

**Universität Leipzig**  
Fakultät für Mathematik und Informatik  
Institut für Informatik

**Beschreibung von Verfahren zur Anbindung von E-Commerce-  
Anwendungen an betriebswirtschaftliche Standardsoftware und  
Evaluation durch eine beispielhafte Implementierung**

**Diplomarbeit**

Ausführung bei: Prof. Dr. Ing. habil. Dipl. Math. Klaus Peter Fähnrich  
Lehrstuhl für Anwendungsspezifische Informationssysteme  
Universität Leipzig

Betreuer: Dipl.-Wirtsch.-Inf. Maik Thränert  
Dipl.-Inf. Frank Werle

Diplomand: Michael Walter  
Matr.-Nr.: 7502383

Abgabe: Leipzig, April 2002

## **Kurzzusammenfassung**

Diese Arbeit gibt einen Überblick über das Spektrum der Enterprise Application Integration. Dazu werden die Grundlagen des E-Business erläutert, um die Notwendigkeit der Integration aufzuzeigen. Eine Aufstellung der verschiedenen Datenformate gibt Aufschluss über die Entwicklungen und die verschiedenen Standards.

In dem Abschnitt Middleware werden die technischen Grundlagen einer möglichen Integration erläutert. Besonders wird dabei auf JMS eingegangen, das in der praktischen Anwendung genutzt wurde.

Die Frameworks bilden ein umfassendes Mittel zur Integration von Applikationen, deswegen werden sie abschließend vorgestellt. Dabei wird J2EE und das neue von Microsoft propagierte .NET präsentiert.

Die Problematik bei der Implementierung wird anhand des Praxis Beispiels herausgearbeitet. Als Verbindung diente SonicMQ, der JMS Broker von Progress.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung .....</b>	<b>1</b>
1.1	Relevanz des Themas .....	1
1.2	Aufbau der Arbeit .....	1
<b>2</b>	<b>Grundlegende Betrachtungen .....</b>	<b>3</b>
2.1	E-Business .....	3
2.1.1	E-Business Klassifikation .....	3
2.1.2	E-Commerce-Modelle .....	4
2.2	Online Shop .....	5
2.2.1	Funktionen eines Online Shops .....	5
2.2.2	Technische Ausprägungen von E-Commerce-Systemen .....	6
2.2.3	Anbieter .....	6
2.3	ERP-Systeme .....	7
2.3.1	Merkmale eines ERP-Systems .....	7
2.3.2	Architektur eines ERP-Systems .....	8
2.3.3	ERP-Systemanbieter .....	9
<b>3</b>	<b>Integration von E-Business in den Geschäftsablauf .....</b>	<b>12</b>
3.1	Grundgedanken für die Integration .....	12
3.1.1	Geschichte der Integration .....	12
3.1.2	Managementkonzepte .....	13
3.1.3	Marktveränderungen .....	14
3.2	Integrationsstrategie .....	14
3.2.1	Integrationsrichtung .....	15
3.2.2	Integrationsstiefe .....	16
<b>4</b>	<b>Enterprise Application Integration .....</b>	<b>17</b>
4.1	Grundlagen .....	17
4.1.1	Begriffsdefinitionen .....	17
4.1.2	Integrationsarten .....	19
4.1.3	Entwicklungstrends .....	22
4.2	Basistechnologien .....	25
4.2.1	XML .....	25
4.2.2	Technische Kernstandards von XML .....	31
4.2.3	Horizontale Standards .....	34
4.2.4	Horizontale E-Business Standards .....	37
4.3	Message-orientierte Middleware .....	49
4.3.1	Messaging .....	49
4.3.2	JMS .....	51
4.3.3	JMS Produkte .....	53
4.4	Frameworks .....	54
4.4.1	Webservices .....	55
4.4.2	J2EE .....	56
4.4.3	Microsoft .NET .....	57
4.4.4	Zukünftige Entwicklungen .....	58
<b>5</b>	<b>Exemplarische Darstellung einer Integration .....</b>	<b>59</b>
5.1	Problemstellung .....	59

5.1.1	Ausgangssituation.....	59
5.1.2	Aufgabenstellung.....	61
5.2	Lösungsansätze und Implementierungsmöglichkeiten.....	63
5.3	Beschreibung der gewählten Lösung.....	65
5.4	Zusatzimplementierung: Integrationsmodul BMEcat .....	69
<b>6</b>	<b>Zusammenfassung .....</b>	<b>71</b>

## Abbildungsverzeichnis

Abbildung 2-1: Elemente des Enterprise Resource Management.....	8
Abbildung 2-2: SAP R3 Komponenten.....	10
Abbildung 3-1: Integrationsstufen.....	13
Abbildung 3-2: Integrierte Informationssysteme.....	15
Abbildung 4-1: Integrationsarten.....	20
Abbildung 4-2: Arten von Enterprise Application Integration.....	20
Abbildung 4-3: Arten von B2B Application Integration.....	23
Abbildung 4-4: Varianten zur B2B-Integration.....	23
Abbildung 4-5: Konflikt bei der Verwendung von Reusables.....	29
Abbildung 4-6: XML-Kernstandards und Vokabulare.....	30
Abbildung 4-7: SOAP Message.....	35
Abbildung 4-8: Klassenstruktur von ETIM.....	39
Abbildung 4-9: Klassenstruktur von eCl@ss.....	40
Abbildung 4-10: Klassenstruktur von UN/SPSC.....	41
Abbildung 4-11: Ebenenmodell der eCo-Architektur.....	47
Abbildung 4-12: Message-orientierte Middleware.....	50
Abbildung 4-13: Zentralisierte Architektur – Hub-and-Spoke.....	50
Abbildung 4-14: Dezentralisierte Architektur – IP Multicast.....	51
Abbildung 4-15: JMS Messaging Domains.....	52
Abbildung 4-16: Garantierte Nachrichtenzustellung.....	53
Abbildung 5-1: Aufbau des omeco <sup>®</sup> webshop.....	59
Abbildung 5-2: Standardmodule und Integrationsmodule des omeco <sup>®</sup> webshop.....	60
Abbildung 5-3: Warenkorb.....	61
Abbildung 5-4: Verbindung SonicMQ und Progress DB.....	63
Abbildung 5-5: Integrationsmöglichkeiten von JMS in omeco <sup>®</sup> webshop.....	64
Abbildung 5-6: Verbindung proALPHA Webserver und proALPHA ERP-System.....	66
Abbildung 5-7: Message-Verarbeitung im proALPHA ERP SYSTEM.....	66
Abbildung 5-8: Message-Verarbeitung im omeco <sup>®</sup> webshop.....	67
Abbildung 5-9: Integration der Echtzeitabfrage in den omeco <sup>®</sup> webshop.....	68
Abbildung 5-10: XML-Dokumente für die Preisfindung.....	68
Abbildung 5-11: Netzstruktur.....	69

## **Tabellenverzeichnis**

Tabelle 2-1: Bereiche des E-Commerce .....	4
Tabelle 2-2: ERP-Systemanbieter .....	9
Tabelle 4-1: Integrationsmatrix .....	22
Tabelle 4-2: Bewertung von Klassifikationsstandards .....	42

# **1 Einleitung**

In den letzten Jahren haben sich in der Geschäftswelt immer wieder neue Trends entwickelt, die auf der Nutzung von Informations- und Kommunikationstechnologie beruhen.

## **1.1 Relevanz des Themas**

Durch den härter werdenden Wettbewerb der Unternehmen wird die Frage nach Optimierungsmöglichkeiten gerade im Bereich der Geschäftsabläufe immer häufiger gestellt. Schlagwörter wie Supply Chain Management (SCM), Customer Relationship Management (CRM), E-Procurement sind aus der Fachliteratur schon fast nicht mehr weg zu denken. Grundgedanke hierbei ist die Integration von Software-Anwendungen, um einen möglichst reibungslosen Ablauf von Geschäftsprozessen zu gewährleisten und den Kunden den besten Service zu bieten.

Dazu zählen die Echtzeitverbindung, die Preisfindung und die Verfügbarkeitsabfrage, die im praktischen Teil behandelt werden. Um diese Funktionen verwirklichen zu können, muss das Frontend-System mit dem Backend verbunden sein.

## **1.2 Aufbau der Arbeit**

In dieser Arbeit sollen zuerst die Grundlagen zu E-Business und den verwendeten Systemen erarbeitet werden. In Kapitel 2.1 erfolgt dabei eine Klassifizierung der verschiedenen Bereiche des E-Business. Die verschiedenen Anwendungen, die sich durch den technischen Fortschritt entwickelt haben, folgen im nächsten Abschnitt. Um das Verständnis für die Integration zu erleichtern, werden dann die zu integrierenden Systeme beschrieben.

Die Gründe für die Integration sollen in Kapitel 3 beschrieben werden. So spielen in jeder Vernetzung von Systemen betriebswirtschaftliche Ideen eine Rolle, die durch die neuen Managementsysteme entstanden sind.

Kapitel 4 spiegelt die technische Realisierung der Integration wieder. Dabei werden in Kapitel 4.1 die Grundlagen gelegt. Die Basistechnologien werden darauffolgend

## 1 Einleitung

---

klassifiziert. Als weiterer Punkt wird Middleware erläutert und zum Abschluss werden verschiedene Frameworks vorgestellt.

Die Umsetzung der in der Theorie gewonnenen Erkenntnisse wird in Kapitel 5 verdeutlicht. Dabei wird zuerst die Ausgangssituation geschildert und dann die gewählte Lösung. Die Problematik, die sich bei der Implementierung stellte, wird in Kapitel 5.3 behandelt.

Ein Überblick über die gewonnenen Erkenntnisse und ein Ausblick in die Zukunft wird in Kapitel 6 gegeben.

## 2 Grundlegende Betrachtungen

In diesem Kapitel werden die Grundlagen gebildet, um eine einheitliche Begriffsnutzung zu gewährleisten. Außerdem werden die Modelle des E-Business erläutert. E-Commerce- und ERP-System werden zum Schluss vorgestellt.

### 2.1 E-Business

Kein anderer Begriff hat die Wirtschaft in den letzten Jahren so geprägt wie E-Business. Trotz allem hat sich in der Literatur keine einheitliche Definition durchgesetzt. Für diese Arbeit wird der Begriff E-Business als

**Nutzung informationstechnischer Medien zur Abwicklung von Geschäftsprozessen**

verstanden. E-Commerce wird als Teilbereich des E-Business identifiziert und wie folgt definiert:

**Nutzung informationstechnischer Medien zur Abwicklung von elektronischem Handel.**

Durch die weitgefaste Definition von E-Business wird nun eine Möglichkeit zur Einteilung gegeben.

#### 2.1.1 E-Business Klassifikation

In der Literatur wird eine Einteilung nach Anbieter und Nachfrager einer Leistung vollzogen, um die Komplexität des E-Business zu reduzieren. Dabei können folgende Bereiche angegeben werden:

- Endkonsumenten (Consumer)
- Unternehmen (Business)
- Öffentliche Verwaltung (Administration)
- Unternehmen als Wiederverkäufer (Reseller)
- Mitarbeiter (Employee).

## 2 Grundlegende Betrachtungen

---

Wenn dies nun als Matrix aufgespannt wird, ergeben sich die verschiedenen Sektoren des E-Business. Die Reseller und die Employee werden nicht berücksichtigt, da sie dem Business bzw. dem Consumer zugerechnet werden können.

<b>Nachfrager Anbieter</b>	<b>Consumer</b>	<b>Business</b>	<b>Administration</b>
<b>Consumer</b>	Consumer-to-Consumer <i>www.ricardo.de</i> <i>www.ebay.de</i>	Consumer-to-Business <i>www.stepstone.de</i>	Consumer-to-Administration
<b>Business</b>	Business-to-Consumer <i>www.quelle.de</i> <i>www.dell.de</i>	Business-to-Business <i>www.verticalnet.com</i>	Business-to-Administration
<b>Administration</b>	Administration-to-Consumer	Administration-to-Business	Administration-to-Administration

Quelle: Nach [1, S. 10]

**Tabelle 2-1: Bereiche des E-Commerce**

Außerdem kann im E-Business auch noch eine Einteilung nach den Funktionen getroffen werden. So gibt es z.B. neben dem E-Commerce auch E-Logistic, E-Procurement u.a., die von verschiedenen Theorien wie dem Supply Chain Management, Customer Relationship Management getragen werden. Dies ist aber für die weitere Arbeit nicht von Bedeutung, weswegen im folgenden E-Commerce-Modelle vorgestellt werden, die sich im B2B und B2C entwickelt haben.

### 2.1.2 E-Commerce-Modelle

Im Bereich des B2B und B2C haben sich verschiedene Modelle zum Verkauf von Produkten entwickelt, die in dynamisch und statisch unterschieden werden. Dabei wird kurz das Powershopping, die verschiedenen Varianten der Auktionen und der Online Shop betrachtet.

Unter Powershopping versteht man ein dynamisches Modell, das Relevanz für den B2C Bereich hat. Ziel dabei ist es, Preisvorteile, die ansonsten nur der Großhändler erhält, direkt an den Kunden weiterzugeben.

Ein weiteres dynamisches Modell stellt die Auktion dar, die verschiedene Varianten annehmen kann. Die bekannteste Auktionsform ist die Englische, bei der der Zuschlag an den Meistbietenden geht. Hierbei kann auch noch zwischen Echtzeit- und Langzeit-Auktion differenziert werden. [51, S. 104] Eine andere Variante stellt die Reverse Auction

dar, bei der nach dem billigsten Angebot entschieden wird, wie z.B. bei Frachtbörsen. Diese Modelle finden sowohl im Bereich des B2C und des B2B Anwendung.

Die aber wohl am meisten genutzte Form im Internet stellt der Online Verkauf dar, wobei zwischen Online Shops und Marktplätzen differenziert werden muss. Unter einem Online Shop wird das Angebot eines Unternehmens im Internet verstanden. Bei einem Marktplatz bieten mehrere Firmen auf einer gemeinsamen Plattform ihre Waren an.

### **2.2 Online Shop**

Wie zuvor erläutert, haben sich verschiedene Modelle des E-Commerce entwickelt, die sich inzwischen durch vorgefertigte Software unterstützen lassen. Die Funktionen, die technischen Ausprägungen und die Anbieter von Online Shops werden nachfolgend erläutert.

#### **2.2.1 Funktionen eines Online Shops**

Die Basis bildet der Produktkatalog, der alle verfügbaren Artikel enthält, die in verschiedene Produktgruppen eingeteilt werden. Durch eine klare Strukturierung der Artikel soll dem Käufer das Finden des Gewünschten erleichtert werden. Im B2B-Bereich wird zudem auf die Nutzung von Produktcodes, die eine eindeutige Zuordnung ermöglichen, gesetzt. So wird professionellen Einkäufern das langwierige Navigieren durch die Produktstruktur erspart. Auch wird dadurch ein Schnellzugriff auf das gewünschte Produkt verwirklicht. [51, S. 121]

Weiterhin ist in jedem Shop ein Warenkorb zu finden. In diesen werden virtuell die gewählten Produkte gelegt. Dem Käufer bietet er eine Übersicht der schon erfassten Artikel. Außerdem ist hier eine Editierung der Bestellung möglich. Dabei kann es sich um eine Änderung der Menge handeln oder ein Löschen. Professionellen Einkäufern wird auch die Option geboten, die verschiedenen Artikel mehreren Lieferadressen zuzuordnen oder einen Wunschtermin anzugeben. [51, S. 121]

Im Bereich des B2B sollte ein Shop auch noch zusätzliche Funktionalitäten unterstützen. Er muss dem Einkäufer Auskunft über die Verfügbarkeit der Ware geben können, möglichst in Echtzeit. Auch sollten schon getroffene Preisabsprachen berücksichtigt werden. Um dies ermöglichen zu können, muss eine genaue Zuordnung des Einkäufers, z.B. über eine Kennung, erfolgen.

### 2.2.2 Technische Ausprägungen von E-Commerce-Systemen

Die Vorteile des E-Commerce liegen in der einfachen Informationsbeschaffung und der schnellen Abwicklung von Geschäftsvorgängen. Dies beruht auf der Verwendung der offenen Standards des Internet wie des Transmission Control Protocol/Internet Protocol (TCP/IP) und des Hypertext Transfer Protocol (HTTP). Auch andere Dienste können zusätzlich genutzt werden, beispielhaft seien hier FTP und SMTP genannt, die je nach Anforderung verwendet werden.

Da bei einem Geschäftsabschluss oftmals sensible Daten verschickt werden, hat sich zur Sicherheit die SSL-Verschlüsselung (Secure Socket Layer) durchgesetzt. Außerdem finden noch andere Verschlüsselungen ihre Anwendung. So werden von Online Shops oft verschiedene Zahlungsarten, wie z.B. eCash, angeboten.

Weiter wird nicht auf die Technik eingegangen, da sie je nach Produkt variiert und jedes Unternehmen sich einen Shop nach seinen Wünschen wählt.

### 2.2.3 Anbieter

Durch Verbreitung des E-Commerce in der Wirtschaft wurden inzwischen die Individualentwicklungen von Online Shops größtenteils durch Standardprodukte oder Mietsoftware, die nicht weiter beschrieben wird, abgelöst. Der Leistungsumfang und der Preis dieser Produkte kann sehr unterschiedlich ausfallen. Nachfolgend werden nun einige Shops von verschiedenen Anbietern kurz vorgestellt.

Bei den Standard-Lösungen existieren Anwendungen für jede Art von Ansprüchen. Zur Einrichtung erhalten die Nutzer meist Hilfe von einem Assistenten, einem virtuellen Berater. Ein solcher ist z.B. der StoreDesignWizard von Intershop Communications, der auf Basis von Autotemplates die Erstellung des Shop erleichtert. [29, S. 103] Um dem Kunden noch einen anderen Mehrwert zu bieten, hat Intershop ein Warenwirtschaftssystem integriert.

Auch IBM bietet mit Net.commerce eine Lösung für Online Shops an, das zudem die Nutzung auf Mall-Basis ermöglicht. „Die Software wird in zwei Ausführungen angeboten: Net.commerce START und Net.commerce PRO.“ [29, S. 105] Das Start-Paket beinhaltet die Grundfunktion, drei vordefinierte Shops und ein Zahlungssystem auf Basis des SET-Protokolls. Für weitergehende Funktionen, wie die Anbindung an SAP oder EDI-Lösungen

benötigen Unternehmen Net.commerce PRO. Durch diese Differenzierung kann IBM den Wünschen unterschiedlicher Kundengruppen entsprechen.

Ein anderer Anbieter ist die omeco GmbH<sup>1</sup>, die Online Shops für den B2B-Sektor entwickelt. Besonders bei omeco webshop hervorzuheben ist die einfache Benutzung, die die Ansprüche an die Web-Browser minimal hält, was für viele Unternehmen wichtig ist. Außerdem wird konsequent auf die Sicherheit der Daten geachtet. Die technischen Merkmale werden in Kapitel 5 bei der praktischen Anwendung noch genauer vorgestellt.

### 2.3 ERP-Systeme

Nachdem Online Shops präsentiert wurden, erfolgt eine Vorstellung von ERP-Systemen, die in den meisten Unternehmen genutzt werden. Dabei wird auf die Grundkomponenten und die Anbieter eingegangen.

#### 2.3.1 Merkmale eines ERP-Systems

Da sich ein Unternehmen einer Vielzahl von Geschäftsabläufen gegenüber sieht, benötigt es Hilfsmittel um dies bewältigen zu können. Diese Aufgabe übernehmen computergestützte betriebswirtschaftliche Informationssysteme, die die betriebswirtschaftlichen Anwendungskonzepte mit der Informationstechnik verbinden [47, S. 4]. Dabei kann man zwischen Administrations-, Dispositions-, Planungs- und Kontrollsystemen unterscheiden.

Ein Enterprise Resource Planning System (ERP-System) vereint all diese Möglichkeiten. Aufgrund seiner Leistungsfähigkeit kann es definiert werden als

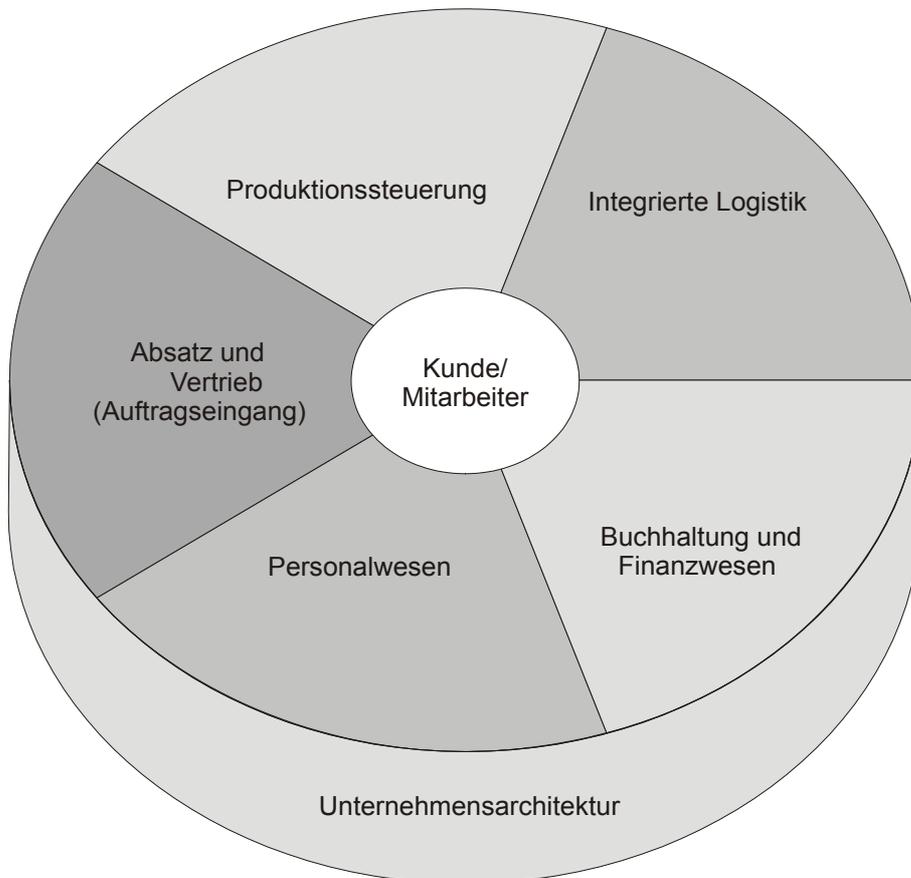
**ein integriertes Gesamtsystem, das alle Funktionen der Administration, Disposition und der Führung in einem Unternehmen unterstützt.**

Durch so ein integriertes Gesamtsystem kann eine größtmögliche Effizienz der Geschäftsabwicklung erreicht werden, da die verschiedenen Module des Systems miteinander kommunizieren können [55, S. 57]. Als Standardkomponenten findet man in ERP-Systemen die Bereiche Personal-, Rechnungswesen und Warenwirtschaft/Logistik. Zudem gibt es viele Branchenlösungen und Zusatzmodule, die das Nutzungspotenzial

---

<sup>1</sup> Siehe auch: <http://www.omeco.de>

erhöhen und oft eine Verbindung zu Komplementärsoftware ermöglichen, wie z.B. CAD-Software [38, S. 56]. Abbildung 1 stellt dies vereinfacht dar.



Quelle: [36, S. 322]

**Abbildung 2-1: Elemente des Enterprise Resource Management**

### 2.3.2 Architektur eines ERP-Systems

Ein ERP-System besitzt eine Client/Server-Architektur, die sich aus drei Grundkomponenten aufbaut:

- Datenmanagement
- Applikation
- Präsentation. [3, S. 191]

Im Datenmanagement werden die Informationen in einer Datenbank verwaltet, um diese den Applikationen zur Verfügung zu stellen. „Diese Ebene ist zuständig für die Ausführung der Abläufe und der Funktionen.“ [54, S. 148] Auf der Präsentationsebene werden die Daten für den Nutzer aufgearbeitet und im GUI (Graphic User Interface) dargestellt.

## 2 Grundlegende Betrachtungen

---

Neben der Hardware-Architektur enthält ein ERP-System eine Organisationsstruktur, die dem Unternehmensaufbau Rechnung trägt. Dabei lassen sich unter anderem Mandant und Buchungskreis unterscheiden. Durch diese Ebenen lassen sich verschiedene Berechtigungen festlegen, was für die Sicherheit der Daten wichtig ist.

### 2.3.3 ERP-Systemanbieter

Der Markt für ERP-Systeme ist sehr breitgefächert, da fast jede Unternehmung von den Vorteilen eines solchen Programms profitieren kann. Da nun aber die Bedürfnisse sehr unterschiedlich sind, haben sich die ERP-Systemanbieter oftmals auf ein Marktsegment spezialisiert. Dabei wird eine Einteilung nach der Unternehmensgröße vorgenommen, Kleinunternehmen mit bis zu 50 Arbeitnehmern, mittelständische Unternehmen mit 51 bis 500 Arbeitnehmern und Großunternehmen mit über 500 Arbeitnehmern. Zudem benötigen viele Branchen speziell auf sie zugeschnittene Programmpakete, um den verschiedenen Aspekten ihrer Unternehmung gerecht zu werden.

**Tabelle 2-2: ERP-Systemanbieter**

soll eine kleine Übersicht über verschiedene Anbieter und ihre Positionierung auf dem Markt bieten.

