}$  und Durchmesser  $d = 10 \text{ mm}$  zugeordnet werden.

Kommt das selbe Merkmal in weiteren Objektgruppen als Basis einer Objektgruppeneigenschaft zur Anwendung, können diese Objektgruppen zu einer übergeordneten Oberobjektgruppe zusammengefasst werden. Die einzelnen Objektgruppen der sich ergebenden Oberobjektgruppe können mittels den verschiedenen Ausprägungen bezüglich diesem Merkmal unterschieden werden. Der Objektgruppe 1 aus Bild B024okIZ ist die Eigenschaft Durchmesser  $D = 15 \text{ mm}$ , der Objektgruppe 2 die Eigenschaft Durchmesser  $D = 12 \text{ mm}$  zugewiesen. Beide Eigenschaften basieren auf dem selben Merkmal (Durchmesser D). Folglich können die beiden Objektgruppen zu einer Oberobjektgruppe zusammengefasst und entsprechend der Ausprägung zum Merkmal D unterschieden werden.

Oberobjektgruppen verhalten sich hinsichtlich diesen beschriebenen Mechanismus wie Objektgruppen.

Auf diese Weise kann entsprechend dem objektorientierten Ansatz eine Objektgruppenhierarchie aufgebaut werden. Die einzelnen Objekte bzw. Objektgruppen besitzen unabhängig davon ein eigenes Verhalten und Struktur, was dem objektzentrierten Ansatz entspricht.

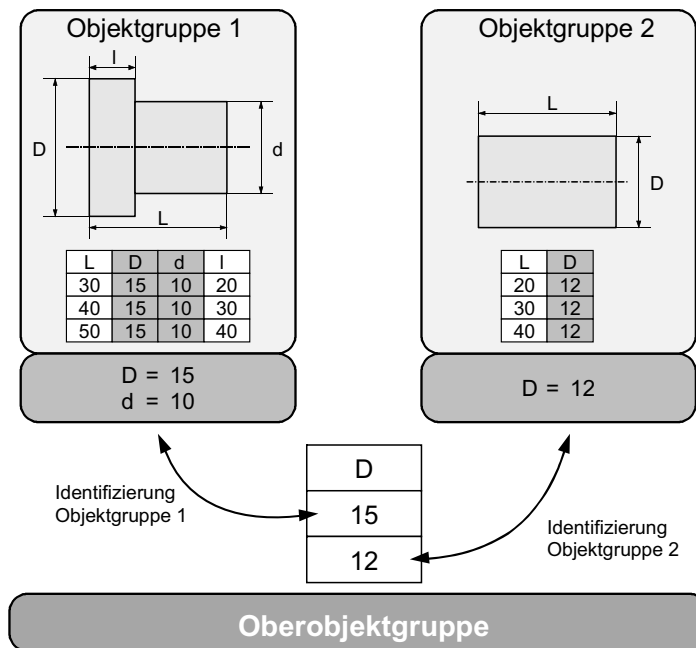


Bild (B024oklZ) Objektgruppeneigenschaften und Objektgruppenmerkmale

Generische Objektsysteme kommen derzeit noch nicht zur praktischen Anwendung. Sie sind vorwiegend Gegenstand der Forschung im Zusammenhang mit dem Aufbau von elektronischen Marktplätzen bzw. verteilten Informationssystemen. Anwendung

## 6.7 Clusteranalyse

Die Clusteranalyse als mathematisch-statistisches (multivariate Statistik) Verfahren der Gruppentechnik dient der Strukturierung einer durch eine Merkmalsmenge  $M$  beschriebenen Objektmenge von  $n$  Objekten in eine optimale Zahl von Klassen (Clustern). Die interessierenden Merkmale müssen wiederum ausgewählt und genau definiert werden. Man erhält somit eine  $m,n$ -Matrix, in der für alle  $n$  Elemente die sie auszeichnenden  $m$  Eigenschaften abgespeichert sind. Auf Basis dieser Datenmatrix wird mit Hilfe sogenannter Distanzfunktionen der Abstand (Distanz  $D$ ) quantifiziert, welcher die Unähnlichkeit von je zwei Objekten im Vergleich zueinander beschreibt. Ergebnis ist eine Distanzmatrix in Dreiecksform, welche die jeweiligen Abstände zweier Elemente im Schnittpunkt der entsprechenden Zeile und Spalte enthält, ähnlich wie in einer Entfernungstabelle, wo der Abstand einer Stadt „A“ von einem Ort „B“ angegeben wird. Diese Distanzmatrix dient als Basis für die eigentliche Clusteranalyse, welche ähnliche Teile zu Clustern zusammenfasst.

Um die Datenmatrix erstellen zu können, müssen in einem ersten Schritt zunächst die verschiedenen Merkmale normiert werden, da diese in der Regel verschiedene Domänen bzw. Skalierungen oder im Extremfall verschiedene Skalierungsarten aufweisen. Aus Bild (B025oklZ) wird der Einfluss der Skalierung auf die spätere Clusterung ersichtlich:

Entsprechend der gewählten Skalierung (Massstab der L- und D-Achse) ändert sich die relative Position zueinander.

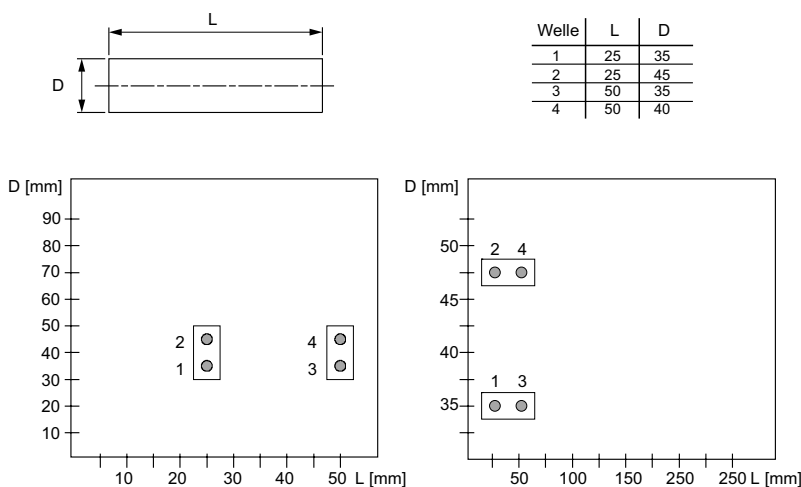


Bild (B025oklZ) Einfluss der Skalierung auf die Gruppenbildung

Jede Merkmalsart erfordert ein entsprechendes Verfahren zur Distanzberechnung, wobei jeweils verschiedene Distanzfunktionen zur Verfügung stehen. Die Distanzberechnung erfolgt zunächst für jedes Merkmal getrennt; anschliessend werden die einzelnen Teildistanzen zu einer repräsentativen Gesamtdistanz verknüpft. Bild B026okIz zeigt häufig verwendete Distanzfunktionen für metrische Merkmale.

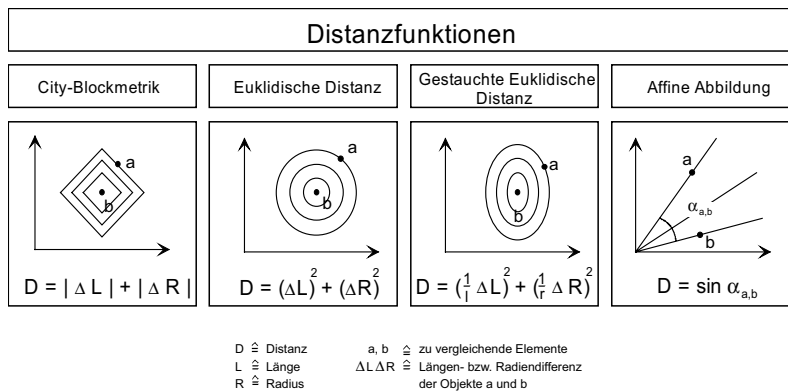


Bild (B026okIz) Distanzfunktionen [FRGR-82]

Die  $n$  Objekte können analog zur kartesischen Geometrie als  $n$  Punkte im  $m$  dimensionalen Merkmalsraum dargestellt werden. In der Distanzmatrix sind die Abstände der Punkte zueinander entsprechend der gewählten Distanzfunktion zu finden. Die von den Objekten gebildete Punktstruktur lässt sich somit ebenfalls im Merkmalsraum darstellen. Ziel der Clusteranalyse ist es nun, Punkthäufungen zu erkennen und zu einer Gruppe zusammenzufassen.

Hierarchische Verfahren untergliedern die Gesamtheit aller Objekte immer feiner, so dass man als Ergebnis eine hierarchische Anordnung der Elemente erhält. Die Gruppenbildung kann graphisch in Form eines Dendogramms dargestellt werden (vgl. Bild B027okIz). Die hierarchische Klassifikation kann entweder agglomerativ oder divisiv erzeugt werden.

Agglomerative Verfahren gehen von den Einzelobjekten aus. Schritt für Schritt werden jeweils jene Objekte oder Objektgruppen zusammengefasst, die am dichtesten beisammen liegen, also das geringsten Distanz  $D$  zueinander aufweisen. Praktischerweise wird die Verknüpfung bis zu einem gewissen Distanzniveau durchgeführt. Das heisst, man gibt ein bestimmtes Grenzdistanzmass  $DG$  vor, ab welchem das Verknüpfen von Objekten und Gruppen abgebrochen wird.

Ergebnis ist eine Anzahl an nicht verbundenen Gruppen, im Dendogramm als nicht verbundene Äste ersichtlich, deren Gesamtheit man als Partition bezeichnet. Ebenso könnte man eine Partition vorgeben und Objekte bzw. Gruppen so lange verbinden, bis der gewählte Grad der Partition erfüllt ist.

Divisive Verfahren gehen von einer Gesamtmenge aus und zerlegen diese sukzessive in einzelne Objektgruppen.

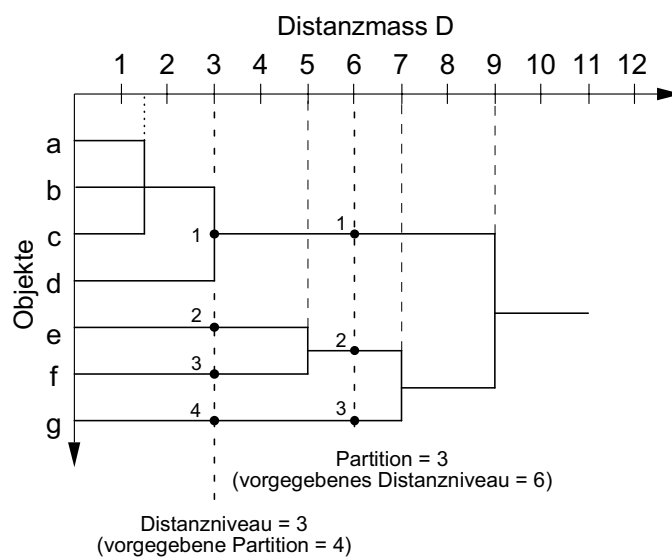


Bild (B027oklZ) Dendogramm

Bei den nicht hierarchischen Verfahren werden die Objekte entsprechend einer gewünschten Partition zunächst mehr oder weniger zufällig auf die entsprechende Anzahl an Gruppen verteilt. Anschließend wird versucht, durch Austausch von Elementen die Homogenität innerhalb der Gruppen zu erhöhen. Die Homogenität einer Gruppierung ist um so besser, je dichter die Objekte einer Klasse aneinander liegen. Ein Gütemass für die „Spannung“ (Heterogenität) innerhalb einer Klasse ist z.B. die Summe der Abstandsquadrate bezogen auf ihren Klassenschwerpunkt.

Häufig kommt eine Mischform der hierarchischen und nichthierarchischen Verfahren zum Einsatz. Mit Hilfe eines hierarchischen Verfahrens wird eine Hierarchie bis zu einem bestimmten Distanzmass aufgebaut. Diese Partition dient dann im folgenden als Ausgangsverteilung für ein Austauschverfahren. Der Vorteil liegt dabei darin, dass

die Startposition problemspezifisch ist und die Gruppen der Endpartition eine vorwählbare Mindesthomogenität einhalten.

Erst nachdem alle Elemente in Gruppen aufgeteilt sind, wird wieder auf die Datenmatrix zurückgegriffen und untersucht, durch welche Merkmale sich die Klassen auszeichnen. Ergebnis ist eine Klassifikation des Objektspektrums und die Ausgabe der charakteristischen Merkmale einer Gruppe. Neu hinzukommende Teile können entweder manuell der jeweiligen Gruppe zugeordnet werden oder die Clusteranalyse wird von neuem durchgeführt.

### 6.7.1 Anwendung der Clusteranalyse

Die Clusteranalyse wird vorwiegend für die generische Klassifizierung verwendet, da sie abgesehen von der Merkmaldefinition vollständig algorithmisierbar und damit weitestgehend automatisierbar ist. Implizite Merkmale müssen auf ordinale Merkmale abgebildet werden, um die Distanzen berechnen zu können, was bei einer grossen Anzahl an impliziten Merkmalen mit einem erheblichen Aufwand verbunden (dieser Vorgang ist nicht algorithmisierbar) und in vielen Fällen nicht einfach zu realisieren ist.

Die Clusteranalyse wird aus diesem Grund vornehmlich innerhalb der Prozessplanung eingesetzt. Hinsichtlich einer praktischen Anwendung innerhalb von PDM-Systemen sind keine praktischen Anwendungen bekannt.

## 6.8 Faktorenanalyse

Bei der Faktorenanalyse handelt es sich wie bei der Clusteranalyse um ein mathematisch statistisches Verfahren mittels welcher Abhängigkeiten zwischen Eigenschaften errechnet werden können.

Die Faktorenanalyse liefert „die einfachste lineare Hypothese über die Struktur, die hinter einer Vielfalt miteinander korrelierter Variablen steht“ [Übe-71]. Die Eigenschaften werden hierfür zunächst in ein mathematisches Modell, dem sogenannten faktorenanalytischen Modell, abgebildet. Geometrisch betrachtet spannen die Eigenschaftsbeziehungen einer Gruppe einen Eigenwertraum auf, dessen Dimensionalität zunächst gleich der Anzahl der Kennwerte ist. Falls zwischen den Kennwerten Beziehungen bestehen, sind diese untereinander teilweise redundant, wodurch die Achsen dieses Raumes nicht orthogonal zueinander liegen, sondern voneinander linear abhängig sind. Stehen die Raumachsen fast parallel, so kann dieser Raum in einen neuen Raum niedrigerer Dimensionalität abgebildet werden. Eine Faktorenanalyse bildet auf diese Weise den niedrigstdimensionalen orthogonalen Unterraum („Faktorenraum“) und setzt dessen Achsen („Faktoren“) genannt in Beziehung zu den Achsen des „Kennwerttraumes“. Der „Faktorenraum“ wird innerhalb des „Kennwerttraumes“ durch Transformationen so rotiert, dass möglichst viele Faktorenachsen parallel zu den Eigenschaftswertachsen stehen [Übe-71]. Auf diese Weise lassen sich die Faktorenwerte aus einer minimalen Anzahl von Eigenschaftsabfragen ermitteln (im Idealfall genauso viele wie Faktoren existieren). Aus der Beziehung zwischen Faktorenraum und Eigenschaftswertraum sind Gleichungssysteme formulierbar, anhand derer es möglich ist, von den Faktorenwerten auf alle Eigenschaften zu schliessen.

In der Praxis trifft der Extremfall vollständig redundanter Kennwerte jedoch nicht zu, d.h. die Achsen des Eigenschaftstraumes sind nicht gänzlich parallel. Dadurch ergibt sich ein Fehler bei der Berechnung der Faktorenwerte aus den abgefragten Eigenschaften, der sich direkt auf die anschliessende Berechnung aller Eigenschaften auswirkt. Ziel des Abfragegenerators ist also eine minimale Redundanz der Eigenschaftsabfragen, d.h. es wird ermittelt, welche Eigenschaften notwendigerweise für den Informationszugriff in dieser Gruppe abgefragt werden müssen. In der Praxis muss daher stets ein Kompromiss zwischen der erforderlichen statistischen Sicherheit der Faktorenanalyse und dem Eingabeaufwand gesucht werden.

Der vollständige Eigenschaftswertsatz wird normiert und mit denjenigen der Objekte der ausgewählten Gruppe verglichen. Als Sucher-

gebnis wird das Element mit der minimalen Distanz  $D$  zum gesuchten Objekt ausgegeben.

Oftmals wird eine detaillierte Vorstellung bezüglich des gesuchten Objektes jedoch noch nicht vorhanden sein. Ein weiterer Nachteil einer solchen expliziten Beschreibung zu Beginn des Suchvorgangs ist ausserdem, dass untergeordnete Eigenschaften, also Eigenschaften die seitens den an das gesuchte Teil gestellten Anforderungen nur eine geringe, oder keine Bedeutung haben, gleichwertig in die Suche mit einbezogen werden. Der Konstrukteur ist damit gezwungen, auch die für seinen Anwendungsfall untergeordneten Kennwerte explizit zu detaillieren, um den Suchvorgang starten zu können. Bei einem unbefriedigenden Erfolg der Recherche muss die Detaillierung neu überdacht, und die Suche erneut gestartet werden.

### 6.8.1 Anwendung der Faktorenanalyse

Die Faktorenanalyse wird in der Regel in Verbindung mit einer Clusteranalyse verwendet. Zielsetzung ist es, den erforderlichen Eingabeaufwand zu verringern. Für den Einsatz von PDM-Systemen hat die Faktorenanalyse keine Bedeutung.



## 6.9 Fuzzy-Sets

Sämtlichen bisher dargestellten Gruppentechniken liegt eine „scharfe Logik“ zu Grunde, d.h. jede Frage kann entweder mit ja (1) oder nein (0) beantwortet werden. Das bedeutet, dass ein Objekt entweder zur Gruppe A oder zu Gruppe B gehört (Ausschliesslichkeit der Gruppenzugehörigkeit).

Die Verwendung unscharfer Zugehörigkeitsfunktionen, den sogenannten Fuzzy-Sets, erlaubt das Miteinbeziehen von Uneindeutigkeiten, welche bei einer Klassifizierung nicht ausgeschlossen werden können und ermöglicht damit, der Realität näher zu kommen.

Die Zugehörigkeitsfunktion  $\mu_A(x)$  drückt das Mass der Zugehörigkeit des Objektes  $X_i$  zur Menge A aus. Der Wert einer Zugehörigkeitsfunktion  $\mu(x)$  liegt immer zwischen 0 und 1.

Wenn  $A(x_i) = 0$  gilt, ist das Objekt  $X_i$  in keiner Weise Element der Menge A. Im Gegensatz dazu ist das Objekt für  $\mu_A(x_i) = 1$  volles Element der Menge A. Im allgemeinen wird  $\mu_A(x_i)$  irgendwo zwischen 0 und 1 liegen und somit den Grad der Zugehörigkeit zur Menge A ausdrücken.

Ist beispielsweise eine Gruppe durch die Merkmalsausprägung „Blechteil mit 90° Abkantung“ definiert bzw. abgegrenzt, würde sich für ein Blech mit einem Abkantwinkel von  $a = 80^\circ$  der Wert der Zugehörigkeitsfunktion zu  $\mu_A(x_{\text{Winkel}}) = 0.75$  ergeben. Das Blechteil mit 80° Abkantwinkel wäre somit mit dem Wert  $\mu_A(x_{\text{Winkel}}) = 0.75$  der Gruppe „Blechteil mit 90° Abkantwinkel“ zugehörig.

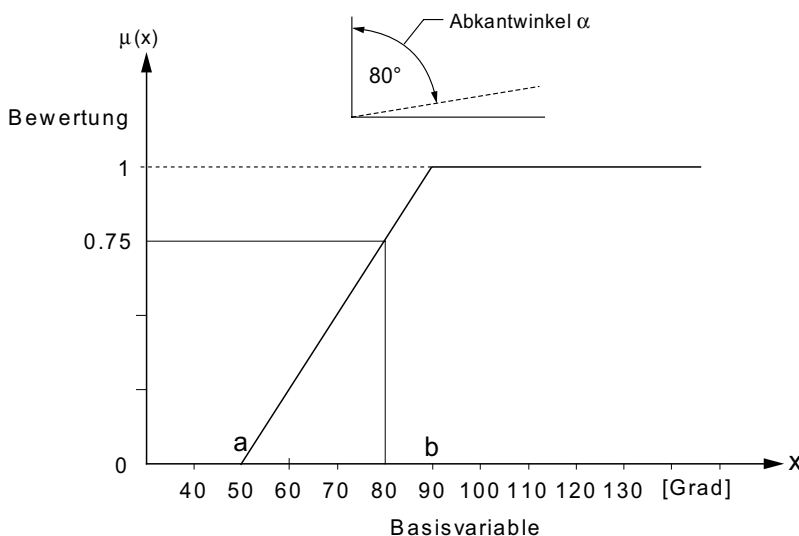


Bild (B028oklZ) Zugehörigkeitsfunktion eines Blechwinkels

Die in Kapitel Clusteranalyse beschriebene Clusteranalyse kann um diese unscharfe Gruppenzugehörigkeit erweitert werden. Das heisst, innerhalb der Normierung der Merkmale muss jedem Merkmal zusätzlich die jeweils geeignete Zugehörigkeitsfunktion zugeordnet werden. D.h. die daraus zu entwickelnde Distanzmatrix enthält als Ergebnis der Distanzberechnung keine festen Distanzen sondern Distanzfunktionen. Als Ergebnis der eigentlichen Clusterung erhält man entsprechend Clusterzugehörigkeitsfunktionen.

### 6.9.1 Anwendung der Fuzzy-Sets

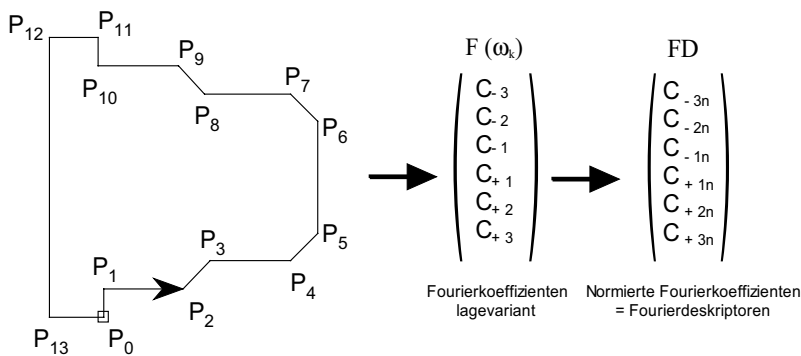
Eine unscharfe Behandlung von Objekten in Bezug auf die Gruppenzugehörigkeit bei der Klassifizierung eines Teilespektrums entspricht wohl am ehesten der Realität. Das Problem liegt allerdings, wie bei den rein mathematisch-statistischen Verfahren, in der Beschreibung eines gesuchten Objektes. Ein systembedingt abzufragendes Merkmal muss auch hier explizit quantifiziert werden, und kann nicht „unscharf“ eingegeben werden. Angewendet wird dieses Verfahren vor allem in der Arbeitsvorbereitung und der Produktionsflussanalyse, in der eine unscharfe Gruppenzugehörigkeit für die Bildung von Teilefamilien von grossem Vorteil ist.

Hinsichtlich einer Anwendung in PDM-Systemen kommen den Fuzzy-Sets keine Bedeutung zu.

## 6.10 Fourierdeskriptoren

Die erstmals von Cosgriff [Cos-60] beschriebene Methode zur Verschlüsselung von Geometriezügen durch Fourierdeskriptoren wurde hinsichtlich einer Fouriertransformation als Methode der Mustererkennung weiterentwickelt, um komplexe geometrische Formen eindeutig und automatisch klassifizieren zu können. Eine Kontur wird dabei durch normierte Fourierkoeffizienten FD lageinvariant beschrieben, welche mittels einer Fouriertransformation des Konturzuges errechnet werden [GRVO-92].

Als Basis dient ein die Kontur beschreibender, räumlicher, geschlossener Linienzug, welcher durch eine Vektorfunktion  $f(a)$  über die Bogenlänge  $a$  repräsentiert wird.



$$A = \overline{P_0}P_1 + \overline{P_1}P_2 + \dots + \overline{P_{13}}P_0 = L$$

Bild (B029oklZ) Ermittlung der Fourierdeskriptoren aus einem Konturzug

Die FD des Vektors sind komplexe Zahlen, die über einen ganzzahligen Index klassifiziert sind. Auf der Basis dieser FD-Vektoren lässt sich die Approximation der Kontur verschlüsseln und der Gestaltunterschied zweier Konturen als reelle Zahl darstellen, was wiederum eine zwingende Voraussetzung für die Anwendung einer automatischen Klassifikation ist.

Aufgrund der globalen Eigenschaft der FD können Detailinformationen, wie z.B. Einstiche, Radien u.ä., nicht klassifiziert werden. Um auch Detailinformationen verschlüsseln bzw. vergleichen zu können, ist somit eine Kopplung des Fourierverfahrens mit lokalen Codierungsverfahren notwendig.

Für den Vergleich von Linienscharen benötigt man zusätzlich Methoden zum Vergleich geometrischer Strukturen (Topologie). Diese beruhen auf der Graphentheorie.

Analog zur eindimensionalen Fouriertransformation für ebene Konturen wird zum Vergleich unterschiedlicher Volumina die zweidimensionale Fouriertransformation angewendet, was allerdings mit einem erheblichen Rechenaufwand verbunden ist. Zur Lagenormierung muss allerdings zuerst die Hauptträgheitsachse  $Q$  des zu transformierenden Volumens berechnet werden. Das kann dazu führen, dass bei kontinuierlichem Aufdicken einer kreisförmigen Platte die Achse mit dem minimalen Trägheitsmoment  $Q_{min}$  von einer bestimmten Dicke der Platte an nicht mehr radial, sondern kolinear zur Zylinderachse liegt. Infolgedessen werden die Fourierdeskriptoren für zwei Körper kurz vor bzw. kurz nach diesem sogenannten Kippunkt, trotz der fast identischen Volumina, sehr verschieden sein.

#### 6.10.1 Anwendung

Das gesuchte Bauteil wird zunächst durch eine Handskizze beschrieben und in das System eingegeben. Anschliessend werden die Fourierdeskriptoren des eingegebenen Konturzug automatisch durch das System bestimmt. Die ermittelten Fourierdeskriptoren werden mit denen der vorhandenen Bauteile verglichen, wofür auf multivariate Methoden zurückgegriffen wird. Dem Benutzer wird daraufhin eine nach dem Grad der Ähnlichkeit geordnete Liste von Teilen ausgegeben.

Der theoretische Vorteil dieser Methode besteht zum einem darin, dass zu keiner Zeit Merkmale definiert, besetzt oder geändert werden müssen, zum anderen kann ein Bauteil sowohl automatisch gesucht als auch klassiert werden.

Aufgrund der Probleme bei der Ermittlung von Fourierdeskriptoren für Volumina ist diese Methode zur Zeit allerdings nur bei ebenen Konturen sinnvoll anwendbar. Desweiteren macht es keinen Sinn, Baugruppen über die jeweilige Kontur zu beschreiben, weshalb sich die Anwendung auf Einzelteile in Form von Blechteilen und einfachen Drehteile (ohne Flächenbearbeitung) beschränkt, was als grösster Nachteile dieses Verfahrens anzusehen ist.

## 7 Zusammenfassung

Das Zusammenfassen von Objekten innerhalb der Produktentwicklung anhand konstruktiver, funktioneller oder fertigungstechnischer Eigenschaften zu Objektgruppen definierter Ähnlichkeit heisst Klassifikation.

Stetig ansteigendes Datenvolumen macht die Möglichkeit, effizient auf Informationen zugreifen zu können, unverzichtbar. Aus dieser Überlegung heraus folgt die Forderung, vorhandene Daten zu strukturieren, z.B. mittels Objektklassifizierung. Dadurch gelangt man in den Genuss der Vorzüge einer Wiederholteilverwendung. Diese sind in erster Linie Verringerung des Aufwandes einer Produktentwicklung in verschiedenen Bereichen der Planung, aber auch der Umsetzung. Ein Klassifizierungssystem soll folgende Eigenschaften aufweisen:

- eindeutige Zugriffsmechanismen
- Definition anwendungsspezifischer Zugriffsmechanismen erlaubt
- flexibel erweiterbar
- Zugriff auf gesamtes Objektspektrum möglich
- in PDM-Systeme integrierbar

Merkmale beschreiben Objekte. Es handelt sich dabei um nicht veränderbare Kennzeichen, die zur Charakterisierung eines Objektes verschiedene Ausprägungen annehmen können. Unter eine Eigenschaft verstehen wir die Verknüpfung von Merkmal und Ausprägung. Mittels Eigenschaften werden Objekte beschrieben und zu Objektgruppen definierter Ähnlichkeit zusammengefasst.

Es existieren verschiedene Verfahren für die Gruppenbildung. Diese werden zum Teil in unterschiedlicher Art und Weise genutzt und können untereinander kombiniert werden. Wichtige Verfahren der Gruppentechnik sind:

- Klassifikationsschlüssel
- Sachmerkmalleisten
- Begriffssysteme
- Generische Objektsysteme
- Clusteranalyse
- Fourierdeskriptoren

Die Wahl eines Verfahrens hängt vom Anwendungsbereich ab.

### Verständnisfrage 1

Worin besteht der Unterschied zwischen generischen und statischen Klassifizierungssystemen?

### Verständnisfrage 2

Wofür werden diese jeweils angewendet?

### Verständnisfrage 3

Warum ist die Benennung für einen effektiven Informationszugriff nur sehr eingeschränkt geeignet?

### Verständnisfrage 4

Was versteht man unter den Begriffen?

- Merkmal
- implizites Merkmal
- explizites Merkmal
- Ausprägung
- Eigenschaft
- Skalierung

### Verständnisfrage 5

Was für Zusammenhänge bestehen zwischen diesen Begriffen?

### Verständnisfrage 6

Welche Verfahren der Gruppentechnik gibt es und welche Methoden basieren auf diesen?

### Verständnisfrage 7

Welche Arten von Begriffsbeziehungen gibt es, wo finden diese vornehmlich Anwendung?

### Verständnisfrage 8

Worin liegen die Vor- und Nachteile von Klassifikationsschlüssel?

### Verständnisfrage 9

Wo werden Klassifikationsschlüssel heute vorwiegend angewendet?

### Verständnisfrage 10

Was versteht man unter einer Sachmerkmalreihe?

### Verständnisfrage 11

Wo werden Sachmerkmalreihen angewendet?

**Verständnisfrage 12**

Worin liegt der Unterschied zwischen der Modellierung und Zielsetzung von Sachmerkmaleisten in einem PDM-System und einer CAD-Normteillbibliothek?

**Verständnisfrage 13**

Mit welcher Methode der Gruppentechnik werden Sachmerkmaleisten oftmals kombiniert und aus welchem Grund?

**Verständnisfrage 14**

Auf welchem Paradigma basieren generische Objektsysteme

**Verständnisfrage 15**

Worin besteht der Unterschied zwischen objektzentrierten und objektorientierten Klassifikationssystemen?

**Verständnisfrage 16**

Was sind die grundsätzlichen Funktionsprinzipien der Clusteranalyse?

**Verständnisfrage 17**

In welchen betrieblichen Funktionsbereichen findet die Clusteranalyse Anwendung?

**Verständnisfrage 18**

Worin besteht die Zielsetzung der Faktorenanalyse?

**Verständnisfrage 19**

Mit welcher Methode der Gruppentechnik wird diese kombiniert?

**Verständnisfrage 20**

Worin unterscheidet sich die Fuzzy-Set basierte Klassifizierung von der Clusteranalyse?

**Verständnisfrage 21**

Worin liegt der Vorteil?

**Verständnisfrage 22**

Worauf beruht die Klassifizierung auf Basis von Fourierdeskriptoren?

### Verständnisfrage 23

Worin liegt der Vorteil - worin die Probleme dieser Methode?

#### Publikationsverzeichnis – Literatur

- [Bra-86] *Brauchle, H.*: Norm- und Wiederholteilsuche bei flexibler Produktion noch interessant? DIN-Mitteilungen 65 (1986) 1, S.25-31. Berlin: Beuth Verlag 1986
- [HiT-71] *Hillman, H.; Traebert, W.E.*: Ordnungsmerkmale und ihre Problematik; Beitrag und Diskussion um DIN 2330. DIN-Mitteilungen 50 (1971) 8, S.347-349. Berlin: Beuth Verlag 1971
- [DIN 1463] *Deutsches Institut für Normung*: DIN 1463 Richtlinien für die Erstellung und Weiterentwicklung von Thesauri. Berlin: Beuth Verlag 1987
- [DIN-2330] *Deutsches Institut für Normung*: DIN 2230 Begriffe und Benennungen. Berlin: Beuth Verlag 1979
- [DIN-2331] *Deutsches Institut für Normung*: DIN 2231 Begriffssysteme und ihre Darstellung. Berlin: Beuth Verlag 1979
- [Opi-66] *Opitz, H.*: Werkstückbeschreibendes Klassifizierungssystem. Essen: Giradet Verlag 1966
- [DIN-4000] *Deutsches Institut für Normung*: DIN 4000 Sachmerkmal-Leisten, Begriffe und Grundsätze. Berlin: Beuth Verlag 1979
- [FrG-82] *Freist, C., Granow, R.*: Ähnlichkeitsuche mit der Clusteranalyse. VDI-Zeitschrift 124(1996) 11 S.921-927 und 13 S.487-495
- [Übe-71] *Überla, K.*: Faktorenanalyse 2.Aufl., Berlin: Springer Verlag 1971
- [Cos-60] *Cosgriff, R.L.*: Identification of Shapes. Ohio State Univ. Res. Found. Columbus, Rep. No.820-11, ASIA (AD254792), 1960
- [GrV-92] *Grabowski, H.; Vogel, H.*: Klassifizieren, Suchen und Ordnen von geometrischen Informationen durch automatische Klassifikation. Konstruktion 44 (1992) 9 S.286-290.





# Objektidentifikation mit Sachnummernsystemen

Autor: Dr. R. Montau



# 1 Überblick

In der betrieblichen Praxis stellen Nummerungen ein wichtiges Hilfsmittel zur Beschreibung produktionstechnischer Daten- und Ablaufstrukturen dar. Die Notwendigkeit der Benummerung ist in der mangelhaften Eindeutigkeit der betrieblichen Begriffsbildung begründet. Für den Zugriff auf ein Objekt werden daher gewisse kennzeichnende Merkmale durch eine Nummer verschlüsselt, die somit eine Kurzbeschreibung darstellt und Aussagen über Klassierung, Variantenbildung, den Änderungszustand usw. enthalten kann.

Zur Benummerung existieren unzählige Vorschläge, die jedoch allesamt keinen begründeten Anspruch auf Allgemeingültigkeit und Ausschliesslichkeit besitzen. Mit Ausnahme einiger grundsätzlicher Prinzipien der Nummerungstechnik - die es in jedem Falle zu beachten gilt - lässt sich die Fragestellung nach dem bestgeeigneten Nummernsystem nicht generell, sondern nur für den jeweiligen Anwendungsfall unter Beachtung der betriebspezifischen Gegebenheiten und Zielsetzungen beantworten.

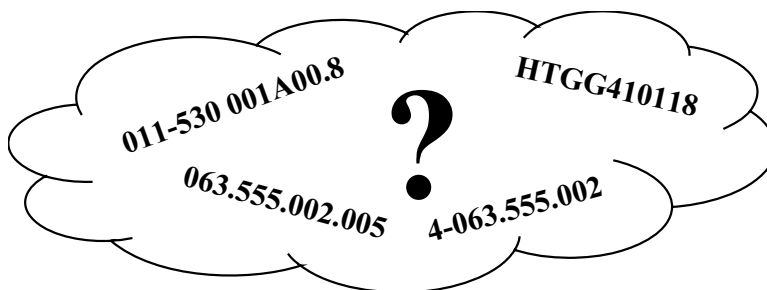


Bild (B001oidZ) Beispiele von Sachnummernsystemen

Mit zunehmender EDV-Durchdringung und insbesondere für die Einführung der PLM-Technologie in Unternehmen müssen historisch gewachsene Nummerungen kritisch analysiert und wiederholt auf ihre Eignung hinsichtlich neuer Anforderungen überprüft werden. Dem seit Schlaffner [Schl-63] allseits propagierten Ruf nach systemfreier Parallelbenummerung kann vielerorts in der Industrie aufgrund organisatorischer Randbedingungen [Dir-73] nicht nachgekommen werden. Ein grundsätzlicher Verzicht auf die mit ca. 75% [EHS-91] weitverbreiteten Verbundnummernsysteme, der nach Möglichkeit zwar immer angeraten werden sollte, ist somit undenkbar.

Bei einer PLM-Einführung muss daher die Benummerungsproblematik von Fall zu Fall immer sorgfältigst analysiert werden, wofür eine detaillierte Kenntnis dieser Thematik erforderlich ist, die nachstehend anhand folgender Gesichtspunkte erarbeitet werden soll:

- *Konzepte der Nummerungstechnik*  
Zunächst erfolgt eine Analyse der grundlegenden Konzepte der Nummerungstechnik, auf die sich sämtliche heute bekannten Benummerungsmethoden zurückführen lassen.
- *Charakterisierung von Sachnummernsystemen*  
Aufbauend auf dem erarbeiteten Grundverständnis können die Vorgehensweise zur Charakterisierung von Sachnummernsystemen und insbesondere die wesentlichen Beurteilungskriterien diskutiert werden.
- *Objektidentifikation in PLM-Systemen*  
Nachdem die anwendungsseitige Benummerung geklärt ist, soll auf die systemtechnische Identifikation in PLM-Systemen eingegangen werden.

## 2 Konzepte der Nummerungstechnik

Die verschiedenen Methoden der Nummerungstechnik lassen sich auf einige grundlegende Konzepte reduzieren, die nachfolgend im einzelnen detailliert betrachtet werden:

- Aufgaben einer Nummer,
- Aufbauprinzipien,
- Wertebereichsbildung,
- Kombinationsprinzipien.

### 2.1 Aufgaben einer Nummer

Bei der *Benummerung* betrieblicher Objekte (z.B. Gegenstände oder Unterlagen) stehen zwei wesentliche Aufgaben im Vordergrund:

1. identifizieren,
2. klassifizieren.

Das *Identifizieren* dient dem eindeutigen und unverwechselbaren Zugriff auf ein Objekt (z.B. Telefonnummer). Das *Klassifizieren* dient dem Ordnen von Objekten in Gruppen und ermöglicht somit deren gesamthafte Identifikation (z.B. Abteilungsnummer).

Neben diesen beiden Hauptfunktionen kann eine Nummer noch zwei weitere Aufgaben erfüllen, die aufgrund ihres geringeren Stellenwertes häufig als Nebenfunktionen bezeichnet werden:

3. informieren,
4. kontrollieren.

Unter dem *Informieren* einer Nummer versteht man eine Aussage anhand unverschlüsselter Merkmale („sprechende Nummern“ wie z.B. DIN-A4 bei Papierformaten). *Kontrollnummern* dienen der Erkennung richtiger Sachverhalte und werden in der Regel über Prüfzeichen realisiert.

In Anlehnung an diese Aufgaben lassen sich entsprechende Nummernarten unterscheiden, wobei insbesondere die Unterscheidung zwischen Identifikations- und Klassifikationsnummern zu beachten ist.

Die *Identifikationsnummer* (kurz Identnummer) als die wichtigste Nummer eines Objekts unterliegt in Anlehnung an [KuW-81, Gru-89] folgenden Anforderungen:

- *Eindeutigkeit*:  
Jedes Objekt muss genau eine Identnummer besitzen (Eineindeutigkeit!). Kein Objekt darf über verschiedene Identnummern ansprechbar sein; verschiedene Objekte können nicht dieselbe Identnummer besitzen.

- *Unveränderlichkeit:*  
Die Identnummer muss über die gesamte Lebensdauer des Objekts konstant, d.h. unveränderlich bleiben und dieses somit „überleben“.
- *Vollständigkeit:*  
Jedes Objekt, das ansprechbar sein soll, muss benummert werden, d.h. eine Identnummer erhalten.
- *Handhabbarkeit:*  
Zur Minimierung der Fehleranfälligkeit empfehlen sich eine konstante Stellenanzahl sowie die Anwendung von Gliederungsmitteln.

Diese Anforderungen lassen sich in idealer Weise durch Zählnummern erfüllen, welche die kürzestmögliche Stellenanzahl sowie eine unbegrenzte Lebensdauer aufweisen, da sie keine zeitabhängigen Bestandteile besitzen.

Im Gegensatz zur Identnummer, die Einzelobjekte identifiziert, dient eine *Klassifikationsnummer* der Identifizierung einer Gruppe! Reine Klassifikationsnummern müssen nur einmal gespeichert werden, weshalb sie weniger kritisch sind als Identnummern, die zur Referenzierung gegebenenfalls an mehreren Stellen redundant gehalten werden müssen. Für den Benutzer stellen Klassifikationsnummern jedoch meist eine unverzichtbare Informationsquelle dar, die folgenden Anforderungen unterliegt:

- *Nicht-Identifikation:*  
Klassifizierungsnummern dürfen in bezug auf ein Einzelobjekt niemals identifizierende Funktionalität besitzen!
- *Ausschliesslichkeit:*  
Um eine eindeutige Zuordnung zu gewährleisten, müssen sich die einzelnen Gruppen gegenseitig ausschliessen (Disjunktheit).
- *Anpassungsfähigkeit:*  
Für allfällige zukünftige Erweiterungen muss die Klassifizierung ausreichende Nummernreserven besitzen (Gefahr des Platzens).
- *Systematik:*  
Anzustreben ist ein systematischer und übersichtlicher Aufbau.
- *Feinheit der Gliederung*  
Aus Gründen der Änderungsanfälligkeit darf diese nur so fein als nötig gewählt werden.
- *Eindimensionalität:*  
Innerhalb einer Nummernstelle darf immer nur ein Merkmal verschlüsselt werden, um die Ausschliesslichkeit nicht zu gefährden.

- *Abhängigkeitsbewahrung:*  
Die Verschlüsselung muss den bestehenden Abhängigkeiten der Klassifizierungsmerkmale Rechnung tragen und diese korrekt abbilden.
- *Zweckmässigkeit:*  
Entsprechend den jeweiligen Zielsetzungen sind nur für diejenigen Begriffe Gruppen zu bilden, die separat abgefragt werden sollen.
- *Handhabbarkeit:*  
Neben einer konstanten Gliederung und Stellenanzahl sowie Gliederungsmitteln sind auch mnemotechnische Begriffe zu empfehlen.

Diese Anforderungen an eine Klassifizierungsnummer lassen sich durch verschiedene Nummern erfüllen. Im Gegensatz zur Identnummer kann daher keine allgemeine Empfehlung für die bestgeeignete Nummer ausgesprochen werden, da Klassifizierungen stark von der verfolgten Zielsetzung und dem dafür eingeräumten Aufwand abhängig sind.

Ein Vergleich der Anforderungen von Klassifizierungs- und Identnummern ergibt als prinzipielle Unterscheidungsmerkmale bzw. Widersprüche:

- *Eindeutigkeit gegenüber Feinheit:*  
Die Klassifikation ist vom Umfang und Detaillierungsgrad abhängig. Zwei in einer Gruppe zusammengefasste Objekte sind lediglich in den festgelegten Klassenmerkmalen ähnlich.
- *Unveränderlichkeit gegenüber Anpassungsfähigkeit:*  
Klassifizierungen sind von ihrem Wesen her stark änderungsgefährdet, da sich im Laufe der Zeit die Ordnungsmerkmale, deren gegenseitige Beziehung und Stellung zueinander oder deren Interpretation ändern können.
- *Eindeutigkeit gegenüber Nicht-Identifikation:*  
Betrachtet man ein Einzelobjekt, so stehen die beiden Hauptaufgaben einer Nummer, Identifikation (trennen) und Klassifikation (zusammenfassen), in funktionellem Gegensatz zueinander.

Das Beispiel einer Bibliothek in Bild B002oidZ verdeutlicht diesen funktionellen Gegensatz, dem eine grundlegende Bedeutung zukommt. Die Funktion des Identifizierens dient dem Trennen von Einzelobjekten, um ein jedes Objekt separat ansprechen zu können. Die Funktion des Klassifizierens besteht dagegen im Zusammenfassen von Objekten zu Gruppen auf der Basis gewisser Gemeinsamkeiten. Eine jede Gruppierung ist dabei prinzipiell änderungsanfällig, da sich die Kriterien der



Gruppeneinteilung und/oder die Eigenschaften eines Objektes mit der Zeit verändern können. Weitere Probleme können aus Mehrfachzuordnungen entstehen (vgl. Bsp. CAD/PLM-Literatur).

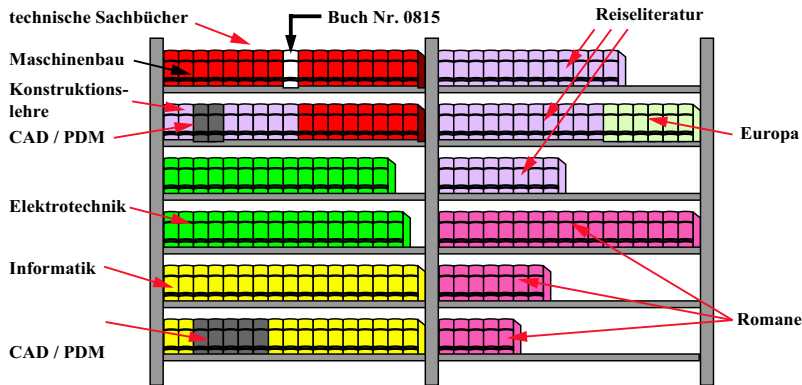


Bild (B002oidZ) Gegensätzlichkeit von Identifikation und Klassifikation

Eine Identnummer kann daher zwar zusätzlich auch klassifizierend sein - Klassifikationsnummern hingegen dürfen niemals identifizierende Funktionalität besitzen! Ansonsten könnte immer nur ein Einzelobjekt und niemals eine Gruppe angesprochen werden (vgl. Tabelle T001oidZ).

Aufgaben	Nummernarten	
	Ident-Nr.	Klassifikations-Nr.
identifizieren	x	keinesfalls
klassifizieren	eventuell	x

Tabelle (T001oidZ) Unterscheidung der Hauptfunktionen

## 2.2 Aufbauprinzipien

Die Diskussion der Aufbauprinzipien erfordert die Beachtung der nummerungstechnischen Begriffsdefinitionen nach DIN 6763. Unter einer *Nummer* ist eine nach bestimmten Regeln gebildete Zeichenfolge zu verstehen, welche aus mehreren Teilen aufgebaut sein kann. Mit *Nummernteil* bezeichnet man aufeinanderfolgende, logisch zusammengehörige Nummernstellen. Als kleinste Einheit einer Nummer sind diese entweder Daten- oder Gliederungsstellen in Form von Ziffern, Buchstaben, Sonderzeichen oder Leerstellen (Bild B003oidZ).

An dieser Stelle sei darauf hingewiesen, dass Gliederungsstellen nicht immer - wie in Bild B002oidZ dargestellt - nur zwischen Nummerteilen angeordnet werden dürfen. Aus Gründen der Übersichtlichkeit können diese durchaus auch innerhalb eines Nummerteiles geführt werden, was jedoch an dessen logischer Zusammengehörigkeit nichts ändert (z.B. Punkt in der Mitte der sechststelligen Zählnummer).

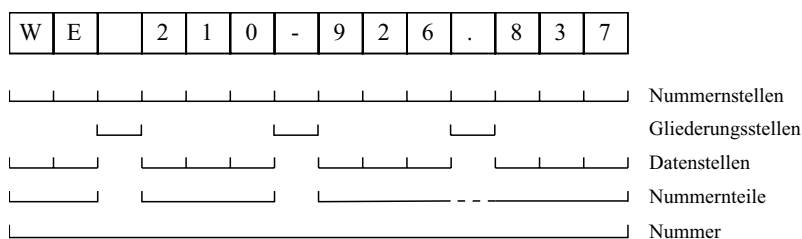


Bild (B003oidZ) Inhalt einer Nummer

Die folgenden Betrachtungen zum Nummernaufbau basieren auf der Einheit Nummerteil. Aufgrund der logischen Zusammengehörigkeit der Nummernstellen darf die Anzahl dieser Stellen keinen Einfluss auf den Aufbau haben. Infolgedessen können die Aufbauprinzipien anhand von einstelligen Nummerteilen erläutert werden, wobei zu unterscheiden ist zwischen dem

- Aufbau von Nummern (zwischen Nummerteilen) und dem
- Aufbau innerhalb von Nummerteilen (Wertebereichsbildung; vgl. nächster Abschnitt).

Der Nummernaufbau charakterisiert die Art und Weise der Zusammensetzung einer Nummer aus mehreren Nummernteilen. Hinsichtlich der Verbindung *zwischen Nummernteilen* unterscheidet man seit Schlaffner [Schl-63] folgende beiden grundsätzlichen Aufbauprinzipien (Bild B004oidZ):

- *hierarchischer Aufbau:*  
Eine Nummernstelle steht in eindeutiger Abhängigkeit von der vorhergehenden Stelle. Eine Aussage bezüglich dem untergeordneten Merkmal ist nur im Zusammenhang mit der vorhergehenden Stelle möglich.
- *unabhängiger Aufbau:*  
Jede Nummernstelle hat ihre eigene Bedeutung. Für Abfragen sind beliebige Kombinationen der Merkmale möglich.

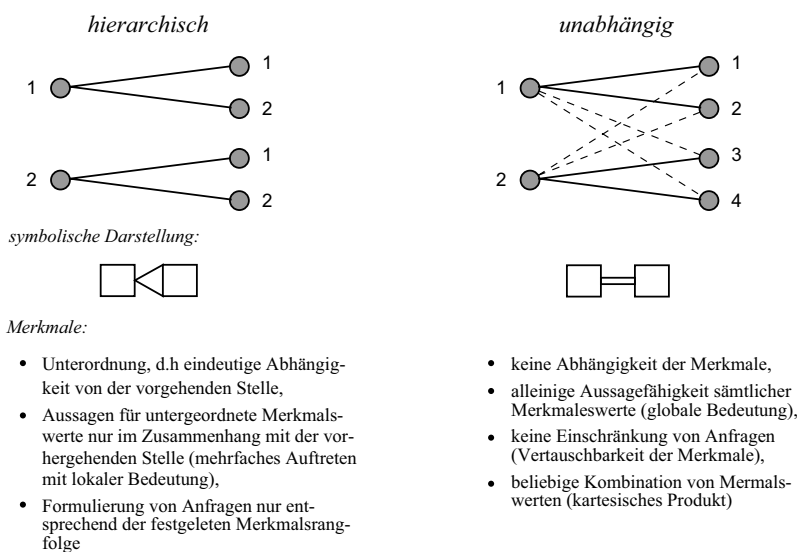


Bild (B004oidZ) Aufbauprinzipien von Nummernsystemen

Die Entscheidung, welche dieser beiden Aufbaumöglichkeiten in einem bestimmten Anwendungsfall am besten geeignet ist, muss jeweils unter Beachtung der bestehenden Abhängigkeiten getroffen werden. Bei Ranggleichheit von Ordnungsmerkmalen ist ein unabhängiger Aufbau zu wählen (z.B. Variantenverschlüsselung bei KFZ) - bei Abhängigkeit ist ein hierarchischer Aufbau vorzuziehen (z.B. Vorwahl bei Telefonnummern).

Hervorzuheben ist, dass die Verwendung eines hierarchischen Aufbaus immer gleichbedeutend ist mit einem Verzicht auf Flexibilität, da

die Rangfolge der Merkmale damit für sämtliche Abfragen unabänderlich vorgegeben ist. Ein hierarchischer Aufbau darf daher nur dann verwendet werden, wenn mit Sicherheit eine eindeutige Abhängigkeit zwischen den Merkmalen besteht und diese Rangfolge auch in Zukunft Bestand hat [Schl-63].

Ist aber, wie z.B. bei Telefonnummern, eine solche Abhängigkeit zwischen zwei Merkmalen mit Sicherheit gegeben, so kann und sollte sie auch auf die Verschlüsselung abgebildet werden. Da eine Telefonnummer ohne Kenntnis der Vorwahl nutzlos ist, sollte hierfür ein hierarchischer Aufbau gewählt werden.

### 2.3 Wertebereichsbildung

Das Prinzip der Wertebereichsbildung dient zur Strukturierung *innerhalb eines Nummernteils*. Die Gruppierung bezüglich eines Merkmals (Klassifikation) erfolgt über die Zuordnung von Eigenschaften zu Wertebereichen einer Nummer, die jeweils durch ihre Anfangs- und Endwerte festgelegt werden (vgl. DIN 6763). Die Bereichsbildung in Nummernteilen lässt sich charakterisieren in:

- *bereichsfrei*  
(rein zählender oder kategorischer Aufbau),
- *eindimensional bereichsweise*  
(Beschränkung auf ein verschlüsseltes Merkmal),
- *bereichsweise disjunkt*  
(mehrdimensional mit Ausschliesslichkeit bei der Gruppenzuordnung),
- *bereichsweise*  
(mehrdimensional mit nicht orthogonalen Unterscheidungsmerkmalen).

Als Beispiel soll die Unterteilung des Wertebereichs einer dreistelligen Zählnummer in die Bereiche 001-299 für zugekaufte Halbzeuge, 300-499 für zugekaufte Baugruppen und 500-999 für Eigenfertigungsteile dienen. Da hierdurch die Merkmale Beschaffungsart und Teileanzahl verschlüsselt werden, handelt es sich um eine mehrdimensionale, disjunkte Wertebereichsbildung. Falls sich verschlüsselte Eigenschaften jedoch gegenseitig beeinflussen, kann durch eine Mehrdimensionalität die Ausschliesslichkeit bei der Gruppenzuordnung beeinträchtigt werden.

Da sich zunehmende Bereichsbildung und Mehrdimensionalität nachteilig auf Unveränderlichkeit und Anpassungsfähigkeit einer Nummer auswirken, ist vom Prinzip der Bereichsbildung nach Möglichkeit abzuraten. Anwendung findet die Wertebereichsbildung häu-

fig zur dezentralen Nummernvergabe, wo einzelne Standorte unterschiedliche Wertebereiche (Standort A: 001-399, Standort B: 400-799 etc.) erhalten und diese z.B. anhand separater Nummernbücher autonom vergeben.

## 2.4 Kombinationsprinzipien

Ein Nummernsystem ist eine gegliederte Kombination von Nummern oder Nummernteilen aus meist unterschiedlichen Nummernarten. Nummernsysteme können daher gleichfalls identifizierende, klassifizierende, informierende und/oder kontrollierende Funktionalität besitzen. Bei der Kombination der Nummernteile wird in der industriellen Praxis sehr häufig eine Mischung von hierarchischem und unabhängigem Aufbau verwendet.

Von besonderer Bedeutung ist bei diesen Kombinationen das Zusammenspiel der Funktionen Identifikation und Klassifikation zwischen den einzelnen Nummernteilen. Nach Schläffner [Schl-63] sind die beiden folgenden Prinzipien zu unterscheiden (vgl. Beispiele in Bild B005oidZ):

- *Verbundnummernsystem:*  
Zur Identifizierung werden Ident- und Klassifikationsnummer benötigt.
- *Parallelnummernsystem:*  
Identifizierung und Klassifizierung erfolgen unabhängig voneinander. Die Identnummer enthält keine (!) klassifizierenden Merkmale.

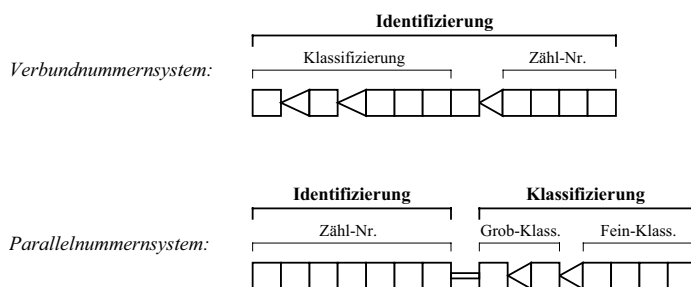


Bild (B005oidZ) Beispielhafte Verdeutlichung der Kombinationsprinzipien

*Verbundnummernsysteme* wurden in der Zeit der ausschliesslich manuellen Verwaltung von Datenbeständen, z.B. in Ordnern oder

Karteien, eingeführt und sind in der Industrie noch immer zu etwa 75% verbreitet [EHS-91]. Zur besseren Überschaubarkeit und für die Verarbeitung grösserer Datenmengen mussten in der Identifizierung unbedingt klassifizierende Angaben mitgeführt werden. Aufgrund ihrer grösseren Aussagefähigkeit werden Verbundnummernsysteme auch heute noch vom Benutzer bevorzugt.

Im Zuge der EDV-Durchdringung haben sich die Anforderungen an Nummernsysteme geändert und die klassifizierenden Aufgaben sind gegenüber der Identifikation mehr und mehr zurückgetreten. Unabhängige Grundvoraussetzung für den Einsatz von Computer- und Datenbanksystemen ist die Eindeutigkeit der Identifizierung. Aufgrund der Änderungsanfälligkeit klassifizierender Aussagen ist der Einsatz von Verbundnummernsystemen generell risikobehaftet. Sind in der Klassifikation lediglich zeitlich konstante Merkmale verschlüsselt (z.B.: Geburtsdatum), so ist dies weniger kritisch als wenn diese zudem zeitveränderlich sind (z.B.: Marktpreis), woraus sich verheerende Folgen ergeben können. Als vordringliches Beurteilungskriterium ist daher die Zeitabhängigkeit der in der Identifizierung mitgeführten klassifizierenden Merkmale anzusehen.

*Parallelnummernsysteme* mit einem systemfreien, rein zählenden Aufbau der Identifizierung sind dagegen für den Computereinsatz prädestiniert, da sie die Vorteile einer Zählnummer zur Identifizierung aufweisen und zudem parallelverschlüsselt klassifizierende Aussagen ermöglichen. Aufgrund der fehlenden Aussagefähigkeit der Identnummer werden Parallelnummernsysteme vom Benutzer jedoch häufig als ungünstiger in der Handhabung empfunden, worin sich ihre geringere Akzeptanz begründet.

Die Unterscheidung in Verbund- und Parallelnummernsysteme ist von herausragender Bedeutung für die Charakterisierung einer Nummer. Die Vor- und Nachteile beider Kombinationsprinzipien werden daher in der einschlägigen Literatur [KuW-81, Gru-89, Opi-71, Jah-79, BeB-85, VDI-72] ausführlich diskutiert (vgl. Tabelle T002oidZ).

	Verbundnummernsystem	Parallelnummernsystem
Charakterisierung	Identifizierung aus der Klassifizierung heraus	Trennung von Identifizierung und Klassifizierung
Vorteil	<ul style="list-style-type: none"> <li>• Aussagefähigkeit (halbsprechende Nummer)</li> <li>• Erkennbarkeit der Klassifizierung</li> <li>• dezentrale Nummernvergabe möglich</li> <li>• gute Merkbarkeit</li> </ul>	<ul style="list-style-type: none"> <li>• Eindeutigkeit der Identnummer</li> <li>• Unveränderlichkeit der Identnummer bei Änderung der Klassifikation</li> <li>• kürzest mögliche Identnummer, die selbständig verwendet werden kann</li> <li>• Anpassungsfähigkeit der Klassifikation</li> <li>• einheitliche Benummerung</li> <li>• datenverarbeitungsgerecht</li> <li>• geringe Fehleranfälligkeit</li> </ul>
Nachteil	<ul style="list-style-type: none"> <li>• Änderung der Klassifikation beeinflusst Identifikation (keine Unveränderlichkeit)</li> <li>• Gefahr des Platzens</li> <li>• Einschränkung der Anpassungsfähigkeit aufgrund fehlender Flexibilität der Klassifikation</li> <li>• längere Stellenanzahl der Identnummer</li> </ul>	<ul style="list-style-type: none"> <li>• hoher Einführungsaufwand</li> <li>• grössere Gesamtstellenanzahl</li> <li>• fehlende Aussagefähigkeit und dadurch geringere Benutzerakzeptanz</li> </ul>

Tabelle (T002oidZ) Gegenüberstellung von Verbund- und Parallelnummernsystem

Anzumerken ist, dass hier für Parallelnummernsysteme die strenge Definition nach Schlaffner Verwendung findet, der in seinen Arbeiten erstmals die grundsätzlichen Unterschiede beider Kombinationsprinzipien herausgearbeitet hat [Schl-63]. Schlaffner und beispielsweise auch [KuW-81] fordern ausdrücklich eine systemfreie Identnummer ohne jedwelche klassifizierenden Angaben als qualifizierendes Merkmal für ein Parallelnummernsystem!

Die grundlegenden Konzepte der Nummerungstechnik sind damit in allgemeiner Form diskutiert. Nachfolgend kann nun auf deren zweckgebundene Anwendung im Bereich der Sachnummerung eingegangen werden.

## 3 Sachnummerierung

Eine der wichtigsten Anwendungen von Nummernsystemen stellt der Bereich der Sachnummerierung dar. Nach DIN 6763 ist die *Sachnummer* eine Identnummer und muss damit in jedem Fall identifizieren. Darüber hinaus kann sie jedoch noch weitere Funktionalität aufweisen - z.B. klassifizieren (vgl. Tabelle T001oidZ).

Da die Sachnummerierung einerseits ein breites Objektspektrum umfasst (Gegenstände und Unterlagen) und zum anderen verschiedene Unternehmensbereiche betrifft, stellt sie immer einen Kompromiss unterschiedlicher Zielsetzungen dar. Diese Zielsetzungen bestehen jeweils aus den Anforderungen an eine Identnummer. Darüber hinaus sind in Abhängigkeit der Aufgabengewichtung auch die Anforderungen an Klassifizierungsnummern (mit Ausnahme der Nicht-Identifikation) sowie weitere anwendungsbezogene Gesichtspunkte zu berücksichtigen.

### 3.1 Anwendungsbezogene Klassifizierungsmerkmale von Sachnummern

Der Anwendungsbereich einer Nummerung hat grossen Einfluss auf die bestehenden Anforderungen. Ein Nummernsystem ist deshalb niemals für alle denkbaren Anwendungsfälle gleichsam geeignet. Dies mag einer der Gründe sein, warum der Begriff Nummernsystem, im Gegensatz zu den Nummernarten, anwendungsbezogen ist (z.B. Sach- oder Personalnummernsystem).

Im Bereich der Sachnummerierung existieren eine Reihe anwendungsbezogener Merkmale, deren Verschlüsselung entscheidenden Einfluss auf Funktionalität und Auswirkung der Benummerung auf Produktionsabläufe und Datenbestände hat. Um dies zu verdeutlichen, sollen im folgenden die wichtigsten anwendungsbezogenen Merkmale von Sachnummern analysiert werden:

- Zuordnung von Nummerungsobjekten,
- Verwendungsabhängigkeit,
- Strukturabhängigkeit,
- Änderungshandhabung.

Anhand der gegenseitigen *Zuordnung von Nummerungsobjekten* lässt sich die Problematik durch die Verschlüsselung anwendungsbezogener Gesichtspunkte aufzeigen. In der Praxis wird diese Zuordnung sehr häufig durch das Prinzip der Gleichheit der Nummer von Sache und Unterlage realisiert. Die einen beliebigen Gegenstand beschreibenden Zeichnungen und sonstigen Dokumente können dadurch



unter derselben Identnummer aufgefunden werden (vgl. [VDI-72, Gru-89]). Die einzelnen Unterlagenarten unterscheiden sich durch eine entsprechende zusätzliche Kennzeichnung (vgl. Bild B006oidZ).

Bei dieser Verfahrensweise steckt die Zuordnungsinformation bereits in der Benummerung, was vorteilhaft für die Benutzertransparenz ist und insbesondere bei geringer EDV-Verfügbarkeit angewendet werden muss. Von Nachteil ist jedoch die Einschränkung der Flexibilität, da Änderungen (von Sachnummer oder Revisionsindex) an mehreren Stellen nachgeführt werden müssen und insbesondere nur (1:n)-Beziehungen zwischen Bauteilen und Unterlagen abgebildet werden können.

Häufig werden an diese Zuordnung aber weitreichendere Anforderungen gestellt, wie etwa (n:m)-Beziehungen (z.B. für Tabellenzeichnungen) oder auch Unterlagenstrukturen. Aufgrund ihrer Flexibilität bieten hier die modernen Datenbanktechnologien grosse Vorteile, da Zuordnungsbeziehungen ohne jede Restriktion und Konsistenzgefahr definiert werden können. Für Abfragen können zahlreiche leistungsfähige Auswertungsfunktionen angeboten werden, wofür jedoch ein Online-Zugriff auf die Datenbank vorhanden sein muss.

Eine Zuordnung über identische Nummernwerte kann somit zwar Vorteile bringen, will jedoch sorgsam überdacht sein und darf nur bei entsprechender hierarchischer Beziehungskardinalität eingesetzt werden. Weitere Anwendungsfälle für eine solche Zuordnung können neben Dokumenten auch Fertigungsmittel oder Produktionsvarianten (z.B. verschiedene Farben) sein.

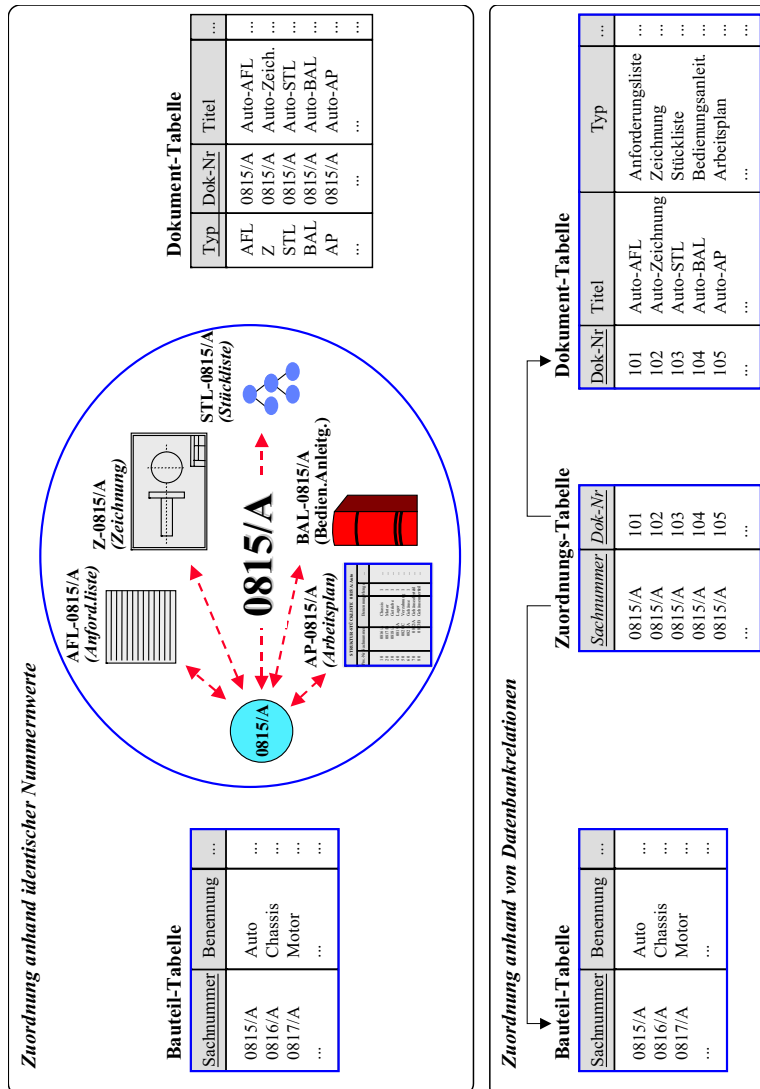


Bild (B006oidZ) Methoden der Zuordnung von Nummerungsobjekten

Unter der *Verwendungsabhängigkeit* einer Nummer versteht man eine Aussage über die Verwendung von Gegenständen in bestimmten Produkten. Hierbei werden alle Komponenten eines Produktes hierarchisch abhängig von der Produkt-Nr. benummert (z.B. 0815; vgl. Bild B007oidZ). Wie jede klassifizierende Aussage ist die Verwendungsabhängigkeit änderungsanfällig und darüber hinaus zeitlich nicht konstant. Die eigentliche Problematik der Verwendungsabhängigkeit liegt jedoch in der Beschränkung auf die Erstverwendung eines Bauteils begründet. Im Falle der Wiederverwendung (in einem anderen Produkt, z.B. 4711) kann nur die Erstverwendung angegeben werden

oder die Sache müsste doppelt benummert werden, wodurch jedoch die Forderung nach Eindeutigkeit verletzt würde.

Für die *Strukturabhängigkeit* gilt im Prinzip dasselbe wie für die Verwendungsabhängigkeit, da diese lediglich eine detailliertere Aussage über die Verwendung darstellt (vgl. Beispiel in Bild B007oidZ). Die Strukturabhängigkeit wird in der Literatur häufig auch mit Baugruppenabhängigkeit bezeichnet, was den Sachverhalt jedoch weniger trifft.

Wie bei der Verwendungsabhängigkeit kann nur die Erstverwendung eines Bauteils beschrieben werden. Bei hohem Wiederverwendungsgrad muss – hierbei schon im selben Produkt (!) - die gewünschte Nummernsystematik durchbrochen werden, was eine grosse Unübersichtlichkeit zur Folge hat und den Nutzen bereits in Frage stellt. Besonders kritisch ist jedoch die prinzipielle Platzgefahr auf jeder verschlüsselten Strukturebene, falls die vorgesehene Stellenanzahl (99 in Bild B007oidZ) nicht mehr ausreichen sollte.

