

Diplomarbeit

Erstellung einer Konzeption und eines Prototyps für die Einbindung
von Business-Daten in Office Produkte am Beispiel von
SAP ERP und IBM Productivity-Tool Symphony.

von Herrn Sebastian Lauffer
Arbeitsplatz IBM Deutschland GmbH, Walldorf
Erstbetreuer Herr Rainer Vetter und Herr Christian Holsing
Zweitbetreuer Dekan Prof. Dr. Herbert Schuster
Abgabetermin 24. November 2008

Heidelberg, den 24. November 2008

Vorsitzender des Prüfungsausschusses

Kurzfassung

Diese Diplomarbeit befasst sich mit der Integration von Geschäftsdaten in Office-Pakete. Der Trend zu offenen Standards, wie zum Beispiel XML, ODF oder Web-Services, verbessert die Chance große, nahezu allgemeine, Integrationslösungen zu schaffen. Das ODF (OpenDocument Format) ist ein ISO/IEC-Standard für den Office-Dokumentenaustausch. Der Markt der ERP-nach-Office-Integrationslösungen ist hart umkämpft und vor allem zwischen Microsoft und IBM aufgeteilt. Letztere setzt mehr auf offene Technologien mit Ihrem Projekt „Atlantic“ und kommt damit auch wesentlich günstiger auf den Markt als Microsoft mit seiner Lösung namens „Duet“.

Bestandteil dieser Diplomarbeit ist ein Prototyp, welcher Daten aus SAP in einem der neuesten Office-Pakete (IBM Lotus Symphony) darstellt. Dabei werden unterschiedliche Lösungswege diskutiert und teilweise implementiert.

Mehrkomponentenlösungen bringen zusätzliche Flexibilität und Modularisierungsmöglichkeiten. Eine zusätzliche Komponente zur Datentransformation in Form eines Web-Services wird vorgeschlagen. Damit wird der Anschluss von anderen Office-Paketen und Systemen möglich und ist ein idealer Ort für weitere Implementierungen geschaffen an dem Rohdaten, z.B. Daten aus unterschiedlichen ERP-Systemen, in ein einheitliches Format transformiert werden können. Mit jeder Datentransformation entsteht die Aufgabe die Datentypen richtig in das neue Format umzusetzen. Deshalb ist die Wahl der Formate besonders wichtig. Hält man sich so früh wie möglich an offene Standards, erspart man sich eventuell später einen Schritt der Datentransformation.

Jede der diskutierten Lösungsvarianten benötigt, als letzten oder auch einzigen Datentransformationsschritt, die Wandlung in ein ODF-Format. Es wird erkannt, dass dieser letzte Transformationsschritt auf zwei Arten durchgeführt werden kann. Zum einen ist es im Falle von ODF besonders einfach (da ODF im Wesentlichen aus gezippten XML-Dateien besteht) die Transformationen unabhängig von der Symphony API durchzuführen und erst anschließend das Dokument in der Anwendung zu präsentieren. Zum andern ist es möglich mittels der Symphony API, das ODF-Dokument zu erzeugen. Die erste Möglichkeit ermöglicht eine klarere Trennung von Generierung und Präsentation und wird daher bevorzugt.

Danksagung

Ich bedanke mich herzlichst bei allen Beteiligten, die mir diese interessante Diplomarbeit ermöglicht haben.

Besonders zu nennen sind seitens der Hochschule Frau Professor Dr. Ute Schäffer-Külz und Prof. Dr. Herbert Schuster in betreuender Funktion sowie Herr Prof. Dr. Dieter Homeister in der Rolle des Prüfungsamtes und die Betreuer und Helfer der IBM, d.h. hauptsächlich Herr Rainer Vetter, Herr Christian Holsing und Herr Bernd Schöner.

Ich bedanke mich bei dem ganzen Team der IBM und auch bei allen anderen Helfern und Unterstützern aus dem universitären sowie familiären Umfeld.

Inhaltsverzeichnis

<i>Kurzfassung</i>	<i>i</i>
<i>Danksagung</i>	<i>i</i>
<i>Einführung</i>	4
1 Ziel / Aufgabenformulierung	7
1.1 Aufgabenformulierungen und erste Forderungen	7
1.1.1 Forderungen an EDV-Lösungen	8
1.1.2 Drei Themenkomplexe - Zuordnen der Ziele	10
1.2 Architektur B2O, B2S und S2S	11
2 Theoretische Grundlagen	16
2.1 Der Begriff Integration	16
2.2 Web-Service	18
2.3 Eclipse	19
2.4 Office-Paket	19
2.5 Der Begriff Business-Daten	20
2.6 Enterprise Resource Planning	20
2.7 Das SAP-ERP-System	24
2.7.1 Kommunikation mit dem SAP-ERP-System	25
2.7.2 Schnittstellen zu SAP-ERP-Systemen	25
2.7.3 Synchrone Kommunikation mit dem SAP-System	27
2.7.4 Der SAP® Java Connector	28
3 Ist-Analyse	29
3.1 Die Office-Pakete (Ziel der Daten)	29
3.2 Erkennbare Trends bei Office-Paketen	30
3.3 Bestehende Lösungen	33
3.3.1 Das Produkt „Duet“ von Microsoft Office und SAP	33
3.3.2 SAP-Integration in Lotus Notes - Projekt „Atlantic“	33
3.3.3 CRM/ERP-Kopplung via Web-Services von CAS Software AG	34
3.4 Der Trend zum Open-Source	34
3.5 Ergebnisse der Ist-Analyse	35
4 Machbarkeits-Analyse	36
4.1 Fehlertoleranz bzw. Interpretationsfreudigkeit	36
4.2 Fehlertoleranz bezüglich der Daten	36
4.2.1 Nützlichkeit von unscharfer Information	37
4.2.2 Die Wissenstreppe – Alles bedarf einer Syntax	37
4.2.3 Fehlertoleranz bis zur völligen Nutzlosigkeit?	38
4.3 Business-Daten nach Office (B2O)	39
4.4 Business-Daten nach Symphony bzw. ODF (B2S)	40

4.5	Daten aus dem SAP-System.....	40
5	<i>Konzeption und Implementierung</i>	41
5.1	Szenario „Einladung zur Jubiläumsfeier“.....	41
5.2	Implementierung SAP-Daten in Symphony-Tabelle	42
5.2.1	SAP nach ODF ohne Symphony-API (B2ODF)	43
5.2.2	SAP nach ODF mit Symphony-API (O2B2ODF)	46
5.3	Aufbau der Implementierung des Prototyp	50
5.3.1	Sicht auf die Java-Paket-Struktur	50
5.3.2	Trennung von Verarbeitung und Design im Prototyp.....	52
5.4	Handlungsempfehlungen	55
5.4.1	Vorteile bei SAP nach ODF ohne Symphony-API.....	55
5.4.2	Vorteile einer zusätzlichen Daten-Transformationsschicht	55
5.4.3	Überlegungen zur Wahl der Kommunikationsform mit SAP.....	56
5.5	Datentransformation via Web-Service	58
5.6	Diagramm mit Fokus auf den Konfigurationsablauf	62
	<i>Ehrenwörtliche Erklärung</i>	64
	<i>Zusammenfassung und Ausblick</i>	65
	<i>Abstract</i>	67
	<i>Quellenverzeichnis</i>	68
	Bücher	68
	Internetquellen	79
	<i>Abbildungsverzeichnis</i>	83
	<i>Quelltextverzeichnis</i>	84
	<i>Tabellenverzeichnis</i>	84
	<i>Anhang</i>	85
	Marktanteile ERP-Software	85
	Liste von ERP Software-Systemen	86
	Liste von Office-Paketen	88
	Liste der Office-Dokument-Beschreibungs-Sprachen	89
	Verwendete Tools - Empfehlungen	90

Einführung

Kurzbeschreibung / Motivation

Diese Diplomarbeit bietet einen Beitrag zur Integration von Information – genauer für den Anwender/Sachbearbeiter relevanten Informationen – in Office-Pakete (typischerweise Text, Tabelle oder Präsentation). Beispielsweise gehört der E-Mail-Versand eines Produktsortiments in Tabellenform zum Tagesgeschäft manches Sachbearbeiters. Dabei werden häufig mehrere komplexe Anwendungen gleichzeitig bedient, wie beispielsweise ein Logistiksystem und ein Ressourcen-Planungssystem (s. Kapitel 2.6 „Enterprise Resource Planning“) und eine Anwendung aus seinem Office-Paket (zum Beispiel eine Tabellenkalkulation).

Allgemein kann man beobachten, dass die Vielfalt und Komplexität des Informationsangebots¹ und der damit steigende Bedarf an Integration für Unternehmen und die Informatik zu einer wesentlichen Aufgabe geworden ist. Dabei ist das Problem der Integration nicht neu und es existieren bereits einige Lösungen, wie z.B. eine mit Microsoft-Office namens „Duet“² oder eine neue Lösung von IBM und SAP mit dem Codenamen „Atlantic“³. Der Transport von Daten in das persönliche Office zur Weiterverarbeitung, beispielsweise zur Angebotserstellung, geschieht seit es Office-Systeme gibt. Jedoch sind aktuelle Lösungen nur eingeschränkt verfügbar und erst wenig verbreitet. Klingelhöller [KlH01, S.121] behauptet sogar, dass „zum Teil immer noch“ uneffizienter Weise mit „Papierbergen“ gearbeitet wird.

Die Bildung von offenen Standards (wie zum Beispiel dem Quasi-Standard ODF) wird zur Vereinfachung solcher und anderer Integrations-Aufgaben beitragen⁴. Hein [He07, S.66] prognostiziert jedoch die Kombination von Informationen als eine wesentliche „Herausforderung der kommenden Zeit“. „Viele Technologien“, welche „in den nächsten zehn Jahren zum Einsatz kommen“ würden, seien „bereits erfunden“ [ebd.].

Da überrascht die Aufgabenstellung dieser Diplomarbeit nicht, nämlich verschiedene Business-Daten [vgl. Information [He07, S.66], insbesondere vom Business-Software-Hersteller SAP, in einem Office „IBM Lotus Symphony“ verfügbar zu machen und sich allgemein mit der Thematik Integration von Business-Daten in Office zu befassen⁵. Der mit dieser Arbeit verfügbar gemachte Prototyp verwendet als Ziel der Datenintegration das Office-Paket „IBM Lotus Symphony“, welches seit Mai 2008 in der Version 1.0 erhältlich ist. „IBM Lotus Symphony“ setzt auf das weit verbreitete Office-Paket „OpenOffice“ auf und unterstützt ebenfalls die neuen OpenOffice-Dokument-Formate (ODF), welche mittlerweile sehr große Verbreitung gefunden haben und vielleicht bald den Microsoft-Formaten den Titel „Quasi-Standard-Formate“ streitig machen werden. Wie Golem am 20.06.2008 [G08] berichtete, will Microsoft nun sogar am ODF-Format mitarbeiten und ihr eigenes Microsoft-Format (OOXML-Format) zurückstellen (ODF habe gewonnen). Außerdem ist IBM Lotus Symphony (wie OpenOffice) kostenlos erhältlich und besitzt eine frei verfügbare API⁶. Aus der Aussage von Saleck [Sa05, S.113], welcher diagnostiziert,

¹ Nach Caspers [Ca04, S.63] ist davon auszugehen, dass sich das weltweit verfügbare Wissen alle fünf bis sieben Jahre verdoppelt.

² Die Duet-Software für Microsoft Office und SAP (betriebswirtschaftliche Software) ermöglicht dem Anwender einen Zugang zu SAP-Daten und -Prozessen über Microsoft-Office seit 2005 [SG08].

³ Software zur Integration von SAP Business-Prozessen und -Daten in den Lotus Notes Client.

⁴ Laut einer Studie der Marktforscher von IDC, vom 31. Mai 2007, wird sich der Markt für „Open Source-Software“ in fünf Jahren verdreifachen [WR08].

⁵ Lösungen dritter finden sich im Kapitel „3.3 Bestehende Lösungen“.

⁶ API steht für „Application Programming Interface“ bzw. für Programmierschnittstelle in der Informatik.

dass sich „auf dem Gebiet der Business-Software“ eine „Open-Source-Entwicklergemeinschaft zusammen [finden, S. L.]“ würde, kann man folgern, dass diese frei erhältliche Symphony-API wahrscheinlich weite Verbreitung finden wird. Bereits innerhalb der ersten Woche nach der Veröffentlichung der ersten Beta-Version von Lotus Symphony, verkündete IBM mehr als 100.000 registrierte Symphony- Benutzer [BDvL08]. Alle Komponenten dieser prototypischen Lösung (Office, Entwicklungsumgebung, Programmiersprache etc.) sind (ausgenommen das SAP-System) kostenlos für jeden verfügbar.

Umfeld und Rahmen der Arbeit

Die vorliegende Diplomarbeit wurde im Rahmen des Diplom-Studiengangs Wirtschaftsinformatik der Fachhochschule Heidelberg und im Hause des „IBM SAP International Competence Center (ISICC) Walldorf“ betreut und angefertigt.

Das ISICC, deren Abteilung „Lotus Business Development“⁷, betreute insbesondere die Entwicklungsarbeit und profitiert direkt von deren technischen Ergebnissen z.B. in Form eines Prototyps. Das ISICC stellte Informationsmaterial, Rechenzeiten, technische Einrichtungen und sonstige Unterstützung, soweit erforderlich und möglich, zur Verfügung.

„Das ISICC ist die zentrale Anlaufstelle für Mitarbeiter von IBM, SAP und Partnern - vor allem jedoch für Kunden, die Lösungen suchen, die auf den SAP Anwendungen in Kombination mit [...] IBM Produkten basieren.“ [ISICC08]

Das ISICC beschäftigt sich mit Fragen der Portierung, Integration und Interoperabilität zwischen den Systemen dieser weltweit operierenden Unternehmen. Die Zusammenarbeit zwischen IBM und SAP existiert seit über 30 Jahren. Im Hardwaresektor wird für die Kompatibilität der SAP-Lösungen mittels IBM-Hardware gesorgt. Im Softwaresektor liegt der Schwerpunkt bei der Integration der Softwareprodukte.