Anbieter	Produkt	Zielgruppe		
		Klein	Mittel	Groß
Baan Deutschland GmbH	I-Baan ERP		x	x
Bäurer AG	B2-Industrie	x	x	x
Clarfeld GmbH	Select Line	x		
Infor Business Solutions AG	Infor-COM	x	x	x
J.D. Edwards Deutschland GmbH	One-World Xe	x	x	x
Navision PC&C Vertriebs GmbH	Navision Financials/Attain	x	x	
	Navision Axapta		x	x
Oracle Deutschland GmbH	E-Business Suite 11i	x	x	x
People-Soft GmbH	People-Soft 8			x
Proalpha Software AG	Proalpha		x	x
Sage KHK Software GmbH & Co. KG	Office Line	x	x	
	Classic-Line	x	x	
SAP Deutschland AG & Co. KG	mySAP.com		x	x
	SAP.Readytowork	x	x	
Software AG	Prodis	x	x	x

Quelle: [40, S. 52ff]

**Tabelle 2-2: ERP-Systemanbieter**

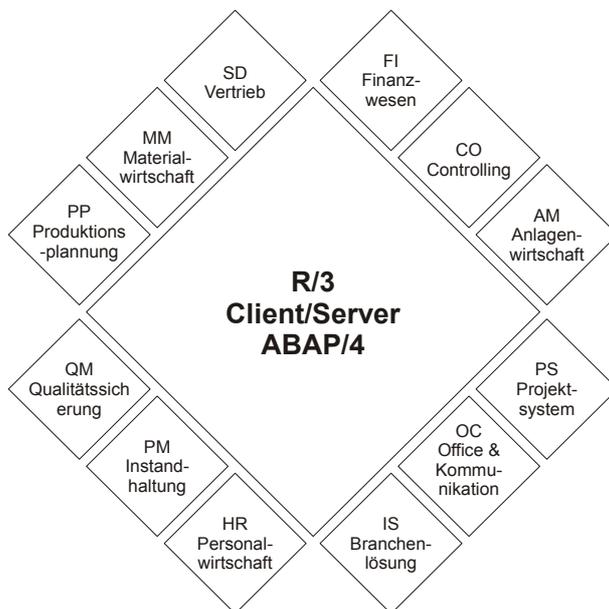
Aus diesen Anbietern sollen nun exemplarisch drei ausgesucht werden, um ihre Merkmale kurz zu beschreiben. Für den Bereich der Großunternehmen wird der Marktführer die SAP AG aufgenommen, der sich vor allem durch seine Branchenlösungen auszeichnet. Die proALPHA Software AG repräsentiert den Mittelstand und wurde wegen der

## 2 Grundlegende Betrachtungen

---

Zusammenarbeit im praktischen Teil gewählt. Die Sage KHK Software GmbH & Co. KG bietet vor allem Produkte für kleinere Unternehmen an.

Die SAP AG<sup>2</sup> hat sich als Marktführer etabliert und bietet jedem Marktsegment eine eigene Lösung an. Hier soll aber nur das Sortiment für die Großunternehmen beschrieben werden. Die SAP R/3 Software enthält alle relevanten Module, die von einem ERP-System erwartet werden. Diese können dann um weitere Zusatzfunktionen für einzelne Branchen ergänzt werden. Als Beispiele wären die Automobilindustrie, das Gesundheitswesen, die chemische Industrie u.a. zu nennen.



Quell: nach [71]

### Abbildung 2-2: SAP R3 Komponenten

Um den neueren Trends auch gerecht zu werden, bietet SAP Erweiterungen z.B. für das Supply Chain Management, das Customer Relationship Management, Mobile Commerce u.a. an.

Das ERP-System von proALPHA<sup>3</sup> unterstützt mit seinen Modulen den Mittelstand. Es beinhaltet alle klassischen betriebswirtschaftlichen Funktionen und kann genau wie SAP R/3 mit Zusatzmodulen zum Supply-Chain-Management, CRM u.a. ergänzt werden. Basis des Systems ist die Datenbank von Progress.

---

<sup>2</sup> Siehe auch: <http://www.sap-ag.de/germany>

<sup>3</sup> Siehe auch: <http://www.proalpha.de>

## 2 Grundlegende Betrachtungen

---

Für kleine Unternehmen ist die Software KHK PC-Kaufmann<sup>4</sup> geeignet. Sie enthält die Funktionen Auftragsbearbeitung, Warenwirtschaft, Anlagen-, Finanzbuchhaltung und Lohn & Gehalt. Passend für kleine Unternehmen ist die Anzahl der Mandanten begrenzt, außerdem werden gewisse Funktionen nicht automatisch ausgeführt.

Da die genutzten Systeme und die Modelle nun bekannt sind, wird im folgenden Kapitel der betriebswirtschaftliche Hintergrund der Integration betrachtet.

---

<sup>4</sup> Siehe auch: [http://www.sagekhk.de/PC\\_Kaufmann/start.html](http://www.sagekhk.de/PC_Kaufmann/start.html)

## 3 Integration von E-Business in den Geschäftsablauf

Die informationstechnische Entwicklung ermöglicht inzwischen eine Integration verschiedener Applikationen. Die Gründe für die Verknüpfungen liegen aber in den betriebswirtschaftlichen Konzepten, die nun im folgenden Kapitel behandelt werden.

### 3.1 Grundgedanken für die Integration

Der zunehmende Wettbewerbsdruck und die Globalisierung zwingen Unternehmen ihre Geschäftsprozesse umzustrukturieren. Die dafür entwickelten Managementkonzepte, wie z.B. Business Reengineering, sollen dabei einen Leitfaden bilden.

#### 3.1.1 Geschichte der Integration

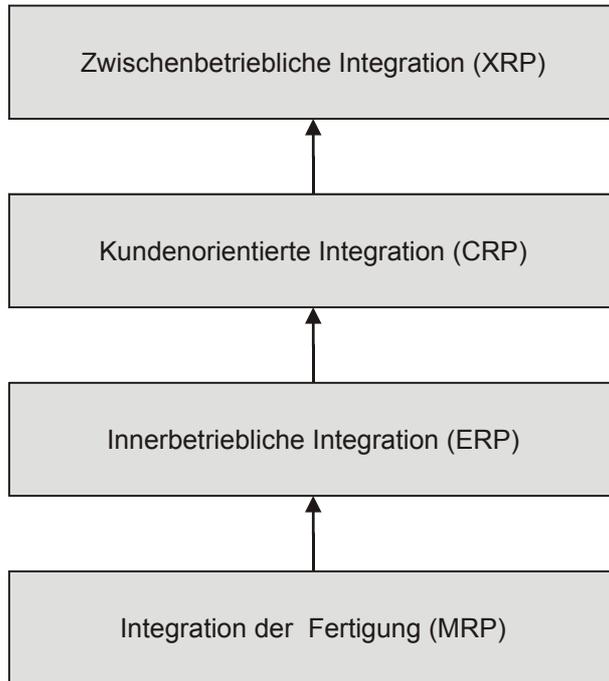
Um die Bestrebungen der Unternehmen nach Integration zu verstehen, muss man sich die Entwicklung des Trends betrachten. Abbildung 3-1 gibt dazu einen Überblick. In den Anfängen haben sich die Material Requirement Planning Systeme (MRP) entwickelt, die sich auf Produktionsprozesse beschränkten. Später ließen sich diese um weitere Unternehmensfunktionen erweitern. Dabei stellten die Unternehmen fest, welchen Nutzen sie aus einer solchen Integration ziehen konnten.

Aus diesem Trend entstanden die ERP-Systeme, die in sich schon eine Integration der betrieblichen Funktionen enthalten. Die gesteigerten Anforderungen der Unternehmen wie die weltweite Koordination von Geschäftsbereichen begünstigte den Trend. Dabei wird versucht, die alten Systeme, die hohe Wartungskosten verursachen, gegen neue auszutauschen. Auch sollen diese System leichter auf verschiedene Umfelder reagieren und die Entscheidungsfindung innerhalb einer Unternehmung erleichtern.

Als nächste Integrationsstufe identifizieren Kalakota und Robinson die kundenorientierte Integration [36, S. 328]. Der Sinn dabei ist, die Bedürfnisse der Kunden an die erste Stelle zu stellen. In der heutigen Zeit bedeutet dies, dass Lieferungen viel schneller erfolgen müssen, da mehr Serviceleistungen erwartet werden. Ermöglicht wird dies durch Customer-Centric Resource Planning (CRP), das eine kontinuierliche Planung zulässt.

In der letzten Integrationsstufe werden die Unternehmensgrenzen überschritten und eine zwischenbetriebliche Integration angestrebt. Dieser Schritt ist für viele Firmen schwer zu bestreiten, da sie immer noch Angst haben, ihre Daten anderen preiszugeben. Aber nur

wenn dieser Sprung geschafft ist, kann eine B2B Integration vorgenommen werden. Dabei werden die Unternehmen durch die Ideen des Supply Chain Management unterstützt. Diese letzte Stufe wird mit Extended Resource Planning bezeichnet.



Quelle: [36, S. 324]

**Abbildung 3-1: Integrationsstufen**

#### 3.1.2 Managementkonzepte

Nicht nur die technischen Möglichkeiten brachten die Unternehmen zum Umdenken, sondern auch verschiedene neue Managementkonzepte, die im Zuge der Technologisierung aufkamen. So fordern Hammer und Champy in ihrem Werk Business Reengineering, dass Unternehmen ihre Geschäftsprozesse überdenken sollen. Diese Prozessorganisation begründet sich auf verschiedenen Faktoren. Unternehmen fokussieren sich immer mehr auf die Kundenwünsche, seien es externe als auch interne. Außerdem muss eine Rentabilitätssteigerung des Unternehmens erreicht werden, damit es dem steigenden Wettbewerbsdruck standhält. Um dies zu bewerkstelligen, müssen die Unternehmen umdenken und ihre bisherige Vorgehensweise effizienter gestalten.

Zur Effizienzsteigerung wird im Business Reengineering auf Computertechnologien gesetzt, sei es durch die Nutzung integrierter Standardsoftwaresysteme oder den Aufbau von Netzwerken. Für eine weitreichende Optimierung sollte nicht nur das eigene Unternehmen in die Prozessorganisation eingebunden werden, sondern auch die Zulieferer bzw. Kunden. Eine Integration der Systeme ist die Grundvoraussetzung.

Auch das Lean Management oder Just-in-Time setzen auf die Unterstützung durch Computersysteme. So müssen Firmen heutzutage ihre IT-Infrastruktur, sowohl intern als auch extern, ausbauen, um wettbewerbsfähig zu bleiben.

Dabei stellt sich die Frage, was sich genau die Unternehmen durch eine Vernetzung versprechen. Die Zeitersparnis ist sicherlich ein Grund dafür, wobei man differenzieren kann zwischen der Zeit, die sowohl durch eine Automatisierung der Prozesse erreicht wird, als auch durch das Wegfallen von Dateneingaben.

Die neueren Produktionskonzepte wären ohne Computerunterstützung nicht denkbar, da z.B. für eine Just-in-Time-Fertigung eine genaue Koordination der Warenströme gewährleistet sein muss. Dadurch können die Produktionsanlagen effizienter genutzt werden.

#### **3.1.3 Marktveränderungen**

Durch die Nutzung von E-Business hat sich das Verhalten der Kunden verändert. So hat sich der Informations-Push in einen Informations-Pull verwandelt. Durch den leichten Vergleich von Angeboten muss ein Konzern nun einen Mehrwert bieten, um das Interesse zu wecken. Woraus sich folgern lässt, dass ein Unternehmen nur mit einem integrierten System bestehen kann, das alle relevanten Daten in Echtzeit zur Verfügung stellt.

Die Serviceleistung eines Unternehmens wird inzwischen auch immer mehr auf die Probe gestellt und nur wer dabei flexibel reagieren kann, hat Chancen auf dem Markt. Deswegen reicht die Nutzung eines ERP-System nicht mehr aus, da es nur für einen mittel- bis langfristigen Planungshorizont entwickelt wurde. Erst in Verbindung mit einem CRP-System, das eine kontinuierliche Planung ermöglicht, kann dies verwirklicht werden.

Um eine Änderung in einem Unternehmen hervorrufen zu können, muss der Planungshorizont bestimmt und eine Strategie festgelegt werden. Der folgende Abschnitt verdeutlicht dabei die Ebenen der Integration.

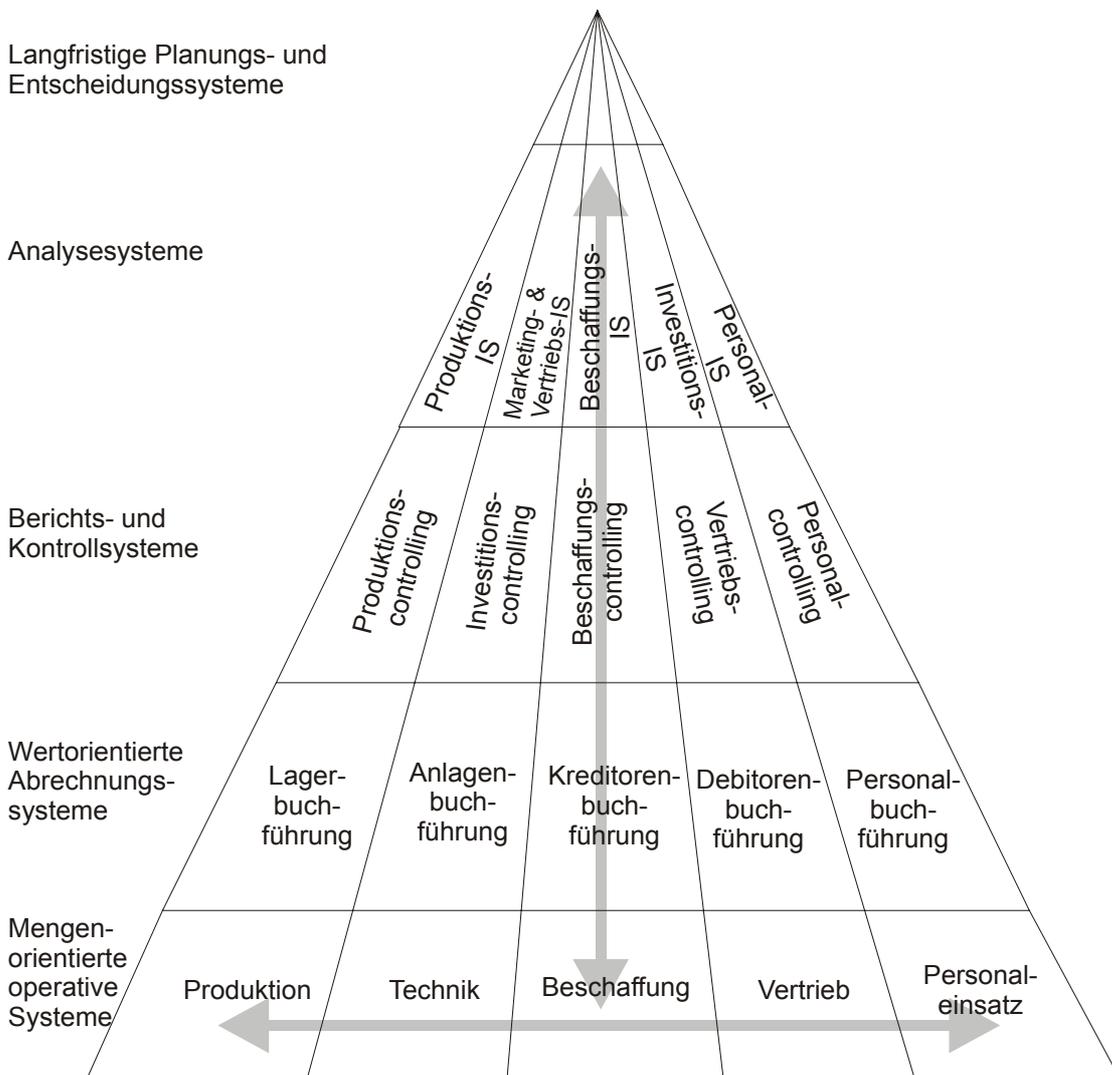
### **3.2 Integrationsstrategie**

Zur Realisierung der zuvor genannten Ziele muss ein Konzern bestimmen, in welcher Art er das E-Business integrieren will. Dabei kann zwischen der Integrationsrichtung und der Integrationstiefe unterschieden werden. Eine Differenzierung in eine inner- und überbetriebliche Integration kann außerdem vorgenommen werden.

#### 3.2.1 Integrationsrichtung

Bei der Integrationsrichtung kann zwischen einer horizontalen und einer vertikalen Integration in der Betriebswirtschaftslehre unterschieden werden. Eine horizontale Integration stellt dabei eine Verbindung entlang der Wertschöpfungskette dar und schließt Zulieferer und Kunden ein. Bei einer Verknüpfung mit Unternehmen auf derselben Ebene der Wertschöpfungskette, wird von einer vertikalen Integration gesprochen. Außerdem besteht auch noch die Möglichkeit, Netzwerke aufzubauen.

Man kann aber auch Integrationsrichtungen an Software Systemen festmachen. Unter der horizontalen Integration wird die Kommunikation der Systeme auf derselben Unternehmensebene verstanden. Die vertikale Integration dagegen integriert die verschiedenen Ebenen. Abbildung 3-2 stellt die Softwareintegration dar.



Quelle: [47, S. 5]

**Abbildung 3-2: Integrierte Informationssysteme**

#### 3.2.2 Integrationstiefe

Eine weitere strategische Entscheidung liegt in der Integrationstiefe. Diese wird am Beispiel der Studie von IBM und der Zeitschrift Impulse<sup>5</sup> beschrieben. Auf Basis der informationstechnischen Nutzung für die Abwicklung der Geschäftsprozesse wird eine Einteilung in 6 Gruppen vorgenommen.

In der ersten Gruppe befinden sich Unternehmen, die weder E-Mail noch Online-Dienste für ihr Geschäft entdeckt haben. Daher sollen sie hier auch nicht weiter betrachtet werden. Ebenso die zweite und dritte Gruppe, in der noch keinerlei direkte Integration in den Geschäftsprozess erfolgt, sondern einfach technische Medien für die Kommunikation und zu Marketingzwecken verwendet werden.

Interessant werden erst die Gruppen vier bis sechs, in denen eine direkte Integration in den Geschäftsablauf erfolgt.

Hierbei kann die Nutzung als zusätzlicher Vertriebsweg als die niedrigste Integrationstiefe angesehen werden, da man vorher noch von keiner Integration im eigentlichen Sinne sprechen kann.

Sobald der Gedanke des E-Business auch auf die Geschäftspartner, seien es Zulieferer oder Kunden, ausgeweitet wird und die einzelnen Systeme miteinander bedingt kommunizieren, wird die nächste Integrationsstufe erreicht.

Vollständig integriert ist das E-Business erst, wenn alle Systeme entlang der Wertschöpfungskette vernetzt sind. Hierbei kann man, wie zuvor erläutert, von einer horizontalen Integration sprechen, bzw. einer überbetrieblichen.[3, S. 154]

Daraus kann man schließen, dass die Integrationstiefe immer mehr zunimmt, je mehr unterschiedliche Systeme miteinander kommunizieren und je stärker die Vernetzung mit dem Geschäftspartner wird.

Es gibt noch andere Integrationsgesichtspunkte, wie z.B. die Integration auf Daten- Objekt- oder Prozessebene. Diese werden im nächsten Kapitel beschrieben, da diese Ebenen sehr eng mit den IT Konzepten verknüpft sind.

---

<sup>5</sup> Siehe auch: [http://www-5.ibm.com/de/mittelstand/download/bericht\\_ebusiness2001.pdf](http://www-5.ibm.com/de/mittelstand/download/bericht_ebusiness2001.pdf)

# 4 Enterprise Application Integration

Die Integration von Anwendungen im Zuge des E-Business steht außer Frage. Nachfolgend werden die Begriffe wie EAI, Middleware und B2B Application Integration genauer erläutert, außerdem werden die zugrundeliegenden Standards und Technologien dargestellt.

## 4.1 Grundlagen

In diesem Abschnitt werden nun die Grundbegriffe definiert und die Entwicklung und Zielsetzung von EAI herausgearbeitet und gegenüber B2B Application Integration und Middleware abgegrenzt.

### 4.1.1 Begriffsdefinitionen

Um die Verständigung zu erleichtern, werden hier die genutzten Begriffe für diese Arbeit definiert, da sich in der Literatur keine einheitliche Nutzung der Begriffe durchgesetzt hat.

#### **Middleware**

In der Literatur wird Middleware auf verschiedene Weise definiert. Am treffendsten kann Middleware als

**„Software für verteilte Anwendungen zur Überbrückung der Heterogenität unterschiedlicher Systeme und Netze“ [60, S. 2]**

verstanden werden.

Des Weiteren kann Middleware nach ihrer Funktionsweise unterschieden werden:

- Datenbankorientierte Middleware
- Funktionsorientierte Middleware
- Transaktionsorientierte Middleware
- Nachrichtenorientierte Middleware
- Komponentenorientierte Middleware

Die datenbankorientierte Middleware ermöglicht die Integration von Daten, die in verteilten Datenbanken vorliegen. Dies erfolgte bis jetzt durch proprietäre Tools der Hersteller. Inzwischen werden immer mehr XML-Schnittstellen angeboten.

Funktionsorientierte Middleware oder RPC-based Middleware ermöglicht die Nutzung von Funktionen auf entfernten Systemen durch RPC (Remote Procedure Call). An RPC erkennt man noch deutlich den Ausgangspunkt der Entwicklung von Middleware, der im Client-/Servermodell lag.

Mit der transaktionsorientierten Middleware wird eine zuverlässige Informationsverarbeitung durch Beachtung der Eigenschaften der Atomarität, Konsistenz, Isolation und Dauerhaftigkeit erreicht.

Nachrichtenorientierte Middleware oder Message-based Middleware geht mit der Übermittlung von Nachrichten einen anderen Weg und vermeidet damit die sonst fälligen Spaghetti-Programme, da sie einen Message-Bus zwischen Applikationen bereitstellt. Auf die genaue Funktionsweise von MOM wird in Abschnitt 4.3 eingegangen.

Komponentenorientierte Middleware oder Object-oriented Middleware stellt den Begriff des unabhängigen Objektes in den Vordergrund. Bekannteste Vertreter dieser Middleware sind CORBA, COM/DCOM und EJB.

Nachdem nun die Grundbegriffe der zugrundeliegenden Technologie kurz erläutert wurden, wird nun der Begriff des EAI vorgestellt, um anschließend gegenüber der B2B Application Integration abgegrenzt zu werden.

### **EAI**

Zu EAI (Enterprise Application Integration) werden in der Literatur verschiedene Definitionen gegeben, wobei nachfolgend einige Ausgewählte vorgestellt werden.

EAI wird bei Ovum<sup>6</sup> folgendermaßen definiert:

„Enterprise application integration (EAI) combines the technologies and processes that enable custom-built and/or packaged business applications to exchange business-level information in formats and contexts that each understands.“ [61]

Bei PricewaterhouseCoopers<sup>7</sup> wird EAI so verstanden:

„Unter dem Begriff Enterprise Application Integration (EAI) werden Technologien zusammengefasst, welche automatisiert die Kommunikation und Interoperabilität zwischen

---

<sup>6</sup> Siehe auch: [www.ovum.com](http://www.ovum.com)

<sup>7</sup> Siehe auch: [www.pwcglobal.com](http://www.pwcglobal.com)

Anwendungen und Geschäftsprozessen innerhalb und zwischen Organisationen ermöglichen.“ [64, S. 7f]

Schließlich definiert Linthicum EAI kurz als:

„... EAI is the unrestricted sharing of data and business processes among any connected applications and data sources in the enterprise.“ [35]

In dieser Arbeit wird EAI nun wie folgt definiert:

**Zusammenfassung von Technologien und Prozessen zur automatisierten Kommunikation und Interoperabilität zwischen Anwendungen und Geschäftsprozessen innerhalb einer Organisation.**

Nachdem der Begriff der EAI klar definiert wurde, kann in der Definition von B2B Application Integration eine Erweiterung von EAI gesehen werden.

### **B2B Application Integration**

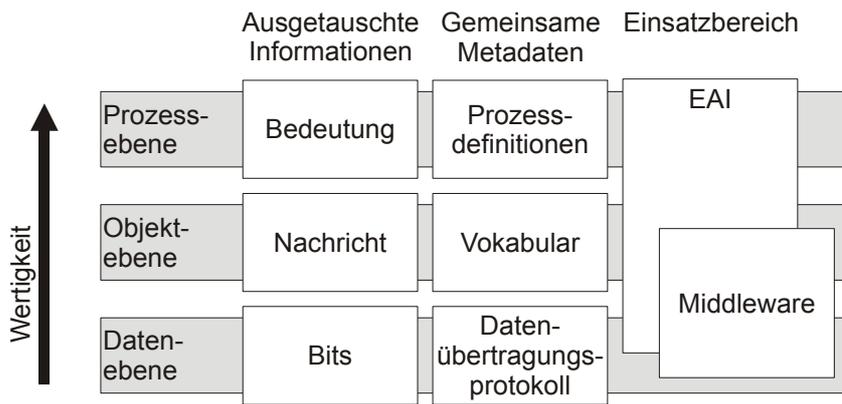
In dieser Arbeit wird B2B Application Integration von EAI abgegrenzt, da unter EAI nur die Verbindung von Applikationen innerhalb einer Unternehmung verstanden wird. Daraus kann man die Definition von B2B Application Integration ableiten:

**„... B2B application integration is the integration of systems between organizations to support any business requirements, ...“** [34, S. 16]

Da aber die Technologien des EAI auch zur Vernetzung von verschiedenen Unternehmen untereinander dienen können, werden in dieser Arbeit auch B2B-Standards vorgestellt.

#### **4.1.2 Integrationsarten**

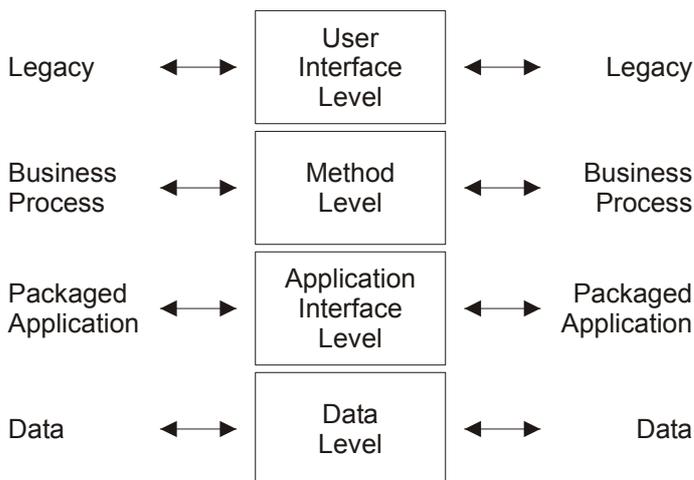
Enterprise Applications können auf verschiedene Weise ins Unternehmen eingebunden werden. Das Zusammenwirken dieser miteinander vernetzten Applikationen kann auf diversen Ebenen stattfinden. Diese werden in der Literatur meist nach Daten-, Funktions- bzw. Objekt- und Prozess-Ebene klassifiziert. Diese prinzipielle Einteilung zeigt Abbildung 4-1 und identifiziert gleichzeitig die Bereiche, die von klassischen Middleware-Lösungen bzw. von EAI-Lösungen abgedeckt werden.



Quelle: Nach [43]

**Abbildung 4-1: Integrationsarten**

Des Weiteren können diese Integrationsarten auch aus der Sicht der zu integrierenden Systeme aufgelöst werden, dabei erweitert sich diese klassische Dreiteilung um eine Integrationsebene, denn die Integration der Altsysteme (legacy-systems) lässt sich nicht den drei anderen zuordnen. Dieser Ansatz, den Linthicum beschrieben hat, wird in Abbildung 4-2 präsentiert.