Des weiteren ist nach DIN 6763 eine Strukturabhängigkeit für Sachnummern nicht erlaubt, da eine strikte Trennung von Sach- und Strukturinformationen gefordert wird! Ungeachtet dessen finden sich jedoch in der Praxis auch heute noch strukturabhängige Nummern insbesondere im Bereich der Zeichnungsnummerung, die auch als Sachnummern geführt werden (vgl. [BeB-85]).

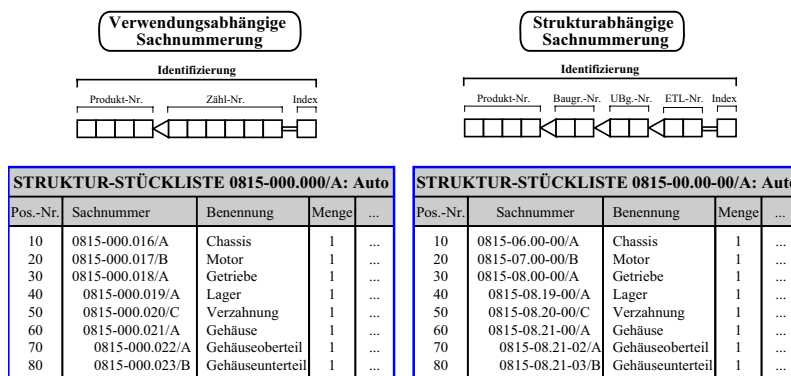


Bild (B007oidZ) Beispiele für Verwendungs- und Strukturabhängigkeit

Die *Änderungshandhabung* im Rahmen der Sachnummerung richtet sich nach dem Ausmass der Veränderung, d.h. nach der Austauschbarkeit der Objekte. Bei austauschbaren Änderung soll die Sachnummer beibehalten werden und lediglich eine Inkrementierung des Änderungsindex erfolgen. Demgegenüber muss bei nicht aus-

tauschbaren Objekten eine unterschiedliche Sachnummer vergeben werden.

Anzumerken ist, dass ein Änderungsindex strenggenommen nicht Bestandteil der Sachnummer sein darf! Im Falle einer austauschbaren Änderung würde ansonsten durch die Inkrementierung des Änderungsindexes unerwünschterweise auch die Sachnummer mitgeändert. Um die geforderte Benummerung in Abhängigkeit der Austauschbarkeit zu erreichen, muss daher anstelle des Änderungsindexes eine von der Sachnummer *separate Änderungsnummer* geführt werden.

Für die Datenhaltung ist dabei die identifizierende Funktionalität der Änderungskennzeichnung entscheidend, wobei folgende beiden prinzipiellen Möglichkeiten der Kennzeichnung bestehen:

aktiv	<i>Identifizierend):</i> Jede Änderung bedingt die Vergabe einer neuen Identifizierung, wodurch jeweils ein neuer Datensatz entsteht.
passiv	<i>(rein klassifizierend):</i> Es ist jeweils nur der neueste Änderungszustand gespeichert, wodurch alte Zustände nicht mehr greifbar sind.

Tabelle (T003nsZ) Möglichkeiten der Änderungskennzeichnung

Im letzteren Fall könnten somit frühere Revisionen nicht aufbewahrt werden, da der alte Datenbestand jeweils überschrieben wird. Aus Gründen der Produkthaftung ist es in der Regel jedoch ein Muss, sämtliche Änderungszustände einer Entwicklung zu dokumentieren (vgl. [DIN9000, DIN9001]), wofür eine *aktive Kennzeichnung* unabdingbar ist. Bild B008oidZ verdeutlicht den funktionellen Unterschied beider Methoden der Änderungskennzeichnung.

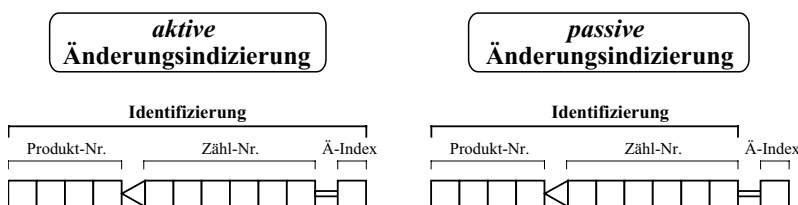


Bild (B008oidZ) Verdeutlichung der grundsätzlichen Methoden zur Änderungskennzeichnung

Weitere anwendungsbezogene Gesichtspunkte, wie z.B. eine Verschlüsselung der Variantenbildung sind stark vom jeweiligen Anwendungsfall abhängig und eignen sich daher weniger zu einer allgemeinen Charakterisierung, für welche die obigen Merkmale ausreichend und zweckmässig sind.

### 3.2 Charakterisierung von Sachnummernsystemen

Eine Beurteilung von Sachnummernsystemen sollte immer im konkreten Anwendungsfall erfolgen, um deren Eignung bezüglich den spezifisch bestehenden Anforderungen bestimmen zu können. Im allgemeinen kann jedoch bereits anhand der oben erläuterten, rein nummerungstechnischen Konzepte aus Abschnitt Konzepte der Nummerungstechnik zusammen mit den anwendungsbezogenen Klassifizierungsmerkmalen eine ausreichend Charakterisierung vorgenommen werden (Tabelle T002oidZ).

	Merkmale	Ausprägungen
nummerungs- technisch	Aufgaben	<ul style="list-style-type: none"> <li>• klassifizieren</li> <li>• informieren</li> <li>• kontrollieren</li> <li>• Kombination</li> </ul>
	Nummernaufbau	<ul style="list-style-type: none"> <li>• hierarchisch</li> <li>• unabhängig</li> <li>• Kombination</li> </ul>
	Werteberichtigung	<ul style="list-style-type: none"> <li>• keine</li> <li>• parallelverschlüsselt</li> <li>• identifizierend</li> </ul>
	Kombinationsprinzip	<ul style="list-style-type: none"> <li>• Verbundnummernsystem</li> <li>• Parallelnummernsystem</li> </ul>
anwendungs- bezogen	Verwendungsabhängigkeit	<ul style="list-style-type: none"> <li>• verwendungsunabhängig</li> <li>• parallelverschlüsselte Verwendungsabhängigkeit</li> <li>• Verwendungsabhängigkeit in der Identifizierung</li> </ul>
	Strukturabhängigkeit	<ul style="list-style-type: none"> <li>• strukturabhängig</li> <li>• parallelverschlüsselte Strukturabhängigkeit</li> <li>• Strukturabhängigkeit in der Identifizierung</li> </ul>
	Änderungskennzeichnung	<ul style="list-style-type: none"> <li>• aktiv</li> <li>• passiv</li> </ul>

Tabelle (T002oidZ) Charakterisierung von Sachnummernsystemen

Da eine Sachnummer definitionsgemäss eine Identnummer ist [DIN6763], kann die Identifizierung nur unabdingbare Voraussetzung und damit kein charakterisierendes Merkmal sein. In bezug auf die Wertebereichsbildung erscheint eine Grobunterscheidung nach dem Ausmass der Bereichsbildung in einem Nummernsystem zweckmässig. Wichtigstes Kriterium dieser nummerungstechnischen Charakterisierung ist das Kombinationsprinzip.

Die verbleibenden Merkmale (Verwendungsabhängigkeit, Strukturabhängigkeit und Änderungskennzeichnung) dienen der anwendungsbezogenen Beurteilung von Sachnummernsystemen.

Aufgrund der mangelhaften Beständigkeit der Verwendungs- und der Strukturabhängigkeit wird hier ebenfalls nach deren Ausmass unterschieden. Eine Strukturabhängigkeit in der Identifizierung kann dabei besonders schwerwiegende Probleme ergeben. Zur Beurteilung der Änderungshandhabung ist eine rein funktionale Unterscheidung in aktive oder passive Kennzeichnung zu treffen.

### 3.3 Beispielhafte Analyse eines Sachnummernsystems

Um die obigen Ausführungen besser veranschaulichen und verifizieren zu können, soll nachfolgend exemplarisch ein Sachnummernsystem aus der industriellen Praxis analysiert werden. Als Beispiel dient eine Nummerung aus einem Schweizer Grossunternehmen [SUL-89], die dort für Eigenfertigungs- und Zukaufteile Anwendung findet. Die Nummer beinhaltet eine Reihe praxisnaher Konsistenzprobleme und eignet sich daher gut zur Verdeutlichung anwendungsbezogener Zusammenhänge, wodurch Vorzüge und Grenzen der Verschlüsselung mit Nummernsystemen deutlich werden.

Die Nummerung besteht einerseits aus einem Sachnummernsystem zur Teileverwaltung und zum anderen aus einem Zeichnungsnummernsystem zur Verwaltung Technischer Zeichnungen. Der Bezug zwischen beiden Nummerungen wird über identische Nummernteile hergestellt, die eine Zuordnung ausdrücken. Die Änderungskennzeichnung erfolgt aktiv anhand einer separaten Änderungsnummer.

Die Sachnummer ist ein Verbundnummernsystem, bestehend aus drei voneinander hierarchisch abhängigen Nummernteilen, der Produktart, der Zählnummer sowie der Kennziffer (vgl. Bild B009oidZ). Mit den Angaben zu Produktart und Kennziffer besitzt die Sachnummer klassifizierende Funktionalität in der Identifizierung. Anhand der Produktart werden verwendungsabhängige Informationen verschlüs-

selt, wie z.B. der Produktbereich oder auch abteilungsbezogene Merkmale. Die Kennziffer hat zudem einen bereichsweisen Aufbau.

Die Zeichnungsnummer ist ebenfalls ein Verbundnummernsystem mit drei Nummernteilen, die einen gemischten Aufbau aufweisen (Bild B010oidZ). Format und Produktart sind parallelverschlüsselt, die Zählnummer ist dagegen von der Produktart hierarchisch abhängig. Sämtliche Nummernteile sind hier rein zählend aufgebaut, d.h. bereichsfrei definiert. Durch die Produktart beinhaltet die Zeichnungsnummer klassifizierende Funktionalität in der Identifizierung. Die Formatangabe ist zudem unverschlüsselt, d.h. informierend.

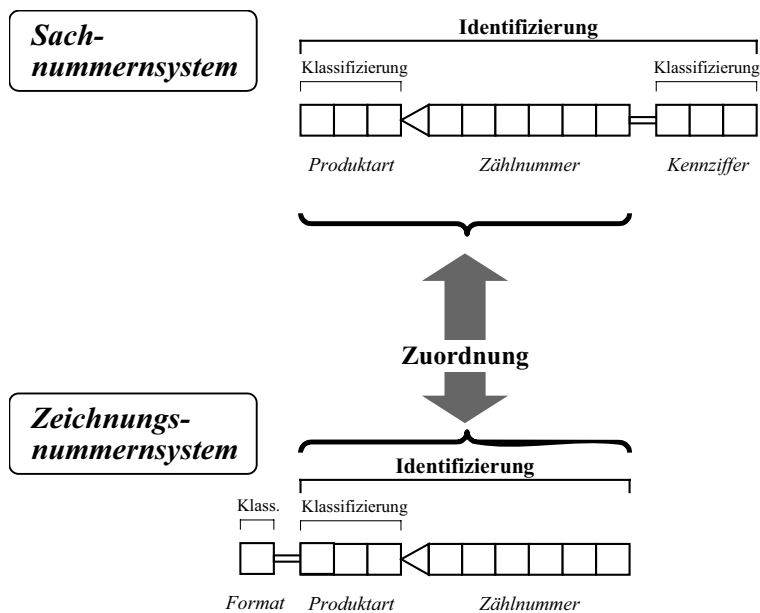


Bild (B009oidZ) Nummerungsbeispiel

001-199	Positionskennzahl
200-399	Ausführungskennzahl
400-449	Rohgusskennzahl
450-499	Rohteilkennzahl
500-599	Zeichnungskennzahl
600-799	Reserve
800-899	Dispositions-kennzahl

Tabelle (T003oidZ) Kennziffer-Wertbereichsbildung

Bei der *Benummerung eines Bauteils* müssen die klassifizierenden Angaben bzgl. Produktart und dem gewünschten Kennzifferbereich vom Benutzer angegeben werden. Daraufhin kann systemintern die Zählnummer in Abhängigkeit der Produktart sowie die hiervon abhängige Kennzahl generiert werden. Hierbei ist einerseits zu beachten, ob bereits eine Zeichnungsnummer für eine Technische Zeichnung des Bauteils vergeben ist. Falls ja, müssen deren Produktart und Zählnummer übernommen werden, um so die konsistente Zuordnung der Stammdaten zu den Unterlagen zu gewährleisten. Der Benutzer hat dadurch lediglich noch den gewünschten Kennzifferbereich zu spezifizieren. Zum anderen dürfen Bauteile, denen evtl. später Zeichnungen zugeordnet werden sollen, niemals dieselbe Zählnummer erhalten, da dies identische Zeichnungsnummern zur Folge hätte.

Bei der *Benummerung einer Technischen Zeichnung* wird die Formatzahl durch das Zeichnungsformat bestimmt und kann z.B. aus einem rechnerintern gespeicherten CAD-Modell abgeleitet werden. Der Benutzer hat lediglich die Produktart anzugeben, woraufhin die abhängige Zählnummer bestimmt werden kann. Falls das dargestellte Bauteil bereits eine Sachnummer besitzt, muss deren Produktart und Zählnummer auch für die Zeichnungsnummer Verwendung finden.

Die obige Analyse verdeutlicht, welche grossen Konsistenzprobleme bei Verbundnummernsystemen auftreten können. Insbesondere die Zuordnung von Sach- und Zeichnungsnummern durch identische Produktart- und Zählnummernwerte ist aufgrund der identifizierenden Funktionalität der Kennziffer und den daraus folgenden Restriktionen als überaus kritisch zu werten. Demgegenüber sind hinsichtlich der Aussagefähigkeit kaum nutzbringende klassifizierende



Merkmale vorhanden, da im digitalen Zeitalter eine Ablage nach Formatgrößen kaum mehr zweckmässig ist und die Kennziffer ebenfalls nur beschränkt hilfreich erscheint. Lediglich die klassifizierende Produktart erfüllt ihren Zweck und ermöglicht eine Grobklassifizierung der Produktpalette.

Nachdem die methodischen Aspekte der Nummerungstechnik hinreichend diskutiert sind, soll abschliessend auf deren Umsetzung und das Zusammenspiel mit der systemtechnischen Identifikation in PLM-Systemen eingegangen werden.

## 4 Objektidentifikation in PLM-Systemen

In Unternehmen wird die Identifikation betrieblicher Objekte (wie z.B. Bauteile oder Dokumente) durch Sachnummernsysteme sichergestellt, die dem Benutzer wohlverständlich sind und daher *die* zentrale Nummer in PLM- und ERP-Systemen darstellen.

Innerhalb eines DV-Systems müssen die Sachverhalte ebenfalls identifiziert werden. Im Falle einer systemfreien Parallelverschlüsselung kann die Identnummer direkt als Primärschlüssel der Datenbankobjekte Verwendung finden. Wird jedoch zur Sachnummerung eine Verbindnummer eingesetzt, so bedeutet dies zwangsläufig gewisse Einschränkungen bezüglich der Anforderungen an eine Identifikation (vgl. Kap. Aufgaben einer Nummer u. Kombinationsprinzipien). Für die Identifizierung innerhalb eines Datenbanksystems z.B. einer PLM-Applikation bestehen daher zwei grundsätzliche Möglichkeiten (vgl. Bild B010oidZ):

- Identifikation durch die Sachnummer,
- Identifikation durch eine interne Identnummer.

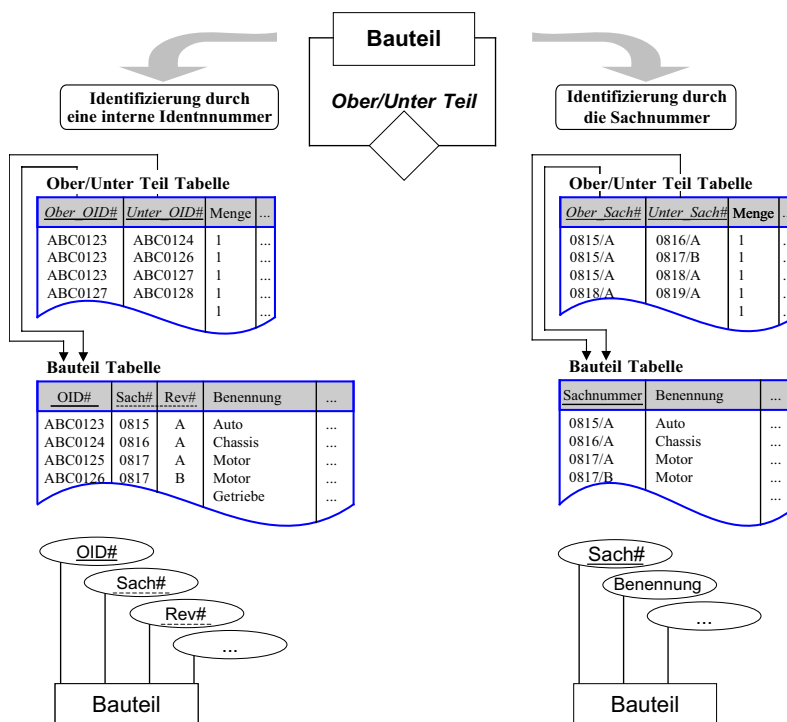


Bild (B010oidZ) Arten der Identifizierung

Da sämtliche klassifizierenden Angaben immer änderungsanfällig sind und ein Primärschlüssel definitionsgemäss aus einer minimalen Kombination von Attributen besteht, dürfen bei Nummernsystemen grundsätzlich nur die identifizierenden Nummernteile als Schlüssel Verwendung finden. Für Verbundnummern sind jedoch Eindeutigkeit und Unveränderlichkeit häufig nicht in ausreichendem Masse gewährleistet, so dass für den Primärschlüssel eine interne Identnummer herangezogen werden muss. Um jedwelche Probleme in Abhängigkeit kundenspezifischer Nummerungen von vorneherein ausschliessen zu können, verwenden heute in der Regel sämtliche PLM-Systeme eine rein systemseitig generierte interne Objektidentifizierung (genannt: Surrogat).

Würde die Sachnummer als Primärschlüssel verwendet, so könnte der Benutzer bei einem Dateneintrag direkt den Wert für diesen Schlüssel eingeben. Im anderen Fall muss die eingegebene Sachnummer in eine interne Identnummer (Surrogat B#) umgesetzt werden (z.B. durch Abfrage des momentan grössten Primärschlüsselwertes in der betreffenden Objekttable (hier: Bauteil) und Erhöhung desselben um 1).

Die interne Objektidentifizierung in Bild B011oidZ ist beispielhaft in der Form „ABC0123..“ ausgeführt. Dies soll andeuten, dass neben der einfachen Inkrementierung einer reinen Zählnummer auch komplexere Algorithmen angewendet werden, um beispielsweise eine weltweit eindeutige Objektidentifikation zu erhalten. Falls diese standort- bzw. vielmehr datenbankübergreifende Eindeutigkeit der OID# nicht sichergestellt ist, muss dies nachträglich vom Verteilungsmodul übernommen werden (vgl. Section 3.5).

Die Information über die Sachnummer dieses Objekts muss ebenfalls hinterlegt werden, damit es vom Benutzer mit dieser ihm bekannten Nummer angesprochen werden kann. Aufgrund der Verwendung einer internen Objekt ID als Primärschlüssel können Sachnummer und Revisionsindex wie ein gewöhnliches Attribut definiert werden. In der Regel wird dabei deren Eindeutigkeit mittels Unique-Definition auf das Datenbankschema übertragen (Uniqueness Constraint in Bild B011oidZ).

Aufgrund der Notwendigkeit einer systeminternen Identifizierung neben der Sachnummer muss somit zwischen der systemseitigen und der anwendungsseitigen Identifikation unterschieden werden, die unterschiedliche Zielsetzungen verfolgen (vgl. Bild B012oidZ).

*Systemseitig* wird jedes Objekt unter einem systeminternen Surrogat abgelegt, worunter eine algorithmisch generierte Identnummer ohne jede Semantik (Bedeutung) für den Anwender zu verstehen ist.

Da diese Identnummer (vorstellbar als reine Zählnummer) keinerlei interessierende (klassierende) Informationen beinhaltet und vom Benutzer i.d.R. verborgen bleibt, kann es niemals einen Grund geben, diese Nummer zu ändern. Die auf Systemebene einzig interessierenden Anforderungen bezüglich Unveränderlichkeit und Eindeutigkeit können somit problemlos erfüllt werden. Im Falle eines objektorientierten PLM-Systems wird die Identifikation Objekt-ID genannt und entlang der Klassenhierarchie vererbt, wodurch beispielsweise ein Bauteil auf jeder Hierarchieebene über dieselbe interne Objekt ID (z.B. ABC0123) identifiziert wird.

*Anwendersseitig* werden hierfür in der betrieblichen Praxis die obenerläuterten Sachnummerungen eingesetzt, welche die unternehmensspezifische Aufbau- und Ablauforganisation [[Wie-89]] widerspiegeln. Diese Sachnummernsysteme sind in der Regel unternehmensweit genormt und vereinigen häufig verschiedene Aufgaben, d.h. Identifizieren, Klassifizieren sowie ggf. noch weitere. Durch ihre grössere Aussagefähigkeit aufgrund klassifizierender Inhalte werden Sachnummernsysteme vom Benutzer bis heute uneingeschränkt bevorzugt, obwohl durch ihre wesensbedingte Änderungs-fähigkeit eine Konsistenzgefahr besteht. Durch die Kombination bzw. gegenseitiger Umsetzung von systeminterner Identifizierung (Surrogat) und benutzerseitiger Identifikation mittels Sachnummernsystemen können die Vorteile beider Methoden miteinander vereinigt werden.

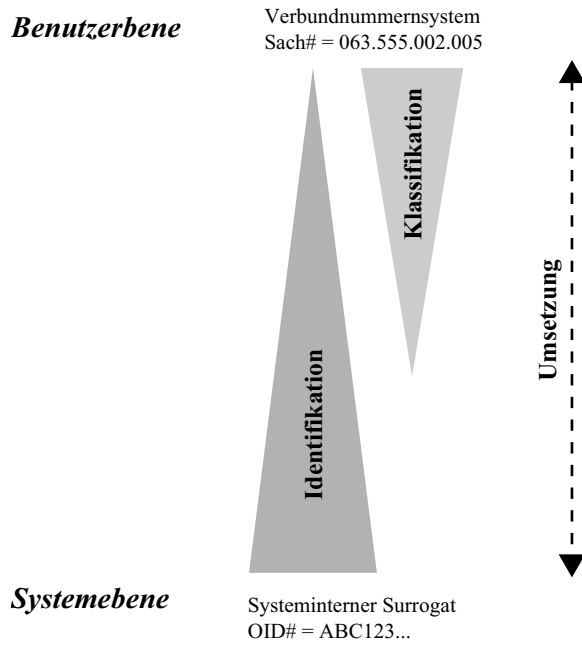


Bild (B011oidZ) Diskrepanz der Zielsetzungen auf Benutzer- und Systemebene

## 5 Zusammenfassung

Für den Zugriff auf ein Objekt werden bestimmte kennzeichnende Merkmale durch eine *Nummer* verschlüsselt. Sie kann Aussagen machen über Klassierung, Variantenbildung, den Änderungszustand usw.

Die Aufgaben einer Benummerung bestehen in der Identifikation (Identifikations- oder Identnummer) und Klassifikation (Klassifizierungsnummer) eines Objektes und in der Information und Kontrolle.

Eine Nummer besteht aus mehreren Nummernteilen, wobei jedes Nummernteil von mehreren Nummernstellen gebildet wird. Der Nummernaufbau charakterisiert die Art und Weise der Zusammensetzung einer Nummer aus den Nummernteilen. Es wird zwischen hierarchischem Aufbau und unabhängigem Aufbau unterschieden. Beim hierarchischen Aufbau steht eine Nummernstelle in eindeutiger Abhängigkeit von der vorhergehenden Stelle. Hat jede Nummernstelle ihre eigene Bedeutung, so spricht man von einem unabhängigen Aufbau.

Das Prinzip der *Wertbereichsbildung* dient der Strukturierung innerhalb eines Nummernteils. Die Gruppierung bezüglich eines Merkmals (Klassifikation) erfolgt über die Zuordnung von Eigenschaften zu Wertebereichen einer Nummer, die jeweils durch ihre Anfangs- und Endwerte festgelegt werden.

Ein *Nummernsystem* ist eine gegliederte Kombination von Nummern oder Nummernteilen aus meist unterschiedlichen Nummernarten. Beim Verbundnummernsystem werden zur Identifizierung sowohl Ident- als auch Klassifikationsnummer benötigt. Beim Parallelnummernsystem erfolgen Identifizierung und Klassifikation unabhängig voneinander. Die Identnummer enthält keine klassifizierenden Merkmale.

*Sachnummerierung* stellt eine der wichtigsten Anwendungen von Nummernsystemen dar. Eine Sachnummer ist eine Identnummer, kann jedoch noch weitere Funktionalitäten aufweisen.

In der Praxis werden betriebliche Objekte durch Sachnummernsysteme identifiziert. Sachnummern bilden dann die zentralen Nummern in PLM- und ERP-Systemen.

**Verständisfrage 1**

Wodurch werden Produktdaten identifiziert?

**Verständisfrage 2**

Welches sind die grundlegenden Konzepte der Nummerungstechnik?

**Verständisfrage 3**

Was ist der Unterschied zwischen Identifikation und Klassifikation?

**Verständisfrage 4**

Was versteht man unter einer „sprechenden Nummer“?

**Verständisfrage 5**

Wozu benötigt man eine dezentrale Nummernvergabe?

**Verständisfrage 6**

Anhand welcher Aufbauprinzipien kann man Nummernteile verbinden und wie kann über deren sinnvolle Anwendung befunden werden?

**Verständisfrage 7**

Welche Grobcharakterisierung wird für Nummernsysteme zweckmässigerweise verwendet und welche Vor- und Nachteile sind dabei zu vermerken?

**Verständisfrage 8**

Was ist das entscheidende Kriterium für ein Parallelnummernsystem?

**Verständisfrage 9**

Welche anwendungsbezogenen Merkmale werden bei Sachnummern häufig verschlüsselt?

**Verständisfrage 10**

Welche Vor- und Nachteile ergeben sich bei der Zuordnung anhand identischer Nummernwerte?

**Verständisfrage 11**

Wodurch unterscheiden sich Verwendungs- und Strukturabhängigkeit?

**Verständisfrage 12**

Worauf muss bei der Änderungskennzeichnung hinsichtlich der Rückverfolgbarkeit geachtet werden?

**Verständisfrage 13**

Welche Möglichkeiten bestehen grundsätzlich zur Objektidentifikation in PLM-Systemen?

**Verständisfrage 14**

Weshalb wird ein systeminternes Surrogat benötigt und was versteht man darunter?



## Publikationsverzeichnis – Literatur

- [BeB-85] *Bernhardt, R.; Bernhardt, W.*: Nummernsysteme: Grundbegriffe und Einführung, Systemvergleiche, praktische Anwendung, speziell bei EDV-Systemen. Sindelfingen: Expert Verlag 1985
- [DIN6763] *DIN 6763*: Nummerung - Grundbegriffe. Berlin: Beuth Verlag 1985
- [DIN9000] *DIN/ISO 9000*: Qualitätsmanagement- und Qualitätssicherungsnormen; Leitfaden zur Auswahl und Anwendung. Berlin: Beuth Verlag 1990
- [DIN9001] *DIN/ISO 9001*: Qualitätssicherungssysteme - Modell zur Darlegung der Qualitätssicherung in Design/Entwicklung, Produktion, Montage und Kundendienst. Berlin: Beuth Verlag 1992
- [Dir-73] *Dirzus, E.*: Die Umstellung auf ein neues Sachnummernsystem: Voraussetzungen - Vorgehensweise - Erfahrungen. Girardet-Taschenbücher Technik Bd. 14, Essen: Girardet Verlag 1973
- [EHS-91] *Eigner, M.; Hiller, C.; Schindewolf, S.; Schmich, M.*: Engineering Database: strategische Komponente in CIM-Konzepten. München: Hanser Verlag 1991
- [Gru-89] *Grupp, B.*: Optimale Verschlüsselung bei Online-Datenverarbeitung. Köln: Verlag TÜV Rheinland 1987
- [Jah-79] *Jahnke, B.*: Gestaltung leistungsfähiger Nummernsysteme für die DV-Organisation. München: Minerva Publikation 1979
- [KuW-81] *Kuhnert, W.; Werner, G.*: EDV-gerechte Verschlüsselung. 2. Aufl., Wiesbaden: Forkel 1981
- [Opi-71] *Opitz, H.*: Die richtige Sachnummer im Fertigungsbetrieb. Girardet-Taschenbücher Technik Bd. 2, Essen: Girardet Verlag 1971
- [Schl-63] *Schlaffner, P.*: Systemfreie Benummerung. DIN-Mitteilungen 42 (1963) H. 12, S. 601-609
- [SUL-89] *Gebrüder SULZER AG*: Werksnormung im SULZER-Konzern: Zweck, Inhalt und Anwendung. Winterthur: Februar 1989
- [VDI-72] *VDI-Fachgruppe Betriebstechnik (ADB)*: Elektronische Datenverarbeitung bei der Produktionsplanung und -steuerung III: Informations- und Stücklistenwesen. VDI-Taschenbücher T 28, Düsseldorf: VDI-Verlag 1972

# CAx/PLM-Integration

Autor: Dr. R. Montau



# 1 Überblick

Der Integration von Erzeugerapplikationen kommt im PLM eine zentrale Bedeutung zu. Hierbei stehen insbesondere CAx-Applikationen und DTP/Office-Applikationen im Vordergrund. Vor allem im M-CAD Bereich liegen komplexe Datenstrukturen und Problemstellungen vor, weshalb der E-CAD Bereich in der Regel eine etwas geringere Beachtung erhält. Die Integration von Software-Entwicklungen soll hier nicht berücksichtigt werden, da deren CASE-Tools (z.B. mit Clearcase) üblicherweise eine physische Datenintegration nicht erlauben und somit eine andersgelagerte Aufgabenstellung darstellt.

Im vorliegenden Abschnitt über die CAx/PLM-Integration werden somit folgende Themenschwerpunkte diskutiert:

- Check In/Check Out-Mechanismus,
- Concurrent Engineering,
- E-CAD Integration mit M-CAD/PLM,
- Visualisierung und Rendering,
- Prozessbezug von PLM-Applikationen.

## 2 Check In/Check Out-Mechanismus

Aufgrund des arbeitsteiligen Charakters einer Produktentwicklung und des Mehrbenutzerbetriebes von PLM-Systemen sind gewisse Sicherungsmechanismen nötig, damit die Benutzer möglichst uneingeschränkt voneinander arbeiten können, ohne sich gegenseitig ihre Datenbestände zu überschreiben (vgl. Bild B014dbtZ).

Da die PLM-Datenbank per se sämtlichen Benutzern zur Verfügung steht, ist beim Zugriff auf ein Objekt mit der Absicht zur Änderung (Update) ein sogenannter „Check Out“ erforderlich, um das Objekt für eine exklusive Schreiboperation zu reservieren. Nach Beendigung eines Update-Vorgangs kann ein sogenannter „Check In“ erfolgen, womit das Objekt wieder sämtlichen übrigen Benutzern zur Verfügung steht, die eine ausreichende Autorisierung besitzen, um gegebenenfalls ihre Operationen durchzuführen (Bild B001caxZ).

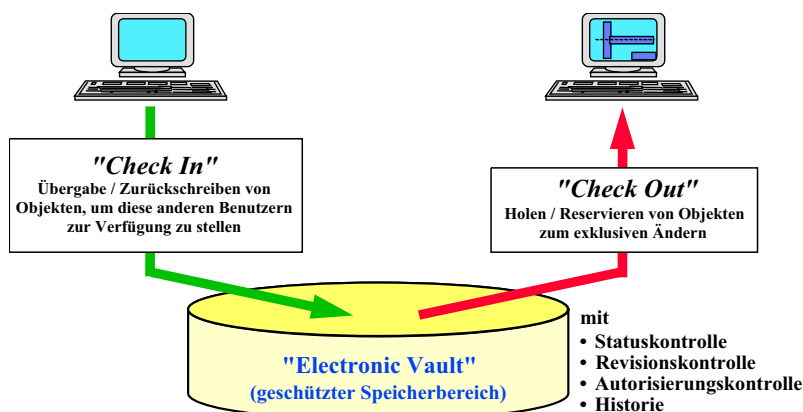


Bild (B001caxZ) Check In/Check Out von PLM-Objekten

Zur genaueren Beschreibung der Check In/Check Out-Funktionalität und deren Anwendung sind folgende Gesichtspunkte näher zu betrachten:

- expliziter und impliziter Check Out,
- Export-Funktionalität beim Check Out,
- Check Out-Sperren versus Prozess-Sperren.

## 2.1 Expliziter und impliziter Check Out

Grundsätzlich können Check In/Check Out-Vorgänge auf zwei Arten ausgelöst werden:

- Expliziter Check In/Check Out  
Dieser wird vom Benutzer bewusst vollzogen, um ein beliebiges Objekt über einen meist längeren evtl. noch unbestimmten Zeitraum exklusiv für sich zu reservieren.
- Impliziter Check In/Check Out  
Dieser wird vom PLM-System während dem Zugriff einer Fremdapplikation auf ein Container-Objekt automatisch durchgeführt, da diese solange die Kontrolle über die Datei(en) besitzt.

Im Gegensatz zum expliziten Check In/Check Out, der prinzipiell auf sämtliche Objektklassen anwendbar ist, kann ein impliziter Check In/Check Out ausschliesslich für Container-Objekte (vgl. Datasets in Bild B001okIZ) erfolgen und zwar genau dann, wenn deren Fremdapplikation aktiviert wird (z.B. Word für eine Text-Datei oder Unigraphics für ein CAD-Modell).

In Abhängigkeit der Leistungsfähigkeit bzw. „Tiefe“ der Integration läuft beim Starten einer Fremdapplikation der Check Out für Dataset-Objekte entweder implizit und damit automatisch im Hintergrund ab oder aber der Benutzer muss dies selbst manuell veranlassen. Die letztere weniger komfortable Methode bedeutet für den Benutzer meist einen zusätzlichen, separaten Schritt.

Falls ein PLM-System keinen impliziten Check Out unterstützt, wird jeder Ladevorgang grundsätzlich mit Read Only-Berechtigung durchgeführt und der Benutzer muss anschliessend noch „von Hand“ einen expliziten Check Out durchführen, um Schreibzugriff zu erlangen. Ein direktes Aktivieren von CAx- oder DTP-Applikationen aus dem PLM-System heraus ist jedoch in der Regel immer möglich.

## 2.2 Export-Funktionalität beim Check Out

Der Check In/Check Out-Mechanismus kann zudem mit einem Exportieren der Objekte aus der PLM-Datenbank kombiniert werden. Insbesondere für den oben erläuterten impliziten Check Out von Dataset-Objekten werden in der überwiegenden Mehrzahl der Fälle die betroffenen Dateien gleichzeitig vom PLM-Vault auf die Betriebssystemebene exportiert.

Zwingend ist dies jedoch nicht und einige PLM-Systeme unterstützen auch das direkte Arbeiten „über“ dem PLM-Vault, woraus zwei Möglichkeiten resultieren:

- Check Out mit Export  
Dateien werden automatisch vom Vault auf die Betriebssystemebene exportiert.
- Check Out ohne Export  
Dateien verbleiben im Vault und werden direkt hierüber bearbeitet.

Aus Sicherheitsgründen wird grundsätzlich in beiden Fällen immer auf einer Kopie der Datei(en) gearbeitet. Im Check Out-Fall ohne Export werden jedoch auf der Betriebssystemebene ausserhalb des PLM-Systems keinerlei Daten sichtbar, so dass OS-User ohne PLM-Zugriff keine Chance zum Lesen oder gar Schreiben erhalten.

Über den Check In/Check Out-Basismechanismus und die Kombination mit einem Datei-Export hinaus bieten PLM-Systeme in der Regel umfangreiche weitere Funktionen an (Auflistung der Check In/Check Out-Historie; Benachrichtigungslisten für automatische System-Mails, wenn ein gewünschtes Objekt wieder eingecheckt wird; Zurücksetzen und Weitergeben von Check Out-Rechten, ohne den Zwang zum Check In; etc.).

## 2.3 Check Out-Sperren versus Prozess-Sperren

Ein Check Out bedingt für das PLM-System zwingend das Setzen einer Sperre. Diese gilt dabei für das gesamte Dataset-Objekt, d.h. sowohl für die relational gespeicherten Meta-Daten (Verwaltungsdaten, welche auch die Check Out-Information beinhalten) als auch für die dazugehörige physische(n) Datei(en).

Das diesbezügliche Äquivalent für operationale Datenbankzugriffe ist die Anforderung einer Schreib-Sperre. Diese vom Applikationsprogrammierer definierten Sperren müssen jedoch von den anwenderseitig angeforderten Check Out-Sperren sorgfältig unterschieden werden:

- *Datenbank-Sperren (Check In/Check Out-Sperren)*  
werden vom Benutzer einer PLM-Applikation durch eine Anwendungsfunktion zur Laufzeit angefordert und sind in der Datenbank persistent gespeichert anhand einem sogenannten Checked Out-Attribut. Entsprechend der binären Charakteristik des Check Out-Mechanismus' (ausgecheckt: ja oder nein) kann dieses Attribut nur zwei Werte annehmen und ist entweder „null“ oder hat den Wert „Yes“. Eine Checked Out-Information kann dabei zeitlich unbegrenzt in der Datenbank verbleiben, bis ein Anwender es irgendwann wieder zurücksetzt.
- *Prozess-Sperren*  
werden vom Applikationsprogrammierer im SQL-Zugriffscodex definiert, um die Datenbankintegrität im parallelen Mehrbenutzerbetrieb zu gewährleisten (vgl. Transaktionsverwaltung, Kap. 3.3). Sie werden daher von einem Laufzeit-Prozess gewissermassen automatisch gesetzt und wieder freigegeben. Dennoch können auch Prozess-Sperren eine PLM-Benutzersitzung überdauern, da im Falle eines Absturzes eine reguläre Freigabe von Prozess-Sperren meist nicht mehr möglich ist. Die Sperren solcher „toten Prozesse“ müssen dann durch spezielle Eingriffe beseitigt werden, wofür i.d.R. systemspezifische Utilities zur Verfügung stehen.

Grosse Bedeutung kommt diesen Sperrmechanismen und der diesbezüglichen Interoperabilität von PLM und CAx insbesondere bei parallelen Mehrbenutzerzugriffen während einer Produktentwicklung zu, wenn Produktmodelldaten effizient und concurrent im Entwicklungsteam bearbeitet werden sollen.

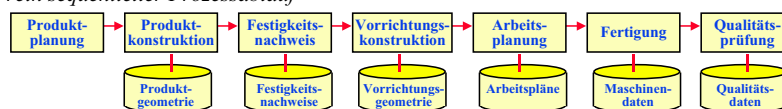


### 3 Concurrent Engineering

Zielsetzung des Concurrent Engineerings ist die weitgehende Parallelisierung von Entwicklungstätigkeiten, um die Auftragsdurchlaufzeiten zu verkürzen. Voraussetzung hierfür ist eine bereichsübergreifende Datenintegration und die Abkehr von einer dokumentbasierenden Vorgehensweise (vgl. Bild B001plmZ) hin zu einer durchgängig rechnerintegrierten Produktentwicklung. Deren letzte Konsequenz stellt das sogenannte „Digital Mockup“ dar, wo unter Verzicht auf jede materielle Modellanfertigung ein Produkt rein digital bis hin zur Produktionsreife entwickelt wird.

Ergebnis einer jeden Entwicklungstätigkeit ist hierbei nicht mehr die Erstellung eines Dokumentes, sondern der Aufbau eines rechnerinternen Produktmodells, wovon bei Bedarf verschiedene Dokumente (z.B. Technische Zeichnungen, Stücklisten etc.) abgeleitet werden können. Sämtliche Mitglieder eines Entwicklungsteams arbeiten somit gemeinsam an der Fortschreibung und Konkretisierung des Produktmodells, wobei unzählige Parallelzugriffe erfolgen, die von der Transaktionsverwaltung des zugrundeliegenden Datenbanksystems anhand von Sperrmechanismen koordiniert werden. Der entscheidende Vorteil des Concurrent Engineerings ist ein erheblicher Zeitgewinn (Bild B002caxZ), der im heutigen Konkurrenzdruck einen ganz entscheidenden Wettbewerbsvorteil darstellt (vgl. Bild B003plmZ).

rein sequentieller Prozessablauf



"Concurrent Engineering" auf der Basis eines integrierten Produktmodells

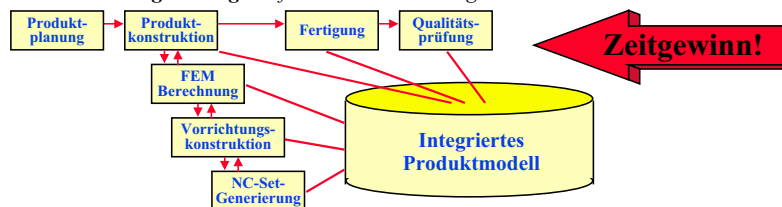


Bild (B002caxZ) Zeitgewinn durch Concurrent Engineering

Voraussetzung für das Concurrent Engineering ist eine Datenintegration, welche mit der PLM-Technologie durch unterschiedliche Abstraktionsbenen der Datenmodelle von PLM und CAx/DTP erreicht wird

(vgl. Kap. 4.1). Bezüglich dem Zusammenspiel von PLM mit Fremdapplikationen sind daher unterschiedliche Integrationstiefen zwischen diesen Datenmodellebenen möglich, wobei zwei wesentliche Leistungsstufen der Einbindung zu unterscheiden sind:

- Einkapselung von Fremdapplikationen,
- „Tiefe“ Integration mit Fremdapplikationen.

### 3.1 Einkapselung von Fremdapplikationen

Da PLM-Applikationen eine Abkehr von den herkömmlichen, abteilungsspezifisch verwalteten Insellösungen bewirken sollen, benötigen sie einen Mechanismus, um verschiedenste Fremdapplikationen einfach und flexibel einbinden zu können. Die Zielsetzung liegt hier in einer möglichst grossen Flexibilität, wobei bezüglich der Funktionalität lediglich gewisse Minimalanforderungen erfüllt sein müssen. Hierfür stehen zwei Aspekte im Vordergrund:

- Einerseits muss der Datenbestand der Fremdapplikationen physisch in der PLM-Datenbank (d.h. dem Vault) gespeichert werden, damit PLM eine Kontrollfunktion ausüben kann (→ Container-Objekte, z.B. Dataset).
- Andererseits muss der Fremddatenbestand mit einer oder auch mehreren Fremdapplikationen verknüpft werden, um festzulegen, welche Applikationen (Tool, z.B. MS Word) für ein bestimmtes Container-Objekt (Dataset-Typ, z.B. Word) zweckmässig angewendet werden kann und wo das ausführbare Programm zu finden ist (→ Tools).

Die Speicherung der physischen Dateien einer Fremdapplikation erfolgt fast immer direkt in deren Originalformat, d.h. ohne jede Konversion. Da die Datenübertragung zwischen PLM-System und dem persönlichen Arbeitsbereich eines Benutzers häufig auf FTP-Funktionen (File Transfer Protocol) basiert, muss in der Regel angegeben werden, ob es sich um eine ASCII- oder eine Binär-Datei handelt. Die Repräsentation von Fremdobjekten im PLM erfolgt durch Dataset-Objekte, die eine oder auch mehrere physische Dateien enthalten können.

Auf der anderen Seite kann einem Dataset-Typ ein Applikationsprogramm zugeordnet werden, wodurch meist durch simples Doppelklicken (Aktion Open) auf ein Dataset-Objekt, dessen Datei(en) in die Fremdapplikation geladen werden können. Hierfür werden Kopien der vom Dataset referenzierten Datei(en) vom Vault in einem persönlichen Arbeitsbereich des Benutzers auf seiner lokalen Maschine bereitgestellt. Diesbezüglich können Datasets als „Transport-Objekte“ bzw.

Koffer oder Taschen angesehen werden, mit welchen Dateien ausgecheckt und im Anschluss an die Bearbeitung in der Fremdapplikation (nach Save und Exit) durch einen Check In wieder in die PLM-Datenbank zurückgeschrieben werden können.

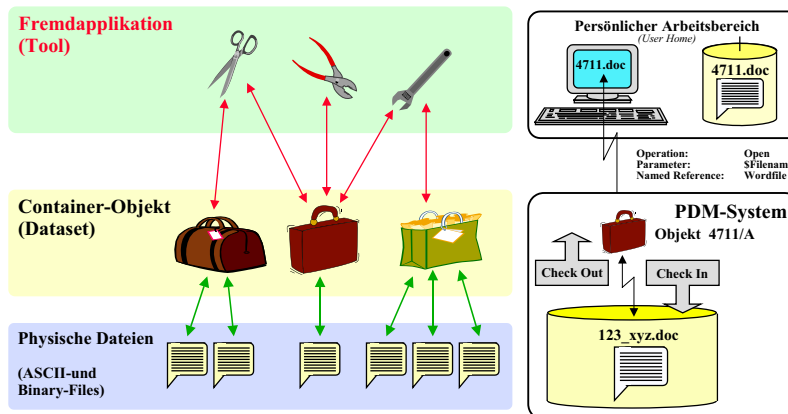


Bild (B003caxZ) Einkapselung von Fremdapplikationen und Check Out-Mechanismus

Der grosse Vorteil bei dieser Einkapselung ist die Flexibilität, da leistungsfähige PLM-Systeme verschiedenartigste Fremdapplikationen mit nur wenigen Eingriffen zum Teil in Minutenschnelle einbinden können, solange es sich lediglich um atomare Dateien handelt.

Die Unterschiede und Nachteile von Einkapselungen gegenüber „tieferen“ Integrationen, die für komplexe CAx-Dateistrukturen zwingend benötigt werden, liegen in der systemübergreifenden Funktionalität begründet. Diese hat entscheidenden Einfluss auf die Interoperabilität im Rahmen des Concurrent Engineerings, wo der Entwicklungsprozess im Vordergrund steht.

### 3.2 „Tiefe“ Integration von Autorensystemen

Das Concurrent Engineering ist im Produktdatenmanagement keinesfalls eine eingeführte und problemlos zu realisierende Entwicklungsmethode. Die diesbezügliche Funktionalität hängt von zahlreichen Faktoren und insbesondere von der Integrationstiefe zwischen den CAx-Applikationen und dem PLM-System ab, wofür zwei Aspekte im Vordergrund stehen:

- möglichst bidirektionale Funktionalität und Assoziativität zwischen CAx und PLM,

- Interoperabilität bezüglich dem Setzen und Freigeben von Objektsperren (Granularität und Dauer von Lese- und/oder Schreib-Sperren).

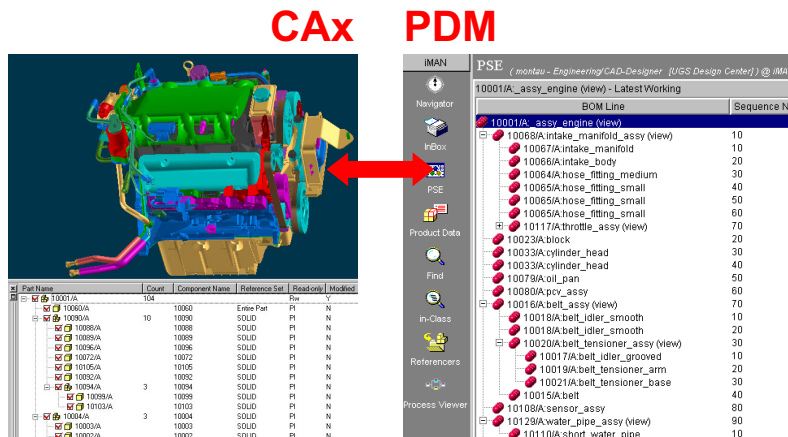


Bild (B004caxZ) Beispielhafte Verdeutlichung der Integrationsproblematik zwischen CAD und PLM

Für die oben beschriebene Funktionalität wird eine PLM-seitige Repräsentation der CAD-Strukturen benötigt, wofür zwei prinzipielle Ansätze bestehen:

- Dokument-Struktur mit 2-stufiger Abbildung:  
Die CAD Assembly-Struktur wird 1:1 auf eine Dokumentstruktur abgebildet, woraus PLM-intern wiederum eine Artikelstruktur generiert wird. Das 3D-Modell wird konzeptionell auf ein Dokument abgebildet. Dokument- und Artikelstruktur werden grundsätzlich separat gehandhabt, können aber in der Regel in Grenzen miteinander gekoppelt werden. Beiderseitige Strukturmodifikationen und Revisionsänderungen im Mehrbenutzerbetrieb sind hierbei problematisch, da die Dokumentstruktur sowohl von der CAD- als auch von der Artikel-Seite modifiziert werden müsste. Um potentielle Konflikte zu vermeiden, bestehen häufig Restriktionen bzgl. der gegenseitigen Synchronisation (z.B. Strukturdefinition/-modifikation nur von CAD nach PLM (Dokumentstruktur)).
- PLM Hybrid-Struktur mit 1-stufiger Abbildung:  
Die CAD Assembly-Struktur wird direkt auf eine PLM Hybrid-Struktur abgebildet, welche sowohl für Artikel als auch für Dokumente genutzt wird. Das 3D-Modell wird konzeptionell auf

ein Artikel abgebildet. Die Synchronisation ist üblicherweise intelligent, d.h. es kann in beide Richtungen synchronisiert werden (von CAD nach PLM und von PLM nach CAD) und nötigenfalls können Komponenten gegenseitig „ausgefiltert“ werden (Umgebungsgeometrie nur in CAD, da nicht Stücklisten-relevant ist; Nicht-Geometrische Position nur in PLM, wie z.B. Öl, ohne CAD-Geometrie). Da PLM-intern nur 1 Struktur gefahren wird, können keine internen Struktur-Konflikte auftreten und es lässt sich in der Regel eine tiefere CAD-Integration erreichen (Parametervariationen, absolute Strukturkontexte etc.).

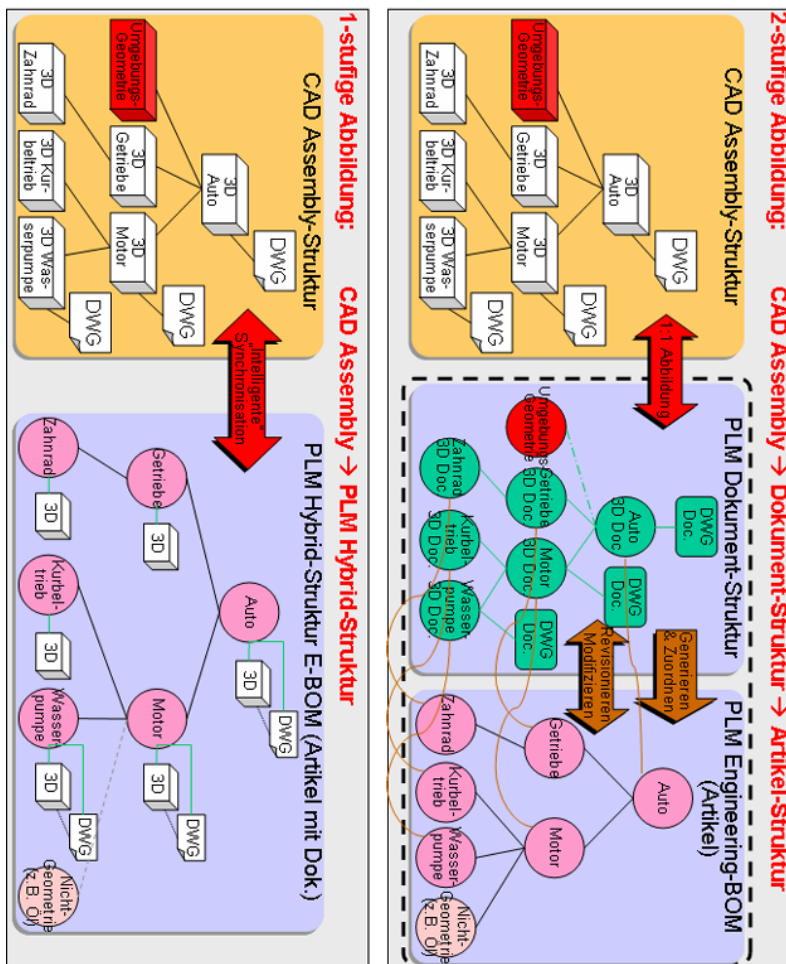


Bild (B017caxZ) CAX-Integration in PLM: 2-stufige Abbildung vs. PLM Hybrid-Struktur

Hinsichtlich der bidirektionalen Funktionalität und Assoziativität zwischen CAD und PLM sind insbesondere diejenigen Informationsmengen zu betrachten, die sowohl in CAD als auch in PLM vorhanden sind. Einerseits ist für die Funktionalität entscheidend, ob dem Anwender sämtliche interessierenden Funktionen sowohl in PLM als auch in CAD zur Verfügung stehen und er frei wählen kann, wo und wie er am effizientesten arbeitet. Andererseits ist aufgrund der Redundanz der Informationen eine systemseitige Kontrolle erforderlich, um das jeweils andere System zu synchronisieren. Um die Leistungsfähigkeit und Integrationstiefe einer PLM/CAD-Schnittstelle beurteilen zu können, müssen daher eine Reihe von Fragen untersucht werden:

- Kann eine Produktstruktur sowohl PLM- als auch CAD-seitig bearbeitet werden?
- Wird das Hinzufügen eines Teiles in die PLM-Produktstruktur auch in der CAD-Assembly-Struktur nachgeführt? Falls ja, können zusätzliche Teile in CAD automatisiert eingelesen werden durch Default-Werte zur Positionierung und Orientierung?
- Wird das Löschen eines Teiles in der PLM-Produktstruktur auch in der CAD-Assembly-Struktur automatisch nachgeführt?
- Wird das Hinzufügen eines Teiles in die CAD Assembly-Struktur in der PLM-Produktstruktur automatisch nachgeführt?
- Wird das Löschen eines Teiles aus der CAD Assembly-Struktur in der PLM-Produktstruktur automatisch nachgeführt?
- Kann die CAD-Produktstruktur um nicht-geometrische Teile erweitert werden (z.B. Betriebs- und Schmierstoffe, wie Fett, Öl, etc.), die nicht in die CAD Assembly-Struktur übernommen werden?
- Können mehrere Produktstruktursichten (BOM View Types) mit der CAD Assembly-Struktur synchronisiert werden?
- Kann ein CAD Assembly-Modell in Abhängigkeit PLM-seitig definierter Konfigurationsregeln geladen werden?
- Stehen die PLM-Konfigurationsregeln auch CAD-seitig als Load Options flexibel zur Verfügung?
- Können für mehrfach verwendete Komponenten (z.B. Schrauben mit Menge 6) in PLM unterschiedliche verwendungsbezogene Angaben in den Produktstrukturbeziehungen hinterlegt werden (z.B. Montageanweisung: Anzugsmoment 60 Nm für Schrauben 1 bis 4, 80Nm für Schrauben 5 und 6)?

Für die Concurrent Engineering-Problematik ist jedoch nicht nur die bidirektionale Funktionalität und Assoziativität von Interesse, sondern insbesondere auch die Möglichkeiten für Parallelisierungen im Entwicklungsteam. Um die diesbezüglich interessierende Interoperabilität

bezüglich dem Setzen und Freigeben von Objektsperren beurteilen zu können, muss das Szenario eines Mehrbenutzerzugriffes untersucht werden, wo häufig eine hochgradig parallelisierte CAX-Produktmodellierung auftreten kann, wie in Bild B005caxZ dargestellt.

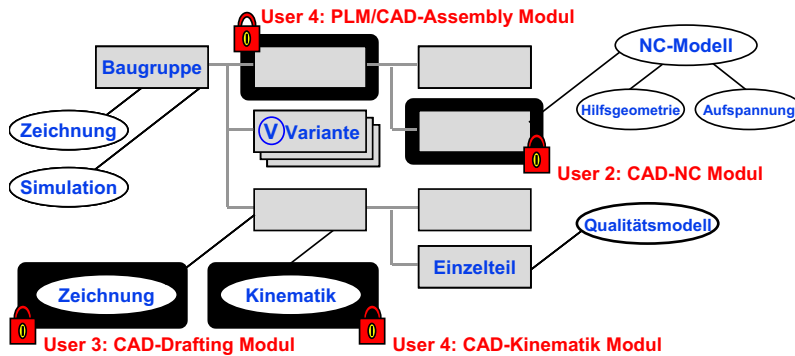


Bild (B005caxZ) Beispiel für ein komplexes Szenario im Concurrent Engineering

Angenommen, ein Entwicklungsingenieur lädt ausgehend von der PLM-Produktstruktur das oberste Bauteil (top Assembly, z.B.: Auto) in eine CAD-Applikation, dann muss PLM- und CAD-seitig eine diesbezügliche Sperre gesetzt werden (impliziter Check Out, vgl. Bild B001caxZ). Entscheidend hierbei ist, ob in beiden Systemen nur das ausgewählte oberste Bauteil gesperrt wird, oder ob aufgrund mangelnder Integrationsstiefe gar die gesamte Produktstruktur gesperrt werden muss. Letzteres würde fatale Konsequenzen mit sich bringen und ein Concurrent Engineering unter Einbezug von CAX-Applikationen verunmöglichen.

Falls nur das oberste Bauteil gesperrt wird und der Entwicklungsingenieur im Rahmen der CAD-Assembly Modellierung auf eine tiefergelegene Baugruppe (z.B. auf den Motor) wechselt, stellen sich eine Reihe weiterer Fragen:

- Können CAX-seitig weitere Sperren vom PLM-System nachgefordert werden?
- Können CAX-seitig nicht mehr benötigte Sperren (z.B. höhergelegener Assembly-Baum) wieder freigegeben und damit für andere PLM-Benutzer verfügbar gemacht werden?
- Wird beim Wechsel auf ein zwischenzeitlich von einem anderen Entwicklungsingenieur modifiziertes Bauteil dies dem Konstrukteur gemeldet und dieser zum Laden des neuesten Modellie-

rungszustandes veranlasst (ansonsten Lost Update-Problematik, vgl. Bild B014dbtZ)?

- Wird sowohl CAx- als auch PLM-seitiges Fehlverhalten von Benutzern konsistent abgefangen?
- Können Lese- und Schreibsperrern unterschieden werden?
- Werden Sperren zum jeweils spätestmöglichen Zeitpunkt gesetzt und anschliessend zum frühestmöglichen Zeitpunkt wieder freigegeben oder unterscheidet sich die Granularität der Systeme?
- Wird beim Sperren lediglich das betreffende Objekt des CAx-Partialmodelles/CAx-Applikationsmodules gesperrt (z.B. Technische Zeichnung) oder müssen sämtliche diesbezüglichen Informationsaspekte mitgesperrt werden (3D-Modell, NC-Programm, Kinematikmodell, Simulationsmodell etc., vgl. Kap. 5.1.1)?

Diese charakteristischen Merkmale der Integrationstiefe und Interoperabilität sind von massgebender Bedeutung für die systemseitige Unterstützung des Concurrent Engineerings. Kommerziell verfügbare PLM/CAx-Lösungen unterscheiden sich diesbezüglich in grossem Masse, weshalb diese Kriterien jeweils sorgfältig analysiert werden müssen, damit nicht durch die Einführung eines PLM-Systems bereits vorhandene CAx-Funktionalität „verloren geht“.

### 3.2.1 Interoperabilität von PLM/CAx-Sperrmechanismen

Zur Veranschaulichung der Problematik bei der Handhabung von Sperrmechanismen in PLM und CAD sollen nachfolgend die wesentlichen charakteristischen Szenarien exemplarisch analysiert werden. Als Anwendungsbeispiel wird ein Motor betrachtet, der in der PLM-Produktstruktur eines Fahrzeugs enthalten ist und CAD-seitig als Assembly-Struktur existiert (Bild B006caxZ).



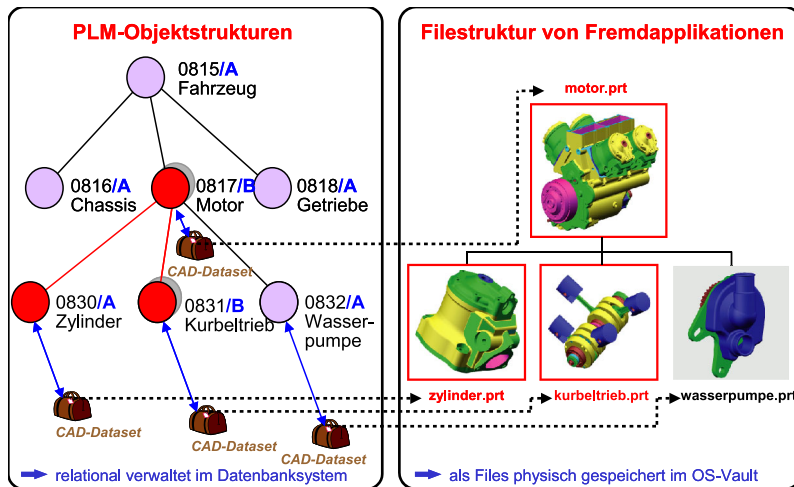


Bild (B006caxZ) Anwendungsbeispiel für ein Concurrent Engineering-Szenario

Betrachtet wird das Zugriffsszenario zweier User A und B, die ausgehend von der PLM-Produktstruktur einzelne Komponenten (Motor, Zylinder, Kurbeltrieb) nach CAD laden, um die Assembly-Struktur zu bearbeiten. Zur Verdeutlichung der Problematik sei darauf hingewiesen, dass beim Laden eines Assembly-Parts, jeweils von Beginn an schon Informationen der Komponenten mitgeladen werden müssen, um den Gesamtzusammenhang der Konstruktion zu visualisieren („Design in Context“) und die Assemblies zum Navigieren in tiefergelegene Baugruppen und Einzelteile anbieten zu können. Problematisch hieran ist, dass naturgemäss Komponenten der Assembly-Struktur von anderen Benutzern immer wieder modifiziert werden, so dass beim Wechsel auf eine solche Komponente ein sogenanntes Reload erforderlich wird, um nicht mit einer veralteten Version zu arbeiten. Der in Tabelle T001caxZ bis Tabelle T003caxZ dargelegte Beispielablauf mit den zugehörigen Systemmeldungen (Bild B007caxZ bis Bild B009caxZ) gibt Aufschluss über folgende Problemfelder:

- Nachfordern von Sperren aus CAD,
- Freigeben von Sperren aus CAD,
- Aktualisierungskontrolle beim Komponentenzugriff,
- Granularität und Dauer von Sperren,
- Priorisierung beim Ausbau paralleler Lese-Sperren.

Nr.	User A	User B
1	User A lädt die Motor-Baugruppe 0817/B von PLM in die CAD-Applikation (Partial Load)	
	Resultat: in PLM erfolgt ein impliziter Check Out (CO = „Yes“) [User A hält Schreib-Sperre für 0817/B]	
2	User A wechselt innerhalb von CAD auf die untergeordnete Zylinder-Baugruppe 0830/A (make workpart)	
	Resultat: nur Lesezugriff (Lese-Sperren werden in PLM nicht angezeigt) [User A hält Schreib-Sperre für 0817/B u. Lese-Sperre für 0830/A]	
3	User A beginnt eine Modifikation mit einer CAD-Modellierungsfunktion	
	iResultat: in PLM erfolgt ein impliziter Check Out für Zylinder-Baugruppe 0830/B [User A hält Schreib-Sperren für 0817/B u. 0830/A]	
4	*** Bemerkung *** Check Out-Sperren werden zum spätest möglichen Zeitpunkt gesetzt!	
5	User A schliesst in CAD die geöffnete Motor-Baugruppe 0817/B (Close Part)	
	Resultat: in PLM wird Check Out-Sperre für die Motor-Baugruppe 0817/B freigegeben [User A hält nur noch Schreib-Sperre für 0830/A]	
6	*** Bemerkung *** Check Out-Sperren in PLM werden aus CAD heraus zum frühest möglichen Zeitpunkt wieder freigegeben!	
7		User B lädt die Motor-Baugruppe 0817/B von PLM in die CAD-Applikation
	Resultat: in PLM erfolgt ein impliziter Check Out für die Motor-Baugruppe 0817/B [User A hält Schreib-Sperre für 0830/A; User B hält Schreib-Sperre für 0817/B]	
8	User A führt in CAD seine Modifikationen an der Zylinder-Baugruppe 0830/A durch	
9	User A speichert seine CAD-Modifikationen an der Zylinder-Baugruppe 0830/A	
	Resultat: in PLM wird Check Out-Sperre für Zylinder-Baugruppe 0830/A freigegeben [User A hält Lese-Sperre für 0830/B; User B hält Schreib-Sperre für 0817/B]	

Tabelle (T001caxZ) Beispielablauf im Concurrent Engineering (Teil 1)

Nr.	User A	User B
10		User B wechselt in CAD auf die untergeordnete Zylinder-Baugruppe 0830/A
	Resultat: nur Lesezugriff auf veraltete (!) Version der Zylinder-Baugruppe 0830/A [User B hält Schreib-Sperre für 0817/B; User A hält noch eine Lese-Sperre für 0830/A]	
11		User B versucht an der Zylinder-Baugruppe 0830/A eine Modifikation mit einer CAD-Modellierungsfunktion durchzuführen
	Resultat: User B erhält Warnung, dass in PLM eine neuere Version existiert (vgl. 9)!	
	Resultat: in PLM erfolgt ein impliziter Check-Out für die Zylinder-Baugruppe 0830/A, da User B seine Absicht zur Änderung bekannt gegeben hat! [User B hält Schreib-Sperren für 0817/B u. 0830/A]	

Tabelle (T001caxZ) Beispielablauf im Concurrent Engineering (Teil 1)

Der von User B beim Aufstarten von CAD in den Hauptspeicher geladene Zustand der Zylinder-Baugruppe 0830/A ist mittlerweile veraltet, da User A in den Schritten 8 und 9 eine neuere Version erzeugt und in die Datenbank zurückgespeichert hat. Bild B007caxZ zeigt die diesbezügliche Systemmeldung, welche User B hierauf aufmerksam macht und zu einem Reload veranlassen sollte.



Bild (B007caxZ) Systemmeldung beim Bearbeiten veralteter Datenbestände (Hinweis für Reload)

Nr.	User A	User B
12		User B ignoriert die Warnung und führt in CAD dennoch seine Modifikation an der veralteten Version durch, ohne Reload (!)
13		User B versucht seine Modifikation an der veralteten Version der Zylinder-Baugruppe 0830/A zu speichern → User B erhält eine Fehlermeldung, da er die Änderung nicht speichern kann.

Tabelle (T002caxZ) Beispielablauf im Concurrent Engineering (Teil 2)

Unabhängig von der Korrektheit des Benutzerverhaltens muss in jedem Falle gewährleistet sein, dass niemals etwas in die Datenbank zurückgespeichert werden kann, das auf einem veralteten Zustand aufbaut, der weiter zurück liegt, als der aktuelle (Bild B008caxZ). Ansonsten würde der aktuelle Zustand entsprechend der Problematik eines Lost Update (vgl. Bild B014dbtZ) verloren gehen.

