

# ERP-OnTo-PDM: Konzept und prototypische Realisierung einer ontologiebasierten ERP/PDM Kopplung mittels XML-Technologie

## Dissertation

zur Erlangung des akademischen Grades eines Doktors der  
Ingenieurwissenschaften  
(Dr.-Ing.)

durch den Fachbereich Maschinenwesen  
(Energie- und Verfahrenstechnik, Maschinentchnik)  
der Universität Duisburg-Essen (Standort Essen)

vorgelegt von

Dipl.-Wirt.Inform.  
Uwe Kraus  
aus Sankt Augustin

Essen 2003

Tag des Rigorosums: 28.08.2003  
Erstgutachter: Prof. Dr.-Ing. H.J. Stracke  
Zweitgutachter: Prof. Dr.-Ing. R. Tracht

# Inhaltsverzeichnis

Abbildungsverzeichnis .....	IV
Abkürzungsverzeichnis.....	VII
<b>1 Einleitung und Motivation</b>	<b>1</b>
1.1 Problemstellung . . . . .	4
1.2 Zielsetzung . . . . .	6
1.3 Aufbau der Arbeit . . . . .	7
<b>2 Stand der Forschung auf dem Gebiet der IT-Systeme im Unternehmen</b>	<b>9</b>
2.1 Das unternehmensweite Informationsmanagement . . . . .	10
2.2 Enterprise Resource Planning (ERP)-Systeme . . . . .	11
2.2.1 Begriffsbestimmung ERP-Systeme . . . . .	13
2.2.2 ERP Systeme am Beispiel von SAP R/3 . . . . .	14
2.3 Product Data Management(PDM)-Systeme . . . . .	16
2.3.1 Begriffsbestimmung PDM-Systeme . . . . .	17
2.3.2 Das PDM-System SmarTeam . . . . .	23
2.4 Kopplungsstrategien zwischen ERP- und PDM-Systemen . . . . .	24
2.5 Fazit . . . . .	28
<b>3 Stand der Forschung auf dem Gebiet der Ontologien</b>	<b>30</b>
3.1 Begriffsbestimmung: Ontologie . . . . .	31

3.1.1	Ontologie in der Philosophie . . . . .	32
3.1.2	Ontologie in der Informatik . . . . .	32
3.1.3	Abgrenzung gegenüber verwandten Konzepten . . . . .	37
3.2	Ontologiearten . . . . .	42
3.3	Fazit . . . . .	46
<b>4</b>	<b>Stand der Forschung auf dem Gebiet Internettechnologie</b>	<b>49</b>
4.1	Begriffsbestimmung . . . . .	50
4.1.1	Abgrenzung von XML und SGML . . . . .	51
4.1.2	Abgrenzung von XML und HTML . . . . .	53
4.2	Das Konzept der strukturbeschreibenden Metasprache XML . . . . .	54
4.3	Die Struktur von XML Dokumenten . . . . .	61
4.3.1	Struktur von XML Dokumenten durch DTD . . . . .	63
4.3.2	Struktur von XML Dokumenten durch XSchema . . . . .	65
4.4	Verarbeiten von XML Dokumenten . . . . .	65
4.4.1	SAX- Simple API for XML . . . . .	65
4.4.2	DOM - Document Object Model . . . . .	66
4.5	Resource Description Framework (RDF) . . . . .	66
4.6	Product Data Markup Language (PDML) . . . . .	69
4.7	MatML . . . . .	69
4.8	Fazit . . . . .	70
<b>5</b>	<b>Anforderungen an die Kopplung</b>	<b>72</b>
5.1	Organisatorische Anforderungen . . . . .	74
5.1.1	Konstruktion . . . . .	75
5.2	Technische Anforderungen . . . . .	76
5.3	Fazit . . . . .	83

<i>INHALTSVERZEICHNIS</i>	III
<b>6 Das Konzept ERP-OnTo-PDM</b>	<b>84</b>
6.1 Aufbau des Konzeptes: ERP-OnTo-PDM . . . . .	85
6.2 Abbildung der Ontologie . . . . .	92
6.3 Einbindung des Konzeptes in PDM-Systeme . . . . .	94
6.4 Einbindung des Konzeptes in ERP-Systeme . . . . .	97
6.5 Einbindung des Konzeptes in andere IT-Systeme . . . . .	98
6.6 Kommunikation mittels ERP-OnTo-PDM . . . . .	100
6.7 Ablauf des Datentransfers . . . . .	103
6.8 Fazit . . . . .	103
<b>7 Realisierung und Anwendung von ERP-OnTo-PDM</b>	<b>106</b>
7.1 Verwendete Vorgehensmodelle . . . . .	107
7.1.1 Vorgehensmodell zur Erstellung der Ontologie: ERP-OnTo-PDM . . . . .	107
7.1.2 Vorgehensmodell zur Realisierung der Schnittstelle . . . . .	108
7.2 Anwendung von ERP-OnTo-PDM am Beispiel der Stückliste . . . . .	109
7.2.1 Stücklistendaten aus Sicht des SAP R/3 Systems . . . . .	112
7.2.2 Stücklistendaten aus Sicht von SmarTeam . . . . .	115
7.3 Fazit . . . . .	115
<b>8 Zusammenfassung und Ausblick</b>	<b>117</b>
<b>Literaturverzeichnis</b>	<b>119</b>

# Abbildungsverzeichnis

1.1	Der E-Business-Hypecycle . . . . .	3
1.2	Herkömmliche Kopplung zwischen ERP- und PDM-Systemen . . . . .	5
1.3	Kopplung zwischen ERP- und PDM-Systemen mittels ERP-OnTo-PDM . . . . .	6
1.4	Aufbau dieser Arbeit . . . . .	8
2.1	Zukunft von Informationsverarbeitungssystemen in Unternehmen . . . . .	11
2.2	Abgrenzung zwischen PDM- und ERP-Systemen (nach Vajna) . . . . .	12
2.3	Einstiegsbildschirm SAP R/3 . . . . .	16
2.4	Einbindung von EDM/PDM . . . . .	20
2.5	PDM Systemarchitektur . . . . .	22
2.6	Informationen SmarTeam . . . . .	23
2.7	Einstieg SmarTeam . . . . .	24
2.8	Prozesskette (Quelle: Siemens AG) . . . . .	25
2.9	Systemarchitektur . . . . .	25
2.10	SAP Kopplung PDM . . . . .	29
3.1	Vom Dokumentenmanagement zum Wissensmanagement - Ausbaustufen von EDM/PDM-Systemen . . . . .	31
3.2	Semiotisches Dreieck . . . . .	33
3.3	Beispiel eines Thesaurus nach DIN1463(2) . . . . .	39
3.4	Klassen von Ordnungssystemen . . . . .	40
3.5	Einordnung . . . . .	41

3.6	Arten von Ontologien . . . . .	42
3.7	Zusammenhang: Ontologie-Wissen/Dokumente-Informationen . . . .	47
3.8	Prinzip einer Ontologie . . . . .	48
4.1	Zweiteilung: Daten und Deklaration von Sprachelementen . . . . .	53
4.2	Das XML-Konzept 1 . . . . .	54
4.3	Das XML-Konzept 2 . . . . .	57
4.4	Das XML-Konzept 3 . . . . .	58
4.5	Zusammenhang zwischen logischer und physikalischer Struktur ei- nes XML-Dokuments . . . . .	62
4.6	Das XML-Konzept 4 . . . . .	63
4.7	Das XML-Konzept 5 . . . . .	64
4.8	Das RDF Model . . . . .	68
5.1	Vorgehen beim Entwickeln und Konstruieren nach VDI 221 . . . . .	77
6.1	Aufbau des Konzeptes . . . . .	86
6.2	Beispiel für den Aufbau der ERP-OnTo-PDM-XML Datei zum Aus- tausch von Stücklistendaten . . . . .	87
6.3	Die zwei Ebenen der Datenintegration . . . . .	90
6.4	Abbildung der Ontologie . . . . .	94
6.5	Datentypen XSL . . . . .	95
6.6	Arten des Datenaustauschesinnerhalb des E-Business . . . . .	99
6.7	Aufbereitung der ERP-OnTo-PDM Daten . . . . .	100
6.8	Aufbau der Steuerungsdatei . . . . .	102
6.9	Prozesskette: Material anlegen . . . . .	104
7.1	xml Spy 1 . . . . .	109
7.2	Einstiegsbildschirm ERP-OnTo-PDM . . . . .	110
7.3	ERP-OnTo-PDM Funktion: Feldauswahl . . . . .	111
7.4	ERP-OnTo-PDM Funktion: Anzeige ERP-OnTo-PDM File (IE) . . . .	111
7.5	Hilfsfunktion Hochladen der ERP-OnTo-PDM-Datei im SAP-System	112

7.6	Ergebnis der Hilfsfunktion:Stücklisten anzeigen . . . . .	114
7.7	Einstieg SmarTeam2 . . . . .	115

# Abkürzungsverzeichnis

**ABAP/4** Advanced Business Application Programming

**AM** SAP R/3 Modul: Anlagenwirtschaft

**API** Application Programming Interface

**ATS** Application Transaction Sets

**B2B** Business to Business

**BC** SAP R/3 Basis Components

**BC-CCMS** SAP R/3 Modul: Basis Computing Center Management System

**BC-COM** SAP R/3 Modul: Basis Kommunikationskomponente

**BC-DB** SAP R/3 Modul: Basis Datenbank-Dienste

**BC-DS** SAP R/3 Modul: Basis Development System

**BC-GUI** SAP R/3 Modul: Basis Graphical User Interface

**BC-IM** SAP R/3 Modul: Basis Information Model

**BDE** Betriebsdatenerfassung

**bzw.** beziehungsweise

**CAD** Computer Aided Design

**CAE** Computer Aided Engineering

**CAM** Computer Aided Manufacturing

**CASE** Computer Aided Software Engineering

- CAO** Computer Aided Office Automation
- CAP** Computer Aided Planning
- CAQ** Computer Aided Quality Assurance
- CIM** Computer Integrated Manufacturing
- DCF GML** Document Composition Facility Generalized Markup Language
- DESIRE** Development of a European Service for Information on Research and Education
- DOM** Document Object Model
- DTD** Dokumenttyp-Definition
- EDM** Engineering Data Management
- EDMS** Engineering Document Management System bzw. Engineering Data Management System
- eLib** Electronic Libraries
- ERP** Enterprise Resource Planning
- GML** Document Composition Facility Generalized Markup Language
- HTML** Hypertext Markup Language
- HTTP** Hypertext Transferprotocol
- ISO** International Organization for Standardization
- IT** Informationstechnologie
- KI** Künstliche Intelligenz
- MRP** Material Requirement Planning
- MRP-II** Manufacturing Resources Planning
- PDM** Produkt Daten Management bzw. Product Data Management
- PDMS** Produkt-Daten-Management-Systeme
- PIM** Product Information Management

- PPS** Produktionsplanung- und -steuerungs-System
- PDML** Product Data Markup Language
- QBIC** Query by Image Content
- R/3** Produktbezeichnung der Firma SAP für das Realsystem 3
- RDF** Resource Description Framework
- SAP** Akronym der Softwarefirma: Software, Anwendungen und Produkte in der Datenverarbeitung
- SAX** Simple API for XML
- SML** Sachmerkmal-Leisten
- SGML** Standard Generalized Markup Language
- TIS** Technisches Informationssystem
- URI** Uniform Resource Identifier
- VDMA** Verband Deutscher Maschinen- und Anlagenbau e.V.
- VPDM** Virtual Product Development Management
- W3C** World Wide Web Consortium
- WWW** World Wide Web
- XML** eXtensible Markup Language
- XQL** XML Query Language
- XSLT**
- XSL** XML Stylesheet Language
- XSL-FO**
- XPOINTER** XML Pointer
- XPATH** XML Path
- XLINK** XML Link
- XHTML** XML HTML 4.0
- XSchema** XML Schema

Mein Dank gilt insbesondere Herrn Prof. Dr.-Ing. H.-J. Stracke, der mich bei der Durchführung dieser Arbeit auf vielfältige Weise unterstützt hat. Herrn Prof. Dr.-Ing. R. Tracht danke ich für sein Interesse an der Thematik, sowie für seine Bereitschaft zur Übernahme des Koreferats und für die rasche Anfertigung des Zweitgutachtens.

# Kapitel 1

## Einleitung und Motivation

Durch die zunehmende Globalisierung und den dadurch gewachsenen Wettbewerbsdruck für Unternehmen, wurden seit Mitte der 80er Jahre verschiedene organisatorische und technische Konzepte entwickelt, um den gestiegenen Anforderungen gerecht zu werden. Zu den Bereichen mit dem höchsten Wirkpotential für solche Umstrukturierungsmaßnahmen gehören vor allem die indirekten Unternehmensbereiche. Untersuchungen haben gezeigt, dass etwa 60% - 80% der Durchlaufzeit eines Produktes in den indirekten Unternehmensbereichen determiniert werden [Fär81, 23 ff.]. Darüber hinaus werden in den indirekten Bereichen eines Unternehmens ca. 70% der Kosten eines Produktes festgelegt [Eve96, 32].

In der letzten Dekade haben sich drei wesentliche technische bzw. organisatorische Konzepte in den Unternehmen durchgesetzt:

1. Der Einsatz von Enterprise Resource Planning (ERP)-Systemen.
2. Das Internet als Trägertechnologie für die Unternehmenskommunikation.
3. Das Wissensmanagement als Grundlage einer Wissens- und Informationslogistik.

### **ERP**

Mit Beginn der 90er Jahre haben sich für die Verarbeitung betriebswirtschaftlicher Daten in den Unternehmen die sogenannten ERP-Systeme als Standard etabliert.

Aufgabe dieser Systeme ist vor allem die Speicherung und Verarbeitung betriebswirtschaftlicher Daten und das Management von betriebswirtschaftlichen Prozessen. Hierdurch wurde eine integrierte Planung der Produktionsprozesse auf einem unternehmensweiten Datenbestand möglich. Die Entwicklung dieser Systeme ist noch nicht abgeschlossen. Ihre Architektur wird mehr und mehr durch einen modularen und komponentenorientierten Aufbau gekennzeichnet. Dadurch wird eine flexible Konfiguration der Systeme möglich. Zusatzfunktionalitäten durch Drittanbieter können in die ERP-Systeme einfach integriert werden. Im technischen Unternehmensbereich können so spezielle Systeme zur Verwaltung von Produkt- und Prozessdaten mit ERP-Systemen gekoppelt werden. Diese speziellen Softwaresysteme zur Verwaltung und zum Management von Produkt- und Prozessdaten werden als (Produkt Daten Management bzw. Product Data Management (PDM))-Systeme bezeichnet.

Eines der Hauptziele für den Einsatz von PDM-Systemen, ist in der Interaktion zwischen Erzeugersystemen, die vor allem technische Produktdaten liefern, und betriebswirtschaftlichen Systemen zu sehen. Demzufolge kommt den PDM-Systemen vor allem die Aufgabe eines Integrationswerkzeuges zu. In ihnen werden sowohl technische Produkt- und Prozessdaten, als auch betriebswirtschaftliche Daten gesammelt und gemanagt. Die Bedeutung einer effizienten Kopplung zum integrierten Management der betriebswirtschaftlichen Daten aus den ERP-Systemen und den technischen Produkt- und Prozessdaten ergibt sich nicht zuletzt durch die Tatsache, dass 70% der Herstellungskosten schon bei der Konstruktion eines Produktes festgelegt werden [Eve96, 32]. Ein effizienter Zugriff auf die gespeicherten Informationen und das Erfahrungswissen mittels PDM-System ist eine grundlegende Voraussetzung für eine wettbewerbsfähige Produktentwicklung.

Ferner ist eine Verteilung der Konstruktionstätigkeiten im Rahmen von Collaborative Engineering bzw. Electronic-Engineering zu beobachten. Hierbei wird der Konstruktionsprozess auf global agierende Teams verteilt. Diese Verteilung und der damit verbundene Austausch von Daten wird mittels verschiedener Dienste des Internets realisiert.

## **Internet**

Das Internet hat mit seinen verschiedenen Diensten die Art und Weise, wie Geschäfte getätigt werden, verändert. Die Euphorie des E-Business ist zwar in den vergangenen drei Jahren stark gebremst worden, jedoch wird nach dieser Konsolidierungsphase vermutlich wieder eine Wachstumsphase folgen. Dieser Zyklus wird

durch den in Abbildung 1.1 dargestellten E-Business-Hypecycle verbildlicht.

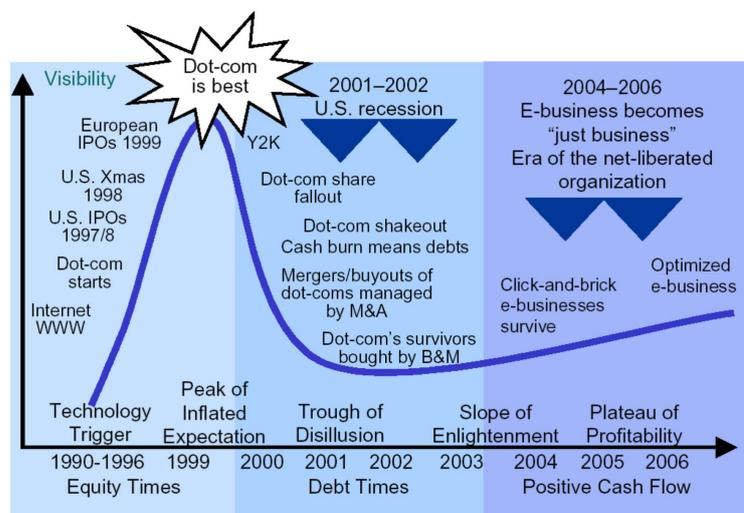


Abbildung 1.1: Der E-Business-Hypecycle nach [Gar03]

Es wird vermutet, dass gerade der Bereich des Business to Business (B2B) nach dieser Konsolidierungsphase stark expandieren wird. Um die Erfolgsaussichten dieser Business-Anwendungen zu erhöhen, ist es notwendig, dass die Dienste und Protokolle des Internets weiterentwickelt werden. Die bisher verwendeten Standards zur Implementierung von Internet-Anwendungen und insbesondere die Hypertext Markup Language (HTML) sind für den Bereich einer ERP-PDM-Kopplung nicht im ausreichenden Maße geeignet.

## Wissensmanagement

Das dritte technische bzw. organisatorische Konzept, um den geänderten Anforderungen an Unternehmen Rechnung zu tragen, ist das Wissensmanagement. Die Bereitstellung der richtigen Information zur richtigen Zeit und am richtigen Ort ist für produzierende Unternehmen von großer Bedeutung. Wissen wird zunehmend als Produktionsfaktor betrachtet und als eine der wichtigsten Unternehmensressourcen identifiziert. Im Unternehmen wird das Management von Wissen und Informationen durch deren stark wachsende Mengen erschwert. Diese Beobachtungen gelten uneingeschränkt für den Bereich der Konstruktion und Entwicklung.

Im Rahmen dieser Arbeit wird daher untersucht, wie durch den Einsatz einer geeigneten Form der Wissensrepräsentation eine semantische Kopplung von ERP- und PDM-Systemen realisiert werden kann.

## 1.1 Problemstellung

Im Bereich des Produktdatenmanagements sind demzufolge zwei Hauptproblemstellungen auszumachen:

1. Erhöhte Anforderungen an das Datenmanagement innerhalb eines Unternehmens.
2. Erhöhte Anforderungen für eine verteilte Produktentwicklung auch über Unternehmensgrenzen hinaus.

Der Prozess der Produktentwicklung wird durch die geänderten Rahmenbedingungen und die daraus resultierenden Forderungen nach immer kürzeren Produktlebenszyklen und damit einhergehend kürzeren Entwicklungszeiten geprägt. Zur Unterstützung der Tätigkeiten im Bereich der Produktentwicklung wurden, nicht zuletzt durch die Computer Integrated Manufacturing (CIM) Bestrebungen der 80er Jahre, eine Vielzahl von Informationstechnologie (IT)-Systemen installiert. Dies sind beispielsweise Computer Aided Design (CAD)-, Computer Aided Planning (CAP)- und Computer Aided Engineering (CAE)-Systeme. Durch den Einsatz dieser Systeme ist die Menge der erzeugten Daten und Informationen während des Entwicklungsprozesses erheblich gestiegen. Neue bzw. erweiterte Gesetze, Verordnungen und Normen in Bezug auf die Produkthaftung, die eine möglichst lückenlose Dokumentation über die Produkte und deren Herstellungsprozess erfordern [Ger00], führen zu einem zusätzlichen Anstieg der zu verarbeitenden Informationen. Ein Beispiel ist die Qualitätsmanagement-Norm International Organization for Standardization (ISO) 9000 ff [Tei], die für eine Zertifizierung des Unternehmens eine vollständige Dokumentation des Produktentstehungsprozesses fordert. Eine weitere Erhöhung der Anforderungen ist von Seiten der Kunden auszumachen, die verstärkt auch eine Verfügbarkeit der Produktinformationen über das Internet im Rahmen des E-Commerce verlangen. Aus diesem Grund müssen die Produktinformationen auch in einem für das Internet kompatiblen Format zur Verfügung gestellt werden. Aufgrund der Tatsache, dass in PDM-Systemen überwiegend technische Produktdaten aus dem Bereich der vorgelagerten Erzeugersysteme wie z. B. CAD, CAP und CAE gespeichert werden und betriebswirtschaftliche Produktdaten in ERP-Systemen gehalten werden, ist eine Kopplung dieser Systeme eine unerlässliche Voraussetzung für die Schaffung eines integrierten Produktdatenmanagements.

Durch den Einsatz isolierter, nicht direkt gekoppelter ERP- und PDM-Systeme ist die Schaffung expliziter Schnittstellen bzw. Dateikonverter für die verwendeten

Systeme nötig. Dies führt für den Datenaustausch zwischen diesen Systemen zu der in Abbildung 1.2 verdeutlichten Situation.

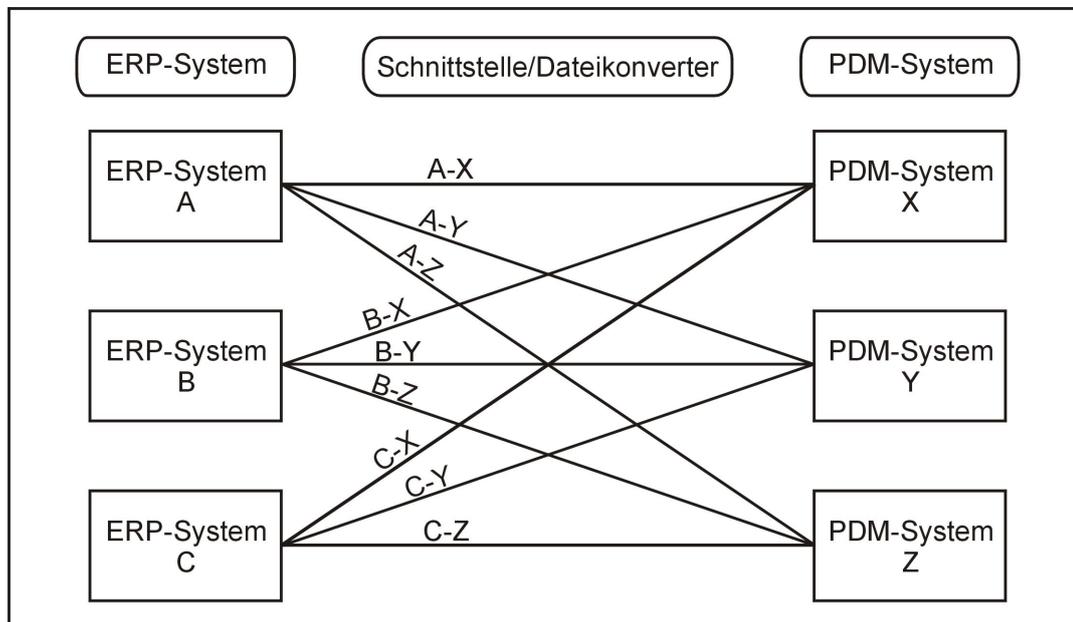


Abbildung 1.2: Herkömmliche Kopplung zwischen ERP- und PDM-Systemen

Es muss eine Vielzahl von Schnittstellen für den Datenaustausch entwickelt werden. Die Anzahl der Schnittstellen in einer Umgebung mit  $n$  beteiligten Systemen beträgt  $[n \times (n - 1)]$ . Diese Schnittstellen sind meist so spezifisch, dass eine Wiederverwendung selten möglich ist. Dies ist darin begründet, dass es keine semantisch einheitliche Definition der Datenstrukturen innerhalb der einzelnen Systeme gibt. Die innerhalb der einzelnen Systeme verwendeten Konzepte werden in der Regel durch diese unterschiedlich interpretiert. Daher müssen systemspezifische Konvertierungsmechanismen implementiert werden, um einen semantisch korrekten Austausch (semantic mapping) der Daten zwischen unterschiedlichen Systemen zu ermöglichen.

Ein weiterer Problemkreis ergibt sich aus kollaborativen Arbeitsformen. Bei der verteilten Produktentwicklung kommt es immer mehr zur Zusammenarbeit von Unternehmen, die weltweit agieren und den Entwicklungsauftrag in Projektteams bearbeiten. Hierbei müssen die Entwickler, die aus unterschiedlichen Kulturkreisen stammen können, einen einheitlichen Begriffsraum (Ein einheitliches Verständnis über die Bedeutung von Begriffen) besitzen.

Einen Lösungsansatz für beide vorgenannten Problemfelder stellt das im Bereich der Künstliche Intelligenz (KI) entwickelte Konzept der Ontologien dar.

## 1.2 Zielsetzung

Diese Arbeit verfolgt das Ziel, eine ontologiebasierte Wissensrepräsentationsform zu entwickeln. Diese wird im Folgenden mit dem Akronym ERP-OnTo-PDM abgekürzt. Dabei handelt es sich um eine Anwendungs-Ontologie, die als Interlingua-Server zwischen unterschiedlichen ERP- und PDM-Systemen eingesetzt werden kann. Die ERP-OnTo-PDM ist dabei produktunabhängig. Die Arbeitsweise und Anwendbarkeit von ERP-OnTo-PDM wird mittels einer prototypenhaften Implementierung für die Systeme SAP R/3 (als ein ERP-System) und SmarTeam (als ein PDM-System) evaluiert.

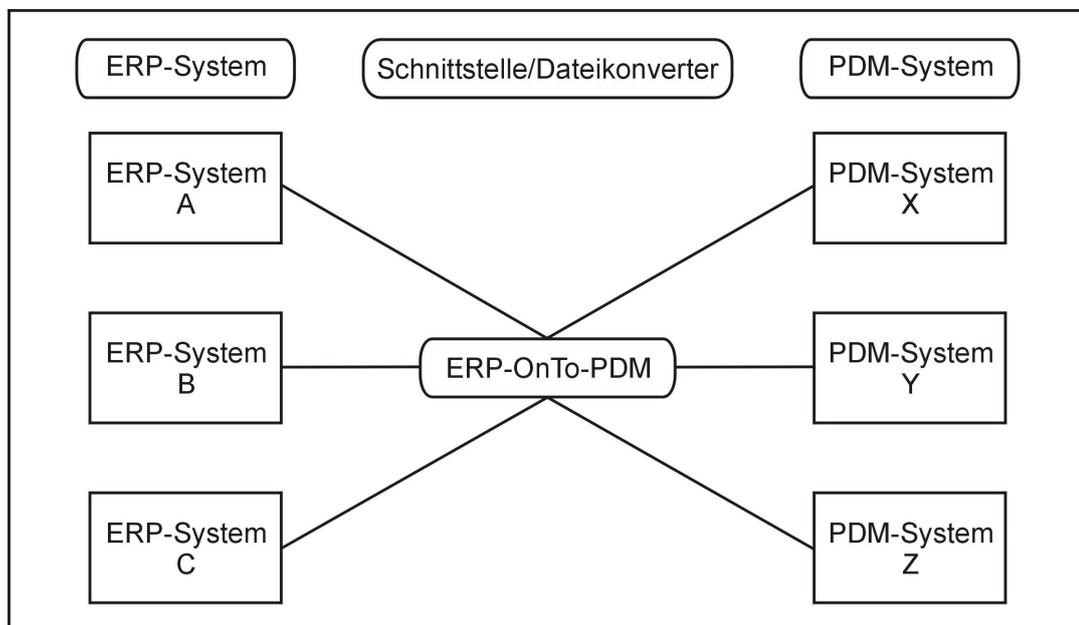


Abbildung 1.3: Kopplung zwischen ERP- und PDM-Systemen mittels ERP-OnTo-PDM

Die Funktionsweise als Interlingua-Server von ERP-OnTo-PDM wird durch die Abbildung 1.3 verdeutlicht. Durch ERP-OnTo-PDM wird eine einheitliche Definition der Konzepte sichergestellt. Daher wird in einem Gesamtsystem von  $n$  beteiligten IT-Systemen nicht  $[n \times (n - 1)]$  Schnittstellen benötigt, sondern nunmehr lediglich eine Schnittstelle zu ERP-OnTo-PDM pro IT-System, um ein semantisches Mapping zwischen allen beteiligten Systemen zu erreichen. Des Weiteren stellt ERP-OnTo-PDM die Basis für einen einheitlichen Begriffsraum dar.

### 1.3 Aufbau der Arbeit

Im anschließenden Kapitel 2 werden der Stand der Forschung im Bereich der ERP- bzw. PDM-Systeme, sowie Strategien zur Kopplung dieser Systeme aufgezeigt. Dazu wird zunächst für den jeweiligen Bereich ein einheitliches Begriffsverständnis geschaffen und darauf aufbauend jeweils stellvertretend ein Produkt für die jeweilige Systemart vorgestellt. Die Betrachtung der Produkte erfolgt dabei in Hinblick auf deren Einsatz bei der Realisierung von ERP-OnTo-PDM in Kapitel 7. Darüber hinaus werden in diesem Kapitel bestehende Strategien zur Kopplung von IT-Systemen eingehender betrachtet und deren Schwachstellen aufgezeigt.

Das Kapitel 3 stellt den Stand der Forschung auf dem Gebiet der Ontologien vor. Zunächst wird in diesem Kapitel das Konzept Ontologie vorgestellt, welches im Rahmen des Wissensmanagements eine immer größere Bedeutung erhält. Ontologien bilden den Kern des im Rahmen dieser Arbeit erarbeiteten ERP-OnTo-PDM-Konzeptes, einer ontologiebasierten, auf eXtensible Markup Language (XML)-Technologie aufbauenden Implementierung zum semantischen Produktdaten-Austausch zwischen verschiedenen IT-Systemen, die produktrelevante Daten managen. Dazu werden neben den definitorischen Grundlagen der Begriffswelt des Forschungsgebiets der Ontologien besonders deren Einsatzmöglichkeiten im Bereich des Datenaustausches von Produktdaten vorgestellt.

Der Stand der Forschung auf dem Gebiet der Internettechnik ist Gegenstand des Kapitels 4. Hier wird die XML mit ihren assoziierten Standards dargestellt. Von besonderem Interesse zur Implementierung des Konzepts ERP-OnTo-PDM ist das Resource Description Framework (RDF), welches die Definition von Metadaten ermöglicht, um den semantikbasierten Produktdatenaustausch zu ermöglichen. Neben den Grundlagen von XML und RDF wird auch das Gebiet der XML-basierten Produktdatenaustauschformate behandelt und deren Schwachstellen aufgezeigt.

In Kapitel 5 werden aus den Schwachstellen, die in den drei vorhergehenden Kapiteln aufgezeigt wurden, die Anforderungen an eine semantische Kopplung von IT-Systemen mittels einer Ontologie abgeleitet und erläutert. Dazu werden die Anforderungen an eine Kopplung von ERP- und PDM-Systemen in drei Bereiche untergliedert. Zunächst werden Anforderungen, die sich aus der Unternehmensorganisation ergeben, dargestellt. Daran anschließend werden Anforderungen, die sich durch den Einsatz der beiden Informationsverarbeitungssysteme ergeben, aufgezeigt. Im dritten Anforderungsbereich sind technische Anforderungen zusammengefasst.

Das auf Grundlage der aufgezeigten Schwachstellen und den daraus abgeleite-

ten Anforderungen an eine semantische Kopplung von Informationsverarbeitungssystemen erarbeitete Konzept ERP-OnTo-PDM ist Gegenstand des Kapitels 6. Zunächst wird in diesem Kapitel eine Auswahl der zur Realisierung einer ontologiebasierten Kopplung geeigneten Basistechnologie getroffen. Das Konzept wird mittels eines RDF-Modells in diesem Kapitel veranschaulicht. Des Weiteren bildet RDF die Basistechnologie zur Realisierung des Konzeptes ERP-OnTo-PDM, welches in Kapitel 7 beschrieben wird. In diesem Kapitel wird zunächst auf das Vorgehensmodell zur Erstellung der Ontologie für das Konzept ERP-OnTo-PDM eingegangen. Daran anschließend wird die Anwendbarkeit durch eine prototypische Implementierung nachgewiesen.

Den Abschluss dieser Arbeit bildet das Kapitel 8. In diesem werden die durch diese Arbeit gewonnenen Erkenntnisse zusammengefasst und ein Ausblick auf mögliche weitere Forschungsarbeiten in dem Bereich des Produktdatenaustausches gegeben.

Der Aufbau dieser Arbeit wird schematisch in Abbildung 1.4 dargestellt.

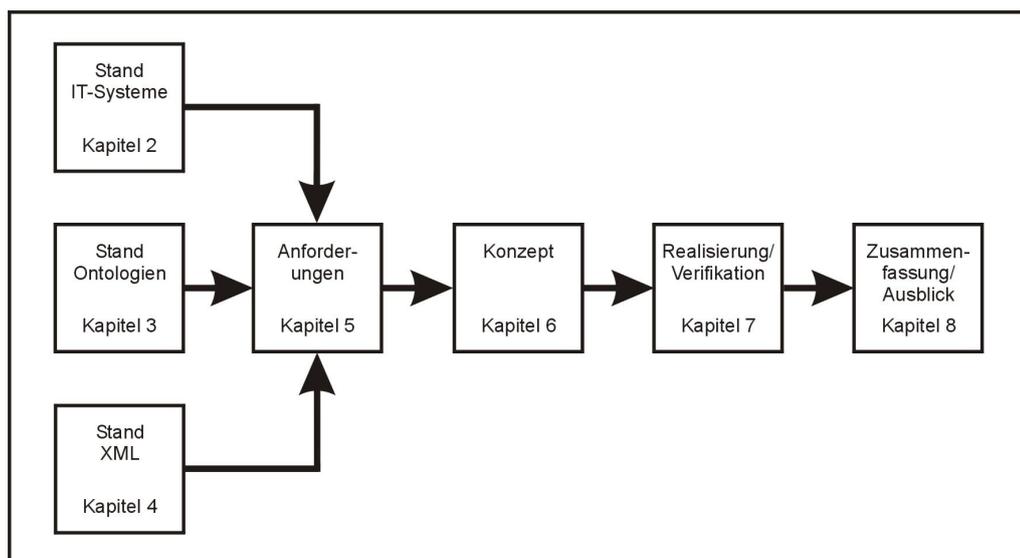
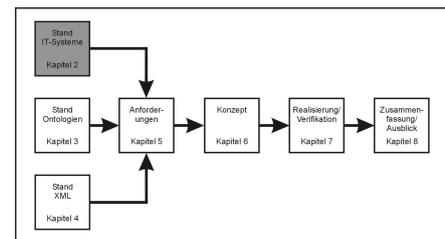


Abbildung 1.4: Aufbau dieser Arbeit

# Kapitel 2

## Stand der Forschung auf dem Gebiet der IT-Systeme im Unternehmen

Als Grundlage für den Einsatz von ERP-OnTo-PDM werden in diesem Kapitel ERP- und PDM-Systeme näher betrachtet. Zunächst wird die Entwicklungstendenz des unternehmensweiten Informationsmanagements erörtert. Darauf aufbauend wird auf ERP-Systeme eingegangen. Dabei wird der Begriff ERP-System definiert und gegen andere Informationsverarbeitungssysteme abgegrenzt. Beispielhaft wird der Aufbau des ERP-Systems R/3 der Firma SAP vorgestellt, welches das weltweit am häufigsten eingesetzte ERP-System ist [Erp02].



In gleicher Weise wird der Begriff PDM-System definiert, von anderen Informationsverwaltungssystemen für den technischen Bereich abgegrenzt und das PDM-System SmartTeam der Firma Smart Solutions vorgestellt.

Beide Systeme werden für die Realisierung von ERP-OnTo-PDM im Kapitel 7 exemplarisch verwendet.

Anschließend werden die existierenden Ansätze zur Kopplung von Informationsmanagementsystemen im Allgemeinen und für ERP-PDM-Systeme im Besonderen kritisch betrachtet.

## 2.1 Das unternehmensweite Informationsmanagement

Wie bereits erwähnt, sind Wirtschaftsunternehmen heute stärker als je zuvor mit einem raschen und beständigen Wandel der gesamtwirtschaftlichen Rahmenbedingungen konfrontiert. Dies führt unter anderem zu immer kürzeren Innovationszyklen, die in Kombination mit einem ständig steigenden Kostendruck eine permanente Optimierung aller Geschäftsabläufe erforderlich macht. Aufgrund der Tatsache, dass das immaterielle Gut „Wissen“ inzwischen als vierter Produktionsfaktor neben den klassischen Produktionsfaktoren<sup>1</sup> Boden, Arbeit und Kapital angesehen wird, ist ein integriertes Informationsmanagement in diesem Umfeld von strategischer Bedeutung für die Unternehmen. Das unternehmensweite Informationsmanagement wird in diesem Zusammenhang als Informationslogistik bezeichnet.

Hierbei besteht das größte Problem nicht in der Speicherung der schnell anschwellenden Daten- und Informationsflut, sondern in dem Wiederauffinden und dem Wiederverwenden bzw. Weiterverarbeiten dieser Daten und Informationen. Die technische Produktdokumentation ist ein Beispiel für eine solche durchgängige Verarbeitung. Die Dokumentation ist für verschiedene Zwecke und Medien aufzubereiten. Als Betriebsanleitung ist sie laut EG-Maschinenrichtlinie für industrielle Erzeugnisse vorgeschrieben [Ger00, 3]. Diese wird auf Papier als integraler Bestandteil dem Produkt beigelegt. Gleichzeitig kann eine multimedial aufbereitete technische Dokumentation innerhalb des Internetauftritts einer Firma einen erhöhten Kundennutzen bzw. ein Alleinstellungsmerkmal gegenüber dem Wettbewerb bedeuten. Wünschenswert ist somit eine einmalige Speicherung der technischen Produktdokumentation in einem Format, welches die Weiterverarbeitung für verschiedene Medien und IT-Systeme gestattet, ohne jeweils spezielle Konvertierungsprogramme benutzen zu müssen.

Das Internet entwickelt sich von einem reinen Informationsmedium zu einer technologischen Basis für ein unternehmensweites Informationsmanagement. Die zukünftige Landschaft der einzelnen Informationsverarbeitungssysteme ist aus Sicht des Autors in Abbildung 2.1 schematisch dargestellt.

Die Daten bzw. Informationsbestände der einzelnen Anwendungssysteme wer-

---

<sup>1</sup>Die Volkswirtschaftslehre unterscheidet die Produktionsfaktoren Boden bzw. Natur, Arbeit (geistige und körperliche) und Kapital[MK89]. Die Betriebswirtschaftslehre [Gut79] unterscheidet die dispositiven Faktoren Planung, Organisation, Kontrolle und die Elementarfaktoren objektbezogene Arbeitsleistung, Betriebsmittel und Werkstoffe

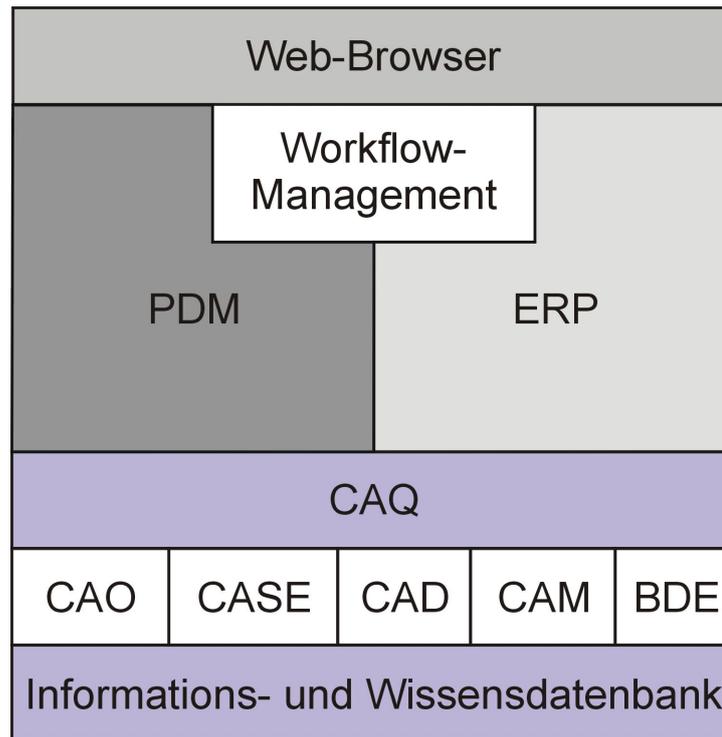


Abbildung 2.1: Zukunft von Informationsverarbeitungssystemen in Unternehmen

den logisch zu einer unternehmensweiten Informations- bzw. Wissensbasis zusammengeführt. Für das Informationsmanagement sind auf der betriebswirtschaftlichen Seite die ERP-Systeme verantwortlich, während für die technischen Applikationen die PDM-Systeme benutzt werden. Für das Management der Aufgabenausführung werden Workflowmanagementsysteme eingesetzt. Als Frontend wird ein Web-Browser Verwendung finden. Als Basistechnologie für diese Systemintegration werden verschiedene Dienste und Protokolle des Internets Verwendung finden.