Die Abkürzung „SAP“ steht als Synonym für Systeme, Anwendungen und Produkte in der Datenverarbeitung“ [IM06, S.155, siehe auch Kapitel 2.6 „Das SAP-ERP-System“]. SAP ist weltweit führender Anbieter von Geschäftslösungen. Eine mehrere Seiten starke Liste von SAP-Lösungen und Anwendungen findet sich auf <http://www.sap.com/germany/solutions> (z.B. Lösungen und Anwendungen in den Bereichen Applikations- und Plattform- Integration, Supply Chain Management oder klassischem ERP etc.).

Im Folgenden wird unter SAP-ERP⁸-Lösungen, in Anlehnung an den erst genannten Begriff ERP-Lösungen, nicht das Produkt „SAP ERP“ verstanden, sondern alle ERP- bzw. Geschäftslösungen von SAP. Die klassischen SAP- Produkte, wie beispielsweise das Produkt „SAP ERP“ zählen „inzwischen zu den am weitesten verbreiteten ERP-Systemen“ [vgl. IM06, S.155]. Noch deutlicher als die Spitzenposition bzgl. weltweiter Verbreitung ist die Marktherrschaft innerhalb

⁷ Das Entwicklerteam der 'Global IBM Lotus - SAP Alliance' im IBM Kompetenz-Center in Walldorf beschäftigt sich derzeit vorrangig mit dem "Atlantic" Produkt, einer Integration von SAP Business Prozessen und Daten in den Lotus Notes Client.

⁸ Nach Meyer [CSmEK02, S.63/64/78] zählt ein „ERP-System“ (z.B. SAP, JD Edwards, ...) zu den „Back-Office“- Systemen und enthält (im klassischen Sinne) „keine Front-Office-Funktionen“, sondern lediglich die klassischen Module wie „Produktion, Finanz- und Betriebsbuchhaltung, human Resources [sic! korrekt „human resources“, Anm. d. Verf.] etc.“. Allerdings verwischt die Trennung zwischen Front-Office- und Back-Office-Funktionen mittlerweile [siehe WNFsu06, S.130].

Europas [vgl. ♣SPP06]. Diese Aussagen werden durch folgende Tabellen im Anhang belegt (Tabelle „Marktanteile ERP-Software“ auf Seite 85).

SAP Enterprise Resource Planning Business-Lösungen, wie das Produkt „SAP ERP“, sind durch einen modularen Aufbau gekennzeichnet und werden meistens in den Bereichen Finanzbuchhaltung, Controlling, Produktion und Logistik eingesetzt.

Die Fakultät für Informatik der Fachhochschule Heidelberg besteht bereits seit über 30 Jahren. Das Leitbild der Fakultät ist geprägt von drei grundlegenden Aspekten: Praxisorientierung, Innovation und Internationalisierung. Im Studiengang der Wirtschaftsinformatik⁹ werden über die ganze Studienzeit hinweg SAP-spezifische Lehrveranstaltungen angeboten. Derzeit sind dies u. a. im 1. Trimester „SAP Application“, im 2. Trimester „SAP Customizing and Development“ und im 3. Trimester „SAP Technology“ [♣SSWi08].

Kapitelunterteilung

Nach der Einleitung, mit einer kurzen Beschreibung der Diplomarbeit und deren Ziele und der Ausgangslage, werden die Ziele genauer in Kapitel 1 definiert.

Im ersten Kapitel werden die angesprochenen Ziele in Form eines Pflichtenheftes detaillierter dargestellt. Kapitel 1.1 führt diese stichpunktartiger aus. Auch die nächsten Unterkapitel dienen letztlich der Beschreibung der Ziele, jetzt jedoch in anschaulicherer Form. Die Voraussetzungen für die Umsetzung der Ziele finden sich ebenfalls im Kapitel 1 unter „Architektur Business-Daten nach Symphony (B2S)“ und im darauf folgenden Kapitel 2 „Theoretische Grundlagen“.

Von der Zielsetzung über die „Ist-Analyse“ (Kapitel 3) zur „Machbarkeits-Analyse“ (Kapitel 4) bis zum (Kap. 5 „Konzeption und Implementierung“) werden Ziele/Themen aufgebaut, aus denen dann Lösungsansätze entstehen.

Daher finden sich in diesen Kapiteln ähnlich klingende Überschriften. So wird beispielsweise die Überschrift „SAP-Daten nach Symphony“ im fast allen Kapiteln (wieder-) verwendet. Es werden, zwecks der Wiedererkennung, Schlüsselphrasen ganz bewusst wörtlich wiederholt.

In der „Ist-Analyse“ (Kapitel 3) wird untersucht inwieweit die Ziele bereits von anderen Anbietern ähnlich oder identisch umgesetzt sind. Und in der „Machbarkeits-Analyse“ (Kapitel 4) werden insbesondere Ziele betrachtet, welche derzeit noch von niemandem umgesetzt sind und vielleicht auch prinzipiell nicht umsetzbar (machbar) sind.

Von da an geht es weiter mit der Umsetzung. Und zwar in konzeptioneller Form (Kap. 5 Konzeption und Implementierung) so wie auch in Form einer prototypischen Lösungen. Der konzeptionelle Inhalt ist von der Darstellung der prototypischen Lösungen meist nicht deutlich getrennt.

⁹ Anm. d. Verf.: dies studiert der Verfasser als Diplomstudiengang

1 Ziel / Aufgabenformulierung

Die hier, stichpunktartig, mit Kürzel versehenen, formulierten Ziele wirken sich auf spätere Kapitel aus. Forderungen/Konsequenzen der Ziele werden später in Kapitel 4 „Machbarkeits-Analyse“ diskutiert und in Kapitel „5 Konzeption und Implementierung“ abgeleitet und weiter konzeptionell umgesetzt.

1.1 Aufgabenformulierungen und erste Forderungen

Die „Erstellung einer Konzeption und eines Prototyps für die Einbindung von Business-Daten in Office Produkte am Beispiel von SAP ERP und IBM Productivity-Tool Symphony“ ist der Titel dieser Diplomarbeit und die vom ISICC vorgegebene Aufgabe.

Der Prototyp muss Eclipse-basierend mit dem Symphony SDK zu erstellen sein. Und die Konzeption und der Prototyp müssen gewisse Freiheitsgrade vorsehen:

- unterschiedliche Quellen, wo die Business-Daten herkommen
 - Daten aus SAP (Zugriff mittels RFC/BAPI) via JCo¹⁰
 - Daten aus SAP via Web-Service
 - Daten aus sonstigem Web-Service.

Diese Anforderungen an Konzeption und Prototyp habe ich in folgende, (d.h. selbst formulierten) stichpunktartige Fein-Ziele eingebettet:

Selbstaufgelegte Feinziele dieser Diplomarbeit:


Überlegung und Diskussion

- (Ziel A) einer **allgemeinen Lösung**
für die Einbindung von Business-Daten in Office Produkte
- (Ziel B) **einer Lösung**
für die Einbindung von SAP-ERP-Business-Daten in das Office IBM Lotus Symphony
 - (Ziel B.1) **Architektur** des Lösungsansatzes
 - (Ziel B.3) **Vorgehensweise** des Lösungsansatzes
 - (Ziel B.3) **Vorläufige Applikation, d.h. Prototyp** von Ziel B.
 - (Ziel B.3.a) Prototyp ist **lauffähig** (nicht marktreif).
 - (Ziel B.3.b) Prototyp ist **Eclipse-basierend** und mit dem **Symphony SDK** zu erstellen
 - (Ziel B.3.c) Möglichst **viele unterschiedliche Datenquellen**¹¹ (Stichwort: Vollständigkeit) sollen unterstützt werden. Z.B.:
 - Daten aus SAP (Zugriff mittels **RFC/BAPI**) via **JCo**
 - Daten aus SAP via **Web-Service**
 - (Ziel B.3.d) Als Ausgabeformat sollen die wichtigsten in Office IBM Lotus Symphony verfügbaren Formate unterstützt werden (**Tabellenkalkulation bzw. ODS, Text-Dokument bzw. ODT, Präsentation bzw. ODP**).


¹⁰ Diese Abkürzungen werden im Kapitel „Theoretische Grundlagen“ erklärt.

¹¹ Als Datenquellen (hier für ein Data Warehouse) werden Systeme verstanden, aus denen das System die benötigten Daten bezieht. Dies sind für mittelständische Unternehmen meist ERP-Standardsysteme, Office Produkte oder Datenbanken.




1.1.1 Forderungen an EDV-Lösungen

Entwickler von EDV-Lösungen sollten sich bei Ihrer Zielfindung, soweit wie möglich, auch an folgenden Forderungen orientieren [vgl. Forderungen an ein Data-Warehouse siehe VvDwa07]:


i. **Plattformunabhängigkeit**


Eine Lösung sollte auf möglichst vielen Betriebssystemen¹² lauffähig sein. Nach Bresik [VvDwa07, S.14] werden selbst „in mittelständischen Unternehmen [...] mehrere unterschiedliche Betriebssysteme eingesetzt.“

ii. **Datenquellen-Vielfalt**


Die Anzahl der unterstützten Datenquellen¹¹ machen meist den größten Wert von Integrationslösungen aus [vgl. mit Forderungen zu „Data Warehouse“- Lösungen, VvDwa07, S.14]. Hierbei zählen zu den erwünschten Datenquellen [ebenso nach Bresik] nicht nur Schnittstellen¹³ zu den wichtigsten ERP-Lösungen, sondern auch Schnittstellen zu **Office Produkten** oder **Datenbanken** oder zu **Daten aus dem Internet** oder zu so genannten „**Flat Files**“¹⁴. Zu den wichtigsten ERP-Lösungen zählen, nach Bresik [VvDwa07, S.14], die Unternehmen „SAP, Microsoft, Infor Global Solution, Sage, Abas Software“ und andere Anbieter, welche als Datenquelle unterstützt werden können „sowie Daten aus den Datenbanken DB2, Oracle, Oracle Express, MS SOL Server, Synbase, Informix, dBASE, InterBase, Lotus Notes, MySQL, SOLite, Teradata sowie sonstige Datenbanken“. Auch Daten mit Binärdaten, insbesondere Bildern sollen, nach Bresik [VvDwa07, S.19], berücksichtigt werden: „Es ist darauf zu achten, dass in der Modellierung auch nicht formatierte [sic! Bilder sind auch formatierte Daten, Anm. d. Verf.] Daten gespeichert werden können. Besonders Bilddaten spielen in vielen Unternehmen eine wichtige Rolle.“

iii. **Schnittstellen¹³-Vielfalt**

Nach Bresik [VvDwa07, S.14] ist die „Voraussetzung der Integration der benötigten Daten“ die „Unterstützung verschiedener Schnittstellen“. Viele ERP-Systeme bieten mehrere Schnittstellen.

Nach Bresik [VvDwa07, S.14] werden „Heute [...] größtenteils indirekte Schnittstellen (z. B. **ODBC**) verwendet, da diese nicht laufend an die Systeme angepasst werden müssen. Diese sind sowohl in Großunternehmen als auch in mittelständischen Unternehmen weit verbreitet. Zusätzlich sollte die SAP Schnittstelle BAPI sowie OLE DB bzw. OLE DB FOR OLAP und sonstige Schnittstellen unterstützt werden.“

iv. **Fehlertoleranz**

Fehlerbehandlung ist notwendig, „Da nicht davon auszugehen ist, dass die Integration der Daten immer reibungslos funktioniert“ [VvDwa07, S.16]. Meist will man keinen Abbruch des Imports bei kleineren Fehlern, wie es beispielsweise bei der Interpretation von Web-

¹² Im Falle einer Office-Integration sind die wichtigsten Betriebssysteme Windows, Macintosh und Linux.

¹³ Die Schnittstelle oder das Interface (englisch „Grenzfläche“) ist der Teil eines Systems, der der Kommunikation dient. Der Begriff stammt ursprünglich aus der Naturwissenschaft und bezeichnet die physikalische Phasengrenze zweier Zustände eines Mediums.

¹⁴ Nach http://www.virenschutz.info/Flat-file-Techniklexikon-bei-Virenschutz-info_690.html

ist mit „Flat file“ eine Datei gemeint, welche nur eine kleine Struktur aufweist und besonders im World Wide Web zu finden ist. Nach http://www.altova.com/de/dev_portal_flatfile_mapping.html

„Flat File“ ist eine generische Bezeichnung für Textdateiformate wie z.B. CSV (comma separated value), durch Tabulatorzeichen getrennte Dateien, Dateien mit fester Länge, usw.

Seiten normalerweise der Fall ist (siehe Kapitel 4.1 „Fehlertoleranz bzw. Interpretationsfreudigkeit“). Eine möglichst intelligente fehlertolerante Behandlung ist normalerweise erwünscht, jedoch sollten dabei rechtliche Folgen von potenziell vorkommenden Fehlinterpretationen bedacht werden. Zu den dabei eher unkritischeren Fehlern der fehlertoleranten Interpretationen zählen normalerweise leichte Fehler nicht-inhaltsbezogener Daten (wie zum Beispiel Farbe und Layout).

v. **Konfigurations-Einfachheit**

Jede Konfiguration sollte (für Normalanwender, wie auch Superadmins) einfach und, ebenso nach Bresik [VvDwa07, S.15/17], „möglichst ohne Programmierkenntnisse“ durchführbar sein.

vi. **Erweiterbarkeit**

Meines Erachtens sollte eine gute Lösung, so weit als möglich, auf zukünftige Aufgaben vorbereitet sein. D.h. beispielsweise indem sie es ermöglicht neue Schnittstellen **mittels einfacher Konfiguration** einzubinden [vgl. VvDwa07, S.18]. Ebenso sollte die **Implementierung** (für Entwickler am Quellcode) leicht erweiterbar sein, indem sie z.B. objektorientiert¹⁵ ist und Mechanismen wie Vererbung auch konsequent und durchgängig anwendet.

vii. **Nicht-proprietäre Speicherung**

Zukunftsweisend ist die Nutzung von Formaten und Standards (für die Speicherung), welche offen und nicht-proprietär verfügbar sind. Zu nennen sind beispielsweise ODF, XML, SGML, LaTeX.