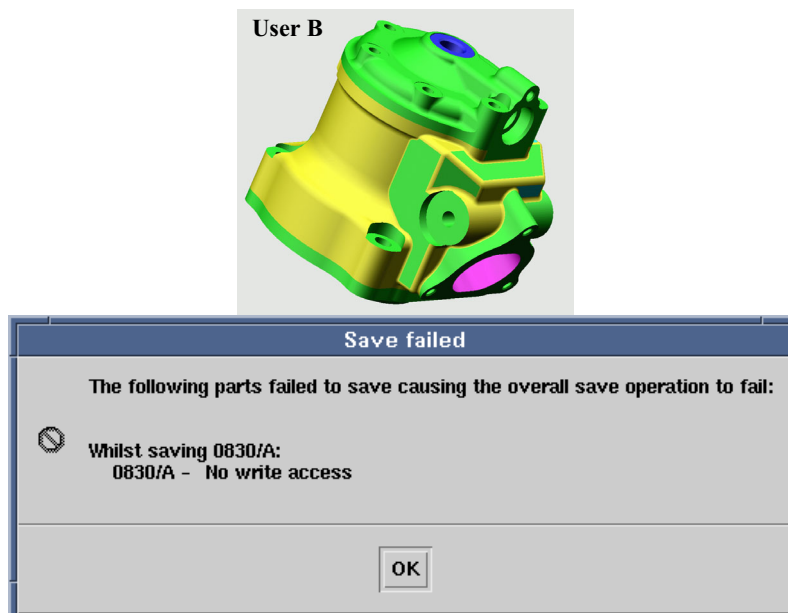


Bild (B008caxZ) Fehlermeldung beim Versuch, nicht integre Datenbestände zu speichern

Nr.	User A	User B
14	<p>*** Bemerkung ***</p> <p>Die Konsistenz des Datenbestandes ist jederzeit gewährleistet! User B kann die Modifikation nur unter einer anderen Nummer bzw. anderem Namen abspeichern!</p>	
15		<p>User B erkennt, dass auch fehlerhaftes Benutzerverhalten abgefangen wird (was er prüfen wollte) und schliesst die Zylinder-Baugruppe 0830/A ohne zu speichern</p>
	<p>→[User B hält nur noch Schreib-Sperre für 0817/B]</p>	
16		<p>User B wechselt innerhalb von CAD auf die Kurbeltrieb-Baugruppe 0831/B</p>
	<p>*nur Lesezugriff (Lese-Sperren werden in PLM nicht angezeigt) [User B hält Lese-Sperre für 0831/B u. Schreib-Sperre für 0817/B]</p>	
17	<p>User A wechselt innerhalb von CAD auf die Kurbeltrieb-Baugruppe 0831/B</p>	
	<p>*nur Lesezugriff (Lese-Sperren werden in PLM nicht angezeigt) [User A hält Lese-Sperre für 0831/B; User B hält Lese-Sperre für 0831/B u. Schreib-Sperre für 0817/B]</p> <p>*** Bemerkung ***</p> <p>Beide User halten gleichberechtigte Lese-Sperren auf 0831/B. Solange keine Modifikation erfolgt, stören sich die beiden User nicht.</p>	
18	<p>User A beginnt eine Modifikation mit einer CAD-Modellierungsfunktion</p>	
	<p>→in PLM erfolgt ein impliziter Check Out für Kurbeltrieb-Baugruppe 0831 B [User A hält Schreib-Sperre für 0831/B; User B hält Schreib-Sperre für 0817/B]</p>	
19		<p>User B beginnt eine Modifikation mit einer CAD-Modellierungsfunktion</p>
	<p>→User B erhält eine Warnung, dass User A die Kurbeltrieb-Baugruppe 0831/B gesperrt hat (Schreib-Sperre)!</p>	
29	<p>*** Bemerkung ***</p> <p>User A (Montau) hat als erster eine Schreib-Sperre angefordert und damit das Vorrecht. Falls erwünscht kann User B zwar weiterarbeiten, ist jedoch vom Verhalten von User A abhängig:</p> <p>Fall 1: Falls User A das Teil modifiziert und speichert, kann User B seine etwaigen Modifikationen nur unter einer anderen Nummer speichern (Fehlverhalten würde wieder zur Fehlermeldung von Schritt 13 führen)!</p> <p>Fall 2: Falls User A das Teil später ohne zu speichern schliesst, kann User B seine etwaigen Modifikationen speichern!</p>	

Tabelle (T003caxZ) Beispielablauf im Concurrent Engineering (Teil 3)

Die User A und B greifen maximal lange parallel und ungestört voneinander auf die Kurbeltrieb-Baugruppe 0831/B zu. Erst als auch User B mit einer Modifikation beginnen will, wird er über die Schreib-Sperre von User A informiert. Letztere wird daher erst zum spätest möglichen Zeitpunkt (in Schritt 18) gesetzt. Für den Fall, dass User A später aus der Kurbeltrieb-Baugruppe aussteigt, ohne zu speichern, wird User B sogar noch die Möglichkeit geboten, seine Modifikationen doch noch zurückzuspeichern (vgl. Bild B009caxZ). Dies verdeutlicht die sehr „tiefe“ Integration von PLM und CAD im untersuchten Beispielszenario, wodurch die Notwendigkeit erneuter Ladervorgänge in CAD – die insbesondere bei grossen Baugruppen (Large Assembly-Thematik) sehr zeitaufwendig sein können – auf ein Minimum reduziert wird.

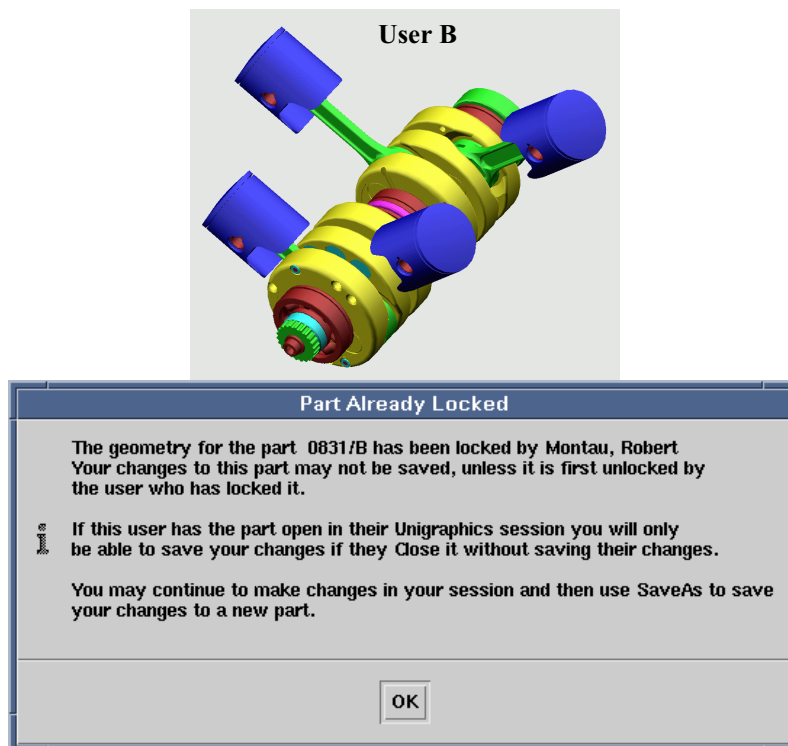


Bild (B009caxZ) Systemmeldung beim Versuch, eine bereits vergebene Schreib-Sperre anzufordern

Eine fast identische Situation ergibt sich bei der zusätzlichen Betrachtung weiterer CAx-Partialmodelle, wie z.B. Zeichnungen, CAM- oder FEM-Modelle (Ellipsen in Bild B005caxZ). Für die Zugriffskontrolle im

Concurrent Engineering sind dies ebenso Objekte, wie 3D-Modelle einer Assembly-Struktur, die gesperrt werden können oder auch nicht.

## 4 Integration von E-CAD mit PLM/M-CAD

Für die Produktentwicklung mittels Elektro-CAD Applikationen sind bzgl. der Integration die Themenfelder M-CAD und PLM zu beachten, wofür hier ein Einblick gegeben werden soll. Zunächst wird im E-CAD ein logisch-funktionales Elektroschema definiert, welches üblicherweise mittels Bibliothekskomponenten aufgebaut wird, die rein funktional verbunden werden mittels Signalen zwischen Ports. Da Elektroschemata sowohl im Leiterplattendesign als auch für Starkstrom- bzw. Schaltschrank-Thematiken grundsätzlich immer rein funktionalen Charakter haben, wird für sämtliche räumlich-gegenständliche Arbeiten ein M-CAD System benötigt. Für das M-CAD System sollten aus E-CAD die Komponenten- und Verbindungsstrukturen zur Verfügung gestellt werden.

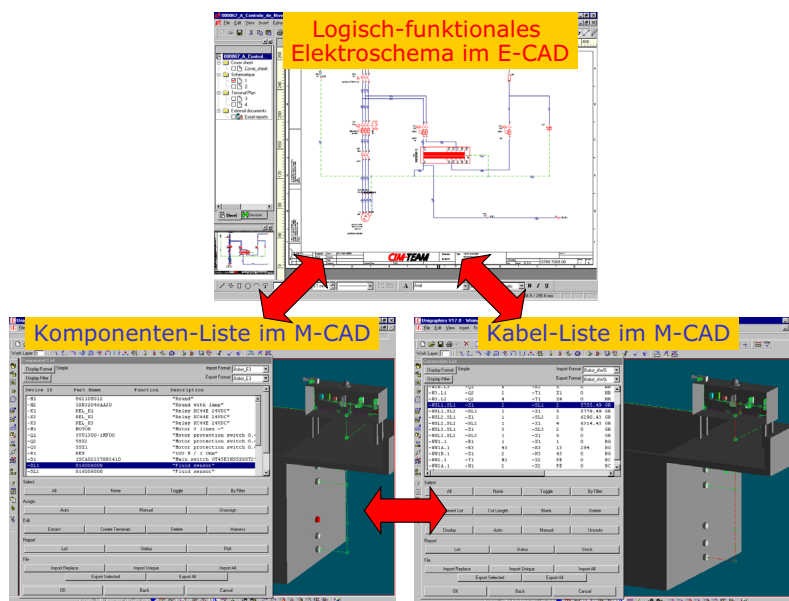


Bild (B010caxZ) Zusammenspiel von E-CAD und M-CAD

Im M-CAD werden zunächst die Elektro-Komponenten im Raum positioniert. Für eine jede Verbindung wird dann eine räumliche Trajektionsskurve definiert, entlang welcher die Verkabelung geführt wird, die abschliessend auch physisch dargestellt werden kann. Hiernach können Einbau- und Kollisionsanalysen geführt sowie die einzelnen Kabellängen bestimmt werden.



Im Gegensatz zur mehrstufigen M-CAD Stücklistenstruktur wird auf der E-CAD Seite nur eine einstufige Struktur benötigt für die Komponenten-Struktur als auch für die Kabel-Struktur (Verbindungen), was PLM-seitig mittels BOMView Types abgebildet wird.

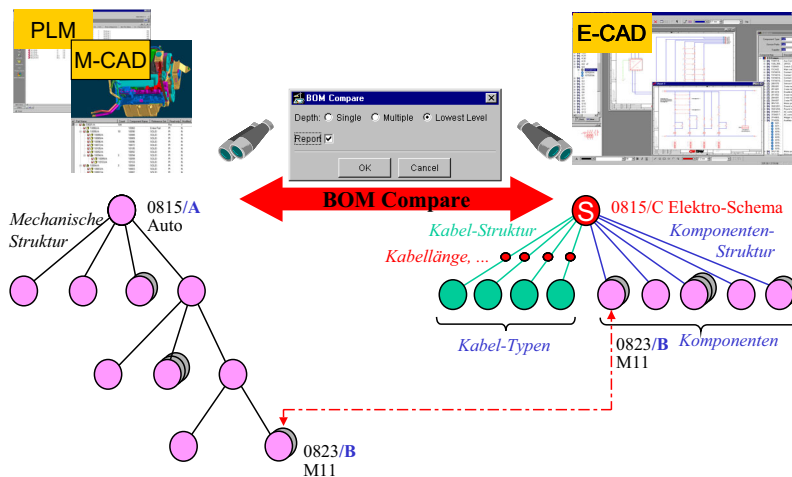


Bild (B011caxZ) Mehrstufige M-CAD Struktur gegenüber einstufiger E-CAD Strukturen

Die Zielsetzung einer E-CAD/M-CAD Integration liegt hierbei in folgenden Gesichtspunkten:

- Digital Mockup für die Verkabelung  
(ohne Notwendigkeit für einen physikalischen Prototyp)
- Bestimmen der Kabellänge im M-CAD
- Raumanalysen und Kollisionsbetrachtungen.

Primäre Aufgabe des PLM-Systems hierbei ist die Zuordnung der einzelnen Objekttypen und das Management der unterschiedlichen Struktursichten.

## 5 Visualisierung und Rendering

Ausgehend von den nativen Quellformaten datenerzeugender Applikationen (z.B. Unigraphics V18-Format oder Word-Format) werden heute zunehmend neutrale Formate gehalten (z.B. JT- oder TIFF-Format, bzw. PDF-Format), welche vorwiegend zwei Zielsetzungen dienen:

- Visualisierung und/oder
- Langzeitarchivierung (digitales Archiv gegenüber Mikrofilm).

Im folgenden soll zunächst auf die Generierung (Rendering) neutraler Formate eingegangen werden und im Anschluss auf das eigentliche Viewing und Markup (Redlining).

### 5.1 Generierung neutraler Formate (Rendering)

Arbeitsergebnis des Produktentwicklungsprozesses sind Produktdaten im proprietären Format des eingesetzten CAD-Systems (z.B. Unigraphics V18-Format). Aus verschiedenen Gründen werden häufig zusätzliche neutrale Ausgabeformate erzeugt, um z.B. unabhängig von Release-Änderungen oder gar einem späteren Wechsel des CAD-Systems zu werden, um Nicht-CAD Anwendern eine Visualisierung zu ermöglichen oder um ein Lesen der Daten auch noch z.B. in 20 Jahren sicherstellen zu können.

Grundvoraussetzung hierfür ist die Verwendung eines genormten Standards, der mit Sicherheit auch noch in der Zukunft Bestand hat. Beispiele hierfür sind z.B. das Rasterformat TIFF (Tagged Image File Format), das Vektorformat CGM (Computer Graphics Metafile), das Plotdatenformat HP-GL (HP Graphics Language) oder neuerdings die Produktmodellchnittstelle STEP (Standard for The Exchange of Product model data). Das jeweils bestgeeignete Format muss unter Berücksichtigung des unternehmensspezifischen Anwendungsfalles ausgewählt werden.

Für die Erzeugung neutraler Ausgabeformate bestehen grundsätzlich vier Möglichkeiten:

- interaktiv durch eine Benutzerinteraktion (evtl. nötig in Kombination mit Methoden 3 und 4),
- automatische Generierung durch das Triggern eines Save User Exits (beste Datenverfügbarkeit, aber laufzeitkritisch für Large Assembly-Anwendungen),
- automatische Generierung durch Triggerung in einem Freigabeprozess

(lauzeiteffizient und besonders geeignet zur Archivierung).

- Automatische Generierung durch asynchronen Batch-Prozess (sehr lauzeiteffizient und besonders geeignet zur Archivierung).

In der Mehrzahl der Fälle wird das automatische Generieren in einem Freigabeprozess die bestgeeignete Lösung sein. Hierbei werden neutrale Ausgabeformate vorwiegend in einer späten Phase eines Freigabeprozesses erzeugt, wenn beispielsweise ein erfolgreicher Abschluss des Prozesses bereits feststeht oder aber kein Schreibzugriff mehr existiert. Ebenso kann es jedoch erforderlich sein, bereits zu Beginn des Freigabeprozesses ein neutrales Format zu erzeugen, um beispielsweise einem beliebigen Benutzer ohne CAD-Lizenz einen Lesezugriff zu ermöglichen.

## 5.2 Viewing und Markup

Charakteristisch für einen Freigabeprozess oder auch für beliebige andere Workflows ist die Prüfung der zugehörigen Objekte durch eine oder mehrere weitere Personen, bevor ein SignOff ausgeführt wird. Analog zur herkömmlichen Kontrolle einer Technischen Zeichnung oder eines Festigkeitsnachweises in Papierform durch z.B. einen Konstruktionsleiter soll im heutigen Informationszeitalter dieser Prüfschritt elektronisch unterstützt werden. Hierfür werden auf dem Markt sogenannte Viewing&Markup-Tools angeboten, die einen Zugriff ermöglichen, i.d.R. unabhängig von der ursprünglichen Erzeuger-Applikation (z.B. CAD-System X), des vorliegenden Betriebssystems (z.B. Unix, NT oder Windows 2000) und der netzwerktechnischen Einbindung (z.B. firmeninternes LAN oder Internet). Beispielhaft erwähnt für solche Systeme seien z.B.:

- Myriad (Fa. Informative Graphics),
- Spicer (Fa. Imagination),
- Teamcenter Visualization (Fa. EDS PLM Solutions) etc..

Viewing & Markup-Systeme sind heute in verschiedensten Preis- und Leistungsklassen sowie als Public Domain Software verfügbar. Schwerpunkt der Anwendung ist heute immer noch das Viewing und Markup von 2D-Zeichnungsableitungen, obwohl insbesondere seit dem Einzug der Internet-Formate (z.B. VRML) das 3D-Viewing zunehmend an Popularität gewinnt (vgl. Bild B012caxZ und B013caxZ).

Grundlage für das Viewing sind i.d.R. neutrale Ausgabeformate, die vorab für ein Native-Format (z.B. Unigraphics Part File) auf irgendeine Weise erzeugt werden müssen. Ein guter Viewer unterstützt häufig eine ganze Liste von Raster- und Vektor-Formaten (CGM, TIFF, HPGL, GIF, JPEG, DXF, usw.).

Falls ein Prüfer mittels Viewing irgendeinen Mangel feststellt und dies mit einem Kommentar dokumentieren möchte, kann er anhand der Markup-Funktionalität ein sogenanntes Redlining ausführen. Hierbei werden häufig Texte, Kreise, Linienzüge und andere geometrische Elemente in mehrheitlich roter Farbe wie auf einem separaten Layer über die 2D-Darstellung gelegt und schlussendlich möglichst in ein separates File gespeichert.

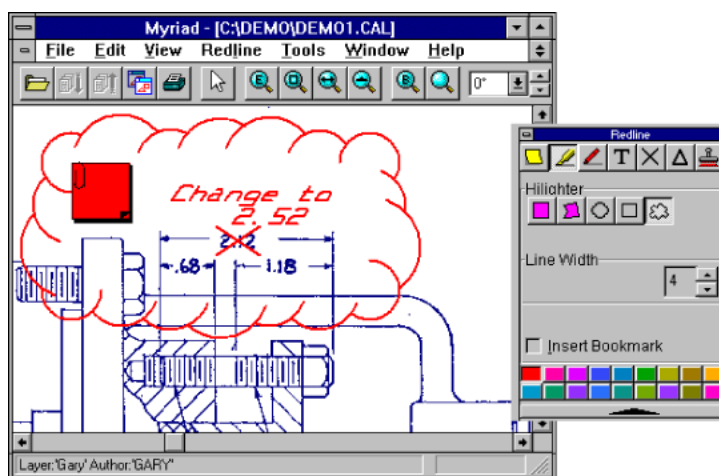


Bild (B012caxZ) Beispiel für 2D-Viewing und Markup

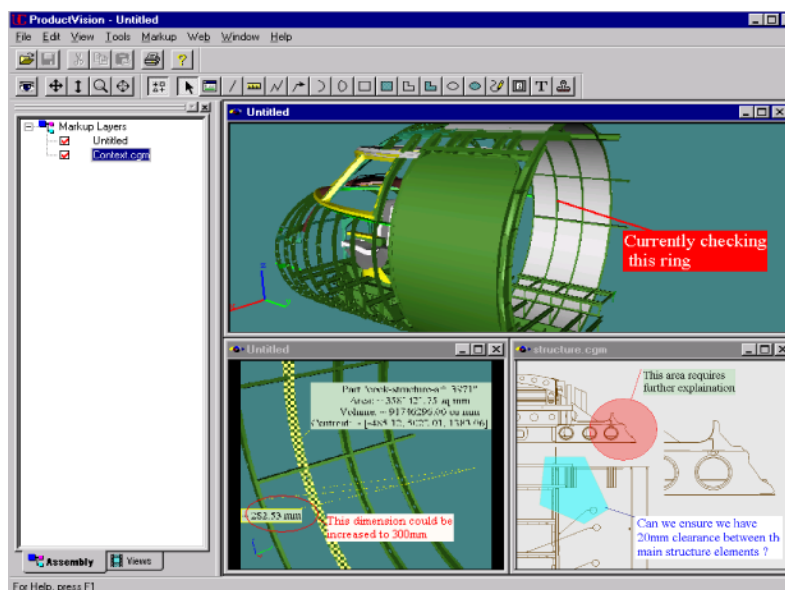


Bild (B013caxZ) Beispiel für 3D-Viewing und Markup

Für den Fall, dass mehrere Personen zu einem Image-Format ein Markup erstellen, ist bezüglich der Handhabung der Markup's durch das PLM-System die Anforderung zu stellen, dass von einem Image-Objekt eine 1:n-Beziehung zu mehreren Markup-Objekten unterstützt wird. Hierdurch kann mittels gegebener Autorisierungsfunktionalität des PLM-Systems (vgl. Kap. 4.5) sichergestellt werden, dass die verschiedenen beteiligten Personen nur jeweils auf ihr eigenes Markup Schreibzugriff besitzen und sich nicht gegenseitig die Markup-Kommentare überschreiben (vgl. Bild B014caxZ).

Um eine solche Handhabung von Markups zu ermöglichen sind folgende grundsätzliche Funktionen zu unterstützen:

- Markup: Erzeugen eines Markups bezüglich einem Dokument,
- View Markup: Anzeigen eines (oder aller) Markups bzgl. einem Dokument (Image File),
- List Markups: Auflisten sämtlicher Markups zu einem Dokument,
- Update Markup: Synchronisieren der Markups mit dem neuesten Dokument (Image File).

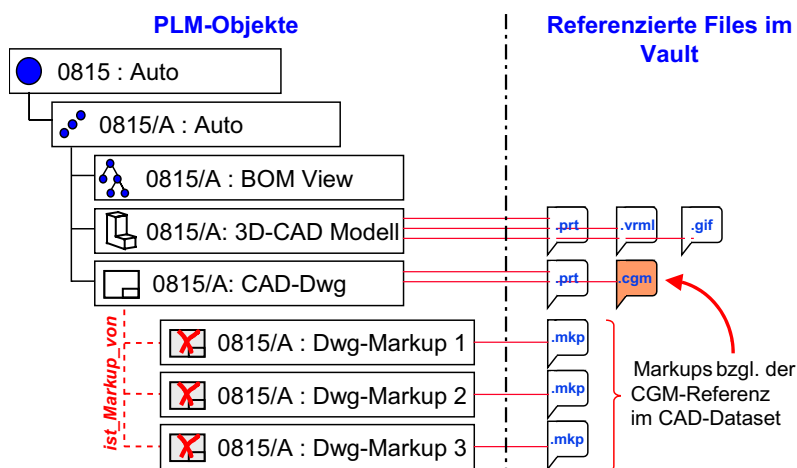


Bild (B014caxZ) Zuordnung von Markup's als eigenständige PLM-Objekte

Diese und weitere Dokumentenmanagement-Funktionen werden derzeit in dem Industrie-Standard ODMA (Open Document Management API) festgeschrieben, der es zukünftig ermöglichen soll, PLM-Systeme möglichst einfach und einheitlich mit beliebigen View & Markup-Tools zu kombinieren.

## 6 Prozessbezug von PLM-Applikationen

Um den Prozessbezug von PLM-Applikationen bzw. die Unterstützung der rechnerintegrierten Produktentwicklung beurteilen zu können, ist zunächst eine Gegenüberstellung von herkömmlicher, dokumentbasierender und rechnerintegrierter Produktentwicklung anzustellen.

Bei der konventionellen Vorgehensweise ist hervorzuheben, dass als Arbeitsergebnisse Dokumente erzeugt werden, die als Eingangsinformationen für nachfolgende Prozessschritte dienen. Mangels Integration einzelner Prozessschritte müssen hierbei gegebenenfalls Informationen wiederholt erarbeitet werden, wie z.B. im Falle eines Modellbaus, wofür eine Rückgewinnung von CAD-Informationen durch Digitalisierung erforderlich sein kann.

Im Gegensatz dazu werden bei der rechnerintegrierten Produktentwicklung keine Dokumente erzeugt, sondern es wird gemeinsam vom Entwicklungsteam ein rechnerinternes Produktmodell erarbeitet, womit zunehmend sämtliche Produkteigenschaften simuliert und visualisiert werden können. Bei Bedarf können hiervon mehr oder weniger automatisiert verschiedene Arbeitsdokumente abgeleitet werden, falls z.B. eine Technische Zeichnung oder eine Stückliste in Papierform benötigt wird.

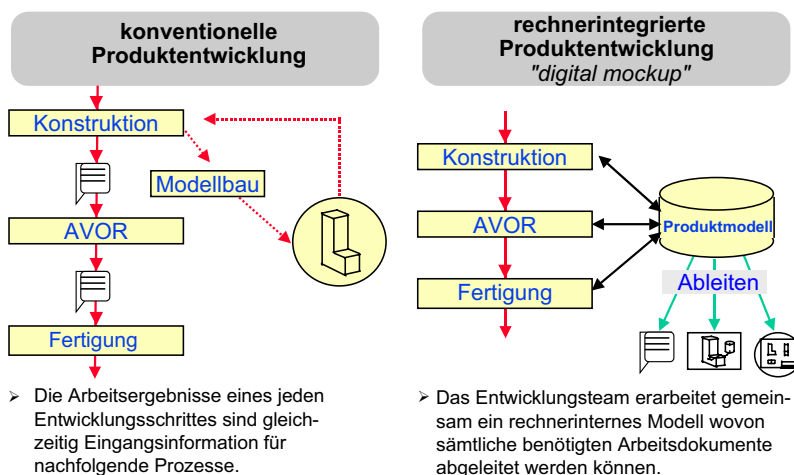


Bild (B015caxZ) Verdeutlichung der integrierten, modellbasierenden Vorgehensweise

Für eine systemtechnische Unterstützung der rechnerintegrierten Produktentwicklung ist insbesondere der Informationsaustausch zwi-

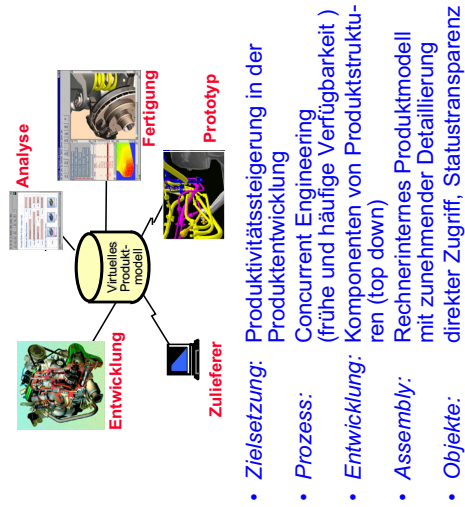
schen den einzelnen Prozessschritten und dem Produktmodell von Bedeutung. Dies bedingt ein effizientes Zusammenspiel zwischen den CAx-Erzeuger-Applikationen und dem PLM-Verwaltungssystem und damit eine grösstmögliche Integrationstiefe.

In bezug auf diese, für die Unterstützung des Produktentwicklungsprozesses entscheidende Integrationstiefe lassen sich PLM-Applikationen in zwei charakteristische Systemgruppen einteilen:

Bevor sich die ursprünglich eher als elektronische Zeichenbretter zu beschreibenden CAx-Applikationen zu den heutigen komplexen 3D-Produktmodellierern mit z.B. Assembly-Funktionalität entwickelt haben, lag die Zielsetzung von Verwaltungslösungen rein in der Administration von Files (Vaulting mit passivem Datenzugriff). Die in diesen Files hinterlegten realen Anwendungsobjekte sind auf der Verwaltungsebene aufgrund mangelnder Integrationstiefe nicht transparent. Demzufolge kann bei der Produktentwicklung nur eine bottom up-Vorgehensweise angewendet werden, wobei ausgehend von den Einzelteilen schrittweise der Baugruppenzusammenhang erarbeitet werden muss.

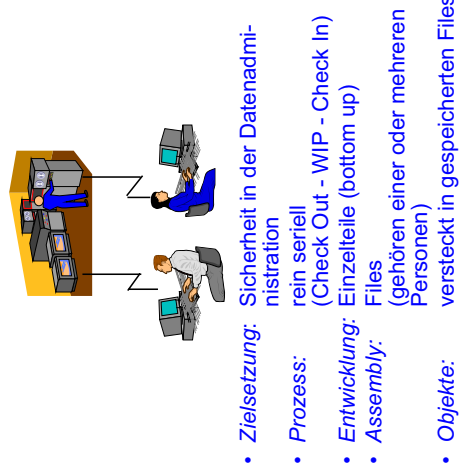
Demgegenüber können bei einer tiefen Integration zwischen PLM und CAx direkt Komponenten von Produktstrukturen entwickelt werden, da diese als transparente Objekte zur Verfügung stehen (interaktive Datennutzung). Basierend auf einer Interoperabilität der Sperrmechanismen kann mittels Concurrent Engineering Methoden ein hochgradiger Parallelzugriff realisiert werden, wodurch erhebliche Verbesserungen hinsichtlich Fehlerhäufigkeit und Durchlaufzeiten realisierbar werden. Wesentlich ist weiterhin die Möglichkeit zur Entwicklung von Produktstrukturen nach einer top down-Vorgehensweise, worauf aufbauend die CAD-Modellierung beliebig ausgeführt werden kann.

**Interaktive Datennutzung**  
(Concurrent Engineering)



- **Zielsetzung:** Produktivitätssteigerung in der Produktentwicklung
- **Prozess:** Concurrent Engineering (frühe und häufige Verfügbarkeit)
- **Entwicklung:** Komponenten von Produktstrukturen (top down)
- **Assembly:** Rechnerinternes Produktmodell mit zunehmender Detaillierung
- **Objekte:** direkter Zugriff, Statustransparenz

**Passiver Datenzugriff**  
(Vault basierende Vorgehensweise)



- **Zielsetzung:** Sicherheit in der Datenadministration
- **Prozess:** rein seriell (Check Out - WIP - Check In)
- **Entwicklung:** Einzelteile (bottom up)
- **Assembly:** Files (gehören einer oder mehreren Personen)
- **Objekte:** versteckt in gespeicherten Files

Bild (B016caxZ) Charakterisierung von PLM-Applikationen bezüglich ihrer CAX-Integrationsstiefe



## 7 Zusammenfassung

Im Rahmen der rechnerintegrierten Produktentwicklung kommt der Integration von Erzeugerapplikationen (z.B. CAx) in PLM-Systeme grosse Bedeutung zu.

Mittels eines *Check In/Check Out-Mechanismus* in der PLM-Datenbank wird ein Objekt zwecks Änderung für eine exklusive Schreiboperation reserviert (Check-Out). Nach Beendigung dieses Updates wird das Objekt freigegeben und steht wieder allen Benutzern zu Verfügung (Check-Out). Somit wird ein Mehrfachzugriff auf ein Objekt und infolgedessen entstehende Inkonsistenz der Datenbasis verhindert.

*Concurrent Engineering* hat zum Ziel, Entwicklungstätigkeiten zu parallelisieren und somit Auftragsdurchlaufzeiten zu verkürzen. Bei der rechnerintegrierten Produktentwicklung ist das Ergebnis einer jeden Entwicklungstätigkeit nicht mehr ein Dokument, sondern ein rechnerinternes Produktmodell, wovon verschiedene Dokumente abgeleitet werden können. Das gesamte Entwicklungsteam kann gemeinsam am Produktmodell arbeiten. Parallelzugriffe werden von der Transaktionsverwaltung des zugrundeliegenden Datenbanksystems koordiniert.

Zwecks Visualisierung und Langzeitarchivierung werden ausgehend von den nativen Quellformaten datenerzeugenden Applikationen neutrale Formate erzeugt. Die Generierung neutraler Formate heisst *Rendering*. Bevor ein Objekt freigegeben wird, muss es begrüft werden. Dies geschieht im sogenannten *Viewing und Markup (Redlining)*. Gängige Viewing&Markup-Tools ermöglichen in der Regel einen Zugriff unabhängig von der ursprünglichen Erzeuger-Applikation, des vorliegenden Betriebssystems und der netzwerktechnischen Umgebung.

### Verständisfrage 1

Welche Hauptunterscheidung kann für Vorgänge des Prozessmanagements getroffen werden?

### Verständisfrage 2

Welcher Mechanismus unterstützt die arbeitsteilige Bearbeitung von Objekten (Produktdaten) in einem Entwicklungsteam?

### Verständisfrage 3

Was versteht man unter Check In/Check Out?

#### Verständisfrage 4

Für welche Objekte ist ein impliziter Check Out möglich und welchen Vorteil bringt dieser?

#### Verständisfrage 5

Was versteht man unter einem Check Out mit Export?

#### Verständisfrage 6

Welche Arten von Sperren unterscheidet man und wodurch unterscheiden sich diese?

#### Verständisfrage 7

Was versteht man unter Concurrent Engineering und worin liegt der Hauptnutzen?

#### Verständisfrage 8

Welche Grundvoraussetzung muss für Concurrent Engineering erfüllt sein?

#### Verständisfrage 9

Was versteht man unter Digital Mockup?

#### Verständisfrage 10

Welche Leistungsstufen unterscheidet man bei der Einbindung von Fremdapplikationen?

#### Verständisfrage 11

Welche Aspekte müssen zur Einkapselung einer Fremdapplikation mindestens erfüllt sein?

#### Verständisfrage 12

Welche minimale Kenntnis muss über die Daten- bzw. Speicherungsstruktur einer Fremdapplikation vorhanden sein und wofür wird diese benötigt?

#### Verständisfrage 13

Was sind die Vor- und Nachteile einer Einkapselung?

**Verständisfrage 14**

Welche Aspekte interessieren bei der Beurteilung der Integrationstiefe zwischen PLM und CAx?

**Verständisfrage 15**

Geben Sie Beispiele, womit sich der Sperrmechanismus zwischen CAx- und PLM hinterfragen lässt?

**Verständisfrage 16**

Geben Sie Beispiele, womit sich die Funktionalität zwischen CAx- und PLM hinterfragen lässt?

**Verständisfrage 17**

Skizzieren Sie ein mögliches Szenario für einen Mehrbenutzerzugriff im Concurrent Engineering?

**Verständisfrage 18**

Welche Methoden sind zum Kopieren von Objekten mit Beziehungen zu unterscheiden?

**Publikationsverzeichnis – Literatur**

- [DIN6772] *DIN 6772: Änderung von Dokumenten und Gegenständen*. Berlin: Beuth-Verlag 1981
- [EHS-91] *Eigner, M.; Hiller, C.; Schindewolf, S.; Schmich, M.: Engineering Database: strategische Komponente in CIM-Konzepten*. München: Hanser Verlag 1991
- [EIS-01] *Eigner, M.; Stelzer, R.: Produktdatenmanagement-Systeme: Ein Leitfaden für Product Development und Life Cycle Management*. Berlin: Springer Verlag 2001
- [VDI-72] *VDI-Fachgruppe Betriebstechnik (ADB): Elektronische Datenverarbeitung bei der Produktionsplanung und -steuerung III: Informations- und Stücklistenwesen*. VDI-Taschenbücher T 28, Düsseldorf: VDI-Verlag 1972

# Workflow Management

Autor: Dr. R. Montau



# 1 Überblick

Das Workflow Management stellt die dynamische Komponente von PLM-Systemen dar, worunter sämtliche ablaufbezogenen Vorgänge und Mechanismen zu verstehen sind, die systemseitig gesteuert werden. Hiervon abzugrenzen sind rein benutzerseitig initiierte Vorgänge, die sich wie folgt unterscheiden:

- benutzerseitige Anwendungsvorgänge  
Diese werden vom Benutzer nach eigenem Ermessen frei ausgeführt und lediglich durch universell einsetzbare Systemmechanismen unterstützt bzw. geführt (z.B. Editiervorgang).
- systemseitig gesteuerte formale Workflow-Prozesse  
Diese werden vom System gesteuert entsprechend einer vorgängig festgelegten Prozessdefinition, die meist unternehmensspezifischen Charakter hat und in der Regel eine Dokumentation sämtlicher Schritte erfordert (z.B. Freigabeprozess).

Formale Workflow-Prozesse unterscheiden sich von den benutzerseitig initiierten Anwendungsvorgängen somit durch die vorgängige Vorgabe einer Prozessdefinition. Zielsetzung ist die Einhaltung dieses fest vorgeschriebenen Ablaufs, was durch ein PLM-System überwacht und koordiniert werden soll.

Von den für PLM-Systeme relevanten produktbezogenen Prozessen sind insbesondere die folgenden beiden Prozessarten oder Bereiche zu beachten, welche im vorliegenden Abschnitt über das Workflow Management behandelt werden:

- Freigabewesen,
- Änderungswesen.

## 2 Freigabewesen

Das Freigabewesen ist von zentraler Bedeutung für die Produktentwicklung und dient der Überprüfung und Genehmigung von Objekten (z.B. Bauteilstamm oder Dokument) mit dem Ziel der Einfrierung des aktuellen Zustandes. Hierauf aufbauend kann mit weiterführenden Prozessschritten (Betriebsmittelkonstruktion, Fertigung etc.) begonnen werden. Die Freigabe erfolgt in der Regel durch die Projektleitung und ist ein entscheidender Meilenstein in der Produktentwicklung. Hierdurch muss der Konstrukteur sein Schreibrecht an den Produktdaten abgeben und deren Stand wird gemäss den Forderungen der Produkthaftung (vgl. Section 2.2) langfristig dokumentiert.

Das Freigabewesen wird im Rahmen der betrieblichen Ablauforganisation unternehmensspezifisch festgelegt, wobei gegebenenfalls verschiedene Prozessdefinitionen zu unterscheiden sind, für z.B. sicherheitsrelevante Teile oder Mechanik-/Elektronikkomponenten. PLM-Systeme erlauben in der Regel eine Konfiguration von Abläufen in Abhängigkeit der vorliegenden Objektklasse (z.B. Bauteilstamm, Technische Zeichnung) und Benutzergruppe (Entwicklungsteam 1, 2 etc.).

Mit der Freigabe, bzw. bereits mit der Initiierung eines Freigabeprozesses erfolgt ein Wechsel in die nächste Hauptphase des Lebenslaufes, womit sich insbesondere die Zugriffsrechte für die betroffenen Objekte ändern, was Vorrang hat vor jeglichen Default-Zugriffsrechten oder auch benutzerseitig geänderten Rechten. Mit herkömmlichen Anwendungsfunktionen eines PLM-Users (auch des Eigentümers) lassen sich diese im Prozessmodell definierten Rechte weder zurücksetzen noch modifizieren, da die elektronische Freigabe einen juristischen Charakter hat. Bezüglich der Autorisierung unterscheidet man

- Zuständigkeiten bzw. Berechtigungen für das Ausführen einzelner Ablauffunktionen,
- Zugriffsrechte auf Objekte, die meist in Abhängigkeit des Status' variieren.

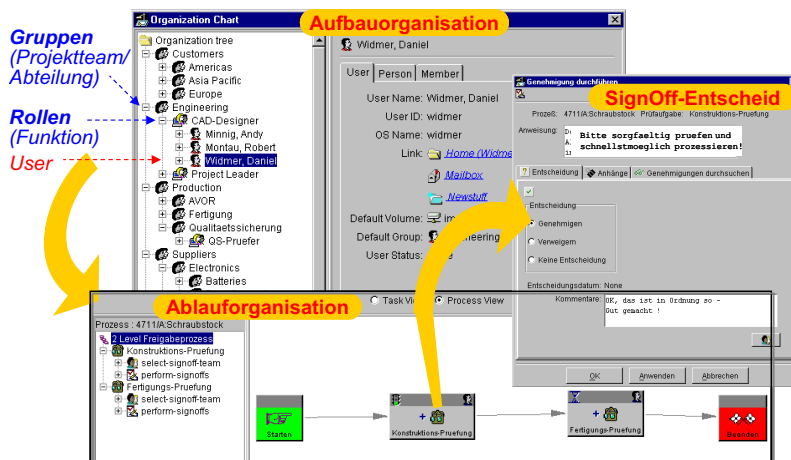


Bild (B001wfmZ) Zusammenhang von Aufbau- und Ablauforganisation

Der Zusammenhang zwischen Aufbauorganisation und Zuständigkeiten bzw. Zugriffsrechten in der Ablauforganisation wird durch Bild B001wfmZ verdeutlicht. Gemäss vorgegebener Prozessdefinition darf für einen „SignOff“ (elektronische Unterschrift für einen Prozessschritt) nur ein adäquater Benutzer als zuständig ausgewählt werden (z.B. Gruppe: Engineering/Rolle: Konstrukteur). Für dieses Accessor-Profil sind zudem gewisse Zugriffsrechte festgelegt. Zuständigkeiten und Zugriffsrechte von Freigabeprozessen haben somit einen direkten Bezug zur Aufbauorganisation.

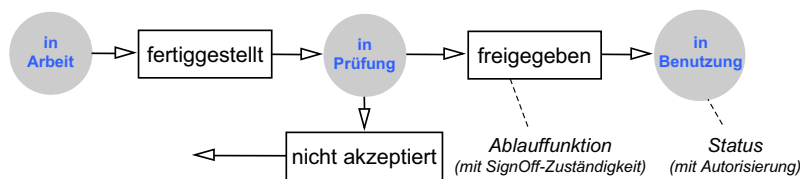


Bild (B002wfmZ) Beispiel für eine allgemeine Ablaufmodellierung

Um die Dynamik eines Prozesses beschreiben zu können, ist ein detailliertes Ablaufmodell erforderlich, welches das dynamische Verhalten in den Vordergrund stellt. Ablauffunktionen und Stati stellen dabei die Grundelemente des Ablaufmodelles dar, welches im Gegensatz zur fast ausschliesslich statischen Informationsmodellierung ein dynamisches Verhalten beschreibt. *Ablauffunktionen* repräsentieren Zeitpunkte, die durch eine einzelne Benutzeraktion (vgl. Drücken der



Enter-Taste) vorstellbar sind. Demgegenüber beschreibt ein Status einen Zustand, der für einen gewissen evtl. längeren Zeitraum Gültigkeit besitzt.

Im Rahmen der Definition eines Freigabeprozesses sind die Zugriffsrechte auf die Objekte in der Regel fest vorgegeben. Hierfür sind häufig für mehrere Benutzergruppen statusabhängig unterschiedliche Zugriffsrechte erforderlich. Hinsichtlich der Berechtigungen bzw. Zuständigkeiten finden sich mehrheitlich fest vordefinierte Prozesse, in welchen z.B. die Prüfpersonen fest vorgegeben sind. Gegebenenfalls werden die Zuständigkeiten jedoch erst zur Laufzeit des Freigabeprozesses bestimmt, wobei die Auswahl durch verschiedene Randbedingungen (Benutzergruppe, Rolle) eingeschränkt ist.

Erwähnt seien an dieser Stelle die möglicherweise bestehenden Abhängigkeitsverhältnisse zwischen Objekten, die durch verschiedene Objektrelationen hergestellt werden. Hierdurch kann die Freigabe einer Objektinstanz (z.B. Baugruppe) von gewissen Bedingungen abhängig gemacht werden (z.B. alle untergeordneten Baugruppen/ Einzelteile müssen freigegeben sein). Desweiteren können gewisse andere Objekte (z.B. BOMView, 3D CAD-Modelle) automatisch mitfreigegeben werden, ohne dass diese explizit zur Freigabe ausgewählt werden müssen. Ergebnis einer Freigabe ist in jedem Fall der Entzug jeglicher Schreibrechte auf ein Objekt und die Zuweisung einer charakterisierenden Statusinformation, wie z.B.: „Released“.

Um das Systemverhalten bei der Ausführung einer Ablauffunktion an anwendungsspezifische Anforderungen anpassen zu können, stellen PLM-Systeme generische Workflow Features zur Verfügung, die über sogenannte Rule Handler (Überprüfen einer Bedingung) oder Action Handler (Ausführen einer Aktion) angestoßen werden (vgl. Bild 8.3).

- **Notification**
  - Versenden von Mails an User, Gruppen, Rollen
- **Zeitbezogene Triggerung**
  - Definition eines Endtermins (Deadline = "Due Date") oder einer Dauer "Duration" für eine Workflow Task
  - Bei Verletzung
    - Send Mail zu User oder Adressliste
    - Kennzeichnung von "Late Tasks" (z.B. farblich)
- **Resource Pool**
  - Direkt-Zuweisung von Tasks an Gruppen, Rollen etc.
  - Jeder qualifizierte User kann eine Task ausführen, die einem Resource Pool zugewiesen ist
- **"Out of Office" Assistent**
  - Weiterleiten von Tasks zu Stellvertreter(n), während Abwesenheit
- **Bedingungsgesteuerte Prozessverzweigungen**
  - Steuerung von Verzweigungen im Prozessablauf mittels Bedingungen (z.B. True/False Evaluation einer DB-Query)
- **Mehrheitsentscheide**
  - Nur eine Untermenge aller SignOff Personen muss zustimmen

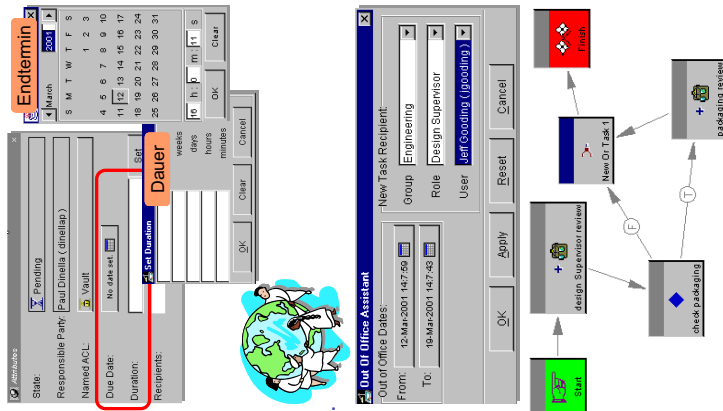


Bild (B003wfmZ) Häufig benötigte Features im Workflow Management

Im Rahmen eines Freigabeprozesses werden in bezug auf die freizugebenden Objekte häufig noch eine Reihe untergeordneter Prozessschritte ausgeführt, welche an dieser Stelle jedoch nicht erläutert werden sollen:

- Generierung eines neutralen Formats zur Langzeitarchivierung (vgl. Section 7),
- Viewing und Markup (Redlining) von Dokumenten (vgl. Section 7),
- Datenübergabe an ein ERP-System.

Falls einmal freigegebene Objekte in irgendeiner Form modifiziert werden müssen, bedarf dies einer umfangreichen Änderung, deren Ablauf im Rahmen des betrieblichen Änderungswesens festgelegt ist.

### 3 Änderungswesen

Aufgrund externer und interner Störeinflüsse ist die Produktentwicklung in der industriellen Praxis meist unvorhersehbaren Änderungen unterworfen. Obwohl eine jede Änderung nicht nur kostspielig ist, sondern insbesondere die Gefahr einer nicht vollständigen Beseitigung der im Umlauf befindlichen überholten Dokumente mit sich bringt, ist diesbezüglich in der Industrie überraschenderweise eine grosse Bereitschaft und Entschlussfreudigkeit zu.

Das Änderungswesen  
Wir ändern morgen, wir ändern heut,  
wir ändern wütend und erfreut.  
Wir ändern, ohne zu verzagen,  
an fünf von sieben Wochentagen.  
Wir ändern teils aus purer Lust,  
mit Vorsatz teils, teils unbewusst.  
Wir ändern gut und auch bedingt,  
weil ändern immer Arbeit bringt.  
Wir ändern resigniert und still,  
wie jeder es so haben will.  
Die Alten ändern und die Jungen,  
wir ändern selbst die Änderungen.  
Wir ändern was man ändern kann,  
und stehen dabei unsern Mann.  
Ist auch die Konstruktion gelungen,  
bestimmt verträgt sie Änderungen.  
Wir ändern deshalb früh und spät,  
alles, was zu ändern geht.  
Für Änderungen stets bereit,  
zum Denken bleibt uns wenig Zeit.

(Änderungen vorbehalten)  
Wopfli

Ein Änderungsprozess besteht in der Regel aus mehreren Prozessschritten, wobei diverse Genehmigungsphasen durchlaufen werden müssen und verschiedene Änderungsdokumente entstehen (vgl. Bild B004wfmZ).

Ausgangspunkt ist irgendeine Störfallmeldung, die in Form eines Änderungsantrages eingereicht werden muss. Der Änderungsantrag beinhaltet zumindest die Ursache der Änderung, die Dringlichkeits-

stufe und die betroffenen (freigegebenen) Objekte. Weitere Angaben können die Austauschbarkeit betreffen oder die schätzungsweise anfallenden Kosten und sich ergebenden Termine.

Im Rahmen eines ersten Genehmigungsschrittes muss dieser Änderungsantrag zunächst genehmigt werden, woraufhin eine sogenannte Änderungsanweisung ausgearbeitet wird. Diese beinhaltet eine detaillierte Beschreibung der auszuführenden Änderungen und geht an einen oder auch mehrere Entwicklungsingenieure, welche daraufhin die Änderungen durchführen.

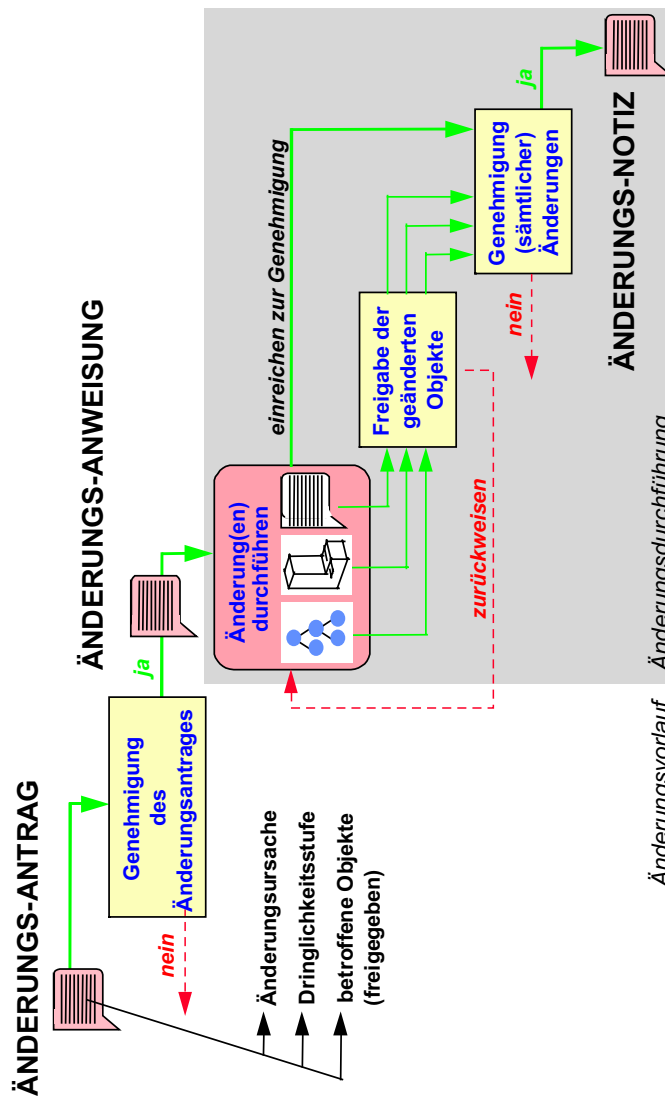


Bild (B004wfmZ) Prinzipieller Ablauf einer Änderung

Nach Abschluss einer jeden Änderung an einem Objekt, muss dieses im Rahmen eines untergeordneten Freigabeprozesses begutachtet und freigegeben werden.

Sind sämtliche Änderungen durchgeführt, so kann der gesamte Vorgang zur abschliessenden Genehmigung eingereicht werden. Im Falle eines positiven Entscheides werden die Änderungen entweder sofort oder mit einer bestimmten Effektivität (Datum, Serien-Nr.; vgl. Section 4.3) wirksam und an sämtliche involvierten Mitarbeiter im Entwicklungsteam und in den nachfolgenden produzierenden Bereichen ergeht eine abschliessende Änderungsnotiz (Bild B004wfmZ).

Im Gegensatz zum Freigabewesen entstehen Änderungen aufgrund unvorhersehbarer Störeinflüsse, weshalb sich diese weniger an der Aufbauorganisation orientieren. Beispielsweise können seit der Freigabe des zu ändernden Objekts die Zuständigkeiten gewechselt haben, durch ein Hinzukommen oder durch einen Weggang von Mitarbeitern sowie durch organisatorische Umstrukturierungen. Aus diesen Gründen werden die Zuständigkeiten für Ablauffunktionen im Änderungswesen in der Regel ohne Rücksicht auf Gruppen- und Rol-  
lendefinitionen festgelegt.

Charakterisieren lassen sich Änderungen einerseits nach sogenannten Dringlichkeitsstufen, andererseits hinsichtlich der Austauschbarkeit.

Als *Dringlichkeitsstufen* unterscheidet man:

- funktionswichtige Änderungen,
- konstruktive bzw. fertigungstechnische Verbesserungen und
- zeichnerische u.a. Richtigstellungen [[Wie-89]].

Als Folge hiervon ergibt sich eine unterschiedliche Einführung bzw. Einsteuerung der geänderten Objekte in den Produktentwicklungs- bzw. Herstellungsprozess (Einführung sofort, Lagerbestände aufbrauchen, Einführung mit Serien-Nr. x oder zum Datum y).

Die *Austauschbarkeit* beschreibt demgegenüber die Eignung eines Objektes, ein anderes Objekt ohne irgendeine Veränderung, Anpassung oder Auswahl von dritten Objekten zu ersetzen und dabei keinerlei Funktions- oder Betriebsmerkmale zu beeinträchtigen ([DIN 6772, VDI-72]), vgl. Tabelle T001wfmZ).

Im Falle einer austauschbaren Änderung wird lediglich eine neue Revision (neuer Änderungszustand) desselben Objektes erzeugt, das sich bei gleichbleibender Sachnummer nur um einen Änderungsindex unterscheidet. Aufgrund der vollumfänglichen Austauschbarkeit wird die überholte Revision in der Regel in sämtlichen Verwendungen durch die neue ersetzt. Hierfür bieten PLM-Systeme das Konzept der

dynamischen Referenzierung auf die jeweils neueste Revision eines Objektes an.

Ist nach Abschluss einer Änderung das geänderte Objekt mit dem vorherigen nicht beliebig austauschbar, so handelt es sich um ein neues Objekt, welches eine neue Identifizierung (Sachnummer) erhält. Je nach Anwendungsfall wird das neue Objekt das alte nur in einigen Verwendungen ersetzen, in der Regel jedoch nicht in allen.

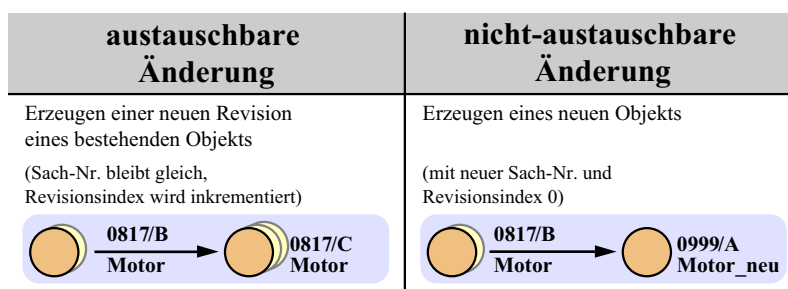
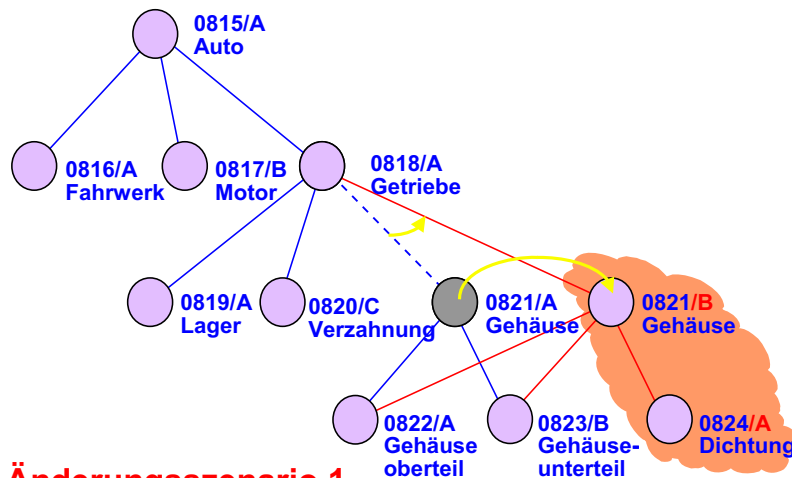


Bild (B005wfmZ) Unterscheidung von Änderungen

Aufgrund der grossen Bedeutung der Austauschbarkeit soll nachfolgend für jede Änderungsart ein exemplarisches Anwendungsbeispiel diskutiert werden.

Bild B006wfmZ verdeutlicht den einfacheren Fall einer austauschbaren Änderung am Beispiel eines Getriebegehäuses, welches Öl verliert. Hierbei sei angenommen, dass lediglich eine Dichtung eingebaut werden muss, wofür keine Änderung an den beiden Gehäusehälften erforderlich ist. Da das Gehäuse anschliessend ein weiteres Unterteil beinhaltet, muss dieses geändert werden. Das neue Gehäuse mit Dichtung kann das alte Gehäuse problemlos in sämtlichen Verwendungen ersetzen (hier im: Oberteil 0818/A Getriebe). Da das alte Gehäuse infolgedessen nicht mehr benötigt wird, handelt es sich um eine austauschbare Änderung, wobei lediglich der Revisionsindex inkrementiert werden muss (von 0821/A auf 0821/B).



### Änderungsszenario 1

**Vorfall:** Getriebegehäuse 0821/A ist undicht!

**Lösung:** Dichtung 0824/A wird zusätzlich eingebaut.

Gehäusehälften benötigen keiner Änderung und bzgl. dem Gehäuse ist die **Änderung austauschbar**.

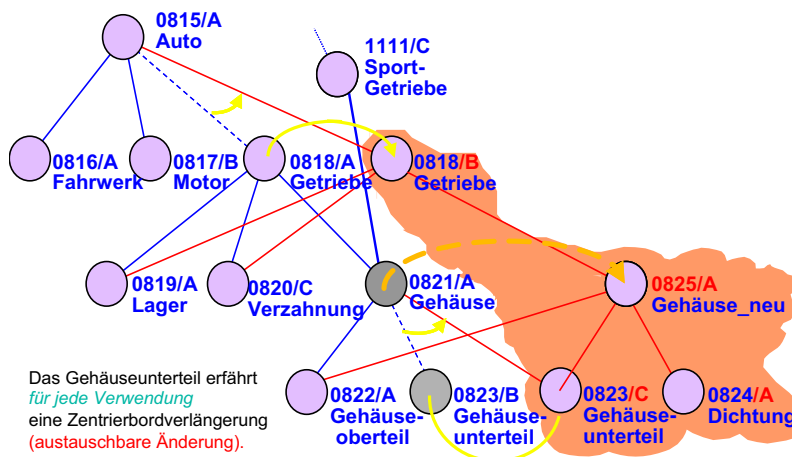
**Resultat:** Für das Gehäuse entsteht eine **neue Revision 0821/B**.

Bild (B006wfmZ) Beispielszenario einer austauschbaren Änderung

Eine nicht-austauschbare Änderung ist demgegenüber immer mit einer etwas komplexeren Problematik verbunden (Bild B007wfmZ). Diesbezüglich sei angenommen, dass der Einbau einer Dichtung nicht für sämtliche Verwendungen möglich ist, da in einem weiteren Ober- teil des Gehäuses (1111/C Sport-Getriebe) z.B. die thermischen Ver- hältnisse dies nicht zulassen. Dieselbe Änderung des Gehäuses 0821/ A ist in diesem Beispiel daher nicht austauschbar und ergibt somit ein neues Objekt (0825/A Gehäuse\_neu), weil nach der Änderung im Sport-Getriebe nach wie vor das alte Gehäuse ohne Dichtung verwen- det werden muss. Für das Getriebe 0818 muss aufgrund des neuen Unterteils eine neue Revision 0818/B eingeführt werden. Nach Abschluss der Änderung wird im Getriebe 0818/B das Gehäuse\_neu (mit Dichtung) verwendet und im Ober- teil 1111/C Sport-Getriebe das Gehäuse 0821/A.

Zusätzlich sei hier noch angenommen, dass am Gehäuseunterteil 0823/B eine Zentrierbordverlängerung erforderlich ist, die in sämt- lichen Verwendungen problemlos eingesetzt werden kann und somit eine austauschbare Änderung darstellt. Das mit einem inkrementier- ten Änderungsindex versehene Gehäuseunterteil 0823/C wird daher sowohl im Gehäuse 0821/A als auch im Gehäuse\_neu 0825/B ver- wendet. Diese austauschbare Änderung ist von rein zusätzlichem Cha-

rakter, d.h. hat keinen Einfluss auf die Unterscheidung der Änderungsszenarien in den Bildern 8.5 und 8.6. Hierdurch soll jedoch ein Eindruck für die in der Praxis häufig vorkommenden komplexen Problemstellungen vermittelt werden, wofür ein fundiertes Verständnis dieser Mechanismen zwingend erforderlich ist.



## Änderungsszenario 2

**Vorfall:** Getriebegehäuse 0821/A ist undicht!

**Lösung:** Dichtung 0824/A wird eingebaut. In einer anderen Verwendung des Gehäuses kann diese Dichtung nicht verwendet werden, d.h.: **nicht austauschbare Änderung!**

**Resultat:** Es entsteht ein **Gehäuse\_neu 0825/A** sowie eine neue Revision für das übergeordnete Getriebe **0818/B**.

Bild (B007wfmZ) Beispielszenario einer nicht-austauschbaren Änderung

Um einen Einblick in die Abbildung eines Änderungsprozesses auf ein PLM-System zu geben, wird in Bild B008wfmZ gezeigt, wie für das obigen Szenario 2 in der Ausgangs-Produktstruktur ein Änderungsprozess definiert wird.

Festgestellt wird das Problem für das Getriebegehäuse 0821/A, wobei hierfür im Szenario 2 der Nicht-Austauschbaren Änderung ein neues Gehäuse (0825/A Gehäuse\_neu) eingeführt werden muss. Dies bedingt eine Revisionserhöhung auf der darüberliegenden Getriebe-Strukturstufe, welche von 0818/A auf 0818/B reversioniert werden muss, um ein neues Teil (0825/A Gehäuse\_neu) einfügen zu können. Daher wird in Bild B007wfmZ die Getriebe-Revision 0818/A als sog. Problem-Item definiert und die resultierende die Getriebe-Revision 0818/B als sog. Affected Item. Die neu erzeugten Teile sowie die weiteren neu angelegten Revisionen werden als sog. Solution Items ange-



geben (0825/A, 0824/A, 0823/C), welche allesamt durch den Änderungs-Workflow freigegeben werden und i.d.R. zusammen eine bestimmte Effectivity erhalten.

Der Zusammenhang der einzelnen Teile im Kontext der Änderungsdefinition wird üblicherweise über unterschiedliche Beziehungen im PLM-Datenmodell abgelegt, welche hiernach flexibel abgefragt und analysiert werden können.

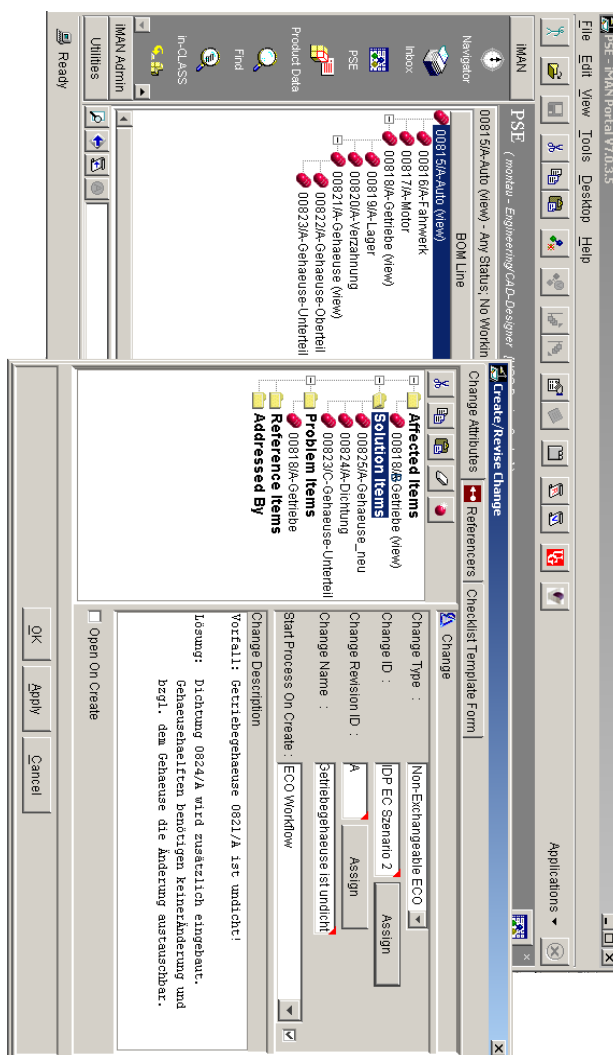


Bild (B008wfmZ) Definieren einer Änderung im PLM-System (obiges Szenario 2)

Nach erfolgreichem Abschluss der Änderung kann im PLM-System die „Ersatz für/Ersetzt durch“ Ablösehistorie in der Produktstruktur untersucht und graphisch dargestellt werden. Für das Getriebe wurde die

bisherige Komponente „0821/A Gehäuse“ durch die Änderung Szenario 2 mit der neuen Komponente „0825/A Gehäuse\_neu“ abgelöst bzw. ersetzt (Supercedure-Relation).

In einem ersten Schritt führt der User die BOM Add- und Cancel-Modifikationen aus, was vom PLM-System überwacht bzw. automatisch protokolliert werden kann. Um eine Supercedure-Beziehung auch im Falle mehrerer Additionen und Subtraktionen auswerten zu können, muss der User anschliessend manuell eine „Ersetzt durch/ Ersatz für“ - Zuordnung vornehmen, um die zugehörigen Komponenten zu verbinden (Bild B009wfmZ).

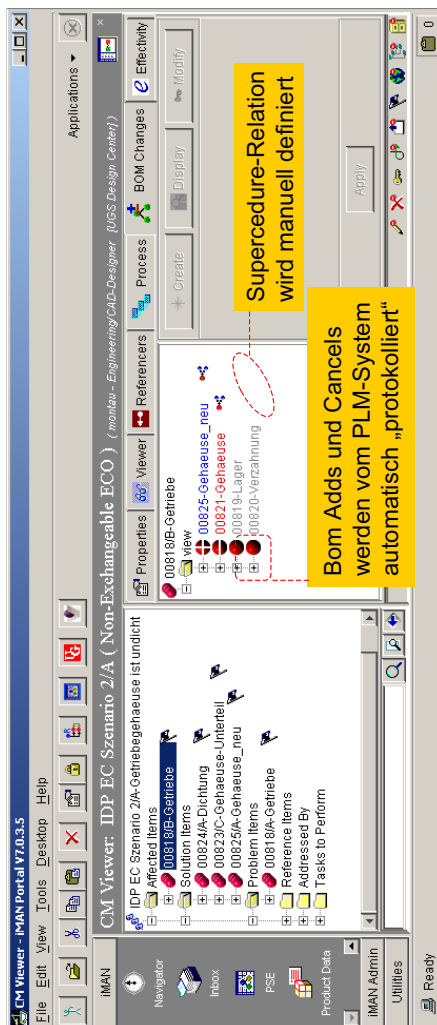


Bild (B009wfmZ)

Unterscheidung von BOM Add und Cancel gegenüber Supercedure-Relation

Im PDM-Datenmodell entspricht die Ablösung dem Navigieren entlang der Superscedure-Relation in Richtung „Supersceded by“, d.h. Ersetzt durch. Die im rechten oberen Bereich in Bild B010wfmZ dargestellte Ablösehistorie kann hierbei sowohl ausgehend vom neuen Komponententeil (0825/A) abgefragt werden als auch vom alten Komponententeil (0821/A), was jeweils im Kontext der Produktstruktur erfolgen muss. Bild B010wfmZ zeigt somit das Analysieren der strukturellen Einflüsse eines Änderungsprozesses auf der Basis der sogenannten BOM Changes, d.h. BOM Additionen und Subtraktionen.

Nachdem in einem ersten Schritt die BOM Add und Cancel Modifikationen ausgeführt wurden und in einem zweiten Schritt manuell etwaige Superscedure-Zuordnungen vorgenommen wurden, ist hierauf aufbauend eine graphische Analyse der strukturellen Ablösehistorie einer Änderung (dargestellt als gelbe Ellipse in Bild B010wfmZ) möglich.

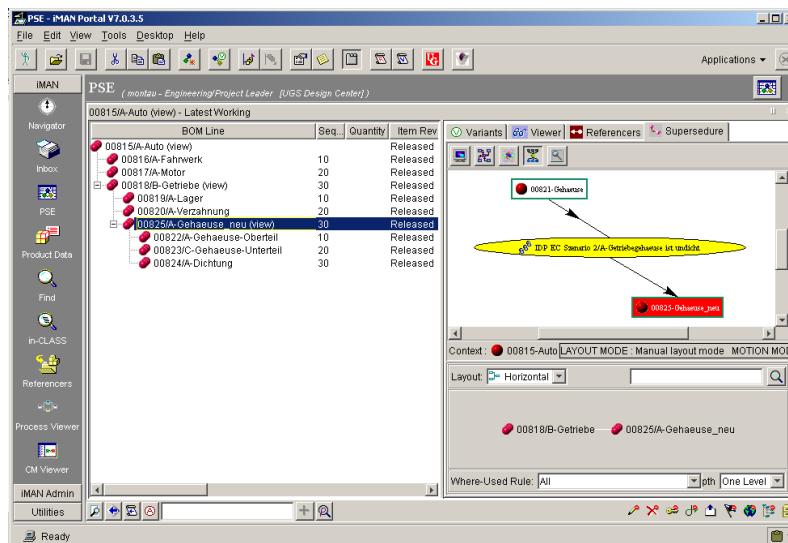


Bild (B010wfmZ) Graphische Analyse der strukturellen Ablösehistorie einer Änderung

Gesamthaft können an einer BOM-Struktur nicht nur Add und Cancel, sondern prinzipiell die folgenden Änderungen durchgeführt werden:

- BOM Add
- Hinzufügen eines Einzelteils oder einer Baugruppe.
- BOM Cancel
- Löschen eines Einzelteils oder einer Baugruppe.
- Move
- Eine Komponente wird geometrisch im Raum verschoben, d.h. erhält neue Werte für die Transformationsmatrix, welche in der PLM-Datenbank gespeichert wird.
- Reshape
- Eine Komponente wird geometrisch verändert, d.h. erhält eine neue Form, welche durch unterschiedliche Visualisierungs-Datasets sichtbar werden.