Quelle: 208

**Abbildung 4-2: Arten von Enterprise Application Integration**

Unter User-Interface-Level EAI wird die Integration von alten Systemen, wie Mainframe-Applikationen, verstanden, die keine Datenbank bzw. Business-Prozess-Schnittstellen besitzen und deren Programme nicht erweiterbar sind. Diese können nur über ihr User-Interface integriert werden, d.h. ihnen werden über Emulatoren die notwendigen Benutzereingaben simuliert. Dies nennt man screen scraping. Nach Linthicum ist dies zwar primitiv, aber lässt sich in einigen Fällen nicht vermeiden, wenn die Altsysteme nicht ersetzt werden können.

Im Gegensatz zu der User-Interface-Integration werden die klassischen Integrationsarten in der Literatur nicht einheitlich bezeichnet. Während für den Daten-Level noch Einverständnis herrscht, findet man für den Prozess-Level auch die Bezeichnung als Method-Level und anstelle von Funktions-Level wird oft auch Object-Level verwendet. Dabei kann man diese Bezeichnungen durchaus als Synonyme füreinander annehmen, da z.B. Objekte eine public Schnittstelle, die aus Methoden und Funktionen besteht, aufweisen und so dieselben Kriterien besitzen, die dem Funktions-Level zugesprochen werden.

Die Integration auf Datenebene tauscht Daten zwischen verschiedenen Datenspeichern, meist Datenbanken aus. Dazu werden die Daten aus dem einen System ausgelesen und in dem anderen System wieder eingespeichert. Oft wird noch eine Datentransformation dazwischen geschaltet. Diese Art der Integration ist einfach und billig, da keine Applikationen verändert werden müssen.

Auf der Objekt- oder Funktionsebene werden Schnittstellen von Applikationen benutzt, um Informationen auszutauschen und Business Logik bereitzustellen. Die Integration wird bei dieser Art durch den Funktionsumfang der zu verbindenden Applikationen eingeschränkt. Solche Schnittstellen findet man meist in Standardsoftware (packaged application) vor, deren typische Vertreter SAP, PeopleSoft, Baan usw. sind. Zur Integration müssen Systeme gefunden werden, die die Informationen von dem Quellsystem abfragen können und dann in ein Format transformieren, das von dem Zielsystem verstanden wird. Beim Stand der heutigen Entwicklung werden dafür Message Broker bevorzugt. In Kapitel 5 wird eine derartige Integration in der praktischen Anwendung vorgestellt.

Letztendlich stellt die Prozessebene die höchsten Anforderungen an eine Integration, denn hier wird die Business Logik von mehreren Applikationen verwendet. Durch die gemeinsame Nutzung einer Methode von mehreren Applikationen verringert sich der Aufwand für die Implementierung. Als Grundlage für diese Ebene dienen zum einen Application Server, die als zentraler Server alle benötigten Methoden bereitstellen. Zum anderen können Methoden, die von verschiedenen Applikationen bereitgestellt werden, von allen anderen gemeinsam genutzt werden, z.B. mit Hilfe von distributed objects.

Schinzer geht in [49] besonders auf die Integration mit XML und den Einsatz von sogenannten EAI-Werkzeugen ein und stellt besondere organisatorische und technische Aspekte für jede Integrationsart in einer Integrationsmatrix, siehe Tabelle 4-1, heraus, die

einen guten Überblick über die Anforderungen an die IT in einem Unternehmen zur Enterprise Application Integration geben.

<b>Integrationsbereich</b> <b>Integrationsgegenstand</b>	<b>Organisatorische Aspekte</b>	<b>Technische Aspekte</b>
<b>Datenintegration</b>	Direktimport und -export durch semantische Mappingtabellen  Verständigung über Datenstrukturen und die inhaltliche Bedeutung	Innerbetrieblicher Datenaustausch auf Basis von XML  Zwischenbetrieblicher Datenaustausch auf Basis von XML und EDIFACT
<b>Funktionsintegration</b>	Logische und inhaltliche Abstimmung von Laufreihenfolge und -häufigkeit  Ablaufsteuerung der Planungskomponenten	Funktionsaufruf aus ERP, DW-, CRM- oder EC-Anwendungen  Einsatz von Enterprise Application Integration-Werkzeugen
<b>Prozeßintegration</b>	Anwendungsübergreifende Ereignissteuerung  Continuous Process Engineering zur Identifikation Integrationsfähiger Prozeßbestandteile	Kopplung von Anwendungsbausteinen über Prozeßmodelle  Steuerung über APIs/RFCs, Einsatz von Workflowsystemen

Quelle: [49, S. 3]

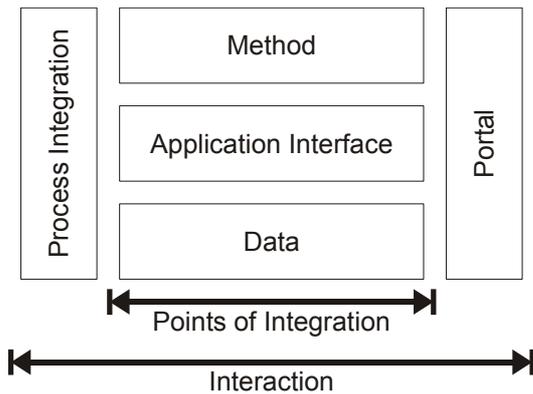
**Tabelle 4-1: Integrationsmatrix**

Nachdem die Integrationsarten für EAI verdeutlicht wurden, gibt der folgende Abschnitt einen Überblick über aktuelle Entwicklungstrends.

### 4.1.3 Entwicklungstrends

Wie im Abschnitt 3.1.1 angesprochen, weitet sich die Integration immer mehr auf Systeme zwischen verschiedenen Unternehmen aus. Diese derzeit vollführte B2B Application Integration kann nach Linthicum in 5 Bereiche gegliedert werden, die in Abbildung 4-3 dargestellt sind. Wobei die untersten Stufen mit denen der EAI gleichzusetzen sind und die Process Integration die Integration von Business Prozessen zwischen Unternehmen beschreibt und die Method Integration die Business Prozesse innerhalb des Unternehmens anspricht.

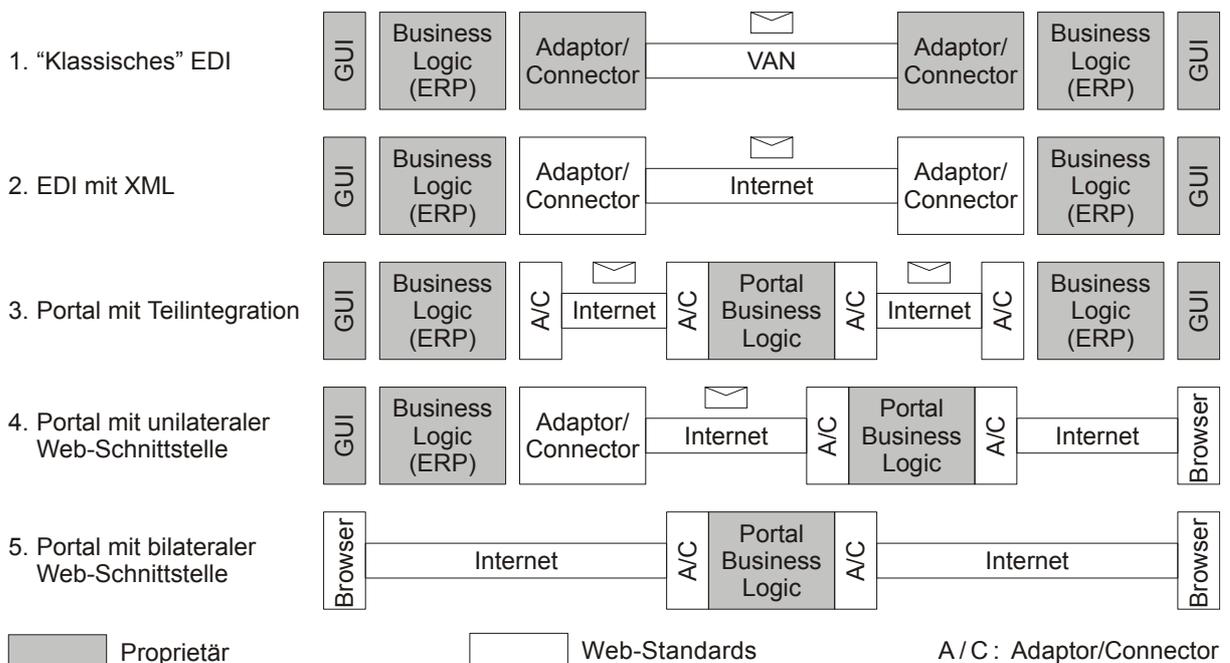
## 4 Enterprise Application Integration



Quelle: [34, S. 27]

**Abbildung 4-3: Arten von B2B Application Integration**

Die Kommunikation erfolgt meist über Portale, die sich im Integrationsprozess von Applikation zwischen Unternehmen als vorteilhaft herausgestellt haben. Wie Abbildung 4-4 zeigt, durchlief die Entwicklung verschiedene Stufen.



Quelle: [36, S. 669]

**Abbildung 4-4: Varianten zur B2B-Integration**

Diese können nach der Kommunikationsbeziehung und der Koordination zwischen den Unternehmen unterschieden werden.

Eine 1:1 Verbindung der IT-Systeme zweier Unternehmen findet bei der Peer-to-Peer-Kommunikation statt, währenddessen bei Extranets, Portalen, Katalogen und Beschaffungssystemen eine 1:N Verbindung vorherrscht, bei der ein Unternehmen mehreren anderen einen Zugang bietet. Bei Marktplätzen und Prozessportalen wird eine N:M

Verbindung realisiert, d.h. mehrere Teilnehmer auf der Kunden- und Anbieterseite werden miteinander verbunden.

Bei den beiden ersten Stufen in Abbildung 4-4 erfolgt die Verbindung der Unternehmen entweder über Value-Added Networks (VAN) bei der Nutzung von EDI oder über das Internet bei der von XML. Die gesamte Anwendungslogik befindet sich auf der Seite der Teilnehmer.

Die Schritte 4 und 5 zeigen die B2B-Integration bei großen Unternehmen, die über gewaltige ERP-Systeme verfügen. Diese müssen große Datenmengen von EDI- und XML-Dokumenten verarbeiten und eine Auslagerung der Anwendungslogik würde eine Auslagerung der ERP-Systeme nach sich führen und damit würde es zu unternehmenskritischen Schnittstellen kommen.

Deswegen fungiert in Schritt 3 der Hub als zentraler Knoten des Systems zur Konvertierung der Nachrichten. Er könnte entweder eine grundlegende Formatkonvertierung, z.B. von XML nach EDIFACT, oder eine Konvertierung von einzelnen Datenelementen durchführen. Bei der Kommunikation eines großen Unternehmens mit mehreren kleineren, wäre auch die Variante nach Schritt 4 denkbar, bei der die Webschnittstelle des einen Unternehmens die Nachrichtenschnittstelle des anderen ist. Unter anderem können über die WEB-Schnittstelle Aufträge von kleinen Unternehmen erfasst werden und über das Portal, z.B. als EDI-Nachricht, an das große Unternehmen gesendet werden. Dieses nimmt damit alle kleinen Kunden als einen virtuellen, großen Kunden wahr.

Vor allen bei kleinen Unternehmen lässt sich der letzte Schritt gut einsetzen, der die Anwendungslogik zentral im Hub konzentriert. Die Schnittstelle zum Portal bildet dabei der WEB-Browser, der über das Internet mit dem Portal kommuniziert. Diese Lösung vermeidet die kostspielige, lokale Einrichtung der Anwendungslogik bei den kleinen Unternehmen.

Der Abbildung 4-4 kann entnommen werden, dass von Schritt 1 und 2 bis zum Schritt 5 immer mehr Anwendungslogik im Hub erforderlich ist. Anfangs verbindet der Hub die ERP-Systeme der Teilnehmer und stellt dafür Anwendungslogik zur Verfügung. In den weiteren Schritten schreitet die Zentralisierung voran, d.h. es „... konzentrieren sich immer mehr Anwendungskomponenten auf dem Portal, ...“ [36, S. 671] Schließlich besteht in

Schritt 5 nur noch ein leichtgewichtiger, flexibler Zugang über den Browser zu dem sich von seinen Teilnehmern abgabelten Portal.

### 4.2 Basistechnologien

Die Grundlage für jegliche E-Business stellt XML dar. In diesem Kapitel soll eine Klassifizierung der Basistechnologie erfolgen. Abbildung 4-6 zeigt die Gliederung von XML, angefangen vom Basisstandard bis zu den Spezifikation für einzelne Branchen.

#### 4.2.1 XML

Aus der Notwendigkeit heraus, eine Auszeichnungssprache für strukturierte Daten, die einen geringeren Umfang als SGML hat, für das WWW zu entwickeln und zugleich genauso einfach ist wie HTML, ist XML entstanden. XML<sup>8</sup> (eXtensible Markup Language) ist sowohl eine Metasprache zur Definition von Datenrepräsentationssprachen als auch eine Datenrepräsentationssprache. Dabei ist nicht vorgegeben, welchen Inhalt die enthaltenen Daten annehmen, sondern nur, dass sie eine beschreibbare Struktur aufweisen müssen.

Als Initiator von XML gilt Tim Bray, der mit Charles Goldfarb, Paul Prescod, James Tauber und Robin Cover, die alle aus dem SGML-Umfeld stammen, XML entwickelte und in enger Zusammenarbeit mit Tim Berners-Lee, dem Gründer des WWW, beim W3C standardisieren ließ. Die Entwicklung und Koordinierung von XML und weiteren Standards, wie VRML, HTML, Java u.a., erfolgt am W3C<sup>9</sup> (World Wide Web Consortium) mit Sitz am MIT (Massachusetts Institute of Technology) in Boston. Neben Vertretern aus Forschung und Lehre sind auch Vertreter bekannter Soft- und Hardwarehersteller am W3C tätig.

Die Ziele, die mit der Entwicklung von XML verfolgt wurden, können in [7] nachgelesen werden und sollen hier kurz vorgestellt werden:

- XML soll sich im Internet auf einfache Weise nutzen lassen.
- XML soll ein breites Spektrum von Anwendungen unterstützen.
- XML soll zu SGML kompatibel sein.
- Es soll einfach sein, Programme zu schreiben, die XML-Dokumente verarbeiten.
- Die Zahl optionaler Merkmale in XML soll minimal sein, idealerweise Null.

---

<sup>8</sup> Siehe auch: <http://www.w3.org/TR/REC-xml/>

<sup>9</sup> Siehe auch: <http://www.w3.org/>

- XML-Dokumente sollten für Menschen lesbar und angemessen verständlich sein.
- Der XML-Entwurf sollte zügig abgefasst sein.
- Der Entwurf von XML soll formal und präzise sein.
- XML-Dokumente sollen leicht zu erstellen sein.
- Knappheit von XML-Markup ist von minimaler Bedeutung.

Ein XML-Dokument besteht i.A. aus einer DTD (Document Typ Definition) und den Elementen für sich. Die DTD beschreibt die Syntax für die Elemente und die Verschachtelung selbiger sowie, welche Attribute dem Element zugewiesen werden können und den Inhaltstyp eines Elements oder Attributs. Die Elemente können wiederum Subelemente oder Attribute beinhalten sowie einen Inhalt haben.

Unterschieden wird bei XML-Dokumenten zwischen wohlgeformten und gültigen Dokumenten. Als wohlgeformt wird ein XML-Dokument nur dann bezeichnet, wenn alle Elemente einen Start- und einen dazugehörigen Endtag haben oder bei leerem Inhalt kann der Endtag entfallen und wird durch einen abschließenden Backslash im Starttag ersetzt. Außerdem müssen alle Attributwerte in Anführungszeichen stehen und die Elemente ohne Überlappung verschachtelt sein. Für ein wohlgeformtes XML-Dokument ist keine DTD notwendig. Anders dagegen bei den gültigen XML-Dokumenten, die zwingend eine DTD besitzen müssen, da ihr Inhalt gegen diese DTD validiert wird. Demzufolge werden solche Parser als validierende Parser bezeichnet. Die hierfür benötigte DTD wird entweder im Dokument selbst, lokal auf dem Rechner, im Netz oder im Browser bereitgestellt.

Da XML aus der Dokumentenverwaltung abstammt, wird in der DTD festgelegt, welchen Elementen Subelemente oder Text untergeordnet wird. Die Möglichkeiten dieses Inhaltsmodells reichen dabei von Text, nur Element, leer, Mixed bis zu Any. Das Typensystem von XML ist relativ schwach und unterscheidet nur Text und die XML spezifischen Typen, wie identifizierende Attribute für Elemente, Referenzen und Mehrfachreferenzen. Es fehlen aber die von anderen Programmiersprachen her bekannten Datentypen, wie numerische Typen, die aber auch beim Entwurf von XML nicht gefordert waren. Genauso können Attribute in ihrer Verwendung spezifiziert und somit als benötigt oder optional gekennzeichnet werden, außerdem kann bei einer Aufzählung ein Default-Wert festgelegt werden. Es gibt noch Entities, die reine Textersetzungen sind. Diese beziehen sich auf einzelne Zeichen, Textabschnitte innerhalb eines Dokumentes oder

dokumentenweite Einbettungen. Entities können somit gut zur Zusammenfassung von Reusables in einer Datei genutzt werden.

Damit nun die XML-DTDs von verschiedenen Autoren miteinander verzahnt werden können, wurde das Konzept der Namensräume<sup>10</sup> eingeführt. Damit können Elementnamen eindeutig durch Namenszusätze gekennzeichnet werden.

In [36] werden einige Vorteile, Nachteile und Einschränkungen von XML genannt, die sich im Bezug auf eine Eignung im B2B oder internen Integration abzeichnen.

### **Vorteile**

Als Hauptvorteil von XML sieht Merz zu Recht die Vereinheitlichung der Repräsentation von Dokumenten, die nun alle auf der gleichen Syntax für Marken beruhen. Somit kann jedes XML-Dokument mit jedem allgemeinen XML-Werkzeug automatisch verarbeitet werden. Auch fördert XML die Automatisierung der Dokumentenverarbeitung, indem der Parser den erheblichen Anteil an Arbeit übernimmt und die Validierung mit Hilfe der DTD oder XML-Schema des Dokumentes vornimmt. Aus diesen Gründen setzt sich XML heute als Standardformat durch und wird bei Neuentwicklungen als Format für strukturierte Dokumente bevorzugt. Die leichte Verständlichkeit von XML-Dokumenten für den Leser ist ein weiterer Vorteil, da sich durch sprechende Bezeichner nachvollziehen lässt, um welche Daten es sich handelt. Auch die Einfachheit von XML trägt zu starker Akzeptanz bei, aber die große Menge der darauf aufbauenden Standards erschwert den Überblick.

Redundanz wird sonst nicht unbedingt als Vorteil empfunden, aber trägt zu einigen oben genannten Vorteilen bei, da sich neben dem Dokumenteninhalte auch viele Metainformationen befinden. Dieser Vorteil wird durch einen größeren Dokumentenumfang begleitet, der aber für die typischen Einsatzgebiete keinen Nachteil darstellt. Mit seinem nur 30-seitigen Standarddokument ist XML ein effizienter Standard in dem Sinne, dass mit minimalem Aufwand ein maximaler Nutzen erreicht wird, wie er beispielsweise mit dem 500-seitigen SGML-Standarddokument nicht möglich wäre.

Durch die Koordination des W3C ist XML ein unabhängiger Standard, der durch die Mitarbeit von Microsoft, Sun, IBM u.a. geprägt ist, und meist im gemeinsamen Konsens publiziert wurde. Als Internet-Hype der letzten Jahre besitzt XML eine weltweit aktive Entwickler-Community, die durch den Open Source Charakter die Entwicklung rasant

---

<sup>10</sup> Siehe auch: <http://www.w3.org/TR/REC-xml-names/>

vorantreibt. Somit sind XML-Tools preiswert und allgemein verfügbar, was sich aber nur auf Parser, Editoren, XSL-Prozessoren bzw. -Design-Tools, Datenbank-Integrationswerkzeuge u.ä. bezieht, die deutlich günstiger als EDI-Tools sind. Die Anwendungen selber liegen aber im gleichen Preisbereich wie EDI.

### **Nachteile**

Von XML wird meist zu viel erwartet, denn eine Softwareentwicklung fängt nicht bei XML an, sondern hört dort auf. Dazwischen liegen weitere Schritte, angefangen bei der Analyse der Geschäftsprozesse, dem Entwurf einer übergreifenden Softwarearchitektur, die Zerlegung in operative Komponenten und Schnittstellen und der Definition von Datentypen. „Grundsätzlich kann XML seine Vorteile nur ausspielen, wenn die darunter liegende Architektur der Unternehmenssoftware entsprechend flexibel ist.“ [36, S. 231]

Durch die vielen parallel laufenden Entwicklungen wird die XML-Standardisierung in der Nähe von Anwendungen immer beschwerlicher, da mehrere Alternativen zu prüfen sind. Im Vergleich zu EDI-Dokumenten ist die Verarbeitung von XML sehr zeitaufwendig. Dies stellt aber nur für große Unternehmen mit mehreren tausend Transaktionen pro Stunde ein Problem dar.

Weil XML als dokumentenorientierte Metasprache entwickelt wurde, weist es nur ein sehr schwaches Typsystem auf. Es gibt keine Möglichkeit, die von Programmiersprachen her bekannten Typen zu definieren, dies wird erst mit dem Übergang zu XML-Schema gelöst.

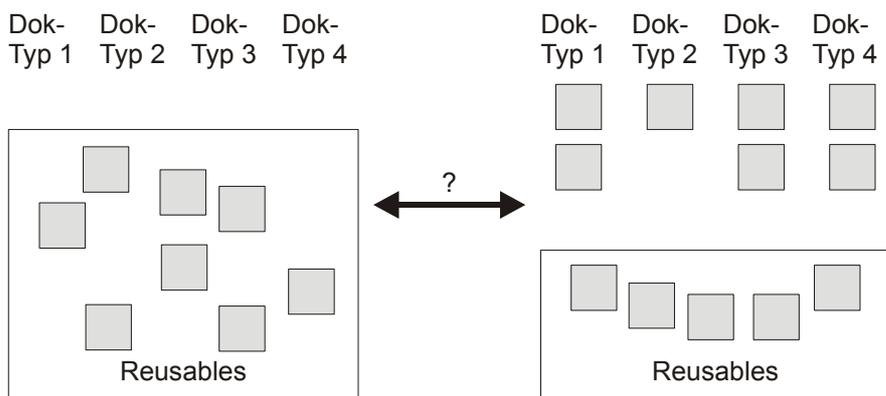
### **Einschränkungen**

Die fehlende Objektorientierung von XML stellt nur eine Einschränkung dar und wird beim Übergang von DTDs auf das XML-Schema behoben. Im Gegensatz zu UML ist XML keine Modellierungssprache, wird aber gerne dazu gezählt. Richtig ist, dass XML zur Speicherung von Modellen benutzt wird.

### **Standardisierung**

Die Standardisierung von XML-Dokumenten vereint meist zwei Bereiche, einerseits die Definition von Nachrichten oder Schemata für Dokumententypen und andererseits die Vereinheitlichung von Prozessen. Dabei ist der Dokumententypenentwurf direkt mit dem Softwareentwurf verbunden, da alle miteinander kommunizierenden Anwendungen, die zuvor definierten Elementtypen interpretieren müssen. Dabei treten einige Probleme auf, die im nachfolgenden erklärt werden. Bei der Definition der verschiedenen Dokumente für

einen Anwendungsfall werden meist einige Elemente immer wieder benötigt, sie haben also Stammdatencharakter. Dies umfasst solche Elemente wie für Parteien, also Besteller, Anbieter, Rechnungsadresse, Lieferadresse, Spediteur u.ä., die in B2B-Anwendung häufig anzutreffen sind. Sinnvollerweise lassen sich diese als Reusable bezeichneten Elemente in eine eigene DTD auslagern und können dann in allen Dokumenten wiederverwendet werden. Dabei kann es leicht zu Konflikten bei der Extraktion von Reusables kommen, wie Abbildung 4-5 zeigt. Dabei muss entschieden werden, welche Elemente zu spezifisch sind und damit zu klein, um als Reusable verwendet werden zu können.



Quelle: [36, S. 273]

**Abbildung 4-5: Konflikt bei der Verwendung von Reusables**

Weitere Konfliktpotentiale bei der Standardisierung entstehen bei der Entscheidung bezüglich der Redundanzfreiheit gegenüber dem Interpretationsspielraum sowie der Verwendung eines Wertes als Attribut oder Element.

Bei der Nutzung von Reusables ist es sinnvoll, einige der Subelemente als optional anzugeben, da sie nicht in jedem Dokument gleichermaßen benötigt werden. „Das Problem ist dabei, dass diese Interpretation selbst nicht in XML ausgedrückt werden kann.“ [36, S. 274] Trotz des Interpretationsspielraums gelingt damit eine schnellere Integration, weil die Implementierung der Reusables nur einmal vorgenommen werden muss. Bei dem Verzicht auf Reusables ergeben sich unweigerlich Redundanzen, da die gleichen Subelemente in mehreren Definitionen parallel genutzt werden, d.h. dieselben Subelemente kommen in einem Dokument mehrmals vor. Dadurch wird eine höhere Genauigkeit in der Verwendung des Dokumentes zu Lasten eines größeren Aufwands erreicht.

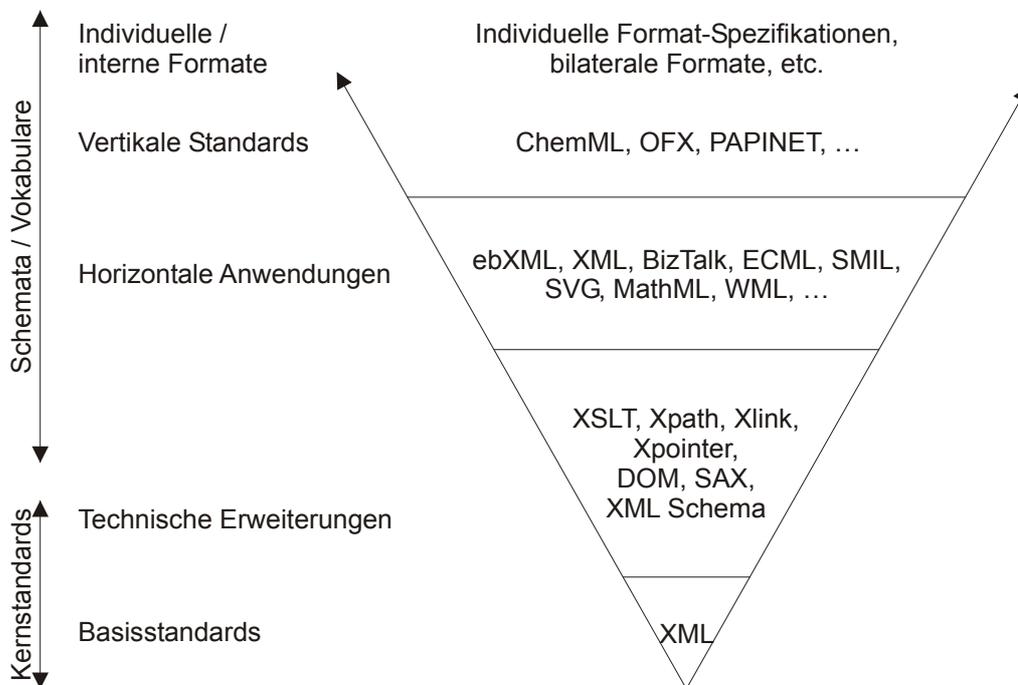
Die Entscheidung, ob ein Wert als Attribut oder Element verwendet wird, muss speziell für jedes Problem getroffen werden. Als Kriterien dafür kann die Reduzierung des

Schreibaufwandes durch Attribute genannt werden oder die Möglichkeit der Einschränkung des Wertebereichs bei Attributen.