¹⁵ klassisches Beispiel: Programmiersprache Java

1.1.2 Drei Themenkomplexe - Zuordnen der Ziele

Für die weitere Arbeit wird der Bereich zwischen den allgemeinen Zielen (A) und den konkreteren Zielen (B) einer einfachen dreigliedrigen Top-down-Zielestruktur zugeordnet, welche das Voranschreiten zu konkreteren Ergebnissen (letztlich einem Prototyp) erleichtern soll¹⁶.

Zunehmend weniger abstrakte Ziele:

- (1) Business-Daten nach Office (B2O)
- (2) Business-Daten nach Symphony (B2S)
- (3) SAP-Daten nach Symphony (S2S)

Diese zunehmend weniger abstrakt werdenden Ziele sind in fast allen folgenden Kapiteln immer wieder namentlich aufgeführt.

Kommentare zu den drei obigen Zielen.

(1) Business-Daten nach Office (B2O)

Zielbeschreibung: Diskussion der grundsätzlichen Möglichkeiten, Idee für eine vollkommen allgemeine/generische Lösung und deren damit verbundenen Einschränkungen. Siehe Kapitel 4 „Machbarkeits-Analyse“.

(2) Business-Daten nach Symphony (B2S)

Zielbeschreibung:

Dieses Ziel stellt eine Einschränkung (gegenüber den in Ziel (1) genannten) Anforderungen bezüglich des Zielsystems dar. Hier wird nun keine Lösung für alle Office-Pakete mehr, sondern nur für das IBM Lotus Symphony gefordert. Diskussion der Grundsätzlichen Möglichkeiten, Idee für eine vollkommen allgemeine/generische Lösung und deren damit verbundenen Einschränkungen. Behauptung: Es könnte (mit deutlichen Komfort-Einschränkungen) gehen.

(3) SAP-Daten nach Symphony (S2S)

Zielbeschreibung:

Dieses Ziel stellt eine Einschränkung (gegenüber den in Ziel (2) genannten Anforderungen) bezüglich des Quellsystems bzw. der Datenquelle dar. Hier wird nun keine Lösung für alle möglichen Datenquellen mehr gefordert, sondern lediglich für Daten aus SAP-ERP-Systemen. Hierzu existiert ein lauffähiger Prototyp (siehe Kapitel 5 „Konzeption und Implementierung“).

¹⁶ Wünschenswert (und bis zu einem bestimmten Abstraktionsgrad durchaus möglich) wäre auch die Entwicklung von zusätzlichen Prototypen gewesen, welche eher den allgemeineren Anforderungen, wie Sie in (Ziel A) beschrieben sind, als den sehr konkreten Anforderungen von (Ziel B), entsprochen hätten. Die Erstellung solcher Prototypen war mir im Rahmen dieser Diplomarbeit mindestens zeitlich nicht möglich. Inwieweit solche überhaupt produzierbar sind, wird im Kapitel 4 „Machbarkeits-Analyse“ diskutiert.

1.2 Architektur B2O, B2S und S2S

Die übliche Schichtenabgrenzung nach Lassmann [WNfSu06, S.149] im „Bereich der Anwendungssysteme“ wird in nächster Abbildung ersichtlich.

Schichten einer Client/Server-Architektur

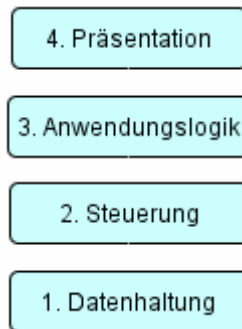


Abbildung 9: Schichten einer Client/Server-Architektur [vgl. WNfSu06, S.149]

Vereinfacht kann man die drei Architekturen (B2O, B2S, S2S), wie in folgender Abbildung darstellen. Die Integrationskomponente (Komponente in der Mitte) stellt die noch nicht auf dem Markt verfügbare noch fehlende Lösung an, welche Präsentationsschicht (Office-Paket) und Datenquelle (Business-Daten meist ERP-Systems) miteinander verbindet. Im Diagramm ist außerdem angedeutet das für die spätere Lösung unterschiedliche Anwender-Typen (Sachbearbeiter, Super-User) berücksichtigt werden sollen. Der Super-User legt die zur Auswahl verfügbaren Einstellungen für den Normalanwender fest (als unterschiedliche Szenarios bzw. Profile).

Zur Darstellungsform der Diagramme:

Jeder blaue Bereich (mit führendem Plus-/Minus-Zeichen, bzw. $\boxed{+}$ / $\boxed{-}$ -Zeichen) stellt eine Gruppe dar. Eine Gruppe kann geöffnet (der erste äußere Rahmen ist eine offene Gruppe) oder geschlossen¹⁷ sein. Geschlossene Gruppen besitzen oben links ein „ $\boxed{+}$ “-Zeichen und offene ein „ $\boxed{-}$ “-Zeichen. Geschlossene Gruppen (also Gruppen ohne sichtbaren Inhalt) zeigen, dass zu diesem Funktionsbaustein, an anderer Stelle, ein Diagramm existiert, welches deren Inhalt (also als offene Gruppe) darstellt oder weist manchmal auch nur darauf hin, dass sich hinter dem Funktionsbaustein ein komplexer Vorgang verbirgt.

¹⁷ Während der Modellierung/Diagramm-Erstellung, mittels „yEd Graph Editor“, können die Gruppen mit +/- auf bzw. zugeklappt werden.

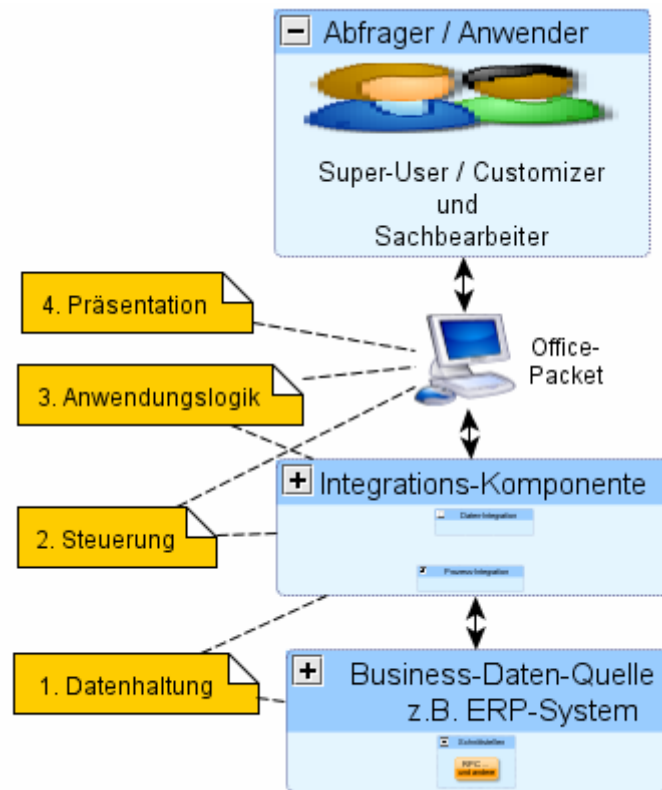


Abbildung 1: Vereinfachte Architektur Darstellung

Anmerkung: Im praktischen Teil wird nur von Business-Daten aus einem SAP-ERP-System die Rede sein und nicht mehr von Business-Daten allgemein.

Die nächsten drei Architektur-Abbildungen beschreiben je eines der Ziele aus Kapitel 1.1.2 „Drei Themenkomplexe - Zuordnen der Ziele“.

Architektur Business-Daten nach Office (B2O)

Diese Architektur-Abbildung veranschaulicht das Ziel (1) „Business-Daten nach Office (B2O)“.

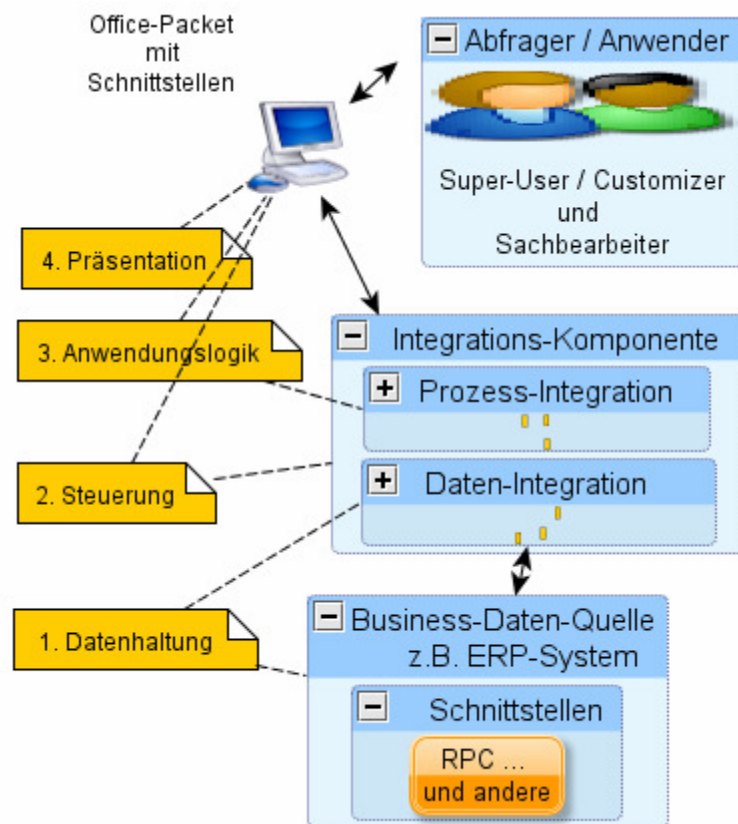


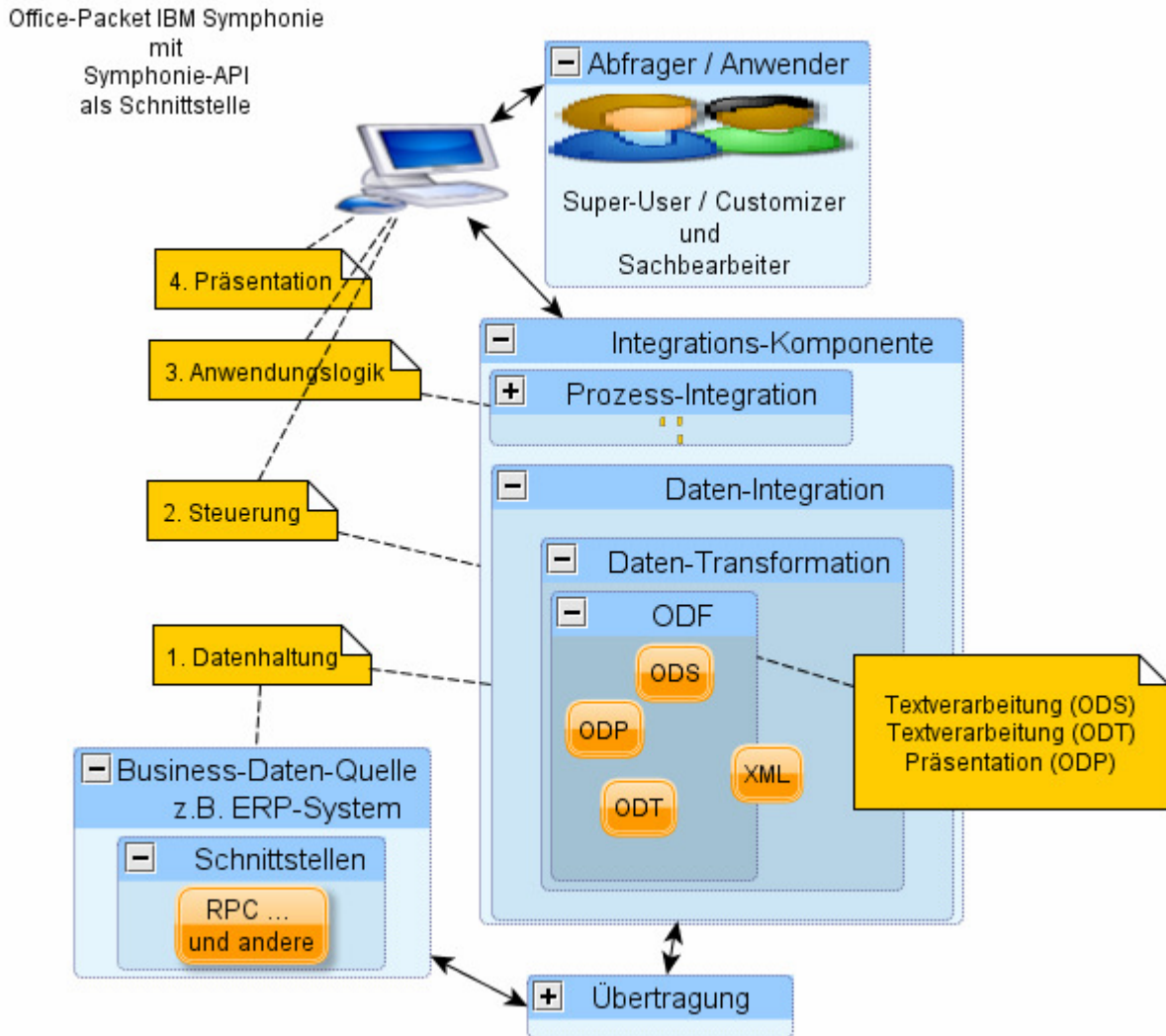
Abbildung 2: Architektur Business-Daten nach Office (B2O)

Zur Positionierung der Integration:

Im Gegensatz zur Prozessintegration ist es gut vorstellbar, dass die Datenintegration entkoppelt von der Prozessintegration implementiert wird und an diese dann potentiell unterschiedliche Prozessintegrationen/Lösungen anschluss finden können. Diese ferne Datenintegration würde so einen zusätzlichen Service, bzw. einen zusätzlichen Zugang zu den Daten liefern. Außerdem bietet sich hier die Gelegenheit zur Standardisierung von proprietären Formaten.

Architektur Business-Daten nach Symphony (B2S)

Folgende Architektur-Abbildung veranschaulicht das Ziel (2) „Business-Daten nach Symphony (B2S)“.