Falls ein PLM-System sämtliche prinzipiell möglichen Änderungen (Add, Cancel, Move, Reshape) spezifisch handhaben kann, sind auch weitergehende graphische Analysen von Änderungen möglich. Bild B011wfmZ zeigt die graphische Untersuchung der Änderungseinflüsse zwischen zwei Revisionen einer Baugruppe.

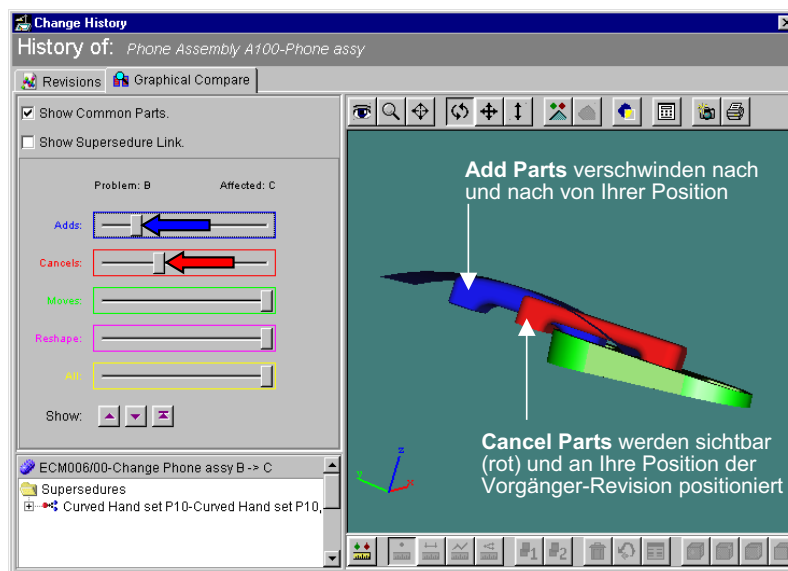


Bild (B011wfmZ) Analyse der graphischen Auswirkung einer Änderungshistorie

### 3.1 Methoden zum Kopieren von Objekten zur Revisionierung

Darüber hinaus können für jedes zu ändernde Anwendungsobjekt die betroffenen Informationsmengen variieren, wobei in den Beispielen von Bildern B004wfmZ und B006wfmZ bislang lediglich die Produktstrukturen (Bauteilstammdaten) betrachtet wurden. Einer Bauteilrevision sind in der Regel jedoch verschiedene weitere Informationen zugeordnet (z.B. Datasets), welche ebenfalls von der Änderung betroffen sein können. Beispielsweise kann bei Festigkeitsproblemen entweder der Werkstoff geändert werden oder die Geometrie, was eine Modifikation des CAD-Modells erfordern würde. Bild B008wfmZ verdeutlicht diesen Zusammenhang, wobei zunächst zweierlei Vorgehensweisen zu unterscheiden sind:

- Kopieren auf neue Revision
- Es wird eine neue Revision eines Objektes erzeugt, welche die alte in der Produktstruktur ersetzt und wahlweise bestehende Objektbeziehungen (Strukturbeziehungen zu untergeordneten Baugruppen und/oder Zuordnungsbeziehungen) mitführt.
- Kopieren auf neues Objekt

Von entscheidender Bedeutung für beide obigen Vorgehensweisen ist hierbei die Art des Mitführens der Beziehungen, bzw. was tatsächlich kopiert wird: Es wird ein neues Objekt erzeugt, wobei Objektbeziehungen (Strukturbeziehungen zu untergeordneten Baugruppen und/oder Zuordnungsbeziehungen) wahlweise mitgeführt werden.

- Kopieren von Objekten und deren Referenzen  
Da sowohl Produktstrukturobjekte als auch Container-Objekte (Datasets mit zugeordneten physischen Files) eigenständige Objekte sind, die einander über Referenzen zugeordnet sind, bleibt beim Kopieren dieser Referenzen (mittels copy) nach wie vor derselbe Dataset und damit dieselbe Datei zugeordnet! Infolgedessen würde eine Modifikation des CAD-Modelles der neuen Revision (bzw. des neuen Objektes) gleichermassen auch die ursprüngliche Objektrevision betreffen, da beide denselben Dataset referenzieren!
- Kopieren von Objekten und referenzierten Dateien  
Falls Produktstrukturobjekte und Dataset-Objekte kopiert werden (mittels save\_as), existieren nach dem Kopiervorgang weitere physische Dateien, wodurch die Produktstrukturobjekte völlig unabhängig voneinander sind und Änderungen - auch der CAD-Geometrie - sich nicht auf das andere Objekt auswirken.

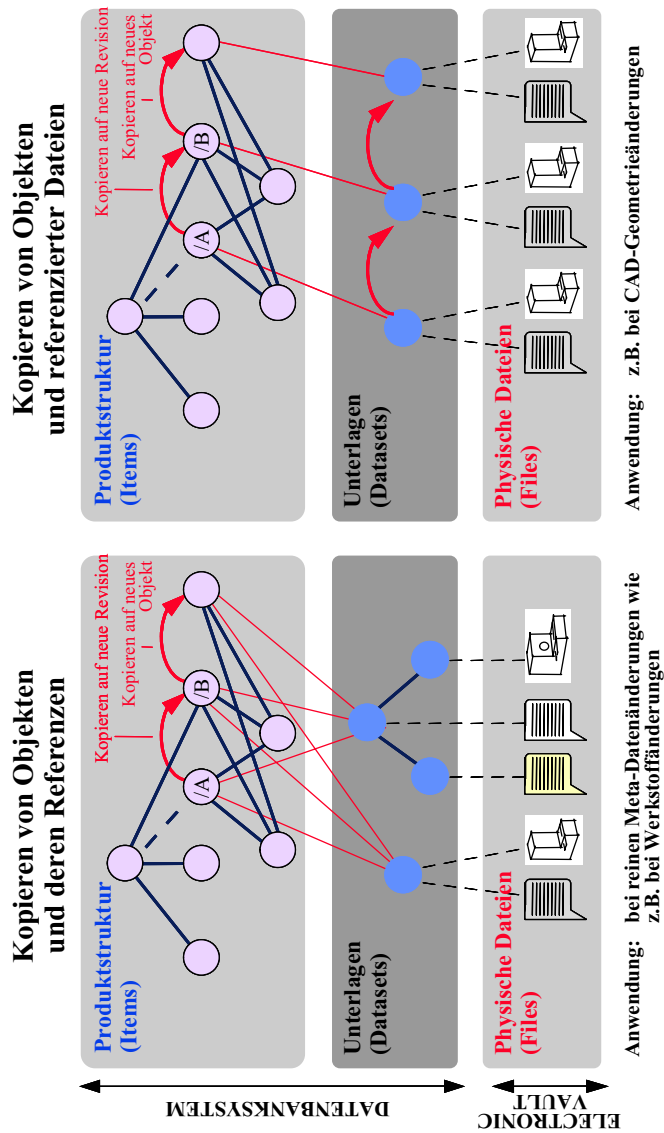


Bild (B012wfmZ) Methoden zum Kopieren von Objekten mit Beziehungen

Für das Beispiel der zusätzlichen Änderung des Gehäuseunterteils 0823 in Bild B006wfmZ muss aufgrund der Zentrierbordverlängerung insbesondere die CAD-Geometrie modifiziert werden, weshalb hierfür ein Kopieren einschliesslich der referenzierten CAD-Modelldatei zu wählen ist. Sollte eine Änderung jedoch nur die relational verwalteten Meta-Daten betreffen, ist dagegen ein Kopieren der Referenzen sinnvoll, um unnötige Daten-Redundanzen zu vermeiden. Beide grundlegenden Methoden werden daher gleichermassen benötigt, um dem PLM-Anwender zweckmässige und leistungsfähige Mechanismen zur Verfügung zu stellen.

Die diskutierten Änderungsszenarien verdeutlichen die komplexen Zusammenhänge im Änderungswesen, die insbesondere im Falle herkömmlicher, dokumentbasierender Produktentwicklungen grosse Probleme aufwerfen können. Der Unterschied zwischen der konventionellen, dokumentbasierenden Vorgehensweise gegenüber der rechnerintegrierten Vorgehensweise und die Unterstützung der letzteren durch PLM/CAX-Technologien soll nachstehend daher nochmals verdeutlicht werden.

## 4 Zusammenfassung

Das *Workflow Management* in PLM-Systemen beinhaltet das Freigabe- und das Änderungswesen. Diese ablaufbezogenen Vorgänge und Mechanismen sind systemseitig gesteuert und werden von rein benutzerseitig initiierten Vorgängen abgegrenzt.

Im *Freigabeprozess* wird ein erstelltes Objekt überprüft und genehmigt, bevor sein aktueller Zustand eingefroren wird. Mit der Freigabe bzw. der Initiierung des Freigabeprozesses erfolgt ein Wechsel in eine neue Lebensphase. Die Zugriffsrechte für das Objekte werden neu vergeben.

Ein *Änderungsprozess* besteht in der Regel aus mehreren Prozessschritten, wobei diverse Genehmigungsphasen durchlaufen werden müssen und verschiedene Änderungsdokumente entstehen. Ausgangspunkt ist eine Störfallmeldung. Auf einen Änderungsantrag hin werden Änderungsanweisungen ausgearbeitet und schliesslich Änderungen durchgeführt. Nach Abschluss der Änderungen muss das Objekt wieder den Freigabeprozess durchlaufen.

### Verständnisfrage 1

Welche Hauptunterscheidung kann für Vorgänge des Prozessmanagements getroffen werden?

### Verständnisfrage 2

Welche formalen Workflow-Prozesse sind von Bedeutung?

### Verständnisfrage 3

Was versteht man unter einer Freigabe, wozu wird sie benötigt?

### Verständnisfrage 4

Welches sind die Grundelemente einer dynamischen Ablaufmodellierung?

### Verständnisfrage 5

Welche zwei Aspekte sind im Freigabenwesen bezüglich Autorisierung zu unterscheiden?

### Verständnisfrage 6

Was versteht man unter einem SignOff?



### Verständnisfrage 7

Aus welchen Hauptphasen besteht ein Änderungsprozess?

### Verständnisfrage 8

Welche Arbeitsdokumente werden im Rahmen eines Änderungsprozesses benötigt?

### Verständnisfrage 9

Welche Informationen beinhaltet ein Änderungsantrag?

### Verständnisfrage 10

Ab wann werden Änderungen für die Produktion wirksam?

### Verständnisfrage 11

Wie kann man technische Änderungen unterscheiden?

### Verständnisfrage 12

Was versteht man unter einer Supercedure-Relation bei einer Änderung?

### Verständnisfrage 13

Welche Methoden sind zum Kopieren von Objekten mit Beziehungen zu unterscheiden?

### Publikationsverzeichnis – Literatur

- [DIN6772] *DIN 6772: Änderung von Dokumenten und Gegenständen.* Berlin: Beuth-Verlag 1981
- [EiS-01] *Eigner, M.; Stelzer, R.: Produktdatenmanagement-Systeme: Ein Leitfaden für Product Development und Life Cycle Management.* Berlin: Springer Verlag 2001
- [VDI-72] *VDI-Fachgruppe Betriebstechnik (ADB): Elektronische Datenverarbeitung bei der Produktionsplanung und -steuerung III: Informations- und Stücklistenwesen.* VDI-Taschenbücher T 28, Düsseldorf: VDI-Verlag 1972
- [Wie-89] *Wiendahl, H.-P.: Betriebsorganisation für Ingenieure.* 3. Auflage, München: Hanser Verlag 1989

# Schnittstellen im Digitalen Produkt

Autor: Dr. E. Zwicker



# 1 Überblick

Die Integration verschiedener CAx-Systeme in die rechnerintegrierte Produktentwicklung mit dem Ziel einer durchgängigen Informationsverarbeitung erfordert die Möglichkeit Informationen austauschen zu können. Hierfür werden definierte Schnittstellen benötigt.

Der Begriff „Schnittstelle“ bezeichnet die Verbindungsstelle (Nahtstelle) zwischen zusammenwirkenden Systemen. Zur Festlegung des Begriffs Schnittstelle soll folgende Definition gelten [GAG-86]:

„Eine Schnittstelle ist ein System von Bedingungen, Regeln und Vereinbarungen, das den Informationsaustausch zweier miteinander kommunizierender Systeme oder Systemkomponenten festlegt“.

Innerhalb der rechnerintegrierten Produktentwicklung werden eine Vielzahl von *CAx-Schnittstellen* benötigt (vgl. Tabelle T001srpZ) um den anfallenden vielschichtigen Informationsaustausch zu ermöglichen. Entsprechend den auszutauschen Informationsinhalten bzw. den Systemen oder Systemkomponenten zwischen welchen diese Informationen auszutauschen sind, wurden spezialisierte Schnittstellen definiert. Von grosser Bedeutung innerhalb der rechnerintegrierten Produktentwicklung sind Schnittstellen zum Austausch von Produktdaten und Prozessdaten.

Für den geforderten Austausch produktbezogener Daten stehen zur Zeit prinzipiell eine Vielzahl an genormten und nicht genormten (de-facto Standards) *Produktdatenschnittstellen* zur wie z.B. IGES bzw. VDA-IS, VDA-FS, DXF, etc. zur Verfügung. Diese bisher veröffentlichten Schnittstellen sind jedoch aufgrund der jeweils zugrundeliegenden Konzepte und angestrebten Modellinhalte (z.B. Handhabung von Produktmetadaten, Parameterisierung, etc.) nur bedingt oder gar nicht zur Realisierung des in diesem Zusammenhang geforderten Informationsflusses geeignet.

Die auf einem sogenannten integrierten Produktmodell basierenden Schnittstelle *STEP* (ISO-10303) soll es erlauben, sämtliche innerhalb des gesamten Produktlebenszyklus erzeugten Produktdaten austauschen, transformieren, speichern und archivieren zu können. Zum jetzigen Zeitpunkt bietet STEP die Möglichkeit, komplexe CAD-Modellstrukturen sowie die zugehörigen Metadaten auszutauschen. Wie bereits in Kapitel 1 „Einführung in die rechnerintegrierte Produktentwicklung“ spielt in diesem Zusammenhang die Standardisierung eines allgemein gültigen integrierten Produktmodells die entscheidende Rolle. Mittel- bis Langfristig wird der CAx-Markt massgeblich durch die Entwicklungen in diesem Bereich geprägt werden.

Mit der immer weiter fortschreitenden Globalisierung der Märkte ist auch die Produktentwicklung einem entsprechenden Wandel unterworfen. Die *Integration von Norm- und Zukaufteile* und der hierfür benötigte überbetriebliche Austausch von Informationen innerhalb einer Produktentwicklung gewinnt in diesem Zusammenhang zunehmend an Bedeutung. Im Mittelpunkt steht die zielsystemneutrale Beschreibung von Teilefamilien innerhalb von Teilebibliotheken bzw. elektronischen Teilkatalogen mit den zugehörigen administrativen Informationen wie zum Beispiel die Identifizierung des Lieferanten, die Bestellnummern, Verfügbarkeiten oder Preisinformationen.

Komplementär zu STEP wird aus diesem Grund die Schnittstelle *Parts Library* (ISO 13584) entwickelt und genormt. Im Mittelpunkt steht die Repräsentation von Teilebibliotheken bzw. Zukaufteilkataloge sowie die Bereitstellung von Mechanismen und Definitionen zum Austausch, der Benutzung und der Aktualisierung einzelner Bibliotheksinhalte. Der überbetriebliche Austausch von Bibliotheksinhalten soll dabei unabhängig vom jeweils verwendeten Zielsystem (z.B. CAD-Systemen) sowie Soft- und Hardwareumgebung (z.B. Betriebssystem, Prozessorarchitektur, Netzwerk, etc.) innerhalb des gesamten Produktlebenszyklus, also von der Konstruktion, der Fertigung, Montage, Wartung und Beseitigung ermöglicht werden.

CAx-Hardware Schnittstellen

System	Kopplung	Bus	Netzwerk
Charakteristika	Anschlussgeometrie Signal Kanal Topologie Übertragungslänge	Spezialisierung Steuerung Breite/Adress- raum Synchronisation	räuml. Ausdehnung integrierte Systeme Topologie Dienste Protokoll
Beispiele	V24 SCSI	AT-Bus Profi-Bus	Internet SNA

Tabelle (T001srpZ) Hardware-Schnittstellen der rechnerintegrierten Produktentwicklung

CAx-Software Schnittstellen

Schnittstelle	intern	extern
Charakteristika	Übertragungsbreite Taktfrequenz	Informationsinhalte: <ul style="list-style-type: none"> <li>• Produktbezogen</li> <li>• Prozessbezogen</li> <li>• Auftragsbezogen</li> </ul> Austauschpartner: <ul style="list-style-type: none"> <li>• System/Anwender</li> <li>• System/System</li> </ul>
Beispiele	GKS OPEN GL	STEP DXF

Tabelle (T002srpZ) Software-Schnittstellen der rechnerintegrierte Produktentwicklung

## 2 Motivation und Zielsetzung

Bezüglich den geforderten Möglichkeiten zum Austausch von Produktdaten innerhalb des gesamten Produktlebenszyklus werden folgende Zielsetzungen unterschieden:

- Systemtechnische Zielsetzungen
- Austausch zwischen einzelnen Komponenten von CAx-Systemen
- Austausch zwischen CAx-Systemen sowie anderen Programmsystemen (z.B. PDM oder PPS-Systemen)
- Betriebsinterne Zielsetzungen
- firmeninterner Austausch zwischen einzelnen Abteilungen
- firmenexterner Austausch zwischen kooperierenden Unternehmen
- Organisatorische Zielsetzungen hinsichtlich der technischen Anwendung
- Verfügbarkeit bzw. Integration von Norm- und Zukaufteilen
- systemunabhängige Modellarchivierung

Von zentraler Bedeutung für die Anwendung bzw. als Investitionsschutz für ein Unternehmen ist der Status der Schnittstellenspezifikation in Bezug auf die Normung. Die Normung von Schnittstellen erfolgt durch verschiedene nationale und internationale Normungsgremien (vgl. Tabelle T003srpZ).

Im allgemeinen zielen die Normungsaktivitäten auf die Festschreibung einer hinsichtlich des jeweiligen Anwendungsgebietes allgemeingültigen Schnittstellenspezifikation sowie der öffentlichen Bereitstellung dieser Spezifikation ab. Durch die öffentliche Verfügbarkeit der Schnittstellenspezifikation werden die Stabilität die Reproduzierbarkeit sowie etwaige Zertifizierungen ermöglicht bzw. unterstützt.

Im Gegensatz zu nicht genormten Schnittstellen zweier Systeme untereinander weisen genormte Schnittstellen folgende Vorteile auf:

- geringer Entwicklungsaufwand:  
Ist keine genormte Schnittstelle definiert, müssen für  $N$  verschiedene CAx-Systeme  $N(N-1)$  einzelne Schnittstellen implementiert werden. Für jedes neu hinzukommende System müssten entsprechend wiederum  $2N$  Schnittstellen implementiert werden.

Dagegen erfordert eine genormte Schnittstelle lediglich  $2N$  Implementierungen dieser Schnittstelle; jedes neu hinzukommende CAx-System erfordert nur 2 zusätzliche Implementierungen (vgl. Tabelle T003srpZ).

- geringerer Wartungs- und Pflegeaufwand:  
Mit jeder Schnittstellenimplementierung ist ein dauernder Wartungs- und Pflegeaufwand verbunden.

	Spezielle Schnittstelle	Genormte Schnittstelle
Kopplungsverfahren		
Anzahl Prozessoren für N Systeme	Anzahl = $N(N+1)$	Anzahl = $2N$
Aufwand für N+1 tes System	Anzahl = $2N$	Anzahl = $2$

Tabelle (T003srpZ) Datenaustausch zwischen N bzw. N+1 CAx-Systemen [AND-93]



### 3 CAx-Schnittstellen der rechnerintegrierten Produktentwicklung

Abhängig von dem jeweiligen Verwendungszweck und deren Integration in die jeweiligen CAx-Systeme können CAx-Schnittstellen zunächst wie folgt klassifiziert werden (vgl. auch Tabelle T001srpZ und T002srpZ):

- CAx-Hardwareschnittstellen

Die Hardware von CAx-Systemen besteht aus mehreren Komponenten (CPU, Coprozessoren, Speichereinheiten, usw. sowie periphere Geräte wie Drucker, Bildschirm, Tastatur etc.), die über Hardware-Schnittstellen miteinander verbunden sind. Um einheitliche Anschlüsse verfügbar zu haben, sind möglichst genormte Hardwareschnittstellen (z.B. Centronics oder V24) erforderlich.

Hardwareschnittstellen werden durch ihre jeweiligen mechanischen Merkmale (Anzahl und Form der Kontaktstifte bzw. Kontaktflächen), durch die Signalcharakteristiken (Codierung, Signalpegel, Trägerfrequenzen, etc.) sowie durch den Übertragungskanal (Art des Leiters, Bandbreite, Übertragungsrate, etc.) geprägt.

Besondere Hardwareschnittstellen zur Kopplung von Komponenten eines Rechnersystems sind Busse. Hardwareschnittstellen zur Kopplung mehrerer Rechnersysteme werden als Rechnernetze bezeichnet auf welche in Kapitel 2 getrennt eingegangen wurde.

- CAx-Softwareschnittstellen

Softwareschnittstellen dienen zum Austausch von Informationen zwischen einzelnen funktionalen Softwarebausteinen von CAx-Systemen (interne Schnittstellen) bzw. zwischen CAx-Systemen und seiner Umwelt (externe Schnittstellen).

Auf diese CAx-Software Schnittstellen soll im folgenden detailliert, speziell im Hinblick auf den Austausch von Produktdaten innerhalb der Produktentwicklung, eingegangen werden.

### 3.1 Interne CAx-Softwareschnittstellen

Im allgemeinen bestehen CAx-Systeme im wesentlichen aus den folgenden drei Bausteinen (vgl. Bild B001srpZ):

- **Kommunikationsbaustein**  
Er verwaltet, steuert und überwacht alle Methoden zur Dialogführung mit dem Benutzer und übergibt die Funktions- und Steuerparameter an die ausführenden Methodenprogramme. Wichtiger Bestandteil des Kommunikationsbausteins ist ein Graphikmodul für die Darstellung der technischen Lösung auf dem Bildschirm.
- **Methodenbank**  
Die Methodenbank dient der Erzeugung und Modifikation rechnerinterner Modelldaten durch die Bereitstellung, Überwachung und Steuerung einer Menge von Methoden. Eine Methode wird als ein Algorithmus verstanden, der in einer programmtechnischen Realisierung verfügbar ist.
- **Datenbankbaustein**  
Das Datenbankbaustein dient der Verwaltung der Daten des Produktmodells entsprechend dem definierten Schema zur Speicherung der technischen Lösung. Im Datenbankbaustein wird der elementare Zugriff auf die Daten festgelegt.

Diese Bausteine sind in ihrer Funktionalität nicht unabhängig voneinander. Derartige Abhängigkeiten werden über die verschiedenen CAx-Softwareschnittstellen festgelegt und geregelt. Die Softwarestruktur wird durch die Gesamtheit aller Bausteine und Schnittstellen festgelegt.

Für den Datenaustausch zwischen den funktionellen Komponenten eines CAx-Systems werden eine Reihe *interner Schnittstellen* benötigt. *Externe Schnittstellen* bestehen zwischen Benutzer und CAx-System sowie zum Datenaustausch mit anderen CAx- bzw. Computer-Systemen.

Die internen Schnittstellen S1 bis S4 spielen vor allem im Hinblick auf die Austauschbarkeit und Erweiterbarkeit von Softwarekomponenten eine wesentliche Rolle.

- **Ein- und Ausgabeschnittstelle**  
Sie stellt die geräteunabhängige, graphisch-interaktive Ein- und Ausgabeschnittstelle zwischen Anwendungsprogrammen und peripheren Geräten dar. Hauptbestandteil dieser Schnittstelle bildet das durch DIN 66252 für 2D Linien- und Rastergraphiken genormte Graphische Kernsystem GKS. Neuere Entwicklungen stellen GKS-3D, PHIGS (Programmer's Hierarchical Interactive

Graphics Specification), PEX (PHIGS-Extensions to the X-Window System ) oder neuerdings OPEN GL dar.

- Methodenablaufsteuerungs- und Methodenaufrufsschnittstelle  
Die Verbindung zwischen Kommunikations- und Methodenbanksystem durch logisch verbundene Definitions-, Verwaltungs- und Sequenztabelle wird durch die Methodenablaufsteuerungs- und Methodenaufrufsschnittstelle realisiert. Eine allgemeingültige Schnittstelle zum geometrischen Modellieren wurde durch CAM-I spezifiziert und als Anwendungsschnittstellenspezifikation veröffentlicht [AND-93].

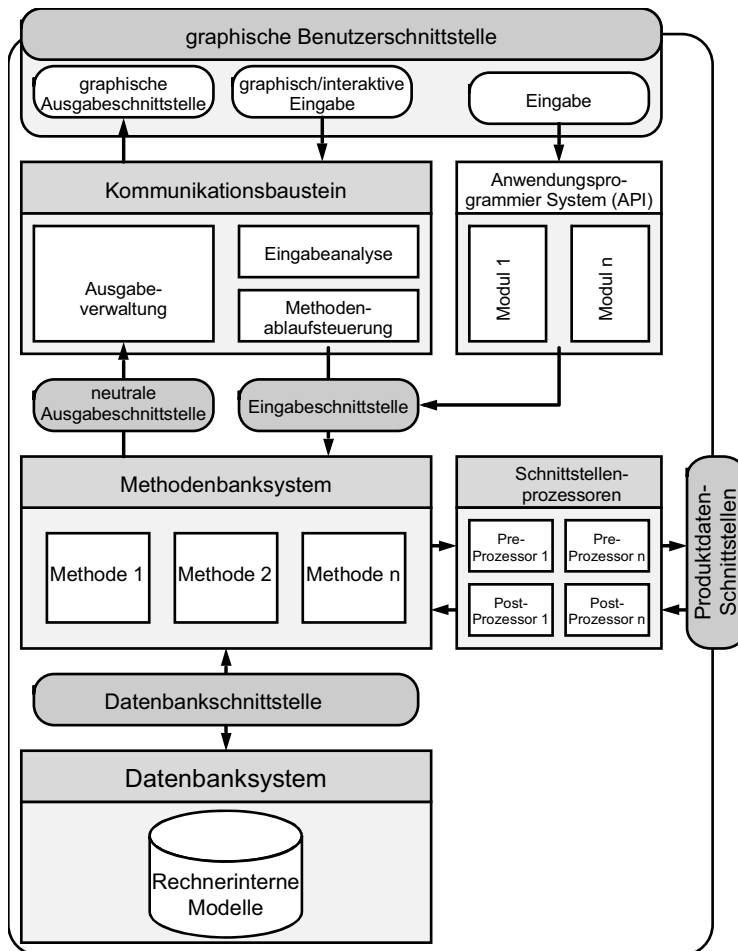


Bild (B001srpZ) Aufbau und Schnittstellen von CAD-Systemen /ANDE-93/

- Neutrale Ausgabeschnittstelle  
Die neutrale Ausgabeschnittstelle stellt die Verbindung zwischen Methoden- und Kommunikationssystem durch Ausgaben in neutraler Form dar (d.h. unabhängig von der graphischen Darstellung am Bildschirm) und dient zur Übergabe der Ausgabedaten als geordnete Datenmengen an das Ausgabeverwaltungssystem.
- Datenbankschnittstelle  
Die Datenbankschnittstelle dient zur Verbindung von Methoden- und Datenbank entweder als elementare Datenbankmanipulationsschnittstelle (DBMS) mit den logischen Funktionen „Einfügen“, „Holen“, „Ändern“ und „Löschen“ über den syntaktischen Grundeinheiten oder als höhere Datenbankmanipulationsschnittstelle (HDBMS) zur Manipulation komplexer Objektstrukturen (bestehend aus verschiedenen syntaktischen Grundeinheiten).

## 3.2 Externe CAx-Softwareschnittstellen

Die externen CAD-Schnittstellen können abhängig von der Art und der Zielsetzung des Informationsaustausches unterschieden werden in:

- Dialog- und Programmierschnittstellen zum Informationsaustausch zwischen CAx-System und Anwender.

Die *Dialogschnittstelle* dient der graphisch-interaktiven Eingabe von Befehlen in systemgeführter (z.B. über Funktionsmenüs) oder benutzergeführter Dialogform (z.B. mittels einer Kommandozeile).

Über die *Programmierschnittstelle* können während der Arbeit mit dem CAD-System eingebundene externe Programme aufgerufen werden, welche Funktionen des CAD-Systems aktivieren. Sie erlaubt beispielsweise mittels einer Varianten- oder Makrosprache die Beschreibung von Erzeugungslogiken zur Erstellung technischer Lösungen und ersetzt die Interaktionen mit dem CAD-System, die der Anwender ausführen müsste, um diese Modelle aufzubauen.

Die Dialog- und Programmierschnittstelle sind fester Bestandteil eines CAx-Systems und unterliegen in der Regel keiner direkten Normungsaktivität.

- Datenschnittstellen  
zur Übertragung bzw. Austausch von Produktdaten zwischen  
einzelnen CAx-System bzw. weiteren Computer-Systemen (z.B.  
PPS Systemen)

### 3.3 Externe CAx-Softwareschnittstellen zum Austausch von Produktdaten

CAx-Softwareschnittstellen zum Austausch von Produktdaten sind Programmbausteine, die auf den systeminternen Schnittstellen eines CAx-Systems aufsetzen bzw. deren Funktionalität nutzen. Zielsetzung ist die Transformation bzw. Abbildung von Produktdaten vorliegend in der jeweiligen systemspezifischen Repräsentation in eine systemfremde Repräsentation bzw. im umgekehrten Falle ausgehend von einer systemfremden Repräsentation in die interne Repräsentation.

Handelt es sich bei der systemfremden Repräsentation um eine zielsystemspezifische Repräsentation, spricht man von einer Systemkopplung.

Für den Produktdatenaustausch, basierend auf einer einheitlichen systemneutralen Datenmodellspezifikation, werden zwei Programmbausteine benötigt:

- Der *Preprozessor* überträgt die Daten des jeweiligen rechnerinternen Modells in das jeweilige definierte systemneutrale Datenmodell.
- Der *Postprozessor* überträgt das einheitliche, neutrale Datenmodell in das jeweilige CAx-systemspezifische rechnerinterne Modell.

Pre- und Postprozessor gehören zur Software des jeweiligen CAx-Systems.

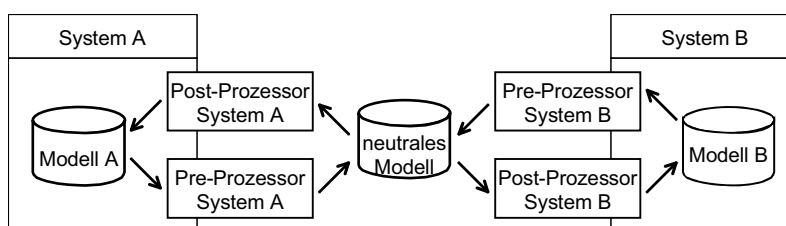


Bild (B002srpZ) Pre- und Postprozessor

Entsprechend den jeweils unterschiedlichen Informationsinhalten, welche innerhalb des Produktentwicklungsprozesses benötigt werden, unterscheidet man zwischen

- produktbezogenen bzw. modellbezogenen Informationsinhalten und deren Austausch zwischen CAx-Systemen. Im Mittelpunkt steht die geometrische Beschreibung des Produktes sowie weitere benötigte Informationen wie zum Beispiel tech-

nologische, funktionale, darstellungsbezogenen, fertigungs-technische oder organisatorische Informationen.

- *prozessbezogene Informationsinhalten* und deren Austausch innerhalb der technischen Prozesskette (z.B. zwischen CAD- und CAM-System). Wesentliches Merkmal prozessbezogener Informationen ist der direkte oder indirekte Bezug auf den Parameter Zeit.
- *auftragsbezogene Informationsinhalten* und deren Austausch innerhalb der betriebswirtschaftlichen Prozesskettenseite bzw. zwischen der betrieblichen und der technischen Prozesskettenseite.



### 3.4 Schnittstellen zum Austausch von Produktdaten

Für den geforderten Austausch stehen zur Zeit prinzipiell eine Vielzahl an genormten und nicht genormten (defacto Standards) Austauschformate zur Übertragung der produktdefinierenden Daten wie z.B. IGES bzw. VDA-IS, VDA-FS, DXF, etc. zur Verfügung (vgl. Tabelle T003srpZ).

Diese Schnittstellen können unterscheiden sich im wesentlichen hinsichtlich

- der zu übertragenen Modelle und Modellinhalte,
- der Art der Spezifikation sowie
- dem spezifizierten Dateiformat

Allgemein liegt eine Tendenz zu zunehmend komplexeren Modellinhalten vor (vgl. Tabelle T005srpZ bzw. Bild B003srpZ). Die Schnittstellen IGES, SET und VDA-FS waren im wesentlichen auf geometrische Daten, Berechnungsdaten sowie Daten der technischen Zeichnung beschränkt. Innerhalb der Entwicklung der Schnittstelle PDDI wurden zusätzlich Werkstückmodelle erarbeitet, welche die Übertragung von Toleranzen, Materialeigenschaften und Oberflächeneigenschaften erlauben. Die Entwicklung der Schnittstelle STEP soll die Beschreibung aller innerhalb des Produktlebenszyklus anfallenden Produktdaten ermöglichen.

Abkürzung	Organisation	Nation/Sitz
NIST	National Institute for Standards and Technology	USA/Washington
ANSI	American National Standards Institute	USA/Washington
CAM*I	Computer Aided Manufacturing International	USA
VDA	Verband Deutscher Automobilhersteller	Deutschland/Frankfurt
AEROS	Aerospatial	Frankreich
U.S. Air	U.S. Air Force	USA
DIN	Deutsches Institut für Normung	Deutschland/Frankfurt
ISO	International Organisation for Standardisation	Genf
AFNOR	Association Francaise de Normalisation	Frankreich/Paris
ESPRIT	European Strategic Program for Research and Development in Information Technology	Europäische Union/ Brüssel

Tabelle (T004srpZ) Normungsgremien

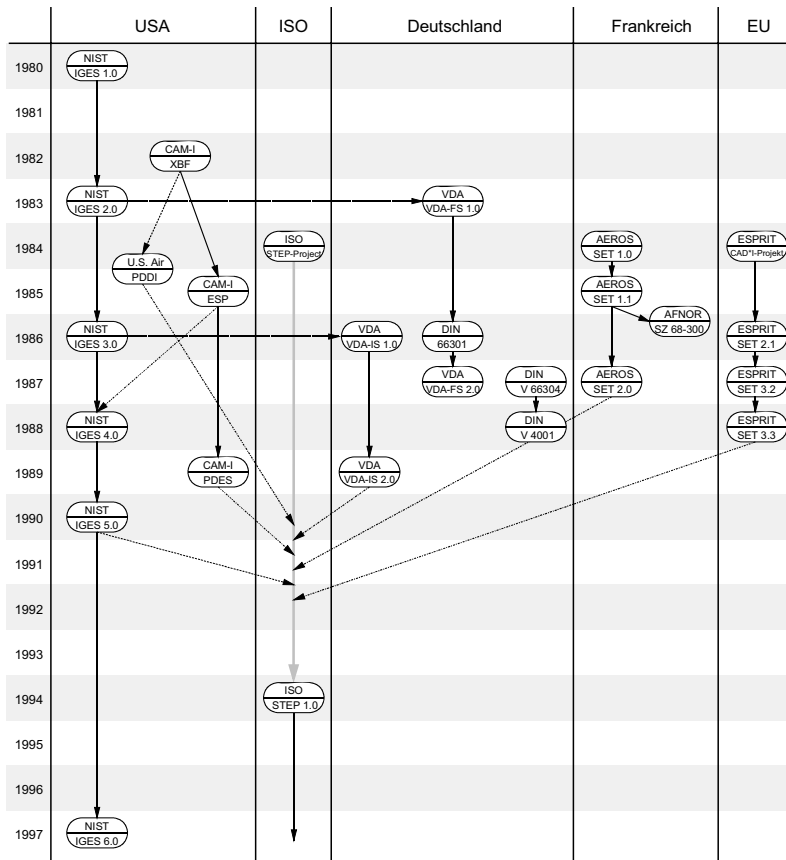


Bild (B003srpZ) Schnittstellen-Entwicklung der rechnerintegrierten Produktentwicklung

	<b>IGES</b> (Initial Graphics Exchange Specification)	<b>SET</b> (Standard d'Exchange et de Transfer)	<b>VDA-FS</b> (VDA Fachenschnittstelle)	<b>PDDI</b> (Product Definition Data Interface)	<b>CAD1</b> (Schnittstelle ES-PR1-Projektes CIM 322)	<b>STEP</b> (Standard for the Exchange of Product Model Data)	<b>DXF</b> (Drawing Interchange Format)	<b>EDIF</b> (Electronic Design Interchange Format)	<b>VNS</b> (Verfahrensneutrale Schnittstelle)
<b>Modelle</b>	Kantenmodelle Analyt. Flächenmodell Volumenmodelle Berechnungsmodelle Techn. Zeichnungen	Kantenmodelle Analyt. Flächenmodell Volumenmodelle Berechnungsmodelle Techn. Zeichnungen CAD/CAM-Daten	Flächenmodelle (Freiformflächen)	Volumenmodelle Fertigungstechnik	Kantenmodelle Analyt. Flächenmodell Volumenmodelle Berechnungsmodelle	Partialmodelle des gesamten Produktlebenszyklus	Kantenmodelle Techn. Zeichnungen	Produktdatenmodell für den gesamten Produktlebenszyklus elektronischer Produkte	Symbolbibliotheken Elektrotechnische Schaltpläne
<b>Umfang der Spezifikation</b>	Lose Elementensammlung	Lose Elementensammlung	Verknüpfbare Elemente zur Beschreibung von Freiformflächen	Formale Sprachdefinition Form. Beschreibung der Elemente Modelle zur Handhabung fertigungs-technischer Informationen Prozessorbausteine Zugriffsoftware	Formale Sprachdefinition Form. Beschreibung der Elemente Modelle für Volumen und FEM-Übertragung Prozessorbausteine Zugriffsoftware	Formale Sprachdefinition Form. Beschreibung der Elemente Modelle für kompletten Produktlebenszyklus Prozessorbausteine	Lose Elementensammlung	Formale Beschreibung der Elemente Modelle für kompletten Produktlebenszyklus	Lose Elementensammlung
<b>Dateiformat</b>	Definiertes Dateiformat (Sektionen)	Definiertes Dateiformat (Blockstruktur)	Definiertes Dateiformat	Definiertes Dateiformat	Formale Definition des Dateiformats	Formale Definition des Dateiformats	Definiertes Dateiformat (Sektionen)	Formale Definition des Dateiformats	Definiertes Dateiformat

Bild (B054srpZ) Vergleichender Überblick über relevante Schnittstellen

	IGES (Initial Graphics Exchange Specification)	SET (Standard d'Exchange et de Transfer)	VDA-FS (VDA Flächenschnittstelle)	PDDI (Product Definition Data Interface)	CAD1 (Schnittstelle des ESPRIT-Projektes CIM 322)	STEP (Standard for Exchange of Product Model Data)	DXF (Drawing Interchange Format)
Modellinhalte	Funktionale Zusammenhänge	Elemente für Schemapläne	-	-	-	Elemente für Schemapläne	-
	Gestaltbeschreibung	Kanten-/Flächenmodell CSG-Modell Zeichnung	Flächenmodell mit Topologie	Kanten-/Flächenmodell B_REP-Modell	Kanten-/Flächenmodell B_REP-Modell CSG-Modell	Kanten-/Flächenmodell B_REP-Modell CSG-Modell	Kantenmodell
	Formelemente	-	-	-	Verbundmaterial Fertigungstechnik Blech Rotationssteile	-	-
Abgeleitete Modellinhalte	Toleranzen/Oberflächen	Bestandteil Bemessung	-	Toleranzmodell	-	Toleranzmodell	Bestandteil Bemessung
	Materialeigenschaft	Bestandteil Bemessung	-	-	-	Materialmodell	Bestandteil Bemessung
	Lebenszyklus/Metadaten	Bestandteil Bemessung (z.B. Zeichnungskopf)	-	-	-	Produktlebenszyklusmodell	Bestandteil Bemessung (z.B. Zeichnungskopf)
Abgeleitete Modellinhalte	Produktdarstellung	Zeichnungsmodell	-	-	-	Produktlebenszyklusmodell	-
	Berechnung	FEM-Modell	-	-	FEM-Modell	FEM-Modell	-
	Verwaltung	Bestandteil Bemessung (z.B. Zeichnungskopf)	-	Bestandteil Bemessung (z.B. Zeichnungskopf)	-	PSCM-Modell	-

Bild (B055srpZ) Modellinhalte ausgewählter Schnittstellen

### 3.5 Schnittstellen zum Austausch von Prozessdaten

Sie werden insbesondere für CAD/CAM-Prozesse benötigt und liegen in verschiedenen Sprachniveaus bzw. Auflösungsstufen vor. Die wesentlichen Schnittstellen für die CAD/CAM-Kopplung sind:

- APT (Automatically Programming Tools zur Erstellung von NC-Teilprogrammen entsprechend der maschinellen NC-Programmierung),
- CLDATA (Cutter Location Data als Zwischenausgabe von NC-Prozessoren in Form von maschinenunabhängigen Werkzeugverfahrwegen ohne Technologiedaten),
- Steuerbefehle für Werkzeugmaschinen nach DIN 66025 (Sequenzen von maschinenabhängigen Befehlen für NC-Werkzeugmaschinen) sowie
- Kontur- und Funktionsmakros nach DIN 406.

Die einzelnen Schnittstellen sollen im folgenden hinsichtlich der jeweiligen Modellinhalte beschrieben werden. Hinsichtlich der praktischen Anwendung innerhalb der rechnerintegrierten Produktentwicklung wurde in Kapitel 10 "CAD/CAM-Prozesskette" detailliert eingegangen.

#### 3.5.1 Steuerbefehle für Werkzeugmaschinen nach DIN 66025

Die Schnittstelle DIN 66025 spezifiziert die Repräsentation von maschinen- bzw. steuerungsabhängigen Befehlen zur Steuerung von NC-Werkzeugmaschinen.

#### Informationsinhalte

Die DIN 66025 umfasst folgende Klassen von Informationen:

- geometrische Anweisungen (Zielkoordinaten des Werkzeugs)
- technologische Anweisungen (Vorschubgeschwindigkeit, Hauptspindeldrehzahl, Werkzeugnummer)
- Fahrhinweise (Wegbedingungen)
- Schaltbefehle (Kühlmittel, Rechtslauf-Linkslauf, etc.)
- Korrekturaufrufe (Werkzeuglängenkorrektur)
- Zyklen- und Unterprogrammaufrufe

Der Programmaufbau ist nach DIN 66025 wie folgt festgelegt:

- Kennzeichnung des Programmbeginns (%-Zeichen)
- eine Folge von Anweisungen, als Wörter bezeichnet, zusammengefasst in Sätzen
- Kennzeichnung des Programmendes.

Ein Anweisung setzt sich aus einen Adressbuchstaben und einer Ziffernfolge, die entweder als lineare oder rotatorische Massangabe oder als Schlüsselzahl verwendet, wird zusammen.

Technologische Angaben, die abhängig von einer bestimmten Werkzeugmaschinensteuerung sind müssen ergänzt werden.

### Austauschformat

Als Datenformat ist ein 8-spuriges Lochstreifenformat, basierend auf dem EIA-Code (EIA: Electronic Industries Association) oder dem ISO-Code, definiert. Ausserdem kann der Austausch über eine ASCII-Datei erfolgen.

### Aufbau der Norm

Die DIN 66025 setzt sich aus zwei Blättern zusammen:

- Blatt 1 „Programmaufbau für numerisch gesteuerte Arbeitsmaschinen; Allgemeines“ spezifiziert den Aufbau von Programmen zur Steuerung numerisch gesteuerter Werkzeugmaschinen. Insgesamt werden der Aufbau des Programmes, des Satzes, des Wortes, der Zeichenvorart und die Programmierverfahren für die Interpolation definiert.
- Blatt 2 „Industrielle Automation; Programmaufbau für NC-Maschinen; Wegbedingungen und Zusatzfunktionen“ beinhaltet die Festlegung von Schlüsselzahlen für Wegbedingungen und Zusatzfunktionen bei der Programmierung von NC-Maschinen.

### 3.5.2 CLDATA (Cutter Location Data nach DIN 66215 )

Im Gegensatz zu dem NC-Anweisungssatz nach DIN 66025 spezifiziert CLDATA die Repräsentation von maschinen- und steuerungsunabhängigen Werkzeugpositionen. Bei CLADATA handelt es sich um einen Zwischencode, der von NC-Prozessoren erzeugt wird.

Auf CLDATA setzen maschinenspezifische NC-Postprozessoren auf die entsprechend den gewählten Werkzeugen und technologischen Parametern (Vorschub, Drehzahl etc.) daraus maschinen- und steuerungsabhängige NC-Daten erzeugen. Dabei kann es sich um steuerungsspezifische Formate oder um genormte Formate wie z.B. DIN 66025 handeln.

### Informationsinhalte

CLDATA enthält im wesentlichen die berechneten Bahnkoordinaten des Werkzeugmittelpunktes (CLDATA 1) in Form von Anweisungen, Hauptwörter genannt. Einzelne Hauptwörter sind zu Gruppen zusammengefasst. Die wichtigsten sind:

- NC-Postprozessor Anweisungen
- Flächendaten
- Werkzeugposition
- Toleranz oder Werkzeugangabe
- Achsart; Einheiten
- Werkzeugweg
- Werkstückkonturbeschreibung
- Werkzeugbeschreibung
- Werkstoffbeschreibung
- Maschinenbeschreibung

Innerhalb CLADATA 2 werden der detaillierte Aufbau und Inhalt der NC-Postprozessoranweisungen aus CLDATA 1 festgelegt. Zu den in CLDATA 1 festgelegten Hauptwörtern für Postprozessor-Anweisungen werden in diesem Teil der Norm die Nebenteile definiert. Mittels diesen Nebenteilen werden die einzelnen Anweisungen (Hauptwörter) hinsichtlich technologischer Informationen spezialisiert.

### Austauschformat

Der Austausch von CLDATA-Informationen erfolgt über eine ASCII-Datei.

### Aufbau der Norm

DIN 66215 setzt sich aus zwei Blättern zusammen:

- Blatt 1 „Allgemeiner Aufbau und Satztypen“ entspricht auf internationaler Ebene der ISO 3592 und spezifiziert den Aufbau von CLDATA-Texten für die Verwendung im Zusammenhang mit Programmiersprachen für NC-Maschinen. Für jeden NC-Postprozessor soll es möglich sein, CLDATA-Texte von einem in dieser Norm festgelegten Aufbau zu erzeugen.
- Blatt 2 „Nebenteile des Satztyps 2000“ entspricht auf internationaler Ebene der ISO 4343 und spezifiziert zu den im Blatt 1 (ISO 3592) festgelegten Hauptwörtern für Postprozessor-Anweisungen Nebenteile.



### 3.5.3 Höhere NC-Programmiersprachen

Höhere NC-Programmiersprachen erlauben die Erstellung von Teilprogrammen entsprechend der maschinellen Programmierung. Mit der Entwicklung dieser Programmiersysteme haben sich eine Anzahl nicht standardisierter Industriestandards etabliert.

Weitverbreitet sind:

- APT (Automatically Programmed Tools) mit den verschiedenen Subsets:
  - EXAPT (Extended Subset of APT)
  - MINIAPT
- COMPACT

### 3.5.4 DIN 406 Makros

Die Schnittstelle für NC-Geometriemakros entsprechend DIN 406 stellt vordefinierte NC-Geometriemakros bereit.

Die Schnittstelle für NC-Funktionsmakros entsprechend DIN 406 beinhaltet vordefinierte Fertigungs- und Funktionskomplexe.

## 4 Übersicht über ausgewählte Produktdatenschnittstellen

Für die industrielle Praxis sind eine Reihe der entwickelten Schnittstellen aufgrund der geringen internationalen Verbreitung von untergeordneter Bedeutung (die Schnittstelle SET kommt beispielsweise vorwiegend in Frankreich zum Einsatz). Bei anderen wie z.B. CAD\*I, PDES oder PDDI handelt es sich um Forschungsprojekte, deren Resultate massgeblich die Entwicklung der internationalen Norm STEP beeinflusst hat bzw. direkt übernommen wurde.

Die für die industrielle Anwendung relevanten Schnittstellen sollen im folgenden kurz beschrieben werden. Der Schwerpunkt liegt dabei jeweils auf den zugrunde liegenden Konzepten und Modellinhalten sowie dem spezifizierten Austauschformat. In diesem Zusammenhang sind IGES und VDA-FS als standardisierte Schnittstellen sowie DXF als Industriestandard zu nennen.

Erwähnt sei an dieser Stelle noch die Schnittstelle SET, die erstmals den Produktdatenaustausch zwischen CAx-Systemen und einer zentralen Produktdatenbank konzeptuell in die Überlegung mit einbezogen. SET 1.1 wurde im Hinblick auf den Bau des Airbus A320 in Frankreich entwickelt und erreichte bereits ein Jahr vor dem Erscheinen der IGES Version 3.0 (1986) ungefähr dessen Funktionalität. Die Verbreitung blieb jedoch auf Frankreich bzw. den Airbuskonzern beschränkt und ging schliesslich ebenfalls in der STEP-Entwicklung auf.

Zukünftig wird die Schnittstelle STEP mit Sicherheit eine entscheidende Bedeutung hinsichtlich des ganzheitlichen Produktdatenaustausches und -archivierung einnehmen. Die STEP-Entwicklung und Standardisierung wird dabei die weitere Entwicklung von CAx-, PDM- und PPS-Systemen beeinflussen, wenn nicht sogar lenken. Aus diesem Grund soll auf die Entwicklung und Methodologie des STEP-Standards detaillierter eingegangen werden.

## 4.1 IGES

IGES wurde 1980 erstmals vorgestellt (Version 1.0) und 1981 als ANSI-Norm Y14.26M genormt. IGES war ursprünglich zum Austausch von 2D-Zeichnungen konzipiert, wurde jedoch mit den folgenden Versionen auf die Übertragung von 2D- und 3D-Kantenmodellen, Flächenmodellen, Volumenmodellen und technischen Zeichnungen ausgerichtet. Schwerpunkt bildet die mechanische Konstruktion, spezielle Elemente unterstützen auch die Anwendungsbereiche Elektrik/Elektronik und FEM.

Eine erste Erweiterung hinsichtlich der Abbildung von Volumenmodellen (CSG- und B-Rep Modelle) stellte innerhalb der IGES-Entwicklung das Experimental Boundary File XBF (1982) dar. Die XBF-Entwicklung wurde vom IGES-Komitee aufgegriffen und die Schnittstelle ESP (Experimental Solid Proposal) parallel zur IGES Version 3.0 (1993) entwickelt. Ziel der ESP-Schnittstelle war wiederum die Beschreibung von Volumenelementen im IGES-Format. Die Ergebnisse dieser Entwicklung gingen in die nachfolgenden IGES-Versionen ein. IGES Version 4 (1988) unterstützte zunächst nur GSG-Geometrien; in die Version 5 (1990) wurden auch Volumenmodelle basierend auf einem B-Representation Modell integriert.

Parallel zu IGES Version 5 wurde aufbauend auf der ESP-Schnittstelle eine weitere Schnittstelle mit einer umfassenderen, von IGES unabhängigen Datenstruktur unter dem Namen PDES (Product Data Exchange Specification) entwickelt. Die Entwicklung wurde im Gegensatz zu den früheren Entwicklungen (XBF, ESP) nicht von IGES übernommen, sondern ist inhaltlich in der STEP-Entwicklung aufgegangen. Die Bezeichnung PDES wurde in „Product Data Exchange using STEP“ abgeändert.

IGES ist heute in den USA stark verbreitet, in Europa wird es vorwiegend zum Austausch von Zeichnungen genutzt.

1997 wurde mit der Version 6.0 als ANSI-Standard die Entwicklung eingestellt. Zukünftige Normungsaktivitäten werden auf STEP verlagert.

### Aufbau, Modellinhalte und Konzeption

Innerhalb von IGES werden alle produktdefinierenden Daten als Datenmodellelemente definiert. Es werden dabei Elementtypen unterschieden, die über vorgegebene Typnummern (z.B. TYP116=Punkt) festgelegt sind. Für verschiedene Elementtypen sind unterschiedliche Definitionsformen vorgesehen. Diese werden über sogenannte Form-

nummern unterschieden und innerhalb einer IGES-Datei (Directory Entry Section) durch einen Eintrag markiert.

IGES unterscheidet bei der Definition von Datenmodellelementen zwischen Elementen denen

- der Modellraum als dreidimensionaler euklidischer Raum mit einem global festgelegten Koordinatensystem (Koordinatensystem  $x,y,z$ ), auf welches das gesamte Modell bezogen wird und
- der Definitionsraum (ebenfalls ein euklidischer Raum), Koordinatensystem  $XT, YT, ZT$ ), welcher elementspezifisch ist und sich von Element zu Element ändern lässt, zugrunde liegen kann.

Die einzelnen Elemente werden im elementeigenen Definitionsraum beschrieben und über eine Transformationsmatrix in die Gebrauchslage im Modellraum transformiert.

Aufgrund der grossen Anzahl an IGES-Elementen welche wiederum verschiedene Definition aufweisen werden oftmals von den verschiedenen Systemanbietern nicht alle Elemente mit all den verschiedenen Definitionen implementiert. So kommt es häufig zu Informationsverlusten bei der Übertragung von Produktdaten.

Der Verband Deutscher Automobilhersteller (VDA) hat aus diesem Grund zu Beginn der 80er Jahre eine definierte Untermenge (Subset) der von IGES bereitgestellten Elemente definiert und in einigen Fällen hinsichtlich der Definition präzisiert (VDA-IGES Subset VDA-IS), welche ein VDA-IS-Prozessor unterstützen muss.

### Dateiformat

Der Datenaustausch basiert auf einem genormten Datenmodell und erfolgt über eine normierte IGES-Datei.

Der Aufbau einer IGES-Datei sieht die folgenden Abschnitte (Sections) vor:

- In der Flag Section wird angegeben ob die IGES-Datei im komprimierten ASCII-Format oder im Binärformat vorliegt. Liegt die IGES-Datei im Standard ASCII-Format vor, entfällt die Flag Section.
- Die Start Section bildet den Anfang und beinhaltet einen vom Menschen lesbaren Kommentar.
- Innerhalb der Global Section wird der verwendete Preprozessor sowie das sendende System beschrieben. Ausserdem werden jene Informationen bereitgestellt, die von dem Postprozessor zur Dateiverarbeitung benötigt werden.
- Die Directory Entry Section enthält ein Verzeichnis der zu Übertragenden Modellinhalte sowie für jedes Element eine Informa-

tionen hinsichtlich der Einbettung in das jeweilige Produktmodell (z.B. Linienfont, Schraffuren, etc.).

- Innerhalb der Parameter Data Section sind die Parameter für die Elemente der Directory Entry Section verzeichnet.
- Die Terminate Section markiert das Ende einer IGES-Datei.

## 4.2 VDA-FS

Die Flächenschnittstelle des Verbandes der Deutschen Automobilindustrie (VDA-FS) wurde zu Beginn der 80er Jahre von Automobilherstellern und -zulieferern speziell zur Übertragung von Freiformflächen und -Kurven beliebigen Grades definiert.

Die VDA-FS zeichnet sich durch ihre Mächtigkeit und den geringen Umfang der Dokumentation aus. Die Version 1 hat normativen Charakter (DIN 66301).

Ende der 80er Jahre wurde eine Ergänzung zur VDA-FS Version 1 als Version erarbeitet und über den VDA und dem VDMA (Verband Deutscher Maschinen- und Anlagenbauer) veröffentlicht.

### Aufbau, Modellinhalte und Konzeption

Innerhalb der VDA-FS Version ist die Übertragung von Freiformflächen und -kurven beliebigen Grades in Polynomdarstellung definiert.

Geometrielemente	nichtgeometrische Elemente
<ul style="list-style-type: none"> <li>• Punkt</li> <li>• Punktfolge</li> <li>• Punktvektorfolge</li> <li>• Kurve in Polynomdarstellung</li> <li>• Fläche in Polynomdarstellung</li> </ul>	<ul style="list-style-type: none"> <li>• Anfangskennung</li> <li>• Kommentar</li> <li>• Strukturierung (SET)</li> <li>• Endkennung</li> </ul>

Tabelle (T007srpZ) Modellelemente der VDA-FS Version 1

Um neben der eigentlichen Freiformkurven- und Flächengeometrie auch deren Berandungen und topologische Zusammenhänge abbilden zu können, wurden sowohl die geometrischen als auch die nichtgeometrischen Elemente entsprechend erweitert (vgl. Tabelle T008srpZ))

Geometrielemente	nichtgeometrische Elemente
<ul style="list-style-type: none"> <li>• Punkt</li> <li>• Punktfolge</li> <li>• Punktvektorfolge</li> <li>• Kurve in Polynomdarstellung</li> <li>• Fläche in Polynomdarstellung</li> <li>• Kurve auf Fläche</li> <li>• begrenzte Fläche</li> <li>• Flächenverband</li> </ul>	<ul style="list-style-type: none"> <li>• Anfangskennung</li> <li>• Kommentar</li> <li>• Strukturierung (SET, Gruppe)</li> <li>• Transformationsgrössenmatrix</li> <li>• Transformationsliste</li> <li>• Endkennung</li> </ul>

Tabelle (T008srpZ) Modellelemente der VDA-FS Version 2

### Dateiformat

Das innerhalb der VDA-FS spezifizierte Datenmodell wird auf eine sequentielle Daten bestehend aus einer Folge von Datensätzen abgebildet. Jeder Datensatz besteht aus 80 ASCII-Zeichen. Die Spalten 73 bis 80 sind für die Eintragung der Datensatznummer bestimmt. Als Datentypen sind Integer, Real, Namen und Befehlswoorte definiert. Die Syntax der Dateneintragung lehnt an die NC-Programmiersprache APT (Automatically Programmed Tools) an.

Die VDA-FS ist in die drei Abschnitte strukturiert:

- Der *Dateikopf* beinhaltet die Anfangskennung mit dem Namen der Datei sowie die Angabe der Anzahl der nun folgenden Texteinträge mit den organisatorischen Informationen (Senderfirma, Projekt-/Dateiname, Gültigkeitsdatum, etc.).
- Die Liste der geometrischen Elemente welche zu übertragen sind. Einzelne Geometrielemente können mittels eines Gruppierungsmechanismus zusammengefasst werden.
- Die Liste der benötigten nicht geometrischen Elemente.

### 4.3 DXF

Bei DXF handelt es sich um einen nicht normativen Industriestandard der amerikanischen Firma Autodesk (Produkt AutoCAD). DXF umfasst ähnliche Informationsinhalte wie die frühen IGES-Versionen (vornehmlich 2D-Zeichnungen sowie 3D-Linien und Flächenmodelle) beschränkt sich dabei jedoch auf weniger Konstrukte (vornehmliche jene, welche Bestandteil der AutoCAD internen Datenstruktur sind). Die DXF-Dokumentation ist nicht öffentlich. Dennoch ist DXF auch bei Softwareprodukten anderer Hersteller als AutoCAD verbreitet.

Aufbau, Modellinhalte und Konzeption

Die DXF-Schnittstelle umfasst im wesentlichen jene Elemente, welche von AutoCAD unterstützt werden. Es kann dabei zwischen Elementen, die nicht hinsichtlich einer Bezugsebene definiert sind und deren Koordinaten sich auf das Weltkoordinatensystem beziehen (z.B. LINE, POLYLINE, 3D FACE, etc.), Elementen die hinsichtlich einer Bezugsebene definiert sind und deren Koordinatenwerte sich auf das jeweilige Elementkoordinatensystem beziehen (z.B. CIRCLE, ARC, TEXT etc.), Bemassungselementen welche sich auf das Weltkoordinatensystem oder auf Weltkoordinaten beziehen sowie Elementen, die unabhängig von Koordinatensystemen sind, unterschieden werden.

Daten- und Dateiformat

Der Datenaustausch basiert auf dem ASCII Datenformat oder einem Binärformat. Die DXF-Datei ist in 4 Abschnitte sowie einer Markierung des Dateiendes strukturiert:

- Die Header Section enthält allgemeine Angaben für eine zu übertragende Zeichnung.
- In der Table Section werden innerhalb einzelner Definitionstabellen Linientypen, Layer, Textarten, Ansichten, Benutzerkoordinatensysteme, Ansichten und Zeichnungsdarstellung angegeben.
- Die Block Section umfasst verwendete Blockdefinitionen wie z.B. Schraffuren oder assoziative Bemassungen.
- Innerhalb der Entity Section sind alle zu übertragenden Datenaustauschelemente aufgelistet.



## 5 Die ISO 10303 „STEP“

Die ISO-Norm 10303 STEP (Standard for the Exchange of Product Model Data) steht in der Tradition einer ganzen Reihe von Normen und Forschungsprojekten zum Austausch von Produktdaten für welche jeweils Produktdatenmodelle entwickelt wurden. Diese Produktdatenmodelle erlauben die Beschreibung statischer Zustände von Produktdaten mittels Dateien, die für den Austausch von Produktdaten zwischen verschiedenen Anwendungsprogrammen verwendet werden können.

## 5.1 Zielsetzung

Zielsetzung von STEP ist die Bereitstellung eines international standardisierten Referenzmodells, das alle im Lebenszyklus eines Produktes eines Produktes anfallenden Informationen beinhaltet. Basierend auf diesem Referenzmodell sowie Regeln hinsichtlich der Anwendung dieses Modells soll es möglich sein, Produktdaten einheitlich zu beschreiben und unter verschiedenen austauschen zu können. Dazu gehört insbesondere die Archivierung und der Austausch mittels Dateien und Produktdatenbanken.

Die STEP-Entwicklung wird international durch das Gremium ISO TC184/SC4 unter dem offiziellen Namen „International Automation Systems and Integration - Product Repräsentation and Exchange“ organisiert sowie durch die Arbeiten verschiedener nationaler Gremien unterstützt. Ein Grossteil der Entwicklung wurde innerhalb verschiedener internationaler Forschungsprojekte erarbeitet und in die Standardisierung eingebracht.

Mit dem Umfang und der Komplexität wurde vollkommenes Neuland in der Standardisierung betreten, weshalb zunächst eine klar definierte Architektur des Standards konzipiert werden musste.

Der grösste Unterschied zwischen STEP und anderen Standardisierungsbemühungen ist die Entwicklung eines in die Zukunft orientierten Standards. Dabei sollten die bis zu diesem Zeitpunkt gewonnenen Erfahrungen und Erkenntnisse in die Entwicklung eingebracht werden.

Hinsichtlich der Entwicklung wurden folgende Anforderungen gestellt:

- *Vollständigkeit:* Die Norm soll alle Produktdaten abbilden können. Zielsetzung ist die ganzheitliche Betrachtung und Handhabung von Produktinformationen im Produktlebenszyklus.
- *Archivierungsfähigkeit:* Neben dem Austausch von Produktdaten ist die Archivierung von Produktdaten Bestandteil von STEP.
- *Erweiterbarkeit:* Die Schnittstelle soll so konzipiert sein, dass sie hinsichtlich neuer Anwendungsbereiche oder technischen Voraussetzungen angepasst bzw. erweitert werden kann.
- *Effizienz:* Das zu spezifizierende Austauschformat bzw. dessen zugrunde liegende Datenstruktur soll zum einem möglichst effizient bezüglich des Laufzeitverhaltens der Prozessoren sein. Zum anderen sollte die Übertragungsdateigrösse möglichst gering sein.
- *Kompatibilität mit anderen Normen:* Bei der Entwicklung von STEP soll auf die Kompatibilität mit anderen ISO-Normen oder Standards geachtet werden. Hervorzuheben sind in diesem

Zusammenhang insbesondere die internen Schnittstellenstandards GKS OPENGL oder PHIGS.

- *Minimum an Informationseinheiten:* Die Anzahl an Informationseinheiten (Entities) sollte auf das nötigste beschränkt bleiben und diese geringstmögliche Überschneidungen untereinander aufweisen.
- *Entkopplung von logischer Datenstruktur und physischem Speicherformat:* In Anlehnung an die ANSI/SPARC-Dreischichten Architektur wird eine Drei-Schichten-Architektur vorgeschlagen, die zwischen der physischen Ebene (Dateiformate), der logischen Ebene (formale Spezifikation) und der Anwendungsebene unterscheidet.
- *Anwendungsspezifische Leistungsstufen:* Es werden hinsichtlich der verschiedenen Anwendungsgebiete Untermengen von Produktdaten gebildet. Prozessoren für diese speziellen Anwendungsgebiete decken nur das entsprechende Teilgebiet ab.
- *Dokumentation, Validierung und Zertifizierung:* Es sollen einheitliche Methoden und Werkzeuge zur Dokumentation, zum Test sowie zur Validierung der Schnittstellenspezifikation entwickelt werden.

Diese Anforderungen hatten massgeblichen Einfluss auf die Architektur der STEP-Norm und führte dazu, dass sich die Norm in vielen Aspekten von ihren Vorgängern unterschied. Die grundlegenden Auswirkungen bzw. den daraus resultierenden Konzepte sollen an dieser Stelle hervorgehoben werden:

- Die Spezifikation eines international anerkannten Produktdaten-Referenzmodells, dem sogenannten *integrierten Produktmodell:*

Im Mittelpunkt steht die eindeutige, implementierungsunabhängige Beschreibung von Produktdaten in einem integrierten Produktmodell. Zielsetzung ist die redundanzfreie Abbildung aller Informationen welche innerhalb des gesamten Produktlebenszyklus anfallen bzw. generiert werden. Die Redundanzfreiheit des Modells soll gewährleisten, dass die von einem CAx-System erzeugten, erweiterten oder veränderten Daten unmittelbar für alle weiteren CAx-Systeme zur Verfügung stehen und entsprechend der jeweiligen Funktionalität des CAx-Systems weiter detailliert bzw. mit weiteren Produktinformationen versehen werden können. Das integrierte Produktmodell selbst wird in allgemeine und anwendungsabhängige Basismodelle sowie Anwendungsprotokolle unterteilt. Die allgemeinen Basismodelle (integrated resources) bilden die Grundlage für die Pro-

duktbeschreibung und sind anwendungsunabhängig. Auf sie bauen die anwendungsabhängigen Basismodelle (Integrated Application Resources) auf. Sie erweitern die allgemeinen Konstrukte, so dass sie für eine ganze Reihe von Anwendungen verwendet werden können. Auf die Basismodelle setzen die Anwendungsprotokolle (Application Protocols) auf. Da für bestimmte Anwendungen nicht alle in den Basismodellen definierten Elementen benötigt werden und zusätzliche anwendungsspezifische Daten nötig sind, beschreiben die Anwendungsprotokolle Ausschnitte aus den Basismodellen mit Spezialisierungen für diese Anwendungen.

- Die Beschreibungs- und Entwicklungsmethoden zur formalen Computer verarbeitbaren Spezifikation des Produktmodells:  
Innerhalb der STEP-Norm ist zum einen die Vorgehensweise bzw. der Ablauf bei der Entwicklung bzw. der Spezifikation anwendungsspezifischer Partialmodelle definiert. Zum anderen werden die hierfür benötigten Methoden und Werkzeuge festgelegt wie zum Beispiel IDEF0 einer Beschreibungsmethode für funktionelle Abläufe, welche auch unter dem Namen SADT (Structured Analysis and Design Technique) bekannt ist.  
Die objektorientierte Informationsmodellierungssprache EXPRESS bildet die Grundlage sämtlicher genormter Modellbeschreibungen innerhalb der ISO 10303. EXPRESS ermöglicht die Beschreibung der Modellstruktur sowie die Beschreibung von gültigen Modellzuständen durch Constraints. Obwohl EXPRESS viele programmiersprachliche Konzepte umfasst, handelt es sich nicht um eine Programmiersprache (dazu fehlen z.B. Ein- Ausgabeoperationen, ein Hauptprogramm sowie Funktionen zur Speicherverwaltung), sondern vielmehr um eine Daten- bzw. Informationsbeschreibungssprache.
- Die Implementierungsmethoden zum Produktdatenaustausch mittels Dateien, Hauptspeicherformaten oder Datenbanken:  
Während bisherige Dateiformate eher dem Lochkartenformat gleichen (z.B. das von IGES, bei dem die Spalten 1-73 die Daten, Spalte 74 einen Abschnittsidentifikator und die Spalte 75-80 die fortlaufende Zeilennummerierung enthält) spezifiziert STEP ein flexibles und auch für den Menschen lesbares Dateiformat, basierend auf der ASCII 7 Bit Kodierung.  
Ausserdem ermöglicht STEP nicht nur den Produktdatenaustausch per Datei, sondern unterstützt auch den Aufbau von Produktdatenbanken, auf welche mittels einer genormt

Schnittstelle von verschiedensten CAx-Systemen bzw. PDM-Systemen oder Archivierungssystemen zugegriffen werden kann.

- Die Konformitätstests (Conformance Testing Methodology and Framework) zur Überprüfung und Validierung der STEP unterstützenden Software, wie z.B. Prozessoren.