Eine Abgrenzung der Aufgaben innerhalb des Informationsmanagements zwischen ERP- und PDM-Systemen wird durch die Abbildung 2.2 dargestellt.

## 2.2 Enterprise Resource Planning (ERP)-Systeme

Die Entwicklung zu den ERP-Systemen kann als ein Evolutionsprozess von Informationsverarbeitungssystemen angesehen werden. In den 1960er Jahren war der Anwendungszweck von Informationsverarbeitungssystemen im Bereich der Produktion auf Systeme beschränkt, die zur Verwaltung der Bestände eingesetzt wur-

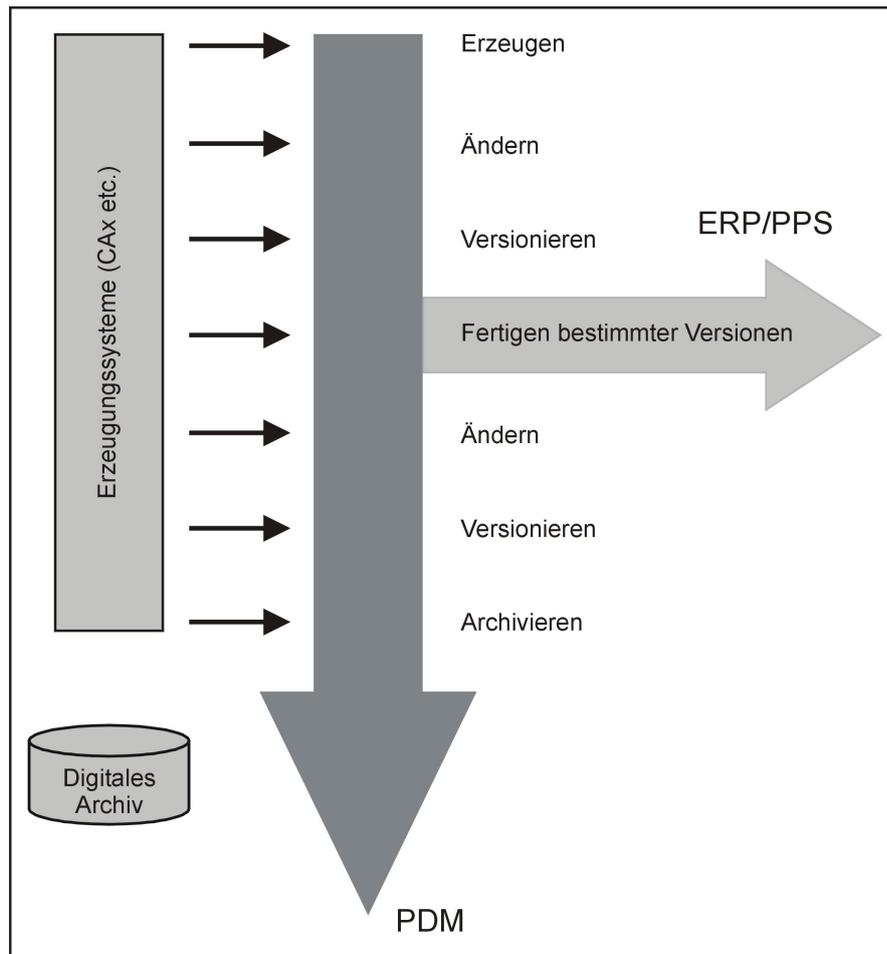


Abbildung 2.2: Abgrenzung zwischen PDM- und ERP-Systemen (nach Vajna)

den. Diese wurden in den 70er Jahren weiterentwickelt und der Schwerpunkt dieser Systeme ging durch das Material Requirement Planning (MRP)-Konzept auf die Planung von Sekundärbedarfen über, die durch Stücklistenauflösung mittels des MRP-Konzeptes aus den Primärbedarfen ermittelt wurden. Eine genaue Betrachtung der Arbeitsweise des MRP-Konzeptes liegt außerhalb des Fokus dieser Arbeit und es sei an dieser Stelle auf die einschlägige Literatur wie beispielsweise [KS93, 223] verwiesen. In den 80er Jahren wurde dann das Konzept des Manufacturing Resources Planning (MRP-II) in diese Systeme integriert. Das MRP-II Konzept ist eine Erweiterung des MRP-Konzeptes und bettet die Planungs- und Steuerungsproblematik in den Gesamtzusammenhang einer Logistikkette ein [Sch89, 37].

Aus der Erkenntnis, dass nicht nur die Produktion einer genauen Planung und Steuerung bedarf, sondern sämtliche Ressourcen in einem Unternehmen, entwickelten sich dann Anfang der 90er Jahre die ERP-Systeme. Neben dem Bereich der

Produktion wurden unter anderem auch die Bereiche der Personalplanung und -entwicklung, des Finanzwesens, sowie des Projekt Management in einem integrierten Informationsverarbeitungssystem unterstützt.

### 2.2.1 Begriffsbestimmung ERP-Systeme

Der Begriff der ERP-Systeme stammt aus dem amerikanischen Sprachraum. Er wird sowohl in der wissenschaftlichen Literatur als auch in der Praxis verwendet, um Informationsverarbeitungssysteme zu beschreiben, die eine unternehmensweite Informationsverarbeitung über die gesamte Logistikkette hinweg erlauben. Es ist festzustellen, dass es keine einheitliche Definition des ERP Begriffes gibt, sondern eine Vielzahl von Definitionen, die sich jedoch nur marginal voneinander unterscheiden.

Eine Definition des Begriffes Enterprise Resource Planning wird von Clewett et. al. gegeben:

„ERP (Enterprise Resource Planning): A client/server application that is designed to coordinate and act as an information resource to an entire organisation or enterprise.“[CFM98, 345]

Diese Definition spiegelt mehr die Aspekte aus Sicht der Wissenschaft wieder und geht vor allem auf die Systemarchitektur dieser Systeme ein. Eine Definition, die mehr die Praxis-Sicht widerspiegelt, lautet:

„Software solution that addresses the enterprise needs taking the process view of organizational goals tightly integrating all functions of an enterprise.“[Erp02]

Im deutschsprachigen Raum hat sich mit der starken Ausbreitung dieser Systeme seit Mitte der 90er Jahre auch der Begriff der ERP-Systeme eingebürgert. Stellvertretend für eine ganze Reihe von Definitionen soll hier die von Klahn wiedergegeben werden:

„ERP-Systeme (Enterprise Resource Planning) unterstützen die gesamte Prozesskette, angefangen bei der Angebotserstellung über die Fertigungssteuerung bis zur Rechnungsstellung.“ [Kla99, 22]

Allen diesen Definitionen sind folgende Eigenschaften gemeinsam:

- Unternehmensweites Daten- und Informations-Management.
- Unterstützung aller betriebswirtschaftlichen Unternehmensfunktionen.
- Skalierbarer Systemaufbau durch Client/Server Technologie.

Festzustellen ist, dass ERP-Systeme eine starke betriebswirtschaftliche Ausrichtung besitzen. Eine Unterstützung von technischen Unternehmensfunktionen wird von diesen Systemen meist nur unzulänglich angeboten. Oft wird diese Unzulänglichkeit durch das Bereitstellen von Schnittstellen zu anderen technischen Informationsverarbeitungssystemen behoben.

### 2.2.2 ERP Systeme am Beispiel von SAP R/3

In den 90er Jahren dominierte das Konzept der ERP-Systeme das Gebiet der betriebswirtschaftlichen Informationsverarbeitungssysteme. Neben der weiten praktischen Verbreitung haben ERP-Systeme auch Einzug in Forschung und Lehre erhalten. Der Erfolg dieser Standardanwendungssysteme für eine integrierte, unternehmensweite Informationsverarbeitung ist weltweit und insbesondere im deutschsprachigen Raum mit dem Produkt R/3 der Firma SAP (Software, Anwendungen und Produkte in der Datenverarbeitung) verbunden.

Bei dem von der SAP entwickelten Softwaresystem R/3 handelt es sich um ein inzwischen zum Standard avanciertes branchenneutrales Softwaresystem, welches die gesamten betriebswirtschaftlichen Geschäftsprozesse eines Unternehmens unterstützt. Dieses auf der Client/Server-Technologie basierende System hatte bereits im Jahre 1998 eine weltweit installierte Basis von 10.000 Systemen, [MK98] die bis zum Jahr 2001 auf fast 20.000 gestiegen ist [GLW02] und gilt damit als das ERP-System mit dem größten Marktanteil [Erp02].

Alle Betrachtungen und Beschreibungen in dieser Arbeit bezüglich des SAP R/3 Systems nehmen Bezug auf das Release 4.0b.

Für folgende betriebswirtschaftliche Bereiche bietet das integrierte Informationssystem R/3 Module zur Unterstützung an:

- **Logistik:**

Vertrieb	(SD sales and distribution),
Materialwirtschaft	(MM material management),
Produktionsplanung	(PP production planing),
Projektsystem	(PS project system),
Qualitätsmanagement	(QM quality management),
Instandhaltung	(PM plant maintenance);

- **Rechnungswesen:**

Finanzbuchhaltung	(FI financial accounting),
Anlagenwirtschaft	(AM assets management),
Investitionsmanagement	(IM investment management),
Controlling	(CO controlling);

- **Personalwirtschaft:**

Personalplanung und -entwicklung	(HR-PD human resources planing and development),
Personaladministration und -abrechnung	(HR-PA human resources personal accounting).

Diese betriebswirtschaftlichen Anwendungsmodule des SAP R/3-Systems können sowohl einzeln als auch in Kombinationen eingesetzt werden. Durch diese modulare Struktur kann dieses Standardinformationsverarbeitungssystem an die jeweiligen kundenindividuellen Erfordernisse angepasst werden. Ferner wird die Adaption an die Anforderungen des Benutzers durch den sogenannten *Customizing-Prozess* unterstützt, der vor der Einführung eines jeweiligen Moduls durchlaufen werden muss. Dieser Prozess wird auch als *Konfiguration der Geschäftsprozesse* bezeichnet. Während dieses Prozesses werden die kundenindividuellen Parameter eingestellt. Für weiterführende Information über den Customizing-Prozess siehe [CDI94, S.330] und [BEG95, S. 156]. Neben den rein betriebswirtschaftlichen Modulen bietet das SAP-System ein sogenanntes Basis System (BC Basic Components) an, welches grundlegende Dienste beinhaltet, die dem gesamten System zur Verfügung stehen. Die Basis-Komponenten lassen sich in nachstehende Gruppen untergliedern:

- Datenbank-Dienste (BC-DB data base),
- Graphical User Interface (BC-GUI graphical user interface),
- Kommunikationskomponente (BC-COM communication),

- Computing Center Management (BC-CCMS computing center management system),
- Entwicklungssystem (BC-DS development system),
- Informationsmodell (BC-IM information model).

Diese Komponenten stehen in jedem R/3-System, unabhängig von anderen Komponenten, zur Verfügung.

Innerhalb dieser Arbeit stehen besonders die Funktionen des Moduls MM im Mittelpunkt der Betrachtung.

Innerhalb dieser Arbeit stehen besonders die Funktionen des Moduls MM im Mittelpunkt der Betrachtung. In der Abbildung 2.3 ist der Einstiegsbildschirm in das SAP R/3 System Release 4.06 abgebildet. Für den Bereiche dieser Arbeit ist der SAP Bereich Logistik, mit seinen Funktionen zur Verwaltung der Stammdaten für Materialien, Stücklisten, Arbeitspläne etc., relevant.

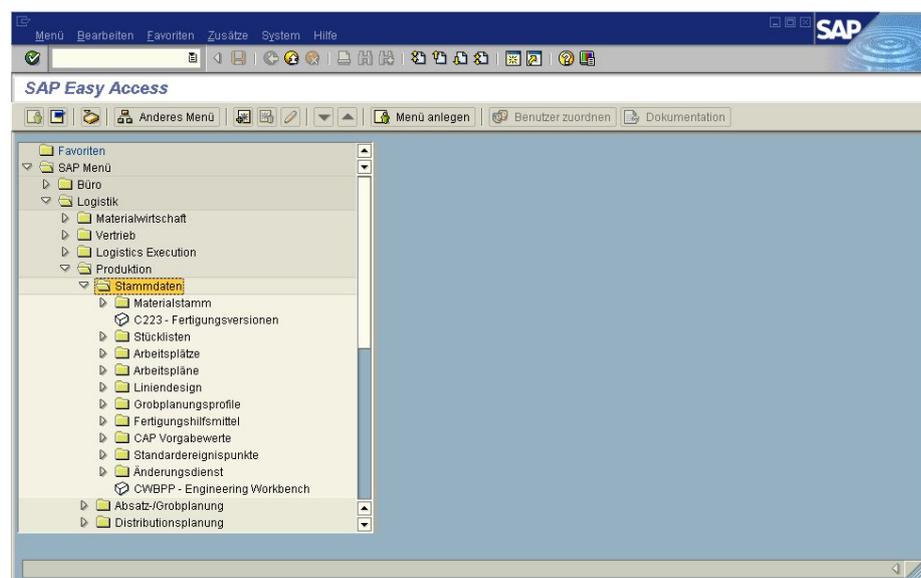


Abbildung 2.3: Einstiegsbildschirm SAP R/3 Release 4.06

## 2.3 Product Data Management(PDM)-Systeme

Die Menge an Informationen bzw. Daten, welche ein Produkt und dessen Entstehungsprozess beschreiben, wächst aufgrund der schon in der Motivation zu dieser

Arbeit beschriebenen, sich ändernden Rahmenbedingungen exponential an [Ger00, 5]. Ein effektives und durchgängiges Management dieser Produkt- bzw. Produktionsprozessdaten über den gesamten Produktlebenszyklus hinweg wird immer wichtiger. Nicht zuletzt aus diesem Grund wurden die sogenannten Engineering bzw. Produkt Daten Management Systeme (EDM-/PDM-Systeme) entwickelt. Diese Systeme haben sich bereits in der Praxis erfolgreich etabliert und werden heute allgemein als Schlüsseltechnologie mit hoher strategischer Bedeutung angesehen [KR98].

Zunächst wird eine begriffliche Grundlage für das Gebiet der PDM-Systeme geschaffen. Darauf aufbauend wird das PDM-System SmarTeam der Firma Smart Solutions vorgestellt. Anschließend wird der Stand der Forschung auf dem Gebiet der PDM-Systeme aufgezeigt.

### 2.3.1 Begriffsbestimmung PDM-Systeme

Wie in vielen Bereichen der Wirtschaftsinformatik existiert im Bereich des Produktdatenmanagements und der PDM-Systeme keine einheitliche Begriffsdefinition. Es gibt eine Vielzahl von Definitionen und Begriffen, die synonym Verwendung finden, teilweise jedoch inhaltlich unterschiedliche Bedeutungen besitzen. Aus diesem Grund werden an dieser Stelle die wichtigsten Begriffe und Definitionen erläutert und somit eine einheitliche Begriffsgrundlage für diese Arbeit geschaffen.

In seinem Ursprung stammt die Bezeichnung *PDM* bzw. *PDMS* aus den USA [EHSS91, 5]. Das Akronym PDM steht für *Product Data Management* bzw. *Produkt-Daten-Management* und wird in der Literatur synonym mit der Abkürzung PDMS für Produkt-Daten-Management-Systeme verwendet. Weitere gebräuchliche Begriffe für die Bezeichnung von PDM-Systemen, die aus dem amerikanischen Sprachraum stammen, sind Engineering Document Management System (EDMS) sowie Product Information Management (PIM). Diese Begriffe und Akronyme bezeichnen mit marginalen Unterschieden alle die gleiche Art von System zum Management von Produkt- und Prozessdaten, wobei sich im Laufe der Zeit der Begriff PDM-System durchgesetzt hat und die größte Verbreitung in Literatur und Praxis gefunden hat.

Die beiden folgenden Definitionen aus dem amerikanischen Sprachraum sind anschaulich und erscheinen aus Sicht des Autors besonders gelungen:

- „A Product Data Management system (PDM) is a software tool that manages engineering information, supports management of

product configurations, and supports management of the product engineering process. The engineering information includes both database objects and “document” objects - sets of information stored in files that are opaque to the PDM system. This information may be associated with specific products or specific product designs, or more generally with product families, the production process or the engineering process itself. The engineering process support usually includes workflow management and concepts of engineering change and notification. In many manufacturing organizations, the PDM is the central engineering information repository for product development activities“ [N.N00].

- „Product Data Management (PDM) is a tool that helps manage both product data and the product development process. PDM systems keep track of the masses of data and information required to design, manufacture or build, and then support and maintain products - whether your product is an petrochemical plant, highway, railway system, pharmaceutical, automobile, consumer product, or service. PDM is used effectively in a multitude of industries.“[N.N98].

Im deutschsprachigen Raum haben sich, synonym zum PDM-Begriff, die Begriffe des *Engineering Data Management* (EDM), des *Engineering Data Management System* (EDMS), der *Engineering Database* (EDB) sowie der Begriff *Technisches Informationssystem* (TIS) etabliert. Dies kann darauf zurückgeführt werden, dass mit PDM-Systemen das Informations- und Datenmanagement über die gesamte Prozesskette eines Produktes von der Produktidee bis zur Entsorgung und dem Recycling des Produktes unterstützt werden soll und dies als *Engineering* bezeichnet wird.

Die Unterscheidung zwischen PDMS und PDM sowie zwischen EDMS bzw. EDB und dem Begriff EDM kann darauf zurückgeführt werden, dass einige Autoren mit PDM/EDM den eigentlichen Prozess des Datenmanagement bezeichnen und mit PDMS/EDMS bzw. EDB die Datenmanagement Systeme. Dies wird durch die Definition des Verbands Deutscher Maschinen- und Anlagenbau e.V. (VDMA) der Begriffe PDM und PDMS unterstrichen:

„Produkt Daten Management oder PDM bezeichnet die ganzheitliche Verwaltung aller Daten, die bei der Entwicklung neuer Produkte

oder der Aktualisierung bisheriger Produkte anfallen, bearbeitet und weitergeleitet werden müssen, verbunden mit der Fähigkeit, den Prozess der Bearbeitung und Weiterleitung zu steuern und zu kontrollieren. Auf dieser Definition aufbauende Produkt Daten Management Systeme oder PDMS sind also Rechnersysteme zur Verwaltung technischer Daten und Dokumente von Produkten über ihren gesamten Lebenszyklus.“[uAVE95, 15f]

Die Abkürzung *EDM* für Engineering Data Management führt jedoch gerade im angelsächsischen Sprachraum zu Fehlinterpretationen, da hier dieses Akronym für Electronic Document Management verwendet wird [Kah00]. Außerdem wird EDM auch als abkürzende Schreibweise für Enterprise Data Management und Engineering Document Management verwendet.

Eine Unterscheidung der Begriffe EDM und PDM wird dagegen von Vajna [Vaj99, 25] und Abramovici [Abr99, 210] vorgenommen. Beide orientieren sich dabei an der VDI Richtlinie VDI 2219<sup>2</sup>. Diese führt die von beiden verwendete Bezeichnung EDM/PDM-System ein.

Vajna macht unterschiedliche Einsatzschwerpunkte beider Systeme aus. EDM Systeme werden danach hauptsächlich zum Management des Arbeitsprozesses und die ihn beschreibenden Daten eingesetzt. Wohingegen PDM-Systeme hauptsächlich die das Produkt beschreibenden Strukturen, Dokumenten und Daten umfassen. Abramovici verwendet die Bezeichnung EDM/PDM-Systeme und versteht darunter technische Datenbank- und Kommunikationssysteme,

„[...] die dazu dienen, Informationen über Produkte und deren Entstehungsprozesse bzw. Lebenszyklen konsistent zu speichern, zu verwalten und allen relevanten Bereichen eines Unternehmens bereitzustellen. EDM/PDM-Systeme stellen daher prozessübergreifende Systeme dar, mit gegenseitigen Beziehungen und Überlappungen sowohl zu den Erzeugersystemen (z. B. CAD- oder Office-Systeme) als auch zu anderen unternehmensübergreifenden IT-Anwendungen, wie z. B. ERP/PPS-, Workflow- oder Projektmanagement-Systemen“[Abr99, 210].

Aus Sicht des Autors kann diese Unterscheidung in der Praxis jedoch nicht so eindeutig getroffen werden. Die meisten der auf dem Markt befindlichen PDM-

---

<sup>2</sup>VDI-Richtlinie 2219: Einführungsstrategie und Wirtschaftlichkeit von EDM/PDM-Systemen (Reihe Datenverarbeitung in der Konstruktion). VDI-Gesellschaft Entwicklung Konstruktion Vertrieb Düsseldorf 1999

Systeme bieten die Möglichkeit sowohl eines Produkt-, als auch eines Prozessdatenmanagements (siehe dazu das in Abschnitt 2.3.2 beschriebene PDM-System SmarTeam). Im Rahmen dieser Arbeit werden, in Bezug auf die weiter oben angeführten Definitionen aus dem Amerikanischen und neueren Veröffentlichungen ([Kah00], [Ger00]), die keine Unterscheidung zwischen PDM- und EDM-Systemen vornehmen, PDM-Systeme und EDM/PDM-Systeme als synonyme Begriffe angesehen. Es wird daher im weiteren Verlauf dieser Arbeit nur noch der Begriff *PDM-System* verwendet.

Aus den Definitionen ist ersichtlich, dass PDM-Systeme eine Integrationsfunktion zwischen erzeugenden Systemen wie (CAD, CAE, CAP etc. ) auf der einen Seite und unternehmensübergreifenden, speziell auf betriebswirtschaftlich-planerische Aspekte konzipierte Systeme wie ERP- und PPS-Systemen auf der anderen Seite übernehmen. Während PDM-Systeme alle anfallenden Versionen und Änderungen in den Produktdaten transparent für den Benutzer mitführen, halten ERP- bzw. PPS-Systeme immer nur eine bestimmte Version der Produktdaten als Basis für ihre Planungsaufgaben. Dies kann z. B. die Erstellung der Fertigungsaufträge mit einer terminlichen und kapazitiven Determinierung sein. Dieser Zusammenhang wird durch die Abbildung 2.4 verdeutlicht.

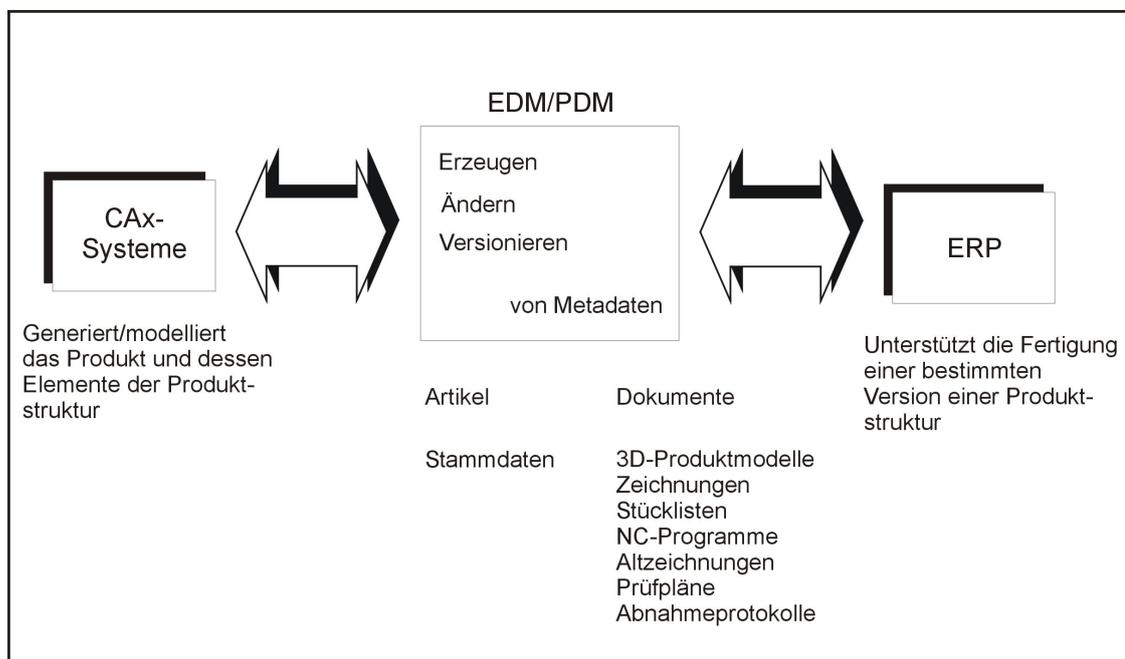


Abbildung 2.4: Einbindung von EDM/PDM [Ler99, 321]

Der Funktionsumfang einzelner PDM-Systeme kann sehr unterschiedlich aus-

fallen, aber von den meisten PDM-Systemen werden die nachfolgend aufgeführten sechs anwendungsbezogenen Funktionsbereiche unterstützt [Vaj99, 33f]:

- **Produktdaten- und Dokumentenmanagement**

Hierunter fällt die allgemeine Verwaltung von Produktdaten und den dazugehörigen Dokumenten aus den technischen Erzeugersystemen wie z. B. CAD-Modelle und Zeichnungen mit ihren verschiedenen Dateitypen und die Kopplung zu den jeweiligen Erzeugersystemen. Des Weiteren gehört dazu ein Versions- bzw. Statusmanagement, die Verwaltung von Projekten in Mappen bzw. Ordnern, sowie die Verwaltung von Metadaten für einzelne Objekte.

- **Produktstruktur- und Konfigurationsmanagement**

Hierunter werden Funktionen zusammengefasst, die die Erstellung bzw. Bearbeitung von Produktstrukturen und die Generierung von Stücklisten bzw. Teileverwendungsnachweisen beinhalten. Weiterhin wird das Management von Produktvarianten, sowie das Management der zeitlichen Veränderungen von Produktstrukturen unterstützt.

- **Klassifizierung und Teilefamilienmanagement**

In diesen Funktionsbereich fallen vor allem Aufgaben zur Generierung von Metadaten. Dies ist unter anderem das Bereitstellen von Sachmerkmalslisten zur Klassifizierung einzelner Teile und die Bereitstellung von Such- und Zugriffsmechanismen zum Wiederfinden von Teile- bzw. Produktinformationen.

- **Prozess- und Workflow-Management**

Dieser Funktionsblock erlaubt neben der Abbildung, Steuerung und Kontrolle von Geschäftsprozessen insbesondere im Fertigungsbereich auch die Abbildung, Steuerung und Kontrolle der korrespondierenden Informationsflüsse zu den einzelnen Geschäftsprozessen.

- **Benutzermanagement**

Darunter werden Funktionen verstanden, die die Abbildung der Organisationsstrukturen erlauben, um mittels Benutzer und Benutzergruppen Zugriffsrechte zu verwalten.

- **Projektmanagement**

Diese Funktionen ermöglichen neben der Planung bzw. Verwaltung von Aktivitäten, Abhängigkeiten und Zeitplänen auch die Verwaltung von Projektinformationen.

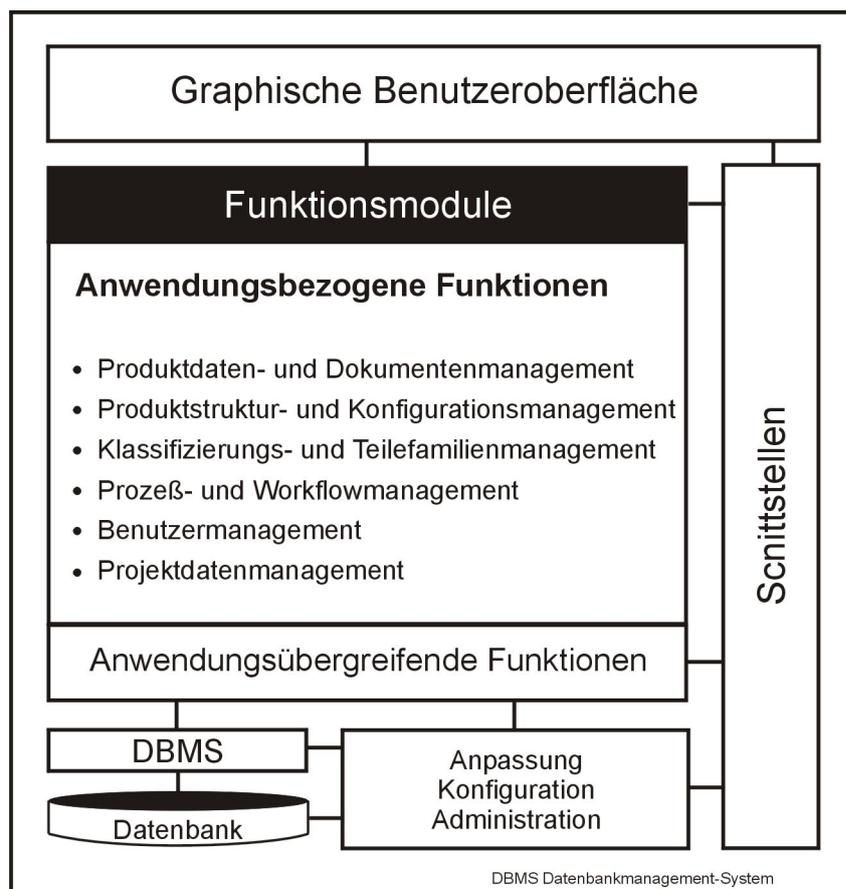


Abbildung 2.5: PDM Systemarchitektur [Ler99, 322]

Der typische Aufbau eines PDM-Systems, sowie die Einordnung der beschriebenen anwendungsbezogenen Funktionsblöcke in diese Systemarchitektur, wird durch Abbildung 2.5 gezeigt.

Daneben wird von einigen Herstellern von PDM-Lösungen eine neue Generation dieser Systeme unter der Bezeichnung *PDM II* offeriert. Diese Systeme sollen neben den traditionellen Aufgaben von PDM Systemen vor allem das *Virtual Product Development Management* (VPDM) unterstützen.

### 2.3.2 Das PDM-System SmarTeam

Bei SmarTeam handelt es sich um ein PDM System der Firma Smart Solutions. Ein effizientes Product Data Management ist essentiell für ein erfolgreiches Unternehmen. Durch SmarTeam wird der Workflow der Dokumente durch die Unternehmensprozesse unterstützt. SmarTeam bietet einen prozessorientierten Ansatz zur Unterstützung der Teamarbeit und der Kommunikation über den gesamten Produktlebenszyklus hinweg. Es zielt auf die Unterstützung aller Mitglieder eines Entwicklungsteams vom Designer über Produktion und Marketing ab.

Wird durch Abbildung 2.6 verdeutlicht.



Abbildung 2.6: Informationen SmarTeam

SmarTeam bietet die Möglichkeit, durch die Nutzung von Templates einen schnellen Einsatz des Systems zu erreichen. Templates sind vorgegebene Datenstrukturen, die dem Benutzer zur Verfügung gestellt werden. Die Templates können durch ein Customizing an die individuellen Bedürfnisse des Kunden angepasst werden.

Projekte stellen die höchste Organisationsebene in SmarTeam da. Jedes Projekt unterteilt sich in verschiedene Informationsebenen.

Wird durch Abbildung 2.7 verdeutlicht.

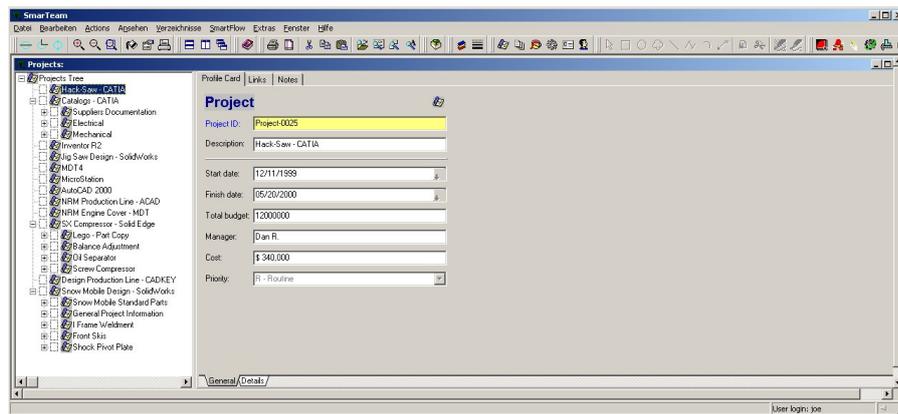


Abbildung 2.7: Einstieg SmarTeam

## 2.4 Kopplungsstrategien zwischen ERP- und PDM-Systemen

Wie in den letzten beiden Abschnitten beschrieben, sind die Aufgabenstellungen an ERP- und PDM-Systeme unterschiedlich. In beiden Systemen werden jedoch zum Teil redundant Daten bzw. Informationen über Produkte und Prozesse gespeichert. Das vom Harrington entwickelte Grundprinzip des Produktlebenszyklus kann EDV-technisch als eine Reihe von Datenverarbeitungsoperationen angesehen werden [Har73]. Ablauforganisatorisch sind diese Prozessschritte eng miteinander verzahnt, und es erscheint unter diesem Gesichtspunkt nicht zweckmäßig, einzelne Funktionen isoliert voneinander zu betrachten. Deshalb wird ein rechner-integrierter Prozeßablauf, mit einer Verknüpfung der einzelnen beteiligten System angestrebt.

Eine Zuordnung der beiden Systeme zu einzelnen Phasen des Produktlebenszyklus wird durch die Abbildung 2.8 gegeben [Ler99, 312].

Der grundsätzliche Systemaufbau von PDM Systemen, sowie die Beziehung zu ERP-Systemen über den Produktentstehungsprozess, wird durch Abbildung 2.9 wiedergegeben.

In der Literatur werden grundsätzlich zwei Strategien zur Verknüpfung von IT-Systemen unterschieden [Sch88]:

1. Koppelung von IT-Systemen und
2. Integration von IT-Systemen.

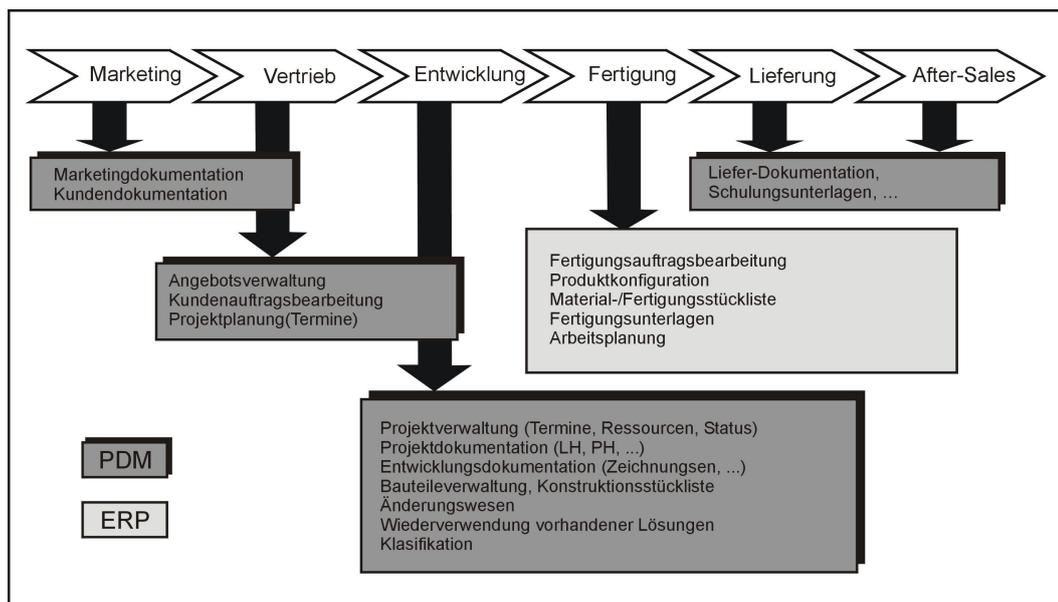


Abbildung 2.8: Prozesskette (Quelle: Siemens AG)

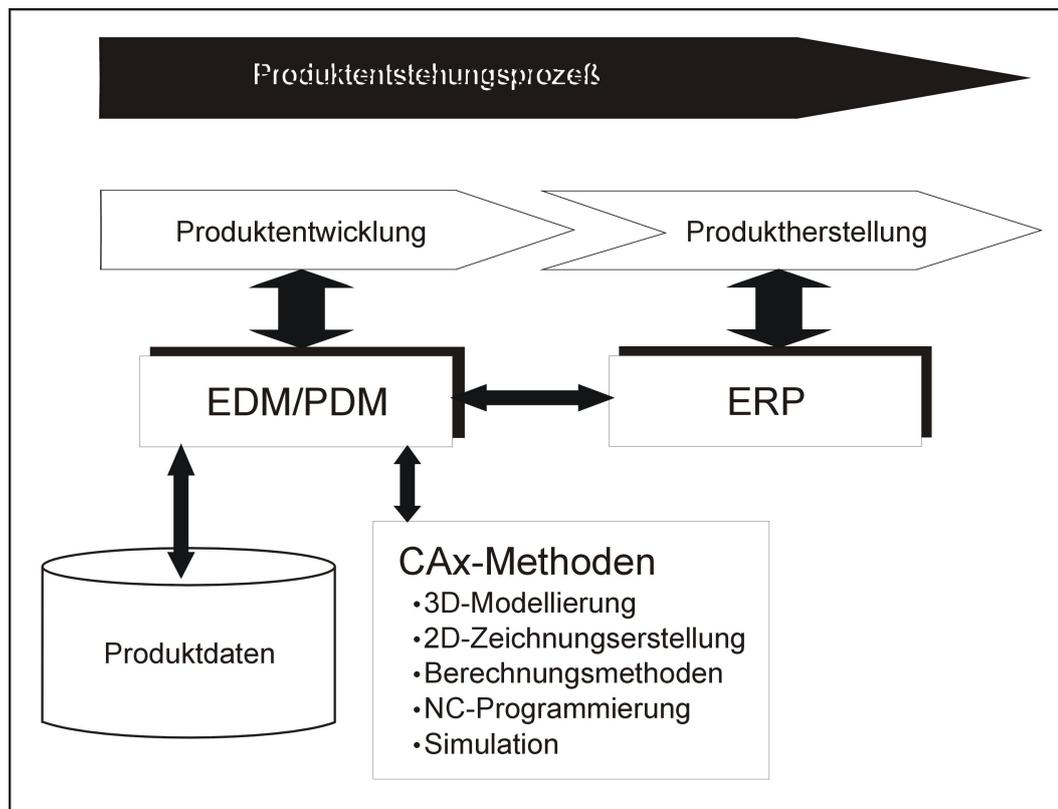


Abbildung 2.9: Systemarchitektur [Ler99, 324]

Der Begriff *Verküpfung* wird im Rahmen dieser Arbeit wertfrei verstanden. Mit ihm kann sowohl Kopplung als auch Integration gemeint sein.

Dabei wird unter der Kopplung von IT-Systemen lediglich die datentechnische Verküpfung zweier getrennter Anwendungssysteme verstanden. Die Anwendungssysteme besitzen in der Regel unterschiedliche und getrennte Datenstrukturen. Die Systeme werden über eine Zwischendatei und/oder durch eine Kopplungsprozedur, die Ausgabedaten des ersten Systems übernimmt und in Eingabedaten des zweiten Systems umwandelt mit einander verküpft. Wie in der Motivation dieser Arbeit aufgezeigt beträgt die Anzahl der Kopplungsprozeduren für  $n$  beteiligte Systeme beträgt  $[n \times (n - 1)]$ . Die jeweiligen Prozeduren sind sehr spezielle.

Die Integration unterscheidet sich von der Kopplung dadurch, dass dem Gesamtsystem, d.h. allen Anwendungen, ein einheitliches Modell mit einer allgemeinen für alle Anwendungen gültigen Daten- und Speicherungsstruktur zugrunde liegt. Die Anwendungen kommunizieren miteinander.

Zur Verküpfung lassen sich die nachfolgend aufgeführten Konzepte unterscheiden:

- **Verküpfungselement Mensch** Hierbei handelt es sich a priori nicht um eine EDV-technische Verküpfung, sondern um eine organisatorische Verbindung EDV-technisch nicht verküpfter Systeme. Diese Art der Verküpfung ist in der heutigen Zeit nicht mehr hinnehmbar. Der Mensch muß zwar in den Prozeß der Verküpfung integriert werden, darf aber nicht die Aufgabe der Verküpfung übernehmen.
- **Verküpfungselement formatierte Datei**  
In der Regel unterscheiden sich die Daten- und Speicherstrukturen von Applikationen. Die Applikation A schreibt Daten in einem festgelegten Format in eine Datei. Dieses Format muß auch der Applikation B bekannt sein, damit sie die Daten lesen und die Daten weiterverarbeiten kann.
- **Verküpfung durch Konverter**  
Diese Lösung der Kopplung von zwei Applikationen beinhaltet ein Konvertierungsprogramm, das die Aufgabe hat, die Datenstrukturen der Applikation A in die Datenstrukturen der Applikation B umzuwandeln.
- **Verküpfung über Datenbanken**  
Diese Lösung besteht aus dem Austausch von Daten über ein Datenbanksystem. Grundsätzlich bestehen hier zwei mögliche Kombinationen.

1. Die autonomen Applikationen besitzen eigene Datenbestände und legen in einer gemeinsamen Datenbasis nur solche Daten ab, die für die Kommunikation notwendig sind.
2. Alle Datenbestände von allen Applikationen können über ein zentrales, einheitliches Datenbanksystem benutzt werden.

Die zweite Variante hat in der Praxis eigentlich nur theoretischen Wert, da jedes Anwendungssystem sein eigenes Datenbankmanagementsystem (DBMS) verwendet. Nur bei reinen eigenentwicklungen oder in Ausnahmefällen besitzen alle Beteiligten Systeme das selbe DBMS.

Bei der Integration im Bereich der Produktentwicklung können in Anlehnung an [GHS96] und [Abe98] fünf verschiedene Ebenen unterschieden werden. Dies sind:

- die Organisationsebene (z. B. Mitarbeiter, Aufbauorganisation),
- Prozessebene (z. B. Ablauforganisation),
- Funktionsebene (z. B. IT-Werkzeuge),
- Informations- bzw. Datenebene (z. B. Modelle, Dokumente) und
- die Systemebene (z. B. Kommunikationsinfrastruktur).