18

Abbildung 3: Architektur Business-Daten nach Symphony (B2S)

Obwohl nur die Übertragung zu/von den Business-Daten gesondert hervorgehoben ist, stellt im Grunde jeder Pfeil eine Übertragung von Daten dar.

Hinweis: Obwohl unter dem Begriff Business-Daten oft vereinfacht nur die Daten aus einem ERP-System verstanden werden – und dies in den meisten konkreten Fällen zutreffen könnte – ist der Begriff Business-Daten allgemeiner zu verstehen. Daten aus einem ERP-System sind als Spezialfall von Business-Daten zu verstehen (siehe Kapitel 2 „Theoretische Grundlagen“).

Architektur Daten von SAP-ERP nach Symphony (S2S)

Im Folgenden wird ein Ziel (S2S) (siehe Kapitel „1.1.2 Drei Themenkomplexe - Zuordnen der Ziele“) Architektur dargestellt.

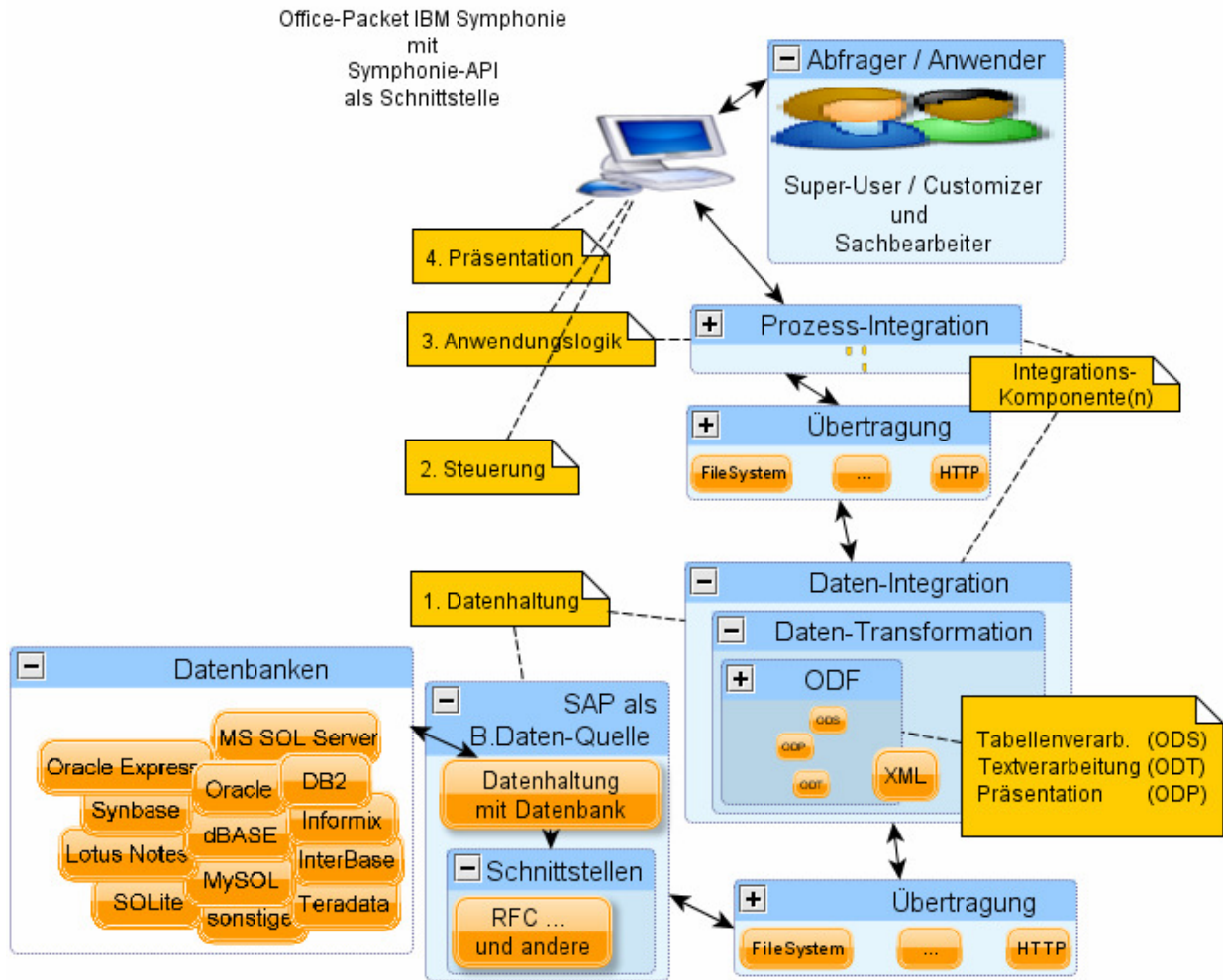


Abbildung 4: Architektur Daten von SAP-ERP nach Symphony

Anmerkung: ERP-Systeme halten Ihre Daten, in der Regel, in Datenbanken [vgl. ERPuS08, S. 9]. Insbesondere den oben am Rande angegebenen [Vdwa07, S.14]. Deren Spezifika werden jedoch normalerweise durch Schnittstellen verdeckt und so eine Unabhängigkeit von der speziellen Datenbanksprache (SQL-Varianten) erreicht. Das Bild mit den Datenbanken soll darauf hinweisen.

2 Theoretische Grundlagen

Dieses Kapitel kann auch unabhängig von anderen Kapiteln gelesen werden.¹⁹

2.1 Der Begriff Integration

„Kernthema ist heute die Integration der Daten in funktionsorientierten Anwendungen.“, sagten Teich/Kolbenschlag/Reiners in ihrem Kapitel 9.1 „Trends der IT-Treiber“ [DrWzS08, S.223]. Weil der Begriff der Integration „wesentlich bestimmend für ERP-Systeme ist“ [ERPuS08, S. 6] und es sich bei dieser Arbeit im Wesentlichen um Integration handelt, werden im Folgenden verschiedene Formen der Integration umrissen.

Aspekte der Integration:

Man unterscheidet Integration nach [vgl. IIPuK02, S.2]:

- **Reichweite**,
- **Richtung** und
- **Gegenstand**

Zur Reichweite der Integration:

Man unterscheidet zwischen Integration **innerhalb** eines Unternehmens und **unternehmensübergreifender** Integration. Die Reichweite d. I. beschreibt also, in wieweit innerhalb eines abgegrenzten organisatorischen Bereiches (eines Prozesses, eines gesamten Unternehmens oder zwischen Unternehmen, also über die Unternehmensgrenze hinaus) integriert wird. Man unterteilt daher die **Integrationsreichweite** wiederum in die **Bereichs-**, **Funktions-**, **Prozess-**²⁰ (Kopplung von Prozessen, nicht Systemen), **innerbetriebliche-** und **zwischenbetriebliche** Integration.

Zur Richtung der Integration:

Man unterscheidet in **vertikale** und **horizontale** Integration. Bei der vertikalen Integration werden für die unterschiedlichen Managementebenen unterschiedliche **Verdichtungsstufen** benötigt bzw. muss es auch möglich sein, von oben nach unten immer genauere bzw. detailreichere Information zu erhalten. Bei der horizontalen Integration werden **Teilsysteme** innerhalb der betrieblichen Wertschöpfungskette integriert (z.B. Fertigung - Vertrieb - Beschaffung - Lagerhaltung - Fertigung - Versand – Kundendienst) [vgl. IIPuK02, S.6].

Der Gegenstand der Integration kann Folgendes umfassen:

- Daten,
- Funktionen,
- Vorgänge,
- Methoden und
- Programme

Diese fünf Modelle haben im Kern die gleiche Funktionalität, sie fügen Daten zusammen und stimmen diese aufeinander ab, so dass zum Beispiel die Arbeitsvorbereitung mit den Daten aus der Konstruktion weiter arbeiten kann.

¹⁹ Die Themenauswahl im Kapitel „Theoretische Grundlagen„ ist in der Art Ihrer Formulierung möglichst allgemein gehalten. Kleinere Querverweise zu anderen Kapiteln oder kleinen Kommentaren zu einer konkreten Aufgabe, Lösung, oder Ähnlichem aus der Diplomarbeit finden sich jedoch dennoch.

²⁰ Laut dem „IC Knowledge“-Glossar [GoICT08] ist „Process Integration“ der Akt der Integration einer Reihe von Prozess-Schritten zur Schaffung eines Prozess-Flows.

Bei der Datenintegration werden Datenbestände logisch zusammengeführt. Bei der Funktionsintegration werden Aufgaben so zusammengeführt und abgestimmt, dass sie über ein gemeinsames Informationssystem aufgerufen und ausgewertet werden können. Bei der **Prozessintegration**, auch **Vorgangintegration** genannt, werden mehrere dieser Aufgaben zu einem Prozess zusammengeführt.

Tabelle 1: Drei Arten der Integration

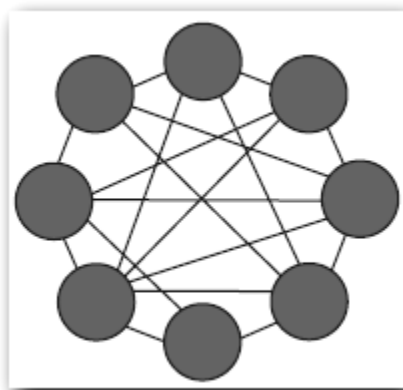


Abbildung 5: Punkt-zu-Punkt-Integration

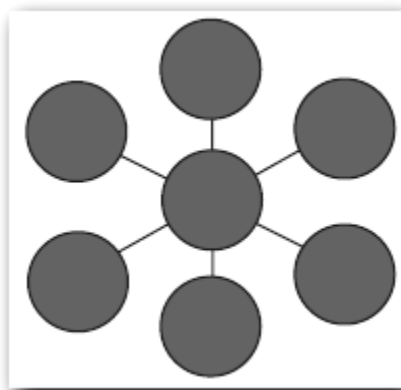


Abbildung 6: Hub-and-Spoke-Integration

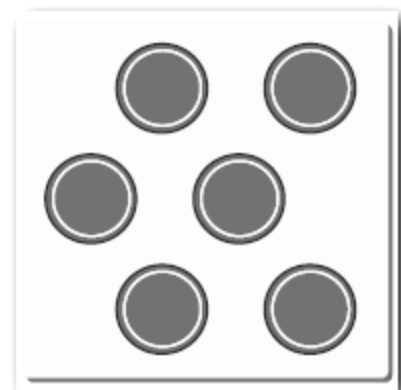


Abbildung 7: Serviceorientierte Integration

Punkt-zu-Punkt-Integration

Nach Stadlbauer [ZAlmi07, S.35] lässt sich Integration im Bereich der Applikations-Kommunikation grundsätzlich in Varianten unterteilen. Nach Stadlbauer ist die erste Variante die Punkt-zu-Punkt-Integration. Charakterisiert ist **Punkt-zu-Punkt-Integration** sei eine Verwendung von spezifischen vielen Schnittstellen zur Kommunikation zwischen jeweils zwei gleichberechtigten Anwendungssystemen. Die hohe Schnittstellenanzahl der **Punkt-zu-Punkt-Integration**, so Stadlbauer weiter [ZAlmi07, S.36] führte zur Entwicklung der Applikations-Kommunikation in Form von middlewarebasierten Integrationstechnologien. Für diese ist die Verwendung einer einzigen zentralen Integrationsplattform typisch, mit der sämtliche zu integrierende Anwendungssysteme verbunden sind. Ein Nachteil der Punkt-zu-Punkt-Integration ist, dass durch die Enge Kupplung von Anwendungen das Nachvollziehen von organisatorischen Veränderungen (z.B. bei In- und Outsourcing) erschwert ist.

Hub-and-Spoke-Integration (Speichennetz)

Bei der Hub-and-Spoke-Integration werden Anwendungen nicht untereinander, sondern über eine Art **standardisierte Middleware Plattform** integriert. Sie stellt eine Weiterentwicklung der Punkt-zu-Punkt-Integration dar. Ein Vorteil der Hub-and-Spoke-Kommunikation ist das zentrale Management, die Standardisierung und die Wiederverwendung. Nachteilhaft an diesem Ansatz ist die relativ aufwendige Einrichtung und Pflege der peripher angeordneten Anwendungen, da die Anwendungen möglicherweise über proprietäre APIs verfügen, welche bei jeder Änderung zusätzlichen Entwicklungs- und Pflegeaufwand erfordern.

Serviceorientierte Integration

Serviceorientierte Architektur (SOA) ist der modernste Ansatz. Für die Implementierung einer SOA müssen die Anwendungen erkennbare, implementierbare und zuverlässige Services bereitstellen.

Durch die einfache Wiederverwendbarkeit von Services und dem einfachen Zusammenbau neuer Geschäftsprozesse sind die entstehenden Kosten beim SOA-Ansatz relativ gering. Mehr zu diesem Thema im nächsten Kapitel und dem Kapitel 5.4.3 „Überlegungen zur Wahl der Kommunikationsform mit SAP“.

2.2 Web-Service

„Web-Services“ ermöglichen den Daten-Austausch zwischen zwei entfernten Anwendungen, welche nicht zwingend auf den gleichen Technologien basieren müssen. Die Kommunikation eines Web-Services basiert stets auf Internet-Protokollen.

Zu dem Begriff Web-Services existieren zahlreiche Definitionen. Das World Wide Web Consortium (W3C) definiert einen Web-Service als: “A software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web-Service in a manner prescribed by its description using SOAPmessages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.” [WSAww04, S.7].

Nach Schädler [EvWSi07, S.32] bieten fast alle renommierten Hersteller von betriebswirtschaftlicher Standardsoftware Web-Services fähige Produkte an.

Entscheidend für die Kommunikationsbeziehung via Web-Service sind dabei folgende Charakteristika:

Programmierbarkeit: Auf Web-Services kann über programmierbare Schnittstellen zugegriffen werden. Gedacht und verwendet werden diese in erster Linie zur Kommunikation von Softwareanwendungen und nicht für die manuelle menschliche Informationsverarbeitung, daher haben Web-Services auch kein graphisches Benutzerinterface.