Ein weiteres Problemfeld stellt sich durch die nichtexistente Versions- und Dokumentationsverwaltung bei der Erstellung von DTDs vor allem in Entwicklerteams. Der dadurch verursachte Koordinationsaufwand behindert das schnelle Vorankommen der Standardisierung.

### Klassifizierung von XML-Technologien

Nachdem XML als Basisstandard erläutert wurde, gibt Abbildung 4-6 eine Einteilung der darauf aufbauenden Standards. Nach Merz zählen dazu die technischen, die horizontalen und vertikalen Standards sowie die bilateral vereinbarten Sprachen.



Quelle: [36, S. 236]

#### Abbildung 4-6: XML-Kernstandards und Vokabulare

In diesem „großen V“ befindet sich XML an der Basis und steht damit der Standardisierung am nächsten. Danach folgen die technischen Erweiterungen und mit zunehmender Öffnung die Schemata und Vokabulare, wie die horizontalen und vertikalen Standards. Dabei entfernen sich die vertikalen Standards immer mehr von der Standardisierung und schließlich öffnet sich das „große V“ zu den bilateral vereinbarten und produktspezifischen Sprachen. Unter Vokabular wird hier die Menge der genormten Bedeutungen, die auf XML-Elemente und Attribute herunter gebrochen werden kann,

verstanden. [36, S.236] Also erfolgt eine einheitliche Festlegung von Begriffen, die dokumentiert den Standard ergeben.

Die technischen Standards sowie die horizontalen Standards werden für ein erfolgreiches E-Business in der heutigen Zeit verstärkt verwendet und werden daher in Abschnitt 4.2.2 und 4.2.3 ausführlich erläutert.

Kein Gegenstand dieser Arbeit sollen die vertikalen Standards sein, da deren mittlerweile entstandene Menge den Umfang dieser Arbeit sprengen würde. Außerdem handelt es sich dabei um verschiedene XML-Dialekte, meist um größere Einzelunternehmen herum, die den Nachrichtenaustausch innerhalb einer Branche oder einer Supply Chain definieren. Laut [60, S. 77] sind dabei XML-Anwendungen wie z.B. Astronomical Instrument Markup Language, Bio-Polymer Markup Language, CML, FinXML, FpML und OSD, zu nennen sowie weitere Standards aus der Logistik- und Chemiebranche sowie der Papierhersteller.

### **4.2.2 Technische Kernstandards von XML**

Aufbauend auf XML koordiniert das W3C verschiedene andere Standards, die im nachfolgenden als technische Kernstandards bezeichnet werden. Dabei sind diese nicht auf bestimmte Anwendungsbereiche beschränkt, sondern dienen als Grundlage zur Verarbeitung von XML-Dokumenten. Einige ausgewählte Standards werden in diesem Abschnitt vorgestellt.

#### **XPath, XPointer und XLink**

Mit XPath<sup>11</sup> wird eine Syntax zur Formulierung von Referenzen auf einzelne Elemente oder Elementmengen in der Knotenhierarchie eines XML-Dokuments bereitgestellt. Mit XPath-Ausdrücken können Werte von einzelnen Attributen oder Elementen sowie von spezifizierten Bereichen abgefragt werden und somit komplexe Anfragen an XML-Dokumente gestellt werden.

Basierend auf XPath dient XPointer<sup>12</sup> dazu, einzelne Knoten oder Bereiche innerhalb eines XML-Dokuments zu identifizieren. Außerdem lassen sich damit Textbereiche und Sprungstellen innerhalb von Dokumenten definieren sowie Zugriffe bis hinunter auf den einzelnen Buchstaben innerhalb eines Elementes ermöglichen.

---

<sup>11</sup> Siehe auch: <http://www.w3.org/TR/xpath/> und <http://www.w3.org/TR/xpath20/>

<sup>12</sup> Siehe auch: <http://www.w3.org/TR/xptr/>

Aufbauend auf XPath und XPointer bietet XLink<sup>13</sup> uni- und bidirektionale Referenzen, die auch außerhalb des eigentlichen Dokumentes definiert sein können, an. Anders als bei XPointer wird hier auf die Repräsentation der Referenz im Ausgangsdokument Wert gelegt.

### **XSL, XSLT und XSL-FO**

Um XML-Dokumente visualisieren zu können, wurde XSL<sup>14</sup> (Extensible Stylesheet Language) entwickelt. Mit XSL ist es möglich, das gleiche XML-Dokument in unterschiedlichen Anwendungskonzepten unterschiedlich darzustellen. Dies geschieht meistens durch eine Transformation eines XML-Dokuments in andere Syntaxen oder gar anders benannte und organisierte Strukturierungselemente. Mit XSLT<sup>15</sup> (XSL Transformation) steht dazu der entsprechende Standard zur Verfügung. Mit XSLT wird ein Quellbaum in einen Zielbaum transformiert. Um dieses zu erreichen, können verschiedenartige Regeln und Schablonen sowie Funktionen, z.B. für die Manipulation der Reihenfolge von Elementen, definiert werden. Nach Merz gibt es aber einige Einschränkungen von XSLT zu beachten. Die für den Bereich der B2B-Anwendung wichtigsten sind die beiden folgenden:

- Keine 1:N bzw. N:1 Transformation: Eine Zusammenführung von mehreren Dateien zu einer Zieldatei ist in XSLT nicht ohne weiteres möglich.
- Keine Konvertierung von Datenwerten: Mit XSLT kann z.B. keine Änderungen des Datumsformates oder Stringoperationen ausgeführt werden.

Trotz dieser Einschränkungen wird XSLT häufig verwendet. Die am meisten verwendete Transformation ist dabei von XML nach HTML, aber auch jedes beliebige andere Format, wie z.B. RTF, PDF oder gar EDI ist möglich. Dazu legt XSL-FO<sup>16</sup> (XSL Formatting Objects) die Formatierung des Ausgabedokumentes in einem neutralen Seitenbeschreibungsfomat fest. Diese Zwischenrepräsentation bei der Formatierung des Dokumentes ist wichtig, da bei der direkten Transformation von XML nach HTML durch XSL keine Seitenformatierungsbefehle erforderlich sind, aber Sprachen wie PostScript, LaTeX, RTF oder PDF diese erfordern.

---

<sup>13</sup> Siehe auch: <http://www.w3.org/TR/xlink/>

<sup>14</sup> Siehe auch: <http://www.w3.org/TR/xsl/>

<sup>15</sup> Siehe auch: <http://www.w3.org/TR/xslt/> und <http://www.w3.org/TR/xslt20/>

<sup>16</sup> Siehe auch: <http://www.w3.org/TR/xsl/slice6.html#fo-section>

### **XML-Schema**

Im Gegensatz zu den DTDs stellt XML-Schema<sup>17</sup> den ausdrucksstärkeren Ansatz zur Definition von Dokumenttypen zur Verfügung. Aufgrund der Einschränkungen von XML 1.0 DTDs wurden neben XML-Schema verschiedene Schemasprachen, wie SOX<sup>18</sup> (Schema for Object-Oriented XML) und XDR<sup>19</sup> (XML Data Reduced), entwickelt, die heute keine Bedeutung mehr haben. Die wichtigsten Eigenschaften beider Sprachen, wie z.B. die Vererbung zwischen Datentypen bei SOX, wurden nachträglich in XML-Schema übernommen. SOX wurde lange Zeit von Commerce One für die Definition von xCBL entwickelt während Microsoft für das BizTalk-Framework XDR entwickelte.

Aufgrund der detaillierten Spezifikation von Dokumenttypen und Inhaltsmodellen können verschiedene Probleme des B2B gelöst werden, insbesondere kann die Dokumentenverarbeitung weiter automatisiert werden. Durch die Vernetzung nicht nur auf inhaltlicher Ebene sondern auch auf Schemaebene eignet sich XML-Schema besonders als Darstellungsstandard für EDI und führt damit zur Dezentralisierung der Standardisierung. Jeder kann sein eigenes Schema definieren und es für andere zugänglich machen.

Die bis jetzt genannten Standards stellen jeweils eine Erweiterung von XML für verschiedene Anwendungsfälle dar. Nachfolgend werden die dazu benötigten Programmierschnittstellen erläutert.

### **DOM und SAX**

Zur Bearbeitung von XML-Dokumente stehen die Programmierschnittstellen DOM und SAX zur Verfügung. Das Document Object Modell (DOM) definiert die logische Struktur eines XML-Dokuments durch eine Knotenhierarchie und bietet Zugriffs- und Navigationsmöglichkeiten auf die einzelnen Knoten. Es existieren verschiedene Implementierungen von DOM, z.B. in C++ oder Java, da DOM in der programmiersprachenunabhängigen Schnittstellendefinitionssprache IDL definiert wurde. Aber auch in ERP- oder Datenbank-Software-Produkten ist mittlerweile DOM implementiert, wie z.B. in Progress 4GL. Dabei wird i. A. die gleiche API unterstützt und somit kann jedes Anwendungsprogramm in gleicher Weise Knoten erzeugen, ändern, löschen oder darauf zugreifen. Die Verarbeitung des gesamten Dokumentes erfolgt dabei im Hauptspeicher und

---

<sup>17</sup> Siehe auch: <http://www.w3.org/XML/Schema>

<sup>18</sup> Siehe auch: <http://www.xcbl.org/sox/sox.html>

<sup>19</sup> Siehe auch: <http://www.w3.org/TR/1998/NOTE-XML-data/> und <http://msdn.microsoft.com/library/en-us/xmlsdk30/html/xmconxmlschemadevelopersguide.asp>

wird durch einmaliges Parsen desselben erreicht. Dies bietet den Vorteil des effizienten Navigierens durch die Knotenstruktur im Hauptspeicher, begrenzt aber die Größe des XML-Dokuments und stellt damit besondere Anforderungen an die Hardwareausstattung des Rechners.

Im Gegensatz zu DOM bietet die Programmierschnittstelle SAX (Simple API for XML) einen ereignisorientierten Zugriff auf die XML-Elemente an und erlaubt dem Anwendungsprogramm das Abonnieren von bestimmten Elementtypen. Sobald beim Einlesen des XML-Dokuments ein solches Element angetroffen wird, wird dies dem Anwendungsprogramm mitgeteilt, das seinerseits auf dieses spezielle Ereignis reagieren kann. Somit lassen sich mit SAX sehr effizient kontinuierliche Datenströme verarbeiten aber im Gegensatz zu DOM keine Objektbäume erzeugen oder manipulieren, da immer nur das jeweils gerade eingelesene Element bekannt ist und alle anderen sofort wieder verworfen werden.

Diese beiden komplementären Techniken werden idealerweise in Kombination implementiert und bei den neueren Parsern wie z.B. Xeres<sup>20</sup> von Apache oder JAXP<sup>21</sup> (Java API for XML Processing) von Sun erfolgt dies schon.

### 4.2.3 Horizontale Standards

Bei den horizontalen Standards, die nicht branchenspezifisch dafür aber anwendungsnahe sind, unterscheidet Merz zwischen den allgemeinen und den E-Commerce-relevanten Standards. Einige der allgemeinen horizontalen Standards werden nachfolgend vorgestellt, weil sie eine der Grundlagen für Application Integration sind. Danach werden im Abschnitt 4.2.4 die E-Commerce-relevanten Standards beschrieben.

#### SOAP

SOAP<sup>22</sup> (Simple Object Access Protocol) ist eine relativ systemnahe Technologie, die aber Möglichkeiten bereitstellt, XML-Dokumente beliebigen Inhalts über Standardprotokolle im Internet zu übertragen. Es „...wurde von Microsoft als Standardrepräsentation für Parameter und Resultate von Prozeduraufrufen (engl. Remote Procedure Call, RPC) entwickelt.“ [36, S. 262] Inzwischen wurde SOAP beim W3C veröffentlicht und liegt dort als Standard vor.

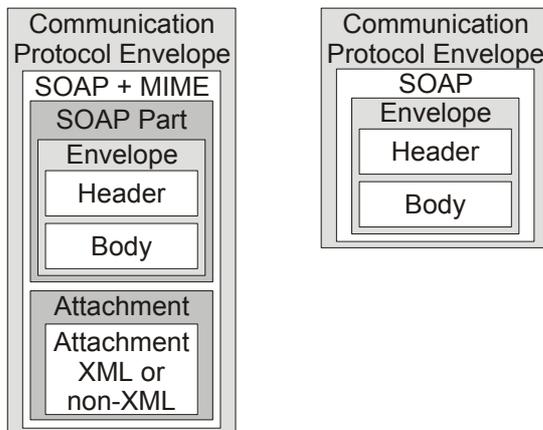
---

<sup>20</sup> Siehe auch: <http://xml.apache.org/>

<sup>21</sup> Siehe auch: <http://java.sun.com/xml/jaxp/index.html>

<sup>22</sup> Siehe auch: <http://www.w3.org/TR/SOAP/>

Ein SOAP-Request besteht aus einem Umschlag, einem sogenannten Envelope, in dem sich ein Header und ein Body befinden. [30, S. 29] Außerdem können diese XML codierten Nachrichten mit und ohne Anhang verschickt werden, wie Abbildung 4-7 zeigt.



Quelle: [22, S. 23]

**Abbildung 4-7: SOAP Message**

Außerdem existieren für SOAP noch Kodierungsregeln, die in SOAP-Encoding definiert werden. Wenn noch für jede Programmiersprache eine Abbildung zwischen den Parameterobjekten und dem SOAP-Encoding entwickelt wird, kann SOAP als universelle Plattform zum Datenaustausch dienen, die nur die Übertragung festlegt, aber den Inhalt der Dokumente offen lässt.

SOAP ist ein Bestandteil der Web Services, die im Abschnitt 4.4.1 beschrieben werden.

### Datenbanken

Da auch im Bereich der Datenbanken XML immer mehr genutzt wird, gibt dieser Abschnitt einen Überblick über die Besonderheiten und stellt am Beispiel von Oracle eine praktische Umsetzung vor.

Zur Speicherung von XML-Dokumenten in Datenbanken gilt es, diese nach datenlastigen oder dokumentenlastigen Inhalt zu unterscheiden.

Die datenlastigen Inhalte stellen kein Problem dar, weil sie direkt aus einer relationalen Datenbank stammen und Produktkataloge beinhalten können. Hier spielt die Reihenfolge der Elemente im Dokument keine Rolle, da sie z.B. in der Datenbank über Schlüssel verknüpft sind. Diese Art von Dokumenten lässt sich mittels einer Abfragesprache extrahieren, die eine Erweiterung für XML besitzt. Microsoft z.B. erweiterte mit MSXML

die Datenbankabfragesprache SQL um weitere Statements, die das Resultat nach XML transformieren. Es können aber auch Dokumente in Datenbanken abgelegt werden.

Bei dokumentenlastigen Inhalten, die meist redaktionellen Ursprungs sind, darf die Reihenfolge der Elemente nicht geändert werden, da sonst die Dokumentmarken ihren Platz verlieren und damit den Sinn des Dokuments entstellen würden. Entweder werden die Fließtext-Elemente als ASCII-BLOB gespeichert oder es werden Zusatzinformationen in der Datenbank abgelegt, die aus den Marken im Fließtext gewonnen werden.

Letztendlich kann durch eine 1:1 Abbildung von Knoten des DOM-Baums auf Datenbankobjekte jedes XML-Dokument ohne Strukturverlust in einer Datenbank abgespeichert werden.

### **Oracle 9i**

Als Beispiel für die Verarbeitung von XML-Dokumenten in relationalen Datenbanken möge Oracle 9i<sup>23</sup> dienen, das mit XML-Dokumenten auf verschiedene Weise umgehen kann. Mit dem iFS (Internet File System) lassen sich Mappings zwischen Datenbankinhalten und XML-Dokumenten festlegen. Des Weiteren lassen sich DB-Schemata aus XML-DTDs erzeugen und damit XML-Dokumente in die Datenbank aufnehmen. Anders herum bietet iFS auch eine DOM-basierte Schnittstelle zu den Inhalten einer Datenbank und ermöglicht zusätzlich noch die Versionierung und Nebenläufigkeitskontrolle.

Weiterhin bietet Oracle mit dem XML SQL Utility for Java die Möglichkeit, Daten zwischen einer objektorientierten Datensicht (SQL3) und XML (DOM) zu transferieren. Letztendlich lassen sich XML-Dokumente auch als BLOB in der Datenbank ablegen.

Zusätzlich zu diesen Möglichkeiten steht der XML Class Generator zur Verfügung, der aus einem Datenbankschema ein JAVA-Package erzeugen kann. Somit kann über eine anwendungsnahe API in dieser Package ein XML-Dokument durch Instanzierung von Objekten auf einfache Art und Weise erzeugt werden ohne dabei auf die DOM-Ebene herabsteigen zu müssen.

Zur Speicherung von XML eignen sich aber auch objektorientierte Datenbanken, wobei die hierarchische Struktur von XML dies favorisieren würde, aber in der Praxis spielen sie

---

<sup>23</sup> Siehe auch: <http://otn.oracle.com/tech/xml/content.html>

derzeit keine Rolle. Neben den OODB haben sich noch reine XML-Datenbanken, z.B. X-Hive/DB<sup>24</sup> oder Tamino<sup>25</sup> von der Software AG, entwickelt.

### **XML-Signature**

Wichtig für E-Business ist eine sichere Datenübertragung sowie eine Authentifizierung der Benutzer und eine Signierung der Dokumente. Mit der Standardisierung von Signaturen und Zertifikaten in XML wird ein Schritt in diese Richtung unternommen. Somit wird es möglich werden, Signaturen zwischen unbekanntenen Parteien auszutauschen oder diese z.B. im E-Business für standardisierte elektronische Schecks zu verwenden.

Interessante Anwendungsbereiche lassen sich erzielen, wenn die zu signierenden Dokumente auch in XML vorliegen. So kann als „Enveloped Signature“ die Signatur selbst in das zu signierende Dokument eingebunden werden oder umgekehrt als „Enveloping Signature“ das Dokument einbinden. [36, S. 260] Ebenfalls denkbar ist, dass sich die Signatur und das zu signierende Dokument selbst in einem Containerdokument nebeneinander befinden und damit die Signatur unabhängig von dem zu signierenden Dokument ist (Detached). Des Weiteren kann sich eine Signatur bei einem XML-Dokument nur auf Teile desselben beziehen oder im umgekehrten Fall mehrere Teildokumente miteinander verbinden.

Mit XML-Signature<sup>26</sup> erfolgt dabei eine Standardisierung durch das W3C und damit eine Grundlage für weitere Anwendungen. Den Erfolg von XML-Signature wird erst die Zukunft zeigen, aber es wird erwartet, dass sich mit zunehmender Durchsetzung von XML als Infrastruktur auch die XML-Signature einen weiten Anwendungsbereich erschließen kann.

#### **4.2.4 Horizontale E-Business Standards**

Nach den allgemeinen Standards werden nun die E-Commerce-relevanten vorgestellt. Diese werden in der Literatur unterschiedlich klassifiziert. Merz unterteilt dabei die horizontalen Standards nicht weiter, während Weitzel u.a. diese in zwei Gruppen gliedert. [60, S. 75ff] Diesem Ansatz folgt auch Brady in ihrer Web-Based Standards Roadmap. [6]

---

<sup>24</sup> Siehe auch: <http://www.x-hive.com>

<sup>25</sup> Siehe auch: <http://www.softwareag.de>

<sup>26</sup> Siehe auch: <http://www.w3.org/TR/xmlsig-core/>

Außerdem haben sich noch Standards zur Produktklassifikation entwickelt. Diese werden nun zuerst beschrieben, da sie von den horizontalen Standards für die Klassifikation genutzt werden.

### **Product Classification**

Ein Vorteil eines elektronischen Marktplatzes besteht in der Verwendung eines lieferantenneutralen, einheitlichen Klassifizierungssystem für Produktdaten. Durch die damit mögliche Suche in dem meist vielfältigen Datenbestand können Produkte schneller und eindeutiger gefunden werden. „Der herstellerübergreifende Ansatz eines Klassifizierungssystems ist zudem Ausdruck der Neutralität des Intermediärs.“ [42, S. 352]

Um dies zu erreichen, müssen folgende zwei Eigenschaften erfüllt sein:

- einheitliche Klassenstruktur
- einheitliches Merkmalssystem

Über eine einheitliche Klassenstruktur können die Artikel systematisch aufgenommen und dargestellt werden. Durch das einheitliche Merkmalssystem werden die Artikel detailliert beschrieben.

Nachfolgend werden die branchenübergreifenden Klassifikationsstrukturen eCl@ss und UN/SPSC und das branchenspezifische ETIM vorgestellt und anschließend bewertet. [21, S. 179]

### **ETIM**

Für die Elektrobranche wurde von den Elektrogroßhändlern in Zusammenwirken mit dem Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO das elektrotechnische Informationsmodell (ETIM)<sup>27</sup> entwickelt.

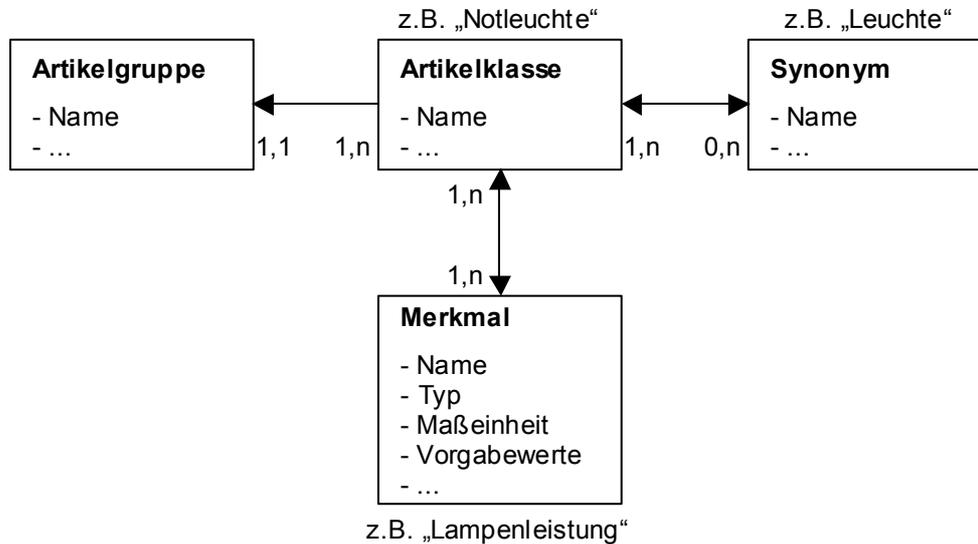
ETIM ist als zweistufige Hierarchie angelegt und beinhaltet Artikelgruppen und -klassen. Mit diesem Ansatz verfolgt ETIM „ein Systematisierungsprinzip, das auf der Verwendung von Synonymen beruht.“ [42, S. 352]

Gleichartige Artikelklassen werden in den Artikelgruppen, die mehr als Oberbegriff für die Artikelklassen zu verstehen sind, zusammengefasst. Im Gegensatz zu den Artikelgruppen können den Artikelklassen technische Merkmale zugeordnet werden. [21, S. 186] Jede

---

<sup>27</sup> Siehe auch: <http://www.etim.de>

Artikelgruppe beinhaltet gleichartige Produkte, die aber auch unterschiedlich und von verschiedenen Herstellern sein können. Die Klassenstruktur von ETIM kann Abbildung 4-8 entnommen werden.



Quelle: Nach [42 und 56]

**Abbildung 4-8: Klassenstruktur von ETIM**

### eCl@ss

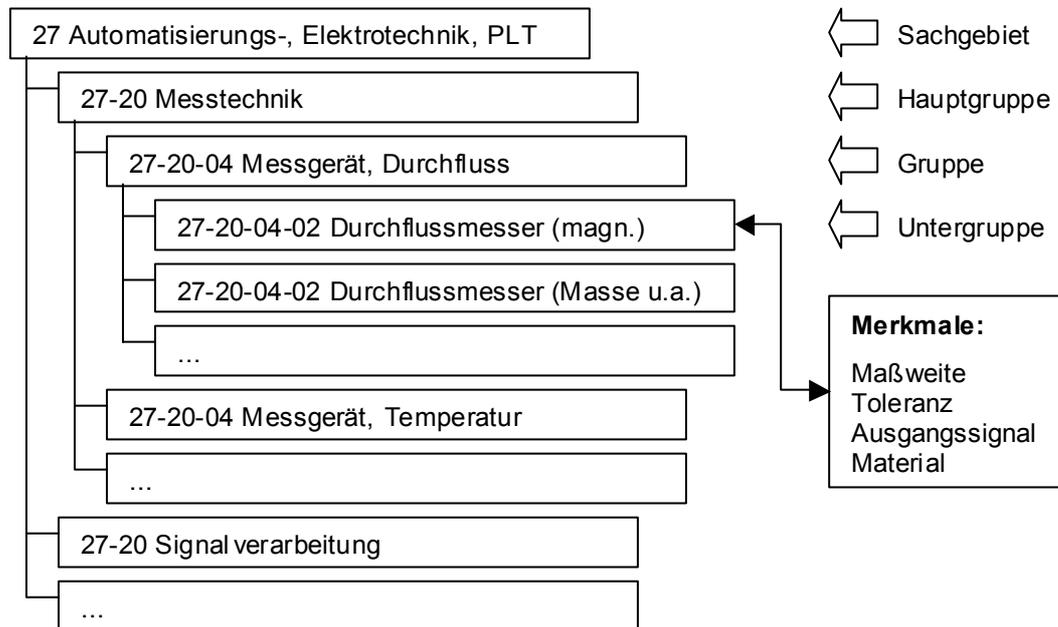
Eine weitere Produktklassifikation ist eCl@ss, das am Institut der deutschen Wirtschaft Köln entwickelt wird, und sich den Anspruch gestellt hat, Transaktionskosten sowohl für die liefernde wie auch die beschaffende Unternehmen zu senken.