## 5.2 Architektur und Aufbau

Die oben beschriebenen grundlegenden Konzepte spiegeln sich in der Architektur und dem Aufbau der ISO 10303 wider.

Insgesamt gliedert sich die ISO 10303 in acht Themengebiete, die als Serien bezeichnet werden. Jede Serie besteht aus einzelnen Teilnormen, bzw. Dokumenten die als Parts bezeichnet werden.

Die *10er-Serie Description Methods* enthält Normdokumente, welche die Spezifikationsmethoden festlegen, mit denen alle Modellbeschreibungen innerhalb der ISO 10303 definiert werden. Innerhalb des Parts 11 *Description Methods: The EXPRESS Language Reference Manual* wird die Datenmodellierungssprache EXPRESS definiert, welche auch ausserhalb der Standardaktivitäten von STEP wie zum Beispiel innerhalb der ISO 13584 (P-Lib) Anwendung findet. Part 12 *EXPRESS-I Language Reference Manual* definiert die Instanzierung von EXPRESS-Daten. Mittels EXPRESS-I können zum einem Objektinstanzen, Schemainstanzen sowie Sammlungen von Schemainstanzen (Modelle) dargestellt werden. Ausserdem können Testdaten (abstract test cases) bereitgestellt werden. Die *0-Serie* beinhaltet lediglich den Part 1 [ISO 10303-1] *Overview and Fundamental Principles*. In diesem Dokument sind die Grundlagen bzw. fundamentalen Prinzipien festgehalten. Es enthält allgemeine Erklärungen, Terminologiedefinitionen sowie Entwurfsprinzipien und -vorgehen für die Definition von Datenmodellen.

Die *20er-Serie Implementation Methods* umfasst Implementierungsmethoden und enthält Normen, die eindeutige Abbildungsvorschriften auf verschiedene Implementierungsformen festlegen. Part 21 *Clear Text Encoding of the Exchange Structure* definiert ein neutrales Austauschformat *Physical File*. Part 22 *STEP Data Access Interface (SADI)* definiert eine navigierende Zugriffsschnittstelle für beliebige Datenhaltungen wie zum Beispiel Datenbanken oder Dateien. Es spezifiziert abstrakte Datenstrukturen und Zugriffsfunktionen. Die Parts 23 und 24 definieren Anbindungen an die Programmiersprachen C und C++. Eine Anbindung an IDL (die Interface Definition Language der Common Object Request Broker Architecture, CORBA) ist als Anhang von Part 23 enthalten.

Die *40er-Serie integrated Generic resources* stellt eine Art Klassenbibliothek bereit, auf denen die branchenspezifischen Datenmodelle (Application Protocols) aufbauen. Gemeinsam mit den anwendungsspezifischen Basismodellen (100er-Serie) bilden diese das eigentliche integrierte Produktmodell. Zielsetzung ist neben einer Reduktion der Zahl von Datentypen oder Funktionen mit gleichem Namen, aber verschiedenen Datenmodellen die Verwendung gemeinsamer Grundklas-

sen die die Interoperabilität von Daten verschiedener Branchen ermöglichen soll.

Die *100er-Serie integrated application resources* beinhaltet die anwendungsabhängigen Partialmodelle, welche die anwendungsneutralen Partialmodelle um anwendungsspezifische Konstrukte erweitern. So nutzt z.B. Part 101 *Draughting* die allgemeinen Konstrukte des Part 46 Visual Presentation und erweitert dieses um die für die Abbildung technischer Zeichnungen notwendigen Konstrukte.

Die *200er-Serie Application Protokolls* besteht aus den bereits erwähnten Anwendungsprotokollen (application protocol). Dabei handelt es sich um funktions- oder branchenspezifische Datenmodelle, welche die einzelnen Konstrukte des integrierten Produktmodells spezialisieren, um den Anforderungen jeweiligen der Anwendung gerecht zu werden. Zwei der zur Zeit geplanten 26 Anwendungsprotokolle liegen derzeit als Standard vor. Es handelt sich dabei um die Parts 201 Explicit Draughting und 203 Configuration Controlled Design. Einen gesamthaften Überblick über die derzeit in Entwicklung befindlichen und bereits abgeschlossenen d.h. als Standard vorliegenden Anwendungsprotokolle gibt 1.5.

Die *300er-Serie Abstract Test Suites* umfasst zu jedem Anwendungsprotokoll der 200er-Serie die Definition abstrakter Testzyklen. Dabei ist für jedes Anwendungsprotokoll mit der Nummer 200+x der Testsatz mit der Nummer 300+x vorgesehen. Innerhalb einer Test Suite werden implementierungsunabhängige Testfälle beschrieben, mittels welchen im Rahmen eines Konformitätstest überprüft werden kann, in wie weit sich eine Implementierung konform zur Norm verhält.

Die *500er-Serie Application Interpreted Constructs* beinhaltet eine Bibliothek interpretierter ARM Konstrukte (auf die ARM-Konstrukte wird in Die Anwendungsmodelle (application resources) eingegangen) als Application Interpreted Construct (AIC) bezeichnet, welche gleichermaßen innerhalb verschiedener Anwendungsprotokolle benötigt werden. Auf diese kann bei der Entwicklung eines neuer Anwendungsprotokolle zurückgegriffen werden. Zielsetzung ist einerseits die geforderte Interoperabilität verwandter Anwendungsprotokolle untereinander. Zum anderen kann der Entwicklungsaufwand für Anwendungsprotokolle reduziert werden. Zur Zeit liegen vornehmlich AIC hinsichtlich der Abbildung von geometrischen Repräsentationen vor (z.B. Part 501 Edge-based Wireframe, Part 504 Draughting Annotations), welche von einer Vielzahl von AP's benötigt werden.

1994 wurde die sogenannte Initial Release von STEP als internationale Norm verabschiedet. Darin enthalten sind die Spezifikationsprache EXPRESS (Part 10), die Definition für den Austausch über

sequentielle Dateien (Part 21) sowie die Grundkonzepte für die Konformitätsprüfung (Part 31). Der Umfang des integrierten Produktmodells der Initial Release beinhaltet neben dem Integrationskern (Part 41), folgende Partialmodelle:

- die Geometrie und Topologie (Part 42),
- die Repräsentationstruktur (Part 43),
- die Produktstruktur und Produktkonfiguration (Part 44),
- die Visualisierung von Produktmerkmalen (Part 46) sowie
- ein Modell für Technische Zeichnungen (Part 101).

Basierend auf diesem integrierten Produktmodell wurde ein Anwendungsprotokoll für den Austausch von technischen Zeichnungen (AP 201: Explicit Drafting) sowie ein Anwendungsprotokoll für die Produktkonfigurationen und Produktstrukturen basierend auf einer 3D Geometrie (AP 203: Configuration Controlled Design) definiert. Die Initial Release stellt gewissermaßen die STEP Version 1.0 dar. Für das Jahr 2000 ist die STEP Release 2 geplant innerhalb welcher die integrierten Ressourcen erweitert und teilweise optimiert werden sollen. Mit der Verabschiedung der überarbeiteten und erweiterten integrierten Ressourcen werden ist es vorgesehen eine Reihe neuer AP ebenfalls zu verabschieden.



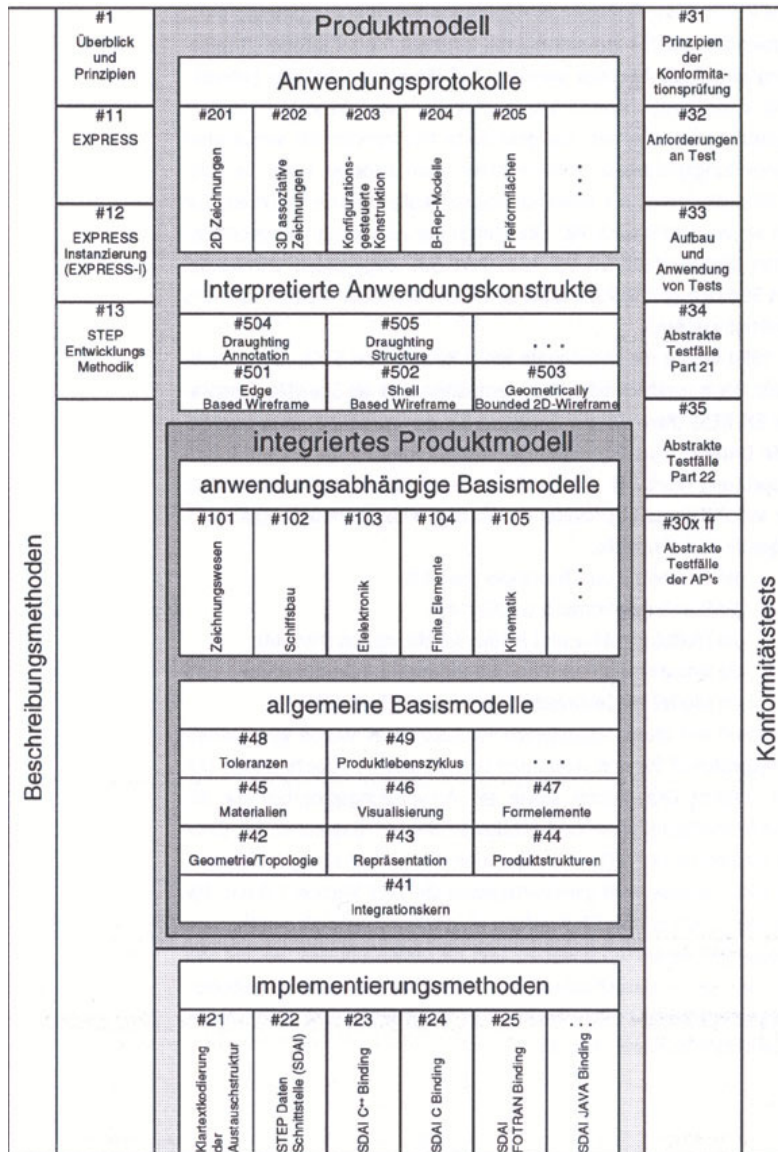


Bild (B056srpZ) Der Aufbau der STEP-Norm

### 5.3 Die Integrierten Ressourcen (integrated resources)

STEP basiert auf einem integrierten Produktmodell, welches sich aus einer Anzahl an Teilmodellen, Partialmodelle genannt, zusammensetzt (z.B. Geometriemodell, Materialmodell, Oberflächenmodell, etc.). Die einzelnen Partialmodelle werden von Experten des jeweiligen Fachgebietes definiert. Dabei werden die zur Beschreibung von Produkten relevanten Merkmale ermittelt und formal spezifiziert. Diese Spezifikation erfolgt in EXPRESS. Die einzelnen Partialmodelle werden schliesslich zu einem einheitlichen, redundanzfreien Gesamtmodell integriert. Dies geschieht mittels einem sogenannten Integrationskern.

Die gesamte Menge dieser Partialmodelle werden als die integrierten Ressourcen (integrated Resources) oder auch allgemeine Basismodelle bezeichnet. 5 dieser Partialmodelle sind bereits industrieller Standard (Part 41, Part 42, Part 43, Part 44 und Part 46). Part 49 steht kurz vor der Verabschiedung als internationaler Standard (Draft International Standard, DIS) Die weiteren Partialmodelle befinden sich teilweise noch in Entwicklung (Part 45 und Part 47) oder die Entwicklung ist aufgrund mangelnder Interesse ins Stocken geraten (Part 48).

Im folgenden soll nun auf die einzelnen Partialmodelle eingegangen werde. Jene Partialmodelle, welche im Hinblick auf das Produkt Daten Management von Bedeutung sind, werden entsprechend detaillierter beschrieben.

#### Der Integrationskern (Part 41)

Der Integrationskern spezifiziert die Grundkonzeption des gesamten integrierten Produktmodells. Die einzelnen Teilmodelle nutzen, detaillieren und ergänzen diese Grundkonzepte. Der Integrationskern gliedert sich in folgende drei Bestandteile:

- Die *allgemeinen Basiskonstrukte* definieren Datenmodellbausteine, die von allen Basismodellen, Anwendungsmodellen sowie Anwendungsprotokollen direkt genutzt werden können. Dazu gehören Datenmodelle zur Abbildung:
  - eines Masseinheitssystems
  - von Gruppieren von Produktdaten
  - von Datum und Zeit
  - von Personen-, Organisations-, und Projektinformationen,
  - Freigabeabläufen und -zuständen
  - von Referenzen auf STEP-externe Spezifikationen sowie
  - eines Grundmusters für Aktivitäten, die zur Änderung der Produktdaten führen.

- Die *allgemeinen Produktbeschreibungsressourcen* beschreiben auf Basis eines 4 Ebenenkonzeptes die Grundstruktur des integrierten Produktmodells, wobei jedes durch ein eigenes Schema repräsentiert wird:
  - Die innerste Ebene ermöglicht die Abbildung des Anwendungskontextes (*application\_context\_schema*). Die Beschreibung eines Produktes kann damit in einen bestimmten Kontext gestellt werden, z.B. einer bestimmten Lebenslaufphase oder der Identifikation des jeweiligen Anwendungsmodells z.B. dem gewählten Anwendungsprotokoll.
  - Die zweite Ebene definiert Mechanismen zur Produktbeschreibung (*product\_definition\_schema*). Zentraler Bestandteil ist die Informationseinheit Produkt, die hierarchisch strukturierbaren Produktkategorien, Produktversionen und sichtenspezifische Produktbeschreibungen. Eine Sicht (z.B. die Produktlebensphase) wird eindeutig über den Anwendungskontext definiert. Ausserdem ist ein Konstrukt zur Abbildung von Beziehungen zwischen einzelnen Produktbeschreibungen definiert. Dieses wird innerhalb des Parts 44 hinsichtlich dem Aufbau von Produktstrukturen spezialisiert. Wichtig ist in diesem Zusammenhang, dass die Produktstruktur nicht direkt auf Basis des Konstruktes *product* definiert wird, sondern über das Konstrukt *product\_version*. Bei dem Konstrukt *product\_version* handelt es sich um ein sogenanntes konfigurierbares Produkt (vgl. Kapitel 5), welches entsprechend der gewählten Konfiguration unterschiedliche Produktstrukturen aufweisen kann. Ein konfigurierbares Produkt stellt somit eine Abstraktion einer Menge konkreter Produkte im Sinne eines Platzhalters dar, der unterschiedlich besetzt werden kann.
  - Auf der dritten Ebene werden Produktmerkmale und deren Verknüpfung mit einem Produkt definiert (*product\_property\_definition\_schema*). Ein Merkmal eines Produktes wie z.B. die geometrische Produktgestalt kann dabei unterschiedlich repräsentiert werden (z.B. durch ein Kanten- Flächen oder Volumenmodell). Derzeit (in der initial Release) ist lediglich die Produktgestalt als Merkmal definiert. Folgen soll u.a. das Material sowie die kinematische Beschreibung.
  - Die Definition der Repräsentation von Produktmerkmalen (*product\_property\_representation\_schema*) erfolgt auf der

vierten Ebene. Ziel der Trennung von der Definition und Repräsentation eines Merkmals ist die Möglichkeit einem Produktmerkmal verschiedene Repräsentationen zuordnen zu können. Dabei kann es sich zum einem um unterschiedliche Geometriemodelle (z.B. Drahtmodell, Flächenmodell CSG, B-Rep., etc.), oder kontextabhängig um verschiedene Gestaltrepräsentationen handeln. Ein Feder kann z.B. abhängig vom jeweiligen Einbauzustand im gespannten (eingebauten) oder entspannt (nicht eingebauter Zustand) repräsentiert werden.

- Die *allgemeinen Verwaltungsressourcen* ermöglichen die Beschreibung von administrativen Daten innerhalb der Anwendungsprotokolle. Die Anwendungsprotokolle können dadurch die entsprechende Basiskonstrukte nutzen bzw. interpretieren.

### Geometrie, Topologie und Geometriemodelle (Part 42)

Innerhalb der Partialmodelle *Geometrie* (`geometry_schema`), *Topologie* (`topolgy_schema`) und *Geometriemodelle* (`geometric_model_schema`) werden die benötigten Konstrukte zur geometrischen und topologischen Beschreibung von Produkten definiert und dienen der expliziten Repräsentation der geometrischen Gestalt eines Produktes.

- Im Geometrieschema werden Vektoren, Richtungen, Punkte, Kurven und Flächen abgebildet. NURBS werden unterstützt.
- Das Topologieschema stellt Konstrukte zur Abbildung von topologischen Zusammenhängen basierend auf einer graphentheoretischen Beschreibung von Nachbarschaftsbeziehungen dar. Das Topologieschema wurde zur Abbildung von B-Rep-Modellen entwickelt.
- Das Partialmodell Geometriemodelle erlaubt den Aufbau von Draht- Flächen und Volumenmodelle unter der Nutzung der im Geometrie- und Topologieschema definierten Konstrukte. Das Schema beinhaltet generative Modelle (CSG- und Halbraummodelle) und akkumulative Modelle (B-Rep).

### Das Repräsentationsmodell (Part 43)

Das Repräsentationsmodell stellt einen allgemeinen Mechanismus zur Repräsentation von beliebigen Produktmerkmalen zur Verfügung. Durch die Trennung der Definition eines Produktmerkmals und den jeweiligen Repräsentationen ist es möglich die Produktgestalt mit verschiedenen Gestaltmodellrepräsentationen z.B. als Drahtmodell, Flächenmodell oder Volumenmodell zu verbinden.

### Das Produktstrukturmodell (Part 44)

Das Produktstrukturmodell dient der Abbildung von Informationen die die Struktur und Konfiguration eines Produktes definieren. Durch die Bereitstellung von neutralen Mechanismen soll eine durchgängige Anwendung innerhalb des gesamten Produktlebenszyklus gewährleistet werden.

Das Produktstrukturmodell spezifiziert verschiedene Mechanismen zur Baugruppenstrukturierung. Dabei handelt es sich um Spezialisierungen des innerhalb des Kernmodell definierten allgemeinen Konstruktes zur Abbildung von Beziehungen zweier Produktbeschreibungen (vgl. Bild B004srpZ)

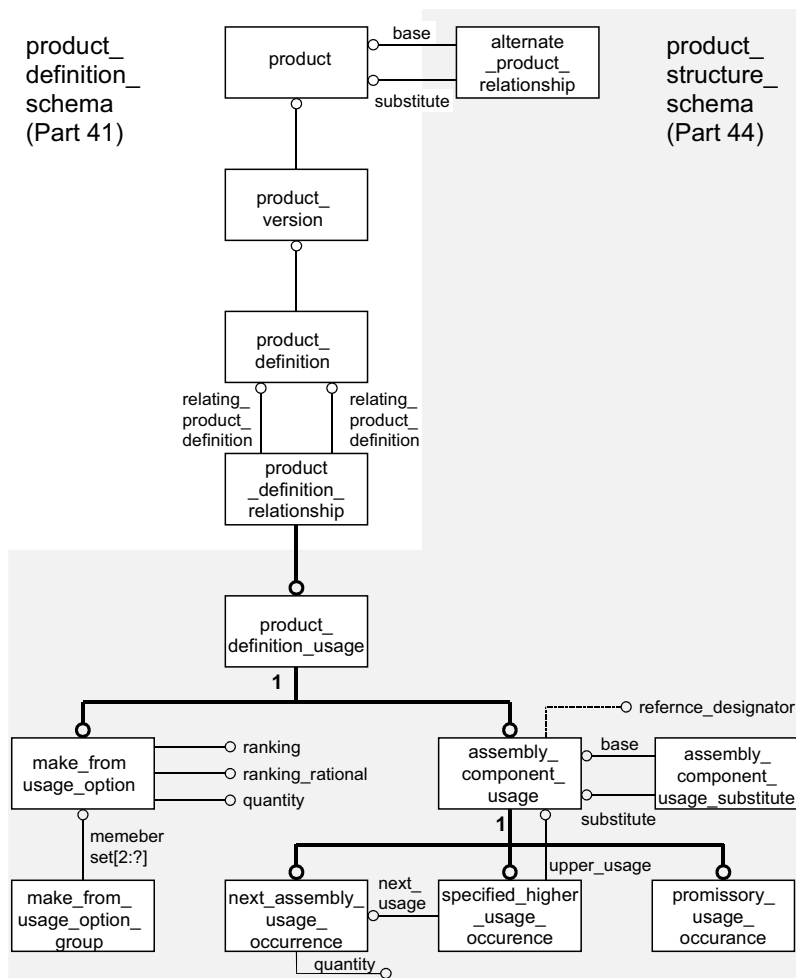


Bild (B004srpZ) Konstrukte zur Beschreibung von Produktstrukturen

- Der Aufbau einer Baugruppenstruktur erfolgt mittels dem Beziehungskonstrukt (`next_assembly_usage_occurence`). Die mehrfache Zuordnung des selben Produktes zu dem übergeordneten Produkt erfolgt durch die entsprechende Anzahl an Instanzen des Beziehungskonstruktes. Um innerhalb einer Produktstruktur zwischen der Verwendung eines Produktes innerhalb einer Unterbaugruppe, die ihrerseits mehrfach in einer übergeordneten Baugruppe verwendet worden ist, differenzieren zu können, kann mittels des Konstrukts `specified_in_higher_usage_occurence` die beliebige Verwendung individualisiert werden (vgl. Bild B0005srpZ). Diese Individualisierung ist deshalb notwendig, damit innerhalb einer Produktstruktur unterschiedliche Produktversionen deselben Produktes verwenden werden können.

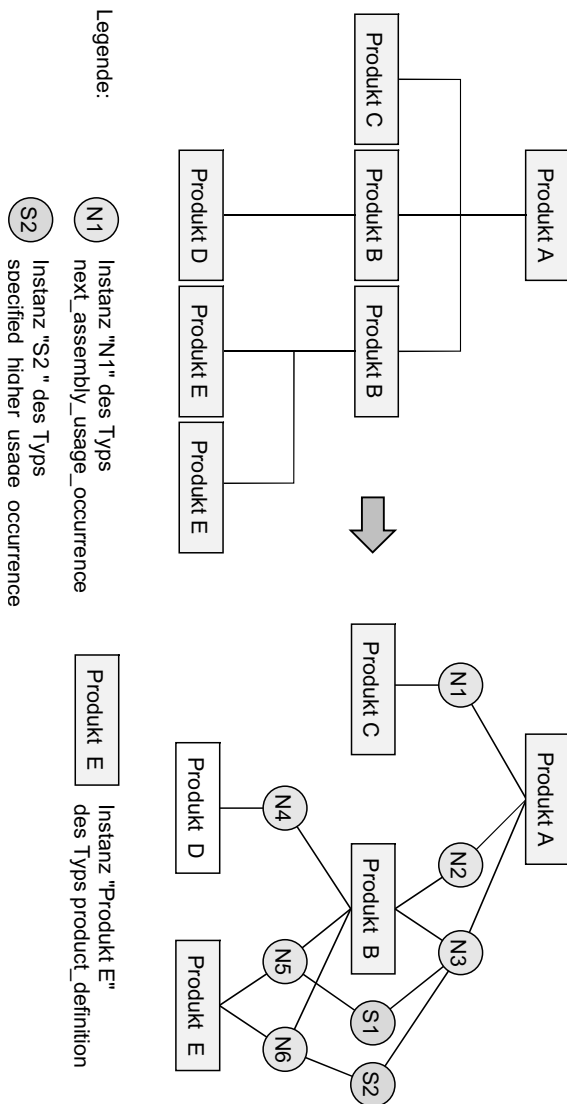


Bild (B005srpZ) Baugruppenstruktur auf Basis der Konstrukte next\_assembly\_usage\_occurrence und specified\_higher\_usage\_occurrence

- Eine vorläufige Baugruppenstruktur kann mittels des Konstruktes promissory\_usage\_occurrence definiert werden. Produkte können dadurch einer Produktstruktur zugeordnet werden ohne zu spezifizieren, auf welcher Stufe der Struktur dieses Produkt schliesslich verwendet werden soll.
- Zur Abbildung von Varianten bietet das Modell zwei Alternativen an:  
Mittels lokaler Alternativen ist es möglich für eine bestimmte Verwendung eines Produktes ein alternatives Produkt anzugeben. Dies erfolgt über das Beziehungskonstrukt

assembly\_component\_\_usage\_substitute, welches sich auf die oben beschriebenen Beziehungskonstrukte bezieht. Die Austauschbarkeit gilt in diesem Fall nur für jene Produkte für welche diese definiert worden ist.

Globale Alternativen erlauben einen von der Verwendung unabhängige Austauschbarkeit. Das entsprechende Konstrukt `alternate_product_relationship` bezieht sich in diesem Fall direkt auf den Typ Produkt.

- Bearbeitungszwischenstände können mittels dem Konstrukt `make_from_usage_option` abgebildet werden. Dabei unterscheidet sich die formale Repräsentation dieses Zwischenteils nicht von dem eines Fertigteils. Mittels des Konstrukts `make_from_usage_option_group` können Gruppen von Rohmaterialien gebildet werden. Auf diese Weise kann zwischen den beiden Ausgangsbauanteilen eines geschweissten Winkels unterschieden werden und diese individualisiert werden.



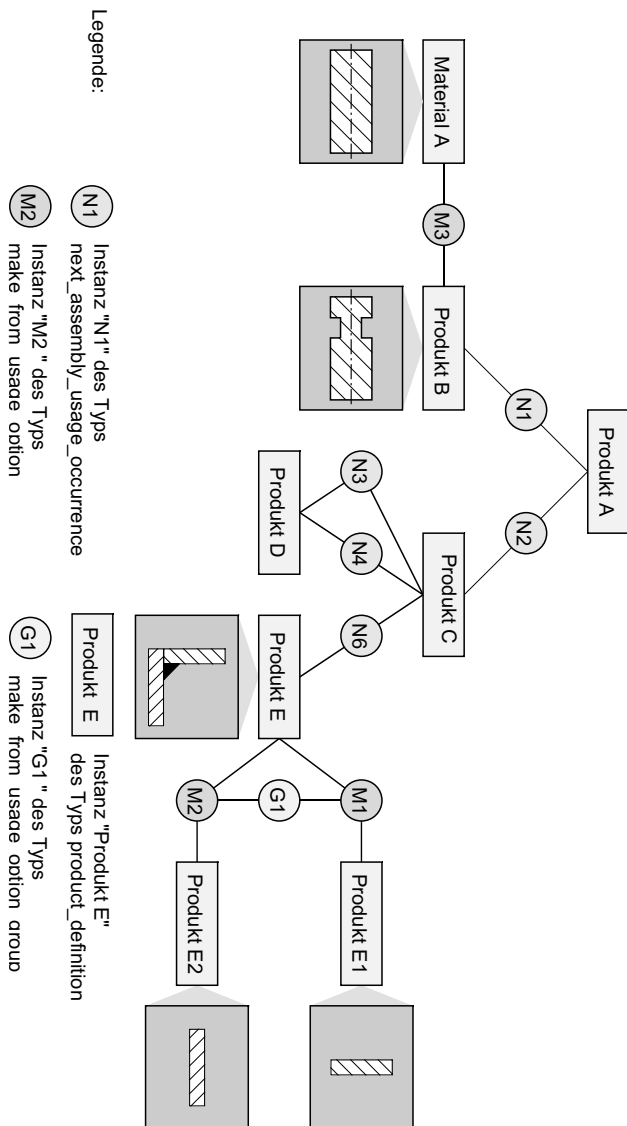


Bild (B006srpZ) Baugruppenstruktur auf Basis der Konstrukte make\_from\_usage\_optionv und make\_usage\_option\_group

- Baugruppenstruktur mit Bearbeitungszwischenzuständen  
Das Produkt Struktur Modell spezifiziert zusätzlich die Abbildung verschiedener Sichten auf eine Produktstruktur (z.B. aus auftragspezifischer Sicht). Hierbei ist die Identifikation von einzelnen Konfigurationen möglich, die jeweils einer bestimmten Produktversion zugeordnet werden können. Abhängig von den Kriterien Start- und Endtermin (date\_effecitivity), Identnummernbereich (serial\_numbered\_effecitivity) und Losgrösse

(lot\_effectivity) können unterschiedliche Produktstrukturen der gekennzeichneten Konfiguration zugeordnet werden.

Insgesamt stellt STEP damit ein sehr mächtiges Produktstrukturkonzept zur Verfügung. Es erfüllt sowohl die Randbedingungen seitens der auf der technischen Prozesskettenseite verwendeten CAx- bzw. PDM-Systemen als auch die der PPS-Systeme welche innerhalb der betriebswirtschaftlichen Prozesskette eingesetzt werden.

#### **Das Werkstoffmodell (Part 45)**

Das Werkstoffmodell dient der Abbildung von Materialeigenschaften eines Produktes. Werkstoffe können über die Materialelastizitätsmatrix und bestimmten Koeffizienten beschrieben werden. Unterstützt wird die Beschreibung sowohl von isentropen als auch anisentropen Werkstoffen. Das Partialmodell Werkstoffe ist nicht Bestandteil der Initial Release of STEP.

#### **Die Visualisierung von Produktmerkmalen (Part 46)**

Das Präsentationsmodell spezifiziert die Konstrukte zur graphischen Präsentation (z.B. Darstellungsattribute wie Farben oder Füllmuster) von Produktmodelldaten. Es wurde unter der Massgabe einer möglichst vollständigen Kompatibilität zu den existierenden internationalen Graphikstandards GKS-3D und PHIGS PLUS entwickelt.

Es wird dabei zwischen zwei den beiden grundlegenden Klassen von Präsentationsobjekten unterschieden:

- Produktgestaltobjekte (Objekte welche die Gestalt eines Produktes definieren) und
- nicht Produktgestalt (Texte, oder geometrische Elemente, die nicht Bestandteil der Produktgestalt sind)

#### **Das Toleranzmodell (Part 47)**

Das Toleranzmodell befindet sich derzeit in Entwicklung und ist damit nicht Bestandteil der Initial Release of STEP. Zielsetzung ist die Entwicklung eines Partialmodells zur rechnerinterpretierbaren Abbildung von Toleranzinformationen (Mass-, Form und Lagetoleranzen) eines Produktes und dessen Integration in das integrierte Produktmodell. Grundlage für das Partialmodell bilden die entsprechenden ISO Toleranznormen.

### Das Formelementmodell (Part 48)

Das Formelementmodell unterstützt die Beschreibung und Abbildung von Formelementen, welche von allgemeiner industriellen Interesse sind, wie z.B. Fasen, Einstiche, Gewinde. Diese können entweder explizit durch die Referenzierung der zugehörigen Geometrieelemente oder implizit über die entsprechende Erzeugungsvorschrift definiert sein. Die Entwicklung des Formelementmodell wird derzeit nicht weiter voran getrieben. Dies nicht zuletzt auch deshalb, weil die wichtigsten Formelemente (z.B. Gewinde) innerhalb einzelner AP's (AP214) spezifiziert und auf Geometrieonstrukte der bereits vorhandenen Integrierten Ressourcen abgebildet wurden.

### Das Lebenszyklusmodell (Part 49)

Das Lebenszyklusmodell (Process Structure and Properties) beinhaltet Modelle zur formalen Beschreibung von Produktdefinitions-, Herstellungs- sowie Demontageprozessen, d.h. im wesentlichen um den gesamten Produktlebenslauf.

Das Lebenszyklusmodell befindet sich derzeit in Entwicklung und ist damit nicht Bestandteil der Initial Release of STEP.

## 5.4 Die Anwendungsmodelle (application resources)

Die Anwendungsmodelle bilden allgemeine anwendungsspezifische Erweiterungen ab, wobei die Modellkonstrukte der allgemeinen Basismodelle um spezifische Konstrukte unter Berücksichtigung allgemeiner anwendungsbezogener Funktionen erweitert werden.

Als internationaler Standard liegt zur Zeit lediglich Part 101 vor. Die Arbeiten an den Parts 102 und 103 ruhen derzeit. Die Parts 104 und 105 befinden sich hingegen noch in der Entwicklung, die Verabschiedung zum Standard wird angestrebt.

### Das Anwendungsmodell technische Zeichnungen (Part 101)

Das Anwendungsmodell technische Zeichnungen (Draughting) stellt Konstrukte für die Speicherung, Verwaltung und den Austausch von technischen Zeichnungen bereit. Es bildet die Grundlage für die Anwendungsprotokolle aus dem Bereich Zeichnungswesen.

Das Modell umfasst Konstrukte zur Abbildung von

- grundlegenden Zusatzinformationen bezüglich technischer Zeichnungen wie zum Beispiel Annotationen oder Gruppen von Annotationen zur Darstellung von Informationen über das Produkt oder zur Erklärung der Zeichnungsinformationen hinsichtlich der Definition, Beschreibung und Administration eines Zeichnungssatzes sowie der enthaltenen einzelnen Zeichnungsblätter
- speziellen Annotationen für die Darstellung von Bemessungen, Toleranzen und darauf bezogenen Massangaben.
- Das Partialmodell *technische Zeichnungen* gliedert sich in drei einzelnen Schemata um die semantischen Unterschiede der spezifizierten Informationen trennen und einzeln hervorheben zu können.
- Das Definitionsschema (`drawing_definition_schema`) spezifiziert die benötigten Ressourcen zur Identifikation, Beschreibung, Organisation und Verwaltung von Zeichnungs- und Zeichnungsblattversionen.

Dabei wird sowohl auf Konstrukte des `presentation_organisation_schema` in welchem die Konzepte zur Definition einer Ansicht sowie Mechanismen von Ansichten spezifiziert sind als auch auf das `support_resource_schema`, welches Konstrukte zur Abbildung von Änderungsaktivitäten, die zu neuen Zeichnungsversionen führen, Verträge unter denen Zeichnungen erstellt, Sicherheitsstufen, verant-

wortliche Personen oder Verantwortungsbereiche bereitgestellt werden, zurück und spezialisiert diese entsprechend.

- Das Elementschema (`drawing_element_schema`) beschreibt die zur Darstellung eines Produktes benötigten von den verschiedenen Ingenieurdisziplinen unabhängigen grundlegenden Elemente wie Texte, Linien und Symbole.

Hinsichtlich der Integrationssystematik greift das Elementschema auf die innerhalb des `presentation_definition_schema` zurück, in welchem Konstrukte zur Darstellung von Objekten definiert sind.

- Das Bemassungsschema beinhaltet sämtliche zur Festlegung einer Bemassung benötigten Konstrukte. Es darin möglich eine Gruppierung von Basiselementen aus dem Elementschema als Bemassungsinformation zu identifizieren, in der z.B. die Masszahl und die jeweils zugeordnete Einheit festgeschrieben sind. Ausserdem können Beziehungen zwischen Bemassungen, wie gemeinsame Bezugspunkte oder gemeinsame Grundlinien abgebildet werden. Die Darstellung der Bemassungsinformation kann mit der bemassten Geometrie in Beziehung gebracht werden (assoziative Bemassung). Dabei werden die Schematas zur Definition von Grösseneinheiten referenziert.

### Das Anwendungsmodell Schiffsbau (Part 102)

Dieses Partialmodell gliedert sich in zwei Teilmodelle. Das Modell zur Übertragung strukturierter Daten für den Schiffsbau (Ship-Model) definiert spezielle Konstrukte zur Beschreibung der Geometrie, Topologie sowie deren Eigenschaften. Im Partialmodell für das Bauwesen (Core-Model) werden Erweiterungen der Basismodelle zur Abbildung bauspezifischer Anforderungen definiert.

Derzeit wird an diesem Anwendungsmodell nicht weitergearbeitet, d.h. das Entwicklungsprojekt innerhalb welchem die beiden Partialmodelle entwickelt wurden ist formal aufgehoben worden. Die bisher erarbeiteten Ergebnisse hinsichtlich des Partialmodells Schiffbau sind direkt in das AP 218 (Ship Structures) eingeflossen und werden dort weiterentwickelt.

Für das Partialmodell Bauwesen wird derzeit ein getrenntes Anwendungsmodell entwickelt (Part 106).

### Das Anwendungsmodell Elektrik/Elektronik (103)

Innerhalb des Anwendungsmodells Elektrik/Elektronik wird der funktionale Zusammenhang zwischen elektronischen Bausteinen abgebildet bzw. die Basiskonstrukte entsprechend detailliert.

Das Modell gliedert sich in die drei Betrachtungsebenen:

- Funktionale Hierarchie,
- Charakteristik und Verhalten sowie
- physikalische Beschreibung von Leiterplatten.

Das Entwicklungsprojekt wurde ebenfalls wie das des Part 102 formal aufgehoben und die inhaltlichen Ergebnisse den entsprechenden AP-Projekten übertragen.

#### **Das Anwendungsmodell Finite Elemente (104)**

Das Finite-Elemente Modell enthält Konstrukte zur Abbildung von FEM-Strukturen. Es unterstützt ein-, zwei-, dreidimensionale Finite Elemente. Weiterhin können Kontrollparameter für die Berechnung sowie Berechnungsergebnisse dargestellt werden.

#### **Das Anwendungsmodell Kinematik (105)**

Innerhalb des Kinematikmodells werden Konstrukte zur Beschreibung kinematischer Strukturen von Produkten mit Hilfe von starren Gliedern und Gelenken bereitgestellt. Dabei werden die kinematischen Freiheitsgrade der Gelenke beschrieben. Für kinematische Analysen können Startbedingungen festgeschrieben werden.

#### **Das Anwendungsmodell Bauwesen (106)**

Das Partialmodell Bauwesen ist aus den Arbeiten innerhalb des Anwendungsmodells 102 (vgl. oben) hervorgegangen. Zielsetzung ist die Entwicklung eines Partialmodells innerhalb welchen alle Produktinformationen hinsichtlich der Planung (z.B. Ausschreibungen, Ressourcenplanung, etc.), der Konstruktion und dem Bau von Bauwerken oder der zur Erstellung von Bauwerken erforderlichen Hilfseinrichtungen benötigt bzw. erstellt werden.

## 5.5 Die Anwendungsprotokolle (application protocols)

Zur Unterstützung von anwendungsspezifischen Prozessketten dienen Anwendungsprotokolle. Mittels eines Anwendungsprotokolls kann damit eine anwendungsabhängige Sicht auf das der ISO 10303 zugrunde liegende integrierte Produktmodell definiert werden. Sie dienen als Grundlage für die Implementierung von STEP-Prozessoren.

Ohne die Anwendungsprotokolle könnten die Systemanbieter beliebige Subsets der integrierten Ressourcen implementieren. Die gleichen Probleme, wie sie diesbezüglich bei bereits existierenden Standards aufgetreten (wie beispielsweise bei IGES) sind, wären die Folge. Die Anwendungsprotokolle stellen aus diesem Grund umfassend alle Anforderungen hinsichtlich einer Implementierung. Dies wird durch die Konstruktion von Anwendungsprotokollen von einer anwendungsspezifischen Interpretation der anwendungsunabhängigen Konstrukte der generischen Ressourcen. Ausserdem werden in Verbindung mit jedem Anwendungsprotokoll die entsprechenden Methoden für die Überprüfung der Konformität definiert.

Aufbau

Bei der Entwicklung von Anwendungsprotokollen werden im wesentlichen drei Spezifikationsphasen durchlaufen:

- Die *Spezifikation der Prozesskette* dient der definierten Abgrenzung des Anwendungsbereiches sowie der funktionalen Analyse der abzubildenden Prozesskette. Ergebnis dieser Phase ist das sogenannte Aktivitätenmodell (Application Activity Model AAM) basierend auf der SADT-Methode (Structured Analysis and Design Technique). Innerhalb diesem werden formal Aktivitäten beschrieben, die Produktdaten erzeugen oder verarbeiten, um die Identifikation der zwischen den einzelnen Aktivitäten fließenden Daten zu ermöglichen. Das AAM hat informativen Charakter innerhalb der Norm.
- Innerhalb der *Spezifikation des anwendungsspezifischen Referenzmodells* wird basierend auf dem Aktivitätenmodell die Struktur des Referenzmodells festgelegt, sowie die anwendungsspezifischen Daten detailliert und formal beschrieben. Darauf aufbauend werden die für die Anwendung benötigten Objekte und Strukturen definiert. Das Ergebnis ist das sogenannte Referenzmodell (Application Reference Model, ARM). Es handelt sich dabei um ein konzeptuelles implementierungsunabhängiges Datenmodell in der Terminologie der jeweiligen Anwendung. Da das ARM die Anforderungen der Anwender widerspiegeln soll, muss bei der Modellierung gewährleistet

sein, dass das Verständnis des Modells auf der Anwenderseite vorhanden ist.

Die Dokumentation dieses Modells erfolgt graphisch in einer formalen Informationsmodellierungssprache wie z.B. EXPRESS-G, IDEF1X oder NIAM, sowie durch eine nicht formale Definition der einzelnen Anwendungsobjekte (Application Objects), ihrer Beziehungen und deren Kardinalitäten (Application Assertions).

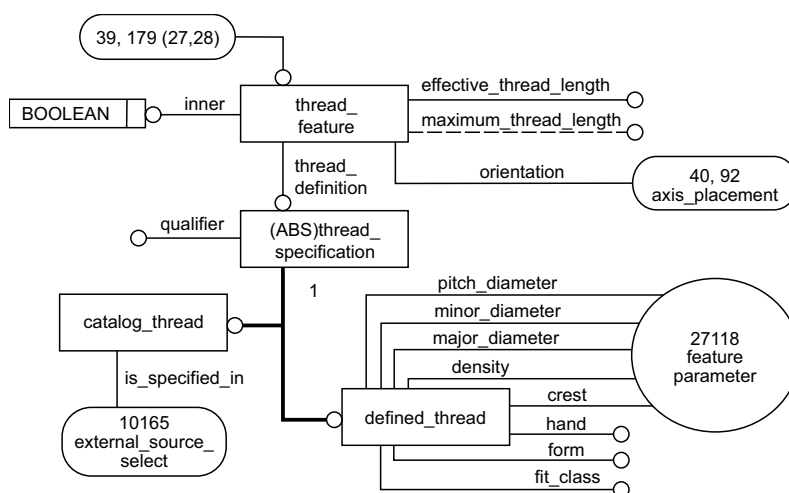


Bild (B007srpZ) ARM-Schema zur Definition von Gewinden (thread\_feature) in EXPRESS-G Notation

Das Gesamtmodell kann modularisiert werden. Dabei können einzelne Anwendungsobjekte sowie deren Beziehungen untereinander, welche ein oder mehrere Konzepte innerhalb des Gültigkeitsbereiches eines ARM umfasst und eine Anwendungsfunktion unterstützen zu einer Unit of Functionality zusammengefasst (aggregiert) werden. Das UoF-Konzept ermöglicht damit zum einem die bessere Strukturierung von ARMs. Ausserdem spielen UoF eine wesentliche Rolle hinsichtlich dem Zusammenspiel verschiedener Anwendungsprotokolle (AP interoperability), da die Überlappungsbereiche anhand der UoF-Definitionen leicht identifiziert und gegebenenfalls direkt verwendet werden können.

- Im letzten Schritt wird *das interpretierte Modell* (Application Interpreted Modell AIM) *aus dem ARM abgeleitet*. Das AIM ist in der Sprache EXPRESS definiert und enthält die Abbildung des Referenzmodells auf die integrierten Ressourcen. Dabei werden die generischen Objekte der Basismodelle entweder direkt



genutzt oder abhängig von den Anforderungen spezialisiert und mit zusätzlichen Zwangsbedingungen versehen. Diese Abbildung wird mittels einer Abbildungstabelle, dem mapping table dokumentiert. Ein mapping table enthält je ARM-Konstrukt eine Zeile in der die korrespondierenden AIM-Elemente, das jeweilige Basismodell und ggf. zutreffende Regeln beschrieben werden.

Identische UoFs müssen die Basismodellkonstrukte gleich interpretieren. Sie werden dann als interpretierte Basismodell-Konstrukte (Application Interpreted Constructs AIC) bezeichnet. Zudem wird eine Bibliothek von interpretierten Basismodell-Konstrukten zur Definition weiterer interpretierter Modelle zur Verfügung gestellt.

© ISO

ISO/CD 10303-214 (E)

#### 4.2.468 Thread\_feature

A Thread\_feature is a machining feature that is specified by a ridge of uniform section on the form of a helix on the external or internal surface of a cylinder.

##### EXAMPLE

Thread features may be used to screw parts together. The data associated with a Thread\_feature are the following:

- \* effective\_thread\_length;
- \* inner;
- \* maximum\_thread\_length;
- \* orientation;
- \* thread\_definition.

##### 4.2.468.1 effective\_thread\_length

The effective\_thread\_length specifies the length of the thread which is usable by the application.

##### 4.2.468.2 inner

The inner specifies if the thread is applied as an inner thread or as an outer thread.

##### 4.2.468.3 maximum\_thread\_length

The maximum\_thread\_length specifies the usable thread length plus the runout. The maximum\_thread\_length need not be specified for a particular Thread\_feature.

##### 4.2.468.4 orientation

The orientation specifies an Axis\_placement whose location represents the start of the thread and whose z-axis points into the direction of the possible runout. See Thread feature to Axis placement for the application assertion.

##### 4.2.468.5 thread\_definition

The thread\_definition specifies the Thread\_specification that contains the parameters specifying the thread. See Thread feature to Thread specification for the application assertion.

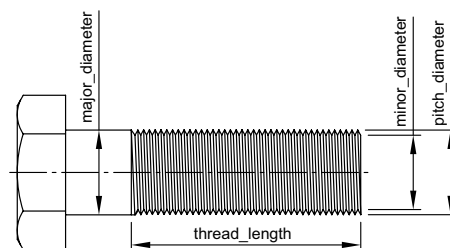


Bild (B008srpZ) Textuelle Beschreibung des benötigten Gewinde-Konstruktes

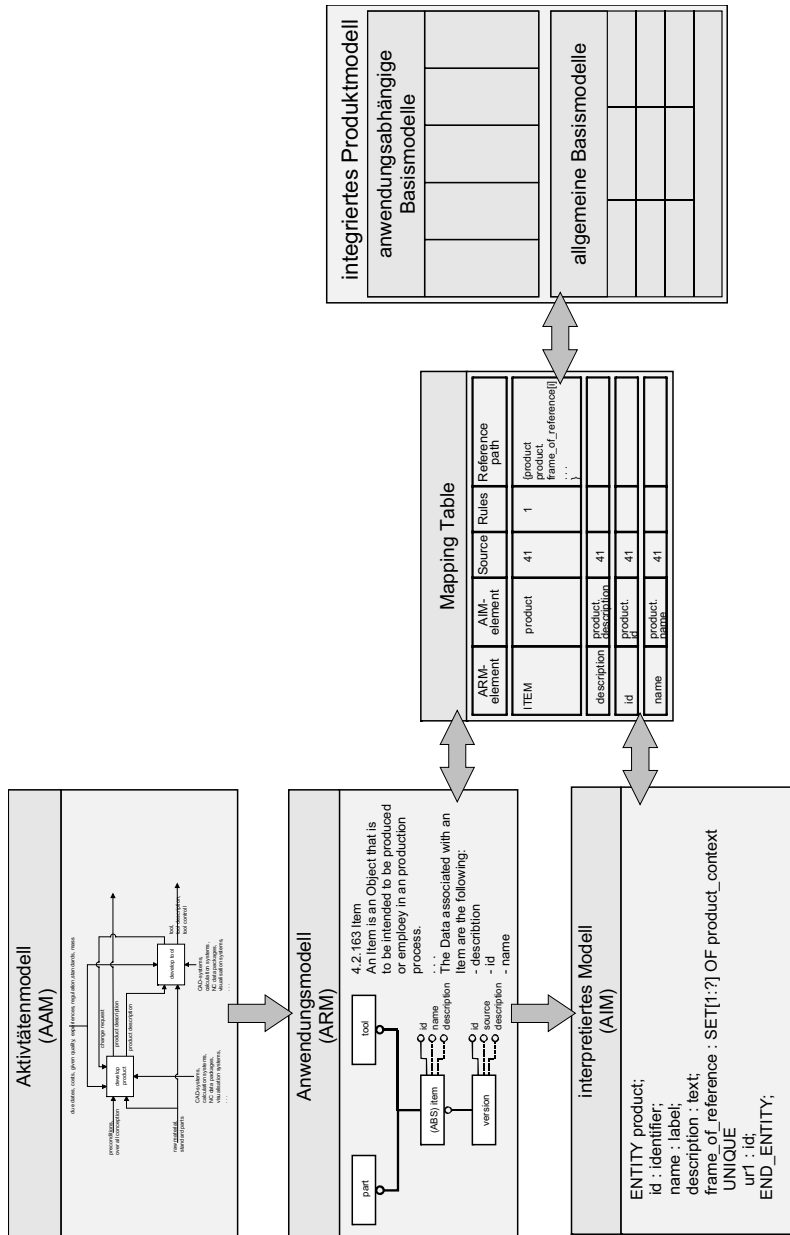


Bild (B009srpZ) Komponenten und Architektur eines Anwendungsprotokolls (AP)

Ein AIM kann in einzelne *Conformance Classes* strukturiert werden. Dies geschieht aus dem Grund da der Gültigkeitsbereich eines AIM in der Regel bei weitem grösser ist als die Funktionalität einzelner Anwendungssysteme. Beispielsweise unterstützen eine Reihe derzeitiger CAD-Systeme noch keine B-Rep-Repräsentationen. Entsprechend können alle Conformance Classes ausser jener innerhalb welcher die B-Rep-Repräsentation definiert ist implementiert und genutzt werden.

Innerhalb einer Conformance Class wird die Untermenge eines AIMS spezifiziert, die von einer Implementierung unterstützt werden muss, um konform mit dieser zu dieser Conformance Class eines AP zu sein. Mit der Initial Release von STEP wurden 3 Anwendungsprotokolle als internationale Norm verabschiedet. Es handelt sich dabei um

- AP 201 „Explicit Draughting“ (Technische Zeichnungen),
- AP 202 „Associative Draughting“ (assoziativ abgeleitete Technische Zeichnungen) und
- AP 203 „Configuration Controlled Design“ (3D Geometrie).

AP 201 und AP 202 haben mit dem Aufkommen leistungsfähiger 3D CAD-systeme praktisch keine Bedeutung bekommen.

Von grosser Bedeutung ist jedoch das sich derzeit noch in Entwicklung befindliche AP 214. Im folgenden sollen AP 203 und AP 214 kurz vorgestellt werden. Tabelle T010srpZ gibt einen Überblick.

### Das Anwendungsprotokoll 203

Das Anwendungsprotokoll 203 *Configuration Controlled Design* zielt auf die Gestaltungsphase mechanischer Produkte ab. Im Fokus stehen dabei neben den verschiedenen Formen der dreidimensionalen Gestaltsrepräsentation, Informationen zum Konfigurationsmanagement und zur Verfolgung von Änderungen des Produktdesigns.

Modellinhalte

- Daten hinsichtlich der Produktdefinition
  - produktdefinierende Daten (Identnummer, Bezeichnung, Version, etc.)
  - organisatorische Daten (Verträge, Sicherheitsstufen, Zulieferer, etc.)
  - Produktstrukturen (Klassifikation, Baugruppen, Konfiguration, Bearbeitungszwischenzustände)
  - Baugruppenkonfigurationen
- Geometrie (Kanten- Flächen- und Volumenmodelle)
- Toleranzen
- Oberflächeneigenschaften

### Das Anwendungsprotokoll 214

Das Anwendungsprotokoll AP 214 Core date for Automotive Mechanical Design Processes beschreibt die Produktdaten von Teilen und Baugruppen in der Automobilentwicklung für den mechanischen Bereich wobei sowohl die Fahrzeugkomponenten als auch die Betriebsmittel für deren Herstellung abgebildet werden.

Das Modell umfasst Prozessketten für die Karosserie, Aggregate, Fahrgestell und Innenaustattung. Die Prozessketten setzen sich aus

den Prozessen Styling, Konstruktion, Produktionsvorbereitung, Betriebsmittel- und Methodenentwicklung, Betriebsmittelherstellung und Qualitätskontrolle zusammen.

Hinsichtlich dieser Informationsinhalte soll

- der Austausch zwischen verschiedenen CAx-Systemen
- das Produkt Daten Management, sowie
- die Langzeitarchivierung

ermöglicht werden.

### Modellinhalte

Das Informationsmodell enthält etwa 560 Objekte, die aus Anwendersicht definiert wurden und etwa 750 Objekte die als konzeptuelles Modell für die Implementierung spezifiziert wurden.

Das AP 214 umfasst dabei folgende Klassen von Produktdaten:

- Daten hinsichtlich der Produktdefinition
- produktdefinierende Daten (Identnummer, Bezeichnung, Version, etc.)
- organisatorische Daten (Verträge, Sicherheitsstufen, Zulieferer, etc.)
- Produktstrukturen (Klassifikation, Baugruppen, Konfiguration, Bearbeitungszwischenzuständen)
- Produktgestalt
  - Geometrie (Kanten- Flächen- und Volumenmodelle
  - Formelemente
- Produktpräsentationen
  - Geometriedarstellung (Farbe, Schattierungen, Strichstärken)
  - Texte
  - Symbole
- Toleranzen
- Oberflächeneigenschaften
- Materialangaben
- Kinematik
- FE-Analysedaten
- Einzelteil- und Baugruppenbeschreibung in Form von technischen Zeichnungen
- Produkteigenschaften (Luftwiderstandsbeiwerte, etc.)

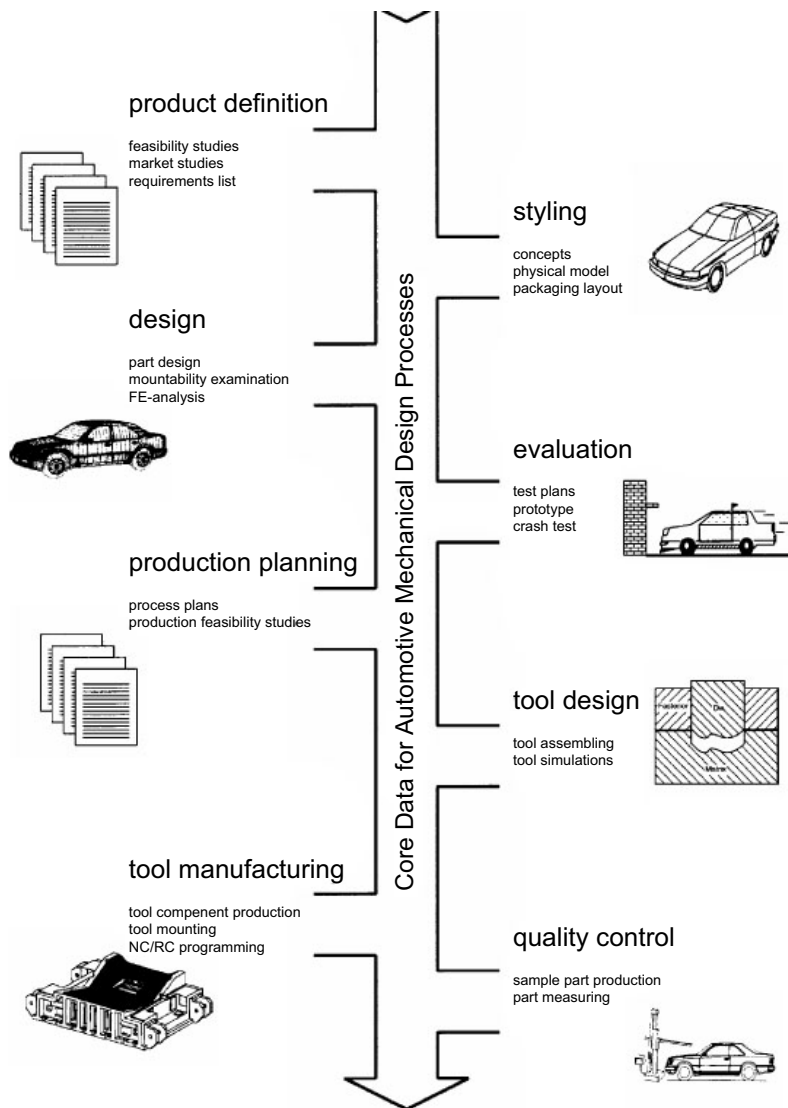


Bild (B010srpZ) Informationsinhalte des AP 214

### 5.5.1 Weitere Anwendungsprotokolle

Tabelle T010srpZ gibt einen Überblick über die derzeit aktuellen Anwendungsprotokolle.

AP	Titel	Anwendung	Status / Implementierung	Bemerkung
201	Explicit Draughting	Zeichnungen mit 2D-Geometrie	IS / nicht bekannt	geringe Bedeutung, Funktionalität von AP 214 abgedeckt
202	Associative Draughting	Zeichnungen mit Assoziationen zwischen 3D-Geometrie und Bemassung	IS / nicht bekannt	geringe Bedeutung, Funktionalität von AP 214 abgedeckt
203	Configuration Controlled Design	s.o.	IS/zahlreiche Implementierungen	von der US-Luft- und Raumfahrt-Industrie initiiert, weite Verbreitung auf dem US-CAD- Markt
204	Mechanical Design Using Boundary Representation	Volumenmodelle mit B-Rep Datenstruktur	Implementierungen nicht bekannt	geringe Bedeutung, Funktionalität von AP 203 bzw. 214 abgedeckt, entstammt der CAD*-Schnittstelle
205	Mechanical Design Using Surface Representation	Flächenmodelle	Implementierungen nicht bekannt	CC1 ist kompatibel zu VDA-FS V1, CC2 zu VDA-FS V2, CC3 entspricht AP 204
206	Mechanical Design Using Wireframe Representation	Kantenmodelle	Projekt gestoppt / nicht bekannt	geringe Bedeutung, Funktionalität von AP 214 bzw. teilweise 203 abgedeckt
207	Sheet Metal Die Planning and Design	Umformwerkzeuge zur Blechbearbeitung	Implementierungen nicht bekannt	geringe Bedeutung, Funktionalität von AP 214 abgedeckt
208	Life Cycle Management - Change Process		Implementierungen nicht bekannt	geringe Bedeutung, Projekt mehrmals mit jeweils veränderter Zielsetzung aufgestartet

Tabelle (T010srpZ) Überblick über die genormten bzw. in Entwicklung befindlichen AP's

AP	Titel	Anwendung	Status / Implementierung	Bemerkung
209	Design through Analysis of Composite & Metallic Structures	FE-Analyse für Verbundwerkstoffe im Bereich FEA-Daten	Implementierungen nicht bekannt	geringe Bedeutung, Funktionalität von AP 214 abgedeckt
210	Electronic Printed Circuit Assembly: Design and manufacture	Geometriedaten hinsichtlich der Leiterplattenentwicklung und -Herstellung	vorhanden, industrielle Nutzung	in Zusammenhang mit AP214 entwickelt
212	Electrotechnical Plants	Eletrotechnischer Anlagebau	Implementierungen nicht bekannt	geringe Aktivitäten
213	Numerical Control Plans for Machinde Parts	Bearbeitung von Dreh- und Frästeilen	Implementierungen nicht bekannt	
214	Core Data for Automotive Design	s.o.	IS / vorhanden	durch internationale Automobilindustrie stark vorangetrieben
215	Ship Arangement	Schiffbau	Implementierungen nicht bekannt	
216	Ship Moulded Forms	Schiffbau	Implementierungen nicht bekannt	
217	Ship Piping	Schiffbau	Pilot-Implementierungen	
218	Ship structures	Schiffbau	Pilot-Implementierungen	
219	-			Projekt abgebrochen
220	Printed Circuit Assembly: Manufacturing Planning	Herstellung von Leiterplatten	Implementierungen nicht bekannt	geringe Aktivitäten
221	Functional Data and their Schematics Representation for Process Plant	Grossanlagenbau, Verfahrenstechnik	Implentierungen vorhanden	

Tabelle (T010srpZ) Überblick über die genormten bzw. in Entwicklung befindlichen AP's

AP	Titel	Anwendung	Status / Implementierung	Bemerkung
222	Exchange of Product data for Composite Structures	Faserverbund	Implementierungen nicht bekannt	
223	Exchange of Design and Manufacturing Product Information for Cast Parts	Giessereitechnik	Implementierungen nicht bekannt	
224	Mechanical Product definition for process Plans using Maching Features	Fertigungstechnik	Implementierungen nicht bekannt	
225	Building Elements using explicit shape representations	Bauindustrie	Implementierungen vorhanden	
226	Ship Mechanical Systems	Schiffbau	Implementierungen nicht bekannt	
227	Plant Spatial Configuration	Grossanlagenbau	Implementierungen nicht bekannt	
228	Building Services: heating, Ventilation and air Conditioning	Bauindustrie	Implementierungen nicht bekannt	
229	Exchange of design and Manufacturing Product Information for Forged Parts	Fertigungstechnik	Implementierungen nicht bekannt	
230	Building structural Frame: Steelwork	Bauindustrie	Implementierungen nicht bekannt	
231	Process Engineering data: Process Design and process Specification of major Equipment	Bauindustrie	Implementierungen nicht bekannt	

Tabelle (T010srpZ) Überblick über die genormten bzw. in Entwicklung befindlichen AP's



AP	Titel	Anwendung	Status / Implementierung	Bemerkung
232	Technical data Packaging Core Information and Exchange	Austausch von PDM-Daten zwischen PDM-Systemen	Implementierungen nicht bekannt	im Zusammenhang mit AP203 entwickelt

Tabelle (T010srpZ) Überblick über die genormten bzw. in Entwicklung befindlichen AP's

## 5.6 Die Datenschnittstelle SDAI

Zu den Entwicklungszielen der STEP-Norm gehörte die bereits oben erwähnte Unabhängigkeit von der Rechnerumgebung sowie die Entkopplung der logischen Datenstruktur von dem physischen Speicherformat. Daraus resultierte die Spezifikation einer Zugriffsschnittstelle mit dem Namen SDAI (STEP Data Access Interface). Die SDAI-Schnittstelle soll es Anwendungsprogrammen erlauben über standardisierte Zugriffsoperationen einheitlich auf STEP-Datenhaltungssysteme zuzugreifen. Die Entwicklung dieser Schnittstelle geschieht vor allem in Hinblick auf die Abkehr von den bisher vorherrschenden dateiorientierten Arbeiten mit Produktdaten hin in Richtung einer Produktdatenverwaltung in Produktdatenbanken mit der Möglichkeit eines gesteuerten gleichzeitigen Zugriffs mehrerer Anwendungen auf die zugrunde liegende Produktdatenbank.

SDAI besteht im wesentlichen aus drei Teilen:

- Die Schemata definieren, wie die Metadaten einer STEP-Datenbank zu organisieren sind. Es handelt sich dabei um insgesamt fünf Schemadeklarationen die den Aufbau einer STEP-Datenbank spezifizieren.
- Die Operationen spezifizieren, wie auf Anwendungs- und Schemadaten zugegriffen wird.
- Es sind insgesamt 105 SDAI-Operationen (z.B. open session, close session zum Öffnen bzw. Schliessen einer STEP-Datenbank) aufgeteilt in 17 Klassen spezifiziert.
- Die Implementierungsebenen legen fest, wie gut eine Datenbank die Forderungen des SDAI erfüllt. Es ist damit wiederum möglich, schrittweise Schnittstellen zu implementieren, deren jeweiliger Leistungsumfang klar definiert ist.

## 5.7 Ausblick und Kritik

Inhaltlich lässt sich einerseits die Aufteilung des gesamten Produktmodells in drei Schichten aus heutiger Sicht kritisieren. Dass die Unterscheidung zwischen allgemeinen und anwendungsabhängigen Basismodellen nicht geglückt ist, ist daran zu ersehen, dass von den letzteren nur noch drei Teile weiterentwickelt werden. Die verbleibenden gehen in den verschiedenen Anwendungsprotokollen auf.

In diesem Zusammenhang ist die Entwicklung bzw. Definition des ARM-Schemas und deren nachfolgende Abbildung auf die integrierten Ressourcen nicht unumstritten. Dieser Vorgang wird als zu komplex und fehleranfällig erachtet, zumal das Mapping als solches interpretierbar ist. Das bedeutet dass verschiedene Personen ein ARM-Konstrukt auf unterschiedliche Weise auf die integrierten Ressourcen abbilden. Derzeit wird dieses Problem dadurch gelöst, dass der Mappingvorgang von einer einzelnen Person kontrolliert wird!!!

Viele Anwendungsprotokollen decken eine sehr breite Funktionalität ab. Die gilt insbesondere für das AP 214. Die Folge ist eine starke Überlappung der einzelnen Anwendungsprotokolle hinsichtlich der jeweiligen Modellinhalte und angestrebten Funktionalität. Die Folge ist eine grosse Anzahl an möglichen AP's zur Übertragung von Produktdaten. Sender und Empfänger müssen jeweils den selben Prozessor zur Verfügung stellen um die Daten ohne Informationsverlust austauschen zu können.

Unter dem Stichwort AP-Interoperability bzw. AP-Harmonisation wird derzeit versucht dieses sich zukünftig ergebende Problem zu lösen. Derzeit tritt dieses Problem deshalb nicht auf da lediglich zwei AP als Industrielle Standard verabschiedet sind und deren Modellinhalte und Funktionalität disjunkt sind.

Unter dem Stichwort AP-Interoperability wird derzeit eine Bibliothek einzelner bereits interpretierter Konstrukte erstellt (AIC) welche von den AP verwendet werden sollen. Unter dem Stichwort AP-Harmonisation sollen zum einem die Modellinhalte einzelner AP abgeglichen werden zum anderen sollen die einzelnen Conformance Classes der AP so definiert sein, dass eine möglichst grosse Anzahl lediglich auf AIC-Konstrukte zurückgreift.

Langfristig ist eine AP-Modularisation angedacht. Die derzeitigen sehr umfassenden AP sollen hinsichtlich der jeweiligen Modellinhalte und Funktionalität sehr stark eingeschränkt und diesbezüglich untereinander disjunkt sein.

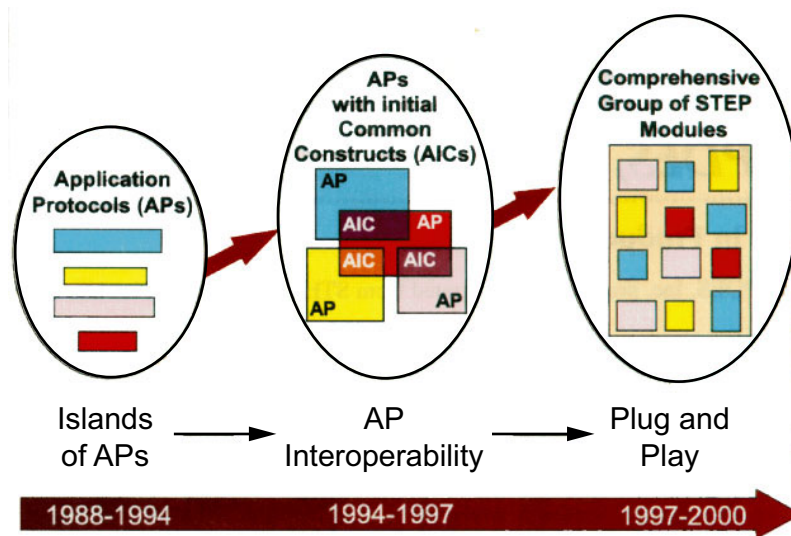


Bild (B011srpZ) AP-Modularisation

Die integrierten Ressourcen unterstützen derzeit nicht die Abbildung parametrischer Geometriemodelle. Ziel ist es daher, die Integrierten Ressourcen in der nächsten Zeit entsprechend zu erweitern. Methodische Ansätze hierfür bestehen bereits. Beispielsweise wurde innerhalb der Entwicklung der ISO 13584 Parts Library die Parameterisierung formalisiert.

Hinsichtlich der angesprochenen Erweiterungen im Rahmen einer kontinuierlichen Weiterentwicklung des Standards STEP stellt sich die der Frage hinsichtlich der Auswärts/Abwärtskompatibilität von STEP. Die Diskussion ist derzeit im Gange und tendiert zur alleinigen Unterstützung der Aufwärtskompatibilität.

Auf der einen Seite haben sich während der Entwicklung von STEP die technischen Möglichkeiten, wie zum Beispiel das Aufkommen parameterisierter CAD-Systeme, die Anforderungen beispielsweise das Arbeiten in virtuellen Unternehmungen sowie die technischen Möglichkeiten (mit dem Aufkommen der plattformunabhängigen Programmiersprache JAVA wird derzeit ein JAVA-Binding in Angriff genommen) immer wieder verändert. Auf der anderen Seite verzögern die demokratischen Strukturen in der internationalen Standardisierung sowie die unterschiedlichen Zielsetzungen einzelner Interessensgruppen (z.B. die Automobilindustrie oder die Luft- und Raumfahrtindustrie) die Entwicklung von STEP.

Als weitere Erschwernis ist der erhebliche Umfang der Dokumentation sowie die starke Verknüpfung der einzelnen Teile der Norm untereinander zu nennen.

Abschliessend soll an dieser Stelle darauf hingewiesen werden, dass trotz aller Probleme welche hinsichtlich der Entwicklung und Normierung von STEP aufgetreten sind, dieser Standard zukünftig mit Sicherheit eine zentrale Rolle im Hinblick auf den Austausch, Speicherung und Archivierung von Produktdaten einnehmen wird.

## 6 Integration von Norm- und Zukaufteile

Die Nutzung bereits im Unternehmen oder auch auf dem Markt vorhandener Lösungen gewinnt heute zunehmend an Bedeutung. Dabei kann es sich sowohl um standardisierte Produkte welche von verschiedenen Anbietern bezogen werden können als auch um spezialisierte Komponenten handeln, welche von einem einzelnen Anbieter angeboten werden. Die Potentiale einer Wiederholteilverwendung wurden bereits in Kapitel 6 „Objektklassifikation“ dargestellt. Diese Komponenten müssen bei den einzelnen Kunden nicht nur in die Konstruktion als solches sondern in das betriebliche Produktdatenmanagement integriert werden.