Die systemtechnische Ebene bezieht sich auf die zugrundeliegende Kommunikationsinfrastruktur sowie die softwaretechnischen Mechanismen für die Kommunikation verschiedener IT-Systeme untereinander. Die Informations- bzw. Datenebene berücksichtigt die Ablage und den Austausch der benötigten Informationen, beispielsweise durch dateibasierten Austausch oder durch gemeinsamen Zugriff auf eine Datenbank. Die Funktionsebene bezieht sich auf die Integration der Funktionen der zu integrierenden Systeme. Die prozesstechnische Ebene fokussiert auf die Daten- und Informationsflüsse und die Organisationsebene auf die Abbildung von Informationen unter Berücksichtigung semantischer Aspekte der Aufbau- und Ablauforganisation, beispielsweise Nummerierungssysteme.

Zusätzlich können bei der Integration von IT-Systemen auf Datenebene verschiedene Granularitäten unterschieden werden. Die Integration von verschiedenen Erzeugersystemen (z. B. CAD) untereinander erfolgt auf sehr feingranularer Ebene, z. B. durch direkte Kopplungsbausteine. Hierbei ist der systemübergreifende Zugriff auf bestimmte Objekte oder auf Attribute bzw. Parameter möglich.

PDM-Systeme leisten eine indirekte Integration von IT-Anwendungen

## 2.5 Fazit

Ziel dieses Kapitel war es, einen Überblick über den Stand der Forschung im Bereich der ERP- bzw. PDM-Systeme sowie Strategien zur Kopplung dieser Systeme zu geben. Zunächst wurden die Entwicklungstendenzen des unternehmensweiten Informationsmanagements erörtert. Es bleibt festzuhalten, dass das Internet sich von einem reinen Informationsmedium zu einer technologischen Basis für ein unternehmensweites Informationsmanagement entwickelt. Anschließend wurde für die jeweiligen Bereiche ein einheitliches Begriffsverständnis geschaffen. Festzustellen ist, dass es keine einheitliche Definition des ERP-Begriffes gibt, sondern eine Vielzahl von Definitionen, die sich jedoch nur marginal voneinander unterscheiden. Eine aus Sicht des Autors besonders den Belangen dieser Arbeit gerecht werdende Definition des ERP wird von Clewett et. al. gegeben:

„ERP (Enterprise Resource Planning): A client/server application that is designed to coordinate and act as an information resource to an entire organisation or enterprise.“[CFM98, 345].

Es hat sich bei der Betrachtung von ERP-Systemen eindeutig gezeigt, dass diese Systeme eine starke betriebswirtschaftliche Ausrichtung besitzen. Eine Unterstützung von technischen Unternehmensfunktionen wird von diesen Systemen meist nur unzulänglich angeboten. Oft wird diese Unzulänglichkeit durch das Bereitstellen von Schnittstellen zu anderen technischen Informationsverarbeitungssystemen behoben.

Für den Bereich der PDM-Systeme ist ebenso keine einheitliche Begriffswelt festzustellen. Die Definition mit der größten Relevanz für den weiteren Verlauf dieser Arbeit und den darin betrachteten Aspekten von PDM-Systemen in Hinblick auf eine Kopplung zwischen ERP- und PDM-Systemen lautet:

„A Product Data Management system (PDM) is a software tool that manages engineering information, supports management of product configurations, and supports management of the product engineering process. The engineering information includes both database objects and “document” objects - sets of information stored in files that are opaque to the PDM system. This information may be associated with specific products or specific product design, or more generally with product families, production process or the engineering process itself. The engineering process support usually includes workflow management and concepts of engineering change and notification. In many

manufacturing organizations, the PDM is the central engineering information repository for product development activities“ [N.N00].

In Hinblick auf die prototypische Realisierung von ERP-OnTo-PDM in Kapitel 7 wurde in diesem Kapitel jeweils stellvertretend ein Produkt für die jeweilige Systemart vorgestellt. Dies sind das SAP R/3 System als Stellvertreter für die ERP-Systeme und das PDM-System SmarTeam der Firma Smart Solutions.

Weiterhin wurden in diesem Kapitel bestehende Strategien zur Kopplung von IT-Systemen eingehender betrachtet.

Um diesen Schwachstellen zu begegnen und einen verbesserten Datenaustausch zwischen IT-Systemen zu erreichen, wird in dieser Arbeit das Konzept der ERP-OnTo-PDM entwickelt. Das für diese Lösung fundamentale Konzept der Ontologien wird in dem folgenden Kapitel 3 eingehender behandelt.

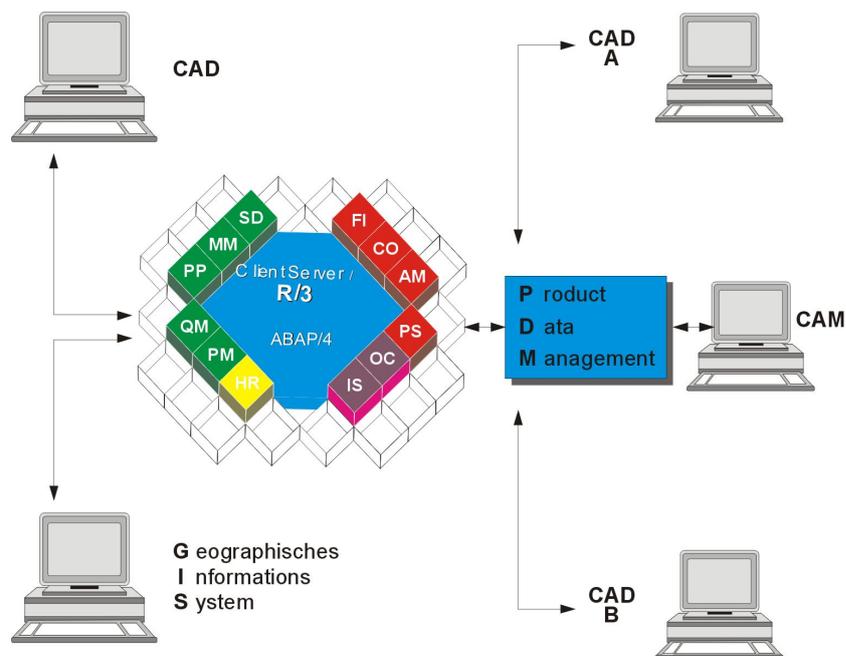
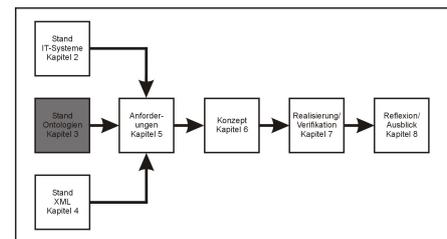


Abbildung 2.10: SAP Kopplung PDM

# Kapitel 3

## Stand der Forschung auf dem Gebiet der Ontologien

Wie im vorherigen Kapitel beschrieben, verschiebt sich der Aufgabenbereich der PDM-Systeme von der reinen Informationsverarbeitung immer weiter zu einer Wissensverarbeitung. Dies wird durch Abbildung 3.1 illustriert.



Eine vergleichbare Entwicklung wird aus Sicht des Autors auch bei ERP-Systemen einsetzen. Werden Daten zwischen diesen beiden Systemen ausgetauscht, ist es sinnvoll, der Entwicklung zu wissensverarbeitenden Systemen Rechnung zu tragen und sich Methoden und Konzepten zu bedienen, die über ein reines Konvertieren von Datenformaten der jeweiligen Systeme hinausgehen und auf einer logisch höheren Ebene angesiedelt sind. Ein solches Konzept lässt sich mittels Ontologien realisieren. Ziel dieses Kapitels ist es, eine präzise Definition des Begriffs *Ontologie* zu geben und verschiedene Arten von Ontologien vorzustellen. Der Fokus der Betrachtungen liegt dabei auf der Eignung der einzelnen Ontologie-Arten für den Einsatz als konzeptionelle Grundlage für die semantische Kopplung ERP-OnTO-PDM.

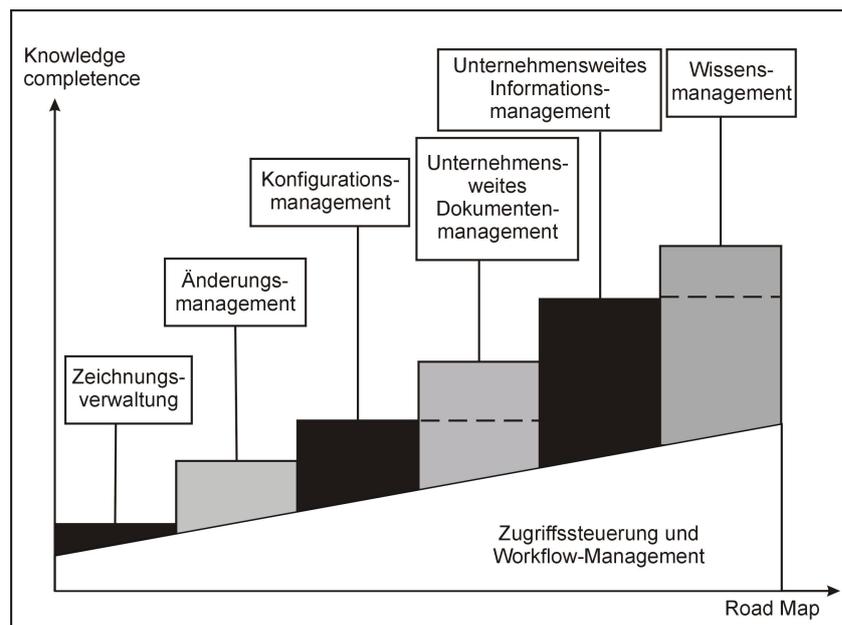


Abbildung 3.1: Vom Dokumentenmanagement zum Wissensmanagement - Ausbaustufen von EDM/PDM-Systemen

### 3.1 Begriffsbestimmung: Ontologie

Wie in der Einleitung zu diesem Kapitel skizziert wurde, gibt es kein einheitliches Verständnis über den Begriff *Ontologie*. Dies ist mit darauf zurückzuführen, dass der ursprüngliche Begriff aus der antiken griechischen Philosophie stammt. Erst seit Anfang der 90er Jahre des zwanzigsten Jahrhunderts sind Ontologien auch Forschungsgegenstand des Forschungsgebietes der KI. Es gibt aber eine gewisse Spannung zwischen dem philosophischen und informationswissenschaftlichen Verständnis und dem Gebrauch des Terminus *Ontologie*.

Gerade deshalb erscheint es dem Autor zu oberflächlich, ein intuitives Verständnis für die *Ontologie*<sup>1</sup> vorauszusetzen. Daher scheint es zweckmäßig, den Begriff *Ontologie(n)* zunächst unter historischem Aspekt zu betrachten.

Dazu wird zuerst das *Ontologieverständnis* im Bereich der Philosophie betrachtet, um darauf aufbauend das Konzept der *Ontologien* im Bereich der Künstlichen Intelligenz vorzustellen, welches im Rahmen dieser Arbeit eine zentrale Stellung einnimmt.

<sup>1</sup>In der Philosophie wird von *der Ontologie* gesprochen, nicht jedoch von *den Ontologien* [ZSS99]

### 3.1.1 Ontologie in der Philosophie

Der Begriff Ontologie wurde im Altertum erstmals in der Philosophie verwendet und lässt sich auf Aristoteles zurückführen. Er steht hier für die Lehre vom Sein bzw. vom Seienden [Gru93]. Er bezeichnet in der Philosophie eine bestimmte Theorie über die Beschaffenheit des Seiens und der Existenz. Es handelt sich also um eine Teildisziplin der Philosophie, die sich mit der Natur von Dingen und der Organisation der Realität beschäftigt. Bereits Aristoteles thematisierte in seiner *ersten Philosophie* Fragen nach dem Sein als Seiendem, d.h. nach seinem objektiven, vom menschlichen Erkennen unabhängigen Wesen und nach den ihm zukommenden Bestimmungen [ZSS99]. Nach Runggaldier/Kanzian [RK98] hat die Ontologie „die begriffliche Erfassung der *allgemeinsten Merkmale der Wirklichkeit* in die Wege zu leiten.“ Die Ontologie wurde in der Wissenschaft im zwanzigsten Jahrhundert vor allem durch Wissenschaftler wie Hartmann, Husserl, Heidegger, Sartre und Quine wiederentdeckt [ZSS99]. Durch die Ontologie sollen insbesondere nachfolgende Fragestellungen geklärt werden:

- Was macht das Dasein aus?
- Was sind die Eigenschaften alles Existierenden?

### 3.1.2 Ontologie in der Informatik

Kommunikation stellt das wesentliche, wenn nicht das einzige Mittel zum Austausch von Wissen und Informationen dar. Diese Kommunikation kann zwischen Menschen und/oder Maschinen stattfinden. Bei der Kommunikation zwischen Maschinen, bzw. wie im in dieser Arbeit untersuchten Fall zwischen zwei Anwendungssystemen, ist das Vorhandensein eines einheitlichen Vokabulars nicht ausreichend, um auch die einheitliche Bedeutung der ausgetauschten Informationen zwischen diesen Systemen zu gewährleisten. Es reicht also nicht aus, nur eine Syntax zum Datenaustausch zur Verfügung zu stellen, sondern es muss auch ein Konzept zur Schaffung eines einheitlichen semantischen Verständnisses zwischen den beteiligten Systemen bereitgestellt werden. Dieser Sachverhalt wird durch das semiotische Dreieck, welches in der Abbildung 3.2 dargestellt ist, verdeutlicht. Es wird ersichtlich, dass in der Sprache durch Symbole (Worte) ein Begriff referenziert wird. Dieser Begriff steht für einen bestimmten Gegenstand (Ding) in der realen Welt.

Diese Beziehung ist jedoch indirekt. Ein Wort ruft bei einem Interpreten (Mensch oder Maschine) eine bestimmte Bedeutung (Begriff) des Wortes hervor

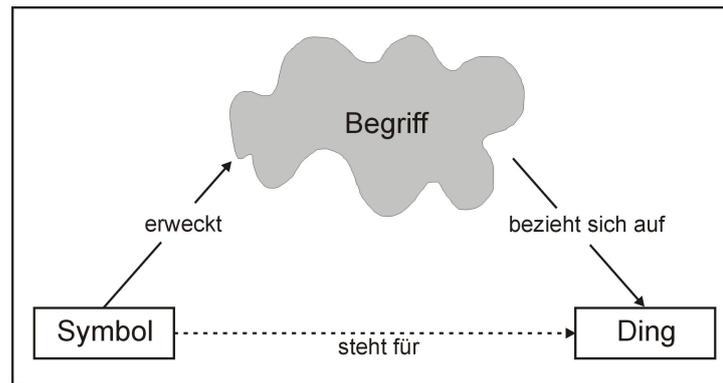


Abbildung 3.2: Semiotisches Dreieck [MSS01, 393]

und diesen Begriff bringt der Interpret dann mit einem bestimmten Gegenstand oder Sachverhalt der realen Welt in Verbindung. Es ist jedoch nicht gewährleistet, dass jeder Interpret auch immer die selbe Verknüpfung zwischen Bedeutung und realer Welt herstellt.

Als Haupthindernisse für eine effektive Kommunikation zwischen Personen, Organisationen und Softwaresystemen werden vor allem unterschiedliche Kontexte, Sprachen sowie Tools und Techniken der Beteiligten angesehen [UG96]. Es muss demzufolge neben der Syntax auch eine Semantik zu den verwendeten Begriffen bereitgestellt werden, um ein gemeinsames Verständnis zu erreichen. Dies führt zu einer Beseitigung von konzeptionellen und terminologischen Unklarheiten [OSS01].

Um ein verteiltes Verständnis in einer abgegrenzten Begriffswelt (domain of interest) zu schaffen, wurde im Bereich der KI-Forschung das Konzept der Ontologien entwickelt. Die Nutzeffekte der Verwendung von Ontologien sind in einer verbesserten Kommunikation, einer verbesserten Wiederverwendbarkeit und Interoperabilität, sowie einer erhöhten Zuverlässigkeit zu sehen.

Studiert man die Publikationen auf dem Gebiet der Ontologie-Forschung, so stößt man auf eine Vielzahl von unterschiedlichen Definitionen des Begriffs der Ontologie. Diese Definitionen unterscheiden sich zum Teil erheblich und es soll daher an dieser Stelle ein Überblick über die wichtigsten Definitionen und eine jeweilige Beurteilung erfolgen. Daneben soll eine für die Zwecke dieser Arbeit sinnvolle Definition des Ontologiebegriffs gegeben werden. Diese soll den besonderen Aspekten der Schaffung einer semantischen Kopplung zwischen ERP- und PDM-Systemen Rechnung tragen.

Im Bereich der KI sind unterschiedliche, konkrete Definitionen des Begriffs

Ontologie vorgeschlagen worden. Die bekannteste stammt von Gruber [Gru93]:

„An ontology is an explicit specification of a conceptualisation. The term is borrowed from philosophy, where an ontology is a systematic account of existence. For AI systems, what „exists“ is that which can be represented“

Eine Ontologie ist demnach für Gruber eine explizite Beschreibung eines Modells. Ausgehend von dieser Definition werden im Folgenden erweiterte und konkretere Definitionen gegeben. Für Wielinga und Schreiber bildet eine Ontologie die Grundlage zur Schlussfolgerung eines Agenten [WS93]. Bei Alberts wird hingegen besonders der Bezug einer Ontologie zu einer bestimmten Aufgabe oder Domäne hervorgehoben[Alb93]:

„An ontology for a body of knowledge concerning a particular task or domain describes a taxonomy of concepts for that task or domain that define the semantic interpretation of the knowledge“.

Durch diese Definition wird weiterhin betont, dass nicht die Terminologie, sondern die semantische Repräsentation der Terme eine Ontologie bilden. Diese Ansätze werden in der Definition von Van Heijst kombiniert[VSW97]:

„An ontology is an explicit knowledge level specification of a conceptualisation, i.e. the set of distinctions that are meaningful to an agent. The conceptualisation - and therefore the ontology - may be affected by the particular domain and the particular task it is intended for.“

Eine Ontologie im Sinne der KI-Forschung verkörpert somit eine bestimmte Art von Weltsicht, welche eine bestimmte Domain berücksichtigt. Diese Weltsicht ergibt sich als eine Menge von Konzepten (Entities, Attribute, Prozesse), ihren Definitionen und den zugehörigen Beziehungen. Ontologien sind üblicherweise in Taxonomien organisiert, d.h. sie werden in einer Baumstruktur mit mehrfacher Verbindung und disjunkten Unterkategorien dargestellt.

Eine Ontologie ist somit ein Konzept zur Darstellung und Formalisierung von Wissen. Es stellt je nach Anwendungszweck ein Modell der Welt bzw. eines Teils der Welt dar.

Der Hauptzweck für die Entwicklung und den Einsatz einer Ontologie liegt in der Schaffung eines *shared understanding* (gemeinsamen Verständnisses). Es

soll die Übertragbarkeit von Wissen unterstützt werden, so dass dieses Wissen mehrfach, auch über Sprachbarrieren hinweg, genutzt werden kann.

Eine Ontologie kann demnach implizit sein, z.B. kann sie in den Köpfen einiger Leute (einer Person) oder in einem Stück Software vorliegen. Im Rahmen dieser Arbeit soll die Ontologie für die Schaffung einer ERP-PDM-Kopplung eine explizite Aufzählung oder Repräsentation einer Konzeptionalisierung bzw. eines Modells widerspiegeln.

Durch die semantische Beschreibung der verwendeten Konzepte machen Ontologien vor allem versteckte Beziehungen zwischen ihnen explizit, die dann zur Interpretation der Information genutzt werden können.

Eine Ontologie im Verständnis dieser Arbeit und in Hinblick auf die Realisierung der Kopplung mittels ERP-OnTo-PDM im Kapitel 7 hat nachfolgende Bestandteile [PB99]:

- **Konzepte**

Sind die Elemente der Ontologie. Sie werden über Relationen miteinander verknüpft und mit Attributen beschrieben.

- **Relationen**

Stellen die Beziehung der Konzepte zueinander dar.

- **Attribute**

Beschreiben die Konzepte.

- **Axiome**

Sind allgemeingültige Sätze (z.B. *Ein Spieler, der bei einem bestimmten Verein ist, hat den Trainer, der bei diesem Verein beschäftigt ist*)

Die strukturierte Entwicklung von Ontologien, auch als *Ontology Engineering* bezeichnet, ist aktuelles Forschungsthema insbesondere im Bereich der Wiederverwendung von Wissen. Ziel der Entwicklung ist es sicherzustellen, dass die entwickelten Ontologien bestimmte Eigenschaften aufweisen, die im engen Zusammenhang mit dem Grad der Wiederverwendbarkeit stehen. In der Literatur [Gru95], [GP96] werden die nachfolgenden Kriterien übereinstimmend als besonders wichtig für die Entwicklung wiederverwendbarer Ontologien angesehen:

- **Klarheit**

Eine Ontologie soll die Bedeutung der beschriebenen Konzepte klar, verständlich und objektiv (d.h. unabhängig vom Umfeld der konkreten Verwendung) darstellen.

- **Kohärenz**  
Eine Ontologie soll möglichst alle Schlussfolgerungen zulassen, die konsistent mit den enthaltenen Definitionen sind.
- **Vollständigkeit**  
Die enthaltenen Konzepte sollen möglichst vollständig über notwendige und hinreichende Bedingungen beschrieben werden.
- **Erweiterbarkeit**  
Die Konzepte einer Ontologie sollen erweitert und verfeinert werden können, ohne dass bestehende Teile der Ontologie verändert werden müssen.
- **Minimale Umsetzungsentscheidungen**  
Eine Ontologie soll auf der Wissensebene beschrieben werden und möglichst wenig Rücksicht auf Aspekte der Realisierung nehmen.
- **Minimale Festlegung**  
Eine Ontologie soll nur den relevanten Teil der modellierten Welt beschreiben, um unnötige Festlegungen zu vermeiden.
- **Unterscheidbarkeit**  
Die Konzepte einer Ontologie sollen disjunkt und unterscheidbar sein.

Wie dargestellt, gibt es in der Literatur eine Diskussion über den Begriff Ontologie. Ein Grund für die Verwirrungen besteht darin, dass verschiedene Terminologien in unterschiedlichem Kontext gebraucht werden. Zum Beispiel gibt es einen engen Zusammenhang zwischen den konzeptionellen Schemata einer Datenbank und einer Ontologie. Aus diesem Grund scheint eine Abgrenzung zu anderen Konzepten der Strukturierung, wie sie z.B. im Ingenieurwesen verwendet werden, erforderlich.

Im Rahmen dieser Arbeit soll eine Definition verwendet werden, die sich an den oben aufgeführten Definitionen orientiert, jedoch zusätzlich den formalen Charakter einer Ontologie betont. Dieser formale Charakter macht es durch die eindeutige Festlegung der Bedeutung elementarer Modellierungsprimitiva möglich, Ontologien effizient zur Wiederverwendung von Wissensmodellen einzusetzen. Die als Grundlage für diese Arbeit verwendete Definition des Begriffes Ontologie ist daher die nachfolgende:

„An ontology is an explicit knowledge-level formalisation of a conceptualisation used to define the semantic interpretation of knowledge-modelling primitives.“

### 3.1.3 Abgrenzung gegenüber verwandten Konzepten

Eines der Hauptaufgabenfelder von Informationsmanagementsystemen ist es, die in ihnen abgelegten Informationen effektiv und effizient zugreifbar zu machen. Diese Aufgabe wird durch eine Gliederung bzw. Strukturierung der Informationen erleichtert. Gerade im Bereich der technischen Informationsverarbeitung steht durch die Gruppentechnologie<sup>2</sup> eine Reihe von Methoden zur Systematisierung, Strukturierung sowie zur Vereinheitlichung zur Verfügung [Ger00, 16 ff.]. Die in Hinblick auf den Datenaustausch wichtigste Methode zur Strukturierung bilden die sogenannten Ordnungssysteme. Weitere Methoden sind beispielsweise die Clusteranalyse, Fourieranalyse oder die Fuzzy Logic. Insbesondere die Ordnungssysteme besitzen für den Bereich der Informationsverarbeitung eine hohe Relevanz und sollen hier eingehender betrachtet werden. Eine Menge von Begriffen, die untereinander Beziehungen aufweisen, werden nach DIN 2230 als ein Begriffssystem bezeichnet. Wird ein Begriffssystem mit dem Ziel der Systematisierung nach bestimmten Ordnungskriterien entwickelt und eingesetzt, bezeichnet man dieses nach DIN 2331 als Ordnungssystem. Ein Ordnungssystem ist demnach ein Hilfsmittel zur Dokumentation von Informationen und besitzt die Aufgabe, einen ungeordneten Zustand mittels festgelegter Ordnungsmerkmale in einen geordneten Zustand zu überführen. Ordnungssysteme lassen sich entsprechend ihres strukturellen Aufbaus, also der Art der Beziehung zwischen den Ordnungsbegriffen untereinander, wie folgt untergliedern [Mew73],[Ing78],[EHSS91, 40 f]:

- **Hierarchische Systeme (z.B. Klassifizierungssysteme)**

Diese Klasse von Ordnungssystemen zeichnet sich durch eine strenge Hierarchie aus. Begriffe mit gleichen oder ähnlichen Merkmalen werden durch einen Oberbegriff zusammengefasst. Durch diese Strukturierung entstehen Überordnungs- bzw. Unterordnungsverhältnisse, welche durch Baum- bzw. Netz-Repräsentationen abgebildet werden können. Nach Art des strukturellen Aufbaus solcher Systeme werden monohierarchische sowie polyhierarchische Systeme unterschieden. Bei den monohierarchischen Systemen ist ein Begriff einer untergeordneten Hierarchiestufe genau einem Begriff einer übergeordneten Hierarchiestufe zugeordnet. Bei den polyhierarchischen Systemen kann ein Begriff einer untergeordneten Hierarchiestufe mehreren Begriffen einer übergeordneten Hierarchiestufe zugeordnet werden. Hinsichtlich der

---

<sup>2</sup>Unter Gruppentechnologie ist eine Methodik für das Zusammenfassen von Objekten (z.B. Bauteile oder Baugruppen) zu Gruppen anhand von funktionellen, konstruktiven oder fertigungstechnischen Merkmalen bzw. Eigenschaften zu verstehen [Vaj92]

Informationsbereitstellung wird in manuelle bzw. automatische Codierung differenziert.

- **Ahierarchische Systeme (z.B. Schlagwörter bzw. Indexkataloge)**

Ahierarchische Systeme sind voneinander unabhängige Zusammenstellungen von Informationen. Die Begriffe sind hierbei gleichrangig und ohne eine inhärente Struktur. Sie können aber beispielsweise alphabetisch geordnet werden. In diese Klasse gehören demnach alphabetisch nach beschreibenden Begriffen sortierte Schlagwortkataloge oder willkürliche Zusammenstellungen von Gegenständen in Tabellen. Diese einfache Art der Dokumentation von verschiedenen Gegenständen hat den Vorteil der gesamtheitlichen Erfassung. Nachteilig ergibt sich aber ein notwendiger, enormer Speicheraufwand, bei gleichzeitig geringer Auflösung bzw. Aussagekraft.

- **Teilhierarchische Systeme (z.B. Thesauri [DIN461-1] DIN1463(2))**

Sie stellen eine Kombination aus hierarchischen und ahierarchischen Ordnungssystemen dar. In Gegensatz zu den ahierarchischen Systemen bieten teilhierarchische Ordnungssysteme den Vorteil der bedingten Verknüpfung von Ordnungsmerkmalen. Dadurch bietet sich die Möglichkeit der Grobgruppierung von Objekten mit dem Vorteil, dass Gruppen nach gewünschten Eigenschaften gezielt ausgewählt werden können. Ein typischer Vertreter dieser Kategorie ist der Thesaurus. Darunter versteht man eine alphabetische Sammlung von Begriffen eines Fachgebietes mit systematischer Ordnung. Die einzelnen Begriffe können hierarchisch dargestellt werden. Thesauri werden in der DIN-Norm 1463 definiert [DIN87a],[DIN87b]. Beispielhaft ist der Aufbau eines Thesaurus in Abbildung 3.3 dargestellt.

Ein Thesaurus ermöglicht es, Informationen zur Unterstützung des Suchvorgangs teilhierarchisch geordnet abzulegen. Dies wird über die Verwendung von Schlagwörtern ermöglicht, die einmal der direkten Nutzung als Suchschlüssel dienen (Deskriptoren) aber auch zusätzliche Informationen (Nicht Deskriptoren wie z.B. Synonyme) enthalten. Somit ist die Möglichkeit der Eingrenzung eines Objektes über gewählte Merkmale gegeben.

Abbildung 3.4 gibt einen schematischen Überblick über die verschiedenen Klassen von Ordnungssystemen. Innerhalb des Ingenieurwesens und dort besonders im Anlagen und Maschinenbau finden Ordnungssysteme die häufigste Verwendung in

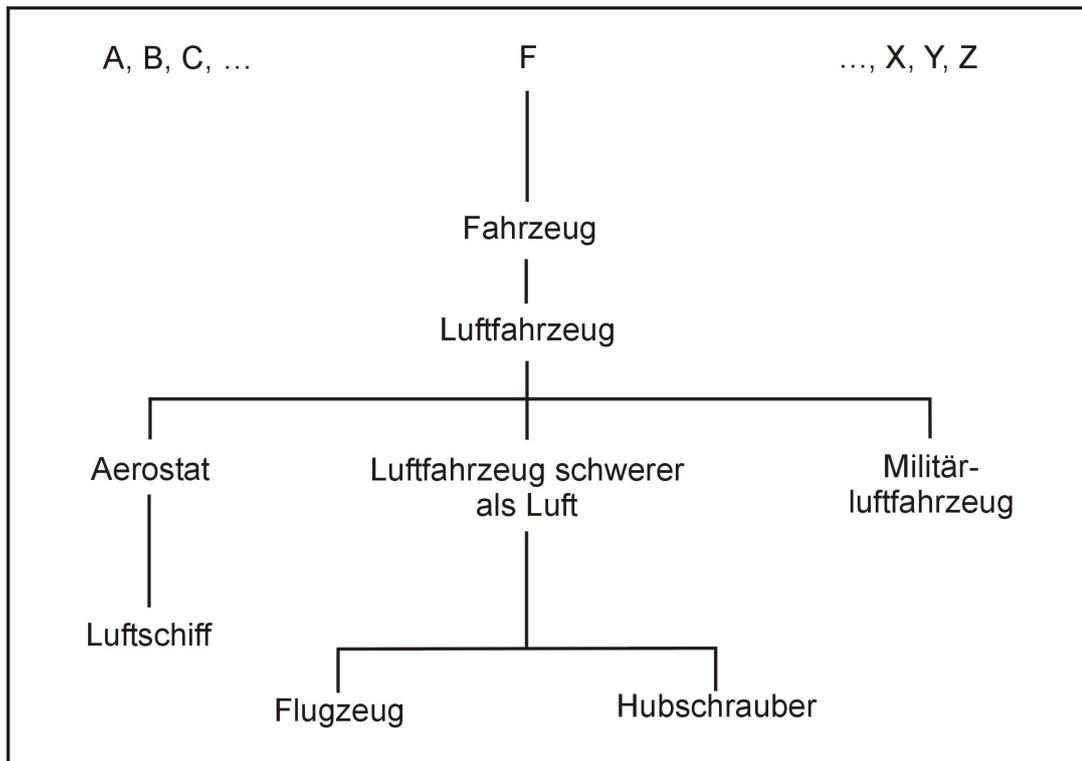


Abbildung 3.3: Beispiel eines Thesaurus nach [DIN87b]

Verbindung mit Sachmerkmal-Leisten<sup>3</sup> (SML). Sachmerkmal-Leisten sind eine spezielle Ausprägung eines hierarchischen Ordnungssystems. SMLs sind beschränkt auf eine einstufige hierarchische Ordnung.

Ontologien können entweder implizit oder explizit spezifiziert sein. Für diese Arbeit sind die explizit spezifizierten Ontologien von entscheidender Bedeutung. Diese können wiederum eine Vielzahl von Formen annehmen, sie werden jedoch notwendigerweise immer eine Liste von Termen und eine Spezifikation derer Bedeutung (Definition) beinhalten [Fen01]. Demnach können die oben angeführten Ordnungssysteme als eine Unterklasse von Ontologien angesehen werden. Eine Ontologie erlaubt, je nach Realisierung über die oben beschriebenen Eigenschaften von Ordnungssystemen hinaus, eine semantische und logische Spezifikation der Terme.

Der Grad an Formalität, mit der die Liste von Termen und Bedeutungen in einer Ontologie beschrieben wird, kann sehr unterschiedlich ausfallen. Es können

<sup>3</sup>Eine Sachmerkmal-Leiste nach DIN 4000 Teil 1 ist die Zusammenstellung und Anordnung der für eine Gegenstandsgruppe relevanten Sachmerkmale [DIN4000-1]

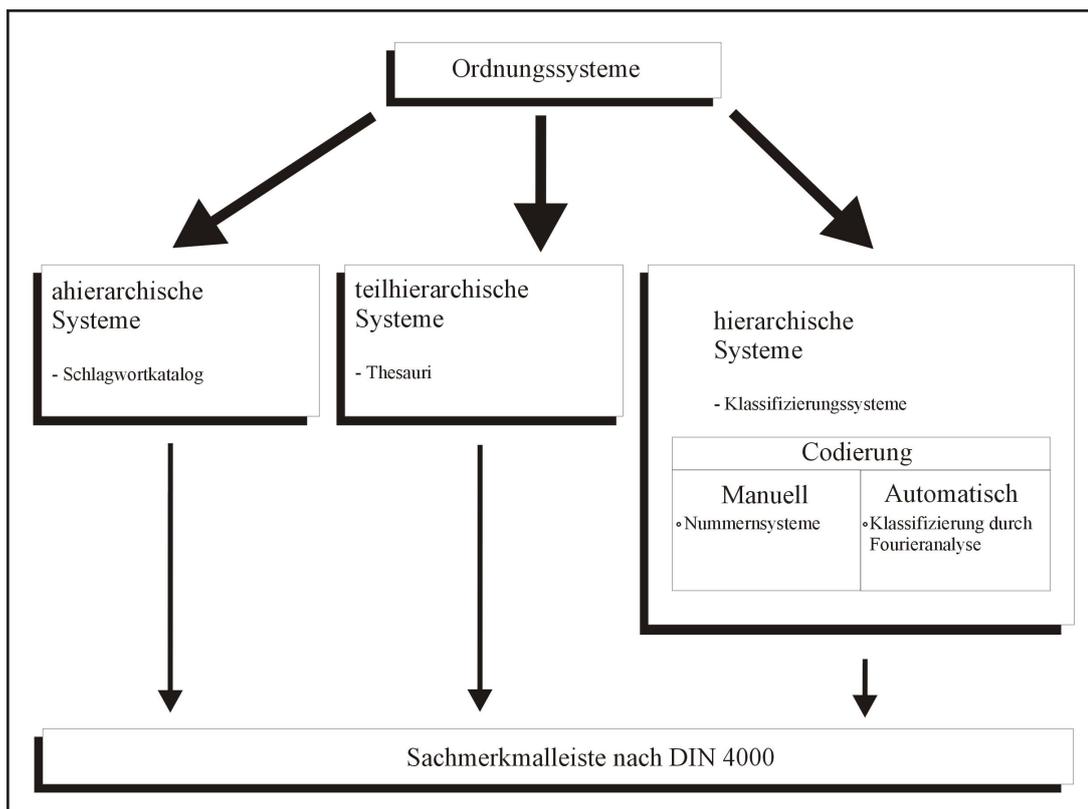


Abbildung 3.4: Klassen von Ordnungssystemen nach [Ing78]

folgende Formalitätsgrade von Ontologien unterschieden werden [KBS94]:

- **highly informal**  
Ausgedrückt durch natürliche Sprache
- **semi-informal**  
Ausgedrückt in einer strikteren und strukturierteren Form der natürlichen Sprache, Klarheit soll durch Reduktion von Zweideutigkeiten geschaffen werden (z.B. Textversion der Enterprise Ontology)
- **semi-formal**  
Ausgedrückt in einer künstlichen, formal definierten Sprache (die Ontolingua Version der Enterprise Ontology)
- **highly formal**  
Akribisch genau definierte Terme mit einer formalen Semantik, Theoremen und geprüft auf Eigenschaften wie Gültigkeit (Soundness) und Vollständigkeit.

Durch die Abbildung 3.5 werden die unterschiedlichen Grade an Formalität, der eine Ontologie unterliegen kann, veranschaulicht.

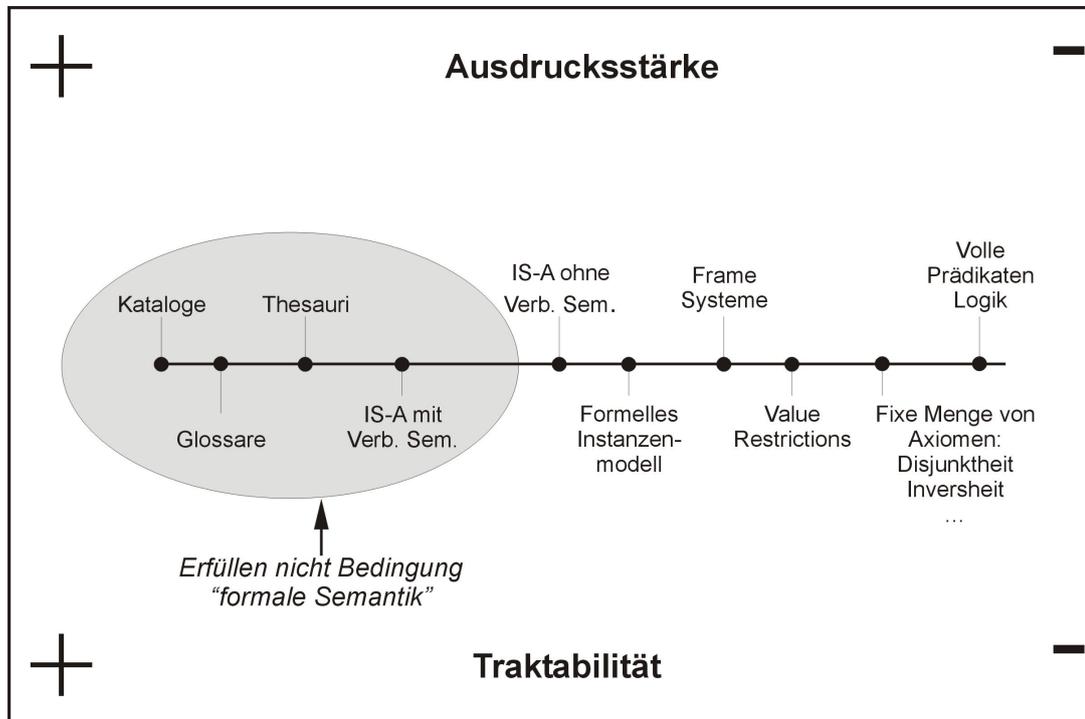


Abbildung 3.5: Einordnung

Durch die Abbildung wird auch verdeutlicht, dass durch das Verwenden von formalen Definitionen in einer Ontologie formal höherwertige Ordnungssysteme geschaffen werden können als die oben beschriebenen. Aufgrund der Tatsache, dass Definitionen von Konzepten deklarativ und formal sind, können bei einer Ontologie im Unterschied zu den beschriebene Ordnungssystemen unerwartete Interpretationen von Konzepten und Relationen vermieden werden. Außerdem wird es möglich, einen Datenaustausch auf einer logisch höher angesiedelten Ebene vorzunehmen und den Grad an Wiederverwendbarkeit der Schnittstelle durch die Einrichtung eines für alle beteiligten Systeme universellen Interlingua-Server zu erhöhen. Ein wesentlicher Vorteil bei der Verwendung von Ontologien zur Integration von Informationsquellen liegt demnach im Bereich der intelligenten Verwendung der verfügbaren Informationen.

## 3.2 Ontologiearten

Wie im vorherigen Abschnitt beschrieben, können sich die Konzepte, die mittels einer Ontologie spezifiziert werden sollen, in Hinblick auf ihren Formalitätsgrad stark unterscheiden. Weiter werden Ontologien auch hinsichtlich ihres Einsatzgebietes unterschieden. Abhängig von ihrem Granularitäts-Grad können verschiedene Typen von Ontologien identifiziert werden. In der Literatur werden so unter anderen von [Gua98],[VSW97] sowie [Fen01, 12f.] verschiedene Ontologie-Typen unterschieden. Eine Unterscheidung wird nicht zuletzt wegen der unterschiedlichen Rollen, die Ontologien im Entwicklungsprozess von Anwendungssystemen einnehmen, notwendig.

Die Eignung des jeweiligen Ontologie-Typen und dessen Nutzung im Rahmen des Knowledge Engineering Prozesses wurden z.B. im KACTUS-Projekt [WSJ<sup>+</sup>94], [GP96], [VSW97] untersucht.

Guarino unterscheidet anhand des Grades der Generalität die in Abbildung 3.6 dargestellten Arten von Ontologien [Gua98].