Selbstbeschreibung: Während der Laufzeit können Metadaten (selbst beschreibende Daten) jedes Web-Services abgefragt werden. Typische Beschreibungselemente sind Name, Beschreibung, Version und Dienstgüte des Services.

Kapselung: Jeder Web-Service ist eine unabhängige, in sich abgeschlossene – mit anderen Worten gekapselte Anwendung – mit klar definierter Aufgabe.

2.3 Eclipse

Der mit dieser Arbeit erstellte Prototyp nutzt als Entwicklungsplattform (IDE²¹) Eclipse und auch die Zielplattform, IBM Lotus Symphony, basiert auf Eclipse. Vielen ist Eclipse nur als IDE bekannt, obwohl Eclipse eine vom Betriebssystem unabhängige Plattform ist, auf der neben Entwicklungsumgebungen auch Client-Applikationen betrieben werden können.

Die bekannteste Verwendung von IDE ist die Nutzung für die Programmiersprache Java. Auch Rich-Client-Applikationen werden auf Basis der Eclipse Rich-Client-Plattform (RCP) entwickelt. Typisch für die Offenheit und Flexibilität von Eclipse ist, dass dafür eine Vielzahl von Plug-ins aus OpenOffice-Source-Projekten, aber auch von kommerziellen Anbietern existieren. Außerdem steht Eclipse unter der Common Public License (CPL) und ist damit für Jeden offen verfügbar. Die CPL ist eine „Freie-Software-Lizenz“²² und gewährt daher das Recht zur freien Nutzung, Weiterverbreitung und auch Veränderung der Software.

2.4 Office-Paket

Bächle [EidW07, S.25] nennt ein „Office-Paket“ auch „integriertes Bürosoftwarepaket“ und definiert dies folgendermaßen:

„Ein integriertes Bürosoftwarepaket (auch: Office-Paket) ist eine Zusammenstellung aufeinander abgestimmter Bürosoftwarekomponenten, welche unterschiedliche Büroaufgaben erledigen oder zumindest unterstützen sollen.“

Und auch Wikipedia²³ definiert fast identisch [Offic08]:

„Office-Pakete sind Zusammenstellungen gebräuchlicher Software für Arbeiten im Büro, welche unterschiedliche dort anfallende Aufgaben erledigen oder zumindest unterstützen sollen.“

Office-Pakete sind also „Zusammenstellungen“ von gebräuchlicher bzw. abgestimmter (Büro-) Software für das Arbeiten im Büro. Zu solchen „Zusammenstellungen“ gehören, laut Bächle [EidW07, S.25], „in der Regel Textverarbeitungs-, Tabellenkalkulations-, Präsentations- und (kleinere) Datenbank-Programme“.

Es existieren sehr viele Office-Pakete. Trotzdem findet man meist nur einen Vergleich zwischen zwei bis fünf von solchen Paketen. Eine „Liste von Office-Paketen“ findet sich im Anhang auf Seite 88.

Ebenso umfangreich ist die Anzahl der von diesen Office-Paketen unterstützten Formate. Allerdings scheint sich das ODF-Format weiter durchzusetzen. Sogar Microsoft unterstützt dieses Format (siehe Kapitel 3.4 „Der Trend zum Open-Source“). Die derzeit (noch) am weitesten verbreiteten Formate sind die proprietären Dateiformate des Marktführers im Bereich Office-Paketen, nämlich Microsoft (MS Office) [EidW07, S.25].

²¹ Eine IDE (Integrated Development Environment) ist ein Anwendungsprogramm zur Entwicklung von Software.

²² Der Sinn einer Freie-Software-Lizenz besteht im Wesentlichen darin, eine möglichst freie Verwendung der Software für jetzt und für die Zukunft zu sichern.

²³ Wikipedia® (Die freie Enzyklopädie)

Zwei Beispiele alternativer, nicht-proprietärer Office-Pakete sind „IBM Lotus Symphony“ und „OpenOffice“ (siehe die „Liste von Office-Paketen“ im Anhang).

Die Integration mit Office-Paketen wird oftmals (so im Falle von MS-Office, OpenOffice und IBM Lotus Symphony) durch eine API des entsprechenden Office-Paketes unterstützt.

2.5 Der Begriff Business-Daten

Es ist nicht klar definiert, was Business-Daten sind und auch Aussagen darüber sind schwer zu finden. Einfacher ist es zu sagen (ex negativo), was Business-Daten nicht sind. In einem Glossar [♣PDSSG08, S.2] werden diese mit „It is not metadata.“ beschrieben, was technisch gesehen (bei der Frage nach dem Datentyp) jedoch auch keine Abgrenzung/Einschränkung darstellt.

So gesehen können Business-Daten alle Arten von Daten sein (verschiedenste Formate, Bilder, Klänge, Videos, etc). Es ist nicht einmal definiert, ob sie immer als eine Bit-Folge darstellbar sein müssen, bzw. in digitaler Form vorliegen. Demnach unterscheidet sich der Ausdruck Business-Daten also lediglich auf semantischer Ebene vom Begriff Daten und nicht in technischer Hinsicht.

2.6 Enterprise Resource Planning

Der Begriff ERP

Unter „Enterprise Resource Planning“ (ERP) versteht man die Planung des Einsatzes/ der Verwendung von Unternehmensressourcen. Das Ziel von „Enterprise Resource Planning“ ist, „die in einem Unternehmen vorhandenen Ressourcen [z.B. Kapital, Betriebsmittel, Personal, Anm. d. Verf.] möglichst effizient zu nutzen“ [vgl. ☐EU07, S.182 und ♣Erp08].

Das ERP-System

Ein ERP-System ist eine komplexe Software, welche die Geschäftsprozesse, d.h. die ERP-Tätigkeit eines Unternehmens unterstützt.

Klassischerweise zählt das „ERP-System“ (z.B. SAP, JD Edwards, ...) zu den „Back-Office“-Systemen [☐CSmEK02, S.63/64/78], d.h. zu den Büroanwendungen im Hintergrund, wie z.B. allen nicht-kundenbezogenen Anwendungen. Sie enthielten, so Meyer 2002 [☐CSmEK02, S.63], „keine Front-Office-Funktionen“, sondern lediglich die klassischen Module wie „Produktion, Finanz- und Betriebsbuchhaltung, human Resources etc.“. Eine klare Trennung in „Back-Office“ und „Front-Office“, sei jedoch, so Lassmann im Jahre 2006, zunehmend schwieriger [☐WNfSu06, S.130] zu ziehen.

Jacob nennt verglichen mit Mayers [☐CSmEK02, S.63 oben], in seinem deutlich aktuelleren Werk aus dem Jahre 2008, wesentlich mehr typische Module.

So decken ERP-Systeme, nach Jacob, eine Vielzahl von Geschäftsprozessen eines Unternehmens aus den Bereichen „Finanz- und Rechnungswesen, Logistik, Vertrieb, Service Management, Produktion, Instandhaltung, Qualitätsmanagement und Human Resources ab.“ [☐EVSVM08, S. 1]

Folgende Abbildung veranschaulicht dies:

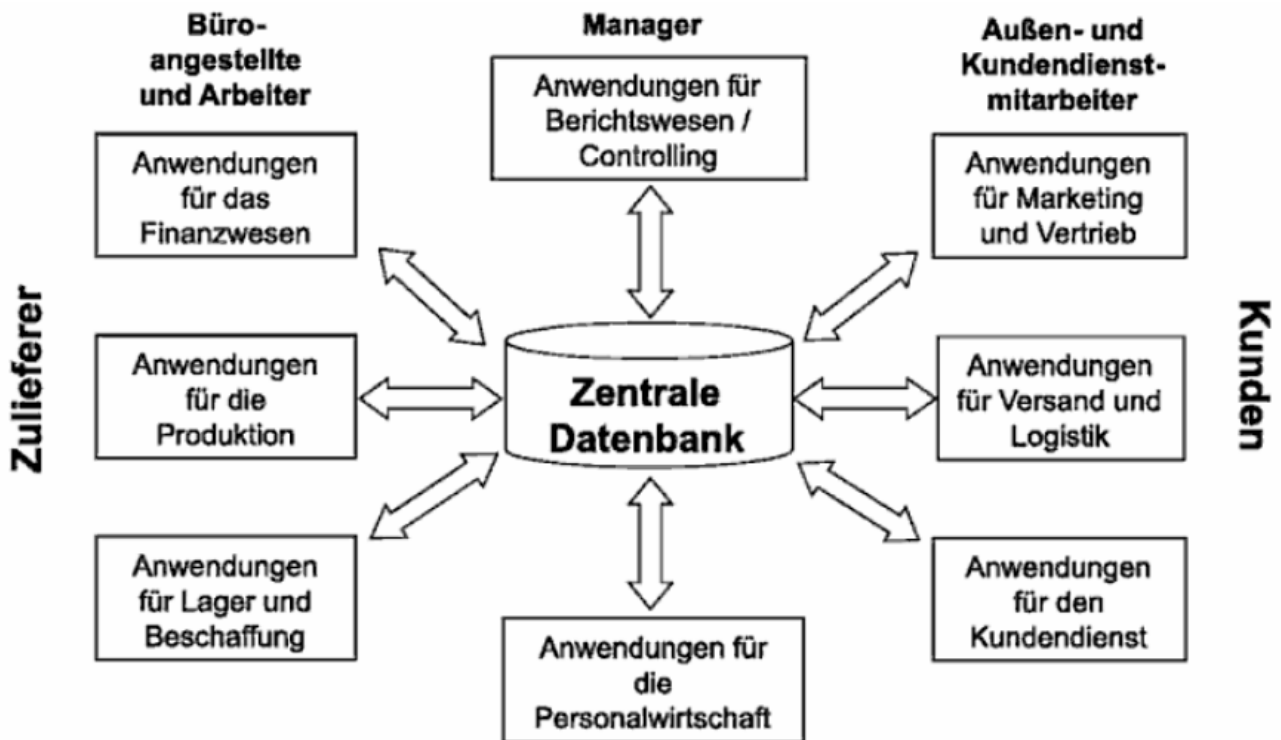


Abbildung 8: ERP-Konzept [GWAME04, S.165]

Ein Anwendungssystem (auch ein ERP-System) wird üblicherweise in Schichten eingeteilt, um die oftmals komplexen Systeme übersichtlich strukturiert eingeteilt in Abstraktionsebenen darstellen zu können. Nach [WNfSu06, S.149] gelten zwischen den Schichten „strenge Maßstäbe für die Zugriffe“ und „die Änderbarkeit, die Wartbarkeit, die Portabilität und die Testbarkeit“ werden dadurch unterstützt.

Die übliche Schichtenabgrenzung nach Lassmann [WNfSu06, S.149] im „Bereich der Anwendungssysteme“ wird in nebenstehender Abbildung ersichtlich:

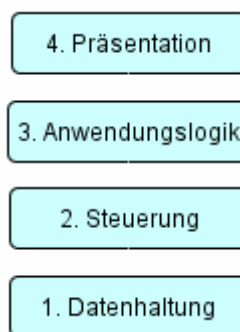


Abbildung 9: Schichten einer Client/Server-Architektur [vgl. WNfSu06, S.149]

Nach Gronau [ERPuS08, S.17] sind an der Bearbeitung einer Geschäftstransaktion immer mindestens zwei Computerprozesse (Client und Server) beteiligt.

Tabelle 2: Mögliche Verteilung von ERP-Aufgaben zw. Client und Server

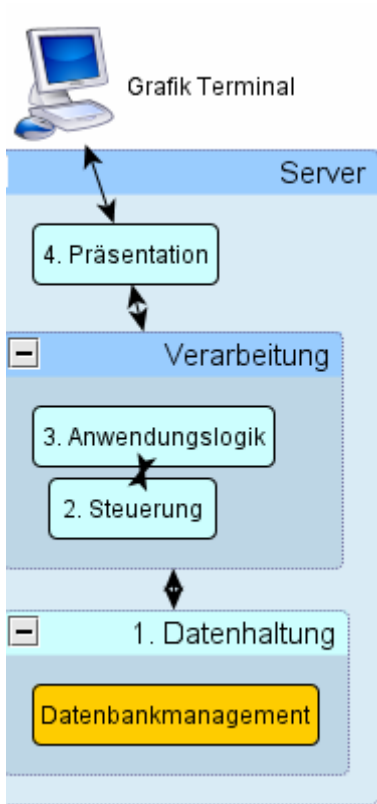


Abbildung 10: ERP-Architektur "Zentrale Präsentation"

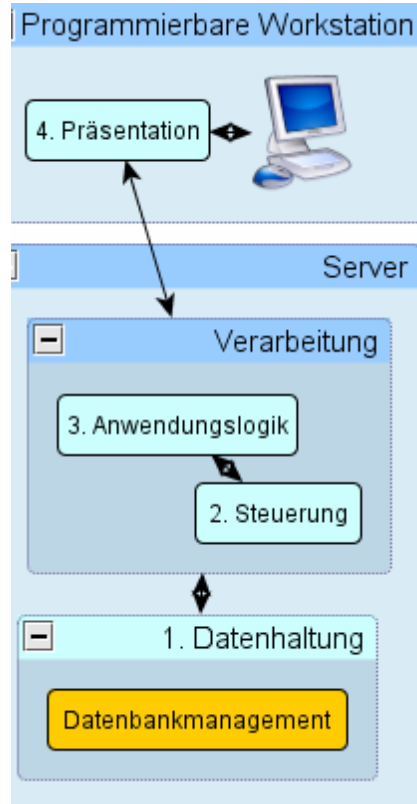


Abbildung 11: ERP-Architektur "Dezentrale Präsentation"

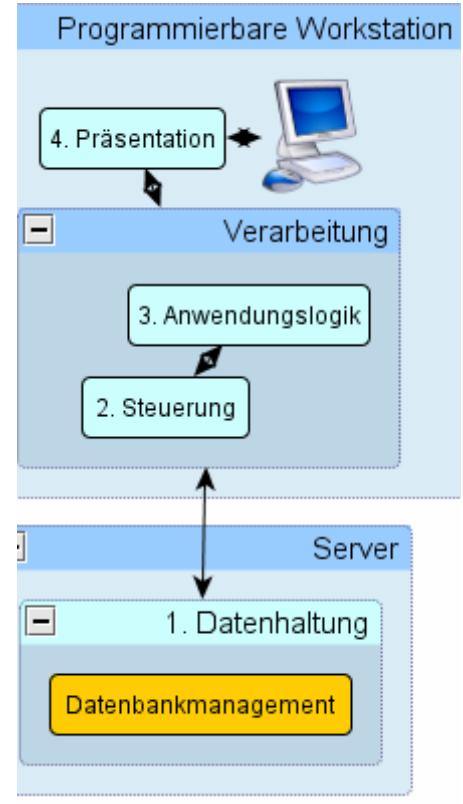


Abbildung 12: ERP-Architektur "Dezentrale Verarbeitung mit zentraler Speicherung"

ERP Systeme haben typischerweise [vgl. ERPuS08, S.9] auch eine Programmierumgebung – und zwar auf der Ebene der Applikationsschicht/Verarbeitungsschicht. Dort lassen sich mittels einer (meist proprietären²⁴) Programmiersprache Anwendungen des Systems modifizieren bzw. erweitern.