Produkte setzen sich aus einzelnen Komponenten zusammen, man spricht von der Produktstruktur. Innerhalb des gesamten Produktlebenszyklus also von der Konstruktion, Fertigung, Inbetriebnahme, Wartung und Entsorgung werden sukzessive Informationen hinsichtlich der in diesem Produkt verwendeten Komponenten erzeugt und dem Produktmodell hinzugefügt (vgl. Kapitel 1 „Einführung in die rechnerintegrierte Produktentwicklung“).

Wird eine Komponente ausserhalb eines Unternehmens beispielsweise durch einen Zulieferer definiert, sind die mit dieser Komponente verknüpften Informationen zunächst unabhängig von denen des Produktes innerhalb welchem diese Komponente verwendet werden soll. Mit der Integration einer Zukaufkomponente in ein Produkt müssen diese Informationen ebenfalls integriert werden. Diese Informationen müssen also zunächst zwischen unterschiedlichen Organisationen und CAx-Systemen ausgetauscht und in das jeweilige Produktmodell integriert werden können.

Der effiziente Einsatz moderner Informationstechnologien in der Produktentwicklung verlangt in diesem Zusammenhang den manuellen Anteil an der Informationsverarbeitung so gering wie möglich zu halten.

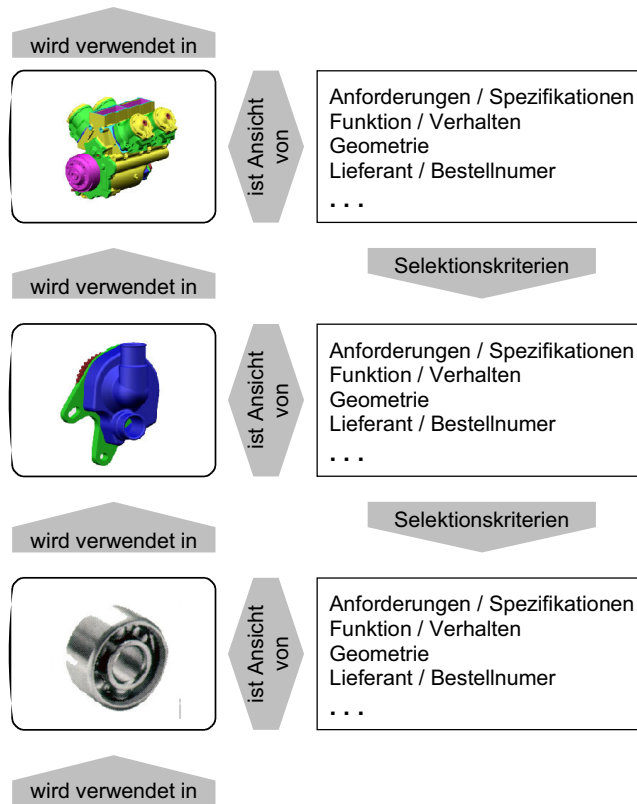


Bild (B012srpZ) Modulbasierte Produktentwicklung

Prinzipiell bestehen zwei Möglichkeiten für den Austausch bzw. zur Integration von Norm- und Zukaufteilen:

- Explizite Beschreibung aller auszutauschenden Elemente in einem neutralen Austauschformat wie zum Beispiel IGES, DXF oder STEP (s.o.).  
Damit muss für jede Bauteilvariante die vollständige geometrische Beschreibung durch den Anbieter erzeugt und übertragen werden. Daraus resultieren extrem hohe Datenmengen die sich hinsichtlich ihrem Informationsinhalt nur sehr gering unterscheiden.
- Implizite Beschreibung von ganzen Teilefamilien in einem neutralen Austauschformat. Hierfür werden ähnliche Bauteile zu Teilefamilien zusammengefasst und für diese eine allgemeingültige parameterisierte Erzeugungsvorschrift sowie eine Liste mit den möglichen Kombinationen der Parameterausprägungen erstellt und ausgetauscht. Der Vorteil liegt dabei in dem erheblich geringeren Datenumfang.

## 6.1 CAD-Teilebibliotheken

Nahezu für jedes kommerziell erhältliche CAD-System wurden aus diesem Grund Teilebibliothekssysteme entwickelt und integriert. Diese sind untereinander in der Regel jedoch nicht kompatibel und hinsichtlich ihrer Funktionalität stark durch den Leistungsumfang des jeweiligen CAD-Systems geprägt.

Teilebibliotheken moderner parameterisierter CAD-Systeme basieren auf einer impliziten Teilebeschreibung und bestehen im wesentlichen aus drei Bausteinen:

- parameterisierte systemspezifische CAD-Modelle,
- CAD-systemunabhängige Parametertabellen,
- ein den Parametertabellen übergeordnetes Ordnungs- bzw. Zugriffssystem.

Die Bibliotheken beinhalten in der Regel eine Vielzahl genormter Maschinen- und Formelemente (Schrauben, Freistiche etc.), die entweder aufgrund ihrer Normung herstellerunabhängig „vorgezeichnet“ und parameterisiert oder vom jeweiligen Lieferanten einzelne CAD-Systeme zur Verfügung gestellt werden können.

Das übergeordnete Ordnungssystem, der Zugriff auf die Parametertabellen sowie die Parameterübergabe zwischen CAD-Modell und Parametertabelle müssen jeweils systemspezifisch mittels der CAD-Programmierschnittstelle angebunden werden. Diese Normteilebibliotheken zeichnen sich daher in der Regel durch eine tiefe CAD-Integration aus.

Vorrangige Zielsetzungen dieser Bibliotheken sind somit Zeiteinsparungen, sowie die Verringerung von Fehlerquellen bei der Modellierung von Normteilen.

Diese Bibliotheken können unternehmensspezifisch angepasst werden. Die Parametertabellen können um zusätzliche Merkmale (z.B. Sachnummer) oder um zusätzliche Datensätze (z.B. Zwischengrößen von Schrauben) erweitert sowie neue Komplettteile (zusätzliche CAD-Parametermodelle und Parametertabellen für z.B. Werknormteile oder häufig gebrauchte Wiederholteile) hinzugefügt und das übergeordnete Ordnungssystem entsprechend erweitert bzw. angepasst werden.

Nachteilig an diesen heute verfügbaren Lösungen ist, dass zu keiner Zeit ein direkter Bezug zu den aktuellen und spezifischen Informationen der Zulieferer dieser Normteile (aktuell verfügbare Sonder- und Zwischengrößen, Werkstoffe, Bestell-Nr. etc.) sowie keine Verbindung der CAD-Normteilebibliothek zum PDM-System, wo Informationen für die weitere Prozesskette benötigt werden (Sach-, Bestellnummer, Herstellerangaben) besteht. Darüber hinaus sind nur häufig verwendete



genormte Standardteile verfügbar, da keine Integrationsmöglichkeiten für herstellerspezifische (oftmals nicht genormten) Teilebibliotheken besteht.

Die Nachteile dieser Entwicklung verdeutlichen, dass systemunabhängige Konzepte erforderlich sind, um die mit einer Norm- und Zukaufteilverwendung einhergehenden Potentiale seitens der Produktentwicklung, den verschiedenen Lieferanten sowie der CAx-Systemanbieter nutzen zu können.

Teilebibliotheken sind aus diesem Grund seit Mitte der 80er Bestandteil der Normung. Zielsetzung ist eine zielsystemneutrale Repräsentation von Norm- und Zukaufteilbibliotheken, die den Austausch von ganzen Bibliotheken ermöglicht. Notwendige Voraussetzung ist die Standardisierung geeigneter Beschreibungsverfahren welche eine effiziente Handhabung von Teilefamilien erlaubt.

Im folgenden soll auf die Grundlegenden Ansätze dieser Entwicklung sowie den wichtigsten daraus hervorgegangenen Normen (DIN V4001 und DIN V66304 - bekanntgeworden unter der Bezeichnung VADPS) eingegangen werden. Die wichtigsten Ergebnisse zur Umsetzung des Gesamtkonzeptes werden in den Normen ENV 40004 auf der europäischen und ISO 13584 auf der internationalen Ebene veröffentlicht. Auf die internationalen Normungsbestrebungen im Rahmen der ISO 13584 soll wegen deren zukünftig grossen Bedeutung detailliert eingegangen werden.

## 6.2 Beschreibungsverfahren

Neben dem Austausch von produktdefinierenden Daten in Form von Datenstrukturen, die die Ausprägung von Produkten repräsentieren, besteht im Zusammenhang mit dem Austausch von Teilebibliotheken die Anforderung, produktdefinierende Daten in Form von Erzeugungslogiken auszutauschen. Diese Erzeugungslogiken zielen auf die Repräsentation von Bauteilvarianten. Sie werden in eine CAD-systemunabhängige Vorschrift zur Verarbeitung der Erzeugungslogik übertragen. Mit der Angabe der konkreten Parameterwerten und der Abarbeitung der Erzeugungslogik wird die gewünschte Variante erzeugt.

Für die Definition von Erzeugungslogiken für Bauteilvarianten muss zwischen folgenden Variantenklassen unterschieden werden:

- Abmessungsvarianten sind dadurch charakterisiert, dass sie nur Abmessungswerte variabel sind. Dies bedeutet, dass die Bauteilausprägung lediglich von den gegebenen Ausprägungen variabler Nennmasse abhängig ist. Änderungen der Bauteilgestalt resultieren nur aufgrund bestimmter Abmessungswerte (z.B.  $L=0$ )
- Gestaltvarianten erlauben eine Variation der Bauteilgestalt. Diese Gestaltänderung resultiert entweder aus einer Kombination kompatibler Geometriekomplexe (Formelemente) oder aus der Abarbeitung vorgegebener Gestaltungsregeln die in einer Gestaltungslogik abgebildet werden muss. Die Bauteilausprägung ergibt sich dann aus den einzelnen Geometrieausprägungen sowie der Topologieausprägung.
- Abmessungs- und Gestaltvarianten stellen die Kombination der beschriebenen Varianten dar. In der technischen Anwendung sind Abmessungs- und Gestaltvarianten meist Bauteile, deren Abmessungen nach vorbestimmten Abmessungswerten variierbar sind und deren Gestaltvariabilität auf der Kombination unterschiedlicher aber kompatibler Geometriekomplexe beruht.

Hinsichtlich einer zielsystemneutralen geometrischen Repräsentation dieser Abmessungs- und Gestaltsvarianten stehen prinzipiell zwei Verfahren der Geometriebeschreibung zur Verfügung:

- deskriptive Geometriebeschreibung:  
Die Beschreibung und Speicherung der Modelldaten und Relationen zueinander erfolgt durch eine definierte Topologie als Datenstrukturmodell, womit ein direkter Zugriff zur Laufzeit auf die Modelldaten möglich ist. Bei der Speicherung von Einzelteilen in massvarianten Datenstrukturmodellen werden die Daten

strukturiert mit definierter Topologie und Geometrie abgelegt. Im Falle von Teilebibliotheken resultieren daraus die oben bereits erwähnten anbieterseitig grossen Datenmengen. Massinvariante Datenstrukturmodelle eignen sich vielmehr zur Speicherung einzelner komplexer Zukaufkomponenten.

Massvariante Datenstrukturmodelle sind gegenüber den Massinvarianten um die Fähigkeit der Parameterisierung erweitert und können damit auch für die Beschreibung von ganzen Teilefamilien verwendet werden. Allerdings existieren derzeit noch keine genormten Schnittstellen zum Austausch solcher massvarianter Modelle.

- prozedurale Geometriebeschreibung:  
Der Aufbau der jeweiligen Datenstrukturmodelle erfolgt durch eine Beschreibung und Speicherung der Erzeugungslogik z.B. in Form von Befehlsfolgen. Die Modelldaten können damit nur nach Ablauf der Befehlsfolgen (also nicht zur Laufzeit) zur Verfügung gestellt werden.  
Variantenprogramme können diesem prozeduralen Ansatz zugeordnet werden. Die Geometrieerzeugungsbefehle der Variantenprogramme enthalten Werte, die durch Variablen repräsentiert werden, welche unabhängig von den Erzeugungslogiken gespeichert werden. Variantenprogramme setzen allerdings den Zugriff auf Dienstprogramme wie Compiler oder Interpreter voraus.

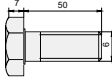
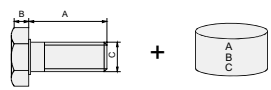
		Geometriebeschreibung	
		deskriptiv	prozedural
Datenstruktur	massinvariant	massinvariante Datenstrukturmodelle  DXF IGES VDAFS ISO 10303 (STEP)	Kommandomakros BEGIN INSERT LINE (0,0,0,0,10,10) INSERT CIRCLE (0,0,0,0,25) END
	massvariant	massvariante (parametrische) Datenstrukturmodelle  (IGES)	Variantenprogramme SUBROUTINE DIN912(A,B,C) ... P1=PNTCAB(X,Y,Z,KFIX) L20LIN2PT(P1,P2,KFIX) END VDA-PS ISO 13584 (PartsLib)

Bild (B013srpZ) Geometriebeschreibungsverfahren in Anlehnung an [GAP-93]

In Bild B013srpZ sind die derzeit meist gebräuchlichen Schnittstellen sowie der jeweils zugrunde liegende Beschreibungsmethode zugeordnet.

### 6.3 Neutrale Parameterrepräsentation auf Basis der DIN V 4001

Aufbauend auf der Sachmerkmaldefinition nach DIN 4000 (vgl. Kapitel 6) wurden für die Nutzung von Sachmerkmalen zur Beschreibung von Norm- bzw. Zukaufteile CAD-spezifische Ergänzungen vorgenommen, die zur Definition sogenannter CAD-gerechter Sachmerkmale führte welche in der DIN V 4001 beschrieben sind.

Auf semantischer Ebene sind innerhalb der DIN 4001 vier Kategorien von Sachmerkmalen vorgesehen:

- Sachmerkmalschlüssel beginnend mit A dienen der Spezifikation von Mass- oder Produktnormen. Sie sind in den entsprechenden Spezifikationen bzw. Normen definiert.
- Sachmerkmalschlüssel beginnend mit B sind Ergänzungsmerkmale, die nicht in Mass- und Produktnormen vorkommen, aber zur Geometriedefinition von Normteilen erforderlich sind
- Sachmerkmalschlüssel beginnend mit C kennzeichnen Funktionsmerkmale, deren Werte erst zum Einbaupunkt des Normteils in die Datenstruktur des CAD-Systems festgelegt werden können.
- Sachmerkmalschlüssel beginnend mit D beschreiben Merkmale, die mit Algorithmen aus den beiden vorangegangenen Kategorien abgeleitet werden können.

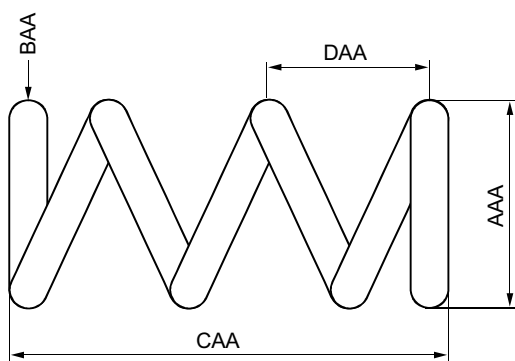


Bild (B014srpZ) Sachmerkmale einer Feder

Das Sachmerkmal AAA (Aussendurchmesser der Feder) in Bild B014srpZ ist beispielsweise innerhalb der entsprechenden Norm bzw. Werksnorm definiert und wird dem Anwender als Auswahlkriterium

angeboten. Dagegen ist der Radius des Federkörpers nicht Bestandteil der Normierung. Für die Erzeugung der geometrischen Repräsentation ist dieser jedoch notwendig. In der Regel sind diese Sachmerkmale für den Benutzer nicht sichtbar. Soll die ausgewählte Feder in einer Konstruktion Verwendung finden muss diese in die bereits vorhandenen Datenstrukturen des CAD-Modells integriert werden. Entsprechend der jeweiligen Einbausituation, muss ein die geometrische Repräsentation eine bestimmte Länge aufweisen. Die Länge der Feder wird aus diesem Grund als Kontextparameter CAA definiert. Zur Vollständigen geometrischen Definition der Feder wird ausserdem noch die Ganghöhe benötigt. Die ist direkt von dem Kontextparameter CAA abhängig, d.h. die Ganghöhe kann auf Basis einer berechnungsvorschrift und dem gewählten Kontextparameter errechnet werden.

Für die Geometriebeschreibung sind fünf Kategorien definiert, die die unterschiedliche CAD-bezogenen Darstellungsformen von Normteilen zu berücksichtigen:

- Kategorie 1, Geometrie-Klasse:  
Die Geometrie-Klasse stellt verschiedene Geometriebausteine (A-Bausteine und B-Bausteine), Komplettteile K bestehend aus Geometriebausteinen sowie Baugruppen G bestehend aus Komplettteilen zur Verfügung. Auf diese Weisen können Gestaltvarianten definiert sowie häufig verwendete Geometrielemente wie z.B. Gewinde, Einstiche etc. in verschiedenen Bauteilen verwendet werden.
- Kategorie 2, Darstellungsgrad:  
Je nach Anforderung der Anwendung lassen sich drei unterschiedliche Darstellungsgrade unterscheiden:
  - M: symbolhaft (Mittellinien, etc.)
  - S: Standard (vereinfachte Darstellung)
  - E: erweitert (detaillierte Darstellung)Der Anwender dadurch kann beim Einsetzen eines Normteils entscheiden, welchen Detaillierungsgrad das in das Produktmodell zu integrierende Normteil aufweisen soll.
- Kategorie 3, Ansicht:  
Entsprechend der DIN 5 Teil 19 werden 7 Ansichten definiert, wobei die erste Ansicht mit der Nummer 1 eine isometrische Darstellung der Normteilegeometrie beinhaltet
- Kategorie 4, Ansichtsvariante:  
Je nach Anwendung werden unterschiedliche Ausprägungen bestimmter Ansichten benötigt (z.B. Schnittdarstellungen), welche datentechnisch als verschiedene Anwendungsvarianten definiert werden.

- Kategorie 5, Einbauzustandsvariante:  
Bei bestimmten Normteilen (z.B. Federn, Pneumatikzylinder, etc.) ergeben sich ein Teil der Normteilabmessungen erst aus deren Einbauumgebung. Somit ergeben sich verschiedene Einbauzustandsvarianten wie z.B:
  - Bausteine deren Geometrie durch Algorithmen variiert werden kann (Schraubenfeder) Bausteine mit 2 oder mehreren Zuständen (Niet roh/gesetzt)
  - Kombinationen der beiden Fälle

Auf syntaktischer Ebene ist innerhalb der DIN 4001 ein strukturiertes Dateiformat zum Austausch von Sachmerkmaleisten definiert:

- Es handelt sich dabei um eine vom Menschen lesbare sequentielle ASCII-Datei, die 80 Zeichen pro Zeile vorsieht.
- Mittels einer Kennung zu Beginn jeder Zeile werden die einzelnen Informationsinhalte gekennzeichnet.
- Eine Datei kann in einzelne logische Blöcke gegliedert werden (BD markiert den Beginn, ED das Ende eines logischen Blockes)
- Ein logischer Block beinhaltet jeweils:
  - die identifizierenden Angaben (Bezeichnung, Version der Norm, Versionen der zugeordneten Programme, etc.) hinsichtlich der verwendeten Norm
  - die formale Parameterdefinition, die Zuordnung der Komplexe sowie die zugehörigen Merkmale
  - die Merkmalbeschreibungssätze, sowie
  - die Merkmaldatensätze.
- Die Erweiterung um weitere Bestandteile wie zum Beispiel unternehmensspezifische Angaben (Werksnorm, Ordnungsnummer, Gültigkeit, etc.) oder Angaben hinsichtlich Tabellenstrukturen.

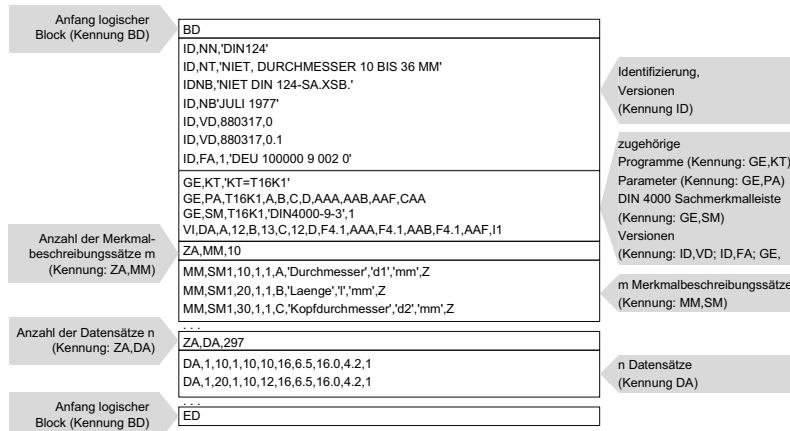


Bild (B015srpZ) Aufbau einer DIN 4001 Sachmerkmalteile



## 6.4 VDA-Programmschnittstelle nach DIN V 66304 VDAPS

Die Programmschnittstelle zur CAD neutralen Repräsentation von Erzeugungslogiken für Norm- und Zukaufteile wurde von einer Arbeitsgruppe des Verbandes der Automobilindustrie (VDA) entwickelt. Sie ist deshalb auch unter der Bezeichnung VAD-PS bekannt. Geometrische Ausprägungen von Normteilen werden in CAD-Systemen über Variantenprogramme generiert. Die Variantenlogik wird dabei in Form eines Programmes formuliert.

Diese Variantenprogramme werden in der Programmiersprache FORTRAN 77 erstellt und sind unabhängig von CAD-Systemen. Zur Anwendung der Variantenprogramme in CAD-Systeme ist die Einbindung erforderlich, wofür eine Programmschnittstelle in FORTRAN erforderlich ist, die den Zugriff auf die CAD interne Methodenbank erlaubt.

Der Austausch von Normteilbeschreibungen zwischen unterschiedlichen CAD-Systemen wird dabei durch ergänzende Funktionen, eingebettet in FORTRAN 77 geleistet. Diese Funktionen umfassen:

- Elemente zur Standard Geometrie Definition
- Elemente zur Standard Geometrie Manipulation
- Elemente zur Standard Geometrie Darstellung (Linienarten, Farben, Strichstärken)
- Elemente zur Erzeugung von Geometriestrukturen, Hilfsfunktionen und Funktionen zur räumlichen Orientierung

CAD-Systeme welche VDAPS unterstützen müssen somit entweder über eine vorhandene Programmierschnittstelle oder direkt die innerhalb der VDAPS definierten Methodenaufrufe bereitstellen sowie die entsprechenden Methoden der CAD-Systeminternen Methodenbank zur Erzeugung der benötigten Geometrieelementen diesen Aufrufen zuordnen. Soll eine Teilebibliothek in ein CAD-System integriert werden, muss diese zunächst kompiliert und dann der kompilierte Code beim Aufstarten des CAD-Systems diesem hinzu gelinkt werden. Desweiteren muss eine Auswahlmaske zur Auswahl der jeweils benötigten Normteile von CAD-System bereitgestellt werden.

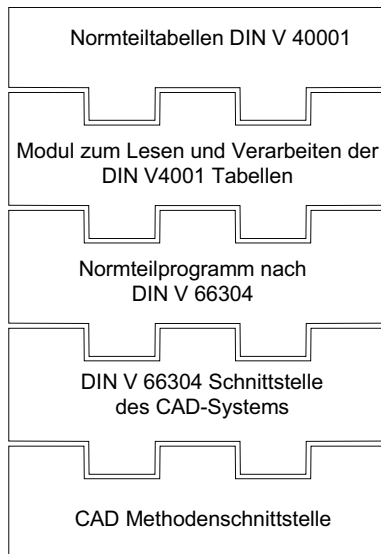


Bild (B016srpZ) Integration der DIN V 4001 und VDA-PS in ein CAD-System

## 7 Die ISO 13584 „Parts Library“

Bei den oben beschriebenen Methoden zur standardisierten Beschreibung von Norm- und Zukaufteilen handelt es sich um nationale Aktivitäten (Deutschland), welche durch die Möglichkeiten der Informationstechnologien zu Beginn der 80er geprägt sind (FOTRAN etc.).

Die Ergebnisse der bis zu diesem Zeitpunkt geleisteten Arbeiten wurden zunächst in die europäischen Normungsgremien mit dem Ziel eingebracht, durch eine auf ganz Europa ausgedehnte Gültigkeit dieser mehr Gewicht zu verleihen und sie langfristig zu sichern.

Um eine langfristige Konformität mit der immer mehr an Bedeutung gewinnenden ISO 10303 zu gewährleisten wurden schliesslich die Aktivitäten im Bereich Repräsentation und Austausch von Teilebibliotheken unter der Bezeichnung ISO 13584 „Parts Library“ (PLib) in die internationale Standardisierungsentwicklung integriert.

Die ISO 13584 wird komplementär zur ISO 10303 (STEP) innerhalb des gleichen Unterkomitees (SC4) der Internationalen Standardisierungsorganisation (ISO) entwickelt und genormt.

Im Mittelpunkt steht die Repräsentation von Teilebibliotheken sowie die Bereitstellung von Mechanismen und Definitionen zum Austausch, der Benutzung und der Aktualisierung einzelner Bibliotheksinhalte. Der Austausch von Bibliotheksinhalten soll dabei unabhängig vom jeweils verwendeten Zielsystem (z.B. CAD-Systemen) sowie Software- und Hardwareumgebung (z.B. Betriebssystem, Prozessorarchitektur, Netzwerk, etc.) innerhalb des gesamten Produktlebenszyklus, also von der Konstruktion, der Fertigung, Montage, Wartung und Beseitigung ermöglicht werden. Das innerhalb der ISO-13584 entwickelte und genormte Gesamtkonzept geht dabei hinsichtlich der Handhabung und Repräsentation von Norm- und Zukaufteilen weit über die oben beschriebenen Ansätze hinaus.

Zielsetzung der ISO 13584 ist im einzelnen:

- die Repräsentation von Teilebibliotheken bzw. der darin abgelegten Informationen,
- die spezielle Handhabung von Einzelteilen sowie Baugruppen innerhalb einer Teilebibliothek
- der Austausch, Speicherung, Transfer, Zugriff sowie Aktualisierung von Teilebibliotheken bzw. der darin abgelegten Informationen (z.B. Produktrepräsentationen, Produktpräsentationen, Produktdefinitionen, etc.),

- die Integration einer beliebigen Anzahl an Teilebibliotheken unterschiedlicher Lieferanten (Produktbibliotheken) in eine integrierte Teilebibliothek,
- die Beschreibung von Teilen bzw. Teilefamilien durch den jeweiligen Lieferanten ohne ein externes Dictionary oder andere Teilebibliotheken zu referenzieren.
- die Definitionen von Mechanismen welche es einer Teilebibliothek erlauben, ein standardisiertes Dictionary zu referenzieren.

Das Konzept geht davon aus, dass ganze Teilekataloge zwischen Lieferant und Integrator bzw. Produktentwickler auszutauschen sind. Diese Teilekataloge besitzen eine vom Lieferanten gewählte Struktur, die den Zugriff auf benötigte Bibliothekselemente unterstützen. Ausgehend von dem ausgewählten Bibliothekselement kann auf die mit diesem verknüpften Produktdaten wie zum Beispiel der jeweiligen geometrischen Repräsentation, der Einbauvorschrift, etc. zugegriffen werden. Die zugeordneten Produktdaten können dabei wiederum zu Klassen zusammengefasst und entsprechend strukturiert werden.

Aus diesem Grund werden zwei grundlegende Modelle definiert, das sogenannte „General Model“ zur Strukturierung und Beschreibung einer Teilebibliothek bzw. deren Inhalte sowie dem „Functional Model“ zur Handhabung der jeweils zugeordneten Produktdaten (vgl. Bild B016srpZ).

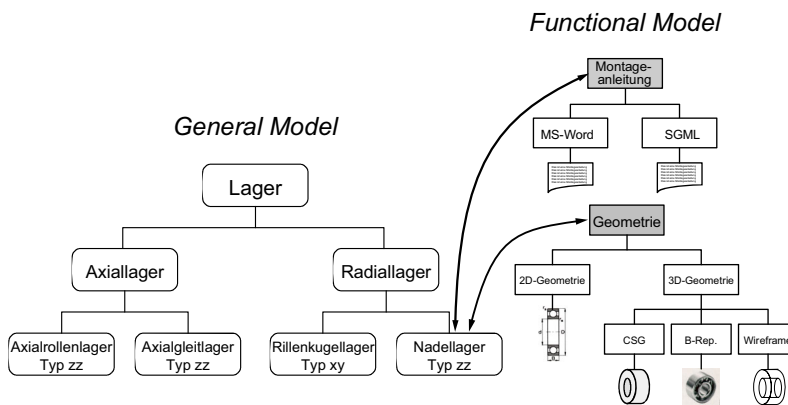


Bild (B017srpZ) Beschreibung des Bibliothekselementes im General Model und Referenzierung zugehöriger Repräsentation im Functional Model

Das *General Modell* enthält damit alle identifizierenden Merkmale der einzelnen Bibliothekselemente und besteht im wesentlichen aus:

- einem implementierungsunabhängigen Metadatenmodell zur Definition einer standardisierten Strukturierung von Teilegruppen,
- Regeln hinsichtlich der Instanzierung dieses Metadatenmodells sowie
- einer allgemeinen Referenzhierarchie, in welche die einzelnen Teilelieferanten ihre Bibliotheken einhängen können bzw. welche durch die verschiedenen Teilelieferanten entsprechend ihres Teilekataloges spezialisiert werden kann.

Der Lieferant des Kataloges kann das Erscheinungsbild seines Teilekataloges weitgehend selbst bestimmen. Die Strukturierung des General Model erfolgt somit produktbezogen entsprechend den Vorgaben des Teilelieferanten.

Ziel der Referenzhierarchie ist es, eine Auswahl von Teilen unter gleichartigen Angeboten unterschiedlicher Lieferanten zu ermöglichen. Dazu muss die Referenzhierarchie in der Lage sein, eine z.B. für Normteile erarbeitete Referenzstruktur aufzunehmen. Diese Referenzstruktur enthält die genormten Teilefamilien mit ihren Merkmalen strukturiert nach genormten Richtlinien. Jeder hier vereinbarten Teilefamilie können Dokumente zugeordnet werden, die deren Bedeutung und ihren Einsatz beschreiben. Dies werden in der Regel Produktnormen sein.

Innerhalb dieser Referenzhierarchie soll eine funktionsorientierte Strukturierung abgebildet werden wie sie in der Entwurfsphase für die Suche einer geeigneten Lösung benötigt wird.

Der Lieferant einer Teilebibliothek kann im Rahmen der Beschreibung seiner Teileklassen einen Bezug zur Referenzstruktur herstellen. Er erklärt damit, dass seine Teileklassen der referenzierten Klasse gleichartig ist, und die in der referenzierten Teileklasse definierten Merkmale ebenfalls für seine Klasse gelten. Dem Produktentwickler wird es damit ermöglicht eine Auswahl unter mehreren Lieferanten zu treffen. Ausserdem kann jeder Lieferant zusätzlich zu den in der Referenzstruktur definierten Merkmalen eigene Merkmale einführen in dem er diese genormten Klassen spezialisiert.

Das *Functional Model* beinhaltet alle erforderlichen Merkmale zur Erzeugung bzw. Handhabung benötigter Produktdaten. Im einzelnen handelt es sich dabei um

- Produktrepräsentationen (CAD-Modelle basierend auf einem parameterisierten Datenmodell),

- verschiedene Detaillierungen und Ableitungen der Produktpräsentation (vereinfachte 3D-Modelle, detaillierte 2D Zeichnungen, FEM-Modelle, etc.),
- Produktmetadaten (Werkstoff, technische Dokumentationen, etc.) sowie
- Produktdefinierende Daten (Lieferant, Bestellnummer, etc.)

Entsprechend der ISO 10303 (STEP) setzt sich die gesamte ISO 13584 aus einer Anzahl von Serien (series) bestehend aus einzelnen Normdokumenten (Parts) zusammen. Dabei spiegelt die Entwicklung und Normung der ISO-13584 die prinzipielle Vorgehensweise beim Datenbankentwurf wider. Zunächst wird ein Ausschnitt der realen Welt welcher von einem Datenbanksystem repräsentiert werden soll definiert und möglichst exakt beschrieben. Dies geschieht in Part 1 und Part 10 der ISO 13584 durch einzelne Referenzmodelle. Innerhalb der folgenden Serien bzw. Normdokumenten werden diese auf implementierungsunabhängige Datenmodelle abgebildet welche durch die verschiedenen Systemanbieter implementiert werden können.

## 7.1 Aufbau und Entwicklung der ISO 13584

Insgesamt gliedert sich die ISO 13584 in acht Themengebiete, die als Serien bezeichnet werden. Jede Serie besteht aus einzelnen Teilnormen, bzw. Dokumenten die als Parts bezeichnet werden.

Die *0-Serie* beinhaltet lediglich den Part 1 *Overview and Fundamental Principles*. In diesem Dokument sind die Grundlagen bzw. fundamentalen Prinzipien festgehalten. Es enthält allgemeine Erklärungen, Terminologiedefinitionen sowie Entwurfsprinzipien und -vorgehen für die Definition von Datenmodellen.

Die *10er-Serie Conceptual Descriptions* enthält Normdokumente innerhalb welcher die zugrundeliegenden Konzepte beschrieben und definiert sind. Die 10er-Serie enthält lediglich Part 10 *Conceptual Model of Parts Library*. Part 10 stellt das Ausgangsdokument der ISO 13584 dar. Bestandteil ist zum einem eine Problemanalyse hinsichtlich der Representation und dem Austausch von Teilebibliotheken und die Anforderungen an den zu entwickelnden Standard. Desweiteren werden die grundsätzlichen Konzepte, die Terminologie sowie die benötigten Mechanismen definiert.

Die *20er Serie Logical Resources* beinhaltet jene Dokumente, innerhalb welcher die logischen Beschreibungen definiert sind. Dazu gehört insbesondere Part 24 *Logical Model of Supplier Library*, eines der Schlüsseldokumente von PLib. Part 24 definiert die generischen Ressourcen für die Repräsentation und dem Austausch von Teilebibliotheken.

Desweiteren umfasst die 20er-Serie Part 20 *General Resources und Part 26 Identification of Supplier Libraries*.

Die *30er Serie Implementation Resources* umfasst alle Normen zur Spezifikation der jeweiligen Implementierung. Dazu gehört derzeit lediglich Part 31 *Programming Interface*. Part 31 stellt die Beschreibung einer zur DIN V 66304 (VDA PS) kompatiblen Schnittstelle zur parametrischen Geometriegenerierung bereit.

Die *40er Serie Description Methodology* enthält Normen hinsichtlich der Beschreibungsmethodik von Teilebibliotheken. Innerhalb von Part 42 *Methodology for Structuring Part Families* wird die prinzipielle Methodologie zur Strukturierung von Teilebibliotheken in Form von 8 Regeln festgeschrieben.

Das Konzept unterscheidet prinzipiell zwischen dem sogenannten „General Model“ zur Strukturierung der Bibliotheksdaten, sowie dem „Functional Model“ zur Handhabung der verschiedenen Repräsentationsformen eines Bauteils.

Die *50er Serie Conformance Testing* umfasst Normen innerhalb welchen Methoden zur Überprüfung der ISO 13584 Konformität definiert werden. Derzeit existiert noch kein Part.

Die *100er Serie View Exchangeprotocol* beinhaltet Normen die den Austausch von Representationen definieren. Part 101 definiert dabei den Repräsentationsaustausch basierend auf der in Part 32 definierten parametrischen Geometriebeschreibung. Part 102 definiert den Repräsentationsaustausch basierend auf dem STEP AP 214.

Die *2000er Serie Standardised Content* umfasst die Definitionen geneormter Bibliotheksinhalte



## 7.2 Konzeptuelles Modell einer Teilebibliothek gemäss ISO 13584

Innerhalb der ISO 13584 wird zunächst ein implementierungs- und modellierungsunabhängiges Konzept als Basis für die Repräsentation und den Austausch von Teilebibliotheken definiert. Dieses Konzept besteht im wesentlichen aus vier einzelnen Referenzmodellen:

- Innerhalb des Actor Reference Model werden die verschiedenen Anwendungsszenarien von Teilebibliotheken definiert.
- Das Architecture Reference Model bestimmt die einzelnen Module einer Teilebibliothek.
- Innerhalb des Information Structure Reference Model werden die Basiskonstrukte zur Informationsstrukturierung festgelegt.
- Mittels dem Information Semantic Reference Model wird ein zweischichtiges Referenzmodell zur Abbildung der jeweiligen Semantik auf die innerhalb des Structure Reference Model definierten Basiskonstrukte definiert.

Im folgenden soll zunächst auf die einzelnen Referenzmodelle eingegangen werden. Von grosser Bedeutung sind dabei das Information Semantic Reference Model und das Structure Reference Model.

### 7.2.1 Das Actor Reference Model

Prinzipiell wird hinsichtlich der Benutzung einer Teilbibliothek zwischen drei Tätigkeitsklassen, Rollen genannt unterschieden, welche in der Regel von bestimmten Personengruppen (Teilelieferant, Integrator, Endanwender) besetzt werden. Die gleiche natürliche Person kann dabei in bestimmten Fällen aufeinanderfolgend verschiedene Rollen annehmen.

- Der *Teilelieferant* (Supplier) beschreibt in einer standardisierten Art und Weise eine Anzahl an Teile bzw. deren jeweilige Repräsentation. Er ist für die Korrektheit dieser Beschreibung verantwortlich. Die Beschreibung dieser Teile in einer Teilebibliothek wird als Teilekatalog (*Supplier Library*) bezeichnet.
- Der *Integrator* (Integrator) integriert eine Anzahl an Teilekataloge in ein Softwaresystem auf welches von verschiedenen Applikationen (z.B. verschiedene CAD- oder PDM-Systeme) zugegriffen werden kann. Es handelt sich damit um eine integrierte Teilebibliothek (Integrated Library).

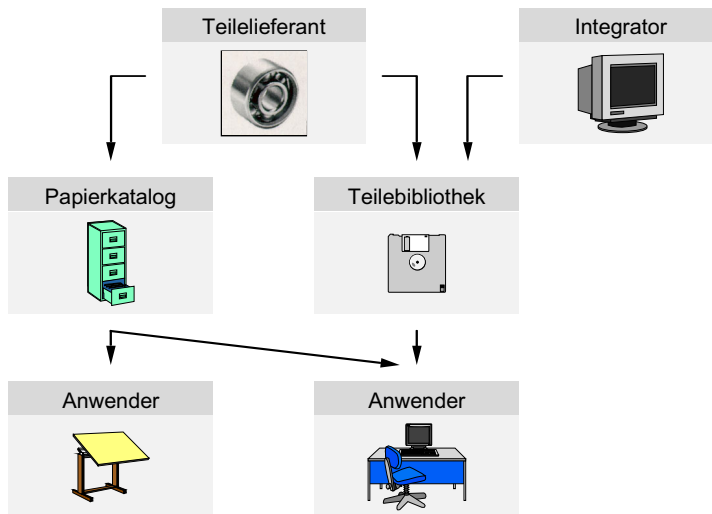


Bild (B018srpZ) Rollen im Zusammenhang mit der Erstellung und Nutzung integrierter Teilebibliotheken

- Der *Endanwender* (End User) greift via seiner CAx-Applikation auf diese integrierte Teilebibliothek zu. Dabei werden im einzelnen folgende Aktivitäten unterschieden:
  - Zugriff auf Informationen der Bibliothek,
  - Auswahl eines vorhandenen Bauteils,
  - Übernahme der ausgewählten Repräsentation des gewählten Teiles in das jeweilige CAx-System,
  - Aktualisierung bzw. Erweiterung einzelner Produktbibliotheken (z.B. Löschen und Hinzufügen von Attributen). Das Ergebnis einer solchen Änderung wird als Benutzerbibliothek (User Library) bezeichnet.

Der Integrator muss somit ein System zur Verfügung stellen, welches

- die automatische Integration von Produktbibliotheken ermöglicht,
- benutzerspezifische Anpassungen dieser Produktbibliotheken zulässt, sowie
- von verschiedene CAx-Applikationen aufgerufen werden kann.

### 7.3 Das Architektur Referenzmodell

Entsprechend der Referenzarchitektur besteht eine integrierte Teilebibliothek aus fünf Subsystemen:

- Die Dialogschnittstelle (Dialoginterface) ermöglicht den Zugriff auf die Integrierte Bibliothek bzw. den graphischen-interaktiven Dialog, sowie die Übernahme und Darstellung der abgefragten Bibliotheksdaten auf den Bildschirm.
- Die Datenschnittstelle (Representation Transmission Interface) dient der Übertragung und Integration der ausgewählten Repräsentation eines Bauteils in die Datenstrukturen der jeweiligen CAx-Applikation.
- Das Dictionary (Semantic Dictionary) stellt die automatische Zuordnung zwischen den externen dem Benutzer auf dem Bildschirm angezeigten Bezeichnungen, und den entsprechenden bibliotheksinternen Bezeichnungen sicher.

Zielsetzung ist dabei:

- die Unterstützung einer Mehrsprachigkeit
- die Bereitstellung verschiedener Synonymen eines Begriffs
- die Sicherstellung der Konsistenz hinsichtlich der Namensgebung für gleiche Elemente verschiedener Lieferanten
- die Vergabe von vordefinierter Bezeichnungen für bestimmte Einträge
- Der Bibliotheksinhalt (Library Content) wird durch die eigentlichen Informationen gebildet, welche in der Bibliothek abgelegt sind. Es kann sich dabei sowohl um Daten als auch ausführbare Programme handeln.
- Das Bibliotheksprogrammsystem (Library Management System) ist jene Software, die es dem Benutzer erlaubt den Inhalt einer Bibliothek zu benutzen.

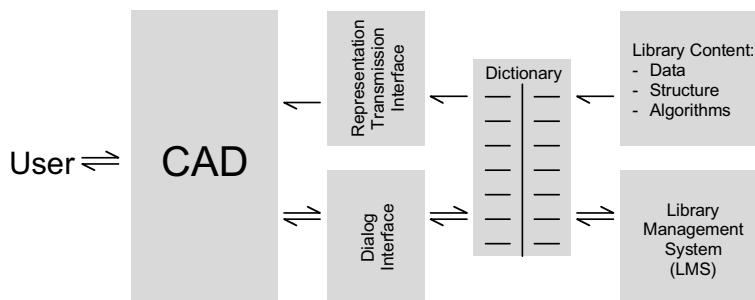


Bild (B019srpZ) Referenzarchitektur einer Teilebibliothek (CAD-Integration)

## 7.4 Das Strukturreferenzmodell

Innerhalb des Strukturreferenzmodells werden sowohl die wesentlichen Informationsinhalte bzw. Informationskonstrukte als auch deren konzeptuelle Strukturierung definiert.

Innerhalb der ISO 13584 spielen diese eine zentrale Rolle. Sie sind vergleichbar mit den generischen Ressourcen der ISO 10303. Auf die für das Verständnis wichtigsten soll im folgenden detailliert eingegangen werden.

### Das Attributekonzept (Attributes)

Bei einem Attribut handelt es sich zunächst um einen abstrakten Ausdruck basierend auf welchen Eigenschaften beliebiger Klassen bzw. Objekte charakterisiert und beschrieben werden können. Ein Attribut innerhalb der PLib entspricht damit einem Merkmal (vgl. Kapitel 6). Attribute können kategorisiert und entsprechend definiert werden. Es handelt sich dabei um entsprechende Spezialisierungen. Innerhalb der ISO 13584 wird jedes Attribut innerhalb einer Klasse definiert (z.B. einer General Model class).

Ein Attribut besitzt

- einen eindeutigen Kode, welcher von dem jeweiligen Lieferanten einer Klasse vergeben wird,
- eine Version, sowie
- eine Beschreibung.

Hinsichtlich der jeweiligen Verwendung unterscheidet man zwischen:

- Free Attributes:  
Attribute welche unabhängig von weiteren Attributen angewendet werden können.
- Identification Attributes:  
Attribute mittels welchen eine Komponente identifiziert werden kann. Es muss sich dabei um ein Free Attribute handeln.
- Derived Attributes:  
Attribute welche mittels einer Ableitungsfunktion aus einem bzw. einer Anzahl weiterer Attribute errechnet werden können.
- Generic Attributes:  
Attribute welche einer generischen Familie zugeordnet sind.
- Inherited Attributes:  
Attribute welche einer Familie von der übergeordneten vererbt wurden.
- Representation Attributes  
Attribute welche entsprechend den B-Merkmalen aus DIN 4001 zur Generierung einer Repräsentation benötigt werden.

- View Attributes  
Mittels den View Attributes können einzelne views einer Klasse unterschieden werden (z.B. vereinfachte Darstellung, exakte Darstellung etc.).

### Das Teilekonzept (Parts)

Bei einem Part handelt es sich um die Abstraktion eines Gegenstandes zur Handhabung einer Menge materieller Gegenstände oder Funktionseinheiten welche hinsichtlich der gewählten Abstraktionsstufe gleich behandelt werden können. Es wird dabei zwischen *Supplier Parts* bzw. *Catalogue Parts* und *Abstract Parts* unterschieden.

- Abstract Parts werden durch eine abstrakte Beschreibung wie zum Beispiel einem internationalen Standard definiert und beschrieben. Es handelt sich dabei nicht um ein direkt existierendes Bauteil, sondern vielmehr um die Spezifikation dieses Bauteils z.B. um eine Schraube DIN 912 M5. *Supplier Parts* werden durch den jeweiligen Lieferanten definiert und mittels einer Spezifikation beschrieben. Es handelt sich dabei um ein Bauteil welches in einer unbestimmten Anzahl an identischen Kopien existiert (z.B. FESTO Zylinder DS 55).

Ein Part

- wird definiert durch die Klasse (Family) welcher es zugehörig ist (*is\_defined\_by*),
- wird identifiziert durch den Identifikator der Klasse sowie durch die Ausprägungen hinsichtlich der identifizierenden Attribute (*Identification Attributes*, *is\_identified\_by*),
- hat eine Menge von weiteren Attributen (*Derived Attributes*) abgeleitet von den identifizierenden Attributen (*has*) und
- ist verknüpft mit anderen Instanzen welche Methoden unterstützen (*is\_linked\_with*).

### Das Teilefamilienkonzept (Part Family)

Bei einer Teilefamilie handelt es sich um eine Anzahl ähnlicher Objekte, welche zu einer Gruppe zusammenfasst werden. Für diese Gruppe muss eine Anzahl an identifizierender Attribute (*Free Attributes* oder *Identification Attributes*) definiert werden. Einzelne Teile dieser Familie können entsprechend den jeweiligen Ausprägungen zu den identifizierenden Attributen voneinander unterschieden werden. Die Zuordnung und Definition von Familien und Merkmalen basiert auf dem Prinzip der gegenseitigen Definition: Eine Familie wird mit Merkmalen beschrieben; die Merkmale sind aber nur im Zusammenhang mit einer Familie sinnvoll zu verwenden.

Innerhalb der Teilefamilien unterscheidet man zwischen:

- einfachen Teilefamilien (Simple Families) und
- generischen Teilefamilien (Generic Families).

Bei einer *Simple Family* handelt es sich um eine Anzahl ähnlicher konkreter Teile, welche der Lieferant zu einer Gruppe zusammengefasst hat.

Der Inhalt einer Simple Family wird mittels einer Tabelle beschrieben. Jedes Teil einer Family wird je eine Zeile, jedem identifizierenden Attribut wird je eine Spalte zugewiesen.

Ein einzelnes Teil einer Simple Family wird demgemäss mittels der jeweiligen Family und den Ausprägungen zu den identifizierenden Attributen identifiziert.

Zusätzlich zu den identifizierenden Attributen können *Derived Attributes* definiert werden, welche sich aus den Identification Attributes ableiten bzw. errechnen lassen und die Auswahl des am besten geeigneten Teiles unterstützen. Hierfür muss die entsprechende Ableitungsfunktion *Derivation Function* definiert werden (vgl. Bild B019srpZ).

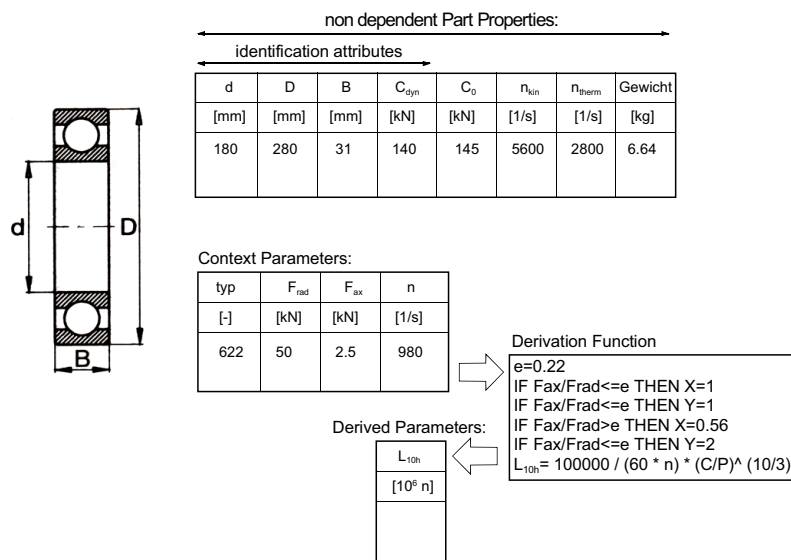


Bild (B020srpZ) Beispiel einer Simple Family

Da diese Tabellen unter Umständen extrem anwachsen können, ist es möglich diese aufzuteilen in

- eine Anzahl an Basistabellen,
- einer Anzahl von Funktionen oder Ausdrücken, mittels welcher weitere Attribute berechnet oder deren Wertebereich eingeschränkt werden kann und
- eine Anzahl an Verknüpfungsregeln sowie den definierten Funktionen und Ausdrücken, mittels welcher ausgehend von einer Anzahl an Basistabellen eine einzelne als virtuelle Tabelle bezeichnete Tabelle gebildet werden kann.

Entsprechend den jeweiligen Auswahlanforderungen wird zur Laufzeit diese virtuelle Tabelle ermittelt und dem Benutzer zur Auswahl des benötigten Bauteils angezeigt.

Schliesslich können mittels einer Ausschlussbedingung (Exclusion Constraint) einzelne Zeilen oder ganze Bereiche einer Tabelle für den Benutzer gesperrt werden.

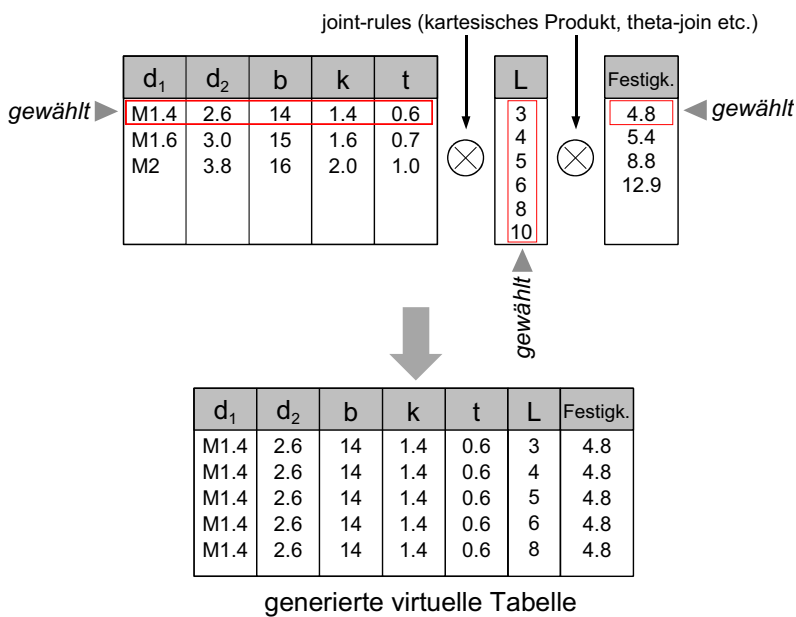


Bild (B021srpZ) Bildung einer virtuellen Tabelle

Simple Families können zu *Generic Families* gruppiert werden, welche wiederum zu weiteren *Generic Families* zusammengefasst werden können. Gemeinsame Eigenschaften einzelner Families (Generic oder Simple) können der übergeordneten *Generic Family* zugeordnet werden. Zielsetzung ist zum einem die Klassifizierung des Bibliotheks-

halts. Zum anderen können auf diese Weise abstrakte Teile definiert und innerhalb der Produktentwicklung verwendet werden.

Teilfamilien können auf verschiedenen Stufen des Ordnungssystems zu generischen Teilfamilien zusammengefasst werden. Man spricht aus diesem Grund von den jeweils übergeordneten (ascendent) Teilfamilien - also jenen Familien welcher eine Teilfamilie zugehörig ist und entsprechend den untergeordneten (descendent) Familien einer generischen Teilfamilie - also jenen Familien aus welchen eine generische Teilfamilie besteht.

Alle Attribute, Funktionen und Regeln werden entsprechend dem Paradigma der Objektorientierung an die darunterliegenden Teilfamilien vererbt.

Die Beschreibung einer Generic Family umfasst:

- den Namen der generischen Teilfamilie,
- die Angabe der zugehörigen untergeordneten Teilfamilien (generisch oder einfach) sowie
- eine Liste von Attributen welche an die untergeordneten Teilfamilien vererbt werden.

Hinsichtlich des Verhaltens der zu vererbenden Attribute wird zwischen *Applicable Attributes* und *Visible Attribute* unterschieden:

- Ist ein Attribut applicable definiert, handelt es sich um ein Attribut, welches für alle dieser generischen Teilfamilie untergeordneten Familien bzw. den diesen Familien zugehörigen Bauteilen definiert sein muss. Applicable Attributes werden von den darunterliegenden Teilfamilien referenziert. Alle Teile dieser Familien müssen eine Ausprägung zu den Applicable Attributes aufweisen. Die Ausprägung kann den NULL-Wert annehmen.
- Ist ein Attribut visible definiert, kann dieses Attribut innerhalb der untergeordneten Familien verwendet werden. Die Verwendung ist nicht zwingend. Die einzelnen Elemente der untergeordneten Familien müssen nicht unbedingt eine Ausprägung zu diesem Attribut aufweisen.



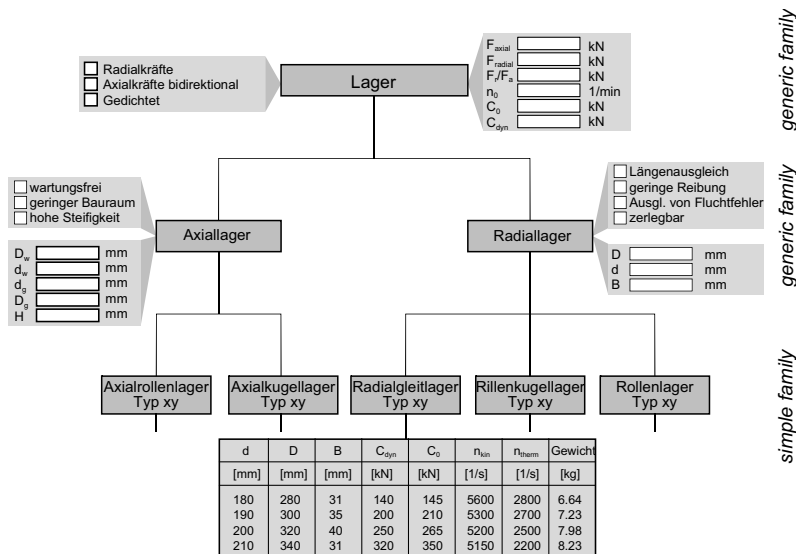


Bild (B022srpZ) Hierarchie bestehend aus Simple und Generic Families

Generische Teilefamilien können instanziiert werden. Es handelt sich in diesem Fall um ein *generisches Bauteil* (Generic Part). Der Anwender legt dabei lediglich fest, dass er beispielsweise ein Lager mit einem definierten Aussendurchmesser und Tragzahl in seine Konstruktion einsetzen will. Den genauen Typ, also die entsprechende Simple Family kann zu einem späteren Zeitpunkt beispielsweise durch den Einkauf festgelegt werden.

Als Voraussetzung für die Instanzierung einer generischen Teilefamilie werden wie bei einer einfachen Teilefamilie Attribute benötigt, entsprechend deren Ausprägung einzelne generische Bauteile unterschieden werden können. Dazu gehören:

- eine Liste von identifizierenden Attributen, mittels deren jeweiligen Ausprägung einzelne generische Bauteile unterschieden werden können (Free Attributes),
- Regeln zur Abbildung logischer Beziehungen zwischen den identifizierenden Attributen (Integrity Constraints) sowie
- Ableitungsfunktionen zur Berechnung von eventuellen Derived Attributes (Derivation Functions).

## Das Sichtenkonzept

Innerhalb des General Models bzw. den einzelnen Klassen des General Models werden die einzelnen Objekte der Bibliothek eindeutig identifiziert.

Mittels des Sichtenkonzeptes können einem Bauteil bzw. einer Teilfamilie des General Models verschiedene Repräsentationen korrespondierend mit verschiedenen Perspektiven des selben Objektes hinzugefügt werden. Diese sollen jeweils einzeln handhabbar und logisch mit dem zugehörigen Bauteil verknüpft sein. Bei diesen Sichten kann es sich um geometrische Repräsentationen, Produktbeschreibungen wie z.B. Montageanleitungen, Kennlinien oder um Preisinformationen handeln (vgl. Bild B022srpZ).

Die Beschreibung bzw. Definition dieser einzelnen Perspektiven erfolgt innerhalb des Functional Models bzw. den einzelnen Klassen des Functional Models. In den Klassen des Functional Models können weitere für die jeweilige View spezifische Merkmale definiert werden. Die zusätzliche Merkmale können ebenfalls als Auswahlmerkmale herangezogen werden.

Über eine Sichtenrelation wird eine Sicht mit einer (generischen) Familie verknüpft. Die Sicht steht dann sowohl für diese Familie wie auch für alle weiteren Unterfamilien zur Verfügung (falls die Sicht einer generischen Familie zugeordnet wurde) und kann nach belieben zugeschaltet werden. Die Wahl einer Sicht kann für die momentane Situation wichtige Merkmale hervorheben und unwesentliche ausblenden. Sichten dienen somit nicht der Strukturierung von Familienhierarchien, sie erweitern vielmehr die Informationsgrundlage für den Endanwender. Beispielsweise könnten innerhalb einer Sicht Merkmale hinsichtlich den Lagerkräften definiert sein und über eine Funktion die sich jeweils ergebende nominelle Lebensdauer als Entscheidungshilfe bei der Lagerauswahl errechnet werden.

Die Identifizierung des Bauteils bleibt dabei unverändert. Das Functional Model ist nur aussagekräftig in Verbindung mit dem zugehörigen General Model.

Sichten können einer Familie auf zweierlei Weise zugewiesen werden:

- Wird eine Sicht von vornherein im Hinblick auf ein Teil definiert, spricht man von einer „a priori“ Sicht. Dies ist dann zweckmässig, wenn die Sicht bzw. die zusätzlich definierten Merkmale nur im Zusammenhang mit einem bestimmten Bauteil bzw. eine Familie sinnvoll sind. Beispiel hierfür sind die Merkmale „Kosten“ oder „Lieferzeit“, welche von einem Anbieter für eine Familie ihres Sortimentes definiert werden.

- Demgegenüber wird bei der „*a posteriori*“ Sichtweise die Verknüpfung zwischen einer Sicht bzw. deren Merkmale mit einer Familie erst zu einem späteren Zeitpunkt von Kataloganbietern oder dem Endanwender vorgenommen. Die entsprechende Sicht wird dabei unabhängig von den etwaig zuzuordnenden Familien definiert. Als Beispiel hierfür sei die vereinfachte geometrische Repräsentation einer Schraube genannt. Diese kann beispielsweise durch dem Integrator definiert und nachträglich vom Anwender einer DIN genormten oder einer ISO genormten Schraubenfamilie zugeordnet werden.

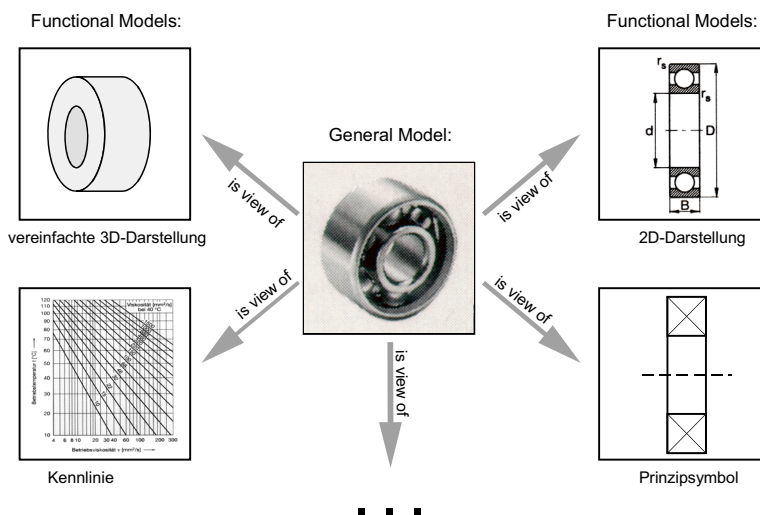


Bild (B023srpZ) Zuordnung verschiedener Repräsentationen zu einem Bauteil

### Definition von Repräsentationen und Repräsentationskategorien

Innerhalb der ISO 13584 werden wie bereits oben erwähnt die Objektbeschreibungen und die Objektrepräsentationen innerhalb paralleler Klassenhierarchien spezifiziert. Innerhalb der General Model classes werden die Objekte spezifizierenden Eigenschaften abgebildet. In denen des Functional Model classes werden dagegen jene Eigenschaften abgebildet, welche die einzelnen Repräsentationskategorien beschreiben bzw. spezifizieren.

Wird zum Beispiel eine Unterlagscheibe durch die drei Parameter Innendurchmesser, Aussendurchmesser und Dicke eindeutig identifiziert, werden diese drei Parameter innerhalb der entsprechenden

General Model class als Identification Attributes der Teilefamilie „Unterlagscheibe“ definiert. Mittels zugeordneten Definitionstabellen kann die Anzahl erlaubter Ausprägungen eingeschränkt werden. Entsprechend den durch den Anwender angegebenen Ausprägungen dieser Identification Attributes, wird eine einzelne Unterlagscheibe der Familie identifiziert (vgl. Bild B023srpZ).

Um jedoch das zugehörige CAD-Modell als eine mögliche Repräsentation der Unterlagscheibe erzeugen zu können, werden noch weitere Informationen benötigt. Als Beispiel hierfür seien die Fasen der Kanten oder die benötigte Ansichtsvariante, in welcher die Unterlagscheibe im CAD-System dargestellt werden soll genannt. Diese Informationen werden im Functional Model innerhalb der Functional Model classes definiert.

Die Beschreibung einer Functional Model class ist der einer General Model class prinzipiell ähnlich mit dem Unterschied, dass eine Functional Model class lediglich Context Parameters, Representation Attributes, Derivation Functions und Methoden zur Erzeugung von Sichten beinhalten kann. Die Methoden sind mit View Logical Names und möglicherweise mit View Control variables assoziiert, welche es dem Anwender erlauben auf diese zuzugreifen.

- Context Parameter dienen der Abbildung verschiedener Anwendungsrandbedingungen wie zum Beispiel Einbaustände (z.B. eingefahren - ausgefahren, etc.) oder auftretende Kräfte (z.B. Lagerreaktionen).
- Representation Attributes spezifizieren zusätzliche Informationen die für die Erzeugung der geometrischen Repräsentation notwendig sind (z.B. die Fasen am Innen- und Aussenradius der Unterlagscheibe).
- Mittels den Derivation Functions können die Ausprägungen von Representation Attributes basierend auf Context Parameters und/oder den Identification Attributes errechnet werden. Beispielsweise könnten die Anfasungen an der Unterlagscheibe basieren auf der Dicke der Scheibe mittels einer Berechnungsvorschrift errechnet werden.

Wie oben beschrieben kann Bauteilen eine beliebige Anzahl an Repräsentationen zugeordnet werden (z.B. ein Volumenmodell, ein Symbol, eine FEM-Modell, etc.). Innerhalb des Functional Models können diese ähnlich wie die Bauteile im General Model abstrahiert und zu Gruppen zusammengefasst werden. Die verschiedenen dreidimensionalen geometrischen Repräsentationen (z.B. CSG-Modell, B-Rep-Modell, Kantemodell) können in abstrakten Repräsentationskategorien (Representation Categories) zusammengefasst und über eine view logical

name identifiziert werden. Innerhalb dieser Representation Categories werden die einzelnen dreidimensionalen geometrischen Repräsentationsformen mittels der View Control Variable identifiziert.

In gleicher Weise wie die Generic Families können einzelne Functional Model classes wiederum zu Gruppen höheren Abstraktionsgrades zusammengefasst und auf diese Weise eine hierarchische Struktur von Repräsentationskategorien aufgebaut werden.

Verschiedene Functional classes korrespondierend mit den verschiedenen Repräsentationskategorien können einer gemeinsamen General Model class zugeordnet werden. Entsprechend kann eine bestimmte Functional Model class verschiedenen General Model classes beispielsweise unterschiedlicher Lieferanten zugeordnet werden.

Der Vererbungsmechanismus kann ebenfalls auf das Functional Model angewendet werden. Auf diese Weise können Eigenschaften welche für mehrere Functional classes gültig sind global für diese Klassen definiert werden. Eine vereinfachte Geometrirepräsentation kann somit übergeordnet für alle Schraubenfamilien definiert werden.

### General view's und Functional view's

Nach der Auswahl des benötigten Objektes in der Struktur des General Model's sowie der gewünschten Repräsentation in der Struktur des Funktional Models, muss diese in das jeweilige Produktmodell integriert werden. Hierfür wird entweder mittels der einem Functional Model zugeordneten Erzeugungsvorschrift die benötigte Repräsentation generiert.

Die zweite Möglichkeit besteht darin, dass die zu integrierenden Repräsenationen in einem geeigneten Datenformat vorliegen und lediglich von der entsprechenden Functional Model class als externes File referenziert werden.

Die durch die Library generierte Repräsentation bzw. die Referenz auf das externe File wird mit zwei Attributen versehen und über sogenannte Representation Transmission Interface an das jeweilige CAx-System übergeben:

- der General View beinhaltet alle Angaben bzw. Informationen der zugehörigen General Model class um das ausgewählte Objekt eindeutig identifizieren zu können.
- der Functional View nimmt die Identifizierung des „Functional Models“ sowie die Werte der der „View Control Variables“ zur eindeutigen Identifikation der ausgewählten Repräsentation wie z.B. die gewählten Werte für Ansicht, Ansichtsvariante und Darstellungsgrad.

Mittels der im General View und Functional View gespeicherten Informationen bleibt der Ursprung des Objektes erhalten. Es kann, ohne noch einmal den Auswahlprozess durchlaufen zu müssen jederzeit weitere Ansichten generiert, bzw. zwischen Darstellungsgraden umgeschaltet werden.

Functional View's werden entsprechend den Objekten im General Model innerhalb sogenannter Functional View Classes durch View Controll variables beschrieben. Beispielhafte Ausprägungen einer View Controll variable sind „geometric\_level = solid“ oder die verschiedenen Ansichten. Functional View Classes können wiederum auf eine Klassenhierarchie abgebildet werden

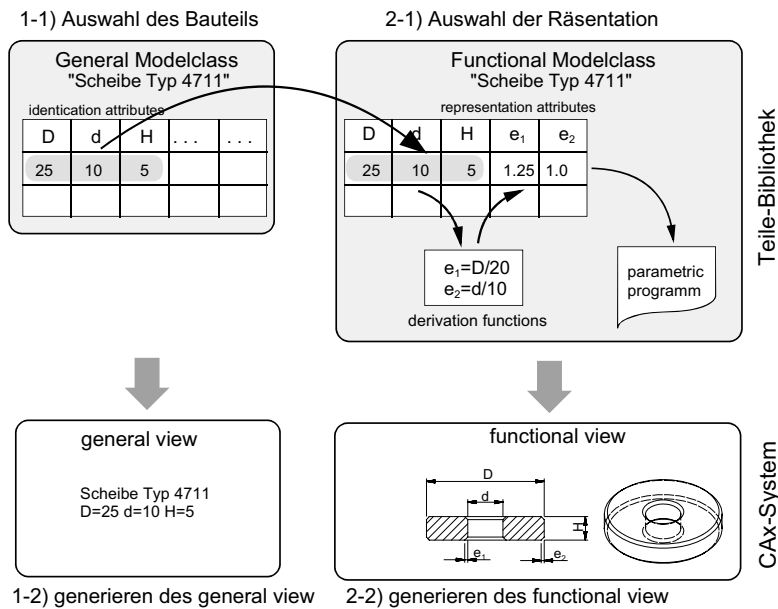


Bild (B024srpZ) Generierung des General View und des Functional View

## 7.5 Das Information Semantic Reference Model

Im Information semantic Reference Model wird im wesentlichen ein Referenzmechanismus definiert, der die strikte Trennung zwischen Informationsbeschreibung und tatsächlichen Inhalt einer Teilebibliothek ermöglicht. Zentraler Punkt dieses Referenzmechanismus ist die sogenannte Basic Semantic Unit (BSU), mittels welcher sämtliche Informationsinhalte (Klassen, Merkmale, Lieferanten, Objektrepräsentationen, etc.) eindeutig identifiziert werden können.