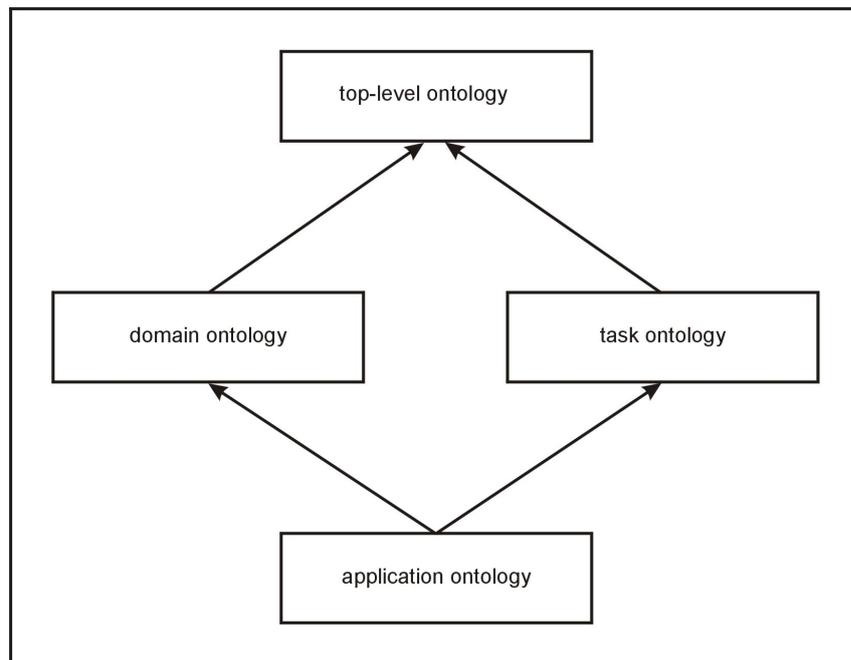


Abbildung 3.6: Arten von Ontologien nach [Gua98, 8]

Aufgrund der Tatsache, dass sich mittels einer Ontologie Konzepte unterschiedlicher Art darstellen und modellieren lassen, kann hiermit sowohl allgemeines als

auch domänenspezifisches Wissen abgebildet werden. In Abhängigkeit vom Gegenstand der Konzeptualisierung lassen sich verschiedene Kategorien von Ontologien unterscheiden [VSW97].

Eine Zusammenfassung und Diskussion der verschiedenen Ontologie-Typen, welche sich in der Literatur herauskristallisiert haben, soll nachfolgend in angemessener Kürze erfolgen.

Zunächst können zwei Dimensionen zur Klassifikation von Ontologien unterschieden werden [VSW97].

Hinsichtlich der ersten Dimension können Ontologien nach der Bedeutung und nach der Struktur der Konzeptualisierung abgegrenzt werden. Die zweite Dimension klassifiziert Ontologien nach dem Subjekt der Konzeptualisierung. Hinsichtlich der ersten Dimension werden demnach folgende drei Kategorien unterschieden [Gua96]:

- **Terminological Ontologies (Terminologische Ontologien)**  
spezifizieren die Terme, die zur Repräsentation des Wissens eines Anwendungsgebiets (domain of discourse), benutzt werden. Diese Art von Ontologien ähneln Lexika. Ein Beispiel einer Terminological Ontologie für das Anwendungsgebiet der Medizin ist das semantische Netz im UMLS (Unified Medical Language System).
- **Information Ontologies (Informations Ontologien)**  
spezifizieren die Datenstrukturen innerhalb einer Datenbank. Als ein Beispiel für diese Klasse von Ontologien kann ein Datenbank-Schema gesehen werden.
- **Knowledge Modelling Ontologies (Wissensmodellierende/ -repräsentierende Ontologien)**  
spezifizieren die Konzeptualisierung von Wissen. Verglichen mit den Information Ontologies besitzen die Knowledge Modelling Ontologies eine komplexere interne Struktur und sind damit eine mächtigere Repräsentation. Des Weiteren sind diese Ontologien oft auf den speziellen Gebrauch des Wissens, welches sie repräsentieren, angepasst. Im Bereich der Forschung und Entwicklung von wissensbasierten Systemen ist dies die wichtigste Klasse von Ontologien.

Die zweite Dimension, die zur Klassifikation unterschieden werden kann, ist das Subjekt der Konzeptualisierung. Hier werden fünf Kategorien unterschieden:

- **Application Ontologie (Anwendungs-Ontologie)**

Mittels dieser Art von Ontologie werden Konzepte und Wissen beschrieben, welche zur Lösung einer bestimmten Aufgabe, in einer gegebenen Domäne, mittels einer bestimmten Anwendung notwendig sind [TEG<sup>+</sup>95]. Diese Ontologien beinhalten demzufolge alle für die Modellierung des Wissens, welches für eine bestimmte Anwendung notwendig ist, benötigten Definitionen.

Anwendungs-Ontologien werden besonders in Hinblick auf die Erstellung eines bestimmten Anwendungssystems geschaffen. Dementsprechend sind Anwendungs-Ontologien auch eine Mischform zwischen einer Domain-Ontologie und einer Generic(Basis)-Ontologie, welche problemspezifisch erweitert wurde. Die Erweiterungen ergeben sich aus der zu lösenden Aufgabe und tragen den zur Lösung verwendeten Methoden Rechnung. Demnach beschreiben Anwendungs-Ontologien sowohl ein spezielles Anwendungsgebiet (Domain-Ontologie) als auch die zu diesem Anwendungsgebiet gehörenden Aufgaben (Task-Ontologien). Guarino beschreibt folgerichtig eine Anwendungs-Ontologie als gemeinsame Spezialisierung von Domänen- und Methoden-Ontologie [Gua96]. Die Konzepte korrespondieren oft mit speziellen Rollen, die Entities in einer Domain während der Ausübung einer Aktivität in diesem Anwendungsgebiet einnehmen. Die problemspezifischen Erweiterungen einer Anwendungs-Ontologie führen dazu, dass diese in der Regel nicht wiederverwendbar sind. Vielmehr können Anwendungs-Ontologien als Resultat der Wiederverwendung und Anpassung anderer Ontologien, die zu einer abstrakten Beschreibung des zu erstellenden wissensbasierten Systems führen, angesehen werden. Ein Beispiel könnte eine Ontologie für eine Diagnose technischer Systeme sein.

- **Domain Ontologie (Domänen-Ontologie)**

In dieser Klasse von Ontologien werden Konzepte bzw. Wissen modelliert, welche speziell für die Analyse und Beschreibung eines bestimmten Problembereichs (Domäne) dienen. Domänen können beispielsweise Elektronik, Medizin, Mechanik oder Kraftwerke mit Dampfturbinen sein [EPM94]. Sie geben mittels Constraints Vorgaben über die Struktur und mögliche Ausprägungen des zu erstellenden Wissensmodells des Problembereichs. Darüber hinaus stellt eine Domänen-Ontologie eine gemeinsame Sprache zur Beschreibung der Domäne dar und unterstützt damit die Akquisition von Domänenwissen. Während Domänen-Ontologien eher der Analyse eines Problembereichs dienen, zielen Anwendungs-Ontologien bereits auf die Erstellung eines Systems, welches eine konkrete Aufgabenstellung lösen soll. Die Konzepte einer

Domänen-Ontologie werden häufig von Konzepten einer vorhandenen Basis-Ontologie durch Vererbung abgeleitet. Hierdurch wird die Weiterverwendung von Strukturen und Eigenschaften einer Basis-Ontologie mit Änderungsaufwand ermöglicht.

- **Top-Level bzw. Basis-Ontologie**

Je nach Autor wird diese Klasse von Ontologien als Top-Level Ontologie, Basis Ontologie, Commonsense Ontologie bzw. Generic Ontologie bezeichnet. Sie ähneln Domänen-Ontologien. Die durch sie spezifizierten Konzepte sind jedoch weniger stark in ihrer Gültigkeit eingeschränkt. Die Ontologie beschreibt grundlegende Konzepte, die in unterschiedlichen Domänen und zur Lösung unterschiedlicher Aufgaben verwendet werden (z. B. physikalische Größen und Maße [Gro97]). Sie zielen auf das Modellieren des allgemeingültigen Weltwissens. Durch sie werden Konzepte wie zum Beispiel Zeit, Raum, Zustand und Ereignis [FNH97], [PS] spezifiziert. Die Konzepte sind unabhängig von konkreten Problemstellungen oder bestimmten Domänen. Daher können Basis-Ontologien in verschiedenen Problembereichen Gültigkeit besitzen. Beispielsweise kann eine Ontologie für Meteorologie in verschiedenen technischen Problembereichen angewendet werden [BA97]. Konzepte in einer Domänen-Ontologie sind häufig eine Spezialisierung von Konzepten einer Basis-Ontologie. Demzufolge ist die Unterscheidung zwischen Basis-Ontologien und Domänen-Ontologien fließend. Eine Unterscheidung ist intuitiv möglich und für die Erstellung von wiederverwendbaren Ontologie-Bibliotheken sinnvoll. Die Bedeutung von Basis-Ontologien liegt vor allem in ihrer hohen Wiederverwendbarkeit. Sie beschreiben Konzepte, die so fundamental sind, dass sie zum einen in fast jeder Anwendung eine Rolle spielen und zum anderen bereits ausreichend verstanden und standardisiert sind, um eindeutig beschrieben werden zu können. Ein gutes Beispiel hierfür sind die natürlichen Zahlen. Für den Prozess der Wissensmodellierung sind verfügbare Basis-Ontologien von entscheidender Bedeutung, da die Formalisierung derartiger Konzepte sehr zeitintensiv ist.

- **Representation Ontologies (Struktur-/ Repräsentations-Ontologie)**

Diese Klasse von Ontologien spezifiziert explizit die Formalismen der Wissensrepräsentation. Sie repräsentiert die grundlegenden Elemente, welche bei der Beschreibung von Konzeptualisierungen verwendet werden. Repräsentations-Ontologie stellen ein Repräsentations-Framework zu Verfügung, ohne Behauptungen über die Welt zu treffen. Domänen- und Basis-Ontologien können durch den Gebrauch von Primitiven, die durch eine

Repräsentations-Ontologie bereitgestellt werden, beschrieben werden. Eine der aus der Literatur bekanntesten Repräsentations-Ontologien ist die durch Gruber beschriebene *Frame Ontology* [Gru93], welche durch die Spezifizierung von Konzepten wie Frames, Slots sowie Slot beschränkende Constraints die Modellierung in einer objektorientierten bzw. framebasierten Sichtweise ermöglicht. Struktur-Ontologien bilden die Grundlage für die Erstellung von Wissensmodellen, indem sie die hierfür verfügbare Sprache festlegen. Besondere Bedeutung kommt Struktur-Ontologien zu, wenn unterschiedliche Repräsentationsformalismen zur Beschreibung der verschiedenen Teile eines Gesamtmodells verwendet werden. In diesem Fall kann die Integration dieser Teile über eine gemeinsame Struktur-Ontologie erfolgen [Gru93] [AH94] [VSW97]. Die integrativen Fähigkeiten von Struktur-Ontologien sind vor allem in komplexen Domänen wichtig, da solche Domänen meist nur durch eine Kombination unterschiedlicher Repräsentationsformalismen und Modelltypen erfasst werden können.

- **Method bzw. Task Ontologies (Methoden- bzw. Aufgaben-Ontologien)**

Diese Klasse von Ontologien bildet das Gegenstück zur Klasse der Domänen-Ontologien, indem sie die relevanten Aspekte einer zu lösenden Aufgabe bzw. die Methode zur Lösung dieser Aufgabe unabhängig von einer konkreten Domäne beschreibt (z. B. modellbasierte Diagnose) [Fen97] [SEG<sup>+</sup>96]. Aufgaben-Ontologien beschreiben die spezifischen Terme für eine bestimmte Aufgabe. Methoden-Ontologien sind Voraussetzung für die Beschreibung wiederverwendbarer Problemlösungsmethoden, da sie es ermöglichen, die Anwendbarkeit einer Methode in einer Domäne zu untersuchen [GTRM94]. Außerdem ermöglichen sie einen Vergleich mit der zu lösenden Aufgabe, wodurch sich ermitteln lässt, ob eine Methode zur Lösung der gestellten Aufgabe geeignet ist.

### 3.3 Fazit

Während Philosophen eine Ontologie als eine sprachenunabhängige und abstrakte Beschreibung betrachten, wird sie in der Welt der Wissensverarbeitung als ein Softwareerzeugnis verstanden, das für einen bestimmten Gebrauch und für eine bestimmte Computerumgebung erstellt wurde.

Eine Ontologie kann damit als eine Schnittstelle zwischen einer Informations-



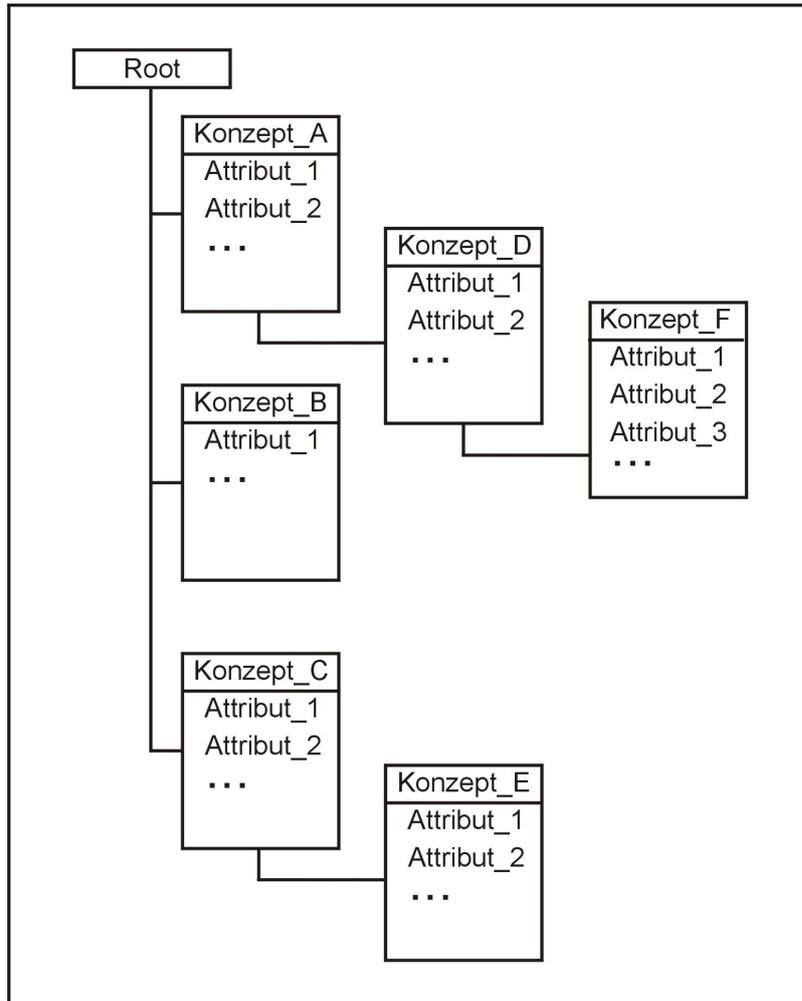
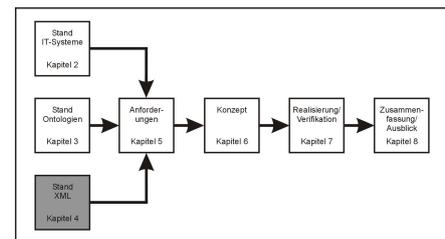


Abbildung 3.8: Prinzip einer Ontologie

# Kapitel 4

## Stand der Forschung auf dem Gebiet Internettechnologie

Mit der zunehmenden Verbreitung und verstärkten kommerziellen Nutzung des Internets sind die Ansprüche an dieses Kommunikationsmedium stark gestiegen. Es wird nicht mehr nur zur Informationssuche eingesetzt, sondern es wird erwartet, dass die gefundenen Informationen auch gleich weiterverarbeitet werden können. Als der primäre Internet-Dienst für Business-Lösungen hat sich das World Wide Web (WWW)<sup>1</sup> mit seiner grafischen Benutzeroberfläche und der damit benutzerfreundlichen Bedienbarkeit etabliert. Bisher wurde für die Implementierung von Inhalten des Webs hauptsächlich die Auszeichnungssprache und das Dateiformat HTML und das damit verbundene Hypertext Transferprotocol (HTTP) verwendet. Ohne die Entwicklung der beiden Web-Standards (HTML und HTTP) hätte das Internet nicht den Erfolg und Stellenwert einnehmen können, den es heute erreicht hat. Für die originären Aufgaben, für die beide Standards entwickelt wurden, sowie den ersten E-Business-Anwendungen waren sie auch ausreichend. Zu den originären Zielen beider Standards gehörte die Schaffung eines Informationsaustausches zwischen Wissenschaftlern. Es sollten wissenschaftliche Dokumente online sichtbar gemacht werden. Diese Texte sollten einfache Textformatierungen und das Einbinden von Grafiken ermöglichen. Des Weiteren sollten Verknüpfungen



<sup>1</sup>Das World Wide Web wird auch kurz als Web bezeichnet. Im Rahmen dieser Arbeit wird dieses Akronym ab jetzt auch zur Bezeichnung des Internet-Dienstes WWW verwendet.

(Links) mittels Hyperlinks möglich sein.

Die Möglichkeiten sowie die Spezifikation von HTML sind für die neuen Anforderungen jedoch nicht ausreichend. Eine genaue Darstellung der Schwachstellen und Unzulänglichkeiten von HTML würde den Rahmen dieser Arbeit sprengen und wird in zahlreichen Publikationen ausreichend betrachtet. Daher wird an dieser Stelle nur auf die wichtigsten Schwachstellen von HTML eingegangen und für eine ausführliche Darstellung auf einige dieser Publikation verwiesen.

Als Schwachstellen von HTML sind vor allem die nachfolgenden Punkte zu nennen [Mic99, 26f.]:

- HTML besitzt praktisch keine Möglichkeit, um die Semantik von Daten abzubilden.
- Von HTML gibt es zu viele proprietäre Varianten - dies führt zu partieller Inkompatibilität.
- Nicht ausreichende Trennung von Inhalt (Struktur) und Layout
- Nicht ausreichendes Konzept zur Verknüpfung (Linkkonzept). Anforderungen von komplexen Informationsnetzen und Informationsmanagements können von HTML nicht erfüllt werden.

Aus diesen Gründen wurde 1996 vom World Wide Web Consortium (W3C) die „eXtensible Markup Language“ (XML) als ein Standard für eine Meta-Auszeichnungssprache verabschiedet. In diesem Kapitel wird dargestellt, welche Rolle XML, die Abfragesprache XML Query Language (XQL) sowie das Resource Description Framework (RDF) für die Nutzung im Bereich des Datenaustausches und des Datenmanagements bei der Produktentwicklung einnehmen und wie dies zur Implementierung des im Rahmen dieser Arbeit entwickelten Konzeptes der ERP-OnTo-PDM genutzt werden kann.

## 4.1 Begriffsbestimmung

Die Abkürzung XML steht für „eXtensible Markup Language“ und bezeichnet eigentlich eine Meta-Auszeichnungssprache zur Beschreibung von Strukturen von Daten und Dokumenten. Im allgemeinen Sprachgebrauch wird das Akronym XML aber auch zur Bezeichnung einer Vielzahl von assoziierten Technologien und Standards verwendet.

In den nachfolgenden Aufzählungen sind die wichtigsten dieser CO-Standards und assoziierten Technologien für den Datenaustausch und die Schaffung einer semantischen Kopplung zwischen ERP- und PDM-Systemen aufgeführt:

- XQL und XML Schema  
Diese Standards dienen als Anfragesprache bzw. zur Typisierung
- XSLT und XSL-FO  
Zur Inhaltskonfektionierung und -gestaltung
- XPOINTER, XPATH und XLINK  
Dienen zur Implementierung weitergehender Verlinkungs- und Verknüpfungskonzepte
- Document Object Model (DOM), SAX  
Ermöglichen eine Implementierung von Anwendungslogik zur direkten Verarbeitung von XML-Dokumenten
- RDF  
Für die Implementierung von Bedeutungsnetzwerken

Der vom W3C verabschiedete XML 1.0 Standard ist ein syntaktischer Grundbaustein für diese anderen Standards der XML-Familie. Die einzelnen Standards werden in den nachfolgenden Abschnitten noch eingehender erläutert.

#### 4.1.1 Abgrenzung von XML und SGML

Als Vorgänger von XML ist die Standard Generalized Markup Language (SGML) zu sehen, die in den siebziger Jahren von Dr. C.F. Goldfarb und zwei seiner Kollegen als (Document Composition Facility Generalized Markup Language (DCF GML) oder GML) entwickelt wurden. Im Jahr 1986 wurde die Entwicklung von der ISO übernommen und ist als ISO-Standard ISO 8879:1986 Information processing – Text and office systems – Standard Generalized Markup Language (SGML) verabschiedet.

Für weitergehende Informationen zum SGML Standard sei an dieser Stelle auf [N.N86] verwiesen. SGML ist eine verallgemeinerte Markierungs- bzw. Auszeichnungssprache. SGML ist ein internationaler Standard für die Definition, Identifikation und Benutzung der Struktur des Inhaltes von Dokumenten. Als eine Metasprache stellt SGML Vorschriften zur Verfügung, um neue Auszeichnungssprachen formal zu definieren. Die Auszeichnungssprachen sind dabei nicht für ein

bestimmtes Medium bestimmt, wie dies beispielsweise bei HTML der Fall ist, sondern medienunabhängig. Dadurch wird ein Datenaustausch zwischen verschiedenen Applikationen möglich und eine direkte Weiterverarbeitung der Daten ohne eine Konvertierung möglich. Dabei verfolgt SGML wie XML eine strikte Trennung zwischen der logischen Struktur der Dokumente und den eigentlichen Informationen und deren Präsentation für die einzelnen Medien.

Damit würde SGML sich als Technologie zur Implementierung einer semantik-basierenden Datenschnittstelle zwischen ERP- und PDM-Systemen auszeichnen, es hat sich jedoch gezeigt, dass der SGML Standard zu komplex ist und sich eine Vielzahl von proprietären Quasistandards entwickelt haben. Aus diesem Grund wurde XML als ein echtes Subset von SGML entwickelt. Die grundlegende SGML-Spezifikation umfasst ca. 500 Seiten, wohingegen die XML-Spezifikation nur etwa 35 Seiten lang ist [Par99, 15],[RR01, 132].

Genau wie bei SGML trennt XML die eigentlichen Daten sowohl von ihrer Struktur, als auch von ihrer Gestalt. Aus diesem Grund sollten beide nicht als Auszeichnungs-, sondern treffender als Inhaltsbeschreibungssprachen bezeichnet werden.

Beide Sprachen sind Werkzeuge zur Definition von Auszeichnungssprachen, deshalb werden sie auch als Metasprachen bezeichnet. Die meisten Dokumente, die mittels XML spezifiziert werden, bestehen wie bei SGML aus zwei Teilen. In der Dokumenttyp-Definition (DTD) ist die Deklaration der Sprachelemente und ihrer Beziehungen enthalten. Im zweiten Teil sind die ausgezeichneten („getaggten“) Daten enthalten. Diese Teilung von XML- sowie von SGML-Dokumenten in zwei Bestandteile wird durch Abbildung 4.1 verdeutlicht.

Das Konzept der DTD zur Beschreibung von strukturierten Informationen wurde in XML von SGML übernommen. Es hat jedoch für den Zweck der Datenrepräsentation, und somit als Werkzeug zum Datenaustausch zwischen datenbankorientierten Systemen wie ERP- und PDM-Systemen, erhebliche Mängel (siehe dazu Abschnitt 4.3.2).

XML besitzt dieselbe Mächtigkeit wie SGML, ist jedoch bei weitem handhabbarer, als dies SGML ist. Wichtig ist festzuhalten, dass jedes XML-Dokument ein gültiges SGML-Dokument ist. Dies ist insofern von Bedeutung, dass XML-Dokumente von SGML-Anwendungen verarbeitet werden können. So wird eine Kompatibilität von XML-Dokumenten, und damit auch von der im Rahmen dieser Arbeit entwickelten ERP-OnTo-PDM, zu einer Vielzahl von bestehenden SGML-Anwendungen hergestellt.

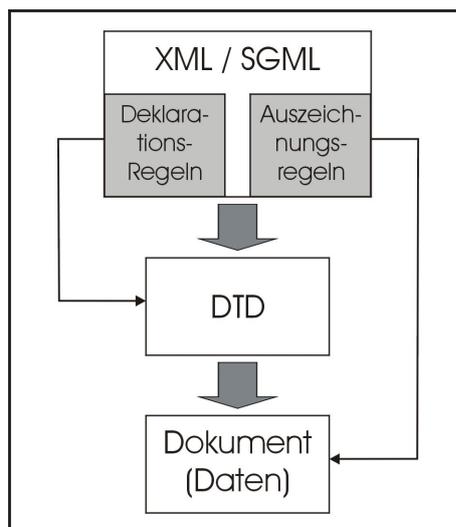


Abbildung 4.1: Zweiteilung: Daten und Deklaration von Sprachelementen siehe[Mic99, 29]

#### 4.1.2 Abgrenzung von XML und HTML

XML war ursprünglich als eine Weiterentwicklung von HTML gedacht. Also als eine Auszeichnungssprache, die sich rein auf Internet- bzw. Intranet-Publikationen beschränkt. XML ist jedoch nach heutiger Spezifikation eine generalisierte Auszeichnungssprache. Dies bedeutet, bei XML handelt es sich wie bei SGML um eine Auszeichnungssprache, die es erlaubt, einzelne Auszeichnungssprachen zu definieren. Dabei wird nicht auf Auszeichnungen, die eine Repräsentation regeln, völlig verzichtet. Auf die Möglichkeiten zur Steuerung des Layouts und der damit verbundenen Repräsentation von XML-Dokumenten auf verschiedenen Medien wird im Abschnitt 4.3 eingegangen. Bei der Spezifikation von XML wurden nachfolgenden Eigenschaften aufgenommen, die nicht in HTML enthalten sind [RR01, 133]:

- Erweiterbarkeit: XML erlaubt es dem Benutzer, eigene Tags oder Attribute zu definieren und zu benutzen, um seine Daten individuell zu strukturieren und mit Zusatzinformationen anzureichern.
- Struktur: XML ermöglicht die Darstellung beliebig tief verschachtelter Strukturen, wie man sie zur Repräsentation von komplexen Hierarchien aller Art benötigt.
- Validierung: XML gibt externen Anwendungen die Möglichkeit, die strukturelle Gültigkeit der zu verarbeitenden Daten zu überprüfen.

HTML ist eine reine Auszeichnungssprache für die Repräsentation von Inhalten im Internet. XML hingegen ist eine Metasprache zur Definition von Auszeichnungssprachen, die auf die Bedürfnisse der jeweiligen Anwender und Anwendungen angepasst sind. Um einige Schwachstellen von HTML zu beseitigen, ist vom W3C XHTML spezifiziert worden. Bei XHTML handelt es sich um eine Weiterentwicklung von HTML und die Spezifikation des HTML 4.0 Standards durch das W3C mittels XML [Mic99, 28]. Dieser Sachverhalt wird durch Abbildung 4.2 verdeutlicht.

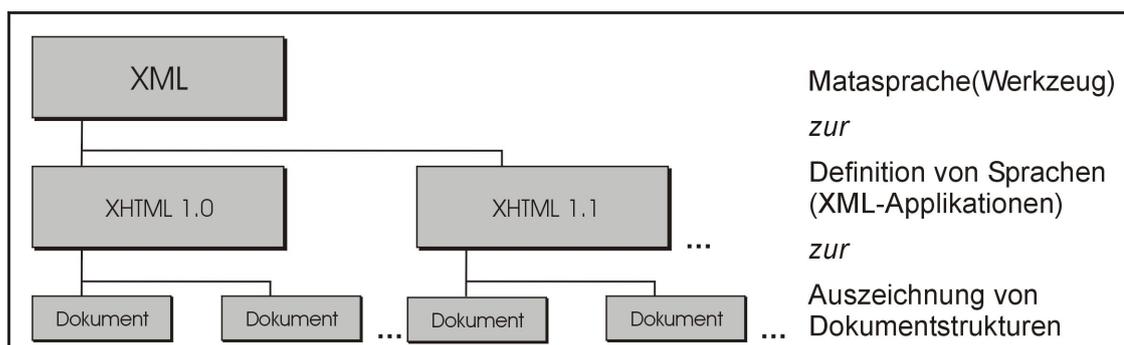


Abbildung 4.2: Das XML-Konzept 1 [Mic99, 28]

XHTML ist nur für die Auszeichnung von Dokumenten, die für das Medium Internet ausgerichtet sind, von Bedeutung und soll deshalb hier nicht weiter betrachtet werden. Für den interessierten Leser sei auf den Standard des W3C zu XHTML 1.0 verwiesen [W3C00c].

## 4.2 Das Konzept der strukturbeschreibenden Metasprache XML

Zunächst wird hier in angemessener Kürze die eigentliche Metaauszeichnungssprache XML vorgestellt. Sie zu diesen Ausführungen auch [Dim00]. Eine vertiefte und weiterführende Darstellung der Grundlagen zu XML kann der Literatur entnommen werden. Bei XML handelt es sich um eine am 17. April 1997 vom WWW-Konsortium (W3C) unter der Aufsicht des W3C Editorial Review Boards als Standard verabschiedete Auszeichnungssprache. Dem interessierten Leser sei an dieser Stelle für eine ausführliche Lektüre der XML-Spezifikation der kommentierte Standard von Tim Bray [Bra98] empfohlen.

Ein deutlicher Nebenzweck war es, die Implementationssünden von HTML nicht zu wiederholen. Bis zum heutigen Tag versuchen die HTML-Parser, HTML-Code in unterschiedlicher Art auszulegen und erzeugen damit unterschiedliche Ergebnisse.

Die Entwicklung dieses Standards verfolgt die im nachfolgenden aufgeführten zehn Entwicklungsziele [Par99, 22 ff]:

1. Leichte Einsetzbarkeit im Internet  
XML soll im Internet gut funktionieren und die Anforderungen von Anwendungen in einer verteilten Netzwerkumgebung berücksichtigen.
2. Unterstützung eines breiten Spektrums von Anwendungen  
XML ist nicht für eine spezielle Art von Anwendungen spezifiziert worden, sondern für eine Vielzahl unterschiedlichster Anwendungen, wie z. B. Auto-  
renwerkzeuge, Maschinen zur Inhaltsanzeige, Datenbankanwendungen.
3. Kompatibilität bezüglich SGML  
Jedes XML-Dokument soll zugleich ein gültiges SGML-Dokument sein. Dadurch können vorhandene SGML-Anwendungen und Werkzeuge weiter benutzt werden.
4. Einfache Entwicklung von Programmen zur Verarbeitung von XML-Dokumenten  
Dieses Ziel dient der Unterstützung der Verbreitung von XML. Die Designer von XML haben erkannt, dass die Verbreitung von XML proportional mit der Verfügbarkeit von Anwendungen zur Verarbeitung von XML-Dokumenten ansteigen wird.
5. Beschränkung der optionalen Merkmale von XML  
Eine der Hauptschwachstellen von SGML ist, dass eine Vielzahl von Optionen spezifiziert wurden, die selten oder gar nicht Verwendung finden. Dadurch wird die Spezifikation des SGML-Standards umfangreich und nahezu unüberschaubar. Des Weiteren wird durch die Beschränkung von Optionen die Gefahr der Inkompatibilität von Anwendungen durch die Nutzung bestimmter Optionen, die nicht in allen Dokumenten Verwendung finden, minimiert.
6. Gute Lesbarkeit von XML-Dokumenten  
Es soll eine gute Lesbarkeit der XML-Dokumente von Menschen erreicht werden.

#### 7. Rasche Bereitstellung des XML-Standards

XML ist aus veränderten und gewachsenen Ansprüchen an eine Web-Sprache entstanden. Diesen Ansprüchen sollte durch eine rasche Spezifikation und Bereitstellung des XML 1.0 Standards Rechnung getragen werden.

#### 8. Formales und prägnantes Design

Die XML Spezifikation sollte möglichst prägnant sein. Dazu wurde der Wortlaut der Spezifikation durch die Verwendung der erweiterten Backus-Naur-Form (EBNF) formalisiert. Bei EBNF handelt es sich um eine Standardform zur formalen Beschreibung von Programmiersprachen.

#### 9. Leicht erzeugbare XML-Dokumente

Ein Ziel des XML-Standards ist, die Erzeugung von XML-Dokumenten möglichst leicht zu machen. Dies gilt sowohl in Bezug auf die Anforderung an die Syntax und Semantik, als auch für Anforderungen an Werkzeuge zur Erstellung von XML Dokumenten. So ist die Erstellung eines XML-Dokuments mittels eines einfachen Text-Editors möglich.

#### 10. Einschränkung der Möglichkeiten für abkürzende XML-Markierungen

Sowohl bei SGML als auch bei HTML ist es ein Problem der Validierung, dass es Möglichkeiten gibt, Auszeichnungsbefehle in einer abkürzenden Schreibweise darzustellen. Dies kann beispielweise das Weglassen der schließenden Markierung sein. Diese Möglichkeiten sind in XML stark eingeschränkt worden.

Ein weiterer Aspekt, der nicht überbetont werden kann, ist das ausdrückliche Ziel, XML-Konstrukte durch möglichst viele Programmiersprachen möglichst gut verarbeiten zu können. Es zeigt eine klare Einsicht in die Tatsache, dass letzten Endes nicht die Datenformate die Qualität einer Lösung bestimmen, sondern die Funktionen der verarbeiteten Programme [RR01, 134].

XML-Dokumente bestehen aus Text, der den Inhalt des Dokuments darstellt und aus Auszeichnungs-Tags, die die Struktur und Informationen über den Content des Dokumentes liefern. Dabei ist dies im Einzelnen eine Kombination aus Elementen, Text, Kommentaren, unparsed character Data, processing Instruktionen und Entities.

Ein XML-Element besteht immer aus einem Start-Tag, Text und/oder untergeordneten Elementen und dem abschließenden End-Tag. Die syntaktischen Regeln von XML legen fest, dass das End-Tag immer mit einem „/“-Zeichen beginnt und dieselbe Schreibweise aufweisen muss, wie das Start-Tag.

Elemente können Attribute besitzen. Durch Attribute können die Angaben zu einem Tag eingeschränkt werden. Die Angabe von Attributwerten zu einem Element geschieht immer durch das Einschließen der Wertangabe in die Anführungszeichen. Attribute und die Angabe von Sub-Elementen, zur Angabe von atomaren Werten eines Elements, sind äquivalent.

Kommentare werden in XML Dokumenten wie folgt dargestellt: `<!-- Hier steht ein Kommentar -->`. Sie können nicht innerhalb eines Tags auftreten.

Mittels Processing Instruktionen können Anweisungen einer Anwendung, die auf das XML-Dokument zugreift, innerhalb dieser angegeben werden.

Durch die Angabe von Entities kann die Angabe von häufig verwendeten Texten verkürzt werden.

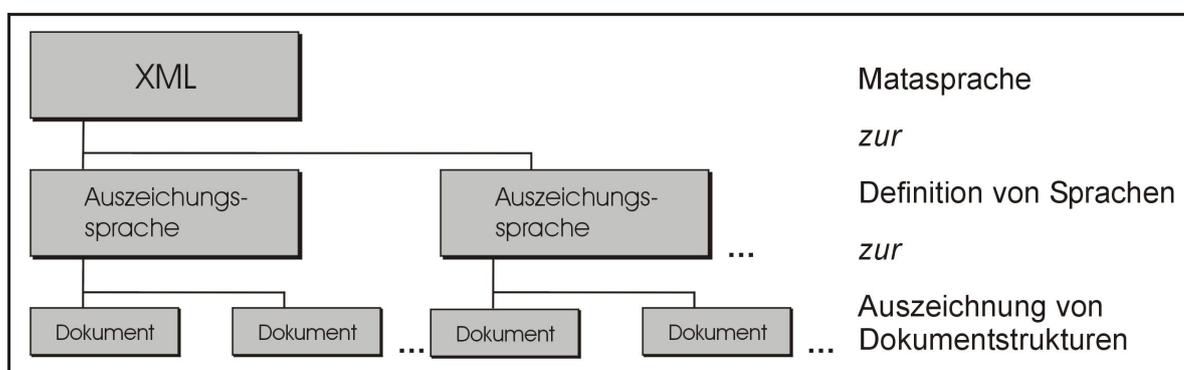


Abbildung 4.3: Das XML-Konzept 2 [Mic99, 22]

Das Konzept von XML und die wichtigen assoziierten Standards zur Spezifikation von XML sollen in diesem Kapitel beschrieben werden. Dazu wird besonders auf das RDF eingegangen, welches die Spezifikation von Metadaten ermöglicht.

In den vorhergehenden Kapiteln wurde dargelegt, dass Informationen nur dann genutzt werden können, wenn sie im benötigten Moment auch gefunden werden. Damit ein Informationssuchender unter den in einer Datenbank abgespeicherten Dokumenten die für ihn relevanten Texte bei einer späteren Informationsrecherche finden kann, müssen die Dokumente entsprechend aufbereitet und in einer Dokumentationssprache beschrieben werden. Mit Hilfe dieser Dokumentationssprache können dann Suchanfragen an die Datenbank formuliert werden. Diese Aufbereitung von explizitem Wissen nennt man inhaltliche Erschließung. Eine Dokumentationssprache ist das Werkzeug, durch welches diese inhaltliche Erschließung der Dokumente erfolgt. Hierbei kann es sich um eine (strukturierte oder unstruk-

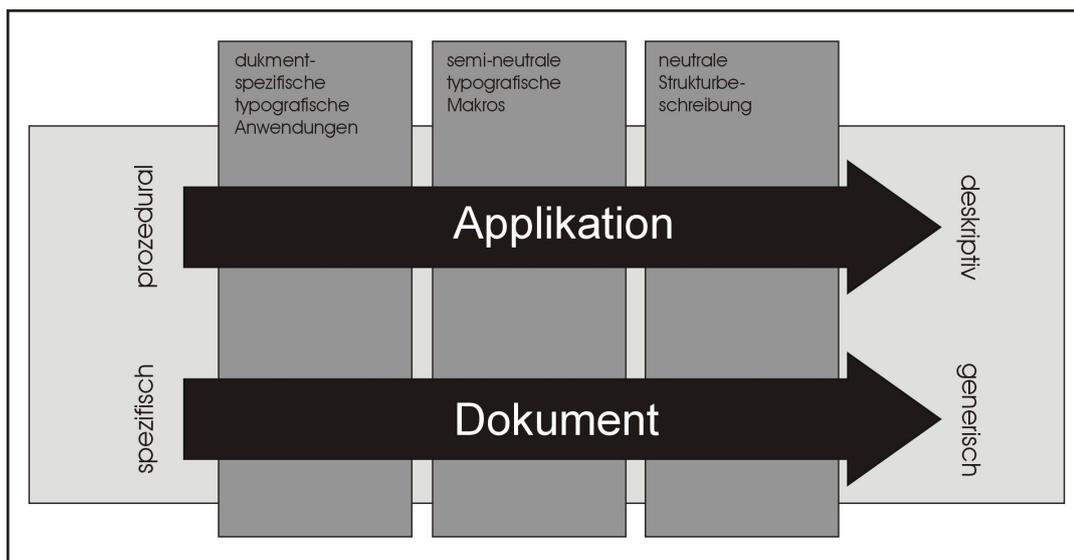


Abbildung 4.4: Das XML-Konzept 3 [Mic99, 22]

turierte) Menge natürlichsprachiger Ausdrücke handeln, oder um eine künstliche Sprache.

Es geht darum, für den Menschen und für Computer verständliche Semantik ins Internet zu bringen. In [Sig01] wird zusammenfassend festgestellt, dass dazu folgende Komponenten benötigt werden:

- Sprachen, in denen man Metainformationen über Dokumentinhalte, für Computer verständlich, ausdrücken kann
- allgemein verfügbare Terminologien, Ontologien, die solche Sprachen verwenden
- Die Integration und Konkordanz zwischen verschiedenen Terminologien
- Werkzeuge, die solche Sprachen und Terminologien verwenden, um Finden, Zugang, Präsentation und Pflege von Informationsquellen zu unterstützen

Gegenstand dieses Kapitels sind die verschiedenen Sprachen, die hierfür verwendet werden können. Solche Sprachen geben Informationen über Informationen und werden unter dem Begriff Metadaten zusammenfasst. Was genau man unter Metadaten versteht, wird im folgenden Abschnitt behandelt.

Aufgrund der Dynamik des Internets stellt die Katalogisierung von Webinhalten eine besondere Herausforderung dar. Nicht nur an den Internetobjekten selbst

und ihrem Umfeld treten laufend Veränderungen auf. Die Notwendigkeit von Metadaten, die diese Internetobjekte beschreiben, wird in [Heb00] und [Bru02] vor allem durch inhaltliche, strukturelle und örtliche Modifikation, sowie Änderungen an Urheberschaft und der Zuständigkeit für diese Objekte begründet. Gleichzeitig wird in [Bru02] betont, dass gerade die Katalogisierung von Webressourcen ein sehr schwieriges Unterfangen darstellt.

Metadaten sind Daten über Daten. Sie beschreiben die zugrundeliegenden Daten. Diese Charakterisierung der Inhalte und Merkmale einer Ressource ermöglicht je nach Qualität und Umfang eine mehr oder weniger genaue Beurteilung der beschriebenen Ressource. Dass das Konzept der Metadaten nicht erst mit Aufkommen des Internets entstanden ist, wurde bereits an anderer Stelle erwähnt. Schon seit Jahrhunderten verwenden Bibliothekare bestimmte Metadaten wie Kurzfassung (Abstract), Schlüssel- und Schlagworte und bibliographische Angaben. Für die Anwendung im WWW haben sich verschiedene Metadatenstandards entwickelt, wobei die in klassischer Weise im Bibliothekswesen verwendeten Angaben, hinsichtlich Art und Umfang der Metadaten verändert und erweitert wurden. In der Regel liegen Metadaten in textueller Form vor. Dies ist jedoch nicht obligatorisch. So existiert beispielsweise mit QBIC (Query by Image Content) von IBM ein Werkzeug, welches ein Bild als Eingabe für Suchanfragen akzeptiert. Eine ausführliche Abhandlung über Metadaten würde den Rahmen dieser Arbeit sprengen. Eine gute Übersicht hierzu bieten z. B. [N.N02b] und [FM02a]. Es gibt gerade für das Internet unzählige Metadaten-Initiativen und Projekte, die in [FM02b] einführend vorgestellt und aufgelistet werden. Damit diese unterschiedlichen Metadaten ihre beschreibende Funktion auch erfüllen können, ist eine standardisierte Struktur der Terminologie notwendig. Die Struktur und der Wertebereich der Metadaten muss festgelegt und allgemein bekannt sein, damit die verschiedenen Metadatenarten interoperabel sind. Die Generierung der Metadaten kann bei der Erzeugung der Ressource durch den Autor erfolgen oder später zum Zeitpunkt der Katalogisierung. Für Webressourcen ist aufgrund der explodierenden Anzahl von Ressourcen letztere Variante höchstens für einen bestimmten (und dadurch begrenzten) Anwendungsbereich möglich.