Zu den am häufigsten genutzten Programmierschnittstellen eines ERP-Systems zählen die so genannten „Remote Procedure Calls“²⁵ [ERPuS08, S.9], über die man das System von außen ansprechen kann. Nach Gronau verfügen alle ERP-Systeme über solche RPCs zur Kommunikation von System zu System [ERPuS08, S.9]. Nekolar [pEuR02, S.55] definiert RPC als Protokoll (später als entferntem Prozeduraufruf), in seinem folgenden Text:

„Unter einem Remote Procedure Call wird ein Protokoll verstanden, welches ein Programm in die Lage versetzt, eine Funktion eines anderen Programmes [sic! Richtig: Programms, Anm. d. Verf.] aufzurufen, unabhängig von den dahinterliegenden Netzwerkdetails.“ Zu einer SAP-eigenen Implementierung von RPC siehe im Kapitel 2.7.3 „Synchrone Kommunikation mit dem SAP-System“.

²⁴ Beispielsweise wird im Fall von SAP-ERP-Systemen dazu seit 1990 meist die Programmiersprache ABAP verwendet und seit der Einführung eines Java-basierten SAP-NetWeaver-Applikationsserver ist ebenfalls dazu die Programmiersprache Java möglich (siehe SAP NetWeaver)

²⁵ Nach Nekolar [pEuR02, S.55] kann der Begriff „Remote Procedure Call“ mit entferntem Prozeduraufruf übersetzt werden.

Zusätzliche Anpassung von ERP-Systemen, und zwar an die Anpassung der Funktionalität von ERP-Systemen an die abgebildeten betrieblichen Prozesse und Datenstrukturen, bietet (üblicherweise) eine Customizing-Schicht [☐ERPuS08, S.9].

Trotz dieser vielen Anpassungsmöglichkeiten hat ein Kunde oft viele noch nicht integrierten Anwendungen im Einsatz. Dies sind beispielsweise oftmals E-Mail-Clients, Anwendungen zur Textverarbeitung oder speziellere Anwendungen wie CAD²⁶. Daher ist das Thema Integration generell immer von Interesse.

Im Anhang befindet sich eine Liste der ERP Systeme (siehe Liste von ERP Software-Systemen und eine Tabelle mit Marktanteilen von ERP-Software).

ERP-Datenhaltung

ERP Systeme setzen in der Regel auf einem Datenbanksystem als Datenhaltung auf, dessen Spezifika allerdings verborgen bleiben [vgl. ☐ERPuS08, S. 9].

Zu den wichtigsten Datenquellen für ERP-Lösungen zählen, nach Bresik [☐VvDwa07, S.14], die nebenstehenden Datenbanken²⁷:

DB2, Oracle, Oracle Express, MS SOL Server, Synbase, Informix, dBASE, InterBase, Lotus Notes, MySQL, SOLite, Teradata sowie sonstige Datenbanken.

Auf der Ebene der Datenbankmanagementsysteme befinden sich „in der Regel“, so Gronau [☐ERPuS08, S.9], auch Schnittstellen, welche so (durch Ihre Existenz) zur Modularisierung beitragen, d.h. so den Zugriff zu anderen Datenbanken vereinfachen. Auch in dieser Diplomarbeit wird nicht direkt mit der Datenbank kommuniziert.

²⁶ CAD (Computer Aided Design) steht für das Entwerfen von Produkten mit dem Computer.

²⁷ Wiederverwendet wird diese bildhafte Darstellung der Datenbanken in einem späteren Architektur-Diagrammen (zwecks besserer Wiedererkennung was mit einer reinen Listendarstellung nicht möglich gewesen wäre).

2.7 Das SAP-ERP-System

SAP ist weltweit führender Anbieter von Geschäftslösungen [vgl. [IM06](#), S.155 und im Anhang „Marktanteile ERP-Software“]. Eine mehrere Seiten starke Liste von SAP-Lösungen und Anwendungen findet sich auf <http://www.sap.com/germany/solutions> (z.B. Lösungen und Anwendungen in den Bereichen Applikations- und Plattform- Integration, Supply Chain Management oder klassischem ERP etc.).

In dieser Arbeit wird unter SAP-ERP²⁸-Lösungen nicht (bzw. nicht nur) das Produkt „SAP ERP“ verstanden, sondern alle ERP- bzw. Geschäftslösungen- Lösungen von SAP.

Klassische SAP Enterprise Resource Planning Business-Lösungen, wie das Produkt „SAP ERP“, sind durch einen modularen Aufbau gekennzeichnet und werden meistens in den Bereichen Finanzbuchhaltung, Controlling, Produktion und Logistik eingesetzt. Das Produkt „SAP ERP“ umfasst vier Lösungen: Personalwirtschaft, Finanzwesen, Logistik und Corporate Services. Den für ERP-Systeme typischen modularen Aufbau zeigt die Abbildung 9 auf S.21 namens ERP-Konzept.

SAP verfügt zu jedem dieser Module über eine Reihe von so genannten Businessobjekten, welche zum jeweils zugehörige Operationen und Schnittstellen zur Verfügung stellen (siehe Kapitel 4.5 „Daten aus dem SAP-System“ auf S.40).

Schnittstellen zu ERP-Systemen

Eine Schnittstelle oder das Interface (englisch „Grenzfläche“) ist der Teil eines Systems, welcher der Kommunikation dient. Der Begriff stammt ursprünglich aus der Naturwissenschaft und bezeichnet die physikalische Phasengrenze zweier Zustände eines Mediums [[DSodl08](#)].

Es existieren zum einen **allgemeine Schnittstellen**, die jeweils als offenes Interface konzipiert sind, und zum anderen **speziellere**, z.B. zur Kommunikation mit bestimmten Programmen (wie beispielsweise Microsoft Office oder Microsoft Project). Oftmals bestehen speziellere Schnittstellen im Kern wiederum aus allgemeineren Schnittstellen. Beispielsweise nutzt die BAPI-Kommunikation von SAP selbst wieder die RFC- Schnittstelle von SAP.

Nach Bresik [[VvDwa07](#), S.14] werden heute [2007, Anm. d. Verf.] größtenteils indirekte Schnittstellen (als Voraussetzung für Integration) wie z.B. ODBC verwendet, „da diese nicht laufend an die Systeme angepasst werden müssen“. Diese seien sowohl in Großunternehmen als auch in mittelständischen Unternehmen weit verbreitet. Außerdem, so Bresik weiter, sollten die SAP Schnittstellen BAPI sowie OLE DB bzw. OLE DB FOR OLAP unterstützt werden. Von allen ERP-Systemen, so Bauer [[CfIKu06](#), S.7], würden „standardmäßig“ „Office- Schnittstellen [...]

²⁸ Nach Meyer [[CSmEK02](#), S.63/64/78] zählt ein „ERP-System“ (z.B. SAP, JD Edwards, ...) zu den „Back-Office“- Systemen und enthält (im klassischen Sinne) „keine Front-Office-Funktionen“, sondern lediglich die klassischen Module wie „Produktion. Finanz- und Betriebsbuchhaltung, human Resources [sic! korrekt „human resources“, Anm. d. Verf.] etc.“. Allerdings verwischt die Trennung zwischen Front-Office- und Back-Office-Funktionen mittlerweile [siehe [WNfSu06](#), S.130].

angeboten.“ Gewünscht seien [ebenso nach Bresik] auch Schnittstellen zu Daten aus dem Internet oder zu so genannten „Flat Files“²⁹.

2.7.1 Kommunikation mit dem SAP-ERP-System

Wie jedes ERP-System [ERPuS08, S.9] verfügt auch ein SAP-System über die Möglichkeit, über so genannte „Remote Procedure Calls“ (RPCs), die Kommunikation zwischen den einzelnen Systemen und externen Programmen durchzuführen. Allerdings wird im SAP-Umfeld weniger von RPCs gesprochen, sondern eher von deren Implementierungen, wie RFCs oder BAPIs, welche auf die RFC-Technik (oder im Falle asynchroner Datenübertragung im sendenden System auf die generierte ALE-IDoc-Schnittstelle) aufsetzen. Alle SAP-Lösungen (die auf einen SAP Kernel ähnlichen Typs aufsetzen), nicht nur „SAP ERP“, verfügen über die gleichen Schnittstellen für den Datenaustausch, sowohl innerhalb des betreffenden SAP Systems, wie auch außerhalb, d.h. also auch zwischen einem SAP-System und Nicht-SAP-System (bzw. externen Programm).

Dabei wird der Datenaustausch eines SAP-Systems entweder synchrone oder asynchrone kommuniziert (siehe folgende Kapitel). Häufiger würde der synchrone RPC verwendet, so Nekolar [pEuR02, S.55].

2.7.2 Schnittstellen zu SAP-ERP-Systemen

ERP-Systeme, nicht nur „SAP ERP“, bieten meist **mehrere Schnittstellen** für die Verbindung zu einzelnen Modulen. Diese unterteilt man in **spezielle** Schnittstellen (für die Kommunikation mit bestimmten Programmen wie beispielsweise „Microsoft Office“, „OpenOffice“ oder „IBM Lotus Symphony“) und Schnittstellen, welche für die **allgemeine** Benutzung und als offene Interfaces konzipiert sind (beispielsweise RFCs oder BABIs).

BAPI

Für die meisten Komponenten eines SAP-Moduls stehen **BAPIs** zu Verfügung. Mittels BAPIs ist der kontextspezifische Informationsaustausch mit einem „SAP ERP-System“ in Bezug auf das entsprechende „Business-Object“ der gewählten Komponente (des Moduls) möglich.

Eine Alternative zu BAPIs stellt die Daten-Übertragung über die **ALE-Schicht** (ALE-Distribution Model) dar. Sie ermöglicht oftmals eine einfache Übertragung an ein anderes SAP-System oder an ein fremdes System, wobei jedoch die Übertragung in die umgekehrte Richtung teilweise beschränkt ist.

ALE

Die „Application Link Enabling“ (ALE) Technologie, wurde ursprünglich von SAP eingeführt, um auf verschiedenen Unternehmensstandorten verteilte R/3-Systeme als ein logisches Gesamtsystem zugreifen zu können. Die Verteilung von Anwendungsmodulen und Datenbanken auf verschiedene Hardwaresysteme wird damit möglich, ohne dass der Endanwender davon etwas bemerkt. Die ALE-Technologie kann aber auch zur Verbindung externer Systeme mit SAP R/3 eingesetzt werden.

²⁹ Nach http://www.virenschutz.info/Flat-file-Techniklexikon-bei-Virenschutz-info_690.html ist mit Flat file ist eine Datei gemeint, welche nur eine kleine Struktur aufweist und besonders im World Wide Web zu finden ist.

Nach http://www.altova.com/de/dev_portal_flatfile_mapping.html :

„Flat File“ ist eine generische Bezeichnung für Textdateiformate wie z.B. CSV (comma separated value), durch Tabulatorzeichen getrennte Dateien, Dateien mit fester Länge, usw.

IDOC

Eine Möglichkeit, um über Schnittstellen und Nachrichtentypen Daten in SAP einzuspielen, stellt die Verwendung von **IDOCs** (Intermediate Documents) dar. Diese Nachrichtentypen haben ihren Ursprung in EDI (Electronic Data Interchange), mit welchem heutzutage Bestellungen oder Lieferabrufe versendet werden. Nach Jung [DuDvA08, S.29] sind IDoc-Nachrichten und BAPIs besonders für fortlaufende Datenübernahmen geeignet. Möchte man mit IDocs arbeiten, so ist eine sogenannte „Partnervereinbarung“ Voraussetzung und der entsprechende Partner, mit dem Daten ausgetauscht werden sollen, muss definiert sein.

Web-Services

Nach Schädlers Aussage vom Juli 2007 [EvWSi07, S.32] sind „Web-Services“ Implementierungen von SAP noch nicht durchgängig in alle Module und Produkte integriert. Meines Erachtens wird dies aber nicht mehr lange dauern, denn nach Jeske [SfJeK05, S.209] sind Web-Services „nicht nur eine modische Erscheinung“. Und weiter „SAP hat diese technologische Strömung konsequent aufgegriffen und auch mitgestaltet“. Die SAP „Web-Services API“-Schnittstelle unterstützt den Zugriff auf eine Reihe von „Business-Objecten“ und Klassen mithilfe der modernen Programmiersprachen wie Java, C# oder C++. Zu den Charakteristika eines Web-Service gehören Programmierbarkeit, Selbstbeschreibung und Kapselung, welche im Kapitel 2.2 „Web-Service“ beschrieben sind.

Damit Web-Services tatsächlich zum Erfolg beitragen, müssen die Geschäftsprozesse von Beginn an so gestaltet werden, dass sie anderen Unternehmen ohne grundlegende Probleme über standardisierte Schnittstellen zur Verfügung gestellt werden können. Hierzu sind als anerkannte Standards WSDL, SOAP und UDDI zu nennen.