In den 4 hierarchischen Ebenen (Sachgebiet, Hauptgruppe, Gruppe und Untergruppe) werden die Klassen durch zweistellige Nummernschlüssel gekennzeichnet. [21, S. 180] Diese beginnen in der ersten Hierarchieebene mit dem Nummernschlüssel 20 für „Verpackung“ und enden mit dem Nummernschlüssel 40 für „Arbeitssicherheit, Unfallschutz“. Merkmale können erst in der 4. Ebene den Klassen zugeordnet werden. Bei eCl@ss wird zwischen

- Basismerkmalsleisten und
- Standardmerkmalsleisten unterschieden.

Allen Produkten können Basismerkmale, wie z.B. für Hersteller und Produktname, zugewiesen werden. Spezifische Eigenschaften von bestimmten Produkten können mit den Standardmerkmalen, z.B. Nennfrequenz, abgebildet werden. Weiterhin können den Klassen Schlagwörter zugeordnet werden, wodurch eine bessere Suchmöglichkeit geschaffen

wurde. Das Konzept wird an dem Beispiel für das Sachgebiet 27 „Automatisierungs-, Elektro- und Prozessleittechnik“ in Abbildung 4-9 dargestellt.



Quelle: [42]

**Abbildung 4-9: Klassenstruktur von eCI@ss**

### UN/SPSC

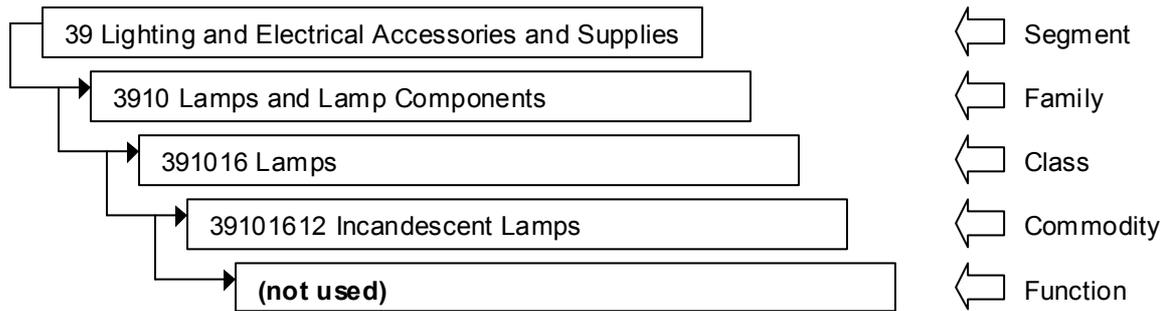
Der United Nations Standard Products and Services Code (UNSPSC)<sup>28</sup> wird von der Electronic Commerce Code Management Association (ECCMA) verwaltet. Er ist durch die Verschmelzung der Klassifizierungsbestrebungen des United Nations Development Program und des amerikanischen Unternehmens Dun & Bradstreet entstanden.

UN/SPSC definiert die folgenden 5 Hierarchieebenen, von denen nur die ersten 4 benutzt werden müssen:

- Segment
- Familie
- Klasse
- Warengruppe
- Funktion

Optional kann die 5. Ebene Funktion innerhalb von Organisationen vergeben werden. Ein Beispiel zu UN/SPSC demonstriert Abbildung 4-10.

<sup>28</sup> Siehe auch: <http://www.un-spssc.net/>



Quelle: [42]

**Abbildung 4-10: Klassenstruktur von UN/SPSC**

### **Andere Klassifizierungsansätze**

Von untergeordneter Bedeutung sind die Ansätze NAMUR und NAPCS.

In Deutschland gibt die Normierungsarbeitsgemeinschaft für Mess- und Regelungstechnik in der chemischen Industrie (NAMUR)<sup>29</sup> nur Handlungsempfehlungen aber keinen Klassifizierungsstandard heraus. In Zukunft möchte NAMUR mit eCl@ss zusammenarbeiten.

Mit dem North American Product Classification System (NAPCS)<sup>30</sup> laufen derzeit in den USA Entwicklungen, eine Verfeinerung des North American Industry Classifications System (NAICS) auf der Produkt- und Dienstleistungsebene zu erlangen. Nach Plan soll das Klassifizierungssystem in diesem Jahr veröffentlicht werden.

### **Bewertung der Product Classification**

Klassifizierungssysteme können nach folgenden 5 Merkmalen bewertet werden:

- Anzahl der Hierarchieebenen
- Merkmalsystem
- Branchenbezogenheit
- Geographische Ausrichtung
- Funktionale Ausrichtung

Wichtig ist eine gut untergliederte Hierarchie, da durch eine höhere Anzahl von Hierarchieebenen die einzelnen Klassen feiner bestimmt werden können. Basis- und Standardmerkmale des Merkmalsystems beschreiben jeden Artikel genauer. Dabei sind

---

<sup>29</sup> Siehe auch: <http://www.namur.de>

<sup>30</sup> Siehe auch: <http://www.census.gov/eos/www/napcs/napcs.htm>

Basismerkmale kaufmännische Stammdaten, wie z.B. Artikelbeschreibung und EAN, die die Grundeigenschaften jedes Artikels näher bestimmen. Im Gegensatz dazu beschreiben die Standardmerkmale technische Stammdaten, wie z.B. Spannungsintervalle oder Abmessungen, die innerhalb einer Artikelklasse eindeutig sind. Die Einsetzbarkeit eines Klassifizierungssystems wird durch seine Ausrichtung auf die Branche und seine Internationalität bestimmt. Unter funktionaler Ausrichtung versteht man die Begrenzung des Klassifizierungssystems auf einen bestimmten Betriebsbereich, wie z.B. auf den Beschaffungssektor.

Einen direkten Vergleich der wichtigsten Klassifizierungsansätze ETIM, eCl@ss und UN/SPSC bietet Tabelle 4-2.

<b>Merkmal</b>	<b>ETIM</b>	<b>eCl@ss</b>	<b>UN/SPSC</b>
Anzahl der Hierarchieebenen	1 (2)	4	4 (5)
Vollständigkeit der Merkmale	+/-	+/-	-
Branchenunabhängigkeit	-	+/-	+
Internationale Anwendbarkeit	-	-	+
Bereichsübergreifende Anwendbarkeit	+	+	+

Quelle: [42, S. 356]

**Tabelle 4-2: Bewertung von Klassifikationsstandards**

Hervorzuheben ist die herstellerübergreifende Suchfunktion bei ETIM, die durch die Verwendung von Synonymen ermöglicht wird. Dabei verzichtet ETIM im Gegensatz zu eCl@ss und UN/SPSC auf eine hierarchische Gliederung. Diese ist bei eCl@ss durch Nummernkreise realisiert, die mit ihren zweistelligen Nummernschlüsseln sehr starr angelegt sind und somit z.B. nachträgliche Einfügungen erschweren. Von manchen Anwendern wird aber dieses Nummernsystem bevorzugt, da durch den häufigen Gebrauch die Identifizierung von Klassen anhand des Nummernschlüssels beschleunigt wird. ETIM und eCl@ss verfügen über ein Merkmalsystem, dessen Anwendbarkeit eingeschränkt ist, weil ETIM auf die Elektrobranche ausgerichtet ist und bei eCl@ss die Hierarchie in einigen Bereichen bereits in der zweiten Ebene endet. Als einziger Standard ist UN/SPSC international anwendbar und branchenunabhängig.

### **Functions**

Nun werden als weiterer Bereich der horizontalen Standards die Functions vorgestellt, die als Vorlagen für bestimmte Geschäftsprozesse über Branchengrenzen hinaus definiert

werden. Dabei fallen die Ähnlichkeiten zu den EDI-Nachrichten auf, wie z.B. Purchase Order- und Invoice-Nachrichten. Zu den Functions lassen sich folgende Standards zählen: BMEcat/openTRANS, gXML, ICE und xCBL.

### **BMEcat / openTRANS**

Ein im deutschsprachigen Raum weit verbreiteter Standard für den elektronischen Produktdatenaustausch ist BMEcat<sup>31</sup>. Zu diesem werden die entsprechenden Geschäftsdokumente im Standard openTRANS<sup>32</sup> entwickelt. Beide Standards werden gemeinsam vom eBusiness Standardization Committee sowie dem Bundesverband Materialwirtschaft, Einkauf und Logistik e. V. (BME) und der Fraunhofer IAO und Universität Essen bli entwickelt. BMEcat liegt derzeit in der Version 1.2 vom 27. März 2001 und openTRANS in der Version 1.0 vom 7. September 2001 vor. Es ist geplant, beide Standards in der Version 2.0 zusammenzuführen. Dem eBusiness Standardization Committee gehören neben dem BME, dem Fraunhofer IAO und der Universität Essen bli viele Unternehmen, wie z. B: Bayer, BMW, Deutsche Bahn, Deutsche Telekom und Lufthansa, an.

BMEcat ist ein offener Standard für Produktkatalogdatenaustausch, wohingegen openTRANS ein offener Standard für Geschäftsprozesse ist.

### **gXML**

CommerceDesk bietet mit Guideline XML (gXML)<sup>33</sup> einen inhaltlich identischen XML-Standard zu EDIFACT- und ANSI X12-Nachrichten. Unterstützt durch RosettaNet sollen existierende EDI-Implementierungen schnell und einfach nach gXML migriert werden können. Dabei wird aber nur der strukturelle Aufbau der Nachricht nicht aber der Austausch der Nachrichten definiert.

### **ICE**

Von der Grapic Communication Association wird das Information and Content Exchange Protocol (ICE)<sup>34</sup> entwickelt. Mitglieder dieser Organisation sind z.B. 3path, Adobe, Cablevision, Vignette und Oracle. ICE ist selbst Mitglied bei dem W3C und dem eCo Framework.

---

<sup>31</sup> Siehe auch <http://www.bmecat.org>

<sup>32</sup> Siehe auch <http://www.opentrans.org>

<sup>33</sup> Siehe auch <http://www.edifecs.com>

<sup>34</sup> Siehe auch <http://www.icestandard.org>

Besonderer Wert wird dabei auf die automatisierte Versendung von der richtigen Information an den richtigen Kunden gelegt. Unternehmen sollen durch das ICE-Protokoll<sup>35</sup> in die Lage versetzt werden, Inhalte zu produzieren und diese effizient mit Geschäftspartnern auszutauschen [36, S. 579]. Es wird lediglich das Übertragungsprotokoll vorgegeben, die Nutzdaten selber können jede beliebige XML-DTD sein. Somit ist ICE vom Inhalt und Format, das zu übertragen ist, unabhängig.

### **xCBL**

Die Common Business Library (CBL) wird heute unter dem Namen xCBL<sup>36</sup> von Commerce One weiterentwickelt. Wichtige Mitglieder von CommerceNet sind Microsoft BizTalk, OASIS, eCo Framework Project and Working Group.

Die Library stellt eine Sammlung von XML-Spezifikationen und -Modulen für Geschäftsdokumente zur Verfügung, wie z.B. für Bestellungen, Verfügbarkeitsprüfungen, Preisfragen, Rechnungen und Produktkataloge. Bei xCBL wird zwischen dem Inhalt eines XML-Dokumentes und dessen Transport strikt unterschieden. Somit ist xCBL nicht an eine bestimmte Transportmethode gebunden. Bei der Umsetzung der Spezifikation wurden Inhalte aus EDIFACT, besonders mit seinen Subsets, EANCOM und SIMPL-EDI, berücksichtigt. [60, S.135f]

### **Frameworks**

Die hier genannten Frameworks unterscheiden sich von denen im Abschnitt 4.4 dadurch, dass sie auf den Austausch von xml-basierten Standards beruhen.

Unter Frameworks versteht man Basistechnologien, deren Fokus viel mehr auf dem gesamten Kommunikationsprozess als auf die bloße semantische Spezifikation von bestimmten Nachrichtentypen ausgerichtet ist [60, S. 76]. Durch die Frameworks werden Infrastrukturen bereitgestellt, um beliebige Geschäftsdokumente automatisch auf XML-Basis verarbeiten zu können. Die derzeit wichtigsten Initiativen sind BizTalk, cXML, ebXML, eCo, OAGIS, RosettaNet und XML/EDI Group.

### **BizTalk**

Auf der Grundlage des Standards XML wird von Microsoft BizTalk<sup>37</sup> entwickelt, dabei wird es von den Größten der Standardsoftware, wie z. B. SAP, Peoplesoft, Baan, J.D.

---

<sup>35</sup> Siehe auch <http://www.w3.org/TR/NOTE-ice>

<sup>36</sup> Siehe auch <http://www.xcbl.org>

Edwards & Company sowie dem großen Anbieter für Internetsoftware CommerceOne unterstützt. Der Grundgedanke von BizTalk besteht im Aufbau eines Framework für Geschäftsdokumente und der Integration verschiedener E-Business-Anwendungen. BizTalk selber besteht aus 3 Komponenten, dem BizTalk-Framework, der BizTalk Schema Library und dem BizTalk Server.

„Das BizTalk Framework definiert eine Menge von XML Formatvorschriften und Elementen (...), die jede ‚BizTalk Message‘ berücksichtigen muss“ [60, S. 81]. Die Bibliothek (Library) selber enthält eine Sammlung von XML-Schemas, die auf dem Portal [www.biztalk.org](http://www.biztalk.org) hinterlegt sind. Das Herzstück von BizTalk ist der BizTalk Server 2000, der als Nachrichtenserver fungiert. Zudem beinhaltet er auch weitere Funktionen, wie das Mapping und die Möglichkeit Entscheidungsregeln festzulegen.

Der Austausch der Daten findet über das BizTalk Dokument statt, das aus einem wohlgeformten XML-Dokument und BizTags besteht. Wichtig ist nur dabei, dass die kommunizierenden Unternehmen sich auf ein BizTalk Schema einigen.

BizTalk kann aber auch in Abschnitt 4.4 eingeordnet werden, da es mit dem BizTalk-Server in Verbindung zu .NET steht.

### **cXML**

Ariba Technologies versucht mit Commerce XML (cXML)<sup>38</sup> kleine und mittelständische Unternehmen besser in den Beschaffungsprozess einzubinden. Wichtige Mitglieder sind z.B. Ariba, Intershop, Microsoft, Philips, Vignette und VISA.

Commerce XML wurde mit dem Ziel entwickelt, EDI-Nachrichten auf der Basis von XML zu standardisieren. Dabei wurde auf die flexible Erweiterbarkeit der Basis-DTDs der Nachrichten geachtet [36, S. 700]. „Basis der cXML-Spezifikation sind Kataloge und Purchase Orders.“[60, S. 122] Neben diesen Dokumenten gibt es noch weitere, jedoch wurden die Branchen Transport und Finanzen nicht eingebunden. cXML wurde mittlerweile in das BizTalk-Framework mit aufgenommen.

### **ebXML**

Electronic Business XML (ebXML)<sup>39</sup> wurde 1999 von der Organisation for the Advancement of Structured Information Standards (OASIS)<sup>40</sup> sowie der UN/Cefact ins

---

<sup>37</sup> Siehe auch <http://www.biztalk.org>

<sup>38</sup> Siehe auch <http://www.cxml.org>

Leben gerufen, um eine offene technische Spezifikation für den globalen Austausch von elektronischen XML Geschäftsdaten zu schaffen. Dabei setzen sich die Entwickler das Ziel, E-Business auch für kleine und mittelständische Unternehmen sowie Entwicklungsländer erschwinglich zu gestalten. Die Initiative wird von SUN, Commerce One, IBM, RosettaNet und vielen anderen unterstützt, wobei sich als einziger Microsoft nicht beteiligt.

Die Spezifikation soll vollständig kompatibel zu dem W3C Standard sein und bei anerkannten Standardisierungsgremien als internationaler Standard eingereicht werden.

„Als Kernstück von ebXML gelten allgemein Repositories, in denen alle Informationen hinterlegt werden, die Business-Partner benötigen, um Geschäfte elektronisch abzuwickeln.“ [53] Darin werden die Geschäftsprozesse mit der Unified Modeling Language (UML) abgelegt.

Zur Versendung der ebXML Dokumente wird inzwischen Simple Object Access Protocol (SOAP) unterstützt, das aber von ebXML erweitert wurde, da es keine Möglichkeit bot, binäre Daten zu verschicken. Dabei wird XML und MIME genutzt.

### **eCo Framework Working Group**

Das eCo Framework<sup>41</sup> wurde ursprünglich von CommerceNet zwischen 1994 und 1999 entwickelt. Seit ca. 1998 schlossen sich CommerceNet und VeoSystems zur eCo Framework Working Group<sup>42</sup> zusammen. Zu den Mitgliedern zählen z.B. IBM, ICE, Microsoft, RosettaNet und Vignette.

Das Ziel dieser Referenzarchitektur ist die Zusammenführung unterschiedlicher, auch proprietärer, E-Commerce-Anwendungen. Gleichzeitig wird eine Verbindung von komplementären Spezifikationen, wie CBL, EDI, ICE, OBI und XML angestrebt. Die eCo-Architektur basierte anfänglich auf einem CORBA-Ansatz, wurde aber um MOM und XML-Nachrichten erweitert.

Die eCo-Architektur wird als Ebenenmodell mit 7 Schichten wie in Abbildung 4-11 dargestellt. „Diese Schichten sollen alle Aspekte abdecken, die erforderlich sind, um eine Geschäftsbeziehung zwischen zwei Unternehmen aufzubauen.“ [36, S. 673] Jede Schicht

---

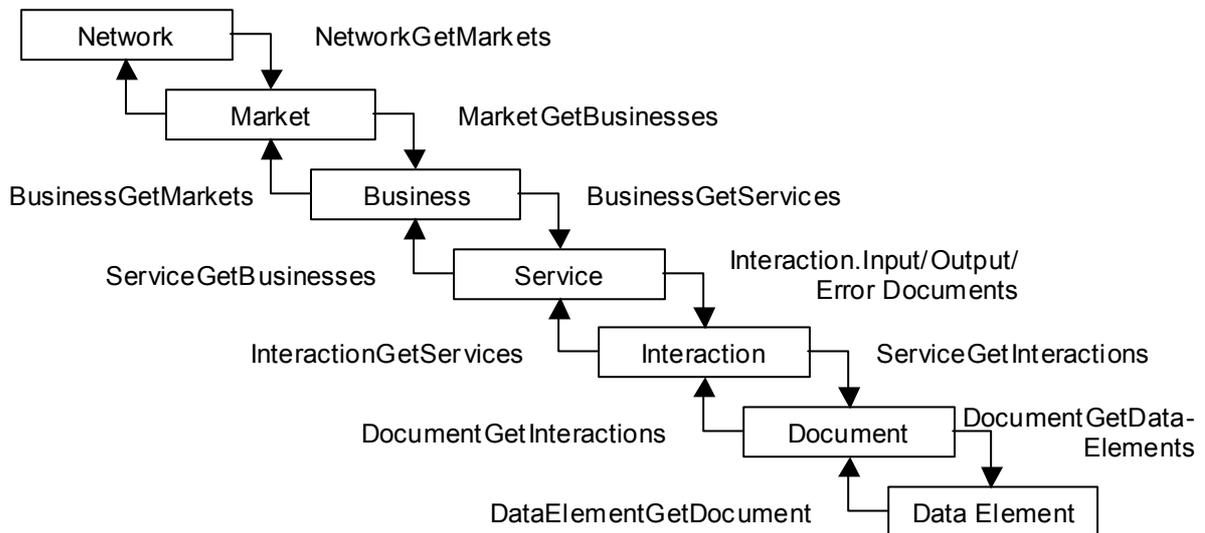
<sup>39</sup> Siehe auch <http://www.ebxml.org>

<sup>40</sup> Siehe auch <http://www.oasis-open.org/>

<sup>41</sup> Siehe auch <http://eco.commerce.net>

<sup>42</sup> Siehe auch <http://www.commerce.net/projects/currentprojects/eco/>

hält Informationen über die jeweils darüber und darunter liegende Schicht vor. So fasst z.B. die Netzwerkschicht mehrere Marktplätze zusammen und die Marktplatzschicht alle Marktplatzobjekte. Damit können jetzt Beziehungen zwischen den Schichten, wie z.B. >>Netzwerk, gib mir alle Marktplätze<< oder aber >>Marktplatz, gib mir dein zugehöriges Netzwerk<<, gezogen werden.



Quelle: [36, S. 675]

**Abbildung 4-11: Ebenenmodell der eCo-Architektur**

Neben den Schichten werden Metainformationen zu jeder Schicht in sogenannten Type Registries abgelegt. Diese „dienen zur Katalogisierung von Dokumentenarten und deren Beziehungen untereinander.“ [60, S. 101]

### RosettaNet

Im Juni 1998 wurde RosettaNet<sup>43</sup> als unabhängiges Non-Profit-Konsortium gegründet. Mittlerweile sind daran über 400 Unternehmen, wie z. B. Microsoft, SAP, IBM und Cisco Systems, aus der IT-Branche beteiligt.

Die Zielsetzung ist die Entwicklung einer gemeinsamen offenen Sprache zur Abwicklung elektronischer Geschäftsprozesse. [15, S. 115] RosettaNet definiert folgende vier Elemente:

- Business Dictionary
- Technical Dictionary
- Implementation Framework (RNIF)

---

<sup>43</sup> Siehe auch <http://www.rosettanet.org>

- Partner Interface Processes (PIPs)

Die Dictionary definieren die Eigenschaften für Produkte, Partner und Geschäftstransaktionen. In dem Implementation Framework werden Protokolle für den Austausch definiert. Mit den PIPs sollen die Geschäftsprozesse aller Supply-Chains der Electronic-Components und Information-Technology-Unternehmen abgebildet werden. [60, S. 128]

### **Sonstige**

Die XML/EDI Group<sup>44</sup> wurde 1997 gegründet. Mittlerweile gehören ihr Firmen wie z.B. Sun, Microsoft, Poet und IBM an.

Ziel des veröffentlichten XML/EDI-Frameworks ist die Überführung von traditionellen EDI in XML-Dokumente. Dabei wurde vor allem Wert auf Rückwärtskompatibilität gelegt. Die Grundlage dieses Frameworks sind die 5 Basistechnologien: XML, EDI, Templates, Agenten und Repositories [44]. Zur Anwendung könnten diese Konzepte im inner- und zwischenbetrieblichen Datenaustausch, Dokumentenmanagement und Data Warehouses kommen.

Als Non-Profit-Industriekonsortium wurde die Open Applications Group (OAG) 1995 von weltweit führenden Softwareunternehmen gegründet. Zu ihr gehören z.B. Microsoft, Oracle, SAP, IBM, J.D.Edwards und PeopleSoft.

Mit der Open Applications Group Integration Specification (OAGIS)<sup>45</sup> wird das Ziel verfolgt, eine einfachere, schnellere und billigere Integration von Softwareprodukten zu erreichen. Dies wurde durch die Entwicklung von Vorlagen für Schnittstellen erzielt. Dabei wurden allgemeine, aber auch spezielle Vorlagen für Geschäftsprozesse entworfen. Es wurden Entwürfe für die innerbetriebliche sowie für die horizontale und vertikale Kommunikation definiert. Diese komplette Beschreibung eines Informationsflusses wird als End to End Integration bezeichnet.

Die Integration dieses mächtigen Frameworks in andere wie z.B. RosettaNet, BizTalk und ebXML wird durch die Mitgliedschaft der OAG in diesen Frameworks unterstützt.

---

<sup>44</sup> Siehe auch <http://www.xmledi-group.org>

<sup>45</sup> Siehe auch <http://www.openapplications.org>

### 4.3 Message-orientierte Middleware

In diesem Abschnitt wird Message-orientierte Middleware (MOM) beschrieben, die eine der Grundlagen für EAI darstellt. Über MOM können unter anderem die zuvor genannten E-Business-Formate ausgetauscht werden und damit eine einfache Integration ins Unternehmen oder über dessen Grenzen hinweg erreicht werden. Dazu wird als Erstes auf den Begriff des Messaging und anschließend auf den einzigen existierenden Standard eingegangen.

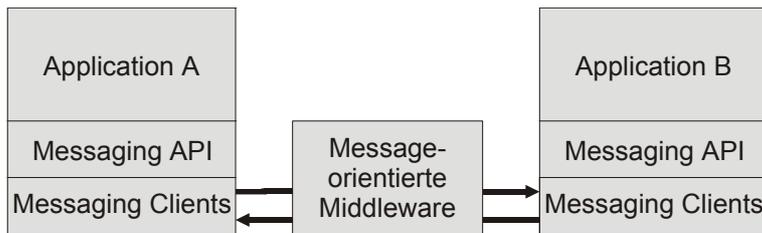
#### 4.3.1 Messaging

„Message-based middleware was one of the first technologies to develop the concept of a common communications mechanism allowing applications to exchange data across the network without having to know the basic primitives.“ [52, S. 31] Mit dieser Beschreibung erfasst Serain das Wesen von Message-orientierter Middleware genau.

#### Asynchrone vs. synchrone Kommunikation

Kommunikation zwischen zwei Prozessen kann sowohl synchron als auch asynchron erfolgen. Synchron steht für eine ständige Verbindung, in der beide Partner online sein müssen und Nachrichten nur dann ausgetauscht werden können, wenn keiner von beiden ausfällt. Dabei wartet der Sender bis er eine Nachricht vom Empfänger erhalten hat. Anders dagegen bei der asynchronen Kommunikation, bei der der Sender seine Arbeit sofort wieder aufnehmen kann, ohne dass er auf eine Nachricht vom Empfänger warten muss.

In [39, S. 2] wird die besondere Bedeutung der asynchronen Kommunikation herausgestellt: „A key concept of enterprise messaging is messages are delivered asynchronously from one system to others over a network.“ [39, S. 2] Dabei zeigt Abbildung 4-12 das Vorgehen einer Anwendung beim Versenden einer Nachricht an eine andere. Dazu benutzt die Anwendung A ein einfaches API, welches die Nachricht an einen Messaging Server übergibt, der sich um die Zustellung der Nachricht an einen oder mehrere Empfänger kümmert.