Metadaten und die Ressource, die durch sie beschrieben werden, können auf drei Arten miteinander verbunden werden:

1. der Metadatensatz kann entweder getrennt von der eigentlichen Ressource in einem separaten Verzeichnis oder
2. in einer Datenbank gespeichert werden oder

3. die Metadaten können direkt in die Ressource selbst integriert werden.

Die Probleme, die einer sinnvollen Verwendung von Metadaten im WWW heute noch im Wege stehen, sind vielschichtig. Viele Anbieter von Webinformationen sind mit den verschiedenen Metadatenstandards wenig vertraut, so dass die Angabe von Schlüsselwörtern häufig ad-hoc und wenig durchdacht erfolgt. Dadurch sind bisherige Metadaten oft wenig bis gar nicht aussagekräftig und deshalb zur Beurteilung der zugrunde liegenden Ressourcen in ihrer Art ungeeignet. Ein weiterer Grund hierfür ist die Vergabe beliebiger Suchwörter, die entweder sehr allgemein gehalten sind oder mit dem eigentlichen Inhalt gar nichts zu tun haben. Diese Verwendung von Metadaten soll eine möglichst hohe Zahl an Hits erzielen und dient nicht in erster Linie der tatsächlichen Beschreibung der Ressource. Die Metadaten werden in diesem Fall also dazu benutzt, bei möglichst vielen Suchanfragen möglichst hoch im Ranking der Suchergebnisse zu erscheinen und nicht dazu, in einer relativ kleinen, thematisch eingeschränkten Ergebnismenge vorhanden zu sein. Ähnliches trifft auch bei den Rating-Attributen zu. Ein sinnvoller Einsatz von Metadaten kann zur Entwicklung von qualitätskontrollierten Informationsdiensten, z. B. Development of a European Service for Information on Research and Education (DESIRE) der EU und des britischen Projekts eLib (Electronic Libraries) genutzt werden. Auf Basis von eLib sind bereits einige Informationsgateways entstanden, die jeweils einen speziellen Themenbereich abdecken und aufgrund der durch Fachleute gesammelten und beurteilten Daten eine hohe Qualität der Information gewährleisten.

Einen Schwerpunkt im Bereich der Metadaten bilden thematische Klassifikationen, die Themengebiete in selbstdefinierten bis hin zu wissenschaftlich optimierten hierarchischen Strukturen organisieren. Solche Klassifikationsmethoden wurden schon im Kapitel 3.1.3 beschrieben.

Wie beschrieben, haben wir es beim heutigen Datenmanagement vor allem mit zwei Hauptaufgaben zu tun [RR01, 89f]:

1. Wie lassen sich komplexe Inhalte und ihre Strukturen adäquat und formal korrekt repräsentieren ?
2. Wie können Struktur und Bedeutung von Inhalten von einer Maschine verarbeitet werden?

Gerade für die zweite Aufgabe eignet sich der Einsatz des in dieser Arbeit vorgestellten Konzeptes der Ontologien. Schließlich gründet diese Aufgabe auf der

Fragestellung, wie der Mensch sich eigentlich verständigt. Diese Fragestellung wurde in Kapitel 3.1.2 mittels des semiotischen Dreiecks schon aufgegriffen. Die maschinelle Verarbeitung von Strukturen beruht darüber hinaus auf den sogenannten formalen Sprachen, die die Grundlage der Informatik bilden und zu denen auch XML gehört.

### 4.3 Die Struktur von XML Dokumenten

Unter der Struktur eines Gegenstandes kann man die Summe der Teile verstehen, aus denen dieser sich zusammensetzt und die Ordnung auf welche Art und Weise diese Teile zusammenhängen.

Eine der wichtigsten und besten Eigenschaften von XML ist die, einem Dokument Struktur zu geben. Für jedes XML-Dokument wird dabei zwischen seiner logischen sowie seiner physikalischen Struktur unterschieden. In der physikalischen Struktur werden die wirklichen Daten, die ein XML-Dokument verwendet, strukturiert. Die logische Struktur ist ein Template, in dem mitgeteilt wird, welche Elemente in einem Dokument vorkommen und in welcher Reihenfolge. Der Zusammenhang zwischen diesen beiden Strukturen wird durch Abbildung 4.5 wiedergegeben.

Die logische Struktur eines XML-Dokuments zeigt, wie ein Dokument aufgebaut ist; also beschreibt es den formalen Aufbau eines Dokumentes. Im Gegensatz dazu gibt die physikalische Struktur an, was eine XML-Dokument enthält.

Neben der Struktur von Dokumenten besitzen Dokumente auch ein Layout. Unter Layout wird die Gestalt dieser Dokumente bzw. einer Klasse von Dokumenten verstanden. Beim Austausch von Dokumenten bzw. Daten zwischen verschiedenen Anwendungen wird oft die Bewahrung des Layouts in den Vordergrund gestellt und dadurch das Problem vergrößert [Mic99, 34]. XML bietet die Möglichkeit, durch die Verwendung von Stylesheets bei der Erstellung und dem Austausch von XML-Dokumenten zwischen Inhalt und dessen Layout zu unterscheiden. Dadurch kann XML als Basis für ein Austauschformat zwischen Anwendungen eingesetzt werden.

Die Verwendung von XML entfaltet im Allgemeinen bei der Beschreibung von konkreten Dokumenten nicht die volle Leistungsfähigkeit, sondern wenn es darum geht, typische Inhalte und Beziehungen für eine Klasse von Dokumenten zu erstellen. Dabei wird der herkömmliche Dokumentenbegriff von XML mittels Abstraktion von einem konkreten Dokument auf eine Klasse von Dokumenten ausgedehnt.

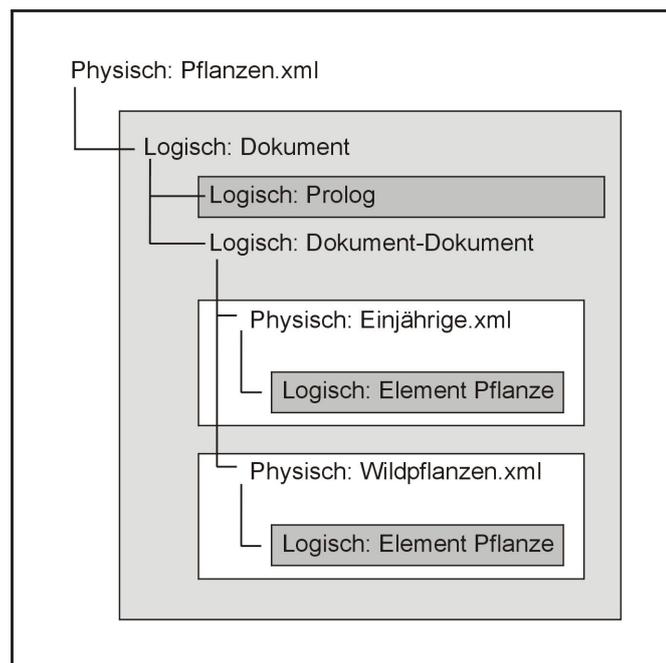


Abbildung 4.5: Zusammenhang zwischen logischer und physikalischer Struktur eines XML-Dokuments [Par99, 30]

Diese Abkehr vom klassischen Dokumentbegriff wird auch durch den allerersten Satz der Einführung in den XML-Standard gezeigt: „ Extensible Markup Language, abbreviated XML, describes a class of data objects called XML documents.“ [W3C00a, 4]. Tim Bray gibt in seiner kommentierten Version des XML-Standards zu der Erweiterung des Dokumenten-Begriffs durch XML folgende Anmerkung [RR01, 134]: „ What do you mean by 'data object'? Good question. The point is that an XML document is sometimes a file, sometimes a record in a relational database, sometimes an object delivered by an Object Request Broker, and sometimes a stream of bytes arriving at a network socket. These can all be described as 'data objects'.“ [Bra98].

Aus dieser Beschreibung des XML-Dokumenten-Begriffs kann entnommen werden, dass XML-Dokumente von einer beliebigen Applikation bei Bedarf on-the-fly erzeugt und dann über ein Netzwerk für anderen Anwendungen zugreifbar gemacht werden können [RR01, 134].

XML bietet also nicht nur die Möglichkeit, Auszeichnungssprachen zu definieren und somit also abstrakte Dokumententypen zu bestimmen. Es bietet weiter die Möglichkeit, die Anwendung dieser Auszeichnungssprachen zu regeln und somit die Auszeichnung konkreter Dokumente mit den Mitteln einer bestimmten Auszeich-

nungssprache zu ermöglichen. Dies wird durch die Abbildung 4.6 verdeutlicht.

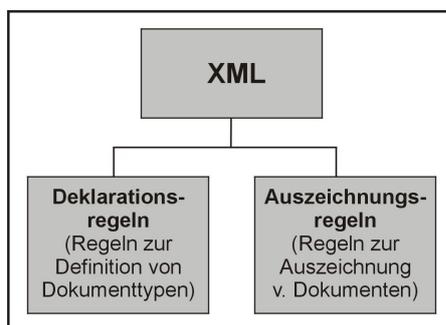


Abbildung 4.6: Das XML-Konzept 4 [Mic99, 36]

### 4.3.1 Struktur von XML Dokumenten durch DTD

Durch die DTD wird die Struktur demnach für eine Klasse von Dokumenten geregelt. Hier werden also die Elemente (Tags) der definierten Auszeichnungssprache definiert. Zwischen der DTD und den konkreten ausgezeichneten Dokumenten besteht dabei eine normative Beziehung [Mic99, 36]. Demnach wird ein konkretes Dokument als gültig bzw. valid bezeichnet, wenn es in Übereinstimmung mit seiner korrespondierenden DTD korrekt ausgezeichnet wurde. Solche validen Dokumente werden auch als Instanz eines Dokumententyps bezeichnet. Die Beziehungen zwischen Instanzen und DTD wird durch die Abbildung 4.7 verdeutlicht.

Die grammatikalische Struktur eines XML Dokuments wird normalerweise durch die DTD definiert. Hier wird angegeben, welche Tags in diesem Dokument enthalten sein dürfen und wie diese zusammen verwendet werden können [Dim00]. Wenn ein Dokument mit einer DTD übereinstimmt, nennt man diese Dokument valid. Die Sprache mit der DTD spezifiziert werden, ist selber kein valides XML-Dokument.

Eine DTD ist zusammengesetzt aus Element- und Attributdefinitionen. Eine Elementdefinition besteht aus dem Namen des Elements und der Angabe, welchen Datentyp dieses Element besitzt. Daneben wird angegeben, in welcher Sequenz diese Elemente auftreten können. Dies wird durch die Angabe der folgenden Symbole erreicht:

- ? steht für Null oder einmal

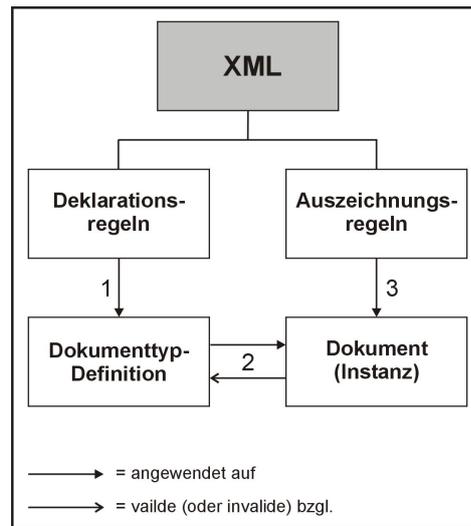


Abbildung 4.7: Das XML-Konzept 5 [Mic99, 37]

- \* steht für Null oder mehrmals
- + steht für mindestens einmal oder mehrmals
- <nothing> steht für genau einmaliges Auftreten.

Bei einer DTD können die Elemente auch Attribute besitzen. Attribute werden bei der Elementdefinition mit dem Tag `<ATTLIST>` *elementname* angegeben. Das Auftreten eines Attributes kann durch die nachfolgenden Parameter gesteuert werden:

- #REQUIRED Mussfeld
- #FIXED Der Wert ist fest
- #IMPLIED ein optionales Attribut

Durch die Angabe des Tag `<!DOCTYPE>` wird in dem XML-Dokument mitgeteilt, welche DTD verwendet wird. Dabei können zwei verschiedene Arten unterschieden werden:

1. durch die Angabe des Parameters SYSTEM wird ein lokales Verzeichnis auf dem lokalen Rechner referenziert.
2. Durch die Angabe des Parameters PUBLIC wird auf eine URL verwiesen.

Eine weitere Möglichkeit besteht in der Spezifikation der DTD innerhalb des eigentlichen XML-Dokuments. Diese Vorgehensweise ist jedoch nicht sehr vorteilhaft, da der gemeinsame Zugriff auf die DTD durch verschiedenen XML-Dokumente unmöglich wird.

### 4.3.2 Struktur von XML Dokumenten durch XSchema

XSL bietet die Möglichkeit das Layout von XML-Dokumenten zu steuern.

Eine weitere Möglichkeit zur Angabe der Struktur von XML-Dokumenten besteht in der Nutzung von XSchema. Mit XML Schema steht eine Sprache zur Spezifikation der Struktur von XML-Dokumenten zur Verfügung, die entwickelt wurde, um die Schwachstellen der DTD-Sprache zu umgehen:

Ein XML Schema wird genau wie eine DTD als Modell für eine Menge von XML-Dokumenten verwendet. In dem Modell wird beschrieben, welche Kombinationen von Elementen, Attributen und Text gültig sind. Wenn ein Dokument mit dem Schema übereinstimmt, wird es als valid in diesem Modell bezeichnet. Genau wie DTD kann XML Schema dazu genutzt werden, neue Auszeichnungssprachen zu definieren. Im Gegensatz zu DTDs, die eine eigene Sprache zur Spezifikation benutzen, sind XML Schemata in XML definiert. Gleichzeitig ist die XML Schemasprache mächtiger und ausdrucksstärker als die DTD-Sprache. Es stehen bereits rund 40 verschiedene Built-In Datentypen zur Verfügung, die jederzeit vererbt und dadurch verändert werden können.

## 4.4 Verarbeiten von XML Dokumenten

Die zur Zeit wichtigsten Möglichkeiten zur Verarbeitung von XML-Dokumenten oder Ausschnitten aus XML-Dokumenten besteht in der Nutzung von SAX oder DOM. Es werden Application Programming Interface (API)s für die meisten populären Programmiersprachen zur Verfügung gestellt, um Teile bzw. gesamte XML-Dokumente zu verarbeiten.

### 4.4.1 SAX- Simple API for XML

Weder SAX 1.0 noch SAX 2.0 sind vom World Wide Web Consortium (W3C) verabschiedete Standards. Es handelt sich jedoch bei ihnen um die meist verbreitetste

API zur Verarbeitung von XML-Dokumenten. SAX basiert auf dem Callback-Programmiermechanismus. Es wird ein Header spezifiziert, welcher bei Eintreten eines Ereignisses ausgeführt wird. Im Falle von XML-Dokumenten können solche Ereignisse das Auftreten von XML-Tags oder Blöcken von Text oder Errors sein.

Die Nachteile von SAX:

- SAX ist ereignisorientiert, dadurch generiert der Parser keine Struktur des XML-Dokuments und Applikationen müssen sich selbst um eine geeignete Repräsentationsform für die Inhalte/Informationen des Dokuments kümmern.
- SAX kann nur Dokumente parsen, es kann jedoch keine existierenden Dokumente ändern bzw. neue Dokumente erstellen.

#### 4.4.2 DOM - Document Object Model

DOM ist eine Sammlung von Interfaces, um auf als Baum strukturierte Dokumente zuzugreifen. Dies macht es für die Verarbeitung von XML sehr geeignet. Der DOM-Parser erstellt eine Baumstruktur des XML-Dokuments. Durch die Erstellung dieser Baumstruktur wird eine für die Verarbeitung von Applikation geeignete Repräsentation der Informationen des Dokuments erstellt.

### 4.5 Resource Description Framework (RDF)

Die Anwendung von XML bietet zwar eine Lösung für das Problem der Strukturierung von Dokumenten, löst aber nicht das Problem der Semantik. Den Sinn oder die Bedeutung eines mit XML modellierten Sachverhalts kann man z. B. mit Hilfe des in diesem Abschnitt vorgestellten RDF ausdrücken. RDF ist eine Empfehlung des W3C und wurde mit dem Ziel entwickelt, Interoperabilität zwischen Applikationen durch den Austausch maschinenverständlicher Informationen im Internet zu ermöglichen. RDF basiert auf XML und ermöglicht dadurch die Nutzung beliebiger eigener Tags. „RDF will retain the capability to exchange metadata between application communities, while allowing each community to define and use the metadata that best serves their needs.“[N.N02a]

Bedeutung wird im RDF durch ein Schema ausgedrückt. Einem Wörterbuch vergleichbar, enthält ein solches Schema Begriffe und ordnet diesen eine spezifische Bedeutung zu. Da Schemata für jeden Kontext beliebig definiert werden

können, Überschneidungen der Begriffsdefinitionen in unterschiedlichen Schemata aber zur Gewährleistung der begrifflichen Eindeutigkeit vermieden werden müssen, nutzt RDF die Funktionalitäten der XML Namespaces. Jedes Prädikat muss exakt einem Namespace bzw. Schema zugeordnet werden. Ein Schema besteht aus Mengen, die jeweils aus drei Elementen entsprechend Subjekt, Prädikat, Objekt bestehen. Dadurch soll der Sachverhalt „bestimmte Dinge haben Eigenschaften mit bestimmten Werten“ verdeutlicht werden. Alle drei Elemente werden jeweils durch URIs ausgedrückt. Jede Beschreibung einer Ressource besteht aus einem Triple der Form (resource, property, value). Weitergehende Informationen zu diesem Thema erhält der interessierte Leser unter [<http://www.w3.org/TR/rdf-schema/>] und [<http://www.cetis.ac.uk/content/20010927172953/viewArticle>].

Das Resource Description Framework bildet das Fundament zum Austausch von Metadaten zwischen verschiedenen Resource Description Communities. Es wird vom World Wide Web Consortium (W3C) entwickelt und liegt in dem Status einer Empfehlung vor [BG99].

Das Konzept einer Metadaten Auszeichnung ist nicht neu. Die Integration von Metadaten wurde mit HTML versucht, führte jedoch nicht zu den gewünschten Ergebnissen. Die meisten Search-Engines sind von der Verwendung von Meta-Tags zur Indexierung von Web-Seiten wieder zurück gegangen und benutzen anderen Verfahren der Volltestsuche [Med00]. Einer der führenden Köpfe und Vordenker, wenn es um Metadaten geht, ist Tim Berners-Lee in seinem Buch „Weaving the Web“<sup>2</sup>

RDF ist eine XML Applikaton für den Zweck der Metadaten Repräsentation.

```
<? xml version=1.0"?'?>
<RDF
  xmlns=""http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
  xmlns:DC=""http://www.purl.org/DC#"
  xmlns:y=""http://www.description.org/schema"''>
  <Description about=""http://www.xyz.de/Example/Smith/"''>
    <DC:date rdf:resource=""July 1999"''/>
    <DC:subject rdf:resource=""Intelligent Agents"''/>
```

---

<sup>2</sup>Ein Designvorschlag für eine neue Form des WWW. Hier wird aufgezeigt welche Verbesserungen durch die Einführung von Semantik in das WWW hat. Siehe für weitere Information

```

        <DC:creator rdf:resource="'http://www.xyz.de/~smith/'"/>
    </Description>

    <Description about="'http://www.xyz.de/~smith/'">
        <DC:name rdf:resource="'John Smith'"/>
        <DC:email rdf:resource="'smith@organisation.de'"/>
    </Description>

</RDF>

```

Die Referenz zur RDF Spezifikation des W3C [W3C99]. Die Referenz zur W3C Spezifikation des RDFS [W3C00b].

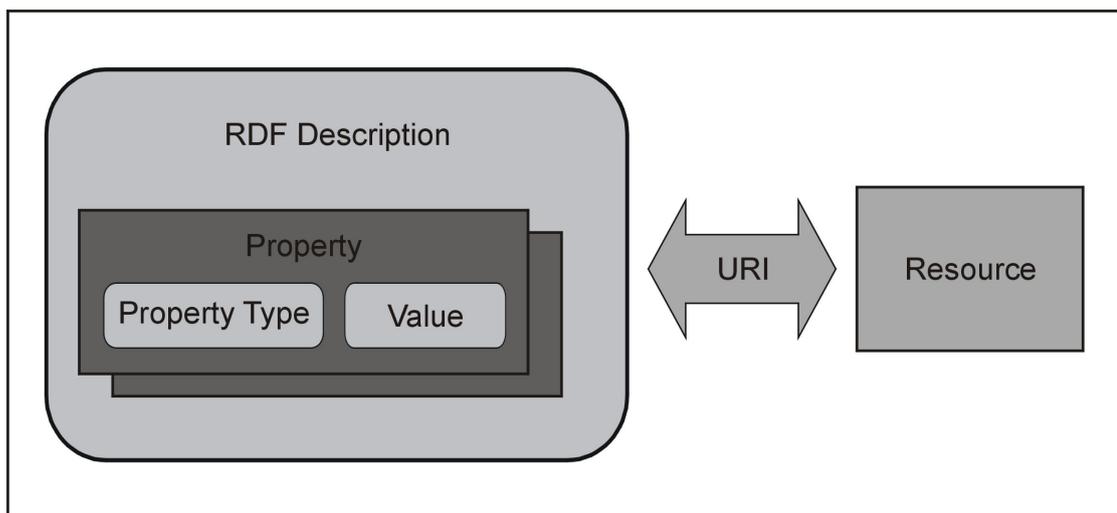


Abbildung 4.8: Das RDF Model [Inn98, 2]

Das Ziel des RDF ist es, die Spezifikation und den Austausch von Metadatenstrukturen zu ermöglichen [Dim00]. Das RDF ist das Resultat eines früheren Projektes, den Web Search Engines nicht nur eine Suche über die Inhalte zu ermöglichen, sondern über die Nutzung von Metadaten bessere Suchergebnisse zu erzielen. Jedes RDF Dokument kann als eine Gruppe von Statements aufgefasst werden, welche eine Ressource beschreiben. In diesem Sinne wird als eine Ressource alles aufgefasst, was durch eine gültige URI (Unified Resource Identifier) eindeutig identifiziert werden kann. Das kann beispielsweise eine gesamte Web-Site, Teile einer Web-Site, ein XML-File, eine Person, ein Buch usw. sein. Ressourcen werden

durch ihre Eigenschaften beschrieben, wobei jede Eigenschaft einen Typ und einen Wert besitzt. Eigenschaften können atomar sein (aus Basis-Datentypen bestehen) oder auch andere Ressourcen beinhalten.

## 4.6 Product Data Markup Language (PDML)

Unter dem Begriff PDML ist eine, innerhalb eines Projektes unter Federführung des US Department of Defence (DoD) entwickelte, Menge an DTD zusammengefasst. Das Ziel der PDML ist es, einen Austausch von organisatorischen und strukturbezogenen Produktdaten in PDM-Systemen zu ermöglichen, wobei der Fokus auf Systemen liegt, die im Umfeld des US DoD Anwendung finden. Die PDML (Version 0.5, April 1999) umfasst sieben (Application Transaction Sets (ATS)), d.h. spezifische DTDs für dedizierte Applikationen, ein Integrationsschema auf EXPRESS-Basis und Mapping-Spezifikationen, um ATS auf das Integrationsschema abzubilden. Den zentralen Baustein der PDML bildet somit das Integrationsschema; eine umfassende DTD basierend auf dem STEP Integrated Resources [BUR]. Ebenso kann EXPRESS zur Definition eines PDML-Schemas genutzt und daraufhin mit Hilfe eines Tools und unter Anwendung entwickelter Konvertierungsalgorithmen zu einer gültigen DTD transformiert werden. Im Gegensatz zur PDM-Enablerspezifikation, die mehr auf Mechanismen für den CORBA-basierten Austausch von Objekten zwischen PDM-Systemen fokussiert und weniger auf Inhalte bzw. Semantik der Daten, ist die Hauptintention der PDML die Bereitstellung einer Art Standard-Vokabular für den PDM-Datenaustausch über das Internet. Prinzipiell könnte ein Datenaustausch in der gleichen Art und Weise auch über eine STEP-Austauschdatei [ISO10303-21] erfolgen. Der Vorteil der XML-basierten Lösung ist, dass eine PDML-Datei Informationen über das zugrundeliegende Schema beinhaltet, mit einem konventionellen Browser ansehbar und leicht interpretierbar, also insgesamt besser und mit Standard-Softwarewerkzeugen handhabbar ist. Nachteilig ist, dass ein festes Schema festgelegt wird.

## 4.7 MatML

MatML ist ein Projekt, um einen XML Dialekt zur Beschreibung und zum Austausch von Produkteigenschaften zu entwickeln, siehe [Beg00]. Ziel ist es, eine Beschreibungssprache für den Austausch von Produkteigenschaften und Produkt-

daten über das WWW auf XML-Basis zu entwickeln. Durch den verstärkten Einsatz und die Entwicklung von XML ist in der Forschungsgemeinschaft auf dem Gebiet des Produktdatenmanagement die Notwendigkeit der Entwicklung einer XML basierten Produktdatenbeschreibungssprache auf XML-Basis erkannt worden. NIST ist für die Entwicklung eines Ansatzes, der als MatML bezeichnet wird, verantwortlich. Das Systems Integration for Manufacturing Applications Program (SIMA) unterstützt die Entwicklung von MatML durch die Einsetzung einer Arbeitsgruppe zur Entwicklung von MatML. Weitere Details können unter <http://www.ceramics.nist.gov/matml/matml.html> nachgesehen werden.

## 4.8 Fazit

Wie beschrieben, haben wir es beim heutigen Datenmanagement vor allem mit zwei Hauptaufgaben zu tun [RR01, 89f]:

1. Wie lassen sich komplexe Inhalte und ihre Strukturen adäquat und formal korrekt repräsentieren?
2. Wie können Struktur und Bedeutung von Inhalten von einer Maschine verarbeitet werden?

Gerade für die zweite Aufgabe eignet sich der Einsatz des in dieser Arbeit vorgestellten Konzeptes der Ontologien. Schließlich gründet diese Aufgabe auf der Fragestellung, wie sich der Mensch verständigt. Diese Fragestellung wurde in Kapitel 3.1.2 mittels des semiotischen Dreiecks schon aufgegriffen. Die maschinelle Verarbeitung von Strukturen beruht darüber hinaus auf den sogenannten formalen Sprachen, die die Grundlage der Informatik bilden und zu denen auch XML gehört.

RDF ist eine Empfehlung des W3C und wurde mit dem Ziel entwickelt, Interoperabilität zwischen Applikationen durch den Austausch maschinenverständlicher Informationen im Internet zu ermöglichen. RDF basiert auf XML und ermöglicht dadurch die Nutzung beliebiger eigener Tags.

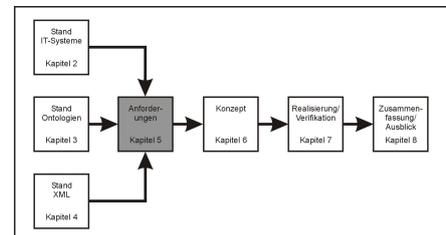
Es wurden die beiden auf XML basierenden Produktbeschreibungssprachen Product Data Markup Language (PDML) und MatML vorgestellt. Weiter wurden deren gemeinsame Hauptschwachstelle, eine fehlende Unterstützung eines semantischen Datenaustausches angesprochen. Das Konzept der Ontologie und deren Umsetzung mittels einer XML-basierten Implementierung durch RDF und die daraus

resultierende Beseitigung der beschriebenen Schwachstellen, soll Gegenstand des folgenden Kapitels sein. Es werden die Anforderungen an einen semantikbasierten Datenaustausch zwischen PDM- und ERP-Systemen aufgezeigt, die als Grundlage für die Entwicklung von ERP-OnTo-PDM dienen.

# Kapitel 5

## Anforderungen an ontologiebasierte Kopplungen von ERP- und PDM-Systemen

Bisher wurden die Aufgabengebiete der beiden Standardanwendungssysteme PDM und ERP aufgezeigt. Dabei wurde die Wichtigkeit einer effizienten Kopplung dieser Systeme herausgearbeitet. Es wurde in Kapitel 3 ein Konzept aus dem Bereich der KI vorgestellt, mittels der eine semantische Kopplung von Anwendungssystemen möglich ist. Durch eine solche semantische Kopplung ist es nicht mehr notwendig, für jedes der an einer solchen Kopplung beteiligten Systeme eine explizite Schnittstelle zu programmieren. Es wird vielmehr durch ein semantisches Mapping der einzelnen Konzepte der beteiligten Systeme ein Inter Lingua Server errichtet. In Kapitel 4 wurde die Internettechnologie XML mit dem Metadaten-Standard RDF vorgestellt. Mittels dieser Techniken wird die Implementierung der ERP-OnTo-PDM erfolgen. In diesem Kapitel werden die verschiedenen Anforderungen an ein ontologiebasiertes Konzept zur Kopplung von ERP- und PDM-Systemen aufgezeigt. Dazu werden die Anforderungen in die folgenden drei Bereiche untergliedert:



- organisatorische Anforderungen,
- anwenderbezogene Anforderungen,

- technische Anforderungen.

Wie schon in dieser Arbeit an verschiedenen Stellen aufgezeigt, ist eine integrierte und durchgängige Daten- und Informationsverarbeitung von immer größerer Bedeutung für die Unternehmen. Allgemein gelten für die Schaffung der ERP-OnTo-PDM die Anforderungen, die an jede Software gestellt werden und als die wichtigsten Kriterien zur Bewertung der Softwarequalität angesehen werden [Sta89]:

- **Änderbarkeit**

Wie schwierig und aufwendig ist es, Änderungen (Anpassungen, Erweiterungen) an dem Programm vorzunehmen?

- **Benutzbarkeit**

Wie einfach läßt sich das Programm durch den Benutzer handhaben (Benutzerfreundlichkeit)?

- **Effizienz**

In welchem Umfang beansprucht das Programm Hardware-Ressourcen (CPU, Hauptspeicher, Peripherie)?

- **Korrektheit**

Wie exakt erfüllt das Programm die Vorgaben lt. Pflichtenheft (Funktionserfüllung)?

- **Portabilität**

Läßt sich das Programm ohne großen Aufwand auf andere DV-Anlagen übertragen (Übertragbarkeit)? (Standardsoftware!)

- **Robustheit**

Wie reagiert das Programm (durch Prüfungen, Kontrollen usw.) auf fehlerhafte Eingaben oder falsche Handhabung?

- **Testbarkeit**

Welchen Aufwand erfordert der Programmtest (Prüfbarkeit)?

- **Verständlichkeit**

Wie gut ist das Programm von Menschen zu lesen (klare Strukturen, überschaubare Module, ausreichende Kommentare)?

- **Wartbarkeit**

Nach welcher Zeit kann das Programm bei Auftreten eines Fehlers wieder lauffähig gemacht werden?

- **Zuverlässigkeit**

Mit welcher Quote (Prozentsatz) arbeitet das Programm beim wiederholten Einsatz im Routinebetrieb korrekt?

Diese Anforderungen an Software sind sehr allgemein und nicht speziell für die Kopplung von ERP- und PDM-Systemen. Um den besonderen Anforderungen für eine semantische Kopplung dieser Systeme gerecht zu werden, werden diese in diesem Kapitel nun eingehender vorgestellt. Zunächst werden die verschiedenen Anforderungen aus den einzelnen beteiligten Unternehmensbereichen aufgezeigt. Anschließend werden jene Anforderungen besprochen, die von den Benutzern von ERP-OnTo-PDM an dieses Konzept gestellt werden. Abschließend wird die Gruppe von Anforderungen diskutiert, die durch die verschiedenen beteiligten technischen Systeme und die verwendeten Technologien impliziert werden.

## 5.1 Organisatorische Anforderungen

In diesem Abschnitt sind Anforderungen zusammengestellt, die aus den Unternehmensbereichen entstehen, die direkt bzw. indirekt mit dem Einsatz von PDM-Systemen konfrontiert sind. Zu diesen Unternehmensbereichen sind insbesondere die nachfolgenden zu zählen:

- Konstruktion
- Arbeitsplanung und -vorbereitung
- Produktion
- Qualitätssicherung

Zur weiteren Erläuterung werden im nachfolgenden die potentiellen Anwendungsmöglichkeiten und die daraus resultierenden Anforderungen an ERP-OnTo-PDM behandelt, die sich aus dem Unternehmensbereich der Konstruktion ergeben. Die Anforderungen aus den weiteren Unternehmensbereichen sollen im Rahmen dieser Arbeit nicht ausführlich abgeleitet werden, sondern in die organisatorischen Anforderungen ohne explizite Ableitung und ausführliche Besprechung aufgenommen werden.

### 5.1.1 Konstruktion

Der Begriff Produktlebenszyklus beschreibt einen Kreislauf aufeinander folgender Lebensphasen eines Produktes <sup>1</sup>.

Wie in Abbildung dargestellt, umfasst der Produktlebenszyklus die Phasen Produktentwicklung, Produktherstellung, Produktnutzung und Produktentsorgung. Die Produktentwicklung stellt eine entscheidende Phase im Produktlebenszyklus dar. Sie hat im besonderem Maße Einfluss auf die Kosten, die Durchlaufzeit und auf die Qualität eines Produktes. In dieser Phase werden zwar nur ca. 10% der gesamten Kosten eines Produktes verursacht, aber ca. 70% der Kosten, die in den folgenden Phasen anfallen, festgelegt. Beispielsweise geschieht dies durch die Bestimmung von Materialien, Fertigungsverfahren etc. (vgl. [VR87]) Anders formuliert liegen Potentiale für erzielbare Kostenreduzierungen eines Produktes im Bereich der Entwicklung bei ca. 50%, während sie in den nachgelagerten Bereichen bei ca. 5% liegen [Bür95]. Die Produktentwicklung kann wiederum unterteilt werden in die Phasen Produktplanung, -konstruktion und -erprobung, die sich jedoch nicht mehr so stark voneinander abgrenzen, sondern durch fließende Übergänge gekennzeichnet sind.

#### Produkterstellung

Die Produkterstellung umfasst alle Belange der eigentlichen Erzeugung des Produktes. Es wird angestrebt, dass die hier gesammelten Erfahrungen in einem iterativen Prozess in die Produktentwicklung zurückfließen, um bei nachfolgenden Entwicklungen Fehler zu vermeiden, die Qualität der Produkte zu steigern und nach Möglichkeit die Herstellungskosten zu senken. Die Phase der Produktentwicklung und Produktherstellung wird auch oft als Produktentstehung zusammengefasst. Der Produktentstehungsprozess als Erweiterung der klassischen Definition des Begriffs „Konstruktion“ ist demnach die Gesamtheit aller Tätigkeiten, angefangen von der Idee bzw. dem Auftrag, über die Entwicklung und Konstruktion, bis hin zum fertigen Produkt und zum Vertrieb.

Die Produktnutzung und die Produktentsorgung schließen den Kreis des Produktlebenszyklus. Aus diesen Phasen fließen idealerweise Informationen in die Produktentstehungsphasen zurück und können so für die Weiterentwicklung und Ver-

---

<sup>1</sup>Der Begriff Produktlebenszyklus ist in der Literatur nicht eindeutig definiert und wird auch von unterschiedlichen Fachdisziplinen (Betriebswirtschaft, Ingenieurwesen etc.) mehrdeutig verwendet. Im Bereich des Ingenieurwesens hat sich die aufgeführte Definition durchgesetzt (vgl. [Bei98] [Gra90] [GLR92][SK97][VR93] [Wes90])

besserung der Produkte verwendet werden.

Bei der Entwicklung von CAD-Systemen ist eine Tendenz hin zu integrierten Systemen erkennbar. Diese modernen CAD-Systeme bieten dem Konstrukteur neben der Unterstützung bei der Zeichnungserstellung auch Unterstützung bei den sekundären Tätigkeiten. Die eigentlichen schöpferischen Tätigkeiten innerhalb der Konstruktion werden in den momentan am Markt verfügbaren CAD-Systemen aber nicht unterstützt. Jedoch sollten durch eine optimale Unterstützung des Produktentstehungsprozesses durch geeignete IT-Werkzeuge die an der Produktentwicklung beteiligten Mitarbeiter mehr und mehr von nicht-kreativen Tätigkeiten befreit werden, um so Freiräume für Kreativität zu schaffen. Bisher werden die Schritte fünf, sechs und sieben von modernen CAD-Systemen unterstützt. Schritte eins, zwei und drei werden weitestgehend im Kopf des Konstrukteurs durchgeführt. Ziel muss eine durchgängige computerbasierte Unterstützung aller Schritte sein. Ein Ansatz dazu kann in der Featuretechnologie gesehen werden. Siehe dazu [VDI99] und [Mee94].

Die Tätigkeit des Konstrukteurs besteht zu 90 Prozent aus der Neu- oder Anpassungskonstruktion von Bauteilen mit bekannten Funktionsstrukturen. Der Anteil tatsächlicher Neukonstruktionen, bei denen die Funktionsstrukturen neu entworfen werden müssen, ist daher sehr gering und durch Anwenderprogramme nur unter großem Aufwand abzubilden.

Der Kerngedanke besteht darin, eine flexible und anwenderzentrierte Lösung unter Nutzung der Web-Technologie bereitzustellen, die das Zusammenwirken verschiedener IT-Systeme auf gleichberechtigter Ebene ermöglicht und dabei die Autonomie der Einzelsysteme nicht beschränkt.

Bisherige Integrationsansätze gehen von einem zentralen Führungssystem, einer hierarchischen Ordnung der gesamten IT-Systemlandschaft und einer harmonisierten Semantik der Informationsmodelle aus, die jedoch insbesondere innerhalb virtueller Kooperationen nicht realisierbar ist.

## 5.2 Technische Anforderungen

Das Internet bildet letztlich nur die Infrastruktur für die Durchführung kooperativer Arbeitsprozesse. Die Mehrzahl der Anwendungssysteme unterstützt die Verteilung und die Nutzung des Internets bislang jedoch nicht im erforderlichen Maße.