2.7.3 Synchrone Kommunikation mit dem SAP-System

Der häufigste RPC (nicht zu verwechseln mit RFC³⁰) sei, zumindest 2002, der synchrone RPC, nicht der weniger bekannte asynchrone RPC (z.B. qRFC), so Nekolar [pEuR02, S.55]. Ein RPC ist ein Spezialfall eines PCs (Procedure Call). Für den Programmierer unterscheidet sich der RPC vom PC in seiner Erscheinung während des Codierens nicht. Dies macht auch den besonderen Charme von RPC aus. So merkt man (äußerlich) keinen Unterschied zum PC.

Bei synchroner Kommunikation wartet der Sender/Aufrufer so lange, bis eine Rückmeldung vom Empfänger/Adressaten zurückgekommen ist. Man spricht hierbei, nach Nekolar [pEuR02, S.55], von einem „blockierenden (synchronen) RPC Aufruf“.

RFC ist die SAP-eigene Implementierung von RPC [dSaITU04, S.110]. RFC wird in erster Linie (und manchmal fälschlicherweise ausschließlich) mit synchroner Kommunikation in Verbindung gebracht [SfJeK05, S.107]. Es existiert jedoch auch ein spezieller RFC-Typ, welcher asynchron arbeitet, namens „queued RFC“ (qRFC). Der „Transactional RFC“ (tRFC) kommuniziert dagegen wieder synchron.

Ursprünglich wurde eine RFC-Kommunikation mit einem SAP-System über eine C-Bibliothek realisiert, die das Fremdsystem einbinden musste [SfJeK05, S.107]. Mittlerweile verwendet man meist Mechanismen/Schnittstellen, welche das RFC-Protokoll nur intern verwenden und nach außen verdecken [SfJeK05, S.107].

Spricht man von einer **RFC-Schnittstelle**, meint man meist einen RFC-Connector, d.h. man meint also eine Schnittstelle/ Bibliothek, welche zwar intern eine C-basierte RFC-Bibliothek nutzt, aber nach außen hin anders zu verwenden ist [SfJeK05, S.107].

Am häufigsten wird für die synchrone Kommunikation per RFC ein Wrapper verwendet. Meist wird der Java-Connector (JCo) verwendet [SfJeK05, S.119]. Der Java-Connector ist kostenfrei und gehört zum SAP-Package dazu. Damit kann ein externes Java Programm so implementiert werden, dass es auf BAPIs zugreifen kann. RFCs können unter anderem auch über die von Microsoft definierten Schnittstellen COM und OLEDB benutzt werden.

BAPIs verwenden, im Falle von synchroner Kommunikation, intern das RFC – Protokoll und stehen für die meisten SAP-Module zur Verfügung. BAPIs bestehen aus Operationen auf Businessobjekten eines SAP-Moduls.

³⁰ Der RFC ist eine Implementierung des RPC.

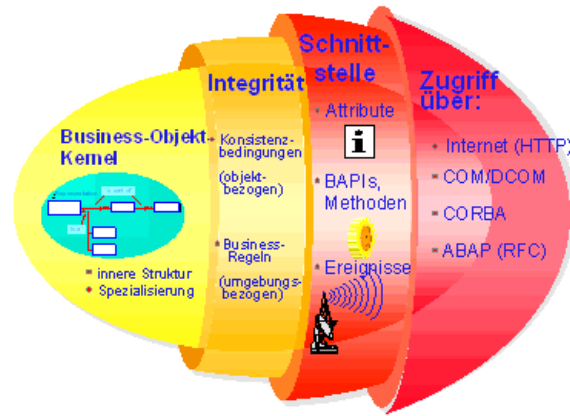


Abbildung 13: Allgemeine Struktur eines Business-Objekts [mHTGu03]

Die heutzutage meist verwendete RFC-Schnittstelle ist der sogenannte „**SAP® Java Connector**“ (JCo) [siehe SfJeK05, S.107 und Kapitel 2.7.4 „Der SAP® Java Connector“]. Es gibt viele Implementierungen der C-basierten-RFC-Schnittstelle für Fremdsystem auch von „kleinen Fremdanbietern“ [SfJeK05, S.107].

2.7.4 Der SAP® Java Connector

Wie bereits in vorigen Abschnitten erwähnt wird für die synchrone Kommunikation per RFC meist ein Wrapper verwendet. Am häufigsten wird der Java-Connector (JCo) verwendet [SfJeK05, S.119]. Dieser so genannte „SAP® Java Connector“ ist im SAP-Package enthalten. Damit kann ein externes Java Programm so implementiert werden, dass es auf BAPIs zugreifen kann.

Die synchrone Kommunikation in der nächsten Abbildung ist auf RFC- oder BAPI-Basis dargestellt. In einer Java-Anwendung werden die BAPI- oder RFC-Aufrufe mit Hilfe SAP® Java Connector (JCo) realisiert. In unterem Modell übernimmt SAP ERP die Rolle des Servers.

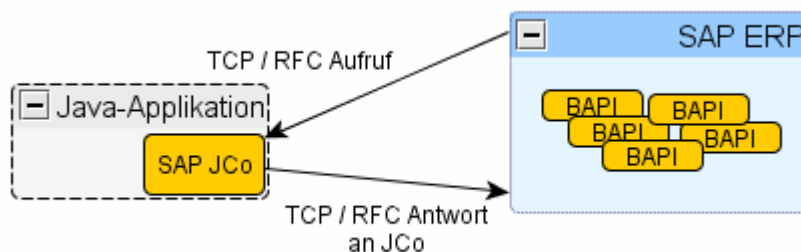


Abbildung 14: Typische Synchrone SAP-Kommunikation

3 Ist-Analyse

Integrationsversuche zur softwaregestützten Integration von Business-Daten in einem Officedokument existieren seit langem. In diesem Geschäft ist derzeit einiges in Bewegung. Im Folgenden der Versuch eine möglichst umfassende Momentaufnahme der sich derzeit rasch ändernden Möglichkeiten darzustellen.

Bezogen auf die Aufgabenstellung stellt sich zuerst einmal die Frage danach, woher Business-Daten kommen können (Quelle) und zum anderen, welche Office-Pakete (Ziel) benutzt werden, beziehungsweise auch, welche Office-Pakete überhaupt existieren.

Führende Anbieter (von ERP Systemen) sind, nach Hain [EU07, S.162], in fallender Reihenfolge SAP, Oracle, Sage und Microsoft. Zusätzlich wird freie und OpenOffice-Source lizenzgebundene ERP-Software angeboten. Diese Reihenfolge spiegelt sich auch in den entsprechenden Marktanteilen wieder (siehe im Anhang unter „Marktanteile ERP-Software“).

Quellen für Business-Daten können prinzipiell alle Datenquellen sein, welche Business relevant sind, da der Begriff Business-Daten so allgemein und unscharf ist (siehe Kapitel 2.5 „Der Begriff Business-Daten“).

3.1 Die Office-Pakete (Ziel der Daten)

Office-Pakete werden als allgemein als Standardsoftware bzw. "standardisierte Software-Produkte der Bürokommunikation" bezeichnet [WNfSu06, S.169].

Es existieren sehr viele Office-Pakete; jedoch nicht mehr als ca. 100 verschiedene (schließt man beispielsweise Varianten des OpenOffice wie z.B. Go-oo³¹ mit ein). Jedoch findet man meist nur Vergleiche zwischen zwei bis etwa fünf solcher Pakete. Allerdings ist vielen Office-Paketen gemeinsam, dass sie das ODF-Format unterstützen. ODF scheint sich durchzusetzen. Sogar Microsoft unterstützt dieses Format.

Die (noch) Quasi-Standards für Office-Dokumente sind die Microsoft-Formate. Bei der Recherche (siehe die „Liste von Office-Paketen“ im Anhang) fiel auf, dass sich die meisten Office-Pakete (z.B. bezüglich ihrer Import-Funktionalität) meist auf die Hersteller Microsoft und OpenOffice bezogen.

Im Bereich der Online-Offices tauchte der Name „Google Text & Tabellen“ am häufigsten auf. Jedoch meinen einige Benutzer, dass „Zoho Sheet/Writer/SQL“ im Online-Officebereich deutlich besser als Google wäre³². Übrigens seien diese („die Online-Offices“) so Rathert [KfMW08] von einer „richtigen Bürosuite noch weit entfernt. [...] Da fehlen noch zwei, drei Entwicklungsjahre.“
Warnung: Unsere Daten sind bei solchen Anbietern nicht unbedingt sicher (Stichwort gläserner User).

³¹ Go-oo ist eine Abwandlung von OpenOffice.org mit einigen Extras.

³² Besonders deutlich äussert sich beispielsweise Dieter Petereit im August 2007: „ZOHO Writer deklassiert Google Text & Tabellen“ auf digitallife.germanblogs.de.

3.2 Erkennbare Trends bei Office-Paketen

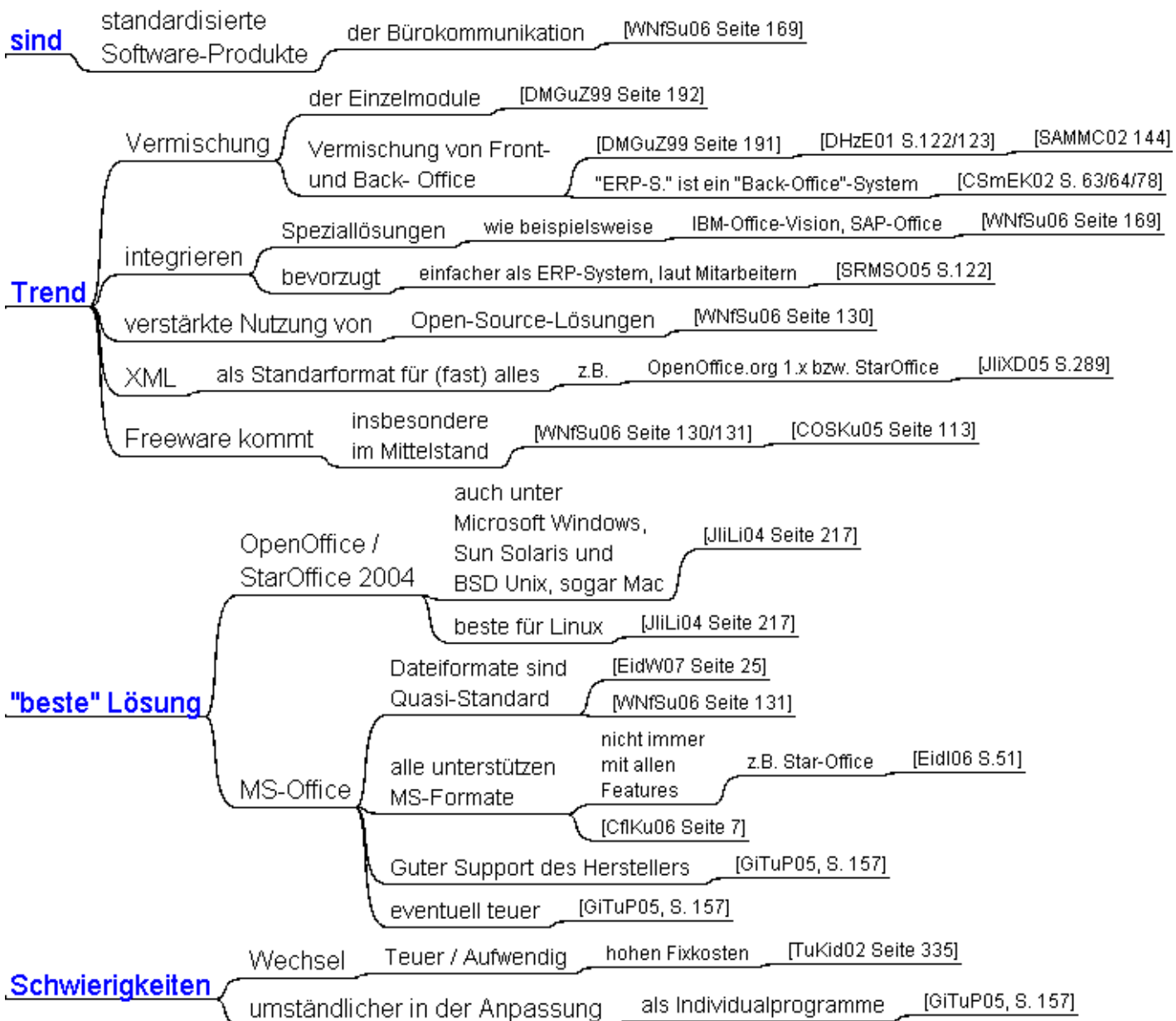


Abbildung 15: Trends bei Office-Paketen

Office-Pakete ersetzen Speziallösungen. Allerdings existieren auch mit Office-Funktionalitäten erweiterte Speziallösungen. Nach Lassmann/Schwarzer „ergänzen“ beispielsweise „IBM-Office-Vision, SAP-Office“ „ihre betriebswirtschaftlich orientierte Software um Funktionen zur Bürokommunikation“.

Jedoch sei ein Trend zur Integration von „Speziallösungen durch die Integration der Office-Produkte“ zu verzeichnen [WNfSu06, S.169].

Lassmann/Schwarzer weisen auch auf ein Wachstum von nicht-proprietären Lösungen (d.h. hauptsächlich Nicht-Microsoft-Lösungen) hin: „Für die Zukunft ist auch im Office-Bereich mit einer verstärkten Nutzung [von OpenOffice-Source-Lösungen, S.L.] zu rechnen.“ [WNfSu06, S.130]. OpenOffice-Lösungen werden proprietäre Lösungen wahrscheinlich jedoch nicht

komplett verdrängen können³³. Speziallösungen gelten grundsätzlich (per definitionem) als anpassbarer und werden daher wohl, auch auf längere Sicht, nicht vollständig verdrängt werden [GiTuP05, S.157].

Der Trend zur Vermischung und Integration hält an:

Das „ERP-System“ (z.B. SAP, JD Edwards, ...) wird immer noch zu den „Back-Office“- Systemen gezählt. Und enthielt ursprünglich „keine Front-Office-Funktionen“, sondern lediglich die klassischen Module wie „Produktion, Finanz- und Betriebsbuchhaltung, human Resources etc.“ [CSmEK02, S.63/64/78]. Back-Office- und Front-Office-Funktionen vermischen sich jedoch ständig mehr und eine klare Zuordnung wird schwieriger [WNfSu06, S.130]. Nach Kampffmeyer [DMGuZ99, S.191/192] nähern sich die Technologien an und spricht von „zunehmenden Vermischung“. „Back-Office-Lösungen“ würden um „Front-Office-Modulen“, wie „Workflow-, Dokumenten-Management- und Groupware-Funktionen“ ergänzt.