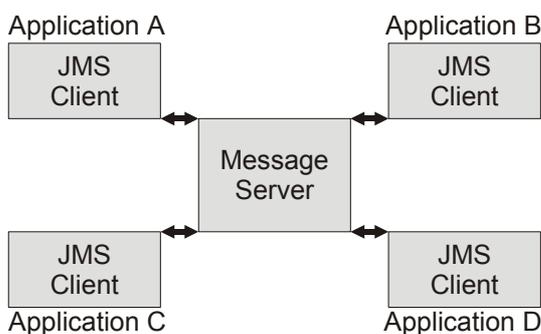
Quelle: [39, S. 3]

**Abbildung 4-12: Message-orientierte Middleware**

### Architektur

Ein Überblick über die verschiedenen Architekturen eines Messaging-Systems kann man [39, S. 3ff] entnehmen. Dabei wird zwischen der zentralisierten oder dezentralisierten Architektur sowie einer Kombination aus beiden unterschieden.

Die zentralisierte Architektur wird meist als hub-and-spoke bezeichnet und verwendet zur Kommunikation einen Message Server, der auch Broker bzw. Router genannt wird, wie in Abbildung 4-13 gezeigt. Dabei kommuniziert jeder Client meist mittels des TCP/IP Protokolls über den Message Server mit den anderen Clients. Dies bewirkt eine Entkoppelung der sendenden und empfangenden Clients wodurch das problemlose Entfernen oder Hinzufügen von Clients möglich wird. Durch die zentralisierte Architektur wird die Anzahl der Netzverbindungen gering gehalten und durch den Einsatz von mehreren Message Servern lässt sie sich in einem Cluster gut skalieren.

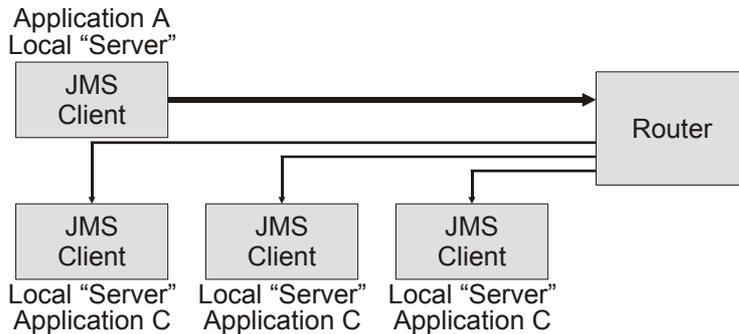


Quelle: [39, S. 4]

**Abbildung 4-13: Zentralisierte Architektur – Hub-and-Spoke**

Im Gegensatz dazu wird bei der dezentralisierten Architektur kein zentraler Server verwendet, sondern die benötigte Funktionalität, wie Persistenz, Transaktionen, Sicherheit usw., wird als lokaler Server in jedem Client integriert. Wie in Abbildung 4-14 dargestellt wird, verwenden derzeitige dezentrale Architekturen das IP Multicast Protokoll und überlassen diesem die Verteilung der Nachrichten. Typisch für Multicast ist, dass

Anwendungen Multicast-Gruppen beitreten können und dann alle Nachrichten erhalten, die an diese Gruppe gesendet werden.



Quelle: [39, S. 5]

**Abbildung 4-14: Dezentralisierte Architektur – IP Multicast**

Bei der hybriden Architektur werden beide Konzepte vereint. Eine denkbare Variante wäre, dass sich die Clients zentralisiert mit einem dedizierten Server verbinden und dieser Server mit anderen Servern dezentral kommuniziert.

### 4.3.2 JMS

Wie zuvor erwähnt, existierte kein Standard für Message-orientierte Middleware, deswegen wurde auf Anregung der Industrie JMS (Java Message Service) entwickelt. Dabei übernahm Sun Microsystems die Führung.

JMS<sup>46</sup> ist ein API (Application Program Interface), aber kein eigenständiges Messaging System. Damit wird der Zugriff auf die MOM abstrahiert und Anwendungen zwischen verschiedenen MOM-Systemen portabel. Mit JMS steht eine robuste Spezifikation zur Verfügung, die einfach und flexibel genug ist, um Messages zwischen Anwendungen auszutauschen. Außerdem bietet sie eine Vielzahl von Nachrichtenübertragungsdefinitionen.

#### JMS Messaging Modelle

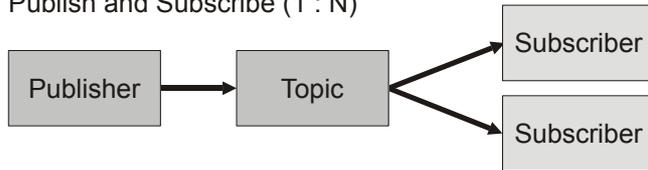
Die JMS Spezifikation sieht zwei Messaging Modelle vor, die dort als Messaging Domains bezeichnet werden. Dabei handelt es sich um publish-and-subscribe sowie um point-to-point, die in Abbildung 4-15 illustriert werden. Des Weiteren definiert die JMS Spezifikation noch die Begriffe des JMS Clients und des JMS Provider. Letzterer ist das Messaging System und ersterer ist der Messaging Client. Mehrere JMS Clients und i.a. ein JMS Provider bilden die JMS Application. Ein JMS Client kann Nachrichten sowohl

---

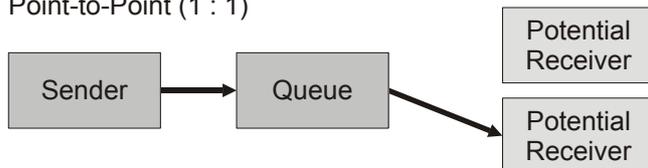
<sup>46</sup> Siehe auch: <http://java.sun.com/products/jms/docs.html>

empfangen als auch versenden, was in JMS Terminologie als Consumer und Producer bezeichnet wird.

Publish and Subscribe (1 : N)



Point-to-Point (1 : 1)



Quelle: [39, S. 7]

**Abbildung 4-15: JMS Messaging Domains**

Das publish-and-subscribe Modell ist ein push-based Modell, in dem ein Producer Messages an mehrere Consumer über einen virtuellen Kanal, in der JMS Terminologie Topic genannt, versendet. Zum Empfangen einer Nachricht müssen sich die Consumer mit diesem Topic verbinden (subscribe), dabei erhalten alle mit einem Topic verbundenen Consumer eine Kopie jeder an dieses Topic gesendete Nachricht. Damit erklärt sich auch das push-based Modell, da jeder Consumer, der Nachrichten abonniert hat, diese automatisch zugestellt bekommt ohne ständig selbst nach neuen Nachrichten fragen zu müssen.

Weiterhin ist es nicht notwendig, dass beim Versenden einer Nachricht der Empfänger aktiv ist. Dabei können Nachrichten verloren gehen, was aber bei bestimmten Anwendungsfällen zu Gunsten einer kürzen Reaktions- und Laufzeit erwünscht ist. Im Gegensatz dazu können aber auch durable subscriptions verwendet werden, bei denen der Client offline gehen kann und sobald er wieder online ist, alle an die von ihm abonnierten Topics gesendeten Nachricht automatisch zugestellt bekommt.

Das point-to-point-Modell ist im Gegensatz zum publish-and-subscribe-Modell ein polling-basiertes Modell, in dem die Nachrichten nicht automatisch durch den Server ausgeliefert werden, sondern jeder Client diese erfragen muss. Einen Wechsel zu einem push-basierten Modell sieht JMS aber vor.

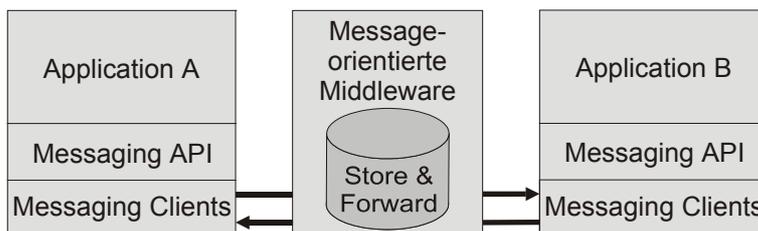
Der Nachrichtenversand erfolgt über einen virtuellen Kanal, genannt Queue. Dies kann asynchron oder synchron erfolgen, aber jede Nachricht darf von einem Empfänger nur

einmal verarbeitet und auch nur von diesem verarbeitet werden. Es können aber durchaus mehrere Consumer die Queue abfragen, aber nur einer bekommt die aktuelle Nachricht. Dieses Prinzip wird von der JMS Spezifikation nicht zwingend vorgeschrieben, aber von vielen Anbietern als load balancing implementiert.

Als besondere Eigenschaft verfügt nur das point-to-point-Modell über die Möglichkeit, anstehende Nachrichten in der Queue noch vor der Verarbeitung zu betrachten und darüber auszuwählen, welche Nachrichten denn abgeholt werden sollen.

### Garantie, Sicherheit und Administration

Ein weiteres Feature von JMS ist die garantierte Nachrichtenzustellung. Wie Abbildung 4-16 zeigt, wird in der MOM dazu jede Nachricht zwischengespeichert, wenn sie nicht zugestellt werden kann. Dies kann der Fall sein, wenn ein Netzwerkfehler oder anderer Grund vorliegt. Sobald der Client wieder erreichbar ist, werden die bis dahin aufgefangenen Nachrichten an ihn ausgeliefert. Somit garantiert JMS, dass jede Nachricht 100-prozentig ihren Empfänger erreicht.



Quelle: [39, S. 16]

**Abbildung 4-16: Garantierte Nachrichtenzustellung**

Während die Zustellungsgarantie ein Feature von JMS ist, wird zur Sicherheit in der Spezifikation nur erwähnt, dass es wünschenswert wäre, aber jeder Hersteller dies selbst zu implementieren hat. Ebenso ist eine Administrationsschnittstelle herstellerspezifisch, da auch ganz verschiedene Architekturen verwendet werden.

### 4.3.3 JMS Produkte

Nachdem die Prinzipien von JMS erläutert wurden, erfolgt jetzt ein Überblick über die am Markt befindlichen Produkte und später im Kapitel 5 wird speziell auf den verwendeten SonicMQ-Broker eingegangen.

### **MQSeries**

Der Marktführer bei den Messaging Systemen ist IBM mit dem Produkt MQSeries<sup>47</sup>. Dieses ist integriert in IBM WebSphere Foundation und lauffähig auf einer Vielzahl von Plattformen. Beide Architekturen, sowohl zentralisiert als auch dezentralisiert, sind in diesem Produkt vereint und es werden beide Messaging Modelle unterstützt.

### **SonicMQ**

Mit SonicMQ<sup>48</sup> steht ein Produkt von Sonic Software, eine Abspaltung von Progress, zur Verfügung, das voll in JAVA implementiert und als erster Message Server J2EE zertifiziert wurde. Beide Messaging Modelle können genutzt werden, außerdem ist als spezielles Feature die DRA (Dynamic Routing Architektur) verfügbar. Diese bietet die Möglichkeit große und skalierbare B2B Infrastrukturen aufzubauen. Ein weiteres Merkmal ist die Erweiterung der TextMessage von JMS um die XMLMessage, die XML Nachrichten entweder als Text oder als DOM-Baum zugreifbar macht.

### **Weitere Produkte**

Weitere Produkte bieten Sun mit iPlanet Message Queue for Java<sup>49</sup>, Fiorano mit FioranoMQ<sup>50</sup>, Softwired AG mit iBus<sup>51</sup> und BEA mit WebLogic Server<sup>52</sup> an. Mit OpenJMS<sup>53</sup> bietet dabei die ExoLab Group eine Open Source Implementierung eines Messages Servers an.

## **4.4 Frameworks**

In diesem Abschnitt werden nun Frameworks dargestellt, die durch ihre umfassende Funktionalität die Entwicklung und Implementierung von EAI erleichtern sollen. Webservice, hierunter SOAP, stellen dabei seit neustem die Grundlage dar, die inzwischen von J2EE und von Microsofts Produkt .NET genutzt wird.

---

<sup>47</sup> Siehe auch: <http://www-5.ibm.com/de/software/websphere/webserver/mqseries.html>

<sup>48</sup> Siehe auch: [http://www.sonicsoftware.com/products/product\\_line.htm](http://www.sonicsoftware.com/products/product_line.htm)

<sup>49</sup> Siehe auch: [http://www.iplanet.com/products/iplanet\\_message\\_queue/home\\_message\\_queue.html](http://www.iplanet.com/products/iplanet_message_queue/home_message_queue.html)

<sup>50</sup> Siehe auch: <http://www.fiorano.com/>

<sup>51</sup> Siehe auch: <http://www.softwired-inc.com/>

<sup>52</sup> Siehe auch: <http://www.bea.com/products/weblogic/server/index.shtml>

<sup>53</sup> Siehe auch: <http://openjms.exolab.org/>

### 4.4.1 Webservices

Da Webservices inzwischen überall diskutiert und inzwischen auch von Sun und Microsoft favorisiert werden, erfolgt nun eine kurz Vorstellung dieser als „... Lingua Franca der System-Integration, innerhalb der Firma, aber auch zwischen Firmen“ [30, S. 26] bezeichneten Architektur. Dabei werden die einzelnen Komponenten, SOAP, WSDL und UDDI, erläutert.

#### SOAP

„SOAP beschreibt ein Protokoll, mit dem man Operationen von entfernten Servern aufrufen kann.“ [30, S. 27] Die Funktionsweise wurde schon zuvor in Abschnitt 4.2.3 beschrieben.

In der Literatur wird SOAP eine große Zukunft vorausgesagt, da inzwischen die wichtigsten Unternehmen in der Computerbranche hinter diesem Format stehen. Auch wird es nicht mehr nur von den Microsoft Produkten genutzt, sondern hält auch in den J2EE Standard seinen Einzug.

#### WDSL

Da SOAP als Aufrufprotokoll alleine nicht genügt, wurde die Web Service Description Language (WSDL) entwickelt, die folgende Komponenten enthält:

- Syntax der Service
- Bindung an das Netzwerkprotokoll
- Festlegung des Übertragungsformats
- Zuordnung zu einer URI. [63, S. 24]

Diese werden in einem XML-Dokument abgelegt, das einer DTD oder einem XML-Schema entspricht.

#### UDDI

Als letzte Komponente zählt UDDI (Universal Description, Discovery and Integration) zu den Web Services. Diese Initiative wird von namhaften Firmen, wie SAP, Sun, IBM und Microsoft unterstützt und stellt einen Verzeichnisdienst dar. Dieser unterscheidet sich allerdings von den sonstigen, da keine Dokumente oder Spezifikationen gesammelt werden, sondern nur Verweise darauf.

Allgemein ist bei den Web Services festzustellen, dass sie immer mehr im Kommen sind, da sie eine offene Plattform für die Integration bieten und selbst von Microsoft unterstützt werden.

### 4.4.2 J2EE

„Die Java-2-Plattform, Enterprise Edition (...) ist ein Framework für die serverseitige Entwicklung und Bereitstellung von Anwendungskomponenten.“ [36, S. 305] Dabei ist zu beachten, dass J2EE kein Produkt sondern nur einen Standard darstellt [58]. Dieser wurde unter der Leitung von Sun Microsystems entwickelt. Als Grundlage diente Java.

Um eine J2EE-zertifiziertes Produkt anbieten zu können, müssen die Hersteller die nachfolgend genannten Komponenten enthalten.

Die Enterprise JavaBeans (EJB) definieren Schnittstellen und Basisklassen und koordinieren zudem die Prozesse bei der Softwareentwicklung und –bereitstellung.

Eine weitere Komponente stellen die Java Servlets dar, die durch eine http-Anfrage aufgerufen werden. Ein Servlet-Engine dient dabei als Verbindung zwischen Java-Programm und Web-Server. Als Alternative kann auch eine Engine eingesetzt werden, die die direkt http-Requests annimmt, wie z.B. Tomcat von der Apache Group. [36, S. 305]

Die Servlets können mit Hilfe der Java Server Pages (JSP), die eine Spracheinbettung in HTML und XML definieren, generiert werden

Java Naming and Directory Interface (JNDI), das eine Definition für Verzeichnisdienste bietet, und Java Transaction API (JTA) zur Kapselung von Funktionen eines Transaktionsmonitors, sind ebenfalls im Standard enthalten.

Die Common Object Request Broker Architecture (CORBA) präsentiert einen Industriestandard und kann wie zuvor genannt in die objekt-orientierte Middleware eingeordnet werden.

Als letzte Komponente ist Java Database Connectivity (JDBC) zu nennen. Hierbei handelt es sich um einen Datenbankschnittstelle, die den Transport von SQL-Statements vereinheitlicht.

### 4.4.3 Microsoft .NET

Parallel zu dem Standard J2EE entwickelte Microsoft ein eigenes Produkt, .NET. „Die .NET-Plattform ermöglicht die Entwicklung und Nutzung von XML-basierenden Anwendungen, Prozessen und Websites als Dienste, die darauf ausgelegt sind, Informationen und Funktionen gemeinsam zu nutzen und zu kombinieren, auf jeder Plattform und auf jedem Smart Device.“ [67]

Die Basis für die .NET-Plattform stellt die Server Familie von Microsoft dar, die nach folgenden aufgezählt wird.

- SQL Server 2000
- BizTalk Server 2000
- Application Center
- Host Integration Server
- Internet Security & Acceleration Server 2000
- Mobile Information Server 2002
- Exchange 2000 Server
- SharePoint Portal Server 2001
- Commerce Server 2000
- Content Management Server 2001

Von diesen Servern soll nun BizTalk herausgegriffen werden, da es an sich schon ein eigenes Framework darstellt. In diesem lassen sich laut Merz sechs Bausteine identifizieren. [36, S.726ff]

Als Prozessmodeller entwirft das BizTalk-Framework XML-Nachrichten oder SOAP-Requests mit Hilfe des Orchestration Designers. Eine Ergänzung durch die Softwareintegration ermöglicht die innerbetriebliche Integration. Die Orchestration Engine, eine Workflow-Komponente, nutzt dies Fähigkeiten um das Zusammenspiel der verschiedenen Applikationen zu steuern.

Ein Mapping-Editor erlaubt die Transformierung der Dokumente in andere Formate, wie z.B. IDOCs, falls es benötigt wird. Die Konvertierungsregeln können dabei im Mapping-Editor visuell bearbeitet werden. Anschließend werden sie vom Mapper ausgewertet, der die Messaging Engine bei Bedarf aufruft.

Außerdem bietet BizTalk noch die Möglichkeit der Serververwaltung, die die Nutzer bei der Organisation der Server, Nachrichten und Steuerinformationen unterstützen soll.

### **4.4.4 Zukünftige Entwicklungen**

Es werden sicher die in den horizontalen Standards genannten Frameworks weiterentwickelt, sodass man sie diesen Frameworks zurechnen kann. Die besten Aussichten dafür hat ebXML und RosettaNet.

Sehr wahrscheinlich ist auch, dass die Web Services immer mehr gefördert werden und sie so eine Menge Standards ersetzen und den Entwicklern die Integration erleichtern werden.

## 5 Exemplarische Darstellung einer Integration

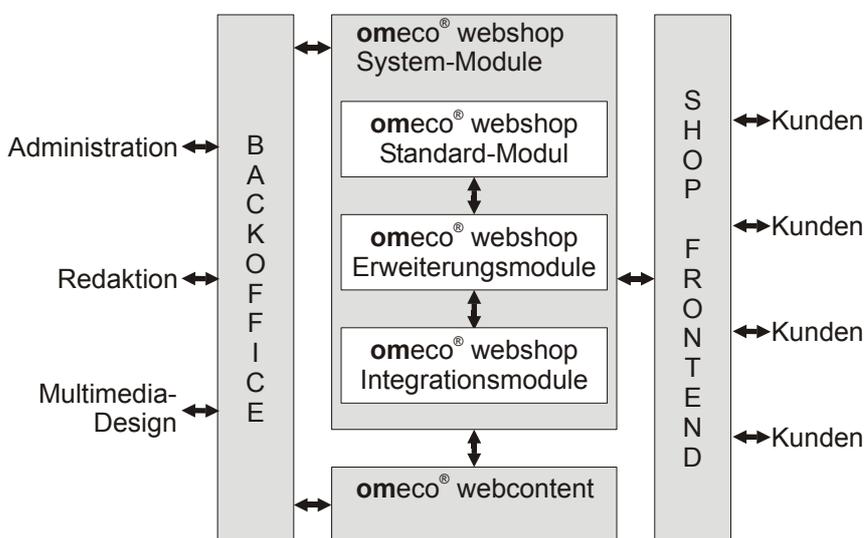
Die zuvor in der Theorie erarbeiteten Grundlagen sollen nun auf ein konkretes Problem in der Praxis angewendet werden. Dabei werden die Bereiche der Enterprise Application Integration und der B2B Integration abgedeckt.

### 5.1 Problemstellung

Zur Lösungsfindung für ein Anwendungsproblem, muss erst die Ausgangssituation dargestellt, das Problem genau definiert und Lösungsvorschläge gesucht werden, was im nachfolgenden Abschnitt schrittweise beschrieben wird.

#### 5.1.1 Ausgangssituation

Das von omeco entwickelte Produkt, omeco<sup>®</sup> webshop 1.2, ist ein leistungsstarkes Online-Shop-System, das auf innovativen Technologien beruht und sich von seinen Mitbewerbern durch Leistung, Sicherheit und Flexibilität abhebt. Durch das modulare Konzept des Webshops, das in Abbildung 5-1 dargestellt ist, ist dieser sowohl für den Business-to-Consumer- wie auch für den Business-to-Business-Markt interessant.

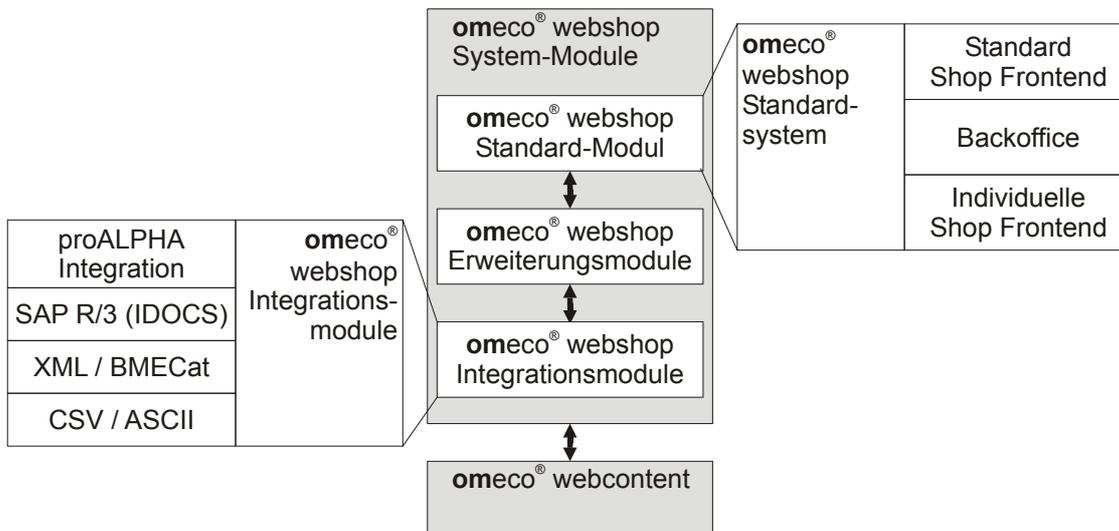


Quelle: [70, S. 2]

**Abbildung 5-1: Aufbau des omeco<sup>®</sup> webshop**

Der omeco<sup>®</sup> webshop besteht aus einem Standardmodul, den Erweiterungsmodulen und einigen Integrationsmodulen. Die Erweiterungsmodule stellen unter anderem die Funktionalitäten für Partnerprogramme, PDF Generator und Cross Selling zur Verfügung. Das Standardmodul, dessen Aufbau Abbildung 5-2 illustriert, stellt das Herzstück der

Shoplösung dar. Es beinhaltet den Shopkern für das Frontend und Backend sowie eventuelle Frontend-Anpassungen.



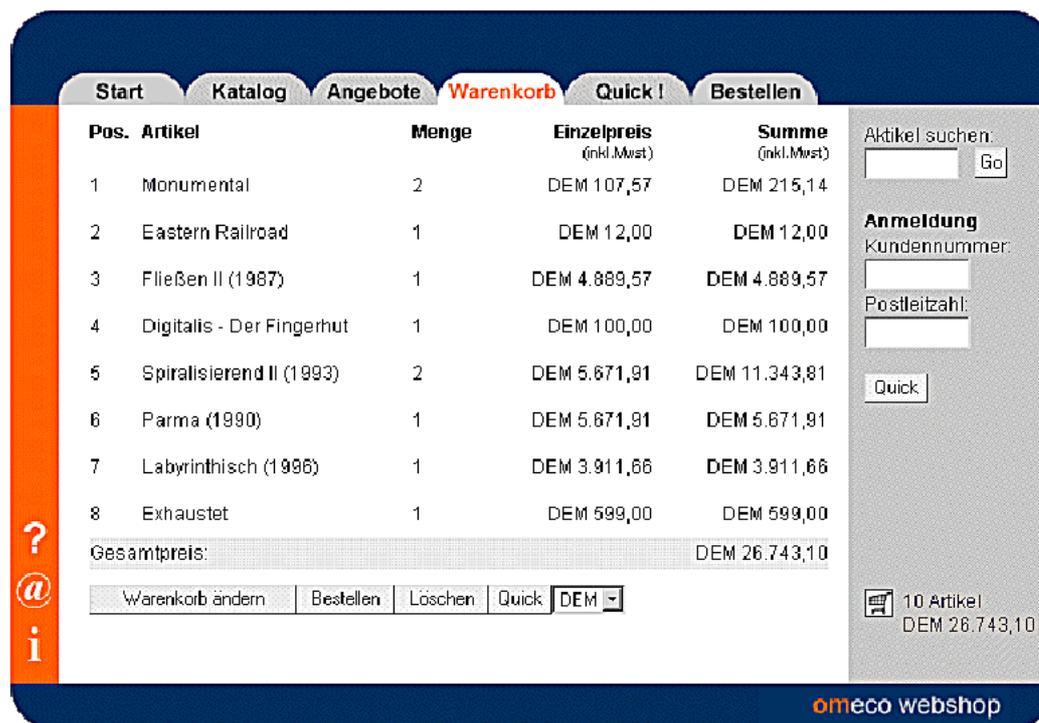
Quelle: Nach [70, S. 6 und S. 16]

**Abbildung 5-2: Standardmodule und Integrationsmodule des omeco® webshop**

Der Systemkern basiert auf PHP und kann wahlweise über ODBC auf verschiedene Datenbanken zugreifen. Für viele Kunden wird dabei die MySQL Datenbank von Vorteil sein, da sie im Web-Umfeld einen schnellen und problemlosen Betrieb mit verschiedenen Web-Servern, wie den hier genutzten Apache, erlaubt. Andererseits ergibt sich für Businesskunden bei der proALPHA-ERP-Systemlösung bei der Nutzung von Progress ein Vorteil, da sie nicht zwei verschiedene Datenbanken warten und pflegen müssen.