Funktionale Anforderungen:

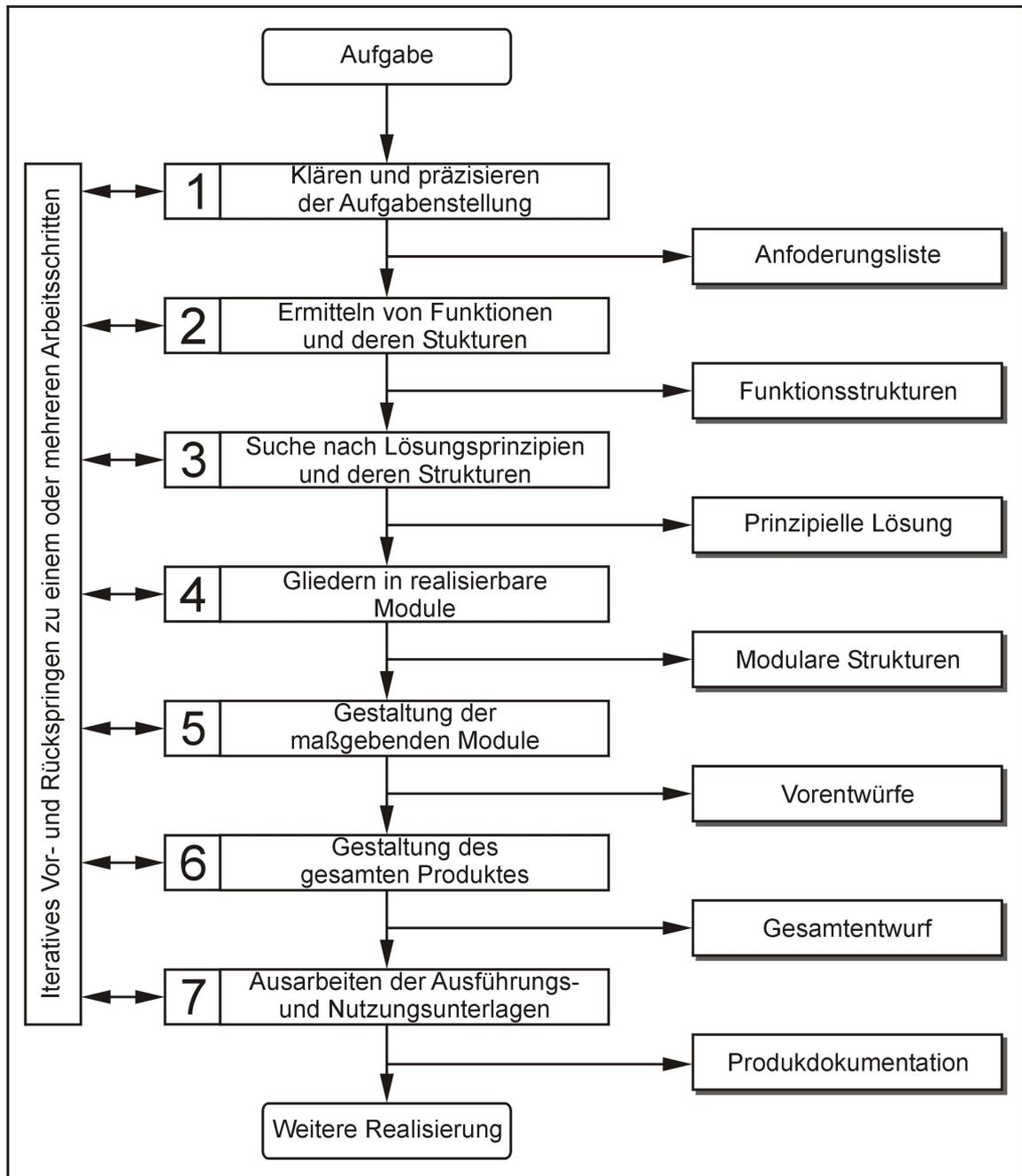


Abbildung 5.1: Vorgehen beim Entwickeln und Konstruieren nach VDI 221

- Online-Integration aller betriebswirtschaftlichen Anwendungskomponenten und -daten im Unternehmen: Isolierte Einzellösungen mit Datenaustausch zwischen den Anwendungen lediglich zu bestimmten Zeitpunkten (Batch-Lösungen) haben ausgedient.
- konsistente Abbildung aller betrieblichen Mengen-, Werte- und Informationsflüsse
- zeitgerechte Bereitstellung aller für die betrieblichen Geschäftsvorgänge relevanten Daten und Informationen.
- durchgängige Unterstützung von Geschäftsprozessen, und zwar sowohl firmenintern als auch im Dreierverbund »Kunde-Firma-Lieferant«
- Nutzung des Internets zur Kommunikation und Informationsverarbeitung
- schnelle und einfache Anpassung betriebswirtschaftlicher Anwendungen an individuelle Anforderungen und sich immer schneller verändernde Geschäftsprozesse (Customizing)
- funktionale Erweiterbarkeit der betriebswirtschaftlichen Anwendungen durch leistungsfähige Entwicklungswerkzeuge
- Daten- und Programm-Integration von Desktop-Komponenten wie z. B. MS-Excel, MS-Word und Lotus Notes mit den betriebswirtschaftlichen Kernanwendungen
- leichte Mensch-Maschine-Interaktion durch intuitiv zu bedienende Benutzungsoberflächen

Technisch-organisatorische Anforderungen:

- Begrenzung der Gesamtkosten (LCO, Lowest Cost of Ownership) für die im Unternehmen genutzte DV-Infrastruktur. Die Gesamtkostenbetrachtung (TCO, Total Cost of Ownership) darf sich allerdings nicht nur auf die technische Infrastruktur beschränken, sondern muss alle Kostenfaktoren berücksichtigen:
  - Beschaffungskosten für Hardware und Software
  - Implementierungszeit und -kosten für die gesamte DV-Lösung
  - Upgrade-Aufwand beim Release-Wechsel

- Aufwand für Anpassungen der Software an sich verändernde Geschäftsprozesse
- Aufwendungen für Mitarbeiterschulung
- kontinuierlich anfallende Betriebs- und Wartungskosten
- hohe Portabilität der Anwendungs-Software. Betriebswirtschaftliche Anwendungen haben mit zehn und mehr Jahren eine deutlich höhere Lebensdauer als viele Hardware-Komponenten. Die Nutzbarkeit der Anwendungs-Software muss auch bei veränderten Hardware-, Software- und Kommunikationstechnologien garantiert sein.
- leichte Anpassbarkeit (Skalierbarkeit) der Hard- und Software-Installation bei veränderten Lastprofilen, z. B. bei steigender Benutzerzahl oder beim Einsatz zusätzlicher Anwendungen
- Unterstützung dezentraler Systemkonfigurationen, z. B. über das Internet
- offene Systemarchitektur durch Unterstützung von verbreiteten Schnittstellen
- kurze, vorhersehbare Antwortzeiten bei Dialoganwendungen
- hoher Durchsatz bei der Verarbeitung von Hintergrundaufträgen
- 24-Stunden-Verfügbarkeit des Gesamtsystems
- kurze Stillstandzeiten bei Release-Wechsel

Im Folgenden werden die Anforderungen an die informationstechnische Unterstützung von Produktionsprozessen bezüglich der Integration von IT-Systemen und des übergeordneten Daten- bzw. Informationsmanagement vorgestellt.

#### Allgemeine Anforderungen

- Sicherheit:  
Die Sicherung der Vertraulichkeit bzw. Integrität von Daten durch Verschlüsselung, sowie die Zuverlässigkeit bei Transaktionen sind unbedingt zu gewährleisten. Die Möglichkeit zur Abbildung von spezifischen, sichtabhängigen Sicherheits- und Zugriffsschutzmechanismen nach bilateralen Vereinbarungen muss gegeben sein.

- **Wirtschaftlichkeit:**  
Ein geringer Integrations-, Anpassungs- und Nutzungsaufwand muss gewährleistet sein, die vorhandene Infrastruktur (Hardware, Software und Netzwerke) muss aus Gründen des Investitionsschutzes beibehalten werden können. Ein Produktivitätsausfall in der Einarbeitungsphase muss durch geringen Schulungs- und Einarbeitungsaufwand minimiert werden können.
- **Leistungsfähigkeit:**  
Die Leistungsfähigkeit, bzw. die Möglichkeiten der Einzelsysteme dürfen sich durch die Integration in einer kooperativen Umgebung nicht verschlechtern. Die kooperative Umgebung darf die Ausführung der eigentlichen Arbeitsinhalte nicht zusätzlich erschweren und muss in einem gewohnten Arbeitsprozess ohne großen Interaktionsaufwand einsetzbar sein.

#### Anwenderbezogene Anforderungen

- **Aktive Unterstützung bei Navigation, Suche und Recherche:**  
Anwender müssen aktiv bei der Suche und Navigation in PDM-Systemen der Kooperationspartner, bzw. in einem verteilten Gesamtsystem, unterstützt werden. Dies erfordert die Bereitstellung direkter, aktiver Suchmethoden (z. B. Merkmalsuche, Volltextsuche).
- **Transparenz:**  
Dem Produktentwickler müssen an seinem Arbeitsplatz alle von ihm benötigten Informationen transparent und aktuell zur Verfügung stehen, damit er sich soweit wie möglich auf seine eigentlichen Aufgaben konzentrieren kann. Produktdaten bzw. Beziehungen zwischen verschiedenen Objekten, Versionen, Varianten, Status müssen systemübergreifend gehandhabt werden können.
- **Integration von Informationen aus externen Systemen:**  
Der Zugriff auf externe Informationen wird zum großen Teil in der Ausarbeitungs- bzw. Detaillierungsphase benötigt. Hier muss ein umfassendes Informationsmanagement gewährleistet sein, das sowohl Systeme von Kooperationspartnern, als auch Web-basierte Produktionskataloge bzw. Engineering Märkte umfasst. Dabei sollte der Anwender sich nicht jedes Mal neu identifizieren müssen, sondern für ihn müssen die verschiedenen Subsysteme wie ein Gesamtsystem wirken.

- **Anwenderfokus:**

Die Bereitstellung aller benötigten Informationen muss immer aus der Sicht eines Bereichs bzw. einer Domäne (profilabhängig) erfolgen. Ferner müssen Funktionen zur Übernahme von Produktdaten aus verschiedenen Quellsystemen in den eigenen Kontext bereitgestellt werden. Zugeordnete Produktdaten sollten dabei entsprechend der implementierten PDM-Konfiguration automatisiert übernommen bzw. adaptiert werden.
- **Einfachheit:**

Ein kooperatives System muss eine einfache intuitive Benutzbarkeit bei geringem Einarbeitungs- bzw. Schulungsaufwand gewährleisten. Der Anwender muss durch das System bei standortübergreifenden Aktionen zusätzlich unterstützt werden. Die graphische Benutzerschnittstelle muss eine aktive, problemorientierte Unterstützung für den Anwender leisten und sich nahtlos in bestehende Anwendungen integrieren. Daher ist aus softwareergonomischer Sicht einerseits die Nutzung vorhandener Standards erforderlich und andererseits die Verfügbarkeit auf allen gängigen Plattformen, die nicht durch die Nutzung proprietärer Funktionalitäten blockiert werden darf. Zudem ist eine visuelle Unterstützung für den Anwender dringend erforderlich, welche Objekte aus verteilten Datenbeständen direkt gegenüber lokalen Informationen sichtbar macht bzw. abgrenzt und damit hilft, Unklarheiten und Fehlerquellen zu vermeiden.
- **Datenaustausch auf Produktmodellebene:**

Der Austausch von Informationen kann nicht feingranular auf Parameter-Ebene geschehen, sondern muss in Form kompletter (statischer) Modelle bzw. Dokumente erfolgen. Das ist auch insbesondere aufgrund der Haftungspflicht bzw. der Nachweispflicht im Rahmen der Produkthaftung erforderlich, wo alle Vorgänge und technischen Dokumentationen über einen langen Zeitraum verfügbar gemacht werden müssen. Produktmodelle, die dynamisch zur Laufzeit erzeugt und nicht permanent gespeichert werden, können dies nicht.
- **Nachvollziehbarkeit:**

Bei der Übertragung von Daten muss ein kontrollierter Import und Export zwischen verschiedenen Kooperationspartnern durch Erfassung, Protokollierung und Archivierung von Ein- und Ausgängen gewährleistet werden, um die Recherchierbarkeit bzw. Rekonstruierbarkeit von Zugriffs- und Austauschvorgängen sicherzustellen. Zudem sind Organisationsstruktur- und Konfigurationsdaten für direkte Weiterverarbeitung bei Datenaustauschvorgängen

mit zu übertragen.

#### Systembezogene Anforderungen

- **Flexibilität:**

Es ist ein hoher Grad an Flexibilität erforderlich, um die häufig und unerwartet auftretenden Änderungen in der IT-Systemlandschaft der Partnerunternehmen innerhalb einer Kooperation aufgrund der konstanten Änderungen im Markt zu unterstützen. Die IT-Werkzeuge einer Umgebung unterliegen einem permanenten dynamischen Wandel durch neue Releases, Mergers oder neue Systeme für bestimmte Aufgaben. Die Integration neuer Werkzeuge, d.h. die Konfiguration, Einrichtung und Anpassung, muss möglichst einfach und schnell durchführbar sein.
- **Anpassbarkeit bzw. Offenheit:**

Verteilte Produktionsprozesse werden durch eine Vielzahl verschiedener Systemtypen (EDM/PDM, ERP/PPS, DMS/WFMS etc. ) unterstützt, die alle parallel und teilweise mit überlappender Funktionalität eingesetzt werden. Neue Systeme müssen eine Architektur bzw. eine Grundkonzeption aufweisen, die eine leichte und flexible Integrationsfähigkeit in die jeweils vorhandene IT-Systemlandschaft ermöglicht. Die breite Mischung unterschiedlicher Alt-Systeme in einem heterogenen Umfeld muss parallel weiter verwendet werden können. Es müssen Informationen aus den unterschiedlichen Systemen inklusive Web-basierter Informationssysteme wie Engineering-Märkte etc. genutzt und miteinander verknüpft werden können.
- **Skalierbarkeit:**

Es müssen modular an verschiedenen Standorten spezifische Funktionalitäten ergänzt werden können. Die temporäre Integration von IT-Systemen der Kooperationspartner einer kooperativen Umgebung muss unterstützt werden. Die Systemumgebung muss sowohl für eine geringe Anzahl, als auch für eine große Anzahl an Anwendern tauglich sein. Technologien, die eine zentralisierte Pflege und Koordination von Konfiguration und Datenmodell voraussetzen, dürfen keine Anwendung finden.
- **Zuverlässigkeit:**

Der Ausfall eines Systems darf das Gesamtsystem nicht beeinträchtigen. Ausfallsicherheit innerhalb der Kooperation bedeutet dabei Unabhängigkeit vom Netzwerk, bzw. von Kommunikationsverbindungen und damit Transaktions-sicherheit.

- **Wartbarkeit:**  
Die Installations- und Wartungsproblematik ist insbesondere bei verteilten Anwendungen ein Faktor, der die Kosten und die Verfügbarkeit der Anwendungen stark beeinflusst.

### 5.3 Fazit

Die in diesem Kapitel vorgestellten Anforderungen dienen als Basis für die Erstellung des Konzeptes der ERP-OnTo-PDM. Die Anforderungen können grob in die drei Bereiche:

- organisatorische Anforderungen,
- anwenderbezogene Anforderungen,
- technische Anforderungen,

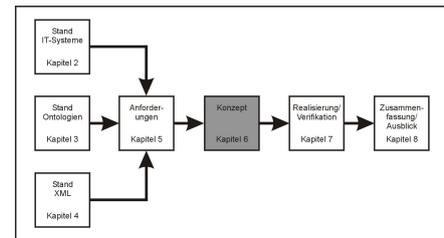
untergliedert werden.

Im nachfolgenden Kapitel wird ausführlich auf die Erstellung des Konzeptes ERP-OnTo-PDM und deren Realisierung eingegangen. Dabei wird immer wieder auf die in diesem Kapitel aufgezeigten Anforderungen an eine effiziente semantische Kopplung von ERP- und PDM-Systemen Bezug genommen.

# Kapitel 6

## Das Konzept ERP-OnTo-PDM

In diesem Kapitel wird, aufbauend auf den in den vorhergegangenen Kapiteln aufgezeigten Schwachstellen und den daraus abgeleiteten Anforderungen, das Konzept ERP-OnTo-PDM zur semantischen Kopplung von IT-Systemen mittels einer Ontologie entwickelt.



Wie aufgezeigt, entstehen an den Produktentwicklungsprozess neue Anforderungen besonders hinsichtlich der Aufteilung bei virtuellen kooperativen Arbeitsformen im Engineering-Bereich [Ger00, 110]. Weiter ist für diese Arbeitsformen der Einsatz von Internet-Standards als Informationsressource und Kommunikationsmedium, durch beispielsweise Engineering-Märkte und Produktinformationssysteme, besonders kennzeichnend. Auf die daraus resultierende strategische Bedeutung von PDM-Systemen als Integrationsmedium wurde bereits eingegangen und die Defizite der bestehenden IT-Systeme und Integrationsansätze zur Unterstützung moderner Produktentwicklungsmethoden wurden aufgezeigt. Der herausragende Vorteil einer semantischen Kopplung mittels ERP-OnTo-PDM besteht darin, dass keine komplette Integration der zu verwaltenden Daten auf Modellebene, bzw. auf Basis eines integrierten Produktmodells erfolgt. Es wird vielmehr ein semantischer Datenaustausch zur Integration von Daten in verschiedenen Erzeuger- bzw. ERP-Systemen über Metainformationen auf Basis einer Ontologie geschaffen.

Zunächst wird der generelle Aufbau dieses Konzeptes, bestehend aus seinen einzelnen Komponenten, beschrieben. Ein weiterer Abschnitt beschäftigt sich mit der Beschreibung, wie die semantische Kopplung von IT-Systemen mittels Ontologi-

en innerhalb des ERP-OnTo-PDM-Konzeptes bewerkstelligt wird. Ferner wird der Prozess des Datenaustausches mittels des ERP-OnTo-PDM-Konzeptes erläutert.

## 6.1 Aufbau des Konzeptes: ERP-OnTo-PDM

Einer der Hauptkritikpunkte an den bestehenden Konzepten zum Datenaustausch zwischen IT-Systemen ist die fehlende Möglichkeit, die Daten mittels einer Semantik zu beschreiben. Diese Schwachstelle führt dazu, dass bei der Schaffung von Systemkopplungen und bisherigen Datenaustauschformate die Syntax und damit einhergehend die Struktur der Austauschdateien zu sehr in den Mittelpunkt der Betrachtung gerät. Es müssen spezielle Sonderzeichen zur Trennung der Datenwerte von Steuerungsinformationen angegeben und ausgewertet werden, die jedoch selbst über keine weitere inhaltliche Bedeutung verfügen. Darüber hinaus ist der Aufbau von solchen Austauschdateien durch eine starre Aufbaustruktur gekennzeichnet. Um einen Datenaustausch zwischen verschiedenen IT-Systemen zu ermöglichen, müssen spezielle Konverter programmiert werden, die genau auf die Aufbaustruktur und die verwendeten Steuerzeichen abgestimmt sind. Ändert sich dieses Austauschdateiformat, müssen erhebliche Änderungen in den einzelnen Convertern vorgenommen werden. Darüber hinaus ist die Wiederverwendbarkeit solcher Converter sehr begrenzt. Des Weiteren ist das Austauschdateiformat nur sehr eingeschränkt für andere Aufgaben als für den Datenaustausch einsetzbar. Werden Mechanismen bereitgestellt, die es ermöglichen, Daten auszuzeichnen, so dass die Bedeutung der Daten ersichtlich wird und weiter ein flexibler Aufbau der auszutauschenden Daten ermöglicht wird, wird der Implementierungs- und Pflegeaufwand erheblich gesenkt. Ferner lässt sich durch eine solches neutrales Datenformat die Anzahl an benötigten Convertern erheblich senken. Das ERP-OnTo-PDM-Konzept zielt genau auf die Beseitigung dieser Schwachstellen ab. Durch die Basierung dieses Konzeptes auf einer Ontologie zur Beschreibung der einzelnen Konzepte eines IT-Systems und die Umsetzung dieser Ontologie durch XML-Technologie wird ein semantisches, neutrales Datenformat zur Datenverarbeitung in den verschiedensten IT-Systemen bereitgestellt.

In der Abbildung 6.1 ist das ERP-OnTo-PDM-Konzept schematisch dargestellt. Im Mittelpunkt des Konzeptes stehen die aus den an der Kopplung beteiligten Systemen erzeugten XML-Dateien.

Die XML-Dateien werden auf Basis der jeweiligen DTD von der jeweiligen ERP-OnTo-PDM-Software-Komponente erzeugt. Die DTD bildet für ein bestimm-

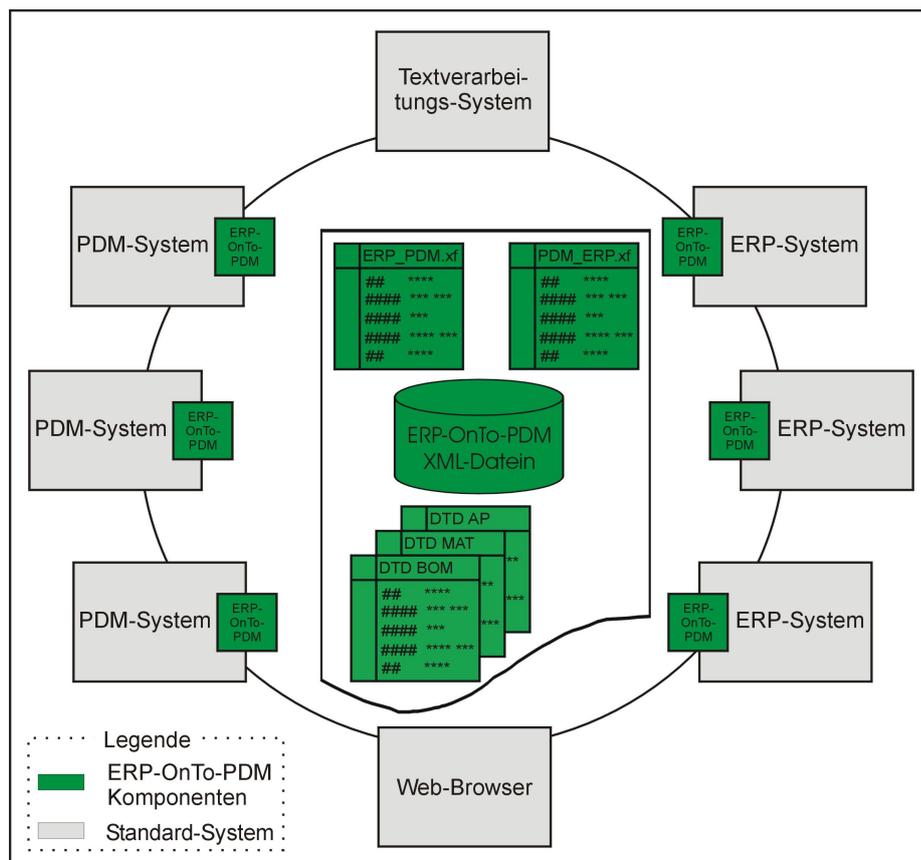


Abbildung 6.1: Aufbau des Konzeptes

tes Konzept eines IT-Systems die syntaktischen und vor allem die semantischen Regeln des jeweiligen Konzeptes ab. Eine solche DTD repräsentiert also durch die explizite Spezifikation der semantischen Constraints eines Konzeptes eine ontologische Repräsentation dieses Konzeptes. Der Aufbau einer ERP-OnTo-PDM-XML Datei wird durch Abbildung 6.2 beispielhaft für eine Stückliste gezeigt.

```

<?xml version = "1.0" ?>
<ERP_OnTo_PDM>
  <BOM_HEAD>
    <Ebene>1</Ebene>
    <Creation_Date>05/19/2003 15:25</Creation_Date>
    <Produkt_Nummer>Fert0010</Produkt_Nummer>
    <Produkt_Bezeichnung>Renntag</Produkt_Bezeichnung>
    <Masseinheit>EA</Masseinheit>
    <Preis></Preis>
    <Menge>1</Menge>
  </BOM_HEAD>
  <BOM_POS>
    <Ebene>2</Ebene>
    <Creation_Date>05/19/2003 15:31</Creation_Date>
    <Produkt_Nummer>Halb0140</Produkt_Nummer>
    <Produkt_Bezeichnung>Gabel</Produkt_Bezeichnung>
    <Masseinheit>EA</Masseinheit>
    <Preis></Preis>
    <Menge>1</Menge>
  </BOM_POS>
  <BOM_POS>
    <Ebene>2</Ebene>
    <Creation_Date>05/19/2003 15:30</Creation_Date>
    <Produkt_Nummer>Halb0130</Produkt_Nummer>
    <Produkt_Bezeichnung>Rahmen</Produkt_Bezeichnung>
    <Masseinheit>EA</Masseinheit>
    <Preis></Preis>
    <Menge>1</Menge>
  </BOM_POS>
</ERP_OnTo_PDM>

```

Abbildung 6.2: Beispiel für den Aufbau der ERP-OnTo-PDM-XML Datei zum Austausch von Stücklistendaten

Ein solches Konzept könnte beispielsweise eine Stückliste darstellen. Für dieses Konzept ist in einem Systemverzeichnis, welches durch eine URI eindeutig referenzierbar ist, eine DTD hinterlegt. Für den genauen Aufbau einer DTD sei an dieser Stelle auf den Abschnitt 6.2 weiter unten in diesem Kapitel verwiesen.

Das ERP-OnTo-PDM-Konzept lässt sich in die folgenden drei Komponenten untergliedern:

- **PDM-Komponente**  
Diese Komponente wird innerhalb des jeweiligen PDM-Systems implementiert. Es dient zur ereignisgesteuerten Erzeugung der XML-Dokumente des ERP-OnTo-PDM Konzeptes. Es handelt sich hauptsächlich um eine Überwachungsfunktion, die beim Eintreffen von bestimmten definierten Ereignissen innerhalb eines PDM-Systems die ERP-OnTo-PDM Dateien erzeugt. Die genaue Arbeitsweise wird in den Abschnitten 6.3 und 6.6 erläutert.
- **ERP-Komponente**  
Diese Komponente wird innerhalb des jeweiligen ERP-Systems implementiert. Es dient zur ereignisgesteuerten Erzeugung der XML-Dokumente des ERP-OnTo-PDM Konzeptes. Es handelt sich hauptsächlich um eine Überwachungsfunktion, die beim Eintreffen von bestimmten definierten Ereignissen innerhalb eines ERP-Systems die ERP-OnTo-PDM Dateien erzeugt. Die genaue Arbeitsweise wird in den Abschnitten 6.3 und 6.6 erläutert.
- **Ontologie-Komponente**  
Die Ontologie des ERP-OnTo-PDM-Konzeptes besteht aus einzelnen DTD-Deklarationen. Es wird für jedes Konzept, welches für einen Datenaustausch in Frage kommt, eine eigene DTD deklariert. Dies kann zum Beispiel für Stücklisten, Materialstammdaten und Arbeitspläne erfolgen. Durch die Spezifikation eines Konzeptes durch eine DTD wird insbesondere dessen Semantik explizit dargestellt. Diese explizite Definition und Repräsentation der Semantik eines Konzeptes stellt somit eine ontologische Repräsentation dieses Konzeptes, wie es im Kapitel 3 angesprochen wurde, dar. Damit sind auch die dort aufgeführten Vorteile einer Ontologie durch das ERP-OnTo-PDM-Konzept realisiert.

Die Aufgaben des ERP-OnTo-PDM-Konzeptes können wie folgt zusammengefasst werden:

- Datentransfer zwischen ERP- und PDM-Systemen
- Datentransfer zwischen PDM- und ERP-Systemen
- Datentransfer zwischen Erzeugersystemen und PDM-Systemen
- Datentransfer zwischen verschiedenen PDM-Systemen
- Datentransfer zwischen verschiedenen ERP-Systemen

- Aufbereitung der Daten in eine Web-Browser gerechte Darstellung
- Aufbereitung der Daten für die Weiterverarbeitung in anderen IT-Systemen

Das ERP-OnTo-PDM-Konzept dient nicht nur für eine vertikale Integration, also zwischen verschiedenen IT-Systemtypen (ERP- und PDM- Systemen), sondern auch für die Schaffung einer horizontalen Integration, also zwischen IT-Systemen desselben Typs. In der Praxis ist oft festzustellen, dass sich nicht nur ein einzelnes PDM-System im Einsatz befindet, sondern nebeneinander mehrere PDM-Systeme zum Einsatz kommen. Die selben Beobachtungen sind auch auf der Seite der ERP-Systeme festzustellen. Die unterschiedlichen Ebenen der Datenintegration und die Nutzung des ERP-OnTo-PDM-Konzeptes wird durch Abbildung 6.3 veranschaulicht.

Wenn diese Systeme nicht von demselben Hersteller stammen, gibt es auch innerhalb dieser IT-Systemklasse unterschiedliche Datenrepräsentationsstrukturen. Bei einem Datenaustausch auf einer horizontalen Ebene sind also dieselben Probleme wie bei einer vertikalen Datenintegration auszumachen. Es müssen hier ebenso Mechanismen zur Abbildung der Datenstruktur des PDM-Systems 1 auf die Datenstrukturen des PDM-Systems 2 bereitgestellt werden. Diese Mapping-Mechanismen müssen für beide Arten der Datenintegration ein flexibles Mapping bereitstellen, das möglichst nicht nur die Systemspezifikationen abdeckt, sondern möglichst universell Verwendung finden kann. Es ist jedoch nicht möglich, vollständig von den systeminternen Datenrepräsentationsstrukturen zu abstrahieren. Wenn demzufolge ein universeller Datenaustauschmechanismus wie ihn das ERP-OnTo-PDM-Konzept bereitstellt, implementiert werden soll, müssen die nachfolgenden beiden Komponenten bereitgestellt werden:

- **Systemabhängige Komponente**

Diese Komponente wird innerhalb des jeweiligen IT-Systems implementiert. Es beinhaltet systeminterne Funktionen, die direkt auf den Datenstrukturen des jeweiligen IT-Systems aufsetzen.

- **Systemunabhängige Komponente**

Diese Komponente beinhaltet Funktionen, die unabhängig von den jeweiligen IT-Systemen sind. Sie werden eigenständig und außerhalb der IT-Systeme realisiert. Zu dieser Komponente zählt vor allem die Ontologie des ERP-OnTo-PDM-Konzeptes mit ihrer Sammlung von XML-DTD, die Steuerungsdateien sowie die erzeugten XML-Dateien.

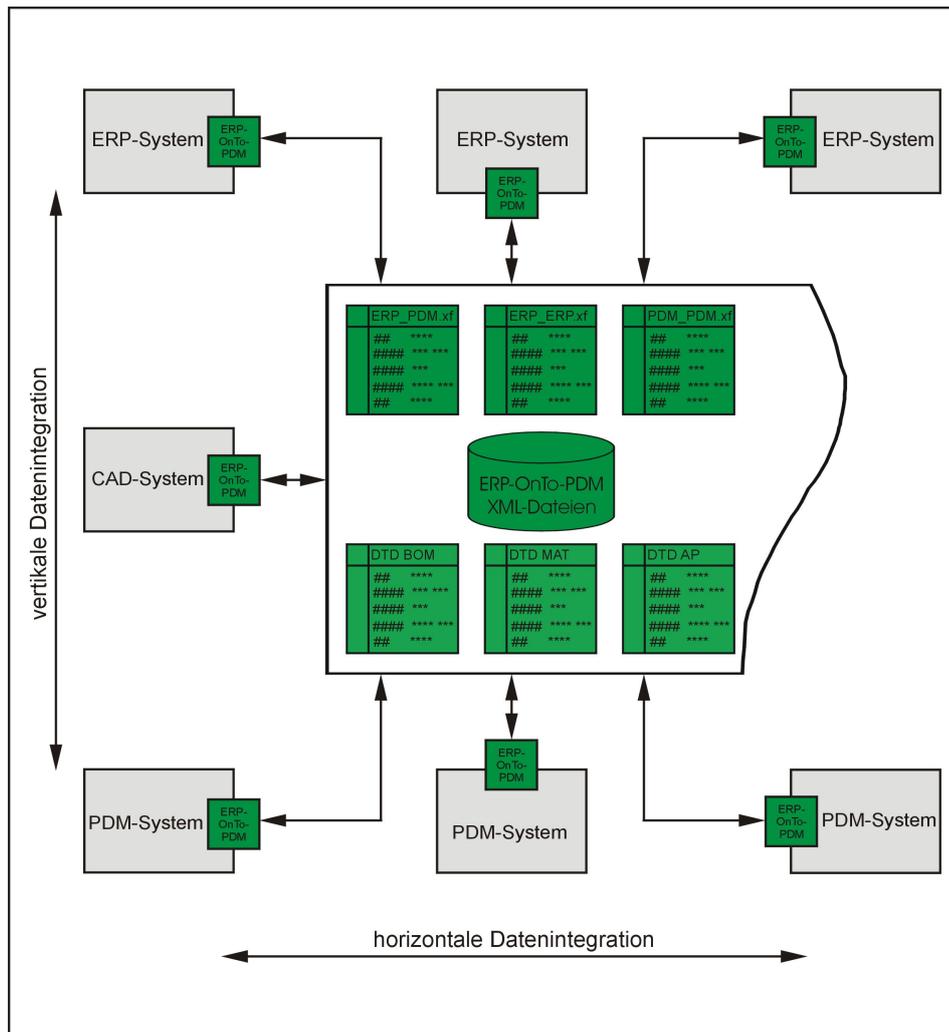


Abbildung 6.3: Die zwei Ebenen der Datenintegration

Aus Sicht einer möglichst flexiblen und wartungsarmen Implementierung ist zu berücksichtigen, dass der Anteil der systemabhängigen Funktionen möglichst gering gehalten werden sollte und möglichst viel Funktionalität in der systemunabhängigen Komponente realisiert wird.

Grundsätzlich ist jede Komponente ereignisorientiert anzulegen. Dies bedeutet, dass innerhalb der Softwarekomponenten ein Event-Handler implementiert werden sollte, der eine Überwachung und ein Auslösen von Ereignissen ermöglicht. Hieraus folgt, dass zur Implementierung dieser Softwarekomponenten Programmiersprachen Verwendung finden sollten, die eine Ereignissteuerung und Ereignisverarbeitung ermöglichen. Des Weiteren sollte diese Programmiersprache in möglichst vielen IT-Systemen integrierbar sein. Um den Pflegeaufwand für die einzelnen Komponenten des ERP-OnTo-PDM-Konzeptes möglichst gering zu halten, sollte die Anzahl von verwendeten Programmiersprachen auf ein Minimum reduziert werden. Da in bestehende IT-Systeme Komponenten zur Manipulationen von deren internen Datenstrukturen zu realisieren sind, sollte eine Script-Sprache Verwendung finden. Gerade bei IT-Systemen, die auf der Basis des Betriebssystems Microsoft Windows aufsetzen, ist in den meisten Systemen eine Schnittstelle für die Integration von Microsoft Visual Basic bzw. Microsoft Visual Basic Script vorhanden.

Die folgenden Ziele stehen im Vordergrund des ERP-OnTo-PDM-Konzeptes:

- **Einfache Integration** Das ERP-OnTo-PDM-Konzept mit seinen Komponenten bietet die Möglichkeit einer einfachen Integration in die vorhandene IT-Infrastruktur bzw. die vorhandene PDM-ERP-Umgebung verschiedener Hersteller innerhalb eines Unternehmens oder eines Unternehmensverbundes mit wenig Anpassungs- und Konfigurationsaufwand.
- **Semantische Anreicherung**  
Die auszutauschenden Daten der einzelnen Konzepte der jeweiligen Systeme werden mit Meta-Daten angereichert, um einen semantischen Datenaustausch mit anschließender Datenverarbeitung zu gewährleisten. Der Datenaustausch auf Basis einer herkömmlichen Dateischnittstelle mittels eines individuellen ASCII-Formats unterliegt schweren Einschränkungen. Unter anderem erweist sich der Aufbau dieser Dateien als sehr unflexibel. Es müssen bestimmte Sonderzeichen als Separator zwischen einzelnen Feldwerten festgelegt werden. Der Aufbau der Austauschdatei selbst unterliegt auch einer starren Struktur. Wird der strukturelle Aufbau der Austauschdatei verändert, muss der Parser zum Einlesen der Austauschdatei unter erheblichem

Aufwand angepasst werden. Durch die Verwendung von ERP-OnTo-PDM und den zugrundeliegenden XML-Mechanismen wird dieser Änderungsaufwand minimiert. Der Zugriff auf einzelne Datenwerte erfolgt mittels des Zugriffs auf die die Werte umschließenden Tags. Durch diese Technik muss die Austauschdatei nicht mehr sequentiell verarbeitet werden. Es ist ein freier Zugriff auf einzelne Werte innerhalb der ERP-OnTo-PDM-Datei über die einzelnen Tags möglich.

- **Medienunabhängigkeit**

Wünschenswert ist eine einmalige Speicherung der technischen Produktdokumentation in einem Format, welches die Weiterverarbeitung für verschiedene Medien und IT-Systeme gestattet, ohne jeweils spezielle Konvertierungsprogramme benutzen zu müssen.

## 6.2 Abbildung der Ontologie

Ontologien bilden den Kern des ERP-OnTo-PDM-Konzeptes. Ontologien dienen hier dem semantischen Produktdaten-Austausch zwischen verschiedenen IT-Systemen, die produktrelevante Daten managen bzw. diese produktrelevante Daten verarbeiten und für die Wiedergabe auf verschiedenen Medien aufbereiten.

Ein wichtiger Punkt bezogen auf den gemeinsamen Datenzugriff bzw. Datenaustausch zwischen verschiedenen Subsystemen ist das semantische Mapping, d.h. die Abbildung von Informationen mit einer unterschiedlichen Bedeutung aufeinander. Verfügbare PDM-Systeme stellen umfangreiche Mechanismen für das Management von komplexen strukturierten Daten und Dokumenten zur Verfügung. Eine der wichtigsten Funktionalitäten dabei ist die Fähigkeit der betriebsspezifischen Anpassung (Customizing), d.h. der Anpassung des internen Datenmodells an die Bedürfnisse des Unternehmens.

Im Bereich Geometrieaustausch, bzw. CAD, hat sich neben den proprietären Datenformaten der einzelnen Systeme das STEP-AP-214 etabliert und bietet auch Mechanismen zum Austausch organisatorischer Daten, wie sie in PDM-Systemen benötigt werden. Im PDM-Bereich gibt es kein einheitliches, sondern proprietäre Datenformate für den Austausch von Informationen zwischen verschiedenen PDM-Umgebungen, da jede Implementierung unterschiedlich ist, selbst wenn sie auf Basis des gleichen PDM-Systems realisiert wurde. Das STEP-PDM-Schema definiert ein zu starres Modell, das nur einen kleinsten gemeinsamen Nenner von benötigten

Informationen abdecken kann. Die PDM-Enablers-Spezifikation für einen gemeinsamen Datenzugriff ist sehr generisch, d.h. lässt eine Menge Implementierungsspielraum. Momentan basiert keines der am Markt verfügbaren PDM-Systeme auf einem Datenmodell, welches exakt das STEP-PDM-Schema abbildet. Daher wären Prä- bzw. Postprozessoren notwendig, die das verwendete Datenmodell auf das STEP-Datenmodell abbilden. Dies gilt nicht pro PDM-System, sondern pro angepasster Implementierung, was für die Systemhersteller einen nicht zu bewältigenden Aufwand darstellt, bzw. wirtschaftlich nicht tragfähig ist. Weder das STEP-PDM-Schema noch die PDM-Enablers stellen also geeignete Mechanismen bereit, um auf semantischer Ebene einen korrekten Transport von Informationen sicherzustellen [EKM99].

Ein Beispiel für eine Attributsemantik ist die „sprechende“ Nummer. Es kann auch sinnvoll sein, dass beispielsweise die Teilenummer aus einem Quellsystem nicht als Teilenummer im Zielsystem verwendet werden soll, sondern als Kommentar (z.B. Herstellernummer). Problematisch beim semantischen Mapping auf Attributebene ist, dass selbst wenn im Quell- und Zielsystem gleiche semantische Attribute identifiziert sind, nicht sichergestellt ist, dass dies für alle Instanzen des Attributes gilt. Es können auch informationstechnische Restriktionen vorliegen, beispielsweise sind für ein Kommentarfeld im Zielsystem weniger Zeichen vorgesehen als für das des Quellsystems, was dann zu einem Informationsverlust führen kann.

Aus diesen Gründen ist oftmals ein vollautomatisches Mapping von Informationen mit unterschiedlicher Semantik nicht möglich. Das Konzept sieht einen Mechanismus vor, mit dem halbautomatisch Daten aus verschiedenen Subsystemen der föderierten Gesamtumgebung aufeinander abgebildet werden. Dabei muss bei einem initialen Datenaustausch die Relation zwischen Informationen manuell konfiguriert werden. Diese Konfiguration wird gespeichert. In nachfolgenden Datenaustauschvorgängen, bei denen Quelle und Ziel gleich sind, steht diese Konfiguration wieder zur Verfügung und kann herangezogen werden.

Wie weiter oben in diesem Kapitel beschrieben, liegt einer der Hauptvorteile des ERP-OnTo-PDM-Konzeptes in der Verwendung von DTDs zur Abbildung der Ontologie. Hierdurch wird das semantische Mapping zwischen den einzelnen IT-Systemen ermöglicht. Weiter wird hierdurch eine flexible Erweiterungsmöglichkeit geschaffen. Durch das Anlegen von DTDs für einzelne Konzepte wird die hinter jedem Konzept stehende Bedeutung explizit spezifiziert und eine semantische Interpretation beigegeben. Dies entspricht der Arbeitsweise von Ontologien. Die Ontologie für des ERP-OnTo-PDM-Konzeptes setzt sich also aus der Summe der

einzelnen DTDs zusammen. Dieser Sachverhalt wird durch die Abbildung 6.4 grafisch veranschaulicht.

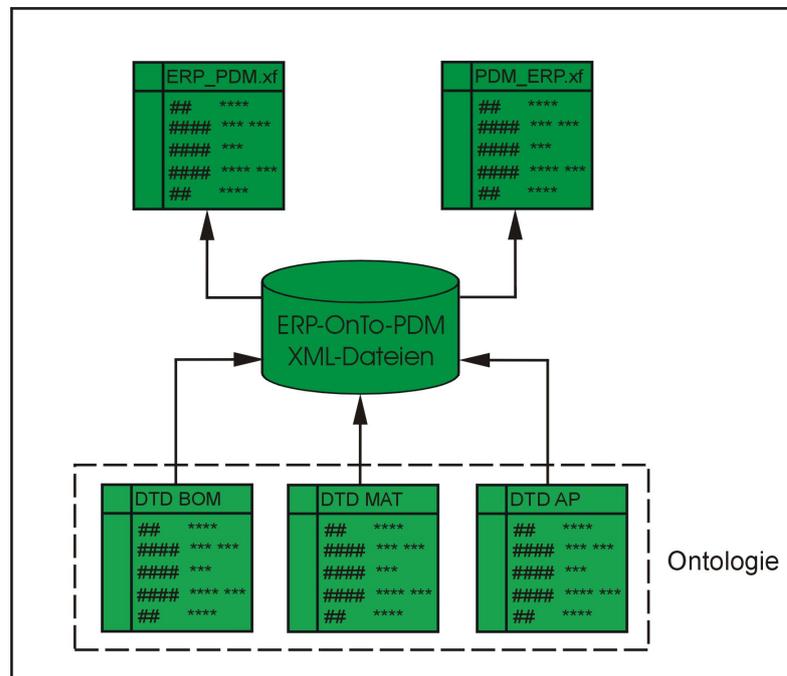


Abbildung 6.4: Abbildung der Ontologie

Die semantischen Constraints für die Ontologie wird durch die Verwendung der in Abbildung 6.5 aufgeführten Datentypen bei der Erstellung der DTDs erreicht.

Um das ERP-OnTo-PDM-Konzept für eine breite Öffentlichkeit zugänglich zu machen, sind die DTDs nicht in die XML-Dateien eingebettet, sondern in einem speziellen Systemverzeichnis abgelegt. Die Einbindung der jeweiligen DTD in die dazu korrespondierende XML-Datei erfolgt unter der Verwendung des Konzeptes von XML-Namensräumen. Durch die Deklaration von Namensräumen in einem XML-Dokument und die Vergabe von Namensraumpräfixen kann die eindeutige Zuordnung von Elementen zu einer semantischen Einheit erfolgen.