Aus der strikten Trennung von Inhaltsbeschreibung und Inhalt resultiert eine zweistufige Abstraktion innerhalb einer Teilebibliothek:

- Innerhalb der ersten Schicht werden die prinzipiellen Konzepte hinsichtlich der Teilefamilien und deren jeweiligen Attribute definiert (semantic description). Mit dieser Definition wird die Ontologie der Bibliothek festgelegt. D.h. es wird definiert welche Art von Bauteile einer Teilefamilie zugeordnet werden dürfen und durch welche Merkmale diese Familie beschrieben wird.
- Ziel der zweiten Schicht (logical description) ist die Festlegung welche konkreten Bauteile welcher Teilefamilien zugehörig sind. Es handelt sich dabei um eine Aufzählung der entsprechenden Bauteile.

Beide Schichten sind unabhängig voneinander. So ist es möglich, dass bestimmte Teilefamilien keine logische Beschreibung aufweisen. In diesem Fall handelt es sich um abstrakte Teilefamilien, welche lediglich abstrakte Bauteile enthalten, beispielsweise beschrieben durch eine Norm oder um Teilefamilien, welche derzeit gerade keinen Inhalt aufweisen.

Die logische Beschreibung einer Teilefamilie kann sich ändern ohne Änderungen der semantischen Beschreibung nach sich zu ziehen. Dies ist der Fall falls Bauteile hinzugefügt oder aus einer Teilefamilie entfernt werden.

Diese zweischichtige Beschreibung ist die Voraussetzung um Teilebibliotheken verschiedener Lieferanten aus dem selben Dictionary referenzieren zu können. Auf diese Weise kann ein Teilelieferant eine dedizierte semantische Beschreibung seitens eines existierenden Standards oder eines anderen Herstellers im Dictionary referenzieren und seine Bauteile unabhängig von eintragen.

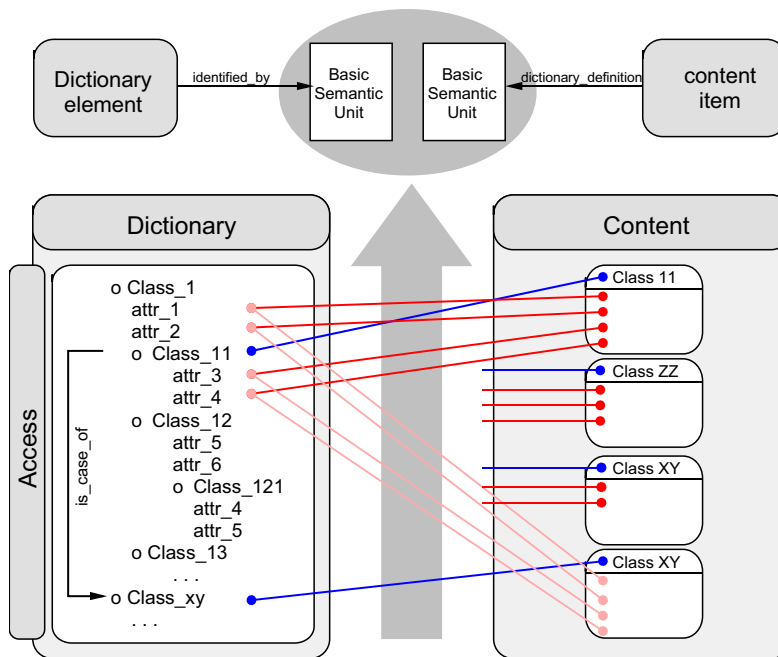


Bild (B025srpZ) zweistufige Abstraktion einer Teilebibliothek

Aufbauend auf dieser zweistufigen Abstraktion wird die Integration von Teilefamilien in eine beliebige Stelle der Referenzhierarchie durch einen Anbieter ermöglicht. Mittels des hierfür vorgesehenen `is_case_of` Konstruktes kann der Zusammenhang abgebildet werden dass die Beschreibung einer Familie A in der Beschreibung der Familie B verwendet werden soll. Für die Suche ergibt sich somit: Wenn Teile nach Merkmalen von B ausgesucht werden sollen, dann werden aufgrund der zuvor definierten `is_case_of` Relation ebenso Merkmale von A mit abgesucht.

Damit einzelne Teilebibliotheken bzw. Teile dieser ausgetauscht bzw. in eine integrierte Bibliothek integriert werden können (z.B. für eine partielle Aktualisierung) muss die Aufteilung auf eine beliebige Anzahl an Austauschfiles ermöglicht werden. Dabei muss zu jeder Zeit sichergestellt werden, dass der Bibliotheksinhalt eindeutig die zugehörigen Beschreibung referenziert.

Da EXPRESS -Attribute nur Referenzen auf Daten innerhalb desselben physischen Files enthalten können, ist es nicht möglich EXPRESS-Attribute für die benötigte Referenzierung zu verwenden. Aus diesem Grund wird das Konzept der `basic_semantic_unit` (BSU) eingeführt. Die BSU gewährleistet zu jeder Zeit die eindeutige Identifikation von Bibliotheksbeschreibungen (vgl. Bild B025srpZ).



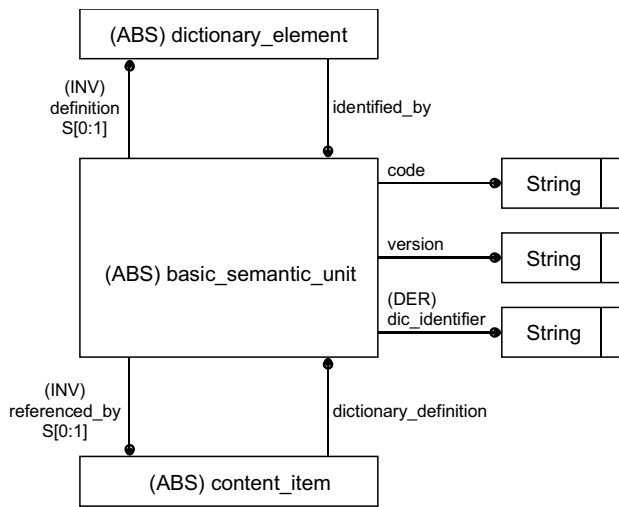


Bild (B026srpZ) Verknüpfung Dictionary/Content

## 7.6 Ausblick

Mit der immer mehr an Bedeutung gewinnenden digitalen Produktkatalogen (z.B. auf CD-ROM oder online auf dem Internet) und zunehmenden Druck auf Zukaufkomponenten innerhalb der Produktentwicklung zugreifen zu müssen werden jedoch Systeme benötigt, welche unabhängig von den eingesetzten CAD-System Teillebibliotheken unterschiedlicher Anbieter aufnehmen können. Innerhalb der ISO 13584 wurden die Grundlagen für leistungsfähige Bibliothekssysteme entwickelt und parallel dazu standardisiert. Es handelt sich dabei um ein sehr leistungsfähiges Konzept. Nachteilig ist die damit verbundenen hohe Komplexität der ISO 13584. Der Umfang der Norm als solches übertrifft damit die Funktionalität heutiger Teillebibliotheken bei weitem. Mit Implementierungen die den vollen Leistungsumfang aufweisen ist somit erst mittel- bis langfristig zu rechnen.

## 8 Zusammenfassung

Durch die Integration von CAx-Systemen in die rechnerintegrierte Produktentwicklung entstehen Schnittstellen. Ziel muss es sein, diese Übergänge zu optimieren, um eine durchgängige Informationsverarbeitung und einen reibungslosen Informationsaustausch sicherzustellen. Es werden Schnittstellen zum Austausch von Produkt- und Prozessdaten unterschieden.

Die ISO-Norm 10303 „STEP“ (Standard for the Exchange of Product Model Data) ist eine auf einem integrierten Produktmodell basierende Schnittstelle. Sie erlaubt, innerhalb des gesamten Produktlebenszyklus erzeugte Produktdaten auszutauschen, zu transformieren, zu speichern und zu archivieren.

Mit der Integration von Norm- und Zukaufteilen wird der Austausch von überbetrieblichen Informationen innerhalb einer Produktentwicklung notwendig. Dies geschieht in erster Linie mittels neutralen Austauschformaten, welche zielsystemunabhängig sind. Die ISO-Norm 13584 „Parts Library“ ist eine zu STEP komplementäre Schnittstelle. Sie regelt die Darstellung von Teilebibliotheken bzw. Zukaufteilkatalogen sowie den überbetrieblichen Datenaustausch.

CAx-Schnittstellen werden abhängig von ihrer Verwendung und der Integration in Hardware- und Softwareschnittstellen klassifiziert

### Verständisfrage 1

Was versteht man im Allgemeinen unter einer Schnittstelle?

### Verständisfrage 2

Worin liegen die Vorteile genormter Schnittstellenformate?

### Verständisfrage 3

In welche Klassen werden CAx-Schnittstellen im Allgemeinen eingeteilt?

### Verständisfrage 4

Aus welchen Bausteinen setzt sich ein CAx-System zusammen?

### Verständisfrage 5

Welche interne Schnittstellen werden benötigt um die Bausteine miteinander zu koppeln?

#### Verständisfrage 6

Welche externe Softwareschnittstellen weist ein CAx-System auf?

#### Verständisfrage 7

Was ist ein Postprozessor?

#### Verständisfrage 8

Was ist ein Preprozessor?

#### Verständisfrage 9

Welcher Prozessor wird benötigt um ein CAD-Modell in eine neutrales Modell zu transformieren?

#### Verständisfrage 10

Was versteht man unter einem neutralen Austauschformat?  
Welche unterschiedlichen Informationsinhalte werden innerhalb des Produktlebenszyklus unterschieden?

#### Verständisfrage 11

Welches sind die wichtigsten Schnittstellen zum Austausch von Prozessdaten?

#### Verständisfrage 12

Welches sind die wichtigsten Schnittstellen zum Austausch von Produktdaten (Name, Normungsorganisation und Modellinhalte)

#### Verständisfrage 13

Unter welcher Zielsetzung wurde die ISO 10303, STEP entwickelt?

#### Verständisfrage 14

Wodurch ist die STEP-Architektur charakterisiert?

#### Verständisfrage 15

Wie spiegelt sich dieser Architektur im Aufbau der Norm wider?

#### Verständisfrage 16

Worin besteht der Unterschied zwischen den Integrierten Ressourcen, IR und einem Application Protocol?

**Verständisfrage 17**

Welches sind die wichtigsten Elemente der Integrierten Ressourcen?

**Verständisfrage 18**

Warum wird innerhalb von Part 44 (Product Structure) das Konstrukt „specified\_higher\_usage\_occurence“ benötigt?

**Verständisfrage 19**

Wodurch zeichnet sich das Produktstrukturmodell aus?

**Verständisfrage 20**

Was versteht man unter der Initial Release von STEP?

**Verständisfrage 20**

Welchen Umfang hat die Initial Release von STEP?

**Verständisfrage 21**

Wofür dienen die Application Resources?

**Verständisfrage 22**

Wodurch ist der Aufbau eines Application Protocols charakterisiert?

**Verständisfrage 23**

Worin besteht der Unterschied zwischen einem ARM und einem AIM?

**Verständisfrage 24**

Was versteht man unter einem Mapping Table?

**Verständisfrage 25**

Wofür wird der Mapping Table benötigt?

**Verständisfrage 26**

Welches sind die wichtigen Application Protocols für den Maschinenbau?

**Verständisfrage 27**

Was versteht man unter SDAI?

**Verständisfrage 28**

Was versteht man unter dem Begriff „AP-Interoperability“

**Verständisfrage 29**

Was versteht man unter dem Begriff „AP-Modularisation“?

**Verständisfrage 30**

Worin besteht die Kritik hinsichtlich der STEP-Architektur?

**Verständisfrage 31**

Worin liegt das Problem beim Mapping?

**Verständisfrage 32**

Wie ist der prinzipielle Aufbau einer CAD-teilebibliothek gekennzeichnet?

**Verständisfrage 33**

Was versteht man unter einer Erzeugungslogik?

**Verständisfrage 34**

Welche Klassen von Varianten können hinsichtlich der Beschreibung durch Erzeugungslogiken unterschieden werden?

**Verständisfrage 35**

Was versteht man unter einer deskriptiven Geometriebeschreibung?

**Verständisfrage 36**

Was versteht man unter einer prozeduralen Geometriebeschreibung?

**Verständisfrage 37**

Was ist der Unterschied zwischen einer massvarianten und einer massinvarianten Geometriebeschreibung?

**Verständisfrage 38**

Ermöglicht STEP die Beschreibung massvarianter Geometrien?

**Verständisfrage 39**

Worauf zielt die DIN V 4001 ab?

**Verständisfrage 40**

Welche Sachmerkmalschlüssel werden innerhalb der DIN V 4001 unterschieden?

**Verständisfrage 41**

Worin liegt die Zielsetzung der einzelnen Sachmerkmaltypen?

**Verständisfrage 42**

Welche Geometrieklassen werden innerhalb der DIN V 4001 unterschieden?

**Verständisfrage 43**

Worin liegt die Zielsetzung der VDA-PS?

**Verständisfrage 44**

Auf welcher Programmiersprache beruht die DIN 66304?

**Verständisfrage 45**

Was ist daran zu kritisieren?

**Verständisfrage 46**

Worin besteht die Zielsetzung der ISO 13584?

**Verständisfrage 47**

Was sind die beiden grundlegenden Elemente der ISO 13584?

**Verständisfrage 48**

Was ist Aufgabe des „General Models“?

**Verständisfrage 49**

Was ist Aufgabe des „Functional Models“?

**Verständisfrage 50**

Wie ist die ISO 13584 aufgebaut?

**Verständisfrage 51**

Was versteht man unter dem „Actor Reference Model“?

**Verständisfrage 52**

Welche Rollen werden innerhalb des „Actor Reference Models“ unterschieden?

**Verständisfrage 53**

Was ist die Aufgabe des „Integrators“?

**Verständisfrage 54**

Welches sind die funktionellen Bestandteile einer Teilebibliothek welche innerhalb des „Architecture Reference Models“ unterschieden werden?

**Verständisfrage 55**

Was ist der Unterschied zwischen einem „free attribute“, einem „Identification attribute“ und einem „Derived attribute“?

**Verständisfrage 56**

Was versteht man unter dem Teilekonzept der ISO 13584?

**Verständisfrage 57**

Was versteht man unter einer Teilefamilie?

**Verständisfrage 58**

Worin liegt der Unterschied einer Simple Family und einer Generic Family?

**Verständisfrage 59**

Was versteht man unter einem generischen Bauteil?

**Verständisfrage 60**

Was versteht man unter einer Sicht?

**Verständisfrage 61**

In welchem Model werden die Sichten definiert?



**Verständisfrage 62**

Worin besteht der Unterschied zwischen einer „a priori“ und einer „a posteriori“ definierten Sicht?

**Verständisfrage 63**

Welchen Zweck haben Repräsentationen und Repräsentationskategorien?

**Verständisfrage 64**

Was versteht man unter einem „Context Parameter“?

**Verständisfrage 65**

Worin unterscheiden sich ein „Context Parameter“ und eine „Representation Attribute“?

**Verständisfrage 66**

Was versteht man unter einem „General View“?

**Verständisfrage 67**

Wofür dient ein „General View“?

**Verständisfrage 68**

Wodurch ist das „Information Semantic Reference Model“ charakterisiert?

**Verständisfrage 69**

Was ist eine „Basic Semantic Unit“?

**Verständisfrage 70**

Warum wird das Konzept der „Basic Semantic Unit“ benötigt?

## Publikationsverzeichnis – Literatur

- [GAG-86] Grabowski H, Anderl, R., Glatz, R.: Schnittstellenproblematik für den Anwender. Wt-Zeitschrift für industrielle Fertigung, Bd. 76, S.212-218, 1986
- [AND-93] Anderl, R.: CAD-Schnittstellen - Methoden und Werkzeuge zur CA-Integration. Habilitationsschrift Uni Karlsruhe, München: Hanser Verlag 1993
- [GAP-93] Grabowski, H., Anderl, R. Polly, A.: Integriertes Produktmodell. Berlin: Beuth Verlag 1993
- [GrA-90] Grabowski, H., Anderl, R.: Produktdatenaustausch und CAD-Normteile. Ehningen: Expert Verlag 1990.
- [ISO 10303-1] ISO IS 10303-1: Product Data Representation and Exchange - Part 1: Overview and Fundamental Principles, Genf 1994
- [ISO 10303-44] ISO IS 10303-44: Product Data Representation and Exchange - Part 44: Integrated Resources: Product Structure Configuration, Genf 1994
- [ISO 10303-214] ISO CD 10303-214: Product Data Representation and Exchange - Part 214: Application Protocol: Core Data for Mechanical Aotomotive Design Processes. ISO TC 184 SC4 WG3 N509
- [ISO 10303-203] ISO IS 10303-203: Product Data Representation and Exchange - Part 203: Application Protocol: Configuration Controlled Design. Genf 1994.
- [ISO 13584-1] ISO CD 13584-1 Parts Library Part 1: Overview and Fundamental Principles ISO TC184/SC4 N519, 20. Dezember 1996.
- [ISO 13584-10] ISO CD 13584-10 Parts Library Part 10: Conceptual Model of Parts Libarary ISO TC184/SC4/WG2 234, 24. Februar 1995.
- [ISO 13584-42] ISO DIS 13584-42 Parts Library Part 42: Description methodology: Methodology for structuring part Families ISO TC184/SC4 , 29. August 1996



# Enterprise Application Integration (EAI)

Autor: Dr. E. Zwicker



# 1 Überblick

Innerhalb der verschiedenen Unternehmensprozesse (Innovationsprozess, Sekundär Entwicklungsprozess, Verkaufsprozess, Einkaufs- bzw. Logistikprozesse etc.) kommen neben den CAx-Systemen zur Produktdatenerstellung und dem PLM-System zur Produktdatenverwaltung weitere jeweils spezialisierte Softwaretools zur Anwendung. Als die wichtigsten sind in diesem Zusammenhang zu nennen:

- Enterprise Resource Planning (ERP)  
zur Anwendung der Planungs-/Steueraufgaben auf die logistisch-/dispositiven Unternehmensprozesse des gesamten Unternehmens;
- Supply Chain Management (SCM)  
zur Anwendung der Planungs-/Steueraufgaben auf die Einkaufsprozesse des gesamten Unternehmens;
- Customer Relationship Management (CRM)  
zur Steuerung und Verwaltung der Kundenbeziehungen von der ersten Kontaktaufnahme bis zur Betreuung innerhalb des Serviceprozesses;
- Spezielle Konfigurationstools  
zur kunden- bzw. auftragsspezifischen Zusammenstellung hochvariantenbehafteter Produkte;
- Visualisierungs- und Virtual Reality-Tools (VR)  
zur Präsentation und Begutachtung von Produkten in den frühen Phasen der Produktentwicklung bzw. der kundenspezifischen Produktpassung.

Die innerhalb des gesamten Produktlebenszyklus benötigten Funktionen, wie z.B. die CAD-Modellverwaltung, die Arbeitsplanerstellung, das Customer Relation Management, die Offerterstellung oder das Erstellen und Pflegen von maschinen- bzw. auftragsspezifischer Ersatzteillisten etc. werden weiterhin in den spezialisierten Subsystemen realisiert.

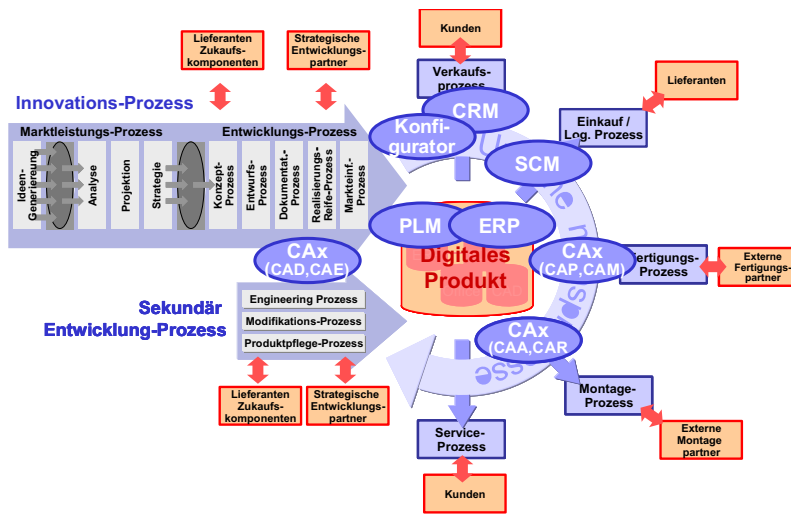


Bild (B001eaiZ) Softwaretools die innerhalb der verschiedenen Unternehmensprozessen zur Anwendung kommen

Unter dem Stichwort „Enterprise Application Integration“ versteht man die Integration der oben genannten jeweils autonomen Applikationen zu einer gemeinsamen Applikation

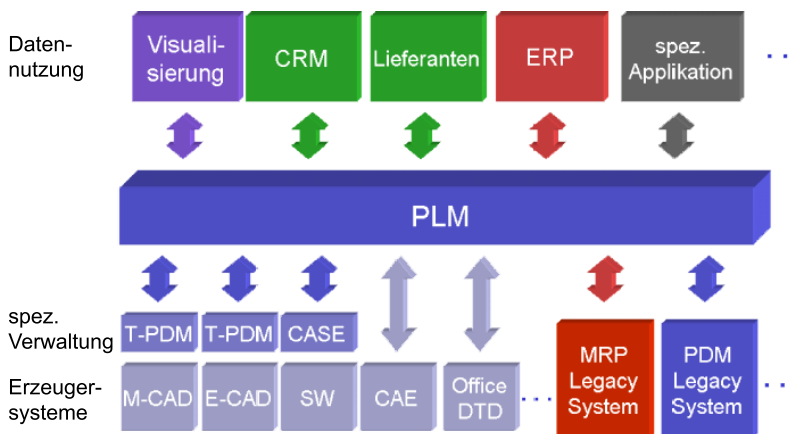


Bild (B002eaiZ) Das PLM-System als zentrale „Integrationsdrehscheibe“

Oftmals erfolgt diese Integration über ein zentrales PLM-System. Über das PLM-System werden diese Subsysteme in ein Gesamtsystem eingebunden und alle global relevanten Produktdaten (z.B. die Artikel bzw. Artikelstammsätze oder die Produktstruktur in den jeweils benö-

tigten Sichten) sowie die einzelnen Prozesse über den gesamten Produktlebenszyklus integriert und den spezialisierten Subsystemen durchgängig zur Verfügung gestellt. Die optimierten Teillösungen sind in ein sehr leistungsfähiges Gesamtsystem eingebunden. Es kann zu jedem Zeitpunkt im Produktlebenszyklus über das PLM auf die jeweils benötigten Informationen der einzelnen Subsystemen zugegriffen werden.

Über das PLM-System können neben den verschiedenen Applikationen ebenfalls vorhandene Alt-Systeme integriert werden. Dies erfolgt in der Regel dann, wenn aufgrund des im Altsystem vorliegenden Datenbestand es keinen Sinn macht, die vorhandene Applikation auf eine neue Applikation zu migrieren. Vielmehr sollen die vorhandenen Daten weiterhin genutzt werden.



## 2 Datenübergabe an ERP

Im folgenden soll anhand der Integration PLM – ERP detaillierter auf die Zielsetzungen, Möglichkeiten und Grenzen einer EAI-Integration eingegangen werden.

### 2.1 Prinzipielle Aufgabenteilung PLM und ERP

Die Aufgabenteilung zwischen dem PLM- und ERP-System ist in einem Unternehmen in der Regel wie folgt definiert:

- Das PLM-System
  - verwaltet alle produktbezogenen Daten über den Produktlebenszyklus;
  - unterstützt sämtliche entwicklungs-/engineeringtechnischen Prozesse zur Produktdefinition durch eine aktive Integration technischer Erzeugersysteme (CAD/CAM/CAE/...) und
  - ermöglicht einen kontrollierten Datenaustausch mit dem ERP-System.
- Das ERP-System
  - verwaltet alle Produkte, Prozesse und Ressourcen für die Produktion und
  - ermöglicht die Definition von Produktionsdaten und Stücklisten spezifisch für jeden Fertigungsstandort.

### 2.2 Informationsfluss PLM–ERP

Dementsprechend ergibt sich zwischen dem PLM- und ERP-system der folgende Informationsaustausch:

- von PLM nach ERP: Engineering/technische Daten
- von ERP nach PLM: Kommerzielle/logistische Daten

Die jeweilige Datenhoheit und Autorisierung für Änderungen müssen klar festgelegt sein.

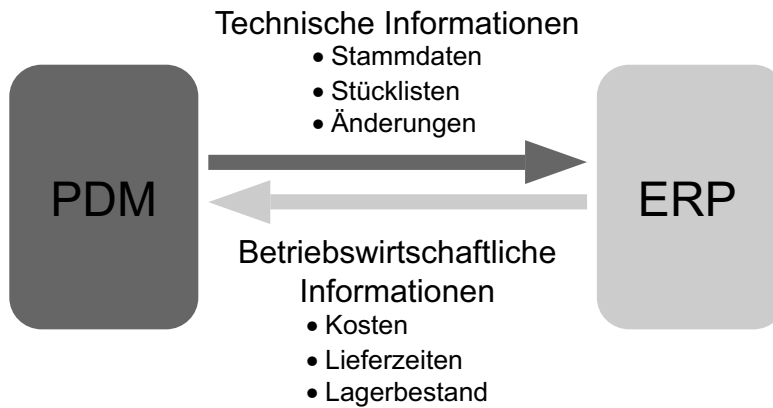


Bild (B003eaiZ) Grobcharakterisierung des Informationsflusses zwischen PLM und ERP (nach CIM Data)

Der Datenaustausch zwischen PLM und ERP hat zwar eine gewisse Ähnlichkeit zum Replikationskonzept bei verteilten PLM-Systemen, hebt sich bei genauerer Betrachtung aufgrund der Verschiedenartigkeit beider Systemwelten doch deutlich davon ab, was folgende Gesichtspunkte verdeutlichen:

- Unterschiedliche Informationsflüsse
  - Produktlebenszyklus bezogene Prozesskette in PLM: Produktdaten aus der Sicht Entwicklung – Produktion – Lieferung – Wartung
  - Ressourcen bezogene Prozesskette in ERP: Ressourcen, Termine, Kosten in den Phasen Entwicklung – Produktion – Lieferung – Wartung
- Unterschiedliche Informationsaspekte (Attribute)
  - produktdefinierende Attribute in PLM (Nummer, Revisionsstand, Konstrukteur, ...)
  - betriebswirtschaftlich/logistische bzw. ressourcenbezogene Attribute in ERP (Nummer, Herstellkosten, Gemeinkosten, Verkaufspreis, Liefertermin, ...)
- Paralleles Arbeiten in PLM und ERP: gleichzeitiges Schreibrecht in beiden Systemen auf verschiedene Attribute eines Objekts
  - Schreibrecht auf produktdefinierende Attribute liegt in PLM
  - Schreibrecht auf betriebswirtschaftliche Attribute liegt in ERP

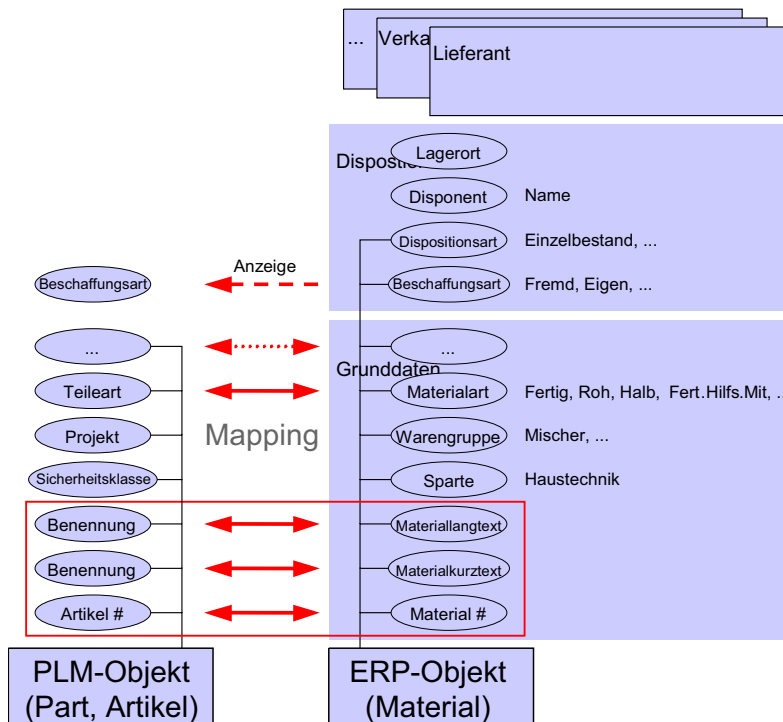


Bild (B004eaiZ) Verdeutlichung der unterschiedlichen Informationsaspekte in PLM und ERP

Ein wesentliches Merkmal von PLM/ERP-Schnittstellen ist die nur teilweise Überdeckung der Stammdaten-Attribute. Bezüglich dieser „gemeinsamen“ Attribute ist eine systemseitige Konsistenzsicherung erforderlich, damit immer nur in einem der Systeme Schreibrecht und die Möglichkeit zum Anlegen neuer Revisionen besteht. Dies ähnelt dem vorgestellten Replikationskonzept, bezieht sich demgegenüber jedoch nicht ausschliesslich auf die Granularität „Objekt“, sondern kann auch spezifisch für ein einzelnes Attribut unterschieden werden. Zur systemübergreifenden Konsistenzsicherung werden heute entweder systemseitige Mechanismen eingesetzt oder auch nur rein organisatorische Regelungen getroffen.

Von besonderer Bedeutung ist hierbei die Handhabung der Sachnummerung, wofür eine ganze Reihe von Möglichkeiten besteht, was die folgende Zusammenstellung verdeutlichen soll:

Gemeinsame unternehmensweite Sachnummer in PLM und ERP:

- Nummernhöhe liegt bei PLM
  - zweckmässig für sämtliche Konstruktionsstämme, die in PLM erzeugt und dann an ERP übergeben werden,

- problematisch für rein ERP-relevante Objekte, die in ERP erzeugt werden,
  - Nummernhoheit liegt bei ERP
    - zweckmässig für sämtliche ERP-relevante Objekte, die in ERP erzeugt werden
    - problematisch für Konstruktionsteile, die in ERP erzeugt werden,
  - Möglichkeiten zur Nummerngenerierung im jeweils untergeordneten System
- methodisch/organisatorisch:
- Methodische Unterteilung im Nummernschema zur dezentralen Nummernvergabe (z.B. hierarchische Bereichsunterteilung in PLM und ERP).
  - Unterschiedlicher Algorithmus in PLM und ERP (z.B. PLM vergibt nur gerade, ERP nur ungerade Nummern).
  - Verwendung vorläufiger Nummern, die nach der erstmaligen Datenübergabe aktualisiert werden (manuell und/oder automatisch).

Systemtechnisch:

- Periodisches Reservieren von Nummernblöcken im voraus (z.B. der nächsten hundert Nummern), die an das jeweils andere System abgetreten werden, welches dann hierfür die Vergabehoheit erhält.
- Online-Zugriff auf das jeweils andere System, beim Lösen einer Nummer (Problematik der Nichtverfügbarkeit; Einschränkung der Autonomie).
- Unterschiedliche Sachnummer in PLM und ERP
  - Bei erstmaliger Datenübergabe wird automatisch eine Nummer generiert (get\_max+1)
  - Bei erstmaliger Datenübergabe wird vom Benutzer die andere Nummer manuell eingegeben.

Da mit der Freigabe technischer Produktdaten ein Wechsel von der Definitionsphase in die Produktionsphase einhergeht, werden hier nach die zugehörigen Informationen auch im dafür zuständigen System benötigt, dem Produktionsplanungs- und -steuerungssystem (ERP). Dieser Datenübergabe von PDM an ERP kommt in heutigen Rationalisierungsbemühungen für Durchlaufzeitverringerungen eine zunehmende Bedeutung zu. Die Hauptzielsetzung besteht in einer automatisierten Übergabe von Konstruktionsstammdaten und Stücklisten, damit diese in PDM bereits digital vorliegenden Informationen nicht auf der ERP-Seite erneut von Hand mühsam und fehleranfällig wieder eingegeben werden müssen. Abhilfe schafft hier ein pro-

grammgesteuertes Anstossen der Datenübergabe (Triggerung) am Ende eines erfolgreich durchlaufenen Freigabeprozesses, woraufhin zunächst eine Reihe ERP-relevanter Validierungen durchgeführt wird (z.B. Mussfelder besetzt?) und dann der Eintrag in die ERP-Datenbank erfolgt.

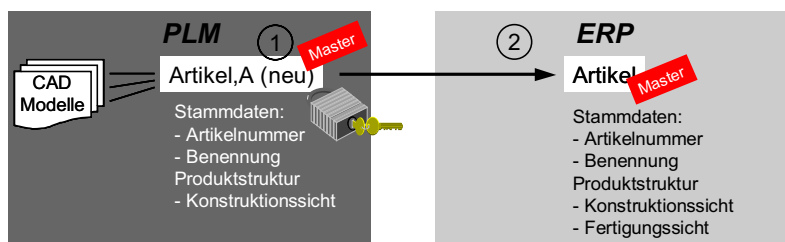


Bild (B005eaiZ) Übergabe Artikelstamm und Stückliste an das ERP mit der Freigabe im PLM

Ebenso wie bei den Attributen und der Benummerung bestehen auch für die Handhabung von Produktstrukturen mehrere Möglichkeiten, wobei zunächst interessiert, welche Struktursicht (Entwicklungssicht: Konstruktionsstückliste, KSL; Herstellungssicht: Fertigungsstückliste, FSL bzw. MSL: Montagestückliste, MSL) an ERP übergeben und dort bearbeitet wird:

- PLM KSL wird an ERP übergeben und dort als FSL eingelesen, die weiterbearbeitet wird.
- PLM KSL wird an ERP übergeben, dort als KSL eingelesen, wovon eine FSL abgeleitet wird, die weiterbearbeitet wird.
- In PLM wird KSL und FSL geführt, FSL wird an ERP übergeben und dort weiterbearbeitet.

In den beiden letzten Fällen wird die übergebene Struktur (in einem der beiden Systeme) aufbewahrt und die Weiterbearbeitung in ERP erfolgt auf einer Kopie dieser Struktur. Daher ist entweder innerhalb von ERP oder von PLM ein Vergleich von KSL und FSL möglich, wofür leistungsfähige PLM- oder ERP-Systeme eine sog. BOM Compare Funktion zur Verfügung stellen.

Die Notwendigkeit solcher Auswertemechanismen und die hierbei in der Praxis auftretende Komplexität anhand Bild B006eaiZ verdeutlicht werden: Während innerhalb der Entwicklungssicht die Stückliste vorwiegend nach funktionalen Gesichtspunkten aufgebaut wird, stehen in der Herstellungssicht fertigungstechnische bzw. montagetechnische Fragestellungen im Mittelpunkt. Ausserdem wird in der

Herstellungssicht die Stückliste um alle zur Herstellung, Montage und Vertrieb notwendigen Materialien wie z.B. Halbzeuge, Rohmaterialien oder Verpackungen erweitert.

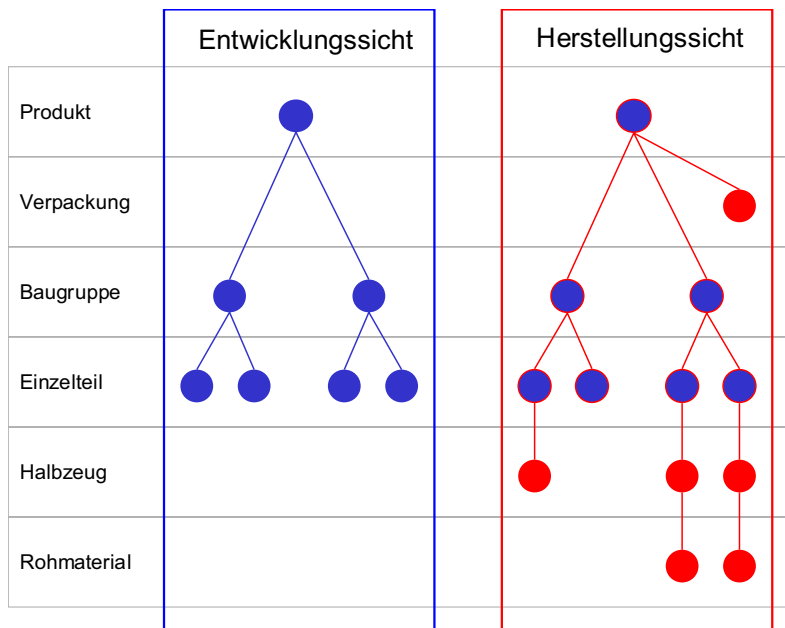


Bild (B006eaiZ) Entwicklungs- und Herstellungssicht

Von grundlegender Bedeutung bei einer jeden Möglichkeit zur Datenübergabe an ERP sowie bei der Generierung eines neutralen Formates zur Langzeitarchivierung und/oder zur Ausführung von Viewing und Markup ist die Einbettung in einen formalen Prozessablauf. Hierdurch kann die Ausführung dieser Prozessschritte systemseitig überwacht und dokumentiert werden.

Falls einmal freigegebene Objekte in irgendeiner Form modifiziert werden müssen, bedarf dies einer umfangreichen Änderung, deren Ablauf im Rahmen des betrieblichen Änderungswesens festgelegt ist.

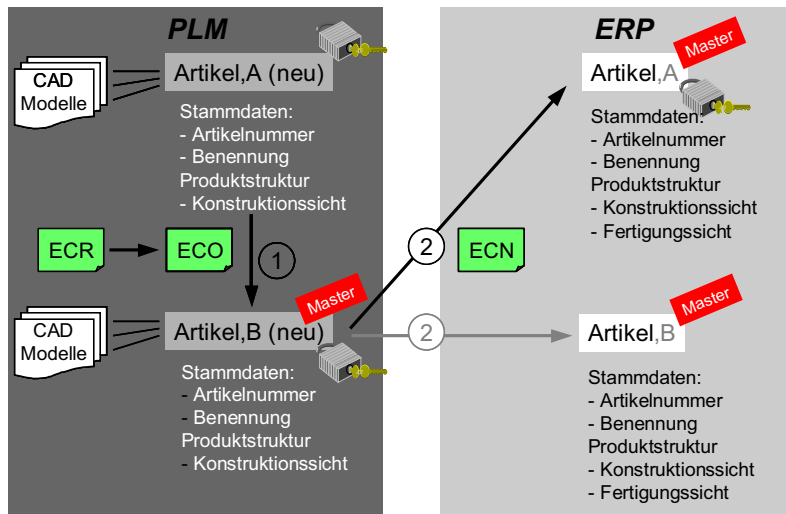


Bild (B007eaiZ) Übergabe Artikelstamm und Stückliste an das ERP mit der Freigabe einer Änderung im PLM

### 3 Systemtechnologie – Motivation

Eine verteilte, rechnerintegrierte Produktentwicklung hat als Kernproblem die Kommunikation zwischen heterogenen, teilweise über mehrere Länder verteilten Applikationen zu lösen. Dabei ist die Schnittstellenproblematik über mehrere Ebenen, angefangen von der physikalischen Verbindung über die Datenübermittlung bis zur semantischen Datenintegration vorhanden. In diesem Kapitel wird deshalb angefangen von der Übertragung elektronischer Signale bis zur Kommunikation über das World Wide Web die gesamte Problematik beleuchtet.

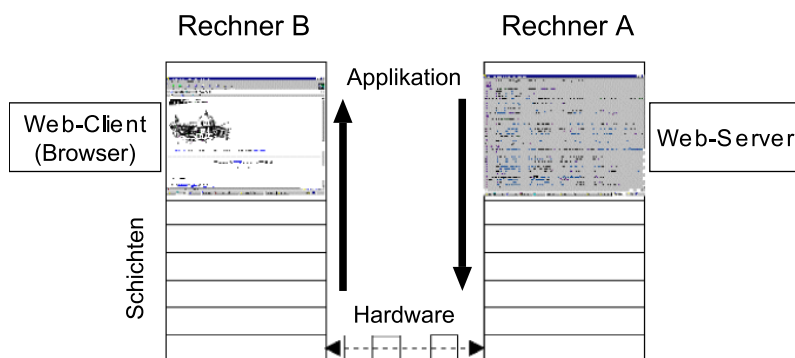


Bild (B008eaiZ) Schichtenmodell für Rechnerkommunikation

Bei einer Kommunikation zwischen zwei verschiedenen Rechnern, hat sich ein Schichtenmodell zur Präsentation der unterschiedlichen Kommunikationsebenen durchgesetzt. Die Schichten stellen auf jeder Ebene einen gewissen, in sich abgeschlossenen Standard (Protokoll) nach oben zur Verfügung, der es erlaubt, jede Ebene für sich zu betrachten und durch einen anderen Standard zu ersetzen. Im folgenden sollen die einzelnen Ebenen von unten nach oben beschrieben werden.

Eine Kommunikation setzt auf unterster Ebene erst einmal eine *physikalische Verbindung zwischen Geräten* voraus. Während man hierbei früher hauptsächlich elektrische Leitungen einsetzte, werden heutzutage auch andere Medien wie Glasfaser und Telekommunikationssatelliten verwendet. Genau wie bei der Kommunikation zwischen Menschen müssen auch bei Rechnern Vereinbarungen über die gemeinsame Sprache (Syntax und Semantik von Zeichen) getroffen werden, man nennt diese *Kommunikationsprotokolle*. Da gewisse



Anwendungen praktisch bei jeder Rechnerverbindung zum Einsatz kommen, haben auf Basis von Kommunikationsprotokollen gewisse *Applikationen*, wie z.B. E-Mail, Filetransfer, World Wide Web etc. zunehmende Bedeutung gewonnen. Durch die immer engere Vernetzung zwischen einzelnen verteilten Hard- und Software-komponenten hat sich unter dem Begriff *verteilte Systeme* eine globale Sicht auf einen Verbund von integrierten Einzelanwendungen durchgesetzt.

### 3.1 Physikalische Verbindungen zwischen zwei Datenendgeräten – High-Low, Ja-Nein und deren Übertragung

Wir wollen hier auf der untersten Schicht der Signalübertragung (der physikalischen) untersuchen, wie beispielsweise über eine serielle Verbindung zwischen zwei Geräten eine Informationsübermittlung stattfindet.

Bei der digitalen Informationsübertragung wird die Information logisch mit Nullen und Einsen bzw. mit einer Spannungspegeländerung übertragen. Eine Vereinbarung, wie sie z.B. in der RS232-Schnittstelle von der EIA (Electronic Industry Association) genormt ist, erkennt eine logische Eins bei einem Spannungspegel zwischen +5V und +15V, eine logische Null bei einem Pegel zwischen -5V und -15V. Um Spannungsabfälle auf der Übertragungstrecke zu tolerieren, erkennt der Empfänger eine Eins ab 3V und eine Null ab -3V. Alle Pegel müssen zur eindeutigen Erkennung eine gewisse Zeit auf der Leitung stabil stehen bleiben, wodurch der Übertragungsgeschwindigkeit eine natürliche Grenze gesetzt ist. Der Strom kann entweder auf der Sende- oder der Empfangsseite eingespeist werden. Die Schnittstelle, die die Stromquelle enthält wird als aktiv bezeichnet.

Um eine bidirektionale serielle Punkt zu Punkt-Verbindung zwischen zwei Geräten herzustellen, sind mindestens zwei Leitungen nötig, eine zum Senden von Impulsen TD (Transmit Data) sowie eine weitere zum Empfangen von Impulsen RD (Receive Data). Bei zwei Datenendgeräten (Source und Destination) werden diese Leitungen überkreuzt miteinander verbunden. TD des einen ist RD beim anderen (Bsp.: PC-Drucker, PC-PC mit PC-Direct). Dabei sind Leitungslängen bis 15m zugelassen.

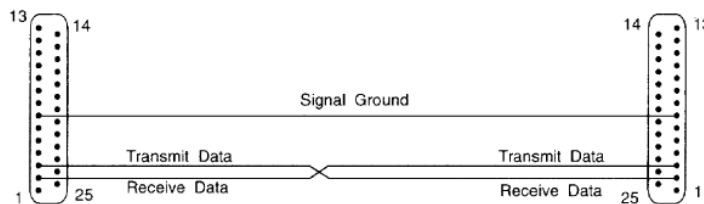
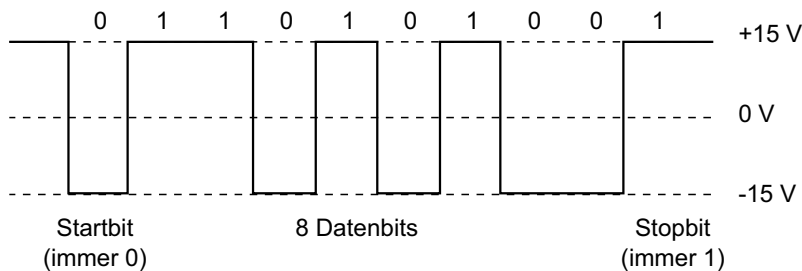


Bild (B009eaiZ) Serielle digitale Informationsübertragung über bidirektionale Verbindung

### 3.2 Physikalische Verbindungen zwischen zwei Datenendgeräten – Digitale Informationsübertragung

Bei der digitalen Informationsübertragung geht es darum, Information, die bereits in Nullen und Einsen codiert vorliegt über eine gewisse Entfernung zu transportieren und eine Verfälschung der Information durch die Übertragung auszuschliessen.

Bei der seriellen Zeichenübertragung wird jeweils ein Zeichen, bestehend aus einer Anzahl Bits, angefangen von der niederwertigsten Stelle, über ein Medium zu einem Empfänger geschickt. Dort wird es wieder zu einem (bspw. acht Bit breiten) Zeichen parallelisiert (z.B. A= 1000001; j=1101010 vergl. Bild B009eaiZ). Dazu werden sogenannte Schieberegister verwendet, welche eine seriell eintreffendes Signal wieder parallelisieren [TiSc-85].

Ein oft benutzter Zeichencode ist zum Beispiel der ASCII-Code. Hierbei handelt es sich um einen 8-bit-Code (=1Byte) welcher somit  $2^8=256$  mögliche Zeichen darstellen kann (In Bild B009eaiZ wird das Zeichen „j“ übertragen).

Dies bedeutet einen ständigen Wechsel des Spannungspegels an der Leitung. Der Empfänger setzt die Spannungsschwankungen in 0/1 Bits und diese in einem Schieberegister wieder in Zeichen zusammen. Meistens werden nicht nur reine Daten, sondern auch Kontroll- und Steuer-Bits mit übertragen, beispielsweise ein Start und Stopp-Bit oder

ein Paritäts-Bit zur Fehlererkennung (gerade Anzahl Einsen=> Parity=1).

### 3.3 Physikalische Verbindungen zwischen zwei Datenendgeräten – Synchrone und asynchrone Übertragung

Wie bereits erwähnt müssen die Spannungen zur Datenübertragung eine gewisse Zeit stabil an der Übertragungsstrecke anliegen, damit sie der Empfänger auch erkennt. Ausserdem müssen Sender und Empfänger noch eine Vereinbarung treffen, wie lange diese Zeiten (Übertragungsrate) sind, damit gewährleistet ist, dass zwischen zwei Pegeländerungen kein Bit verloren geht, oder zweimal dasselbe Signal abgetastet wird.

Bei der sogenannten *synchronen Datenübertragung* benutzen Sender und Empfänger die gleiche Abtastrate bzw. Signalübermittlungsrate. Zur Synchronisation wird eine bestimmte Bit-Folge (Synchronwort) eingefügt, damit der Empfänger den Beginn eines Datenblockes erkennen kann und sich mit seinem Taktgenerator auf das ankommende Signal synchronisieren kann - die Daten liegen in einem sogenannten Übertragungsrahmen zwischen zwei Synchronworten. Manchmal wird auch über eine spezielle Taktleitung der Takt mitgeschickt.

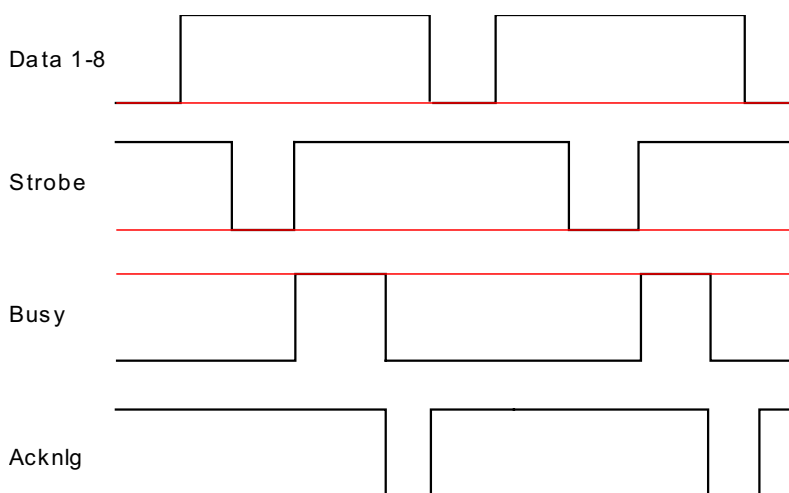


Bild (B010eaiZ) Echte asynchrone Übertragung mit drei Steuerleitungen

Bei der *asynchronen Übertragung* werden Sende- und Empfangstakt nicht synchronisiert, sondern nur ungefähr auf dieselbe Frequenz eingestellt ( $\pm 3\%$ ). Ein Start und Stopbit kennzeichnen den sehr kurzen Übertragungsrahmen (meist ein ASCII-Zeichen). Bei der echten asynchronen Übertragung werden Steuersignale (oft auf getrennten Steuerleitungen) zur Triggerung der Datenübergabe benutzt.

### Beispiel

Eine verbreitete Schnittstelle ist die sogenannte Centronics-Schnittstelle, welche insbesondere drei Steuerleitungen (Bild B010eaiZ) verwendet: Strobe (Datum übernehmen), Busy (Endgerät besetzt), Acknlg (Datum erkannt). Die Verbindung heisst asynchron weil der Zeitpunkt des Eintreffens einer Information dem Empfänger nicht bekannt sein muss, da er über die Steuerleitung sein Signal zum Informationsempfang erhält.

### 3.4 Physikalische Verbindungen zwischen zwei Datenendgeräten – Bussysteme

Ein Bus ist ein Leitungssystem zur Datenübertragung zwischen adressierbaren Einheiten in einem elektrisch und logisch verbundenen System. Während einer laufenden Übertragung sind aber immer nur zwei Teilnehmer ausgewählt und miteinander verbunden. Bussysteme sind im Prinzip asynchrone Schnittstellen mit freiem Zugang für (fast) beliebige Anzahl von Empfängern.

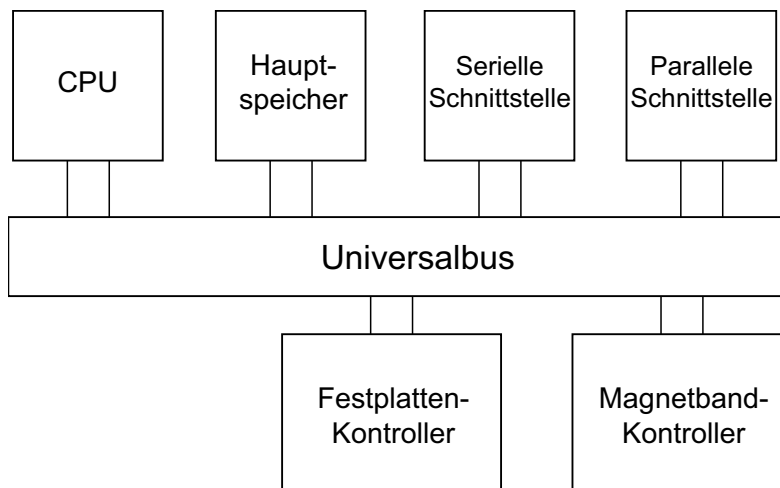


Bild (B011eaiZ) Rechnerarchitektur mit einem Universalbus

Jede PC und Workstationarchitektur baut prinzipiell auf mehreren Bussystemen auf (Adress-, Steuer-, Daten- und externe Bussysteme).

Eine Spezialstellung nimmt der weit verbreitete SCSI-Bus ein. Der SCSI-Bus ist ein Bussystem ohne spezielle Steuerleitungen, jedoch mit einem umfangreichen Befehlsprotokoll. Die Norm besagt welche Befehle ein an den Bus angeschlossenes Gerät verstehen muss. Alle Geräteinformationen – wo ein logischer Block auf der Platte physikalisch zu finden ist, wie die Daten auf dem Medium kodiert sind, etc.- muss im Ziel-Gerät selbst als Software codiert sein. Diese Struktur ähnelt stark der später beschriebenen Client-Server Architektur.

### 3.5 Physikalische Verbindungen zwischen zwei Datenendgeräten – Geschwindigkeit

Wie wir bereits gesehen haben, ist für die Übertragungsraten die Zeit des Anliegens einer Information auf einem Medium verantwortlich. Diese Zeit wird in bit/sec=Baud gemessen (Nicht zu verwechseln mit Byte/sec = 8 Bit /sec). Oft wird auch die Frequenz der möglichen Übertragungsraten in MHz angegeben, dabei ist jedoch wichtig, dass es sich hierbei um eine reine physikalisch mögliche Geschwindigkeit zum Wechseln von einem Pegel zu einem anderen handelt. Nicht berücksichtigt sind Wartezyklen vom Sender und Empfänger sowie Leerzyklen zur Stabilisierung des Signals auf der Leitung.

Die oben beschriebene RS232-Schnittstelle hat folgende übliche Bitraten: 110, 300, 1200, 2400 und 9600 bit/sec.

Ein Standard- SCSI-Bus unterstützt im synchronen Modus eine Taktrate von 5MHz bei einer Datenrate von 5 Mbyte. Da im voll synchronen Modus ohne Synchronisierungszeichen gearbeitet wird, ergibt sich eine Übertragungsrate von 5Mb/sec. Als „Fast SCSI“ werden 10 MHz Taktrate verwendet und als „Wide SCSI“ sind 4 Bytes Datenbreite vorhanden, was bei einer Kombination eine Übertragungsrate von 40 MByte/sec erlaubt.

### 3.6 Physikalische Verbindungen zwischen zwei Datenendgeräten – Modulation und Demodulation (Modem)

Unter Modulation versteht man die Überlagerung eines zu übertragenden Signales auf eine Trägerfrequenz. Aus elektronischen Gründen ist es meist einfacher, aus dieser Trägerfrequenz ein Signal herauszufiltern, als das Signal selbst zu übertragen. Typische Beispiele wie man sie auch aus dem Radiobereich kennt sind die Amplitudenmodulation (AM) und die Frequenzmodulation (FM). In der Grafik Bild B012eaiZ) sieht man eine Amplitudenmodulation eines 0/1 Rechtecksignales.

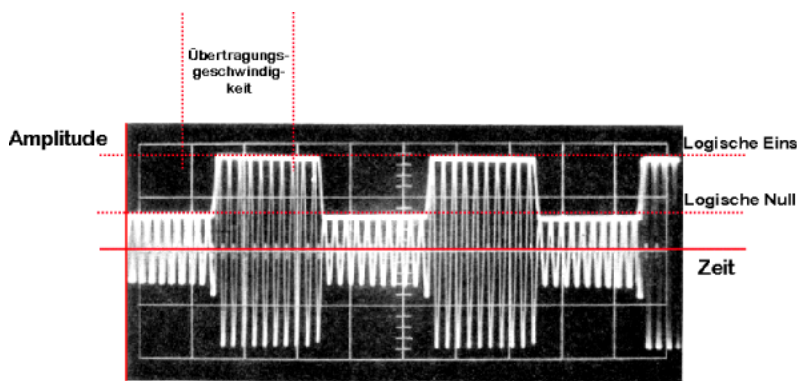


Bild (B012eaiZ) Übertragung einer Folge von 0/1 als Amplitudenmodulation

Dabei werden die Amplituden einer Trägerfrequenz in regelmässigen Abständen gemäss dem zu übertragenden Signal variiert. Im Gegensatz zur Amplitudenmodulation bleibt bei der Frequenzmodulation die Amplitude konstant jedoch ändert sich die Frequenz gemäss Trägersignal.

Folgende Modemgeschwindigkeiten bzw. Protokolle sind verbreitet:

- V.32: Standard für 9600 bps-Modem
- V.34: Standard für 28.800 bps-Modem
- V.42: Ein CCIT-Standard für Fehlerkontrolle, der automatisch eine fehlerhafte Übermittlung erkennt und neu sendet.
- V.42bis: Im Prinzip V.42 Fehlerkorrekturstandard mit Datenkompression. Theoretisch kann ist eine Kompressionsrate von 4:1 möglich, wird jedoch wegen Fehlerkorrekturen selten erreicht.

- V.90: Seit September 98 offiziell als Standard in Kraft, erlaubt die Übertragung von bis zu 56000 bps ohne Kompression
- Prinzipiell müssen sich Modemgeschwindigkeiten immer zwischen Sender und Empfänger automatisch synchronisieren, d.h. die maximale Geschwindigkeit zwischen Modulator und Demodulator bestimmt die Gesamtgeschwindigkeit.

### 3.7 Physikalische Verbindungen zwischen zwei Datenendgeräten – Integrated Services Digital Network (ISDN)

ISDN integriert eine ganze Reihe von Informationssystemen, wie Telekommunikation, Videoconferencing, ISDN Workstations, FAX-Service etc. in einem Protokoll (Integrated Services). ISDN bedeutet eine rein digitale Informationsübertragung von Sprache (Pulse Code Modulation) genauso wie von Daten (Digital Network). Auf der physikalischen Schicht können sowohl konventionelle Telefonleitungen als auch neuartige Medien wie z.B. Glasfaserkabel genutzt werden.

Prinzipiell unterscheidet man zwei verschiedene Anschlussmöglichkeiten:

- den Basisanschluss mit 144 kbit/s sowie den
- Primäranschluss mit 2048 kbit/s Übertragungsrate (Bild B013eaiZ).

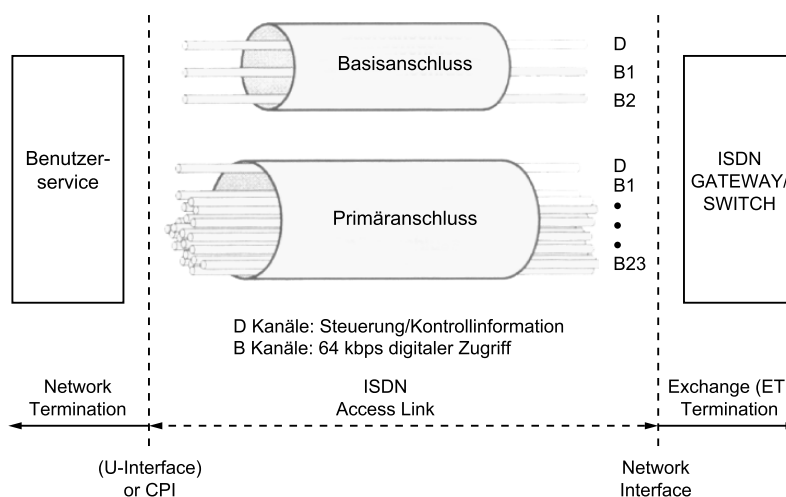


Bild (B013eaiZ) Struktur eines ISDN-Anschlusses



Jede ISDN-Verbindung liefert drei sogenannte Kanäle ein 16kbit/s sowie zwei 64kbit/s. ISDN liefert eine ca. viermal bessere Übertragungsgeschwindigkeit als konventionelle Analogmodems. Ausserdem werden eine Fülle neuer Möglichkeiten durch die digitale Verbindungserstellung geliefert, wie automatische Anrufnummernerkennung, automatisches Wählen bei besetzten Leitungen etc.. Typische ISDN-Anschlüsse kosten zwischen 40 Sfr. und 100 Sfr. im Monat.

### 3.8 Kommunikationsprotokolle

Protokolle sind eine Vereinbarung von Befehlssätzen zur Datenübermittlung. Sie befinden sich eine Schicht über der Zeichencodierung, d.h. mehrere Zeichen zusammengesetzt ergeben ein Steuersignal. Protokolle haben prinzipiell immer drei wichtige Ziele:

- Trennen von Kontrollinformation und Datenübertragung,
- Synchronisation der Protokollendpunkte,
- Effizientes Management der Ressourcen (z.B. Leitungen).

Am Beispiel des TCP/IP-Protokolls (Transmission Control Protocol/ Internet Protocol) soll im folgenden ein Protokollaufbau analysiert werden.

#### 3.8.1 Paketübertragung

Ähnlich wie bei der synchronen Datenübertragung mit Synchronisierungszeichen werden mit dem TCP/IP-Protokoll jeweils einzelne Datenpakete übertragen, die ihre eigene Adressinformation mit sich tragen. Jede Schicht wird über eine spezifische Header-Information angesprochen oder erweitert den Header mit spezifischen Informationen für die darunterliegende Schicht.

#### 3.8.2 Adressierung

Wichtig bei der Adressierung mit dem TCP/IP-Protokoll ist die Unterscheidung zwischen Datenendgeräten und reinen Datenweitergabestationen. Man unterscheidet zwischen Hosts, Gateways und Routern (siehe Grafik). Gateways übertragen Pakete von einem Netz mit unterschiedlichen Protokoll in das andere, während Router zwischen gleichartigen Netzen vermitteln, im Internet heissen sie IP-Router. Es setzt sich jedoch zunehmend auch für Router der Ausdruck Gateway durch.

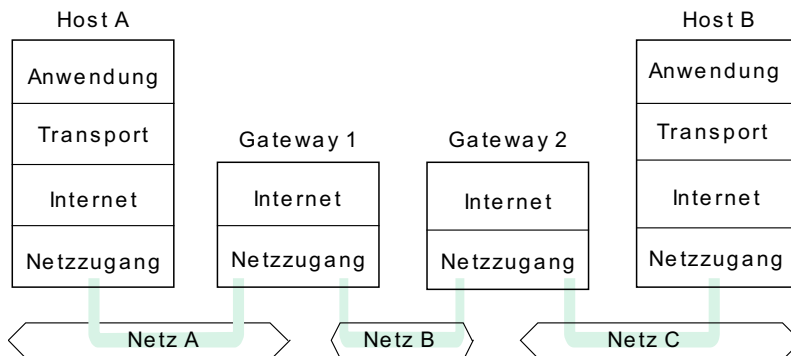


Bild (B014eaiZ) Funktion eines Gateway

IP ist die Adressierungsschicht, welche verantwortlich ist für die Übermittlung der einzelnen Datenpakete über Gateways zu einem Host in einem anderen Netzwerk. Dabei werden die einzelnen Pakete mit einem IP-Header ausgestattet, der beim Empfänger wieder entfernt wird. In der IP-Schicht erfolgt keine Rückmeldung über den Empfang einer Meldung oder erneutes Senden nach einem Fehler. Die in der IP-Schicht verwendeten Adressen werden durch das „Address Resolution Protocol“ dynamisch in die Hardware-Adressen des lokalen Netzwerks umgesetzt. Hierbei werden die Adressen von links nach rechts von übergeordnetem Routern den darunterliegenden bis zum Ziel-Rechner selbst übermittelt. So bedeutet bei der Adresse 132.128.123.11, dass der Hauptrouter die Nummer 132 besitzt und 11 die Netzwerkkarte im angesprochenen Rechner ist. Hierbei hat jeder von ca. 5 Millionen Rechnern im Internet eine eindeutige, nur einmal vorkommende Adresse.

### 3.8.3 Transportprotokoll

Die TCP-Schicht zerlegt einen Datenstrom in einzelne kleinere Pakete, übergibt sie an IP und setzt sie beim Empfänger wieder zusammen. Gehen einzelne Pakete bei der Übermittlung verloren, wird dies durch TCP erkannt und die Information erneut gesendet.

Auf gleicher Ebene wie TCP ist das User Datagram Protocol (UDP) angesiedelt. UDP erlaubt Anwendungsprogrammen den Datenaustausch über IP jedoch ohne das zeitaufwendige Auf- und Abbauen einer TCP-Verbindung. Das Anwendungsprogramm muss jedoch in der Lage sein, selbst Übertragungsfehler zu erkennen und die Sendung zu wiederholen.

Beim TCP spricht man von einem verbindungsorientierten Protokoll, weil der Sender auf die Bestätigung des Empfängers wartet, dass

die übermittelten Daten korrekt empfangen wurden. Solange diese Bestätigung nicht erfolgt, werden die Daten laufend neu gesendet. Das UDP ist ein verbindungsloses Protokoll, welches Daten mit einer Adresse und einer Prüfsumme sendet, ohne auf die Bestätigung des Empfängers über das korrekte Eintreffen zu warten.

Die TCP-Schicht wird in der Praxis manchmal auch durch andere auf IP-aufsetzende Protokolle ersetzt, um beispielsweise per Telefoneinwahl eine Verbindung zu einem Router herzustellen. Meist wird hierzu das SLIP oder PPP-Protokoll verwendet.

#### 3.8.4 Serial Line Internet Protocol (SLIP)

SLIP ist ein nicht offiziell standardisiertes (dennoch weit verbreitetes) Protokoll zum Netz-Zugang über eine Telefonleitung, wobei es erst einmal keine Rolle spielt, ob es sich um eine digitale (ISDN) oder analoge Verbindung handelt. Implementiert wurde das Protokoll erstmals 1984 zur Kommunikation unter SUN Workstations. SLIP ist darauf beschränkt auf eine darunterliegende IP-Protokollschicht aufzusetzen, d.h. es ist nicht möglich gleichzeitig verschiedene Protokolle unter SLIP zu verwenden.

Die Kosten für einen SLIP Anschluss betragen im Moment (1997) unter 30 Sfr./Monat für eine festgelegte Anzahl Stunden Zugriff.

#### 3.8.5 PPP

Das PPP (Point to Point Protokoll) wurde 1989 an der University of California in San Diego vorgestellt. Im Gegensatz zu SLIP ermöglicht dieses Protokoll eigene Fehlererkennung und Datenkompression sowie weitere eigene Funktionalitäten. Aus diesem Grunde ist PPP das mächtigere Protokoll für Netzwerkverbindungen über Telefonleitungen. Besteht die Wahlmöglichkeit, sollte man dem PPP-Protokoll den Vorzug geben[Gil-95].

#### 3.8.6 ISO/OSI 7-Schichtenmodell

Um einen Vergleich bzw. eine Einordnung von Kommunikationsprotokollen zu ermöglichen, hat die „International Standard Organisation of Open System Interconnection“ einen 7-schichtigen Standard definiert in den sich Protokolle einordnen lassen. Dabei baut jede Schicht auf dem Protokoll der unteren Schicht auf, d.h. auf jeder Schicht können eine Vielzahl verschiedener Protokolle zum Einsatz kommen. Ein Protokoll stellt dabei nach oben eine definierte Funktionalität zur Verfügung und setzt auf einer darunterliegenden Funktionalität auf,

ohne deren Realisierung zu beachten (*Encapsulation*). Im folgenden werden die sieben Schichten (Tabelle T001eaiZ) kurz beschrieben.

<i>Rechner A</i> <i>Betriebssystem A</i>		<i>Rechner B</i> <i>Betriebssystem B</i>	
Anwendung	1	Anwendung	1
Darstellung	2	Darstellung	2
Kommunikationssteuerung	3	Kommunikationssteuerung	3
Transport	4	Transport	4
Vermittlung	5	Vermittlung	5
Sicherung	6	Sicherung	6
Bitübertragung	7	Bitübertragung	7
Übertragungsmedium			

Tabelle (T001eaiZ) ISO/OSI-Referenzmodell

- Bitübertragung (Physical):  
Elektrische und mechanische Charakteristik der Physikalischen Verbindung.
- Sicherung (Data Link)  
Synchronisation, Rahmen (Bsp.: Start - Stopbit), Übertragungsfluss (synchron, asynchron etc.)
- Vermittlung (Network):  
Erstellen, Unterhalten und Beenden von Verbindungen. Adressierungs- und Routing-Funktionen.
- Transport (Transport):  
Datenfluss regeln zwischen Endbenutzern, Fehlererkennung und Korrektur, Mapping von Transportadressen zu Endbenutzern.
- Kommunikationssteuerung (Session):  
Bereitstellung der übertragenen Daten, Benutzerzugang zur Übertragung, Benutzer definiert den Grad der Kontrolle.
- Darstellung (Presentation):  
Darstellung der übertragenen Daten in benutzbarer und verständlicher Form.
- Anwendung (Application):  
Beispielsweise FTP, E-mail und andere Anwendungen die auf den darunterliegenden Protokollen aufsetzen.

### 3.8.7 Einordnung von TCP/IP in das ISO/OSI Modell

Da das TCP/IP-Protokoll vor der Normung des 7-schichtigen ISO/OSI-Referenzmodells entstand, ergibt sich eine gewisse Schwierigkeit bei der Einordnung dieses Protokolls. Im allgemeinen ist man sich darüber einig, dass TCP/IP eigentlich über weniger als sieben Schichten verfügt [Hun-95]. Zum besseren Verständnis soll hier dennoch eine Einordnung vorgenommen werden.