Die Integrationsmodule erweitern den Shop um Schnittstellen zu anderen Systemen, wie zu SAP oder um die Möglichkeit Produktkataloge im Standardformat, in diesem Fall, BMEcat, auszutauschen. Durch die frühzeitige Festlegung dieser Funktionalitäten ließ sich der konkrete Exportfilter, wie er in Abschnitt 5.4 beschrieben wird, schnell und zuverlässig implementieren.

Die Benutzung des Shops von omeco ist einfach und intuitiv. Durch eine hierarchische Anordnung von Produktgruppen sowie einer Suchfunktion fällt das Auffinden, der gewünschten Produkte leicht. Durch das klare Design des Shop, behält der Kunde jederzeit den Überblick, wie viel und zu welchen Preis er bestellen kann. Der einfach, aber funktional gehaltene Warenkorb, wie ihn Abbildung 5-3 zeigt, ermöglicht schnelles und bequemes Auswählen der zu bestellenden Artikel.



Quelle: [70]

**Abbildung 5-3: Warenkorb**

Durch die Zusammenarbeit von omeco mit proALPHA wurde das Produkt omeco<sup>®</sup> webshop an die Bedürfnisse der Business-Kunden von proALPHA angepasst und mit der dort verwendeten Technologie kompatibel gestaltet. Dies betrifft vor allem die zugrundeliegende Datenbank, die dazu von MySQL auf Progress geändert wurde, um im Umfeld des proALPHA ERP-Systemes nur ein Datenbankprodukt warten zu müssen.

Die Integration des Shops in die ERP-Systemlösung von proALPHA wurde als Datenintegration vorgenommen, d.h. jedes System operierte auf einer unabhängigen Datenbank, deren Datenbestand in Batchläufen oder manuell aktualisiert wurde. Damit konnte der Shop seine Bestelldaten an das ERP-System liefern und wenn nötig konnten vom ERP-System aus, die Produktdaten des Shops aktualisiert werden. Somit konnte nur eine statische Verfügbarkeitsprüfung von Produkten durchgeführt und Listenpreise, sowie vorher ausgehandelte Rahmenverträge berücksichtigt werden.

### 5.1.2 Aufgabenstellung

Die beiden vorgestellten Systeme, omeco<sup>®</sup> webshop und proALPHA ERP-System, galt es dynamisch in Echtzeit zu verbinden, um eine detailliertere Preisbildung für Businesskunden und eine aktuelle und aussagekräftige Verfügbarkeitsanzeige bieten zu können. Wie schon vorgestellt, konnten Preise nur abhängig von Rahmenverträgen ermittelt

werden. Spezielle Preisbildungen, die das ERP-System in seiner Business-Logik vorsieht, konnten nicht abgefragt werden, da diese in Echtzeit bezogen auf den Kunden und Produkt ermittelt werden mussten. Auch die Verfügbarkeit von Produkten konnte nur statisch abgefragt werden. Meist geschah dies durch nächtlichen Abgleich der Datenbestände zwischen den Datenbanken. Die Aufgabe bestand darin, die bisherige Datenintegration beider Systeme prototypisch in eine Funktionsintegration überzuführen und dies am Beispiel der Preis- und Verfügbarkeitsabfrage zu demonstrieren. Kein Gegenstand der Diplomarbeit sollte eine vollständige Funktionsintegration sein.

Bei der Integration beider Systeme galt es außerdem Sicherheitsfragen zu berücksichtigen. Das wohl wichtigste Argument war die Sicherheit des ERP-Systems, da es für ein Unternehmen hochsensible Daten enthält und einen Hacker-Angriff aus dem WWW ohne weitere Maßnahmen völlig hilflos unterlegen wäre. Des Weiteren bestanden bei den vielen in Echtzeit gestellten Anfragen der verschiedensten Shop-Benutzer Performanz-Bedenken sowohl auf Seiten des Shops als auch des ERP-Systems. Da die Daten über das Internet vom Shop zum ERP-System transportiert wurden, galt es außerdem diese gegen unbefugtes Verändern oder Mitlesen zu schützen.

Eine weitere Anforderung bestand in der Kompatibilität zu den existierenden Systemen, genauer sollte eine Lösung gefunden werden, die die Änderungen in beiden Systemen so gering wie möglich hielt und gleichzeitig eine flexible Erweiterbarkeit für zukünftige Entwicklungen ermöglicht. Auch sollte bei der Lösung auf die Kompatibilität mit der vorhandenen Infrastruktur geachtet werden.

Da diese Lösung später zum Austausch der Produktdaten genutzt werden sollte, galt es verschiedene Datenformate und Standards auf ihre Eignung zu untersuchen. Diese sollten sich nahtlos in die vorhandenen Systeme integrieren lassen und frei zugänglich sein, damit sie zum Austausch von Daten zwischen ERP-Systemen verschiedener Unternehmen dienen können.

Die Lösung sollte unter Berücksichtigung der zuvor genannten Kriterien einfach und kostengünstig sein, um der Positionierung des Webshops auf dem Markt für mittelständische Unternehmen gerecht zu werden.

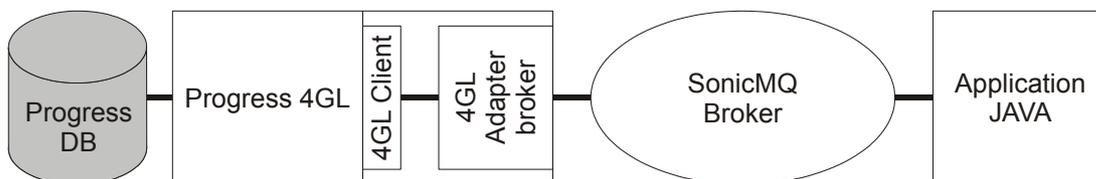
### 5.2 Lösungsansätze und Implementierungsmöglichkeiten

Nach der Beschreibung der Ausgangssituation und Aufgabenstellung mussten die verschiedenen Lösungsvarianten analysiert werden.

Da die bisherige Datenintegration den geforderten Ansprüchen nicht mehr gerecht wurde, wurde die Funktionsintegration gewählt. Wie im Abschnitt 4.1.2 beschrieben, können damit Applikationen auf die anderen Methoden über eine vernetzte Struktur zugreifen. Die Alternative dazu wäre eine Prozessintegration. Diese wurde aber aufgrund ihrer Komplexität zu weit führen, da es sich bei der vorhandenen Aufgabenstellung nicht um eine Integration von Prozessen handelt.

Nachdem sich die Funktionsintegration herauskristallisiert hat, galt es zu prüfen welche technischen Möglichkeiten in Betracht kommen. Es blieb letztendlich nur die Nutzung von Message orientierter Middleware in Betracht zu ziehen, da diese die nötige Flexibilität aufweist und mit den existierenden Message Brokern bewährte Lösungen bietet.

Es blieben also nur noch die verschiedenen Produkten zu evaluieren. Betrachtet wurden die Produkte MQSeries von IBM und SonicMQ von Progress. Durch die Bedingung die Kosten so gering wie möglich zu halten, wurde das Produkt SonicMQ gewählt, da es sich vorzüglich in die vorhandene Infrastruktur integrieren ließ. SonicMQ ist wie im Abschnitt 4.3.3 schon erwähnt ein JMS Broker, der von Progress entwickelt wurde und sich dank dieser Abstammung nahtlos in das Datenbankprodukt von Progress integrieren lässt. Da diese Datenbank die Grundlage für das ERP-System von proALPHA ist, können aufgrund dieser Bündelung Kosten und Aufwand gespart werden. Die konkrete Integration von SonicMQ in Progress geschieht über die Programmierschnittstelle 4GL von Progress und einen 4GL-SonicMQ-Adapter. Diese Funktionsweise wird in Abbildung 5-4 demonstriert.

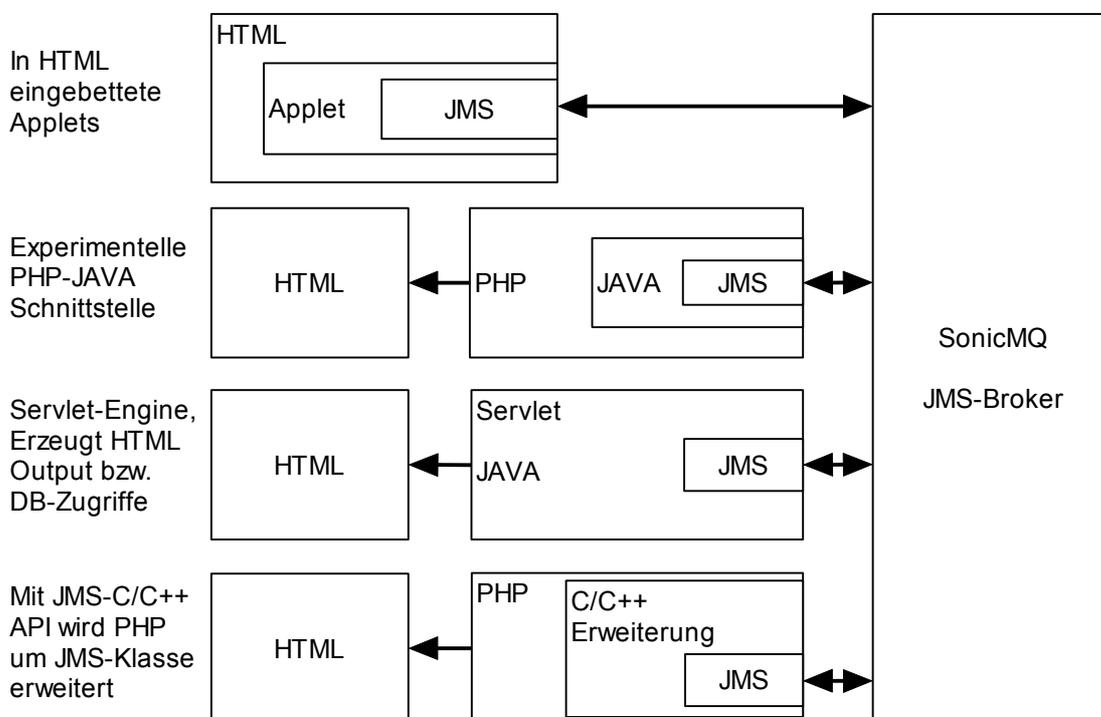


Quelle: Nach [12]

**Abbildung 5-4: Verbindung SonicMQ und Progress DB**

Mit dem SonicMQ-Broker steht ein JMS-Broker zur Verfügung, dessen API ein Java-Interface bietet. Da der bisherige Shop in PHP erstellt wurde, musste ein Weg gefunden

werden, ohne großen Aufwand den SonicMQ-Broker aus dem Shop anzusprechen. Dazu gibt es verschiedene Varianten, die in Abbildung 5-5 detailliert dargestellt werden. Die erste Variante beruht auf der Nutzung von Applets und der Integration der JMS-Schnittstelle in dieses. Aufgrund der Sicherheitseinstellungen für Applets kann sich dieses nur mit dem Server verbinden, von dem es aufgerufen wurde. Damit ist diese Lösung unflexibel, außerdem hätte sich die Ladezeit der Shopseite durch die vielen kleinen Applets deutlich erhöht. Eine andere Variante wäre die Nutzung der experimentellen Java-Schnittstelle in PHP, die aber gerade wegen ihres noch experimentellen Charakters nicht für produktive Zwecke einsetzbar war. Weiterhin wäre es denkbar, eine Servlet-Engine zu installieren und die Kommunikation zwischen den Shop und dem Servlets über http-Redirekts zu realisieren. Dies hätte den Vorteil, die Logik für die JMS-Implementierung vollständig in einem Servlet und damit in JAVA zu kapseln und damit den Änderungsaufwand im Shop auf ein Minimum zu reduzieren. Die letzte evaluierte Möglichkeit ist die Nutzung des von Progress angebotenen C/C++-APIs für SonicMQ und die gleichzeitige Erweiterung von PHP um eine JMS-Klasse. Da dies aber sehr zeit- und arbeitsaufwendig wäre, wurde diese Möglichkeit nicht in Betracht gezogen. Aufgrund ihrer Vorteile wurde sich für die Möglichkeit der Nutzung von Servlets entschieden.



Quelle: Eigene Darstellung

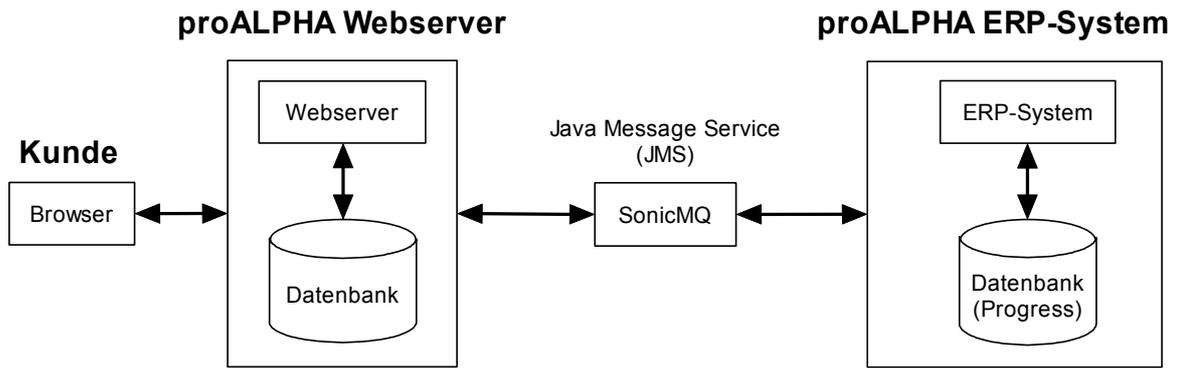
**Abbildung 5-5: Integrationsmöglichkeiten von JMS in omeco® webshop**

Die vorhandene Datenbank von Progress wurde beibehalten, obwohl bei der Entwicklung darauf geachtet wurde, unabhängig von der verwendeten Datenbank zu bleiben. Durch die Nutzung von JDBC ist dies einfach möglich. Damit wird es in Zukunft möglich sein, die Datenbank für den Shop unabhängig vom ERP-System zu wählen und damit Vorteile in der Nutzung und Wartung sowie der Performance zu erzielen.

Bei der Auswahl des Datenformates, welches zwischen dem ERP-System und dem Shop zum Transport der Daten dienen sollte, wurden aktuelle Entwicklungen berücksichtigt. Dies wären z.B. das BizTalk-Framework, BMEcat/openTRANS und andere. BizTalk scheidet aufgrund seiner Größe und Mächtigkeit für diesen Anwendungsfall aus, außerdem ist BizTalk im Vergleich zu dem verwendeten Message Broker von Progress relativ teuer und damit nicht für Mittelständische Unternehmen geeignet. BMEcat selber ist nur ein Produktkatalogaustauschformat und damit nicht geeignet Verfügbarkeiten oder spezielle Preise abzufragen. Mit openTRANS steht zwar ein Standard für Business-Transaktionen zur Verfügung, der aber noch stark in Entwicklung ist und die benötigten Preis- und Verfügbarkeitsabfragen erst in einer späteren Version standardisieren will. Aufgrund mangelnder Standardformate wurde sich für eine eigene und damit proprietäre Lösung entschieden.

### **5.3 Beschreibung der gewählten Lösung**

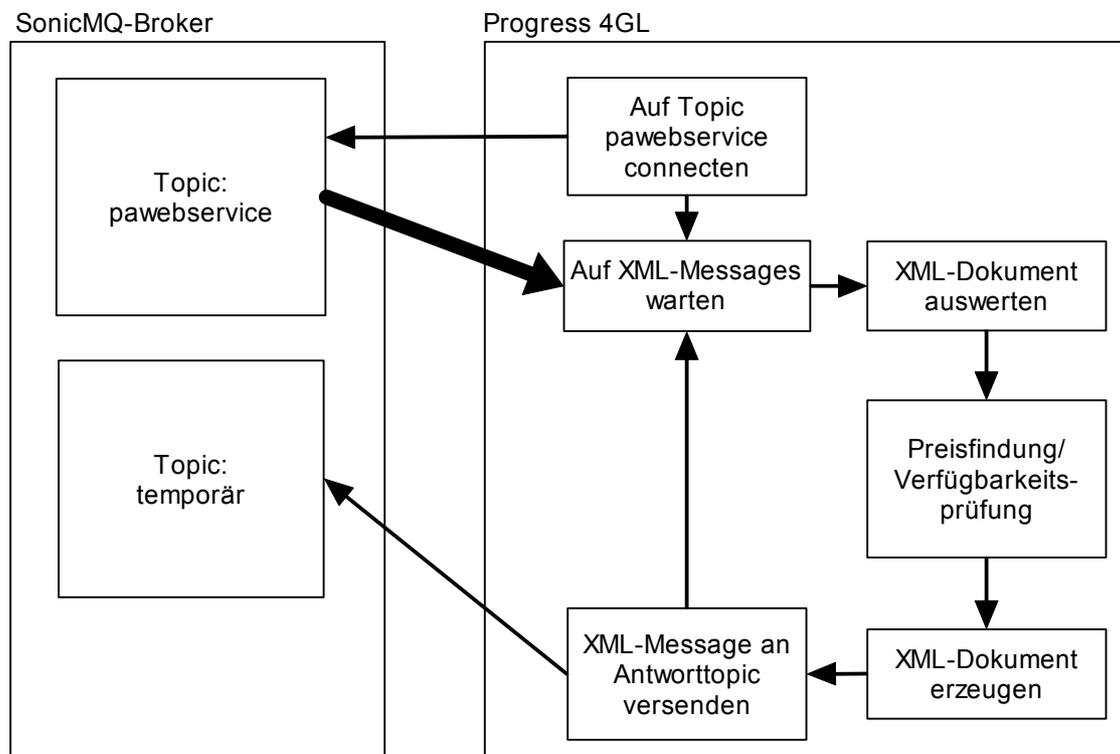
Die Online-Verbindung zwischen dem Webshop und dem proALPHA ERP-System wurde durch SonicMQ, den JMS-Broker von Progress, realisiert. Dieser ist ein rein in JAVA implementierter und von SUN für J2EE zertifizierter Java Message Server. Über diesen können mit einem definierten API Nachrichten ausgetauscht werden. Mit JMS ist dabei nicht festgelegt, wie die Übertragung der Nachrichten zu erfolgen hat, sondern wie man Zugriff auf die übertragenen Nachrichten erhält. Mit dem Publish-and-Subscribe Modell von JMS existiert ein Modell, mit dem mehrere Sender mit mehreren Empfängern kommunizieren können. Dabei muss sich der Empfänger auf ein Topic (Thema) des SonicMQ-Brokers verbinden und bekommt alle Nachrichten zugestellt, die von Sendern an dieses Topic geschickt wurden. Die Nachrichten werden nur dem Empfänger zugestellt, solange er mit dem SonicMQ-Broker verbunden ist. Da kein 100%-iger Empfang der Nachrichten gefordert ist, stellt ein Verlust, z.B. bei einem Übertragungsfehler, keinen Nachteil dar. Durch den Verzicht auf Persistenz können die Laufzeiten gering gehalten werden.



Quelle: Eigene Darstellung

**Abbildung 5-6: Verbindung proALPHA Webserver und proALPHA ERP-System**

Für die Echtzeitermittlung des Preises bzw. der Verfügbarkeit von Artikeln werden die im ERP-System schon vorhandenen Funktionen genutzt. Der JMS-Empfänger in dem ERP-System nimmt die XML-Nachrichten entgegen, wertet sie aus und ermittelt die gesuchten Werte. Diese werden wieder als XML-Nachricht verpackt und an den Sender zurückgeschickt. Dazu bietet sich die Nutzung temporärer Topics auf dem SonicMQ-Broker an, weil diese nur solange existieren, wie die Verbindung des Senders mit dem Broker besteht und somit eine eindeutige Zuordnung der Antwort zur Anfrage möglich ist.

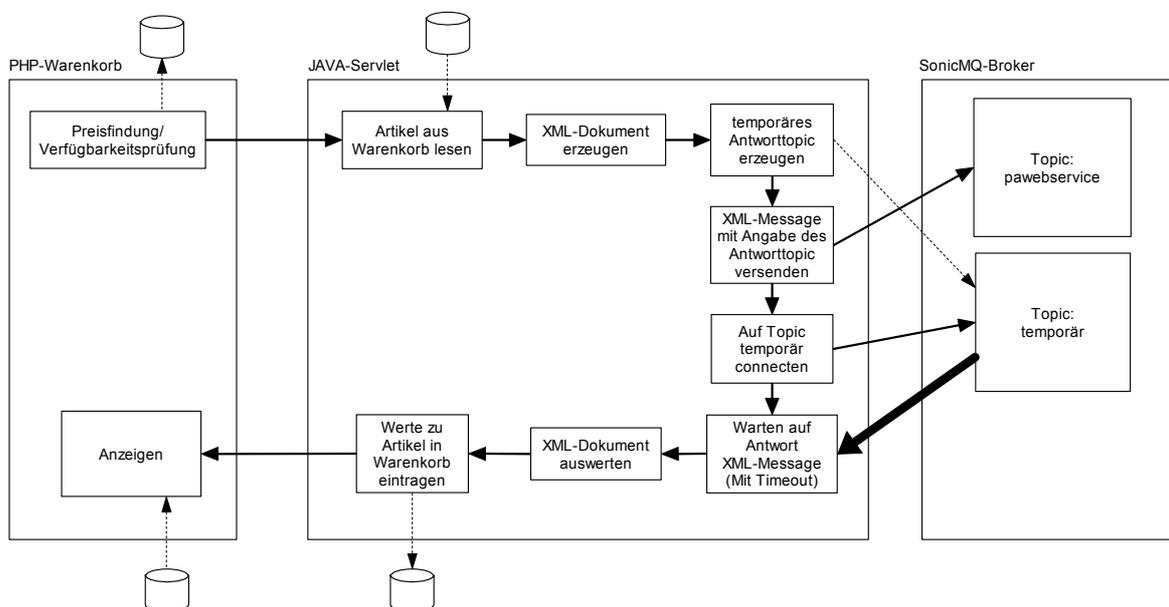


Quelle: Eigene Darstellung

**Abbildung 5-7: Message-Verarbeitung im proALPHA ERP SYSTEM**

## 5 Exemplarische Darstellung einer Integration

Wie vorher beschrieben, erhält das Servlet alle seine Daten aus der Datenbank und erstellt daraus eine XML-Nachricht. Diese wird um das Antworttopic ergänzt und an den Broker verschickt. Jetzt wird eine bestimmte Zeitspanne auf Antwort gewartet. Sollte keine Antwort erfolgen (Verbindungsabbruch o.ä.), wird ein Fehlercode an den Warenkorb zurückgegeben, andernfalls werden die Ergebnisse der Anfrage aus der XML-Nachricht extrahiert und in der Datenbank abgelegt. Nun wird, wie vorher beschrieben, der Warenkorb mit Hilfe der versteckten HTML-Form aufgerufen, der nun die Ergebnisse anzeigt. Wichtig ist, dass die Datumswerte, die vom Warenkorb zum Servlet und umgekehrt weitergereicht werden, positionsbezogen sind und durch die Session ID zugeordnet werden können. Damit kann bei einer Fehleingabe dem Nutzer diese auch nach einem Servletaufwurf angezeigt werden.

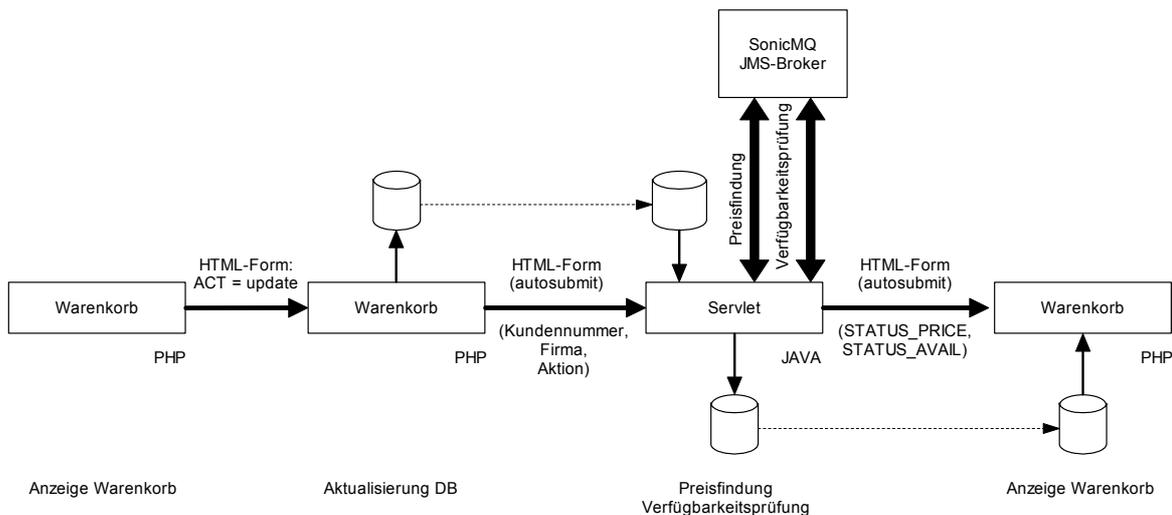


Quelle: Eigene Darstellung

**Abbildung 5-8: Message-Verarbeitung im omeco® webshop**

Im Webshop konnte auf keine vorhandenen Funktionen zurückgegriffen werden. Da SonicMQ ein JMS-Server ist, wurde konsequenterweise die Funktionalität der Echtzeitermittlung in Java realisiert. Dazu übernimmt der Warenkorb im Webshop nur eine Eingabe- und Darstellungsfunktion. Alle Änderungen im Warenkorb werden in die Datenbank geschrieben und dann ein Servlet aufgerufen. Der Aufruf erfolgt dabei über eine HTML Form, die versteckte Formularfelder enthält. Der Inhalt dieser Form wird per Javascript und einem Autosubmit an ein Servlet geschickt, das aus den übergebenen Werten die Aktion, die es ausführen soll, ermittelt. Dabei kann es sich um die Preisfindung

und die Verfügbarkeitsprüfung handeln. Wenn beide angegeben sind, werden sie derzeit noch nacheinander ausgeführt, woraus zwei JMS-Anfragen an das ERP-System resultieren.



Quelle: Eigene Darstellung

**Abbildung 5-9: Integration der Echtzeitabfrage in den omeco® webshop**

Die verwendeten XML-Dokumente enthalten die minimal möglichen Angaben für die Online Preis/Verfügbarkeitsermittlung. Das Zahlenformat ist deutsch, d.h. als Dezimaltrenner werden Kommas genutzt. Die ID's sind nicht zwingend sortiert oder aufeinanderfolgend. Wegen der prototypischen Implementierung und des geringen Dokumentumfangs wurden keine DTDs für diese Dokumente definiert. Sie sind also nur wohlgeformt. In Abbildung 5-10 werden die Dokumente für die Preisfindung dargestellt.