### 6.3 Einbindung des Konzeptes in PDM-Systeme

Wie schon bei der Beschreibung des Aufbaus des ERP-OnTo-PDM-Konzeptes angeführt, sollte die Softwarekomponente innerhalb der PDM-Systeme ereignisorientiert implementiert werden. Damit kann besonders flexibel auf die verschiedenen

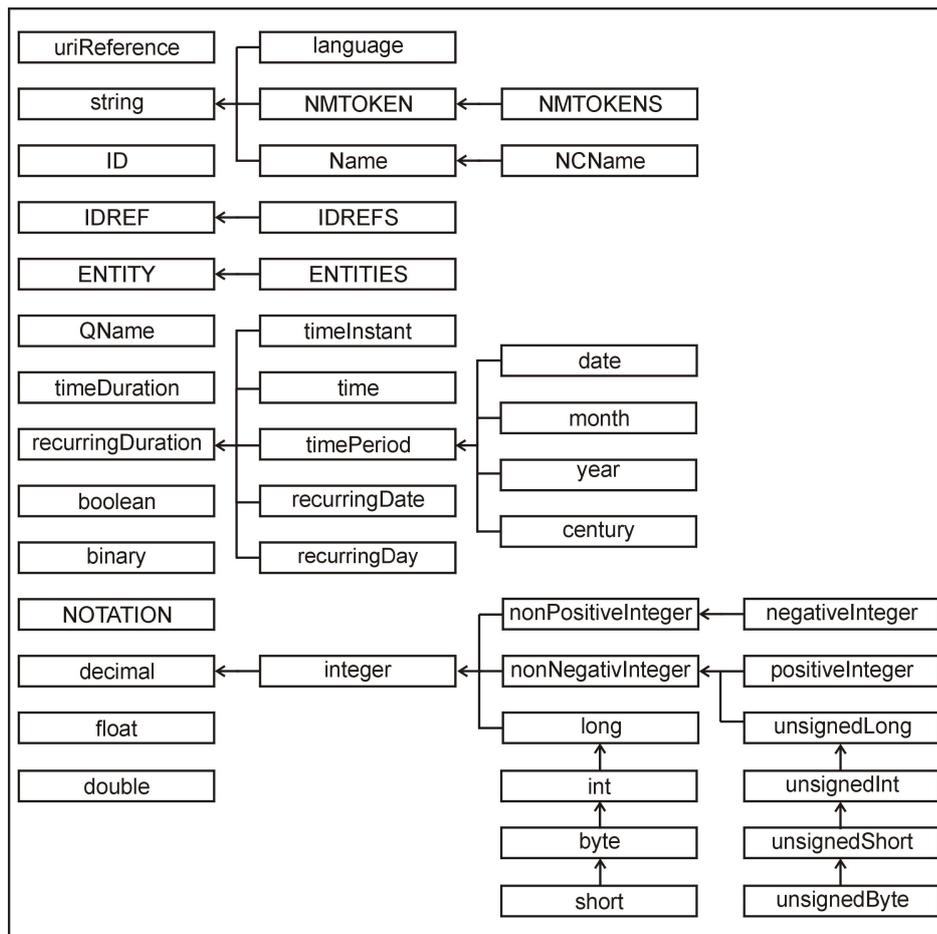


Abbildung 6.5: Datentypen XSL

Arten zum Auslösen eines Datenaustausches innerhalb eines Systemverbundes eingegangen werden. Die Softwarekomponente innerhalb eines PDM-Systems kann in die folgenden vier Funktionsblöcke untergliedert werden:

- **Kommunikationsfunktionen**

In diesen Funktionsblock fällt vor allem der Mechanismus zur Steuerung und Überwachung des Datenaustausches. Diese Funktionen erzeugen bzw. lesen die Steuerungsdateien, wie sie im Abschnitt 6.6 genauer beschrieben werden. Sie dienen zur Realisierung eines ereignisgesteuerten Datenaustausches, der eine flexible Datenkommunikation zwischen den einzelnen IT-Systemen ermöglicht.

- **Datenmanipulationsfunktionen**

Diese Funktionen dienen zum Auslesen der Daten aus den systemspezifischen Repräsentationen und der Konvertierung in das systemunabhängige, ontologiebasierte Datenformat des ERP-OnTo-PDM-Konzeptes.

- **Ausnahmebehandlungsfunktionen** Hierunter sind Funktionen zusammenzufassen, die für ein fehlertolerantes Arbeiten des ERP-OnTo-PDM-Konzeptes sorgen. Sie dienen der Behandlung von Fehlern bei der Datenübertragung, bzw. zur Reaktion auf fehlerhafte Benutzereingaben.

- **Hilfsfunktionen**

Hierunter fallen Funktionen, die einer erhöhten Benutzerfreundlichkeit dienen und eine flexible Nutzung des ERP-OnTo-PDM-Konzeptes unterstützen.

Eine der zentralen Anforderungen zur erfolgreichen Umsetzung von Softwareprojekten stellt zweifelsfrei die Benutzerfreundlichkeit dar. Aus diesem Grund sieht das ERP-OnTo-PDM-Konzept Funktionen zur flexiblen Konfiguration der zu transferierenden Daten vor. Mittels dieser Funktion besitzt der Benutzer die Möglichkeit, für jedes Konzept neben den Mussdatenfeldern eine Reihe von optionalen Daten zum Transfer auszuwählen. Um die höchstmögliche Flexibilität und Konfigurierbarkeit zu erreichen, sieht das ERP-OnTo-PDM-Konzept vor, dass die vom Benutzer zum Datenaustausch selektierten Datenfelder in einer Konfigurationsdatei hinterlegt werden. Diese Konfigurationsdatei wird von der Funktion zum Auslesen der Daten innerhalb der PDM-Systeme dann verwendet und so ein flexibler Datenaustausch gewährleistet.

Bei den PDM-Systemen ist neben der Schaffung von Mechanismen zum Datenaustausch mit ERP-Systemen auch die Möglichkeit zur Realisierung einer Da-

tenintegration mit vorgelagerten Erzeugersystemen, wie sie die CAD-Systeme darstellen, zu berücksichtigen. Eine Implementierung einer solchen Datenintegration würde den Rahmen dieser Arbeit überschreiten, jedoch sei an dieser Stelle betont, dass das ERP-OnTo-PDM-Konzept zur Schaffung eines ontologiebasierten neutralen Datenbeschreibungsformats auch diesen Anforderungen genügen kann und auf diese angepasst werden kann.

Als zentrale Funktion innerhalb der in einem PDM-System zu realisierenden Komponente des ERP-OnTo-PDM-Konzeptes ist ein Mechanismus zum Lesen und Schreiben auf den systeminternen Datenstrukturen zu sehen. Dieser Mechanismus transferiert in der einen Richtung das systemspezifische Datenformat in das neutrale Datenformat des ERP-OnTo-PDM-Konzeptes. In der anderen Richtung wandelt dieser Mechanismus die an ein PDM-System übertragenen Daten aus dem neutralen Datenformat in die systeminternen Datenstrukturen um und ermöglicht so das Neuanlegen von Daten aus anderen Systemen im PDM-System bzw. das Updaten von vorhanden Datensätzen.

Des Weiteren sieht das ERP-OnTo-PDM-Konzept vor, dass der Benutzer den gesamten Prozess des Datenaustausches aus dem PDM-System heraus steuern kann. Es ist demzufolge nicht notwendig, dass der Benutzer die Systemumgebung des PDM-Systems verlassen muss. Vielmehr kann er von der Initialisierung des Datentransfers, bis zur Überprüfung der Ergebnisse in beispielsweise einem ERP-System, in seiner vertrauten Programmumgebung bleiben. Dies führt zu einer höheren Akzeptanz auf Seiten der Benutzer des ERP-OnTo-PDM Systems, da kein wesentlicher Lernaufwand für ein neues Softwaresystem entsteht. Zur Realisierung dieser Anforderung ist die Nutzung von Remote Function Calls (RFC) vorgesehen. Mittels eines RFC-Aufrufs ist es einem Programm möglich, Funktionen eines anderen IT-Systems in dessen Umgebung aufzurufen. Beispielfähig sei hier die Möglichkeit aufgeführt, dass der Benutzer eines PDM-Systems nach erfolgreichem Übertragen von Stücklisten mittels des ERP-OnTo-PDM-Konzeptes an ein ERP-System sich die Stücklisten des ERP-Systems innerhalb des PDM-Systems betrachten kann.

## 6.4 Einbindung des Konzeptes in ERP-Systeme

Analog zu der Softwarekomponente des ERP-OnTo-PDM-Konzeptes, die innerhalb eines PDM-Systems realisiert wird, können auch für die Softwarekomponente, die innerhalb eines ERP-Systems realisiert wird, die folgenden vier Funktionsblöcke

unterschieden werden:

- **Kommunikationsfunktionen**

In diesen Funktionsblock fällt vor allem der Mechanismus zur Steuerung und Überwachung des Datenaustausches. Diese Funktionen erzeugen bzw. lesen die Steuerungsdateien, wie sie im Abschnitt 6.6 genauer beschrieben werden. Diese Funktionen dienen zur Realisierung eines ereignisgesteuerten Datenaustausches, der eine flexible Datenkommunikation zwischen den einzelnen IT-Systemen ermöglicht.

- **Datenmanipulationsfunktionen**

Diese Funktionen dienen zum Auslesen der Daten aus den systemspezifischen Repräsentationen und der Konvertierung in das systemunabhängige, ontologiebasierte Datenformat des ERP-OnTo-PDM-Konzeptes.

- **Ausnahmebehandlungsfunktionen** Hierunter sind Funktionen zusammenzufassen, die für ein fehlertolerantes Arbeiten des ERP-OnTo-PDM-Konzeptes sorgen. Sie dienen der Behandlung von Fehlern bei der Datenübertragung bzw. zur Reaktion auf fehlerhafte Benutzereingaben.

- **Hilfsfunktionen**

Hierunter fallen Funktionen, die einer erhöhten Benutzerfreundlichkeit dienen und eine flexible Nutzung des ERP-OnTo-PDM-Konzeptes unterstützen.

## 6.5 Einbindung des Konzeptes in andere IT-Systeme

Wie schon an verschiedenen Stellen dieser Arbeit angesprochen, wird eine integrierte Datenverarbeitung für die verschiedenen Medien immer wichtiger. Hier spielen besonders die Anwendungen des E-Business und E-Commerce eine wichtige Rolle. Der Bereich des Business-to-Consumer und vor allem der Bereich Business-To-Business mit ihren verschiedenen IT-Systemen und proprietären Daten-Formaten sind hier zu nennen. Produktinformationen, die in einem PDM-System primär für unternehmensinterne Zwecke gespeichert sind, können zur Erstellung von Produktbeschreibungen für den Verbraucher herangezogen werden. Diese Produktbeschreibungen können traditionell in gedruckter Form oder in elektronischer

Form über einen Internetauftritt dem Verbraucher bereitgestellt werden. In beiden Fällen müssen die Produktinformationen aus dem PDM-System jedoch mit für den jeweiligen Medien-Typ speziellen IT-Werkzeugen aufbereitet werden. Eine direkte Verarbeitung dieser Daten war bisher nur sehr eingeschränkt möglich. Die verschiedenen Formen des Datenaustausches bei E-Business-Anwendungen ist durch Abbildung 6.6 dargestellt. Im ungünstigsten Fall wurden die Daten aus dem PDM-System manuell neu in die entsprechenden Werkzeuge eingepflegt. Das ERP-OnTo-PDM-Konzept bietet durch die Verwendung von XML-Technologie die Möglichkeit einer medienunabhängigen Verarbeitung der Daten. Durch den in das PDM-System integrierten Konverter werden die Daten soweit semantisch mit Metadaten angereichert, dass eine automatische Weiterverarbeitung mit Hilfe von Skript- bzw. Stylesheetsprachen ermöglicht wird. Eine den Belange des ERP-OnTo-PDM-Konzeptes entsprechende Stylesheetlanguage ist die XSLT.

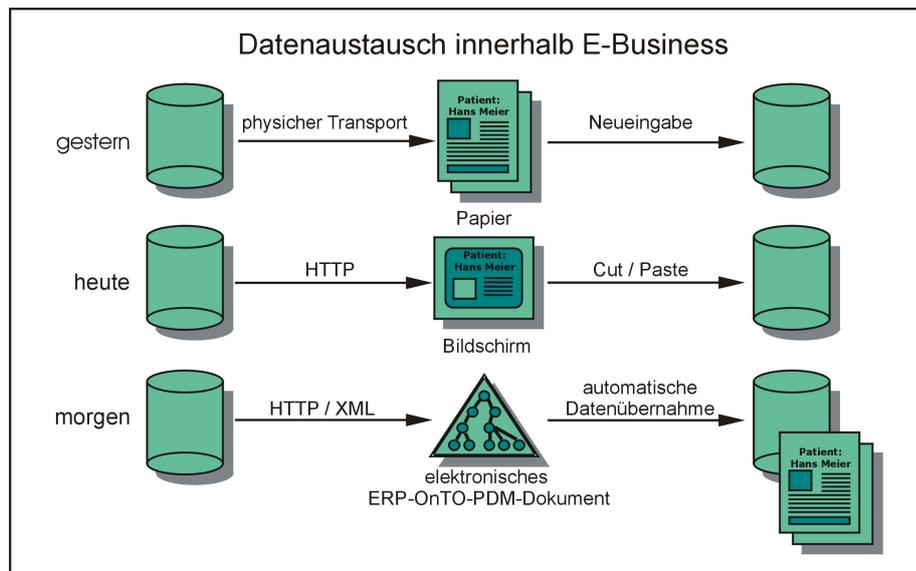


Abbildung 6.6: Arten des Datenaustausches innerhalb des E-Business

Als Anwendungen für die Weiterverarbeitung kommen z.B. die folgenden in Frage:

- **Textverarbeitungssysteme**

Um gedruckte Produktbeschreibungen zu generieren, werden Textverarbeitungssysteme verwendet. Diese sind auf die speziellen Belange des Erzeugens eines ansprechenden Schriftbildes für Druckwerke ausgelegt. Moderne Textverarbeitungssysteme, wie beispielhaft Microsoft Winword 2000 sind auf die

Verarbeitung von XML-basierten Dokumenten ausgelegt. Mittels einer speziellen Stylesheet-Datei ermöglicht das ERP-OnTo-PDM-Konzept die in einer ERP-OnTo-PDM-XML Datei enthaltenen Daten automatisch in ein solches Textverarbeitungssystem zu übernehmen.

- **Web-Autorensysteme**

Diese Systeme sind für die Erstellung und Veröffentlichung von Dokumenten im Internet ausgelegt. Sie ermöglichen die auf die Bildschirmdarstellung optimierte Darstellung von Dokumenten im HTML bzw. XHTML Format. Mittels einer speziellen Stylesheet-Datei ermöglicht das ERP-OnTo-PDM-Konzept die in einer ERP-OnTo-PDM-XML Datei enthaltenen Daten automatisch in ein solches Web-Autorensystem zu übernehmen.

Die Arbeitsweise von ERP-OnTo-PDM mittels Stylesheet Dateien ist in Abbildung 6.7 dargestellt.

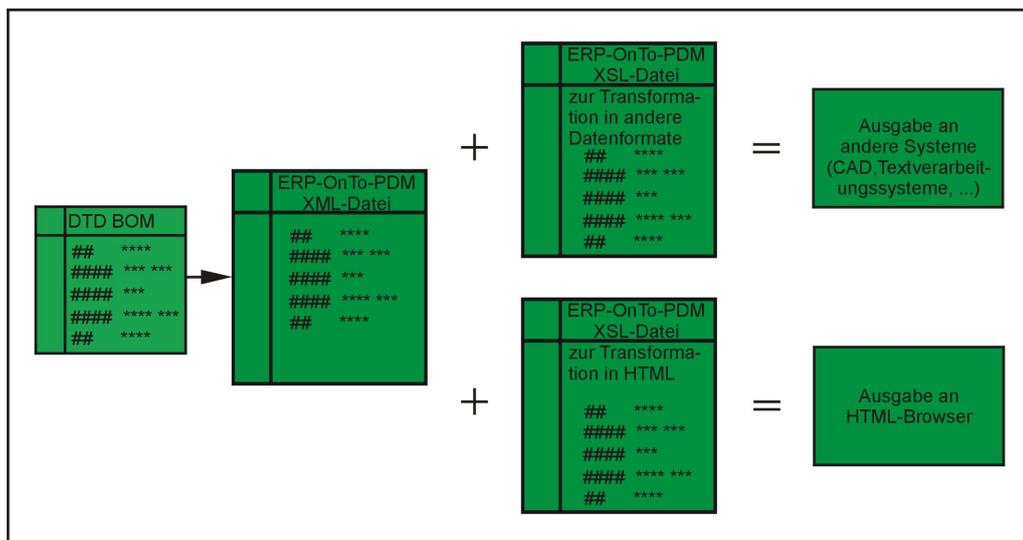


Abbildung 6.7: Aufbereitung der ERP-OnTo-PDM Daten

## 6.6 Kommunikation mittels ERP-OnTo-PDM

Die Kommunikation zwischen den an der Kopplung beteiligten Systemen ist innerhalb des ERP-OnTo-PDM-Konzeptes ereignisorientiert organisiert. Dazu wird innerhalb eines jeden Systems eine Klasse implementiert, die auf entsprechende Ereignisse reagiert. Solche Ereignisse können die folgenden sein:

- Anlegen eines neuen Konzeptes
- Aktualisieren eines Konzeptes
- Anfordern der Daten eines Konzeptes eines anderen Systems

Die Ereignisse können sowohl von einem Benutzer ausgehen, als auch durch ein System-Ereignis ausgelöst werden. Beispielsweise kann ein Benutzer eines PDM-Systems einen neuen Arbeitsplan innerhalb des PDM-Systems anlegen. Das PDM-System übernimmt hierbei die Funktion des Leitsystems. Um einen konsistenten Datenbestand zu gewährleisten, sind jedoch Informationen aus dem ERP-System erforderlich. Es ist unter anderem ein Abgleich der Arbeitsplannummer aus den beteiligten Systemen notwendig. Durch das Benutzer-Ereignis wird so eine Anfrage über eine Arbeitsplannummer generiert. Dieses Ereignis führt zu einer Kommunikation zwischen den beteiligten Systemen, die aus den beiden nachfolgenden Komponenten besteht:

- Kontroll-Datenfluss
- Nutz-Datenfluss.

Die Kommunikation und die damit verbundenen beiden Datenströme werden durch das ERP-OnTo-PDM-Konzept auf Basis einer semantischen Dateischnittstelle realisiert. Das ereignisauslösende System generiert dazu seine Steuerungsinformationen und die zu übertragenden Daten in separaten Dateien. Softwaretechnisch werden diese Dateien in ein global spezifiziertes Austauschverzeichnis abgelegt. Dieses Verzeichnis ist durch ein URI eindeutig zu adressieren. Hierbei wird das weiter oben angesprochene Konzept des XML-Namespaces verwendet. Hierdurch wird der Koordinations-Overhead der beteiligten System auf ein Minimum reduziert. Für die im Rahmen dieser Arbeit besonders fokussierte Kopplung von ERP- und PDM-Systemen können abhängig von der Datenflussrichtung, die in folgenden Dateinamenskonventionen unterschieden werden:

- Von einem ERP-System zu einem PDM-System: ERP\_PDM.xf.
- Von einem PDM-System zu einem ERP-System: PDM\_ERP.xf.
- Von einem ERP-System zu einem ERP-System: ERP\_ERP.xf.
- Von einem PDM-System zu einem PDM-System: PDM\_PDM.xf.

Sowohl die Steuerdatei, als auch die Nutzdatendatei sind valide, wohlgeformte XML-Dateien. Mittels der Steuerdatei werden die angeforderten Aktionen und die dazu korrespondierenden Nutzdateien kommuniziert. Der Aufbau einer solchen Steuerungsdatei ist durch Abbildung 6.8 beispielhaft dargestellt.

```
<?xml version="1.0" standalone="yes" ?>
<Control_File>
  <Event>
    <Command>Create_Mat_Nr</Command>
    <Use_File>Matr123.xml</Use_File>
  </Event>
  <Event>
    <Command>Update_BOM</Command>
    <Use_File>BOM456.xml</Use_File>
  </Event>
</Control_File>
```

Abbildung 6.8: Aufbau der Steuerungsdatei

In dieser beispielhaft angeführten Steuerungsdatei ist die Situation nachgestellt, dass ein Benutzer eine neue Materialnummer anfordert, die mittels der Nutzdatendatei Mat123.xml übertragen werden soll. Weiter soll ein Update der durch die Nutzdatendatei BOM456.xml spezifizierten Materialstückliste vorgenommen werden. Um die Kommunikation durchzuführen, ist in jedem an der Kopplung beteiligten System ein Event-Handler implementiert, der durch eine FileEvent-Steuerung oder ein äquivalentes Konzept das Austauschverzeichnis überwacht, eine vorhandene Steuerungsdatei verarbeitet und löscht bzw. selber eine Steuerungsdatei anlegt. Durch die Verwendung des XML-Namespace-Konzeptes zur Lokalisierung und Identifikation der Steuerungsdatei ist eine einfache und flexible Konfiguration des Austauschverzeichnisses möglich. Der Event-Handler sollte frei konfigurierbar sein und auf die einzelnen Ereignisse sowie sich ändernde Systemzustände anpassbar zu sein. Ein solcher Systemzustand könnte zum Beispiel eine sich ändernde Übertragungsgeschwindigkeit sein. Wenn sich die Performance der Kommunikation aufgrund einer geringeren Übertragungsgeschwindigkeit verschlechtert, sollte die Time-Out-Zeiten heraufgesetzt werden. Nach erfolgreicher Verarbeitung einer Steuerungsdatei wird diese gelöscht. Dementsprechend muss der Event-Handler das Austauschverzeichnis auf existierende Steuerungsdateien überprüfen, bevor er eine Steuerungsdatei in das Verzeichnis speichert. Sollte eine Steuerungsdatei vorhanden sein, deutet dies auf einen noch nicht abgeschlossenen bzw. einen fehlerhaften Datenaustausch hin. Durch das Lesen eines entsprechenden Zustandsfeldes

innerhalb der Steuerungsdatei kann der Event-Handler den Zustand des Datenaustausches feststellen. Wenn dieser fehlerhaft sein sollte, sind entsprechende Fehlerbehebungsmaßnahmen durchzuführen.

## 6.7 Ablauf des Datentransfers

An dieser Stelle soll die Arbeitsweise des ERP-OnTo-PDM-Konzeptes eingehender betrachtet werden. Dazu wird der Ablauf eines Datenaustausches für einen Materialstammsatz zwischen einem PDM- und einem ERP-System mittels ERP-OnTo-PDM beispielhaft verwendet. Ausgangspunkt ist das Ereignis, dass ein Benutzer eines PDM-Systems ein neues Material anlegt. Um einen konsistenten Datenbestand zu gewährleisten, ist es unter anderem notwendig, die Materialnummer mit dem ERP-Datenbestand anzugleichen.

Es sind folgende zwei Fälle möglich:

- Materialdaten sind bereits im ERP-System vorhanden
- Materialdaten sind noch nicht im ERP-System vorhanden

Abhängig davon ob der Materialstammsatz schon in dem ERP-System vorhanden ist oder nicht werden die Daten entweder vom PDM-System an das ERP-System übertragen bzw. vom ERP- zum PDM-System. Der Ablauf dieses Datenaustausches wird durch die in Abbildung 6.9 dargestellte Prozesskette veranschaulicht.

## 6.8 Fazit

Wie aufgezeigt, entstehen an den Produktentwicklungsprozess neue Anforderungen gerade wegen verteilter bzw. virtueller kooperativer Arbeitsformen im Engineering-Bereich [Ger00, 110]. Weiter ist für diese Arbeitsformen der Einsatz von Internet-Standards als Informationsressource und Kommunikationsmedium durch beispielsweise Engineering-Märkte und Produktinformationssysteme besonders kennzeichnend. Auf die daraus resultierende strategische Bedeutung von PDM-Systemen als Integrationsmedium wurde bereits eingegangen und die Defizite der bestehenden IT-Systeme und Integrationsansätze zur Unterstützung moderner Produktentwicklungsmethoden wurde aufgezeigt. Der herausragende Vorteil einer semantischen

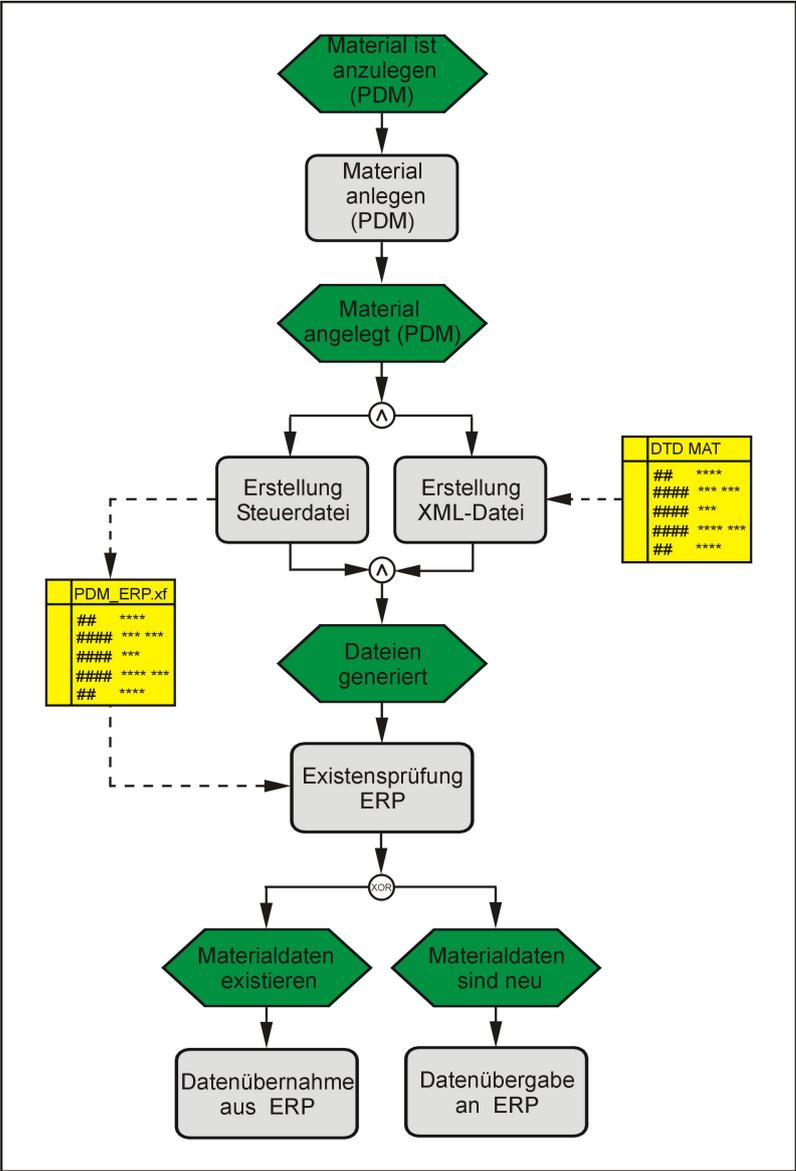


Abbildung 6.9: Prozesskette: Material anlegen

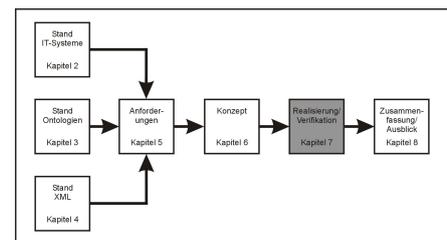
Kopplung mittels ERP-OnTo-PDM besteht darin, dass keine komplette Integration der zu verwaltenden Daten auf Modellebene, bzw. auf Basis eines integrierten Produktmodells erfolgt. Es wird vielmehr ein semantischer Datenaustausch zur Integration von Daten in verschiedenen Erzeuger- bzw. ERP-Systemen über Metainformationen auf Basis einer Ontologie geschaffen.

Das Konzept der ERP-OnTo-PDM ermöglicht so eine verteilte Datenhaltung mit einer kontrollierten Redundanz und einer Überwachung der Konsistenz von Datenbeständen innerhalb dieses Systemverbundes. Es steht so im Gegensatz zu den meisten bisherigen zentralisierten, monolithischen Integrationsansätzen mit einer möglichst kompletten und vollständig automatisierten Integration unterschiedlicher heterogener IT-Systeme.

# Kapitel 7

## Realisierung und Anwendung von ERP-OnTo-PDM

Das auf Grundlage der aufgezeigten Schwachstellen und den daraus abgeleiteten Anforderungen an eine semantische Kopplung von Informationsverarbeitungssystemen erarbeitete Konzept ERP-OnTo-PDM ist im Kapitel 6 eingehend vorgestellt



worden. In diesem Kapitel soll nun beschrieben werden, wie bei der Realisierung des Konzeptes ERP-OnTo-PDM vorgegangen wurde, und es soll die Einsetzbarkeit dieses Konzeptes demonstriert werden. Dazu wird zunächst auf das Vorgehen zur Erstellung der Ontologie für das Konzept ERP-OnTo-PDM eingegangen. Daran anschließend wird die Anwendbarkeit durch eine prototypische Implementierung nachgewiesen. Für die prototypische Entwicklung der Software-Bausteine steht aus Systemsicht eine hohe Flexibilität, ein modularer Aufbau der Funktionen, Betriebssystemunabhängigkeit, sowie die Möglichkeit der Anbindung an bereits vorhandene Systeme im Vordergrund. Aus Anwendungssicht sind besonders die Benutzerfreundlichkeit und eine Ausrichtung auf die potenziellen Anwender durch Benutzung der Sprache des Ingenieurs die wichtigsten Anforderungen, die bei der prototypischen Realisierung von ERP-OnTo-PDM zu berücksichtigen sind.

## 7.1 Verwendete Vorgehensmodelle

Zur Erstellung von ERP-OnTo-PDM wurde ein zweistufiger Erstellungsprozess verfolgt. In der ersten Phase wurde die Kern-Ontologie und in der zweiten Phase die Schnittstelle zwischen den Anwendungssystemen entwickelt. Um ein strukturiertes Vorgehen beider Entwicklungsprozesse zu gewährleisten, wurden für diese Prozesse verschiedene Vorgehensmodelle verwendet. Der Vorteil bei der Verwendung zweier Vorgehensmodelle ist in der Spezialisierung dieser Modelle auf die einzelnen Entwicklungsaufgaben zu sehen. Die verwendeten Vorgehensmodelle mit ihren Besonderheiten werden in den beiden nachfolgenden Abschnitten beschrieben.

### 7.1.1 Vorgehensmodell zur Erstellung der Ontologie: ERP-OnTo-PDM

Es gibt zur Zeit weder Richtlinien, Methoden noch Standards zur Entwicklung von Ontologien. Das Entwickeln von Vorgehensmodellen zur Erstellung von Ontologien bildet einen der zentralen Forschungsschwerpunkte in der aktuellen Ontologieforschung. Um eine möglichst hohe Wiederverwendbarkeit von Ontologien zu erreichen, sollten Ontologien als kleine Module mit einer hohen internen Kohärenz und einer limitierten Anzahl von Interaktionen zwischen den einzelnen Modulen konzipiert werden. Diese und anderen Anforderungen an die Entwicklung von Ontologien sind in den Design Prinzipien für Ontologien zusammengefasst [Gru95][Gua95][UG96].

Auf der Grundlage dieser Design-Prinzipien wurden für die Entwicklung der Ontologie für ERP-OnTo-PDM die nachfolgenden Fragen abgeleitet und schrittweise beantwortet:

- Was soll spezifiziert werden?  
(Eine für das ganze Team einheitliche und eindeutige Definition von Ontologien muss festgelegt werden)
- Wer soll die Ontologie benutzen und auf welche Weise soll die Benutzung erfolgen?
- Welchen Zweck soll die Ontologie erfüllen?
- Wie soll die Ontologie repräsentiert werden?

- Wie sollen Definitionen repräsentiert werden? (Methoden zur Erschließung von Ontologien)

Durch dieses Vorgehen sollte dem Mangel an exakten Richtlinien, Methoden und Standards zur Entwicklung begegnet werden und ein qualitativ hochwertiger Ontologieerstellungsprozess gewährleistet werden. Der eigentliche Erstellungsprozess orientierte sich dabei an den von Uschold und Gruniger identifizierten Phasen einer Ontologieentwicklung [UG96]:

- Die Identifizierung der Ziele und Aufgabenbereiche
- Die Erstellung einer Ontologie
  - Erfassen der Ontologie
  - Kodieren der Ontologie
  - Integration bereits vorhandener Ontologien
- Evaluation
- Dokumentation
- Richtlinien für einzelne Phasen

Als Entwicklungswerkzeug zur Implementierung der Ontologie ERP-OnTo-PDM wurde der XML-Spy-Editor in der Version 5 Release 4 der Firma Altova verwendet. Hierbei handelt es sich um ein auf die Entwicklung von XML-basierten Dokumenten spezialisiertes Entwicklungswerkzeug. In Abbildung 7.1 ist dieses Entwicklungswerkzeug mit einem Auszug der ERP-OnTo-PDM Ontologie abgebildet.

### 7.1.2 Vorgehensmodell zur Realisierung der Schnittstelle

Neben der Ontologie als Basis für den semantischen Datenaustausch mittels ERP-OnTo-PDM mussten folgende zusätzliche Funktionen implementiert werden:

- Funktionen zur Erzeugung eines ERP-OnTO-PDM-Files.
- Funktionen zum Einlesen des ERP-OnTo-PDM-Files in den Anwendungssystemen.

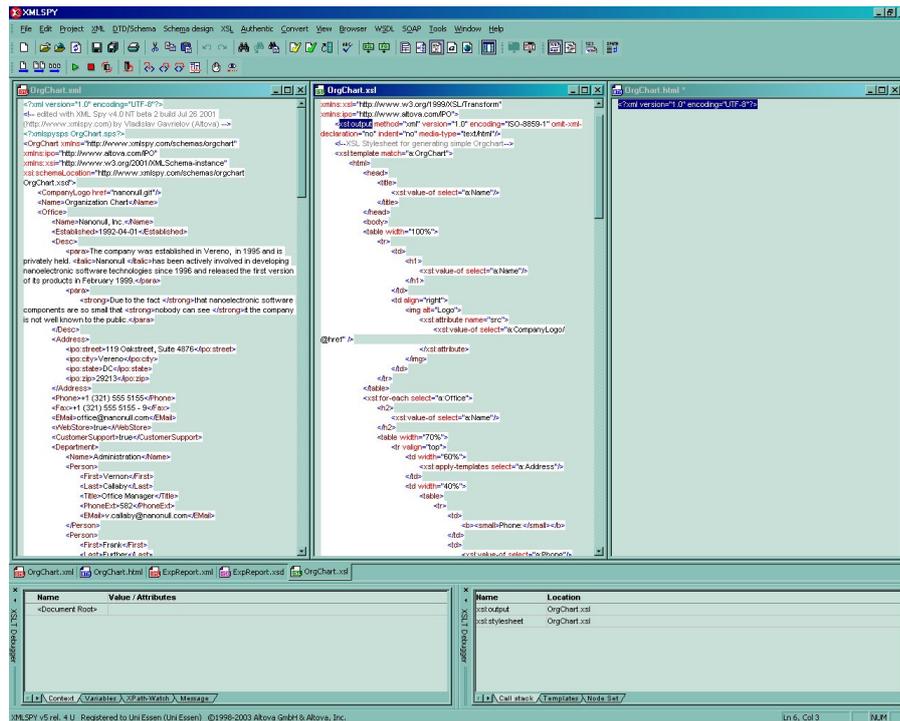


Abbildung 7.1: xml Spy 1

- Hilfsfunktionen zur Evaluierung der Arbeitsweise von ERP-OnTO-PDM.

Die Realisierung dieser Funktionen entspricht einer herkömmlichen Softwareentwicklung. Unter der Berücksichtigung, dass im Rahmen dieser Arbeit nur ein Prototyp erstellt werden kann, wurde ein Vorgehensmodell zur Erstellung von Softwareprototypen verwendet.

## 7.2 Anwendung von ERP-OnTo-PDM am Beispiel der Stückliste

Zur Demonstration der Funktionsweise von ERP-OnTo-PDM wurden nach dem oben beschriebenen Vorgehen die Ontologie sowie die Softwarebausteine entwickelt. Die Funktionsweise soll mittels des Datenaustausches von Stücklisten demonstriert werden. Eine solche Materialstückliste wird im PDM-System SmarTeam angelegt und mittels ERP-OnTO-PDM in das SAP R/3 System übertragen. Des Weiteren wird die universelle Weiterverarbeitbarkeit des ERP-OnTO-PDM-Files durch die Darstellung im Internet-Explorer demonstriert.

In Abbildung 7.2 ist der Einstiegsbildschirm des PDM-Systems SmarTeam dargestellt. Als letzter Eintrag in der Menüleiste ist das Menüfeld der semantischen Schnittstelle ERP-OnTo-PDM zu sehen. In diesem Menü befinden sich die zur Evaluierung der Arbeitsweise von ERP-OnTO-PDM notwendigen Programmfunktionen von SmarTeam.

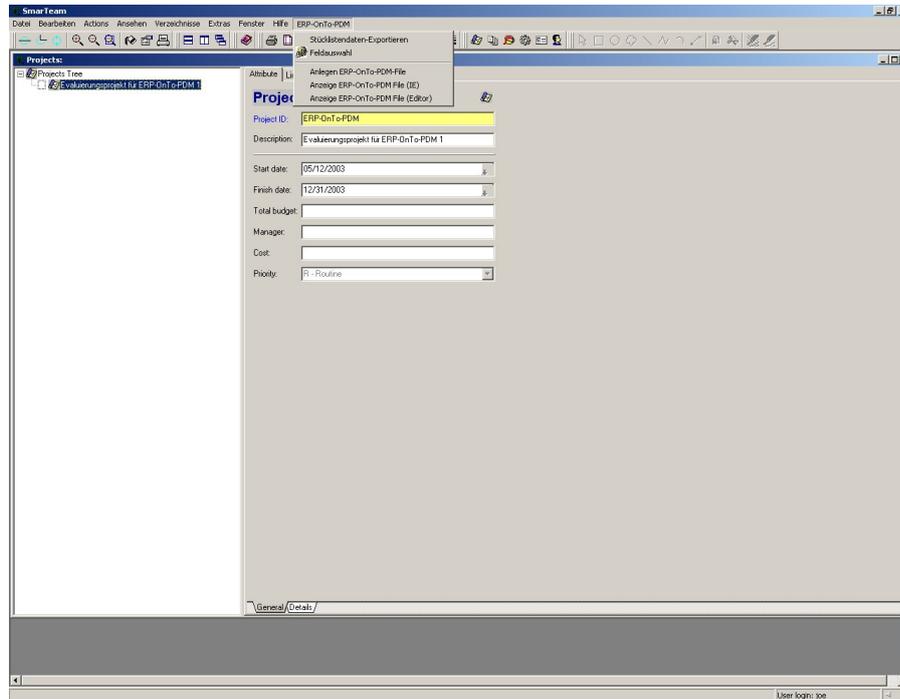


Abbildung 7.2: Einstiegsbildschirm ERP-OnTo-PDM

Dabei handelt es sich um nachstehende Funktionen:

- **Exportieren Stücklistendaten**

Dies ist die Hauptfunktion und führt den gesamten Workflow des Datenaustausches zwischen den Systemen durch.

- **Feldauswahl**

Mit Hilfe dieser Funktion wird es dem Benutzer ermöglicht, die gewünschten Datenfelder der Stückliste zu selektieren, die bei dem Datenaustausch berücksichtigt werden sollen. Der zu dieser Funktion gehörende Bildschirm ist in Abbildung 7.3 dargestellt.

- **Anlegen ERP-OnTO-PDM File**

Diese Funktion generiert das ERP-OnTo-PDM-File. Dieses File beinhaltet

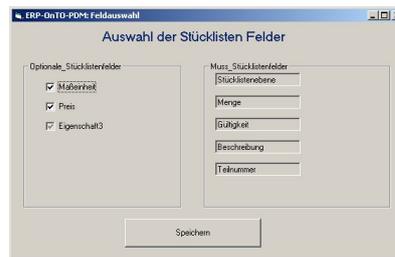


Abbildung 7.3: ERP-OnTo-PDM Funktion: Feldauswahl

die ausgewählten Daten aus der Feldauswahl und wandelt sie in das Format von ERP-OnTO-PDM um.

- **Anzeigen ERP-OnTo-PDM File (IE)**

Diese Funktion ruft das erzeugte ERP-OnTO-PDM File mit den selektierten Daten im Internet-Explorer der Firma Microsoft auf. Ab der Version 5.5 verfügt der Internet-Explorer über einen eingebauten XML-Parser, der eine Validierung des anzuzeigenden XML-Files vornimmt. Dadurch kann überprüft werden, ob das erzeugte ERP-OnTo-PDM File valide und well formed ist. Der zu dieser Funktion gehörende Bildschirm ist in Abbildung 7.4 dargestellt.

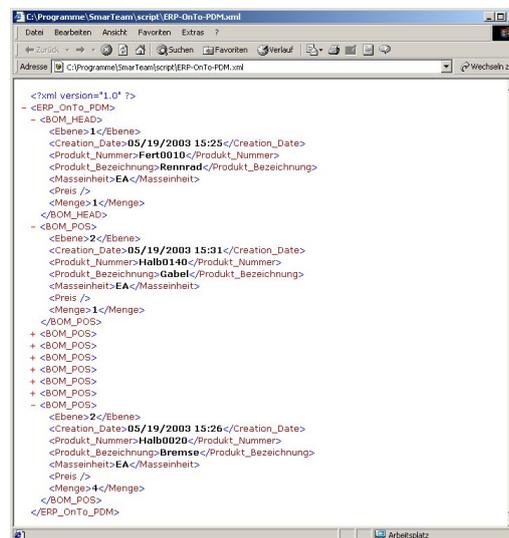


Abbildung 7.4: ERP-OnTo-PDM Funktion: Anzeige ERP-OnTo-PDM File (IE)

- **Anzeigen ERP-OnTo-PDM File (IE)**

Diese Funktion ermöglicht dem Benutzer, sich das erzeugte ERP-OnTo-PDM

File in einem Editor anzeigen zulassen.