Das TCP/IP Protokoll umfasst alle Schichten des ISO/OSI Modells bis zum Application-Layer. Bei der Realisierung des TCP/IP-Protokolls mit Ethernet ist das Adress-Resolution-Protocol (ARP) dafür verantwortlich die IP-Adressen in hardware-spezifische Ethernet-Adressen umzusetzen. Die darunterliegende Schicht verpackt die IP-Datagramme in Ethernet-Rahmen und verschickt diese. Auf das CSMA/CD-Protokoll wird später eingegangen. Man sieht in Bild (B015eaiZ) auch noch eine Besonderheit des IP-Protokolls, welches auf Schicht drei und fünf arbeitet, weil IP einmal die Hardware Ethernet-Adressen verwaltet und auf Schicht fünf das Routing mit Internet-Adressen verantwortlich ist.

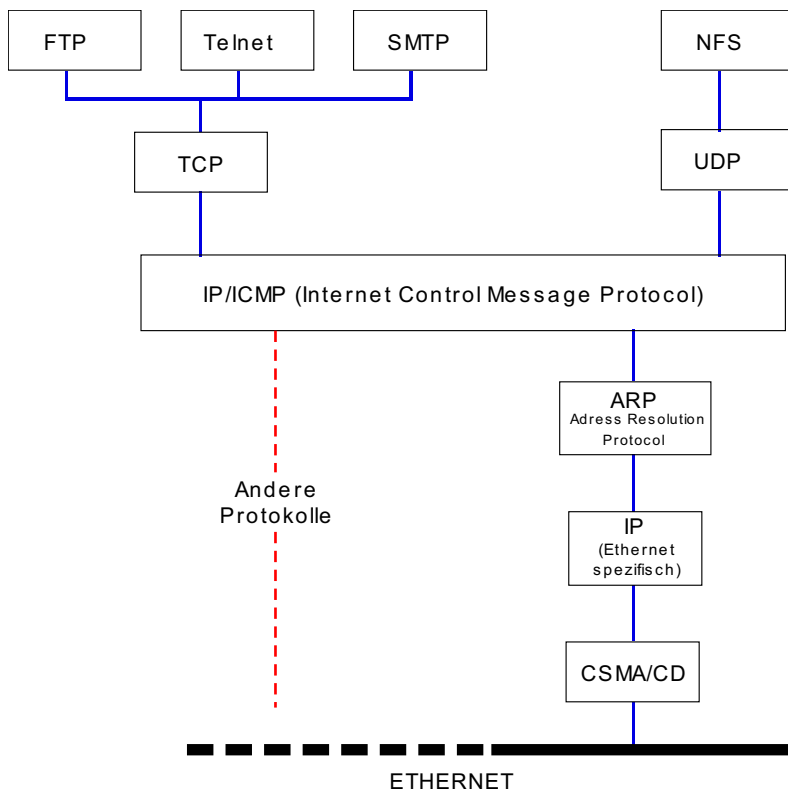


Bild (B015eaiZ) TCP/IP Realisierung mit Ethernet

Die Applikationen FTP, Telnet, SMTP und NFS werden später behandelt.

### 3.8.8 Internet

Auf Basis des TCP/IP Protokolls entstand ein Netz, welches heutzutage unter dem Namen Internet weltweit verbreitet ist.

Grundlage des Internets war 1970 der Zusammenschluss von vier Grossrechnern im Auftrag des amerikanischen Verteidigungsministeriums, dem sogenannten ARPAnet. Die Mitarbeiter erkannten schnell den Nutzen dieses Netzes als Kommunikationsnetz (E-Mail).

1974 wurde aus dem Problem, heterogene Netzwerktechnologien zusammenzubringen, das TCP/IP Protokoll definiert, welche heutzutage die technische Grundlage des Internet darstellt. Auf dieser Basis vernetzen sich immer mehr (hauptsächlich) Hochschulinstitute und so wurde 1980 das Internet geboren. Genutzt werden dabei hauptsächlich die Dienste FTP und E-Mail .

### 3.8.9 Internet-Adressierungsschema

Internet-Adressen beruhen prinzipiell auf dem IP-Protokoll zur Adressierung. In der Praxis wird jedoch eine zusätzliche Übersetzungsschicht zur besseren Transparenz der Adressen benutzt. Diese Schicht setzt benutzerfreundlichere Internet Namen in IP-Adressen um. Der Server, welcher die Tabellen zur Umsetzung der Adressen verwaltet, wird Domain Name Server (DNS) genannt.

#### Beispiel

Beispieladresse: [leonhardt@ikb.mavt.ethz.ch](mailto:leonhardt@ikb.mavt.ethz.ch). Bei dieser Adresse sind drei wichtige Details zu beachten:

1. ikb.mavt.ethz.ch ist keine hardware-abhängige Adresse, d.h. es wird noch eine zusätzliche Umsetzung auf eine hardwareabhängige Adresse vorgenommen. Dies hat zur Folge, dass bei einem Rechnerwechsel sich die E-Mail-Adressen nicht ändern.
2. leonhardt ist ein Betriebssystem-User unabhängiger Name, der intern auf den richtigen 8-Character User-Namen umgesetzt wird.
3. Damit auch versehentlich an leonhard statt an leonhardt gesandte e-mails weitergeleitet werden, ist auch diese Adresse im POP3-Server (siehe Kapitel 2.4.5) registriert.

Die Domain Name Server (DNS) greifen nicht auf eine gemeinsame zentrale Datenbank zu, sondern sind ein Netz von weltweit hierarchisch angeordneten Servern. Innerhalb einer Adresse sind die

Bezeichnungen rechts des Klammeraffen (Master-Space=@) hierarchisch von rechts nach links aufgebaut. Der am weitesten rechts stehende Teil der TCP/IP-Adresse wird Top-Level-Domain genannt, diese heissen z.B. com, edu, mil, etc.. Die Top-Level-Domains werden von sogenannten Root-Servern verwaltet, diese können nur zu einer Top-Level-Domain verweisen (bzw. dem dazugehörigen DNS)..

### 3.8.10 Zugang zum Internet

Es gibt prinzipiell drei verschiedene Möglichkeiten des Zugangs zum Internet.

- Direkter Anschluss mit einer eigenen Domain (connected network)
- über einen Online-Dienst
- über einen Provider der einen Rechner mit Internet-Zugang per Modem zu Verfügung stellt

Der direkte Anschluss zum Internet geschieht im allgemeinen mit einem Antrag bei einem Internet-Dienstanbieter (Provider). Die Kosten für eine eigene Adresse (Domain) betragen je nach Übertragungskapazität zwischen 10.000 CHF und 2.000.000 CHF.

Online-Dienste bieten neben dem reinen Zugang zum Internet noch zusätzliche Informationsangebote. Diese Dienste sind relativ günstig (ca. 20 CHF/Monat), da sie auf vorhandene Infrastrukturen (Telefonnetz zugreifen).

Die Internet Service Provider bieten ausschliesslich die Möglichkeit, per Modemzugang ihren eigenen Rechner mit Internetzugang zu benutzen (Kosten zwischen 10 CHF und 30 CHF)..

Wichtig ist in diesem Zusammenhang ebenso, dass nur ein direkt an das Internet angeschlossener Rechner den vollständigen Zugang auf andere Internet-Rechner erlaubt (Rlogin, X-Terminal, etc.).

## 3.9 Applikationen

Wie wir gesehen haben, befindet sich auf der obersten Ebene im ISO/OSI Protokoll die Anwendungsschicht. Diese umfasst alle Applikationen, die auf Netzwerkkommunikation aufsetzen, aber nur auf darunterliegende Dienste zugreifen und diese nicht selbst implementieren. Für die Kommunikation zwischen solchen parallel laufenden Applikationen ist der Begriff des Prozesses wichtig.

### 3.9.1 Prozess

Falls man bei einem Rechner keine von Neumann-Architektur mit rein sequentieller Ablaufsteuerung vorliegen hat, ist es wichtig, den Begriff des Rechenprozesses näher zu betrachten. „Ein Programm mit allen aktuellen Laufparametern sowie allen Betriebsmitteln ausser eventuell dem Prozessor“ [Wet-87] nennt man einen Prozess. Ein Prozess drückt Parallelität von Aktionen aus. Wieviele Prozesse gleichzeitig auf einem Rechner aktiv sein können (nicht virtuell sondern real) ist alleine durch die Anzahl der Prozessoren bestimmt. Das Betriebssystem übernimmt die Zuordnung von Prozess zu Prozessor. Ein Prozess durchläuft folgende Stadien (siehe Bild B016eaiZ):

- F: Prozess gestartet
- B: Betriebsmittel zugewiesen
- R: Prozessor zugeordnet
- W: Prozessor aufgeben
- N: Prozess beenden

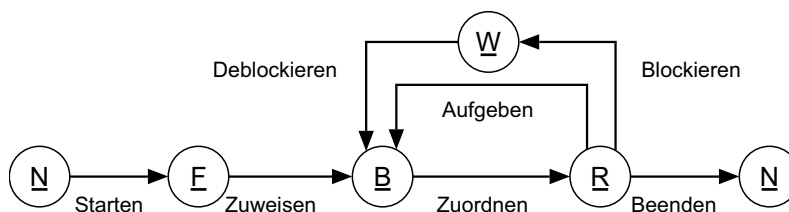


Bild (B016eaiZ) Stadien eines Prozesses

Für jeden dieser Zustände, ausser Prozessor zugeordnet, verwaltet das Betriebssystem eine Warteschlange und ordnet dem Prozess mit höchster Priorität eine freigewordene Ressource zu.



### 3.9.2 Socket Kommunikation

Wie kann man nun einen einzelnen Prozess auf einem beliebigen Internet-Rechner ansprechen?

Man benutzt dazu sogenannte Ports, die eine direkte Verbindung zu einem laufenden Prozess herstellen, welcher diesen Port laufend abhört. Eine Verbindung von Internet-Adresse und Port-Nummer wird als Socket bezeichnet. Sockets sind Kommunikationsendpunkte zwischen parallel laufenden Prozessen auf gleichen oder anderen Rechnern. Zwei jeweils verschiedenen Instanzen zugeordnete Sockets kooperieren (nicht gerichteter Datenaustausch) miteinander und erlauben somit eine Kommunikation zwischen den Prozessen. Es existieren unterschiedliche Arten von Sockets:

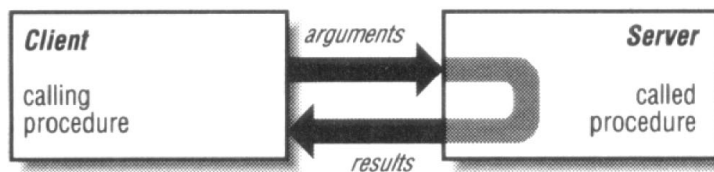
- Stream Socket  
Herstellung einer direkten Verbindung, keine Grenzen in Übertragungslänge, Übertragungssicherung durch TCP.
- Datagramm Socket  
Verbindungslose Übertragung, es wird ein komplettes unabhängiges Paket gesendet. Die Übertragungslänge ist limitiert durch die Grösse des bei einer Transaktion sendbaren Datenpakets.
- Raw Socket  
Erlauben den direkten Zugriff auf das Internet Protocol ohne die Sicherungsschicht.

#### Beispiel:

Ein Benutzer auf dem Rechner 128.66.12.2 baut eine TELNET-Verbindung (Applikation für Remote Login) zum Rechner 192.178.16.2 auf. Eine Ausgangs-Port-Nummer für den sendenden Prozess wird dynamisch vergeben, bspw. 3333. Der Telnet-Port läuft per Standard auf der Nummer 23. Das heisst, dass von Adresse 128.66.12.2.3333 (Socket) zu 192.178.16.2.23 eine Verbindung aufgebaut wird (TCP!) und über diese Verbindung kommuniziert wird.

### 3.9.3 Remote Procedure Call

Das RPC-Protokoll [Sun-88] ist eine Programmierschnittstelle mit deren Hilfe ein Programm direkt Routinen auf einem anderen oder dem gleichen Rechner aufrufen, und als eigenständigen Prozess starten kann. Bild B017eaiZ zeigt die Unterschiede von lokalem und Remote Procedure Call. Zu beachten ist hierbei, dass bei der RPC-Kommunikation zwei eigenständige Prozesse miteinander kommunizieren. Die Kommunikation zwischen den Prozessen erfolgt über das XDR (eXternal Data Representation)-Protokoll [Sun-87], d.h. Argumente und Resultate werden auf Client und Server Seite mit diesen Routinen codiert und auf die lokale Repräsentation decodiert. Somit ist mit RPC eine Programm zu Programm-Kommunikation ohne Rücksicht auf interne Datenrepräsentation und Formate möglich.



#### Local Procedure Call

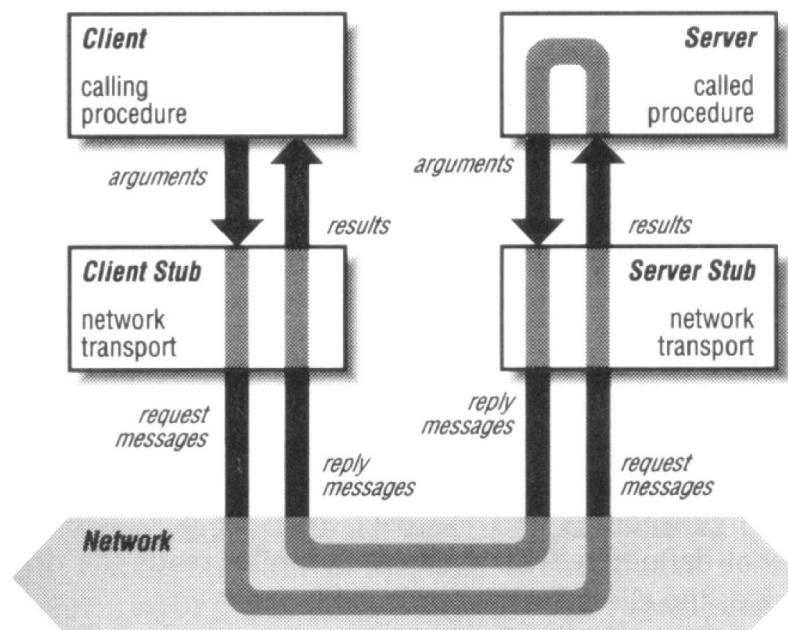


Bild (B017eaiZ) Remote Procedure Call vergl. mit lokalem Procedure Aufruf

Der einen RPC aufrufende Prozess wird als Client bezeichnet. Er ruft einen sogenannten RPC-Server auf einem anderen Rechner auf, welcher einen Prozess mit der angeforderten Routine startet. Nach seiner Beendigung sendet der Server eine Nachricht mit den aktuellen Berechnungsergebnissen an den Client.

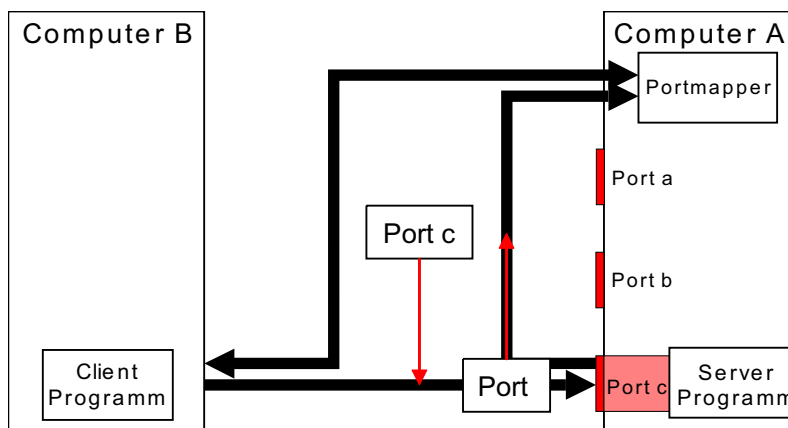


Bild (B018eaiZ) RPC-Kommunikation über Ports

Client und Server kommunizieren über sogenannte Ports (siehe Bild (B018eaiZ) reserviert für RPC) mit einem "callrpc" ein Portmapper genanntes Programm, welches dem Client die Port- Nummer (im Beispiel c) der gewünschten Routine mitteilt. Client und Server kommunizieren über diesen Port miteinander.

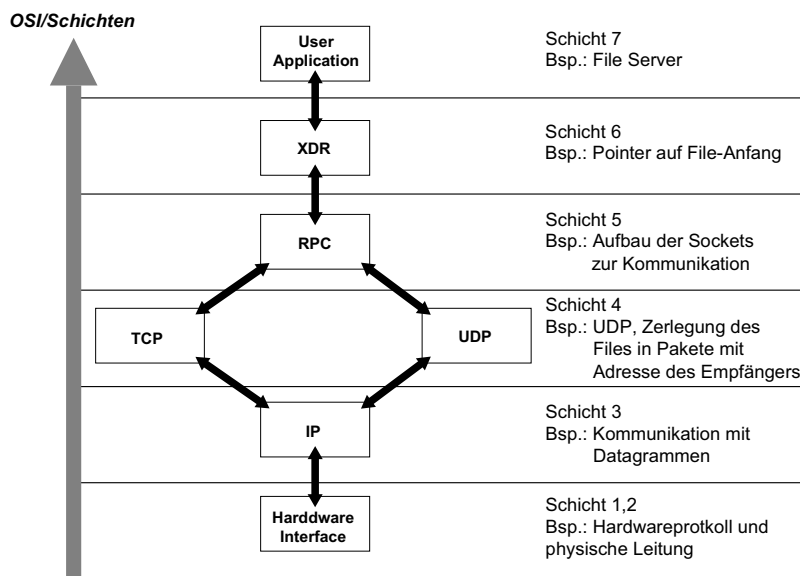


Bild (B019eaiZ) Einordnung von RPC in das ISO/OSI-Modell

Wie Bild B019eaiZ zeigt, setzt RPC auch auf dem bereits behandelten TCP/IP Protokoll auf. Wichtig ist hierbei jedoch, dass über RPC nicht Rechner miteinander kommunizieren, sondern Prozesse auf durch Internet-Adressierung spezifizierten Computern. Daraus folgt, dass eine RPC-Kommunikation durchaus auch auf ein und demselben Rechner stattfinden kann.

### 3.9.4 FTP

Prinzipiell ist FTP eine Client/Server Anwendung zur Übertragung von Dateien. Auf dem Rechner von dem die Daten kommen muss ein Server-Prozess laufen, der für das Überprüfen der Authorisierung, sowie den Transfer der angeforderten files verantwortlich ist. Das Client-Programm, welches sich je nach Hardware sehr unterschiedlich präsentieren kann (Zeilenmodus, Windows-FTP), ist dafür verantwortlich, in welchem Format die Daten übertragen werden, prinzipiell ist der Server entweder auf Binary oder ASCII-Modus eingerichtet.

Im allgemeinen Sprachgebrauch hat sich die Bezeichnung FTP für anonymous FTP eingebürgert. Anonymous FTP arbeitet mit öffentlich bekannten Benutzern auf einem Server, diese heissen „anonymous“ und benutzen als Passwort meist die E-Mail Adresse des clients.

### 3.9.5 E-Mail

Die Applikation E-mail besteht im Prinzip aus den Funktionen Editieren einer Mail, Abspeichern am richtigen Ort, Aufrufen einer Schnittstelle zum Netzwerk, Übergeben der Nachricht und einer Funktion zum Entgegennehmen von Nachrichten.

Diese Funktionen werden vom sogenannten SMTP (Simple Mail Transfer Protokoll) [Pos-82] angeboten. Dieses Protokoll überträgt Nachrichten in Form von ASCII Zeichen.

Ein neues Protokoll ist die sogenannte *Multipurpose Internet Mail Extension (MIME)*. MIME sorgt dafür, dass in einen ASCII-Text eingebettete Binary-files (z.B. Word Files) dem Empfänger als Attachment erscheinen und dieser sie mit dem dazugehörigen Programm öffnen kann.

Falls ein persönlicher Rechner nicht ununterbrochen mit dem Internet verbunden ist, besteht die Notwendigkeit, eingehende mails auf einem Server zwischen zu speichern und bei Bedarf abzurufen. Hierzu ist das Post Office Protocoll (POP oder POP3=Version3) entwickelt worden. Dabei kann auf der Client-Seite (dem lokalen Rechner) eine Server unabhängige Mail-Software installiert sein, welche mails mit dem SMTP-Protokoll zum Server schickt und bei Bedarf mit POP3 neue mails vom Server holt.

## 3.10 Verteilte Systeme – LAN/WAN

Local Area Networks sind charakterisiert durch einen einfachen Netzaufbau auf allen Schichten. Es sind keine Router oder aufwendige Adressierungsmechanismen nötig, da eine sogenannte PEER to PEER Verbindung realisiert wird. Dies bedeutet, dass im Gegensatz zu WAN (Wide Area Network) meist eine ständige Kabelverbindung zwischen den Geräten besteht, welche nur für die Netzwerkverbindung genutzt wird und nicht zum Beispiel auch als Telefonleitung.

Häufige Einsatzgebiete von LANS:

- Druckernetzwerke,
- Lokales File- und Applicationsharing (Apple Share, Microsoft Netzwerk).

Wide Area Networks sind Verbindungen zwischen Computern, die geographisch weit entfernt stehen. Die Kabelverbindung besteht meist nur zum Zeitpunkt der Kommunikation und wird nach Übertragungsende wieder aufgehoben. Dabei wird auf länder-und weltweit bestehende Netze wie zum Beispiel das Telefonnetz zurückgegriffen. Dabei wird je nach Einsatzzweck eine Standleitung (verteilte Filialen

innerhalb einer Firma) oder eine Wählleitung per Modem oder ISDN-Router benutzt.

### 3.11 Verteilte Systeme – Arten von Netzwerken

Für LANs gibt es drei wichtige Aufbauarten (Topologien):

- Stern-Topologie,
- Ring-Topologie,
- Bus-Topologie.

Eng verknüpft mit diesen Topologien sind die LAN-Standards „Token Ring“ und „Ethernet“.

*Token Ring* basiert wie der Name bereits sagt auf einer Ring-Topologie. Charakterisiert wird diese Netzwerkverbindung durch das sogenannte Token, ein Berechtigungssymbol, welches anzeigt, welcher Rechner vorübergehend die Kontrolle über den Ring übernommen hat. Wenn Rechner A an Rechner B Daten versenden will, muss er warten, bis das Token in freiem Zustand ist. Dann versetzt er es in den Zustand belegt und hängt diesem Daten an. Jeder Rechner im Netzwerk erkennt das Token als belegt und überprüft, ob die Sendung für ihn bestimmt ist. Falls ja übernimmt er die Daten und gibt das Token wieder frei.

Der am weitesten verbreitete Standard *Ethernet*, ist durch den Standard IEEE802.3 oder Ethernet II festgelegt. Prinzipiell verwendet dieses Netz die Bus-Topologie, bei der die einzelnen Rechner an einen Kabelstrang angeschlossen werden. Der Zugriff auf den Bus (im vergl. zu rechnerinternen Bussen gibt es ja keinen expliziten Adress- oder Steuerbus) wird durch das sogenannte CSMA/CD (Carrier Sense Multiple Access with Collision Detection = CSMA/CD)-Verfahren geregelt. Dabei überwacht jede Station im Netz, ob bereits Daten übertragen werden (Carrier Sense) und schickt gegebenenfalls Daten ab. Trotz dieser Prüfung kann es zu Zusammenstößen zwischen Datenpaketen kommen. Falls zwei Rechner gleichzeitig prüfen und mit der Übertragung beginnen, wird dies als Kollision erkannt (Collision Detection) und die Übertragung abgebrochen.

Beim Ethernet stehen drei Verbindungsarten zur Verfügung:

Verbindungsart	Kabel	Entfernung zw. Stationene
Thin Ethernet	10Base-2 Koax	0.5 m – 185m
Thick Ethernet	10Base-5 Koax	2.5 m – 500m
Twisted Pair	Verdrillte Doppelader/ Telefonleitung	0.5 m – 100/185m (doppelt abgeschirmte Kabel)

Tabelle (T002eaiZ) Vergleich verschiedener Ethernet-Verbindungen

Bei Thin Ethernet werden die Rechner direkt miteinander verkabelt, d.h. falls neue Rechner ins Netz eingefügt werden, muss das Netz unterbrochen werden. Dies ist jedoch kein Problem, solange keine Anwendung in ein time-out läuft (möglichst kleiner als 2min), da durch das CSMA/CD-Protokoll die Störung erkannt und die Nachricht erneut gesendet wird.

Thick Ethernet schliesst die Rechner über Verbindungseinheiten (externe Transceiver) ans Netz – auch hier muss das Netz unterbrochen werden, um neue Rechner einzufügen.

Neuerdings setzt sich die Verbindung über Twisted Pair-Kabel immer mehr durch, da hierbei normale Telefonleitungen verwendet werden können. Diese sind normalerweise zwar etwas stör anfälliger, was jedoch im LAN mit Leitungslängen von wenigen Metern bis zu hundert Meter kein Problem darstellt. Ab einer grösseren Leitungslänge sollten abgeschirmte Kabel verwendet werden. Eine Übersicht über mögliche Verkabelungen ist aus Grafik [GER-94] S. 582 zu ersehen.

Mit allen Verbindungsarten ist eine Übertragungsrate von bis zu 10Mbit/s erreichbar.

#### Beispiel:

An der ETH werden alle drei möglichen Verbindungsarten sowie Bus- und Ethernet Stern Topologien eingesetzt. Im CAETH-Zentrum sind die CAD Workstations mit Thin Ethernet verbunden. Dies führte in der Vergangenheit bereits mehrfach zu schwer ortbarem Ausfall der ganzen Anlage, weil Studenten ein einzelnes Kabel entfernten. Die Büros im CLA-Gebäude sind über UTP Telefonleitungen sternförmig von einzelnen Repeatern ans Ethernet angeschlossen. In einigen Büros sind über eine einzige UTP-Buchse mehrere Apple Macintosh Rechner

durch Thick Ethernet mit externen Transceivern miteinander verbunden.

### 3.12 Verteilte Systeme – Client/Server

Das Client-Server Modell für verteilte Rechneranwendungen ist mit dem Siegeszug von Unix zu einer weit verbreiteten Technologie geworden.

Im Prinzip kann es auf jede Rechneranwendung übertragen werden, in der ein Prozess eine Aufgabe an einen anderen Prozess zur Ausführung übergibt. Hierbei ist es jedoch nicht zwingend Voraussetzung, dass beide Prozesse auf unterschiedlichen Rechnern laufen. Ein typisches Beispiel ist das X Window System (siehe Fenstersysteme), welches durchaus auch als lokale Fensterapplikation betrieben werden kann.

#### Definition Client/Server [HAN-96]:

.. eine kooperative Informationsverarbeitung, bei der die Aufgaben zwischen Programmen auf verbundenen Rechnern verteilt werden. In einem solchen Verbund können Rechner aller Art zusammenarbeiten. Server (=Dienstleister) bieten über das Netz Dienste an, Clients (=Kunden) fordern dies bei Bedarf an.

Im Prinzip arbeitet die Client/Server Kommunikation immer auf einem Remote Procedure Call Konzept. Der Server hält eine Anzahl von Programmen (Diensten) bereit, die er auf Anfrage des Clients mit dessen Parametern startet und nach dessen Beendigung die Parameter zurückliefert.

In der Praxis werden oft eine Anzahl von Server-Prozessen auch auf einem Hardware-Server - oft einem Rechner mit grösserer Leistungsfähigkeit - zusammengefasst. Je nach Art der Server Prozesse spricht man dann z.B von einem File-Server (überträgt auf Anfrage Files), Datenbank-Server (Enthält die RPC-Serverprozesse für Datenbankfragen), World Wide Web Server (enthält den WWW-Server), Lizenz-Server (prüft die Lizenzberechtigungen), etc.. Meistens muss eine grössere Maschine mehrere Arten von Server-Prozessen betreiben. Dies führt manchmal dazu, dass eine Client/Server-Umgebung zu einer Grossrechnerumgebung mit einem Flaschenhals beim Server mutiert.



### 3.13 Verteilte Systeme – Fenstersysteme

Fenstersysteme sind eine typische Anwendung des Client/Server Prinzips. Ein Server überprüft zum Beispiel dauernd Anfragen auf Mausbewegungen und stellt bei Bedarf die gewünschte Fenster-Grafikfunktion zur Verfügung (Fenster schliessen, verschieben etc.)

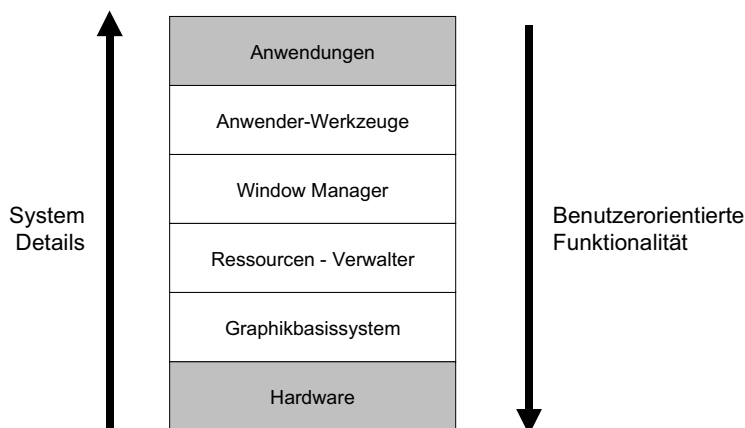


Bild (B020eaiZ) Schichtenaufbau von Fenstersystemen

#### Beispiel

Das X-Window-System erlaubt eine grafische Bildschirminteraktion zu einem Prozess vollkommen hardwareunabhängig zu betreiben. So stellt beispielsweise XSOFTWARE die komplette Fensterfunktionalität nach Bild B020eaiZ auf einem PC zur Verfügung. Beim Aufstarten wird der X-Server (!) initialisiert. Dieser benutzt die PC-spezifischen Graphikbasissysteme (deswegen sehen Fenstersysteme auf Intel-PCs anders aus, als auf einem Macintosh) zum Handling der Client-Anfragen. Wird das Programm geschlossen, obwohl eine auf einem remote Host laufende Applikation noch lokal offene Fenster hat, lautet die Warnung: „do you really want to kill the client(!)-application?“.

Das X-Window System ist eine Applikation, die nicht RPC-basiert, sondern auf der Socket-Kommunikation aufgesetzt ist. Ein zentraler Prozess stellt der Event-Handler dar, welcher zyklisch Events wie z.B. Mausbewegungen oder Tastatureingaben abfragt, und in Warteschlangen verwaltet sowie die jeweils zugeordneten Aktionen triggert.

### 3.14 Verteilte Systeme – NFS und NIS

Das NFS und NIS sind Applikationen auf der obersten ISO/OSI Schicht, die auf darunterliegenden Diensten aufsetzen. Deswegen gibt es sehr unterschiedliche Arten von NFS und NIS-Applikationen je nach Netzwerkcharakteristik.

Im Gegensatz zu verteilten Applikationen mit Datenaustausch auf Applikationsebene ist das NFS eine Schicht höher angeordnet und verwendet eine Kommunikation auf File-Ebene. Das Network-File-System erlaubt einen hardwareunabhängigen Zugriff auf ein verteiltes Filesystem.

Zur Sicherstellung der Zugriffsrechte bzw. Zuordnung der Files zur Hardware wird das Network Information-System (NIS) verwendet, welches auf jedem Rechner einen Server betreibt, der die File-Informationen über das Netz repliziert und konsistent hält. Das NIS erlaubt, ähnlich wie das Domain Name Server System (DNS), eine Zuordnung von Rechnernamen zu Adressen, jedoch nur in lokalen Netzwerken. Die Kommunikation zwischen den sogenannten NIS Slave Servern auf jeder Maschine ist wiederum RPC basiert.

### 3.15 Verteilte Systeme – Datenbanksysteme in einer verteilten Umgebung

Die interne Technologie und Modellbildung relationaler Datenbankmanagementsysteme (DBMS) wird in Kapitel 3 näher erläutert. In diesem Kapitel wird lediglich auf die Kommunikationsschnittstellen und deren Einsatz in verteilter Umgebung eingegangen werden.

Der interne Aufbau eines Datenbankmanagementsystemes (DBMS) -Schicht 1 in Bild B021eaiZ- besagt noch nichts darüber wie die Zugriffsmechanismen auf die Daten von aussen implementiert sind. Während Desktop-Systeme ein komplexes grafisches Interface auf Basis von Sprachen der vierten Generation (4GL), für Datendefinitionen und Datenmanipulationen zur Verfügung stellen, hat sich bei den grossen relationalen DBMS eine Schnittstelle auf SQL-Basis durchgesetzt. Betrachtet man die Vier-Schichten Architektur nach Bild B021eaiZ so ist die SQL-Schnittstelle auf Schicht 1 anzusiedeln, während Desktop-Systeme durchaus alle vier Schichten umfassen.

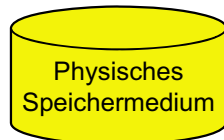
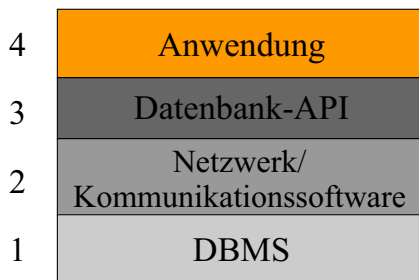


Bild (B021eaiZ) Vier-Schichten Architektur von Datenbanksystemen

Typische Funktionalitäten eines DBMS (Schicht 1) sind SQL-Parser, Optimierer, Ausführungsmodul und Dienste für die Datenverwaltung (Sicherheit, Transaktion, Recovery), welche bei Desktop-Systemen meist nur eingeschränkt implementiert sind.

Die Netzwerk und Kommunikationssoftware (Schicht 2) gewährleistet eine physische Trennung von Anwendung und Datenbank (Bsp. Client-Server Modell). Sie ist heutzutage meist TCP/IP basiert. Die Datenbank-API (Schicht 3) hat hauptsächlich die Aufgabe von einer Anwendung kommende Anfragen in datenbankspezifische Systemaufrufe umzusetzen. Eine Möglichkeit besteht hierbei in der Anwendung des sogenannten embedded SQL. Hier stellt sich die Frage wie man diese Abfragen in einer jeweiligen Programmiersprache (Hostsprache) einbettet und die Anweisungen für den Hostcompiler und das DBMS Laufzeitsystem verständlich macht. Hierzu wird der Quellcode der Hostsprache mit eingebetteten SQL-Aufrufen mit einem DBMS spezifischen Precompiler vorcompiliert und in Aufrufe des DBMS-Laufzeitcodes übersetzt.

Eine andere Möglichkeit für Anwendungsprogrammierung von DBMS besteht in den sogenannten Call Level Interfaces (CLI). Diese bestehen aus Funktionsaufrufen des DBMS in einer 3GL-Programmiersprache die vom Anwendungsprogrammierer direkt verwendet werden können ohne den Quellcode precompilieren zu müssen.

Wesentliche Unterschiede zwischen CLI und embedded SQL (siehe [Gei-92]):

- Keine Unterscheidung von Datenbank-Variablen und Variablen der Programmiersprache
- Klare Trennung von DBMS-Aufrufen im CLI und Anwendungsprogramm
- Kein Precompilieren nötig, sondern lediglich Linken der DBMS-Bibliotheken
- Einfacheres Debugging

In Kapitel Datenbankschnittstellen wird ODBC als Standard eines Call Level Interfaces vorgestellt.

### 3.16 WWW

Das World Wide Web ist im Prinzip nichts anderes, als eine weltweit verteilte Client-Server-Kommunikation mit vereinbarten Definitionen wie:

- Standardisiertes Protokoll zur Informationsübertragung (HTTP),
- Default Port für den Server (Port Nr. 80),
- Default Applikation, die bei einer Anfrage gestartet wird (Transfer einer bestimmten Seite durch HTTP-Protokoll)
- Sprache zur Textdarstellung (HTML),
- Einheitliche Darstellung von Suchpfaden (URL).

Das Hypertext Transfer Protocol sorgt dafür, dass auf Anfrage an einen Server ein ASCII-Textfile (Hypertext) aus dem Server an den Client gesendet wird.

In diesem Dokument sind Steuerzeichen (tags) zur Formatierung des Dokuments enthalten. Diese Steuerzeichen ergeben insgesamt die Hypertext Markup Language (HTML). HTML ist eine Teilmenge von SGML (Standard Generalized Markup Language), welche ursprünglich für den Satz (z.B. Zeitungen) entwickelt wurde.

#### Beispiel:

Hallo in HTML: <I><B>H</B>allo</I>

Die in eckigen Klammern stehenden Zeichen werden „Tags“ genannt und geben Informationen darüber wie die dazwischenstehenden ASCII-Zeichen zu behandeln sind.

Bei der Adressierung des HTML-Dokumentes wird ein System verwendet, welches Uniform Resource Locator (URL) genannt wird. Dieser URL wird zusammengesetzt aus einer base URL und dem File-Namen. Dies bedeutet, dass sämtliche Verzeichnisse, die unterhalb der

Base-URL (Document Root) liegen theoretisch über diese erreicht werden können.

### 3.16.1 WWW-Server

Das WWW funktioniert streng nach dem Client/Server Prinzip, wobei die Unterscheidung zwischen Server und Client so streng ist, dass hierbei vollkommen unterschiedliche Produkte auf der Client- und Server-Seite angeboten werden.

Der WWW-Server hat hierbei prinzipiell allein die Aufgabe, die auf seinem abgefragten Port eingehenden Anfrageparameter des Clients (URL) auf seine Filestruktur (Document-Root) umzusetzen und das File dem Client zu schicken.

Das heisst es läuft ein RPC Server Prozess auf dem Standardport Nr. 80 (es können auch andere ports definiert werden), der als Parameter-Input eine URL bekommt und daraufhin nur ein einziges Programm aufruft, nämlich das, welches ein der URL entsprechendes File zum Client transferiert, sobald es sich um ein bekanntes File-Format handelt (HTML, VRML, GIF etc.).

### 3.16.2 WWW-Client

Da prinzipiell sämtliche Informationen vom Server über das gemeinsame HTTP-Protokoll übertragen wird und vom Client in der maschinentypischen Weise dargestellt werden, ist es für den Ersteller einer HTML-Seite vollkommen gleichgültig, auf welcher Maschine seine Information betrachtet wird.

Folgende Clients (auch Browser genannt) sind u.a. erhältlich:

- Netscape,
- Microsoft Internet Explorer,
- NCSA Mosaic,
- HOT Java

Der zuletztgenannte Browser der Firma SUN setzt auf der Programmiersprache Java auf, die wir im folgenden Kapitel etwas näher betrachten wollen.

### 3.16.3 XML

XML ist die Abkürzung für Extensible Markup Language und eine Teilmenge von SGML. XML wurde vom World Wide Web Consortium für den Einsatz im WWW entwickelt.

Der prinzipielle Aufbau ähnelt stark html jedoch mit dem unterschied, dass nicht nur ein begrenzter Umfang von Tags (logische und semantische) zur Verfügung steht, sondern sogenannte Elementtypen

pen. Diese können auch strukturiert werden, wie aus dem folgenden Beispiel zu ersehen ist:

```
<buch>
  <titel>The Tadpoles of Texarkana</titel>
  <autor>Herbert J. Snoog</autor>
  <publisher>Skudsville Press</publisher>
</buch>
```

Bei XML erfolgt eine Trennung zwischen dem Inhalt eines Dokuments und dessen Aussehen (Design), sogenannte Style Sheets erfüllen die Designaufgabe.

Die Art und Anordnung der Elemente wird über die DTD (Document Type Definition) festgelegt.

Mittels eines HTML templates, welches die Darstellungsweise der Elementtypen definiert können einfach aus XML html-Dateien erzeugt werden.

#### 3.16.4 Java

Wie wir gesehen haben, kann ein WWW-Server prinzipiell alle möglichen Arten von Files übertragen. Über das MIME-Protokoll wird festgelegt, welche lokale Applikation gestartet wird, falls ein gewisses File Format beim Browser ankommt. Ist die Applikation lokal nicht vorhanden, kann das Client-System auch nichts damit anfangen.

Java ist prinzipiell nichts anderes wie eine objektorientierte Programmiersprache (ähnlich C++). Die Besonderheit bei Java besteht in der Möglichkeit sogenannte *Applets* zu erstellen. Ein Applet ist unabhängig vom lokalen Betriebssystem für sich alleine lauffähig. Die ist Möglich weil ein Applet nicht nur das kompilierte Programm, sondern auch Software zum simulieren eines virtuellen Prozessors enthält. Beim Aufrufen eines Applets über das WWW interpretiert der lokale (JAVA-fähige) Browser das Programm und simuliert zur Laufzeit auf der aktuellen Hardware eine Java-Maschine. Dies bedeutet, dass es vollkommen gleichgültig auf welcher CPU und auf welchem Betriebssystem ein Java-Programm erstellt wird, durch den Java-Browser wird sichergestellt, dass das Programm sämtliche lokalen Ressourcen erhält und damit zu einem ablauffähigen Prozess wird. Hierin sind auch die Sicherheitsbedenken gegen Java-Applets begründet, da die Java-Maschine Zugriff auf alle lokalen Ressourcen des Client Rechners hat. Neuere Browser verfügen deswegen über speziell konfigurierbare Java-Sicherheitsoptionen.

Eine ander Technologie sind die sogenannten *Servlets*. Diese sind im Prinzip nichts anderes als Applets, die beim Aufrufen jedoch nicht

auf den client übertragen sondern direkt auf dem Server gestartet werden.

### 3.16.5 Push-Technologie

Die Push Technologie realisiert eigentlich einen aktiven WWW-Server, der unter bestimmten Bedingungen Aktionen auf einem Client ausführt. D.h. solange ein Rechner mit dem Internet verbunden ist, kontaktiert dessen Client-Programm regelmässig einen Server. Falls sich auf dem vom Anwender definierten Kanal eine Änderung ergibt, wird diese automatisch auf den Client übertragen und dort zur Anzeige gebracht.

Genutzt werden kann diese Technologie z.B. als Nachrichtenkanal, für Börsendaten sowie als firmeninternes Kommunikationsmedium. Im Gegensatz zu E-Mail erlaubt diese Technologie die Nutzung von WWW-Browsern für Broadcast-Messages, sofortige Benachrichtigung bei ungewöhnlichen Ereignissen etc.

### 3.16.6 Network-Computer, NetPC

Während früher zu Zeiten der grossen Mainframes der Einzelbenutzer eigentlich keinen Zugriff zum System hatte, aber auch nicht brauchte (die Systeme wurden von Systemadministratoren betreut), haben sich die System- und Software-Verwaltungsaufgaben fast vollständig auf den Endbenutzer verlagert. Dies hat zur Folge, dass durch unsachgemässes Arbeiten sowie durch die notwendige Software-Installation an jedem einzelnen Arbeitsplatz ein PC ohne Anschaffungskosten ca. 12.000 CHF/Jahr reine Unterhalts und Wartungskosten verursacht (Quelle: Andersen Consulting). Man versucht deswegen, wieder mehr zu zentralem System- und Software-Management zurückzukehren. Aus diesen Bestrebungen entstanden unter Nutzung moderner Kommunikationsmitteln die sogenannten Network-Computer.

Die Basis für Network-Computer sind eine Programmiersprache, wie z.B. Java, welche auf einer beliebigen Hardware einen virtuellen Prozessor emulieren können, der ein JAVA Programm verarbeiten kann. Dieser Prozessor ist prinzipiell auch einfach in der Lage, über das Internet zur Laufzeit lokal nicht vorhandene Programmmodule zu laden und abzuarbeiten.

Nach diesem Prinzip funktionieren dann auch die Network-Computer, die lokal keine Festplatte oder andere peripheren Speichermedien, ja sogar nur ein kleines Betriebssystem mit Internet Kern-Funktionalität besitzt. Sämtliche Software wird bei Bedarf von einem Server zentral bezogen.

### 3.17 Systemschnittstellen

Für eine Systemintegration zwischen zwei Datenbanksystemen lassen sich auf unterschiedlichen Ebenen Schnittstellen realisieren. Voraussetzung ist dass die einzelnen Ebenen in sich abgeschlossen und die Schnittstellen offen und dokumentiert sind.

Für einzelne Schichten haben sich auch Standards etabliert, die von immer mehr Systemen angeboten werden. Innerhalb dieser wird die Schnittstellenproblematik von der syntaktischen Ebene auf eine semantische bzw. einer durch Geschäftsprozesse gegebene Ebene verlagert.

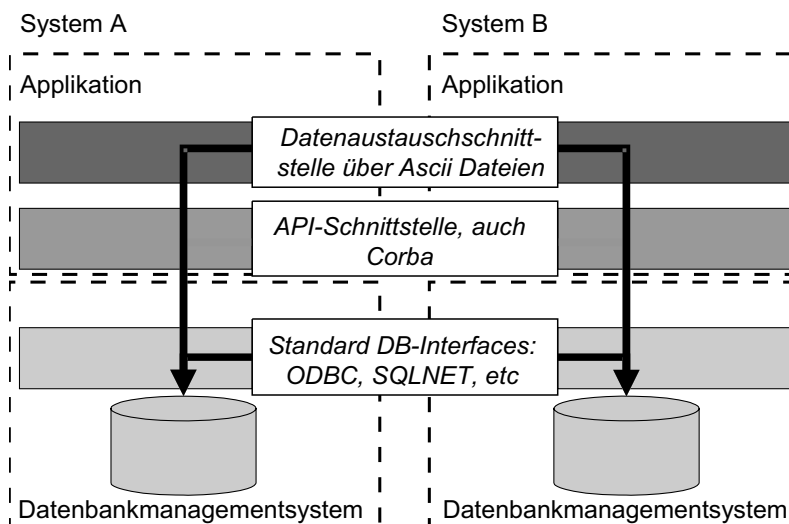


Bild (B022eaiZ) Systemschnittstellen

Auf der obersten Ebene liegen die Schnittstellen, die keine Nutzung der Applikation erfordern. Hier ist es lediglich notwendig, dass das auszutauschende Format der (ASCII)-Datei und der Zeitpunkt definiert werden.

Auf der API (Appication Programming Interface) –Ebene wird direkt-beispielsweise über Remote Procedure Call- eine Routine des fremden Systemes gestartet.

Die Kommunikation zwischen Datenbankmanagementsystemen umgeht die Applikation und repliziert direkt Daten aus Tabellen des einen Systemes in das andere. Dies ist nur möglich wenn die Systeme keine Konsistenzsicherungsmechnismen auf Applikationsebene implementiert haben.



### 3.17.1 Datenbankschnittstellen

An dieser Stelle soll das in der Microsoft Welt weitverbreitete standardisierte Call-Level-Interface ODBC (Open Data Base Connectivity) erwähnt werden. ODBC basiert auf der CLI-Spezifikation der X/Open SQL Access Group und stellt somit eine datenbankunabhängige Programmierschnittstelle zur Verfügung.

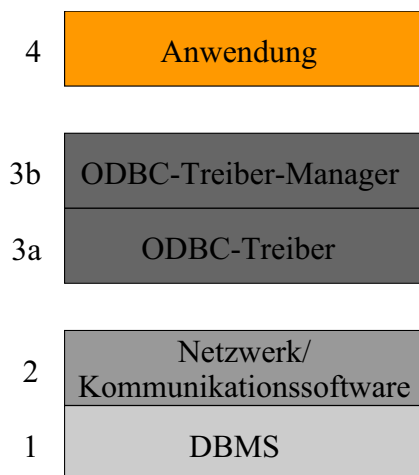


Bild (B023eaiZ) Standard-ODBC Architektur

Die datenbankneutrale Schnittstelle ODBC nutzt die Abgeschlossenheit (Encapsulation) auf der DBMS-Seite, die bewirkt, dass der DBMS-Server kein Wissen über Funktionalitäten auf der Client Seite benötigt, sondern lediglich Funktionen auf Basis eines standardisierten Datenprotokolls für die Netzwerkschicht zur Verfügung stellt. Unter der Vorbedingung hierfür dass eine Mehrschichtenstruktur ohne Seiteneffekte zwischen den Schichten vorliegt, ersetzt ODBC dann lediglich die Schicht 3 (Bild B022eaiZ Datenbank API) der jeweiligen Datenbank-Zugriffsarchitektur. Um eine systemneutrale Schnittstelle zu schaffen wird ODBC noch einmal aufgeteilt in Treiber-Manager und Treiber (Bild B023eaiZ). Der Treiber-Manager verwaltet die Interaktionen zwischen Anwendungsprogrammen und Treibern. Der Treiber verarbeitet ODBC-Funktionsaufrufe, setzt SQL Anfragen für spezifische Datenquellen ab und empfängt die Ergebnisse.

Mittels ODBC lassen sich somit auf Datenbankebene unter Umgehung der Applikationen Systemintegrationen durchführen.

### 3.17.2 Corba

Mit Corba wurde die Möglichkeit geschaffen, Objekte in verteilter Umgebung, basierend auf Standardnetzwerkprotokollen global verfügbar zu machen. Der Standard wurde von der Object Management Group geschaffen. Hierbei können die Objekte in völlig heterogenen Strukturen existieren, in relationalen Datenbanken sowie in Methoden in diversen Programmiersprachen und unterschiedlichster Hardware. Die Objekte kommunizieren über eine Kommunikationsschnittstelle in unterschiedlichen Adressräumen, auf unterschiedlicher Hardware und in heterogener Betriebssystemumgebung.

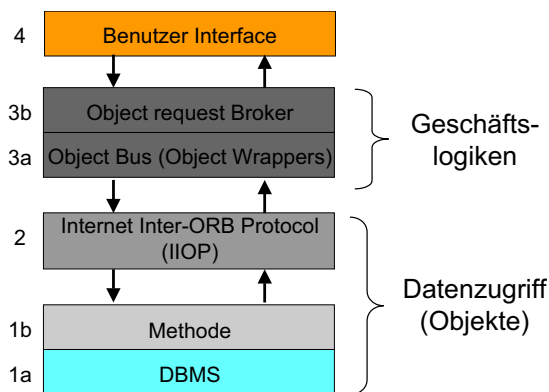


Bild (B024eaiZ) Corba Architektur

Möglich wird dies, indem für diverse Programmiersprachen mittels einer *Interface Definition Language* (IDL) Standardinterfaces zur Verfügung gestellt werden. Mittels dieser kann eine existierende Methode „gewrapped“ werden. Solchermassen eingebettete Objekte stehen auf dem sogenannten Object Bus anderen Anwendungen zur Verfügung und kommunizieren über definierte Schnittstellen (ähnlich Sockets) miteinander. Es können nur Objekte miteinander kommunizieren, die als Kommunikationsobjekte auch mittels Interface Definition Language auf dem Object Bus präsent sind. Der Object Request Broker identifiziert über die sogenannte „object reference“ das für die Anfrage zuständige Objekt. Die Object Reference ist durch semantische Regeln gegeben. In Bild B024eaiZ ist die Kommunikation einer Anwendung mit einem Corba-Objekt via dem TCP/IP basierenden Internet Inter-ORB Protocol zu sehen [LeBaSi-98].

### 3.17.3 XML

XML setzt sich als Standard für die Definition der Dateiformate der Schnittstellen zwischen den Systemen heute zunehmend durch. Viele System bieten Möglichkeiten XML zu generieren und zu lesen. Häufig werden sogar Prozeduren zum Lesen und Schreiben von Datenbankinhalten von den einzelnen Systemen angeboten.

Die Entwicklung und der Einsatz von branchenübergreifenden DTDs (Document Type Definition) erleichtern Organisationen bereits heute den Austausch und die Verwaltung von Daten.

## 3.18 Industrielle Nutzung dieser Technologien – Geschwindigkeit

Ein oft diskutierter Punkt verteilter Netzwerk-Kommunikation ist die Geschwindigkeit der Datenübermittlung (siehe auch vorheriges Kapitel).

Beim Produktdatenaustausch gibt es verschiedene Größenordnungen der auszutauschenden Daten. Diese gehen von wenigen kByte für einfache „Metadaten“ bis zu mehreren MByte für ganze CAD-Modelle.

### Beispiel

Angenommen ein auszutauschendes 3D-CAD-Modell habe 7,2 Mbyte und es liegt eine Übertragung ohne Steuersignale auf direktem Weg zwischen zwei 28800 Baud Modems vor, werden

$28800 \text{ bit/s} = 3600 \text{ Byte/s}$

übertragen. D.h. 7.200.000 Byte benötigen  $2000\text{s} \Leftrightarrow 33,3 \text{ Minuten}$ .

Dieses einfache Beispiel zeigt, dass in einer weit verteilten Entwicklungsumgebung nicht einfach sämtliche Dokumente auf einem Server zentral gelagert und bei Bedarf per FTP transferiert werden können. Hier müssen aus Performance-Gründen zur Konsistenzsicherung möglichst nur Referenzen der beteiligten Dokumente ausgetauscht werden. Bei Bedarf werden dann unter Nutzen von Viewing-Technologien wie VRML (Virtual Reality Modelling Language), Tiff-Viewer und Web-Interfaces zur PLM-Datenbank eine komprimierte Sicht auf die aktuellen Produktdaten ermöglicht.

### 3.19 Industrielle Nutzung dieser Technologien – Kommunikation innerhalb eines verteilten PLM-Systems

Da es eine der Grundaufgaben von PLM-Systemen ist, eine Integration verschiedener Software-Systeme auf unterschiedlichen Hardware-Plattformen herzustellen, kommen hierbei Kommunikationstechnologien auf allen Schichten des ISO/OSI-Protokolls zum Einsatz.

Im allgemeinen sind PLM-Systeme auch heute noch durch die Verwendung einer zentralen relationalen oder objektorientierten Datenbank als Integrationsplattform gekennzeichnet. Hierbei lassen sich zwei charakteristische Client/Server-Technologien unterscheiden:

- Datenbank-Server ↔ PLM-Client Kommunikation
- Datenbank-Server ↔ PLM-Server ↔ PLM-Client Kommunikation

#### Datenbank-Server ↔ PLM-Client Kommunikation

Ein Teil der Systeme verwendet die interne Datenbanktechnologien zur Kommunikation zwischen Client-System und Server. Hierbei ist auf jedem Rechner eine PLM-Software als Client dafür verantwortlich, die jeweiligen Datenbankbefehle an den Datenbankserverprozess abzusetzen.

- Vorteile
  - ein Overhead durch zusätzlichen PLM-Server-Prozess
  - einfaches Update durch Austausch der Client-Software
  - eher zuverlässige Kommunikation (weil Standard-)
- Nachteile
  - Abhängigkeit von der Systemsoftware des DB-Herstellers
  - eine zusätzliche PLM-Funktionalität auf dem Server

#### Datenbank-Server ↔ PLM-Server ↔ PLM-Client Kommunikation

Eine andere Möglichkeit stellt der Aufbau einer eigenen PLM-Client/Server Kommunikation dar. Hier existiert zusätzlich zum Datenbank-Server Prozess noch ein eigener PLM-Server-Prozess. Dieser wartet auf Anfragen seiner PLM-Clients und setzt danach Datenmanipulationsbefehle an den DB-Server ab.

- Vorteile
  - Abhängigkeit von der Datenbanktechnologie
  - zusätzliche PLM-spezifische Kommunikationstechnologien implementierbar
- Nachteile:
  - umständlichere Kommunikation mit dem Server

### 3.20 Industrielle Nutzung dieser Technologien – Kommunikation zwischen weltweit verteilten Systemen

Anhand eines Beispiels aus der Praxis soll hier die durch alle ISO/OSI-Schichten gehende Problematik weltweit verteilter Entwicklungsumgebungen erläutert werden.

#### Beispiel

Eine Firma Y mit Sitz in der Schweiz arbeitet mit mehreren weltweit verteilten Institutionen sehr eng an der Entwicklung einer neuen Produktlinie zusammen. Da der Entwicklungszyklus sehr schnell voranschreitet und alle Partner ständig mit den neuesten Informationen versorgt werden müssen, bzw. diese auch schnell verbreiten müssen, kommt der Kommunikation unter Nutzung des Internet eine sehr wichtige Bedeutung zu. Die Firma Y besitzt zwei Möglichkeiten für den Zugang zum Internet:

- Über analoge Modems mit einer Übertragungsgeschwindigkeit von 28800 Baud zu einem kommerziellen Internet-Provider.
- Im Rahmen eines Forschungsprojektes zur Entwicklung der neuen Maschinengeneration besteht die Möglichkeit über PPP den Dial-In-Service der ETH zu nutzen. Hierbei wird auch der analoge Dial-In mit 28800 Baud genutzt.

Die ETH-Zürich ist über Switch (=Provider) mit einer 2 Mbps (= 2 MBaud)-Leitung mit dem Internet verbunden. Firma XY in Deutschland ist ebenfalls mit einem 28800 bps-Modem mit dem ETH-Dial-In verbunden.

Die Firma Z in China sagt, sie besitzt einen „eigenen Internet-Knoten“.

Über die beschriebenen Kommunikationskanäle werden CAD-Modelle, Word-Files zur Dokumentation, Power-Point Präsentationsunterlagen sowie Software für die Maschinensteuerung ausgetauscht. Das PLM-System an der ETH verwaltet den aktuellen Stand der Produktdokumentation und den Partnern wird über einen Intranet-WWW-Server das Viewing auf den aktuellen Datenbestand ermöglicht.

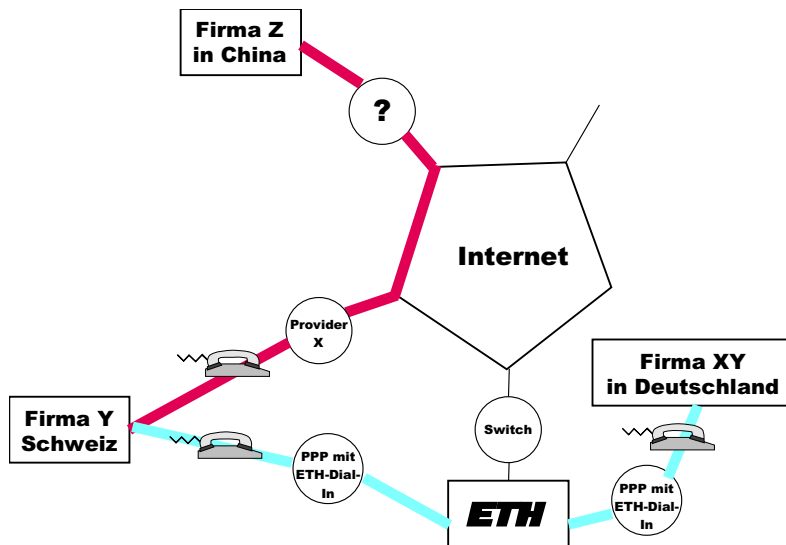


Bild (B025eaiZ) Weltweite Kommunikation mit Modem und Internet

Bei der Kommunikation mit dem chinesischen Partner traten schwere Laufzeitprobleme auf, die teilweise aufgrund von Times-Outs sogar zu einem Abbruch der Verbindung geführt haben. Dies war insbesondere verwunderlich, da hierbei nur relativ kleine Datenmengen im Verhältnis zur sonstigen Kommunikation auftraten. Der Verdacht lag deswegen nahe, dass das Problem in China liegt.

Was bedeutet die Aussage dieser Firma: „Wir haben einen eigenen Internet Knoten?“ Prinzipiell ist damit in der Umgangssprache lediglich gemeint, dass sie ein sogenanntes Class B-Netz betreibt, d.h. sie hat einen Domain-Namen im Internet. Weiter ist jedoch nichts über die Bandbreite ihres Anschlusses ans „Internet“ gesagt, d.h. es besteht durchaus die Möglichkeit, dass hier eine sehr langsame Verbindung vorliegt.

Ein Versuch über den PPP-Anschluss der ETH mit den Projektpartnern in China zu kommunizieren ergab eine erstaunlich schnelle Verbindung, d.h. das Problem liegt beim Provider der Schweizer Firma. Diese besitzt zwar einen 256kBit/s Gateway zum Internet, jedoch zehn 28800 Baud Modems. D.h falls 9 Modems benutzt werden, ist die Kapazität für die Übertragung ausgelastet. Daraus folgt, dass ein weiterer Teilnehmer in einen Wartezyklus gerät.

### 3.21 Industrielle Nutzung dieser Technologien – Kosten

Die Kosten die für einen Internet-Anschluss zu kalkulieren sind, belaufen sich im allgemeinen auf unter 100 CHF. An Hardware sind folgende Komponenten nötig:

- Netzwerkkarte ca. 500.- CHF.
- Modem ca. 500.- CHF.

Der Zugang zum Netz läuft in Europa meist über einen Provider, welcher ein Modem zur Internet-Verbindung zur Verfügung stellt. Hierbei gibt es extreme Unterschiede in der Leistungsfähigkeit, die von der Übertragungsgeschwindigkeit der empfangenden Modems sowie der Bandbreite des Internet-Anschlusses abhängt. Hier variieren die Preise von 20 CHF/Monat (von vielen Anbietern wird dieser Anschluss mittlerweile gratis angeboten) bis zu den Kosten der ETH, die ihren Zugang mit Switch mit 1,5 Millionen CHF/ Jahr bezahlt. Dafür hat die ETH einen direkten Zugang zu einer 2Mbit/s-Verbindung.

### 3.22 Industrielle Nutzung dieser Technologien – Sicherheit

Bei der weltweiten Koordination firmeninterner bzw. geheimer Daten taucht zwangsläufig die Frage der Sicherheit vor Spionage und Sabotage von firmeninternem Know-How auf. So vielschichtig wie die Kommunikation sind dabei auch die Einbruchsmöglichkeiten.

Denkbar ist ein physisches Kopieren der über die Leitung gehenden Daten durch eine Induktionsschleife, ein Einbruch auf Protokollebene durch Anzapfen von Internet-Gateways o.ä. sowie ein illegale Nutzung auf Applikationsebene durch falsche oder gestohlene User-Codes. Entsprechend sind auch die Sicherheitsmechanismen je nach Schicht unterschiedlich.

Eine relativ hohe Sicherheit gewähren Verschlüsselungsalgorithmen auf Zeichenebene. Hierbei ist das Knacken des Codes zwar nur eine Zeitfrage jedoch reicht dies in der Praxis vollkommen aus. Ein Konzept welches basierend auf der IP-Schicht einen Sicherheitsmechanismus liefert, sind die sogenannten *Firewalls (Brandmauern)*.

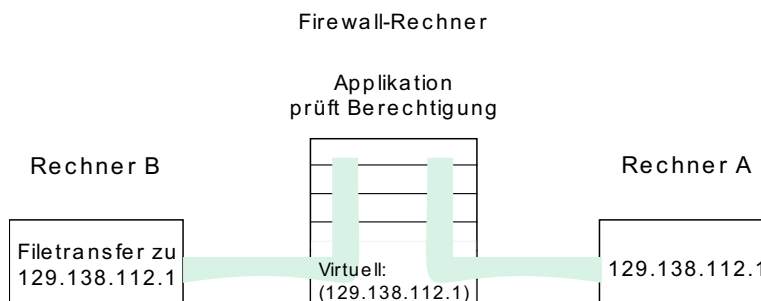


Bild (B026eaiZ) Funktion eines Firewall

Von aussen gesehen arbeiten Firewalls wie Router, bzw. Gateways, jedoch werden intern die Pakete nicht einfach weitergereicht, sondern durchlaufen innerhalb des Firewalls mehrere Netzwerkschichten, bis zu einer Applikation (Bild B026eaiZ). Diese Applikation prüft meist nur ob die Zieladresse eine Berechtigung für das nächste Netz besitzt.



## 4 Zusammenfassung

Neben den CAx-Systemen zur Produktdatenerstellung und dem PLM-System zur Produktdatenverwaltung finden weitere Software-Tools Anwendung zur Unterstützung von Unternehmensprozessen:

- Enterprise Resource Planning (ERP)
- Supply Chain Management (SCM)
- Customer Relationship Management (CRM)
- Spezielle Konfigurationstools
- Visualisierungs- und Virtual Reality-Tools (VR)

Die Integration obiger jeweils autonomer Applikationen zu einer gemeinsamen Applikation ist unter dem Begriff „Enterprise Application Integration“ bekannt. In vielen Fällen geschieht diese Integration über ein zentrales PLM-System.

Bei verteilter, rechnerintegrierter Produktentwicklung ist die Kommunikation zwischen heterogenen, geographisch verteilten Applikationen zentral.

Informationen werden in digitaler Form über physikalische Verbindungen ausgetauscht. Dazu benötigt es eine entsprechende Vorbereitung der Signale, die Festlegung einer Übertragungsart und einer Verbindungstopologie.

Kommunikationsprotokolle legen die Spielregeln des Datenaustausches fest. Sie trennen Kontrollinformationen von der eigentlichen Datenübertragung, synchronisieren die Protokollendpunkte und sind für ein effizientes Management der Ressourcen besorgt.

Applikationen setzen auf eine funktionierende physikalische Verbindung und in der Regel mehrere Kommunikationsprotokolle auf. Dabei laufen mehrere Prozess ab. Ein Prozess wird als ein Programm mit allen aktuellen Laufparametern sowie allen Betriebsmitteln ausser eventuell dem Prozessor definiert.

Für die Auswahl von Kommunikationssystemen sind im Allgemeinen folgende Kriterien von Bedeutung:

- Performance
- Kosten
- Sicherheit

Je nach Einsatz und Anwendung sind diese Kriterien verschieden gewichtet.

#### Verständisfrage 1

Warum erlauben Glasfaserkabel eine höhere Übertragungsgeschwindigkeit als konventionelle Kupferleitungen?

#### Verständisfrage 2

Warum werden bei rein serieller Datenübertragung mit 2 Mhz nicht 2 Millionen bit/sec übertragen?

#### Verständisfrage 3

Wieso benötigt man bei einer elektronischen Kommunikation zwischen zwei Datenendgeräten nur eine Stromquelle?

#### Verständisfrage 4

Worin unterscheidet sich ein SCSI-Bus von einem konventionellen Bussystem?

#### Verständisfrage 5

Wieso ist SCSI kein Echtzeitprotokoll?

#### Verständisfrage 6

Warum kann man Ethernet als Bussystem bezeichnen?

#### Verständisfrage 7

Worin könnte das Problem bestehen, dass obwohl bereits Prozessoren mit 600 Mhz Taktfrequenz existieren, noch keine so hoch getakteten PCs ausgeliefert werden?

#### Verständisfrage 8

Wieso wird für die Telnet-Kommunikation das TCP-Protokoll benutzt?

#### Verständisfrage 9

Wieviele Prozesse können gleichzeitig auf einem Rechner laufen?

#### Verständisfrage 10

Was ist die Grunfunktionalität eines WWW-Servers

**Verständisfrage 11**

Worin besteht der Hauptunterschied einer ISDN-Verbindung und einer konventionellen Telefonverbindung?

**Verständisfrage 12**

Wie kann man einen WWW-Server aufsetzen, der nicht öffentlich zugänglich sein soll?

**Verständisfrage 13**

In welcher Form ist die RPC Kommunikation wichtig für WWW-Applikationen?

**Verständisfrage 14**

Welche Funktion hat ein Provider, wozu wird er benötigt?

**Verständisfrage 15**

Wozu könnte ein RPC innerhalb eines Rechner nützlich sein?

## Publikationsverzeichnis – Literatur

- [Blo-92] Bloomer,J.: Power Programming with RPC, ISBN 0-937175-77-3, O'Reilly & Associates, Inc.
- [Gek-94] Gerding,M.; Kibbel,B: Novell DOS 7 und Personal Netware Betriebssystem und Peer to Peer Netzwerk
- [Gei-92] Geiger, K.: Inside ODBC, ISBN 3-86063-359-7
- [Gil-95] Gilster, P.: The SLIP/PPP Connection, ISBN 0-471-11712-9
- [HAN-96] Hansen, H.R: Wirtschaftsinformatik I Lucius&Lucius, Stuttgart, 1996
- [Hun-95] Hunt,C.: TCP/IP Netzwerkadministration, ISBN 3-930673-02-9, O'Reilly&Associates, Inc.
- [Jam-96] James, P.: Netscape Navigator, ISBN 3-8266-0261-7, International Thomson Publishing, 1996
- [LeBaSi-98] Lewis, G.; Barber, S.; Siegel,E.: Programming with Java IDL, Wiley Computer Publishing, 1998.
- [Schm-96] Schmidt,F.: SCSI-Bus und IDE-Schnittstelle, ISBN 3-89319-849-0, Addison Wesley (Deutschland) GmbH, 1996
- [Pos-82] Postel, J.: Simple Mail Transfer Protocol, RFC 821, USC/ Information Sciences Institute, August 1982.
- [Ros-89] Rose, M. T.: The Open Book: A Practical Perspective on Open Systems Interconnection, Prentice Hall Series in Innovative Computing, Prentice Hall, Englewood Cliffs, 1989
- [Soe-96] Soethe,F.: Intranet mit HTML und Netscape. Intranet mit HTML und Netscape, ISBN 3-89319-954-3
- [Sun-87] Sun Microsystems: XDR: External Data Representation, RFC 1014, June 1987
- [Sun-88] Sun Microsystems: RPC: Remote Procedure Call Protocol Specification Version 2, RFC 1057, June 1988
- [SyLa-97] Ahamed, Syed V. ; Lawrence, Victor B. : Intelligent Broadband Multimedia Networks 1997 Kluwer Academic Publishers
- [TiSc-85] Tietze, U.; Schenk, Ch. : Halbleiterschaltungstechnik, Springer-Verlag 1986
- [Wet-87] H. Wettstein .: Architektur von Betriebssystemen, Springer Verlag 1987