**Anfrage an ERP-System:**

```
<?xml version="1.0" ?>
<PriceRequest>
  <Firma>100</Firma>
  <Kunde>222</Kunde>
  <Preisanfrage ID="1">
    <Artikel>232412</Artikel>
    <Menge>8</Menge>
  </Preisanfrage>
  <Preisanfrage ID="2">
    <Artikel>345412</Artikel>
    <Menge>23</Menge>
  </Preisanfrage>
</PriceRequest>
```

**Antwort vom ERP-System:**

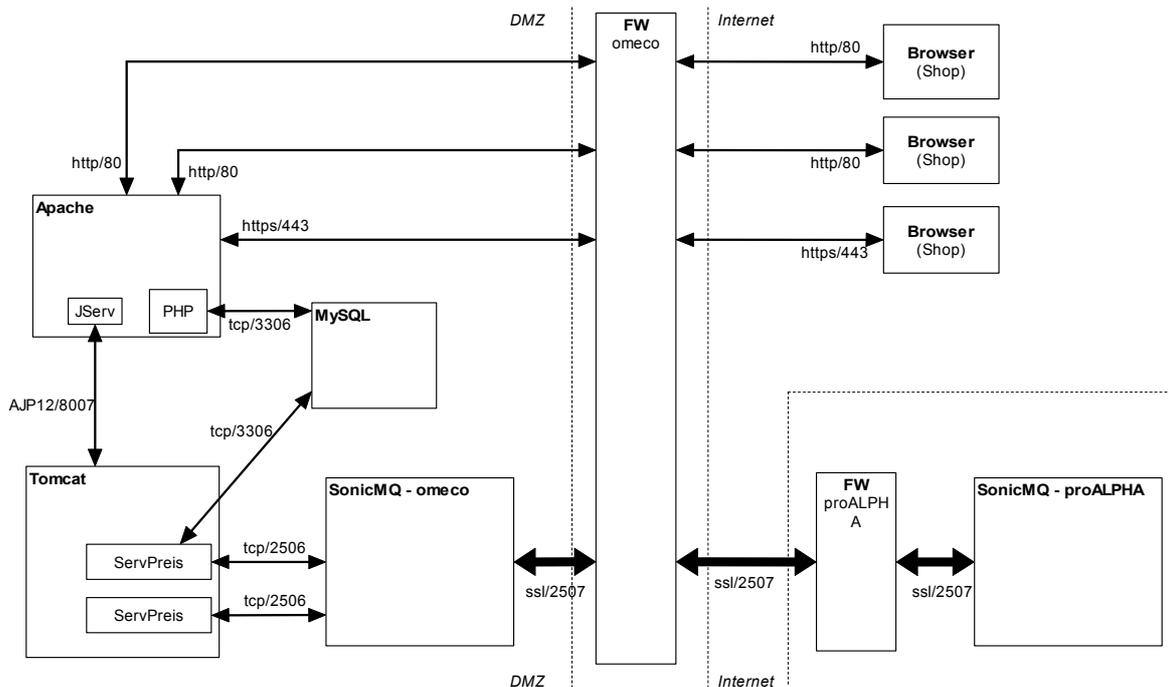
```
<?xml version="1.0" ?>
<PriceReply>
  <Preisanfrage ID="1">
    <Preis>100</Preis>
    <Waehrung>0</Waehrung>
  </Preisanfrage>
  <Preisanfrage ID="2">
    <Preis>200</Preis>
    <Waehrung>0</Waehrung>
  </Preisanfrage>
</PriceReply>
```

Quelle: Eigene Darstellung

**Abbildung 5-10: XML-Dokumente für die Preisfindung**

Der prinzipielle Aufbau des Testnetzes kann Abbildung 5-11 entnommen werden. Es ist ersichtlich das die gefundene Lösung auf einem Apache Webserver, der Datenbank (hier MySQL), der Servlet-Engine Tomcat sowie je einen Message-Broker für den Shop und das

ERP-System beruht. Dabei bilden die beiden JMS-Broker ein Cluster, das über eine durch SSL gesicherte Verbindung die Nachrichten austauscht. Anfragen an den Shop sind sowohl über http als auch gesichert über https möglich. Da alle 4 beteiligten Server, Apache, MySQL, Tomcat und SonicMQ, auf einem abgeschlossenen System liefen, bedurfte es keinerlei Verschlüsselung zwischen den Servern. Der Internetverkehr wurde zusätzlich noch durch eine Firewall abgesichert.



Quelle: Eigene Darstellung

**Abbildung 5-11: Netzstruktur**

Diese Implementierung wurde in den Standardsystemkern von omeco<sup>®</sup> webshop aufgenommen und steht damit einer weiteren Integrationsstufe zur Verfügung.

### 5.4 Zusatzimplementierung: Integrationsmodul BMEcat

Den Abschluss der praktischen Umsetzung des in den vorherigen Kapitel erworbenen Wissen bildete die Implementierung eines Exportfilters für die Produktdaten des Shops. Dieser Export sollte in ein Standardformat erfolgen, welches in der Industrie und besonders im Mittelstand hohen Anklang findet. Aufgrund einiger Nachfragen durch Kunden wurde das Format BMEcat genauer auf seine Eignung dafür untersucht. Da BMEcat sowohl Produktdaten als neuen Katalog als auch als Update eines vorhandenen Kataloges übertragen kann, eignete es sich besonders für die gegebene Aufgabenstellung.

## 5 Exemplarische Darstellung einer Integration

---

Wie im Abschnitt 5.1.1 aufgezeigt wurde, bestand die Schnittstelle für solche Erweiterungen schon. Die Umsetzung erfolge als JAVA-Package, die die neuen Fähigkeiten des in Entwicklung befindlichen omeco<sup>®</sup> webshops 2.0 ausnutzt. Bei diesem wird das Backend in Java entwickelt und somit die Schnittstellen präziser definiert.

## 6 Zusammenfassung

Die Entwicklung in der Geschäftswelt geht immer zu einer Vernetzung aller Unternehmen über. Dabei haben sich inzwischen verschieden Standardisierungsvorhaben gebildet, die sich zum Ziel gesetzt haben, die Integration zu erleichtern. Nicht nur Datenformate werden hierbei betrachtet, sondern auch gesamte Geschäftsabläufe und Lieferketten. Je nach Bedürfnis, kann sich die Integration von der Daten-, über die Funktions- bis zur Prozessebene erstrecken. Die Entscheidung, welche Integrationsart gewählt wird, hängt auch mit von der in der Strategie festgelegten Integrationstiefe ab. Natürlich kann die Integrationsrichtung auch nicht außer Acht gelassen werden, da sie die Effektivität im eigenen Unternehmen bestimmt.

Da die Integration von Software immer mehr zur Chefsache wird, ist das Interesse an dem sich entwickelndem Markt für EAI-Lösungen sehr groß. Dies lässt sich allein schon an den Fachzeitschriften für die IT-Branche und den Managermagazinen erkennen, in denen man zwangsläufig einem Artikel über einen Web Services oder ein neues Datenaustauschformat begegnet. Ganz euphorisch wird hier oft schon von der virtuellen Unternehmung gesprochen, die als die einzig vernünftige Unternehmensform in der Zukunft hingestellt wird. Leider wird dabei viel zu oft außer Acht gelassen, dass der Weg bis dahin noch weit ist und viele Standardisierungsbestrebungen sich gegenseitig behindern und so eine schnelle Einleitung der Zukunft unmöglich machen.

Durch die Vielzahl von Standards, Middlewarearten und Angeboten an Komplettpaketen wird der Markt zudem noch unübersichtlicher und es erfordert viel Zeit, sich über die bestehenden Möglichkeiten zu informieren. Tendenziell aber scheint es, dass sich die Web Services unter den anderen Format hervorarbeiten und die Chance haben, ein plattformübergreifender Standard zu werden. Allein durch die Förderung von Microsoft und Sun steigen ihre Möglichkeiten.

Neben den Standards, die ein feste Verbindung benötigen, entwickeln sich inzwischen immer mehr Übertragungsmöglichkeiten, die losgekoppelt von einer physischen Umgebung sind. So nutzen viele Unternehmen inzwischen Wireless Lan, um ihre Netzwerke leichter aufbauen zu können, oder den Mitarbeitern eine flexiblere Arbeitsumgebung zu bieten. Außerdem wird die Nutzung von Daten auch außerhalb der Büros fortschreiten durch den Einsatz von Technologien wie GPRS oder UMTS.

Erste Ansätze zur Integration von mobilen Applikationen in die unternehmenseigenen bestehen schon. So können Außendienstmitarbeiter schon jetzt ihre Daten ohne größeren Aufwand via Handy an die Zentrale übermitteln. Selbst wenn dies zur Zeit noch teuer erscheint, so steckt doch ein gewaltiges Einsparungspotential dahinter.

Die Ideen vieler Science Fiction Autoren scheinen in absehbarer Zeit wirklich zu werden und niemand muss mehr in einem Büro arbeiten, sondern kann sich egal von welchem Ort der Welt mobile mit seinem Computer verbinden. Das mag zwar im Moment noch nicht vorstellbar sein, aber wenn man sich die Entwicklung in den Großstädten betrachtet, in denen immer mehr Funkmasten für die Datenübertragung aufgebaut werden, sollte es doch irgendwann Realität sein. Und sobald selbst die Unternehmensgrenzen überwunden sind, ist die virtuelle Unternehmung endgültig erreicht.

Die praktische Implementierung stellt einen Kompromiss dar. Einerseits ist sie einfach und effektiv, andererseits trägt das selbstdefinierte Datenformat nicht zu offenen Schnittstellen zwischen den Systemen bei. Es bleibt abzuwarten, in wie weit sich der Standard openTRANS entwickeln wird. Laut Auskunft durch die Entwickler, sollen wahrscheinlich noch in diesem Jahr weitere Business-Dokumente aufgenommen werden, so z.B. die Preis- und Verfügbarkeitsabfrage. Damit wird dann die Kombination von BMEcat und openTRANS für mittelständische Unternehmen interessant. Es bieten sich damit vielfältige Integrationsmöglichkeiten, deren Potential hoffentlich durch weitere Standardisierung noch zunimmt.

## Glossar

API	Application Program Interface
BSS	Betriebswirtschaftliche Standard-Software
CBL	Common Business Library
CORBA	Common Object Request Broker Architecture
CRM	Customer Relationship Management
CRP	Customer-Centric Resource Planning
cXML	Commerce XML
DOM	Document Object Modell
DRA	Dynamic Routing Architektur
DTD	Document Typ Definition
EAI	Enterprise Application Integration
ebXML	Electronic Business XML
EDI	Electronic Data Interchange
EJB	Enterprise Java Beans
ERP	Enterprise Ressource Planning
FTP	File Transfer Protocoll
HTTP	Hypertext Transfer Protocol
ICE	Information and Content Exchange Protocol
IDL	Interface Definition Language
JDBC	Java Database Connectivity
JMS	Java Message Service
JNDI	Java Naming and Directory Interface
JSP	Java Server Pages
JTA	Java Transaction API
KMU	Klein- und Mittelständige Unternehmen
MOM	Message-orientierte Middleware
MRP	Material Ressource Planning
OASIS	Organisation for the Advancement of Structured Information Standards
PPS	Produktionsplanungs-Systeme
RPC	Remote Procedure Call
SAX	Simple API for XML
SCM	Supply Chain Management
SGML	Standard Generalized Markup Language
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SOX	Schema for Object-Oriented XML
SSL	Secure Socket Layer
TCP/IP	Transmission Control Protocol/Internet Protocol
UML	Unified Modeling Language

W3C	World Wide Web Consortium
XDR	XML Data Reduced
XML	eXtensible Markup Language
XSL	Extensible Stylesheet Language
XSL-FO	XSL Formatting Objects
XSLT	XSL Transformation

## Literatur

- [1] Albers, Sönke (Hrsg.) u.a., eCommerce, Einstieg, Strategie und Umsetzung im Unternehmen, 2. Aufl., F.A.Z.-Institut für Management-, Markt- und Medieninformation GmbH, Frankfurt/Main, 2000
- [2] Angeli, Axel, Streit, Ulrich, Gonfalonieri, Robi, The SAP R/3 Guide to EDI and Interfaces, Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig, Wiesbaden, 2000
- [3] Baldi, Stefan, Grundlagen der Wirtschaftsinformatik, R. Oldenbourg Verlag, München usw., 1998
- [4] Becker, Jörg, Informationsmodelle für das Electronic Business, in: Handbuch Electronic Business, (Hrsg.) Weiber, Rolf, S. 39 - 57
- [5] Bliemel, Friedhelm u.a. (Hrsg.), Electronic Commerce, Herausforderungen – Anwendungen – Perspektiven, 2. Aufl., Gabler Verlag, Wiesbaden, 1999
- [6] Brady, Mary, Web-Based Standards Roadmap, 15. Mai 2000, <http://xw2k.sdct.itl.nist.gov/brady/gsa-prod/working/stds.htm>, 6. Januar 2002
- [7] Bray, Tim, et al., Extensible Markup Language (XML) 1.0 (Second Edition), 6. Oktober 2000, <http://www.w3.org/TR/REC-xml>, 10. September 2001
- [8] Brenner, Walter, Keller, Gerhard (Hrsg.), Business Reengineering mit Standardsoftware, Campus Verlag, Frankfurt/Main usw., 1995
- [9] Brinkkemper, Sjaak, Lindencrona, Eva, Solvberg, Arne, Information Systems Engineering, State of the art and research themes, Springer Verlag, London, 2000
- [10] Brown, William J. et al., Anti Patterns, Refactoring Software, Architectures, and Projects in Crises, John Wiley & Sons, Inc., New York, 1998
- [11] Chesher, Michael, Kaura, Rukesh, Electronic Commerce and Business Communications, 2. Aufl., Springer-Verlag, London, 1999
- [12] Coeur, Jean-Michel, Progress 4GL to Java Communication with SonicMQ, April 2001, [http://www.progress.com/profiles/2001\\_04/source.htm](http://www.progress.com/profiles/2001_04/source.htm), 03. April 2002
- [13] Corsten, Hans, Gössinger, Ralf, Einführung in das Supply Chain Management, Oldenbourg Wissenschaftsverlag GmbH, München, Wien, 2001

- [14] Daum, Berthold, Scheller, Markus, Electronic Business, Methoden, Werkzeuge, Techniken und Systeme für den Unternehmenserfolg im Internet, Addison Wesley Longman Verlag GmbH, München, 2000
- [15] Elting, Andreas, E-Commerce II, RosettaNet: Aussichtsreicher E-Business Standard, in: iX, 1/2002, S. 114 – 117
- [16] Fischer, Norbert F., The Future of B2B-marketplaces in Europe, 26. Juni 2001, [http://www.competence-site.de/emarktplaetze.nsf/71947890407337DAC1256AA800612402/\\$File/zukunft\\_b2b-marktplaetze\\_geminiconsulting.pdf](http://www.competence-site.de/emarktplaetze.nsf/71947890407337DAC1256AA800612402/$File/zukunft_b2b-marktplaetze_geminiconsulting.pdf), 30.12.2001
- [17] Fowler, Martin, Refactoring, Improving the Design of Existing Code, Addison-Wesley Longman, Inc, Reading, USA, 1999
- [18] Frank, Ulrich, Standardisierungsvorhaben zur Unterstützung des elektronischen Handels: Überblick über anwendungsnahe Ansätze, in: Wirtschaftsinformatik 43, 3/2001, S. 283 – 293
- [19] Gora, Walter, Mann, Erika (Hrsg.), Handbuch Electronic Commerce, Kompendium zum elektronischen Handel, 2. Aufl., Springer-Verlag, Berlin usw., 2001
- [20] Hammer, Michael, Business reengineering, Die Radikalkur für das Unternehmen, 4. Aufl., Campus Verlag, Frankfurt/Main usw., 1994
- [21] Hentrich, Johannes, B2B-Katalog-Management, E-Procurement und Sales im Collaborative Business, Gallileo Press, Bonn, 2001
- [22] Holubek, Andreas, Sicherer Datenaustausch über das Netz mit JAXM, in: Java Magazin, 11/2001, S. 22 – 24
- [23] Hornberger, Werner, Schneider, Jürgen, Sicherheit und Datenschutz mit SAP-Systemen, Maßnahmen für die betriebliche Praxis, Galileo Press, Bonn, 2000
- [24] Jeckle, Mario, Die besondere Rolle von XML im EAI, 23. August 2001, <http://www.jeckle.de/files/eai2001.pdf>, 01. Januar 2002
- [25] Jung, Jean-Christian, ERP 2000 Germany, ERP Software & IT Services Industry in Germany Markets and Strategies 1998 – 2004, 2001, <http://www.pac-online.de/ger/PDF/ERP2000.pdf>, 7. Januar 2002
- [26] Kalakota, Ravi, Robinson, Marcia, Praxishandbuch E-Business, Der Fahrplan zum vernetzten Zukunftsunternehmen, Addison Wesley, München, 2001

- [27] Kelkar, Oliver, Schmitz, Volker, Otto, Boris, Spezifikation openTRANS<sup>®</sup> Version V 1.0, 7.September 2001, <http://www.bmecat.de/Download/openTRANS-V1.0.pdf>, Kostenlose Registrierung, 04.Januar 2002
- [28] Kobert, Thomas, XML, bhv Verlags GmbH, Kaarst, 1999
- [29] Köhler, Thomas, Best, Robert B., Electronic Commerce, Elektronischer Handel in der Praxis, Addison Wesley Longman Verlag GmbH, Bonn, 1998
- [30] Koschel, Arne, Tritsch, Roland, Web-Services zur Integrations, in: Java Spektrum, 3/2002, S. 26 – 33
- [31] Koushik, Srinivas, Straeten, Detlef, Eine strategische Roadmap zur Implementierung von E-Business-Lösungen, in: Handbuch Electronic Business, (Hrsg.) Weiber, Rolf, S. 93 - 114
- [32] Krause, Jörg, Electronic Commerce, Geschäftsfelder der Zukunft heute nutzen, Carl Hanser Verlag, München usw., 1998
- [33] Leierer, Gudrun A., PHP 4 & MySQL, 2. Aufl., Data Becker, Düsseldorf, 2000
- [34] Linthicum, David S., B2B Application Integration, e-Business-Enable Your Enterprise, Addison-Wesley, London, 2001, Sample-Chapter: <http://www.awl.com/cseng/titles/0-201-70936-8/ch1.pdf>, 19. März 2002
- [35] Linthicum, David S., Enterprise Application Integration, Addison-Wesley, London, 2000, Sample-Chapter: [http://www.awl.com/cseng/titles/0-201-61583-5/sample\\_chapter/eaich1.html](http://www.awl.com/cseng/titles/0-201-61583-5/sample_chapter/eaich1.html), 19. März 2002
- [36] Merz, Michael, E-Commerce und E-Business, Marktmodelle, Anwendungen und Technologien, 2. Aufl., dpunkt-Verlag, Heidelberg, 2002
- [37] Mintert, Stefan, E-Commerce I, ebXML: Grundlage für globalen Handel, in: iX, 1/2002, S. 110 – 113
- [38] Möhrlen, Regine, Kokot, Friedrich, SAP R/3 – Kompendium, Markt und Technik, Haar bei München, 1998
- [39] Monson-Haefel, Richard, Chappell, David A., Java Message Service, O'Reilly & Associates, Inc., Sebastopol, USA, 2001
- [40] Montanus, Sven, Im Schmelztiegel des E-Business, in: e-commerce magazin, 11/2001, S. 52 - 55

- [41] Morgenthal, JP, Enterprise Application Integration with XML and Java, Prentice Hall, Inc., New Jersey, 2001
- [42] Otto, Boris, Beckmann, Helmut, Klassifizierung und Austausch von Produktdaten auf elektronischen Marktplätzen, in: Wirtschaftsinformatik 43, 4/2001, S. 351 – 361
- [43] Pape, Ulrich, EAI-Tools – ein Marktüberblick, 23. August 2001, [http://www.competencesite.com/eaisysteme.nsf/BA297BD2025A1E34C1256AC60043469D/\\$File/010823\\_vortrageai-kongress.pdf](http://www.competencesite.com/eaisysteme.nsf/BA297BD2025A1E34C1256AC60043469D/$File/010823_vortrageai-kongress.pdf), 14. Februar 2002
- [44] Peat, Bruce, Webber, David, XML/EDI – “the e-Business framework”, 1997, <http://www.xmledi-group.org/xmledigroup/startde.htm>, 6. Februar 2002
- [45] Preßmar, Dieter B., Scheer, August-Wilhelm (Hrsg.), SAP® R/3® in der Praxis, Neue Entwicklungen und Anwendungen, Schriften zur Unternehmensführung, Gabler Verlag, Wiesbaden, 1998
- [46] Scheer, August-Wilhelm, Architektur integrierter Informationssysteme, Grundlagen der Unternehmensmodellierung, Springer-Verlag, Berlin usw., 1991
- [47] Scheer, August-Wilhelm, Wirtschaftsinformatik, Studienausgabe, Referenzmodelle für industrielle Geschäftsprozesse, 2. Aufl., Springer-Verlag, Berlin usw., 1998
- [48] Schill, Alexander, Middleware im Vergleich, o.J., [http://www.competence-site.de/eaisysteme.nsf/C937534DE4B6BC14C1256A14005E5D60/\\$File/middleware.pdf](http://www.competence-site.de/eaisysteme.nsf/C937534DE4B6BC14C1256A14005E5D60/$File/middleware.pdf), 13. März 2002
- [49] Schinzer, Heiko, Enterprise Application Integration – Werkzeuge, Juli 2000, [http://www.competence-site.de/eaisysteme.nsf/F9F38A78C910F136C12569EB002E926B/\\$File/eai-werkzeuge.pdf](http://www.competence-site.de/eaisysteme.nsf/F9F38A78C910F136C12569EB002E926B/$File/eai-werkzeuge.pdf), 23. März. 2002
- [50] Schmitz, Volker, Kelkar, Oliver, u.a., Spezifikation BMEcat® Version 1.2, 2001, <http://www.bmecat.de/Download/BMEcatV12.pdf>, Kostenlose Registrierung, 05.Januar 2002
- [51] Schulz, Stefan, Geschäftsszenarien und Grundfunktionen, in: Internet Selling, Integrierte Online-Verkaufslösungen mit SAP, Galileo Press, Bonn, 2000, S. 115 - 143
- [52] Serain, Daniel, Middleware, 2. Aufl., Springer Verlag, London, 1999

- [53] Sommergut, Wolfgang, Wird ebXML globaler B-to-B-Standard?, in: Computerwoche, 23/2001 vom 8. Juni .2001, S. 18 – 19
- [54] Steinbuch, Pitter A., Betriebliche Informatik, Kompendium der praktischen Betriebswirtschaft, Olfert, Klaus (Hrsg.), 7. Aufl., Friedrich Kiehl Verlag GmbH, Ludwigshafen, 1998
- [55] Teufel, Thomas; Röhricht, Jürgen; Willems, Peter, SAP-Prozesse mit Knowledge Maps analysieren und verstehen, Addison-Wesley, München usw., 2000
- [56] Thaller, Georg Erwin, Software-Qualität, Der Weg zu Spitzenleistungen in der Softwareentwicklung, VDE Verlag, Berlin, Offenbach, 2000
- [57] Thome, R., Electronic Commerce, Anwendungsbereiche und Potentiale der digitalen Geschäftsabwicklung, Verlag Franz Vahlen, München, 1997
- [58] Vawter, Chad, Roman, Ed, J2EE vs. Microsoft.NET, A comparison of building XML-based web services, Juni 2001, <http://www.theserverside.com/resources/article.jsp?l=J2EE-vs-DOTNET>, 13. März 2002
- [59] Wamser, Christoph (Hrsg.), Electronic Commerce, Grundlagen und Perspektiven, Verlag Franz Vahlen, München, 2000
- [60] Weitzel, Tim, Harder, Thomas, Buxmann, Peter, Electronic Business und EDI mit XML, dpunkt.verlag, Heidelberg, 2001
- [61] Wells, David, Budd, Michael, Ward-Dutton, Neil, OVUM EVALUATES: ENTERPRISE APPLICATION INTEGRATION, <http://www.ovum.com/go/product/sample/0048531.htm>, Kostenlose Registrierung, 03. April 2002
- [62] Wenzel, Paul (Hrsg.), Betriebswirtschaftliche Anwendungen mit SAP R/3<sup>®</sup>, Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig usw., 2001
- [63] Wieland, Thomas, Beschreibung von WebServices, in: Java Spektrum, 4/2001, S. 22 - 27
- [64] Winkeler, Thomas, Raupach, Ernst, Westphal, Lothar, EAI - Enterprise Application Integration, Die Pflicht vor der E-Business-Kür, [http://www.pwcglobal.com/at/ger/ins-sol/publ/20\\_pwc\\_e\\_bus.pdf](http://www.pwcglobal.com/at/ger/ins-sol/publ/20_pwc_e_bus.pdf), 30. Dezember 2001

- [65] Zerdick, Axel u.a., Die Internet-Ökonomie, Strategien für die digitale Wirtschaft, 2. Aufl., Springer-Verlag, Berlin usw., 1999
- [66] Datenmodell ETIM Klassifikationssystem Version 1.1, 26. April 2001, Datei: Datenmodell-ETIM1.1.doc, in: [http://www.etim.de/html/download/ETIM\\_1.1-CSV.zip](http://www.etim.de/html/download/ETIM_1.1-CSV.zip), 11. Februar 2002
- [67] Integration für ein agiles Business, Microsoft .NET im Unternehmen, 2002 <http://www.microsoft.com/germany/themen/net/default.htm>, 13. März 2002
- [68] Java 2 Platform Enterprise Edition (J2EE), 1999, [http://www11.informatik.tu-muenchen.de/lehre/lectures/ss2001/va/extension/html-kurz/va\\_course2.5.5.html](http://www11.informatik.tu-muenchen.de/lehre/lectures/ss2001/va/extension/html-kurz/va_course2.5.5.html), 13. März 2002
- [69] Java™ Message Service Documentation, 18. März 2002 <http://java.sun.com/products/jms/docs.html>, 20. März 2002
- [70] omeco® webshop, Version 1.2, März 2001, [http://www.omeco.de/webcontent/files/omeco\\_webshop\\_v12\\_d455f6.pdf](http://www.omeco.de/webcontent/files/omeco_webshop_v12_d455f6.pdf), 03. August 2001
- [71] <http://www.siemens.com.pk/images/sap-r3.jpg>

## **Erklärung**

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Leipzig, den 19. April 2002

Michael Walter