Der letzte Schritt besteht im Hochladen der erzeugten ERP-OnTo-PDM-Datei in das SAP R/3 System. Hierbei hat der Benutzer die Möglichkeit entweder die Datei vom Präsentationsserver oder vom Anwendungsserver zu laden. Das zu dieser Funktion korrespondierende Bildschirmbild ist in Abbildung 7.5 dargestellt.



Abbildung 7.5: Hilfsfunktion Hochladen der ERP-OnTo-PDM-Datei im SAP-System

### 7.2.1 Stücklistendaten aus Sicht des SAP R/3 Systems

Im SAP-System werden Baukastenstücklisten verwaltet. Eine Stückliste besteht aus einem Stücklistenkopf und beliebig vielen Stücklistenpositionen. Jede Stücklistenposition kann in weitere Unterpositionen aufgeteilt sein. Stücklisten werden nach Stücklistenarten unterteilt. Dieser Ordnungsbegriff bestimmt u.a. das Verfahren der Nummernvergabe (Nummernkreis) und welche Stammdatenfelder gepflegt werden können. Die Daten im Stücklistenkopf beziehen sich auf das ganze Objekt, bei der Variantenstückliste auf die Stücklisten einer der Varianten, bei der Mehrfachstückliste auf eine Alternative der Stückliste eines Erzeugnisses. Beispiele für Daten im Stücklistenkopf sind Nummer des Materials, Werk, Revisionsstand, Stückliste, Alternative zur Identifikation innerhalb einer Stücklistengruppe, wie z.B. Varianten, technischer Typ, CAD-Kennzeichen etc.. Positionen beschreiben die Bestandteile eines Erzeugnisses. Beispiele für Positionsdaten sind die Positionsnummer, Komponente, Komponentenmenge, Positionstyp, Gültigkeitsdaten, Bau-

gruppe vorhanden, Unterposition vorhanden. Unterpositionen bilden verschiedene Einbauorte der Positionsmenge ab. Die Unterpositionen besitzen keine operative Bedeutung, sie werden z.B. nicht in den Fertigungsauftrag übernommen.

Stücklisten beschreiben in R/3 verschiedene Objekte wie Material, Equipment und Dokumente. Für die in einer Stückliste erfassten Objekte müssen im System gültige Stammsätze vorhanden sein.

Die wesentlichen Stücklistentypen, die im SAP-System verwendet werden, sind:

- **Materialstückliste**  
Die Materialstückliste bezeichnet eine Stückliste, die zu einem Material im Materialstammsatz angelegt wird.
- **Kundenauftragsstückliste.**  
Kundenauftragsstücklisten beziehen sich auf einen Kundenauftrag. Im Kundenauftrag wird - im R/3-Vertrieb (SD)- das konfigurierbare Material als Auftragsposition erfasst und mit einer abhängigen Auftragsstückliste gegliedert. Die Auftragsnummer und die Auftragsposition identifizieren die Kundenauftragsstückliste. Die mehrstufige interaktive Konfiguration sowie technisch bedingte Änderungen erfolgen in der Stücklistenverwaltung. Änderungen an der ursprünglichen Stückliste schlagen sich in der Auftragsstückliste nicht nieder.
- **Dokumentenstückliste.**  
Die Dokumentenstückliste fasst mehrere Dokumente bzw. ihre Informations- und Dokumentationsobjekte zu einem Dokumenteninfosatz zusammen. Ein Beispiel für ein komplexes Dokument: Konstruktionszeichnungen, Fotos, Schaltpläne, Spezifikationen, Ausschreibungsunterlagen, Programme für einen Kundenauftrag.
- **Equipmentstückliste.** Equipments sind technische Objekte der Instandhaltung (PM). Equipmentstücklisten werden benutzt, um die Struktur des Equipments zu beschreiben und um die Ersatzteile für die Zwecke der Instandhaltung zu definieren.
- **Technischer Platz Stückliste.**  
Der Technische Platz ist ebenfalls ein technisches Objekt der Instandhaltung (PM). Die Stücklisten zu technischen Plätzen definieren und fassen die Elemente einer technischen Struktur zusammen. Diese Stücklisten können Materialien und Dokumente beinhalten.

- Technische Stücklistentypen

Für die Abbildung von Produktvarianten und Fertigungsalternativen bietet das R/3-System folgende Typen:

- Variantenstücklisten.

Eine Variantenstückliste beschreibt mehrere ähnliche Produkte, deren Bestandteile nur geringe Unterscheidungen - Veränderungen der Grundausführung - aufweisen. Oder anders formuliert, einen hohen Anteil an identischen Bestandteilen besitzen. Auch unterschiedliche Komponentemengen können mit einer Variantenstückliste abgebildet werden. Variantenstücklisten können - bis auf die Kundenauftragsstückliste - für sämtliche Stücklistentypen angelegt werden.

- Mehrfachstückliste.

Die Mehrfachstückliste beschreibt ein Erzeugnis, das bei unterschiedlichen Fertigungsverfahren aus verschiedenen Komponenten bzw. unterschiedlichen Mengenverhältnissen hergestellt wird. Eine unterschiedliche Losgröße könnte der Grund für eine Mehrfachstückliste sein.

Für die Anwendung von ERP-OnTo-PDM werden Materialstücklisten verwendet.

Werk	Material	Al	Pos	Kompon.	Materialbezeichnung	FT	Menge	ME	ED	Datum
0001	PERST000	1	1		Reinrad MESSIG		1	2,000	PT	06.08.1996
10	BAL0000				Schulung MESSIG		1	1,000	PT	X
20	BAL0000				Schulung MESSIG		1	1,000	PT	X
30	BAL0000				Teillage MESSIG		1	1,000	PT	X
40	BAL0000				Lehrer MESSIG		1	1,000	PT	X
50	BAL0000				Lehrer MESSIG		1	1,000	PT	X
60	BAL0000				Lehrer (Büroal) MESSIG		1	1,000	PT	X
70	BAL0000				Lehrer (Heifen u. Feilpen) MESSIG		1	2,000	PT	X
80	BAL0000				Polster MESSIG		1	1,000	PT	X
90	BAL0000				Gehäl MESSIG		1	1,000	PT	X
0001	PERST000	1	1		Reinrad MESSIG		1	2,000	PT	06.08.1996
10	BAL0000				Schulung MESSIG		1	1,000	PT	X
20	BAL0000				Schulung MESSIG		1	1,000	PT	X
30	BAL0000				Teillage MESSIG		1	1,000	PT	X
40	BAL0000				Lehrer MESSIG		1	1,000	PT	X
50	BAL0000				Lehrer MESSIG		1	1,000	PT	X
60	BAL0000				Lehrer (Büroal) MESSIG		1	1,000	PT	X
70	BAL0000				Lehrer (Heifen u. Feilpen) MESSIG		1	2,000	PT	X
80	BAL0000				Polster MESSIG		1	1,000	PT	X
90	BAL0000				Gehäl MESSIG		1	1,000	PT	X
0001	PERST000	1	1		Teillage MESSIG		1	2,000	PT	06.08.1996
10	BAL0000				Schulung MESSIG		1	1,000	PT	X
20	BAL0000				Schulung MESSIG		1	1,000	PT	X
30	BAL0000				Teillage MESSIG		1	1,000	PT	X
40	BAL0000				Lehrer MESSIG		1	1,000	PT	X
50	BAL0000				Lehrer MESSIG		1	1,000	PT	X
60	BAL0000				Lehrer (Büroal) MESSIG		1	1,000	PT	X
70	BAL0000				Lehrer (Heifen u. Feilpen) MESSIG		1	2,000	PT	X
80	BAL0000				Polster MESSIG		1	1,000	PT	X
90	BAL0000				Gehäl MESSIG		1	1,000	PT	X
0001	PERST000	1	1		City-Ride MESSIG		1	1,000	PT	06.08.1996
10	BAL0000				Schulung MESSIG		1	1,000	PT	X
20	BAL0000				Schulung MESSIG		1	1,000	PT	X
30	BAL0000				Teillage MESSIG		1	1,000	PT	X
40	BAL0000				Lehrer MESSIG		1	1,000	PT	X
50	BAL0000				Lehrer MESSIG		1	1,000	PT	X
60	BAL0000				Lehrer (Büroal) MESSIG		1	1,000	PT	X
70	BAL0000				Lehrer (Heifen u. Feilpen) MESSIG		1	2,000	PT	X
80	BAL0000				Polster MESSIG		1	1,000	PT	X
90	BAL0000				Gehäl MESSIG		1	1,000	PT	X
0001	PERST000	1	1		Reinrad MESSIG		1	2,000	PT	11.09.1996
10	BAL0000				Schulung MESSIG		1	1,000	PT	X
20	BAL0000				Schulung MESSIG		1	1,000	PT	X
30	BAL0000				Lehrer MESSIG		1	1,000	PT	X
40	BAL0000				Lehrer MESSIG		1	1,000	PT	X
50	BAL0000				Lehrer (Büroal) MESSIG		1	1,000	PT	X
60	BAL0000				Lehrer (Heifen u. Feilpen) MESSIG		1	2,000	PT	X
70	BAL0000				Polster MESSIG		1	1,000	PT	X
80	BAL0000				Gehäl MESSIG		1	1,000	PT	X
90	BAL0000				Spezialtyp MESSIG		1	1,000	PT	X

Abbildung 7.6: Ergebnis der Hilfsfunktion:Stücklisten anzeigen

Eine der mittels der SAP R/3 eigenen Programmiersprache ABAP/4 realisierte Hilfsfunktion ermöglicht dem Benutzer von ERP-OnTo-PDM alle Informatio-

nen über den Aufbau der in diesem Mandanten verwendeten Stücklisten anzuzeigen. Diese Hilfsfunktion dient zur Evaluierung des Datenaustausches mittels ERP-OnTo-PDM.

Ein Ergebnis dieser Hilfsfunktion ist in Abbildung 7.6 dargestellt. Es enthält Informationen über die Bezeichnung der jeweiligen Stückliste, der Bezeichnungen der einzelnen Stücklistenpositionen und über die Quantität mit der diese in das übergeordnete Produkt eingehen.

## 7.2.2 Stücklistendaten aus Sicht von SmarTeam

SmarTeam bietet die Möglichkeit, Stücklisten über eine Vererbungsbeziehung zu definieren. Alle Stücklisteninformationen werden zu einer Komponente, im SmarTeam als *item* bezeichnet, angelegt. Zu dieser Komponente können die Subkomponenten angelegt werden, welche dann mittels einer Link-Beziehung als *Part-of* der Komponente definiert werden. Zu dieser Sub-Komponente kann die Quantität angegeben werden, mit der die Sub-Komponente in die Hauptkomponente eingeht. Eine Stückliste im Sinne von SmarTeam ist in Abbildung 7.7 dargestellt.

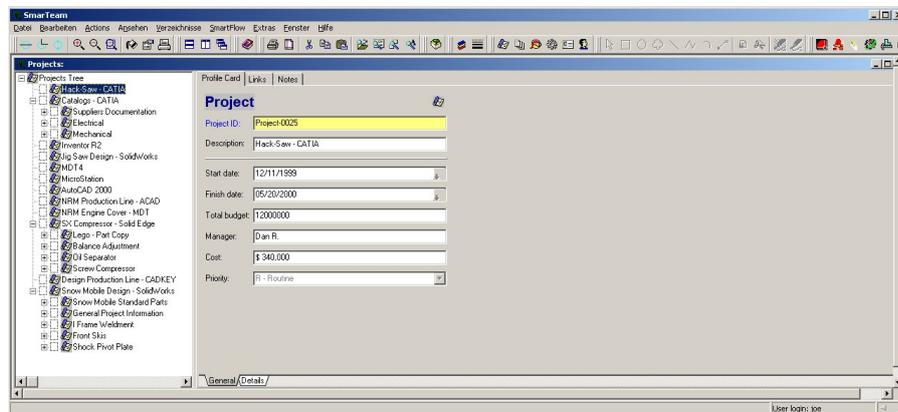


Abbildung 7.7: Einstieg SmarTeam2

## 7.3 Fazit

In diesem Kapitel wurde das Vorgehen bei der prototypenhaften Realisierung von ERP-OnTo-PDM beschrieben, sowie exemplarisch die Funktionsweise der semantikbasierten Kopplung von Informationssystemen mittels des im Rahmen dieser

Arbeit entwickelten Konzeptes ERP-OnTo-PDM evaluiert. Dazu wurde zunächst ein Überblick über die Funktionen von ERP-OnTo-PDM gegeben. Daran anschließend wurde das Konzept *Stückliste* jeweils in der Realisierung der beiden beteiligten Systeme (SmarTeam und SAP R/3) vorgestellt. Darauf aufbauend wurde der Prozess des Datenaustausches mittels ERP-OnTo-PDM für den Datentransfer einer Stückliste aus dem PDM-System SmarTeam in das ERP-System SAP R/3 beschrieben. Es wurde ersichtlich, dass mittels ERP-OnTo-PDM eine Transformation in ein produktunabhängiges, mit semantischen Informationen angereichertes Dateiformat möglich ist. Durch diese Transformation in ein mit der ERP-OnTo-PDM Ontologie konformes Format wurde eine Schnittstelle zwischen diesen Systemen geschaffen, die über einen reinen Datenaustausch weit hinaus geht. Es wurden durch die Transformation der Daten in die Ontologie folgende zwei Hauptvorteile erreicht:

#### 1. **Semantische Anreicherung**

Die Stücklistendaten des PDM-Systems wurden durch die Ontologie um Meta-Daten angereichert, die zu einer produktunabhängigen Repräsentation führen. Diese Daten können nun effektiv in verschiedene Systeme übernommen werden. Es wurde somit die Funktion eines Interlingua-Servers erreicht.

#### 2. **Medienunabhängigkeit**

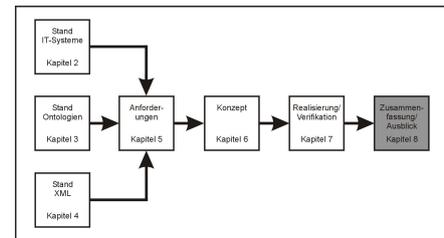
Durch die Transformation in das Format von ERP-OnTo-PDM wurde aufgrund ihrer Realisierung mittels XML-Technologie der Vorteil einer medienunabhängigen Repräsentation erreicht. Die Daten sind einmal gespeichert und können für die im Rahmen dieser Arbeit skizzierten neuen Anforderungen an die Produktdatenverarbeitung genutzt werden. Dies kann die Weitergabe an ein ERP-System sein oder die Einbettung der Produktinformationen innerhalb eines Internetauftrittes.

Mit dieser prototypenhaften Realisierung und der Evaluierung wurden nachgewiesen, dass mittels ERP-OnTo-PDM die Problemstellungen dieser Arbeit gut gelöst werden können.

# Kapitel 8

## Zusammenfassung und Ausblick

In diesem Kapitel werden die durch diese Arbeit gewonnenen Erkenntnisse zusammengefasst, und ein Ausblick auf mögliche weitere Forschungsarbeiten in dem Bereich des Produktdatenaustausches gegeben.



Im Kapitel 2 wurde der Stand der Forschung im Bereich der ERP- bzw. PDM-Systeme sowie Strategien zur Kopplung dieser Systeme aufgezeigt. Dazu wurde zunächst für den jeweiligen Bereich ein einheitliches Begriffsverständnis geschaffen und darauf aufbauend jeweils stellvertretend ein Produkt für die jeweilige Systemart vorgestellt. Die Betrachtung der jeweiligen Produkte erfolgte dabei in Hinblick auf deren Einsatz bei der Realisierung von ERP-OnTo-PDM in Kapitel 7. Des Weiteren wurde in diesem Kapitel bestehende Strategien zur Kopplung von IT-Systemen eingehender betrachtet und deren Schwachstellen aufgezeigt.

Das Kapitel 3 stellte den Stand der Forschung auf dem Gebiet der Ontologien vor. Zunächst wurde in diesem Kapitel das Konzept Ontologie vorgestellt, welches im Rahmen des Wissensmanagements eine immer größere Bedeutung erhält. Ontologien bilden den Kern des im Rahmen dieser Arbeit erarbeiteten ERP-OnTo-PDM-Konzeptes, einer ontologiebasierten auf XML-Technologie aufbauenden Implementierung zum semantischen Produktdaten-Austausch zwischen verschiedenen IT-Systemen, die produktrelevante Daten managen. Dazu wurden neben den definitorischen Grundlagen der Begriffswelt des Forschungsgebiet der Ontologien besonders deren Einsatzmöglichkeiten im Bereich des Datenaustausches von Produktdaten vorgestellt.

Der Stand der Forschung auf dem Gebiet der Internettechnik war Gegenstand des Kapitels 4. Hier wurde die Extensible Markup Language (XML) mit ihren assoziierten Standards dargestellt. Von besonderem Interesse zur Implementierung des Konzepts ERP-OnTo-PDM war das Resource Description Framework (RDF), welches die Definition von Metadaten ermöglicht, um den semantikbasierten Produktdatenaustausch zu ermöglichen. Neben den Grundlagen von XML und RDF wurde auch das Gebiet der XML-basierten Produktdatenaustauschformate behandelt und deren Schwachstellen aufgezeigt.

In Kapitel 5 wurden aus den in den drei zuvor behandelten Kapiteln aufgezeigten Schwachstellen die Anforderungen an eine semantische Kopplung von IT-Systemen mittels einer Ontologie abgeleitet und erläutert. Dazu wurden die Anforderungen an eine Kopplung von ERP- und PDM-Systemen in drei Bereiche untergliedert. Zunächst werden Anforderungen die sich aus der Unternehmensorganisation ergeben dargestellt. Daran anschließend wurden Anforderungen die sich durch den Einsatz der beiden Informationsverarbeitungssysteme ergeben aufgezeigt. Im dritten Anforderungsbereich sind technische Anforderungen zusammengefasst.

Das auf Grundlage der aufgezeigten Schwachstellen und den daraus abgeleiteten Anforderungen an eine semantische Kopplung von Informationsverarbeitungssystemen erarbeitete Konzept ERP-OnTo-PDM ist Gegenstand des Kapitels 6. Zunächst wurde in diesem Kapitel eine Auswahl der zur Realisierung einer ontologiebasierten Kopplung geeigneten Basistechnologie getroffen. Das Konzept wurde mittels eines RDF-Modells in diesem Kapitel veranschaulicht. Des Weiteren bildet RDF die Basistechnologie zur Realisierung des Konzepts ERP-OnTo-PDM welches in Kapitel 7 beschrieben ist. In diesem Kapitel wurde zunächst auf das Vorgehensmodell zur Erstellung der Ontologie für das Konzept ERP-OnTo-PDM eingegangen. Daran anschließend wurde die Anwendbarkeit durch eine prototypische Implementierung nachgewiesen.

Es hat sich gezeigt, dass mittels einer semantischen Kopplung von IT-Systemen die Anzahl an Schnittstellen drastisch reduzieren lässt. Des Weiteren ist ein erheblicher Vorteil von ERP-OnTo-PDM in der verwendeten Internettechnologie zu sehen. Durch den Einsatz von XML bzw. RDF wird eine durchgehende Datenverarbeitung auch über verschiedene Medien ermöglicht. Daraus kann den neuen Anforderungen an ein Produktdatenmanagement Rechnung getragen werden.

# Literaturverzeichnis

- [Abe98] ABELN, O. (Herausgeber): *Innovationspotentiale in der Produktentwicklung*. B.G. Teubner, 1998.
- [Abr99] ABRAMOVICI, M.: *EDM/PDM-Einführungsstrategien - Erfahrungen und Perspektiven*. In: VERTRIEB, VDI-GESELLSCHAFT ENTWICKLUNG KONSTRUKTION (Herausgeber): *Informationsverarbeitung in der Konstruktion '99 - Beschleunigung der Produktentwicklung Durch EDM/PDM- und Feature-Technologie: Tagung München 19./20- Oktober 1999*, Seiten 77–101, Düsseldorf, 1999. VDI Gesellschaft Entwicklung, Konstruktion, Vertrieb (VDI-EKV), VDI Verlag.
- [AH94] ABU-HANNA, A.: *Multiple Domain Models in Diagnostic Reasoning*. Doktorarbeit, University of Amsterdam, 1994.
- [Alb93] ALBERTS, L.: *YMIR: An Ontology for Engineering Design*. Doktorarbeit, University of Twente, 1993.
- [BA97] BORST, W. N. und J.M. AKKERMANS: *Engineering Ontologies*. International Journal of Human-Computer Studies, 46(2/3):365–406, 1997.
- [BEG95] BUCK-EMDEN, RÜDIGER und JÜRGEN GALIMOW: *Die Clinet/Server-Technologie Des Des SAP-Systems R/3: : Basis Für Betriebswirtschaftliche Standardanwendungen*. Addison-Wesley, Bonn Paris, 2. Auflage, 1995.
- [Beg00] BEGLEY, ED: *MatML: XML for Materials Property Data*. <http://>, 2000.
- [Bei98] BEITZ, W.: *Innovationsorientierte Effiziente Produktentwicklungsmethodik*. In: KRAUSE, F.-L. und E. UHLMANN (Herausgeber): *Innovative Produktionstechnik*, Seiten 255–262. Hanser Verlag, 1998.

- [BG99] BRICKLEY, D. und R. V. GUHA: *Resource Description Framework (RDF) Model and Syntax Specification*. W3C Recommendation, 22. Februar 1999. <http://www.w3c.org/TR/RFC-rdf-syntax>.
- [Bra98] BRAY, TIM: *The Annotated XML Specification*. 1998. <http://www.xml.xom/axml.html>.
- [Bru02] BRUGGER, JUDITH M.: *Cataloging the Internet*. <http://ublib.buffalo.edu/libraries/units/cts/Internet/brugger.html>, 2002.
- [Bür95] BÜRCEL, H. D.: *Lean R&D*. Stuttgart, 1995. Schäffer-Peoschl Verlag.
- [CDI94] CDI (Herausgeber): *SAP R/3: Grundlagen, Architektur, Anwendungen*. Markt und Technik, Haar bei München, 1994.
- [CFM98] CLEWETT, ANNETTE, DANA FRANKLIN und ANN MCCOWN: *Network Resource Planning Fpr SAP R/3, BAAN IV, and PeopleSoft. A Guide to Planning Enterprise Applications*. McGraw-Hill, New York San Francisco Washington. D.C., 1998.
- [Dim00] DIMITROV, MARIN: *XML Standards for Ontology Exchange*. 2000.
- [DIN87a] DIN, 1463 TEIL 1: *Erstellung und Weiterentwicklung Von Thesauri*. Beuth-Verlag, 1987.
- [DIN87b] DIN, 1463 TEIL 2: *(Entwurf) Anhang A - Beispiele Für Die Darstellung Von Thesaurus- Relationen*. Beuth-Verlag, 1987.
- [EHSS91] EIGNER, M., C. HILLER, S. SCHINDEWOLF und M. SCHMICH: *Engineering Database: Strategische Komponente in CIM-Konzepten*. Carl Hanser Verlag, München, Wien, 1991.
- [EKM99] EHRLER, A., H. KUNZE und M. MAISER: *Development of Configurable Semantix Mappers*. In: *Proceedings of the European Conference Product Data Technology Days 1999 in Stavanger*, Seiten 247–254, Norway, 1999.
- [EPM94] ERIKSSON, H., A. PUERTA und M. MUSEN: *Generation of Knowledge Aquisition Tools from Domain Ontologies*. In: *Proceedings of 10th Banff Knowledge Acquisition for Knowledgebased Systems Workshop*, Seiten 3–15, Canada, 1994. Banff.

- [Erp02] ERPFANS.COM: *Enterprise Resource Planning (ERP) Do We Need It?* Seiten 1–50, 2002. Also available online: <http://www.erpfans.com/erpfans/erpdefinition/erp008.html>.
- [ES99] ERDMANN, M. und R. STUDER: *Ontologies as Conceptual Models for XML Documents*. In: *Proceedings of the 12th Knowledge Acquisition for Knowledge-Based Systems Workshop (KWA '99) October 1999*, Canada, October 1999. Banff. <http://sern.ucalgary.ca/KSI/KWA/KWA99/papers.html> or <http://www.aifb.uni-karlsruhe.de/mer/projects.html>.
- [Eve96] EVERSHEIM, WALTER (HRSG.): *Prozeßorientierte Unternehmensorganisation - Konzepte und Methoden Zur Gestaltung Schlanker Organisationen*. Springer, Berlin, Heidelberg, New York, 1996.
- [Fär81] FÄRBER, K.: *Auftragsabwicklung in Einer Serienfertigung mit Kundenindividuellen Varianten*, Band VDI-Bericht Band 412. VDI-Verlag, Düsseldorf, 1981.
- [Fen97] FENSEL, D.: *Specifying Knowledge-Based Systems with Reusable Components*. n.n, Berlin New York, 1997.
- [Fen01] FENSEL, D.: *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer Verlag, 2001.
- [FM02a] FELDMAN, SUSAN und JESSICA MILSTEAD: *Cataloging by Any Other Name...* <http://www.onlinemag.net/OL1999/milstead1.html>, Abruf am 12.12.2002.
- [FM02b] FELDMAN, SUSAN und JESSICA MILSTEAD: *Metadata Projects and Standards*. <http://www.onlinemag.net/OL1999/milstead1.html>, Abruf am 12.12.2002.
- [FNH97] FRIDMAN-NOY, N. und C. D. HAFNER: *The State of the Art in Ontology Design*. AI Magazine, 18(3):53–74, 1997.
- [Ger00] GERHARD, DETLEF: *Erweiterung der PDM-Technologie Zur Unterstützung Verteilter Kooperativer Produktentwicklungsprozesse*. Shaker Verlag, Aachen, 2000.
- [GHS96] GAUSEMEIER, J., A. HANH und W. SCHNEIDER: *Kooperatives Modellieren Auf Basis Transienter Objekte*. In: RULAND, D. (Herausgeber):

- Verteilte und Intelligente CAD-Systeme, Tagungsband CAD '96 Kaiserslautern 07.-08.03.1996*, Seiten 311–325. Informatik Xpress, 1996.
- [GLR92] GRABOWSKI, H., G. LANGLOTZ und S. RUDE: *25 Jahre CAD in Deutschland - Standortbestimmung und Notwendige Entwicklung*. VDI-Berichte, 993:1–30, 1992.
- [GLW02] GRABOWSKI, HANS, RALF LOSSACK und JÖRG WEISSKOPF: *Datenmanagement in der Produktentwicklung. Automatische Klassifikation Von Produktdaten Aus 3D-CAD-Systemen, PDM- und ERP-Systemen, XML- und Office-Dokumenten...* Carl Hanser Verlag, München Wien, 2002.
- [GP96] GOMEZ-PEREZ, A.: *Knowledge Sharing and Reuse*. 1996.
- [Gra90] GRABOWSKI, H.: *Bedeutung der Normung Von Produktmodelledaten in CIM*. In: *Rechnerintegrierte Konstruktion und Produktion - Eine Organisatorische, Personelle und Technische Herausforderung Kongress München 22.-24. Oktober 1990*, Seiten 75–106, Düsseldorf, 1990. VDI Gemeinschaftsausschuss CIM, VDI-Verlag.
- [Gro97] GROENBOOM, R.: *Formalizing Knowledge Domains - Static and Dynamic Aspects*. Shaker Publisher, 1997.
- [Gru93] GRUBER, T. R.: *A Translation Approach to Portable Ontologies*. Knowledge Acquisition, 5(2):199–220, 1993.
- [Gru95] GRUBER, T. R.: *Towards Principles for the Design of Ontologies Used for Knowledge Sharing*. International Journal of Human-Computer Studies, 43:907–928, 1995.
- [GTRM94] GENNARI, J., S. TU, T. ROTHENFLUTH und M. MUSEN: *Mapping Domains to Methods in Support of Reuse*. International Journal of Human and Computer Studies, 41:399–424, 1994.
- [Gua95] GUARINO, N.: *Formal Ontology: Conceptual Analysis and Knowledge Representation*. International Journal of Human-Computer Studies, 43(2/3):625–640, 1995.
- [Gua96] GUARINO, N.: *Understanding, Building, and Using Ontologies*. In: *Proceedings of 10th Banff Knowledge Acquisition for Knowledgebased Systems Workshop*, Seiten 3–15, Canada, 1996. Banff.

- [Gua98] GUARINO, NICOLA: *Formal Ontology and Information Systems*. In: *Proceedings of FOIS '98, Trento, Italy, 6-8 June 1998*, Seiten 3–15. IOS Press, 1998.
- [Gut79] GUTENBERG, E.: *Grundlagen der Betriebswirtschaftslehre*, Band 1. Springer-Verlag, Berlin, 1979.
- [Har73] HARRINGTON, J.: *Computer Integrated Manufacturing*. Industrial Press, New York, 1973.
- [Heb00] HEBER, JÜRGEN: *Knowledge Discovery-Broker-Entwicklungsarbeiten Für Das xFIND-Suchsystem*. Diplomarbeit, iicm, 2000. <http://www.iicm.edu/cguetl/education/thesis/jheber/html/diplom.html>.
- [Ing78] INGENIEURE, VEREIN DEUTSCHER: *Elektronische Datenverarbeitung Bei der Produktionsplanung und -Steuerung*. VDI-Verlag, Düsseldorf, 1. Auflage Auflage, 1978.
- [Inn98] INNELLA, RENATO: *An Idiots Guide to the Resource Description Framework*. The New Review of Information Networking, 4, 1998. Also available at <http://www.dstc.edu.au/Research/Projects/rdf/RDF-Idiot.html>.
- [Kah00] KAHLERT, TORALF: *Konzeption Eines Webbasierten Beratungs-Unterstützungs-Systems Am Fallbeispiel Einer PDM-Systemauswahl*. Fraunhofer-Institut für Produktionsanlagen und Konstruktionstechnik, IPK Berlin, Berlin, 2000.
- [KBS94] KNOELEDGE BASE SYSTEMS, INC: *IDEF 5 Methode Report*. Technischer Bericht F33615-C-90-0012, Knowledge Base Systems, Inc. 1408 University Drive East Collge Station, Texas 77849 (409) 260-5274, 1994.
- [Kla99] KLAHN, MARKUS: *PPS und CAD Online Koppeln: Echte Integration Bringt Informations- und Zeitgewinn*. EDM-Report, 5(4):22–27, 1999.
- [KR98] KEMPIS, R.-D. und J. RINGBECK: *Do It Smart*. Ueberreuter Wirtschaftsverlag, Wien, 1998.
- [KS93] KISTNER, KLAUS-PETER und MARION STEVEN: *Produktionsplanung*. Physica-Verlag, Heidelberg, 2. Auflage, 1993.

- [Ler99] LERMER, J.: *Erfolgreiche EDM/PDM-Systemeinführung Durch Gestufte Vorgehensweise*. In: VERTRIEB, VDI-GESELLSCHAFT ENTWICKLUNG KONSTRUKTION (Herausgeber): *Informationsverarbeitung in der Konstruktion '99 - Beschleunigung der Produktentwicklung Durch EDM/PDM- und Feature-Technologie: Tagung München 19./20- Oktober 1999*, Seiten 311–329, Düsseldorf, 1999. VDI Gesellschaft Entwicklung, Konstruktion, Vertrieb (VDI-EKV), VDI Verlag.
- [Med00] MEDEIROS, NORM: *XML and the Resource Description Framework: The Great Web Hope*. Online Magazine, 24(5), 2000. Also available at <http://www.onlinec.com/onlinemag/OL2000/medeiros9.html>.
- [Mee94] MEERKAMP, H.: *Features Als Grundvoraussetzung Für Integrierte Konstruktionssysteme*. In: *Workshop on Feature Technology in Design and Manufacturing*. Kolloquienzentrum Spelzenklamm 29.- 30. August. 1994, Universität des Saarlandes, 1994.
- [Mew73] MEWES, D.: *Der Informationsbedarf im Konstruktiven Maschinenbau*. VDI-Verlag, Düsseldorf, 1973.
- [Mic99] MICHEL, THOMAS: *XML Kompakt: Eine Praktische Einführung*. Carl Hanser Verlag, München Wien, 1999.
- [MK89] MARTINY, L. und M. KLOTZ: *Strategisches Informationsmanagement - Bedeutung und Organisatorische Umsetzung*, Band 12.1. Oldenburg-Verlag, München Wien, 1989.
- [MK98] MÖHRLLEN, REGINE und FRIEDRICH KOKOT: *SAP R/3 - Kompendium: Betriebswirtschaftliche Funktionsumfang und Erfolgspotentiale; Technologiegrundlagen, Internet-Funktionalität, Warehouse-Konzept U.A.; EURO-Special, Strategisches Projektmanagement, Branchenlösungen...* Markt und Technik Buch- und Software-Verl., Haar bei München, 1998.
- [MSS01] MÄDCHE, ALEXANDER, STEFFEN STAAB und RUDI STUDER: *Ontologien*. *Wirtschaftsinformatik*, 4(43):393–395, 2001.
- [N.N86] N.N: *Information Processing – Text and Office Systems – Standard Generalized Markup Language (SGML)*. International Organization for Standardization (ISO), 1986. <http://www.iso.ch/iso/en/ISOOnline.frontpage>.

- [N.N98] N.N.: *Product Data Management: The Definition*. CIMdata Inc., 1998. <http://www.cimdata.com>.
- [N.N00] N.N.: *Product Data Management Enablers Specification Version 1.3 November 2000*. OMG Manufacturing Domain Task Force(MfgDTF), 2000. <ftp://ftp.omg.org/pub/docs/formal/00-11-11.pdf>.
- [N.N02a] N.N.: *Glossar Von Dublin Core Standard*. <http://dublincore.org/documents/2001/04/12/usageguide/glossary.shtml>, 2002.
- [N.N02b] N.N.: *Defining Metadata*. <http://www.getty.edu/research/institute/standards/intro> Abruf am 11.12.2002.
- [OSS01] OPPERMAN, HENRICK, HANS-PETER SCHNURR und RUDI STUDER: *Die Bedeutung Von Ontologien Für Das Wissensmanagement*. Wissensmanagement, (6):33–36, 2001.
- [Par99] PARDI, WILLIAM J.: *XML in Action - Dynamische und Datengestützte Webseiten mit der Neuesten Web-Technologie*. Microsoft Press Deutschland, Unterschleißheim, 1999.
- [PB99] PEREZ, A. G. und V. R. BENJAMINS: *Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods*. In: *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends*, August 1999. Also available at <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-18/>.
- [PS] PIRLEIN, T.H. und R. STUDER: *Integrating the Reuse of Commonsense Ontologies and Problem-Solving Methods*. International Journal of Expert systems: Research and Applications, Seite in pres.
- [RK98] RUNGGALDIER, EDMUND und CHRISTIAN KANZIAN: *Grundprobleme der Analytischen Ontologie*. Schöningh, München Wien Zürich, 1998.
- [RR01] ROTHFUSS, GUNTHER und CHRISTIAN RIED: *Content Management. Mit XML: Grundlagen und Anwendungen*. Springer, Berlin; Heidelberg; New York; Barcelona; Hongkong; London; Mailand; Paris; Singapur; Tokio;, 2001.

- [Sch88] SCHOLZ, BERND: *CIM-Schnittstellen: Konzepte, Standards und Probleme der Verknüpfung Von Systemkomponenten in der Rechnerintegrierten Produktion*. Oldenbourg, München and Wien, 1988.
- [Sch89] SCHEER, A.-W.: *CIM Computer Integrated Manufacturing. Der Computergesteuerte Industriebetrieb*. Springer-Verlag, Berlin Heidelberg New York, 1989.
- [SEG<sup>+</sup>96] STUDER, R., H. ERIKSSON, J. H. GENNARI, S. W. TU, D. FENSEL und M. MUSEN: *Ontologies and the Configuration of Problem-Solving Methods*. In: *Proceedings of the 10th Banff Knowledge Aquisition for Knowledge-Based Systems Workshop*, Canada, 1996. Bsnff.
- [Sig01] SIGEL, ALEXANDER: *Wissensorganisation; Lehreinheit im Rahmen der Postgradualen Studiengänge „Wissensmanagement“ und „Bibliotheks- und Informationsmanagement“*. Donau-Universität Krems:<http://index.bonn.iz-soz.de/sigel/veroeff/Krems-2001/PerlPoint/Wissensorga.pdf>, 2001.
- [SK97] SPUR, G. und F.-L. KRAUSE: *Das Virtuelle Produkt - Management der CAD-Technik*. Carl Hanser Verlag, München Wie, 1997.
- [Sta89] STAHLKNECHT, PETER: *Einführung in Die Wirtschaftsinformatik*. Springer Verlag, Berlin, Heidelberg, New York, Tokyo, 4. Auflage, 1989.
- [TEG<sup>+</sup>95] TU, S., H. ERIKSSON, J. GENNARI, Y. GENNARI und M. MUSEN: *Ontology-Based Configuration of Problem-Solving-Methods and Generation of Knowledge Acquisition Tools*. Artificial Intelligence in Medicine, 1995.
- [Tei] TEIL1, ISO 9000: *Normen Zum Qualitätsmanagement und Zur Darlegung Des Qualitätsmanagementssystems - Teil1: Leitfaden Zur Auswahl und Anwendung*. International Organization for Standardization, Geneva, Switzerland.
- [uAVE95] ANLAGENBAU (VDMA) E.V., VERBAND DEUTSCHER MASCHINEN UND (Herausgeber): *Produkt Daten Management - Grundlagen und Entscheidungshilfe*. Eggebrecht-Press KG, Mainz, 1995.

- [UG96] USCHOLD, M. und M. GRUNINGER: *ONTOLOGIES: Principles, Methods and Applications*. Knowledge Engineering Review, Volume 11(Nummer 2), 1996.
- [Vaj92] VAJNA, S.: *Gruppentechnologie und CIM*. CIM-Management, 8:6, 1992.
- [Vaj99] VAJNA, S.: *Die Neue Richtlinie VDI 2219: Praxiserprobte Hinweise Zu Einführungsstrategien und Wirtschaftlichkeit Von EDM/PDM Systemen*. In: VERTRIEB, VDI-GESELLSCHAFT ENTWICKLUNG KONSTRUKTION (Herausgeber): *Informationsverarbeitung in der Konstruktion '99 - Beschleunigung der Produktentwicklung Durch EDM/PDM- und Feature-Technologie: Tagung München 19./20- Oktober 1999*, Seiten 25–42, Düsseldorf, 1999. VDI Gesellschaft Entwicklung, Konstruktion, Vertrieb (VDI-EKV), VDI Verlag.
- [VDI99] ??? In: VERTRIEB, VDI-GESELLSCHAFT ENTWICKLUNG KONSTRUKTION (Herausgeber): *Informationsverarbeitung in der Konstruktion '99 - Beschleunigung der Produktentwicklung Durch EDM/PDM- und Feature-Technologie: Tagung München 19./20- Oktober 1999*, Seiten 77–101, Düsseldorf, 1999. VDI Gesellschaft Entwicklung, Konstruktion, Vertrieb (VDI-EKV), VDI Verlag.
- [VR87] VDI-RICHTLINIE, 2235: *Wirtschaftliche Entscheidungen Beim Konstruieren - Methoden und Hilfen*. VDI-Verlag, Düsseldorf, 1987.
- [VR93] VDI-RICHTLINIE, 2221: *Methodik Zum Entwickeln und Konstruieren Technischer Systeme und Produkte*. VDI-Verlag, Düsseldorf, 1993.
- [VSW97] VAN, G. HEIJST, A. SCHREIBER und B. WIELINGA: *Using Explicit Ontologies for KBS Development*. International Journal of Human-Computer Studies, 46(2/3):183–292, 1997.
- [W3C99] W3C: *Resource Description Framework (RDF) Model and Syntax Specification W3C Recommendation 22 Februar 1999*. 1999. Also available online : <http://www.w3c.org/TR/1999/REC-rdf-syntax-19990222>.
- [W3C00a] W3C: *Extensible Markup Language (XML) 1.0 (Seconed Edition)*. W3C Recommendation 6 October 2000. 2000. Also available online : <http://www.w3c.org/TR/2000/REC-xml-20001006>.

- [W3C00b] W3C: *Resource Description Framework (RDF) Schema Specification 1.0 W3C Candidate Recommendation 27 March 2000*. 2000. Also available online : <http://www.w3c.org/TR/2000/CR-rdf-schema-20000327>.
- [W3C00c] W3C: *XHTML 1.0: The Extensible Hyper Text Markup Language. A Reformation of HTML 4 in XML 1.0*. 2000. Also available online : <http://www.w3c.org/TR/2000/REC-xhtml-20000126>.
- [Wes90] WESTKÄMPER, E.: *Auftragsentwicklung - Effizienz Vom Auftragseingang Bis Zum Versand*. In: *Rechnerintegrierte Konstruktion und Produktion - Eine Organisatorische, Personelle und Technische Herausforderung Kongress München 22.-24. Oktober 1990*, Seiten 135–164, Düsseldorf, 1990. VDI Gemeinschaftsausschuss CIM, VDI-Verlag.
- [WS93] WIELINGA, B. und A. SCHREIBER: *Reusable and Sharable Knowledge Bases: A European Perspective*. In: *Proceedings of the 1st International Conference on Building and Sharing Very Large-Scaled Knowledge Bases*, Tokyo Japan, 1993.
- [WSJ<sup>+</sup>94] WIELINGA, B. J., A. T. SCHREIBER, W. JANSWEIJER, A. ANJEWIERTEN und F. VAN HAMELEN: *Framework and Formalism for Expressing Ontologies*. Esprit Project 8145 KACTUS, 1994.
- [ZSS99] ZELEWSKI, STAPHAN, REINHARD SCHÜTTE und JUKA SIEDENTOPF: *Ontologien Zur Strukturierung Von Domänenwissen Tagung Des Workshop „Wissen - Wissenschaftstheorie - Wissensmanagement“*. 1999. Also available online: <http://pim.uni-essen.de/vortraege/Berlin.pdf>.