Klingelhöllers Prophezeiung aus dem Jahre 2001 besagt [DHZE01 S.122], dass der „Benutzer von morgen“ „im Normalfall nicht mehr merken“ wird ob im Hintergrund ein DMS läuft, das ihn mit den benötigten Informationen versorgt oder nicht. Dies scheint sich teilweise schon bewahrheitet zu haben (siehe Kapitel 3.3 „Bestehende Lösungen“).

OpenOffice-Source kommt:

Nach Lassmann/ Schwarzer [WNfSu06, S.130] stammt die begründete Behauptung, dass der „Einsatz von Freeware“ steige, „insbesondere in mittelständischen Unternehmen, da sie eine kostengünstige, qualitativ hochwertige Alternative zu proprietären Systemen“ darstelle.

Die attraktivsten „Anwendungsgebiete“ für OpenOffice-Source sind „vor allem Betriebssysteme (z. B. Linux), Webserver (z. B. Apache) und Programmiersprachen“, was sogar auch Anbieter von beispielsweise ERP-Systemen veranlassen würde, so „ihre Software auf OpenOffice-Source-Betriebssysteme zu migrieren“ [WNfSu06, S.131].

Ähnlich fällt die Prognose von Saleck [COSKu05, S.113] aus: „Auch auf dem Gebiet der Business-Software“ findet sich eine „Open-Source-Entwicklergemeinschaft zusammen“. Saleck nennt als Beispiel für eine OpenOffice-Source-ERP-Software „Lx-Office ERP“. Diese „Lx-Office ERP“ ist eine „browserbasierte Warenwirtschafts- und Finanzbuchhaltungssoftware, und OpenOffice-Source-Software.

XML ist weiter auf dem Vormarsch:

Die MS-Office-Dateiformate und MS-Office-Systeme mit einem Marktanteil von ca. 95% halten sich (noch?) hartnäckig. Jedoch ist ein Trend zu offenen, nicht-proprietären Software-Systemen und auch offene Dateiformaten zu erkennen. Seit längerem sehr bekannt und mittlerweile sehr verbreitet sind XML und ODF (ODF enthält XML-Dateien). Die „erste XML-Spezifikation wurde [...] 1998 vom W3C veröffentlicht“. „XML ist eine Teilmenge von SGML“ [XuSX08, S.1].

„IBM Lotus Symphony“, „OpenOffice.org 1.x“ bzw. „StarOffice“ (ab Version 6) setzen auf XML auf „und benutzen dieses Format zur Speicherung von Dokumenten.“ [JliXD05, S.289]. StarOffice ist die von der Firma Sun vertriebene kommerzielle Fassung des OpenOffice.org-Pakets. Beide Pakete benutzen in den Office-Anwendungen daher die gleichen Dokumentformate. Wenn hier von OpenOffice 1.x die Rede ist, lassen sich die Aussagen auch auf StarOffice 6/7 beziehen.

³³ Natürlich können Office-Pakete auch Bestandteil von anderen Lösungen (auch nicht-proprietärer Art) sein.

OpenOffice / StarOffice / Lotus Symphony

„IBM Lotus Symphony“ nutzt Teile des Quellcodes von OpenOffice und unterscheidet eher in der Bedienoberfläche, welche stark an Web-Browser erinnert. So werden die einzelnen Dokumente als Tabs angeordnet und am rechten Rand (SideShelf³⁴) öffnen sich Menüs und Auswahlfelder. Im Gegensatz zu OpenOffice öffnet Symphony alle Dokumente innerhalb eines Fensters und nicht in verschiedenen Fenstern. OpenOffice ist, wie IBM Lotus Symphony, kostenlos. Die kommerzielle Variante von OpenOffice ist „Sun StarOffice“ [JiLi04, S.217].

Das IBM-Büropaket Symphony besteht, im Gegensatz zu seiner Herkunft, dem Ur-Vater OpenOffice, nur aus drei Haupt-Features: Text-Dokument, Tabellen-Kalkulation und Präsentation. Das Symphony-Produkt bietet optional (im Gegensatz) zu OpenOffice einen Support „auf dem Niveau anderer IBM-Software-Produkte“.

Standardmäßig speichert Symphony Dokumente im „OpenDocument Format“ (ODF), ein Format, das auch Microsoft in Zukunft unterstützen wird. Wie Golem berichtet am 20.06.2008 [MODhg08], will Microsoft nun sogar am ODF-Format mitarbeiten und Ihr eigenes Microsoft OOXML-Format (Office Open XML) zurückstellen (ODF habe gewonnen). ODF-Dokumente (wie auch OpenOffice-Dokumente) bestehen zum größten Teil aus XML-Dateien, was man leicht nachprüfen kann, wenn man diese mit einem herkömmlichen Entpacker auspackt.

Betriebssysteme

StarOffice ist auch unter den Betriebssystemen „Microsoft Windows“, „Sun Solaris“ und „BSD Unix“ nutzbar. Eine Version für „Mac OS X“ ist geplant. Damit ist dieses Office-Paket ideal geeignet für Organisationen, welche mehrere Betriebssysteme einsetzen. Auch OpenOffice und „Lotus Symphony“ ist auf diesen Betriebssystemen einsetzbar.

Dateiformat

Das Dateiformat von OpenOffice und StarOffice ist in allen aktuellen Versionen absolut identisch. IBM Lotus Symphony unterstützt das ODF-Format ebenfalls. Das gilt unabhängig davon, unter welchem Betriebssystem die Programme laufen.

Microsoft Office

Nach Bächle/ Kolb [EidW07, S.25] sind die Quasi-Standards im Office-Bereich, immer noch die proprietären Dateiformate von „MS Office“. Daher ist es für alternative Office-Pakete von zentraler Bedeutung diese Formate importieren und exportieren zu können.

Guter Support ist wichtig – insbesondere für Geschäftskunden.

„Tendenziell ist das Risiko, irgendwann bei Problemen, bei Änderungs- oder Erweiterungswünschen alleine und ohne Unterstützung des Herstellers auskommen zu müssen, bei Standardsoftware geringer. Die Softwarepflege durch den Softwarehersteller ist bei Standardsoftware intensiver, was aber auch zu einer Folge schneller (kostenpflichtiger) Updates führen kann“ [GiTuP05, S.157].

Das Support insbesondere für Geschäftskunden wichtig ist hat IBM längst erkannt und bietet für das Office-Paket „IBM Lotus Symphony“ einen kostenfreien moderierten Online-Support an. Darüber hinaus ist ein kostenpflichtiger Service, insbesondere für die Bedürfnisse großer

³⁴ Mit SideShelf ist seitlicher Fensterbereich innerhalb der Office-Anwendung gemeint.

Organisationen nutzbar, welcher uneingeschränkten technischen Support auf dem Niveau anderer IBM Softwareprodukte bietet.

Schwierigkeiten beim Wechsel

Nach Müller/ Eymann/ Kreuzer [TuKid02, S.335] möchte man bestimmte Informationsgüter, vor allem Informationsproduktionsgüter wie z.B. Datenbanksoftware oder Office-Software, nicht gerne wechseln, da solch ein Wechsel einerseits Wechselkosten und andererseits einem aufwendigen Lernprozess (Grund: Verlassen der gewohnten Umgebung) für den Käufer zur Folge hätte.

3.3 Bestehende Lösungen

Die zwei größten Anbieter, welche eine Integrationslösung zwischen einem Office-Paket und SAP bieten, d.h. einem Zugriff direkt aus einem Office-Paket heraus auf SAP-Daten und -Prozesse, sind (jeweils in Zusammenarbeit mit SAP) Microsoft und IBM. Beide Lösungen werden im SAP-Geschäftsbericht 2007 [SGINM08] auf S.34 nebeneinander aufgeführt.

Eine Liste von ERP Software-Systeme findet sich im Anhang auf S. 85.

3.3.1 Das Produkt „Duet“ von Microsoft Office und SAP

„Duet Software for Microsoft Office and SAP

Dank Duet haben Anwender einen direkten und einfachen Zugang zu SAP-Daten und -Prozessen über Microsoft Office. Damit revolutioniert Duet, das im April 2005 erstmals vorgestellt wurde, den Zugriff auf betriebswirtschaftliche Software vom Desktop aus. Das Produkt ist aus einer richtungweisenden Zusammenarbeit zwischen SAP und Microsoft entstanden und ist das erste gemeinsame Produkt der beiden Branchenführer. Im April 2007 kündigten SAP und Microsoft an, dass sie ihre Partnerschaft vertiefen und gemeinsam weitere Geschäftsszenarios, Plattformfunktionen und Entwicklungswerkzeuge für Duet auf den Markt bringen werden.“ [SGINM08, S.34].

3.3.2 SAP-Integration in Lotus Notes - Projekt „Atlantic“

SAP und IBM bieten demnächst eine gemeinsam entwickelte Integrationslösung mit dem Codename „Atlantic“ an [SGINM08, S.34]. Auf der diesjährigen größten Kundenmesse von IBM für Lotus-Kunden wurde der Beschluss von Lotus und SAP angekündigt, gemeinsam ein Produkt zu entwickeln, welches die IBM-Lotus-Welt mit SAP-Business-Systemen verbindet. Möglich wird diese Integrationslösung durch die neue Version 8 des Lotus Notes Clients. Mit Notes 8 setzt IBM weiterhin konsequent auf offene Standards³⁵ [AgZ08].

Atlantic bietet eine komplette Integration von SAP-Business-Prozessen in den Lotus Notes Client und ermöglicht es, nicht nur direkt aus Lotus Notes heraus, Bestellungen freizugeben, sondern auch SAP Workflows zu starten.

Lotus Notes kann unterschiedlichste Anwendungen innerhalb eines Clients betreiben und die einzelnen Anwendungen über ein „Composite Application Framework“ im Lotes Client miteinander vernetzen. Damit kann in relativ kurzer Zeit eine Applikation entwickelt werden und es ist sogar Kunden selbst möglich ihre Atlantic- Installation anzupassen oder zu erweitern. Damit wird

³⁵ In diesem Zusammenhang ist auch das Kapitel „Der Trend zum Open Source“ auf Seite 34 interessant.

Mitarbeitern eine individuelle, hochproduktive Arbeitsumgebung verfügbar sein, welche optimal in Geschäftsprozesse im Backend integriert ist.

Dabei werden gegenüber den Lösungen des größten Konkurrenten (Microsoft) deutlich Kosten gespart [🔍LNaDP08, S.46].

3.3.3 CRM/ERP-Kopplung via Web-Services von CAS Software AG

Für das On-Demand-CRM-Produkt „Pia“, so Frank Niemann, wird es demnächst eine ERP-Schnittstelle geben. Dies ermögliche den Nutzern des Kunden-Management-Programms beispielsweise Aufträge von einer CRM-Software und von einer ERP-Software aus zu übergeben [🔍CSAaa].

Die Pia-Software von CAS Software AG richtet sich an kleinere Unternehmen. Diese Kunden-Management-Software lässt sich über den Web-Browser bedienen. Die Lösung umfasst hauptsächlich ein Kontakt-Management, einen Gruppenkalender und ein Kampagnen-Management und eine Computer-Telefonie-Integration (CTI). Für Nutzer, die eine bestehende ERP-Software anbinden wollen, soll es demnächst eine Erweiterung geben [🔍CSAaa].

Die Kosten pro Monat für einen Nutzer betragen 19,90 Euro [🔍CSAaa].

3.4 Der Trend zum Open-Source

Laut einer Studie der Marktforscher von IDC vom 31. Mai 2007 wird sich der Markt für Open-Source-Software in fünf Jahren verdreifachen [🔍WRfSO08]. Matt Lawton, Direktor des IDC-Forschungsprogramms Open-Source, sieht in diesem Trend bestätigt, dass quelloffene Software den „umfassendsten und langfristigen Trend“ in der Softwareindustrie seit den 1980er Jahren darstellt.

Diese Entwicklung erklärt der „Bundesverband der Wirtschaftsförderung und Außenwirtschaft“ in seiner Broschüre vom 2. November 2008 [🔍DDIRV08, S.90]. Es zeigt sich „im Bereich der Open-Source software-Entwicklung [...] oder der OpenOffice content-Bewegung, dass der ökonomische Markt keineswegs der einzige Mechanismus zur Schaffung der für Innovationen wichtigen Anreizwirkungen ist.“ Auch andere Motive könnten bestimmend werden, wie etwa das Interesse am Zugewinn von „Aufmerksamkeit“ oder „Reputation“ oder auch nur „Neugier und Vergnügen am Erfolg einer innovativen Anstrengung“.

Gewinnt das Format „OpenDocument“ oder das „Office Open XML“ von Microsoft?

Stuart McKee³⁶ (Microsoft) gestand Anfang Juni 2008 auf dem „Red Hat Summit“ die Niederlage des proprietären Microsoft-Formates OOXML gegenüber ODF ein. McKee sagte „ODF hat ganz klar gesiegt“, so berichtete die US-amerikanische Newssite InfoWorld. Er bezog sich damit auf die Aussage von Microsoft, künftig ODF direkt in Office unterstützen zu wollen. OOXML hat zu Beginn des Jahres die Zertifizierung der ISO als Standard erhalten, worauf sogleich Brasilien, Indien, Südafrika und Venezuela Einspruch erhoben.

³⁶ Stuart McKee ist nationaler Technikchef bei Microsoft USA

Allerdings wurde dann doch am 15. August 2008 die Publikation des Standards ISO/IEC DIS 29500 (Bezeichnung des OOXML-Standards) offiziell genehmigt. Das „OpenDocument Format“ (ODF) ist bereits seit zwei Jahren ISO-Standard (ISO 26300) und wird von der Europäischen Kommission empfohlen.

Zahlreiche Länderregierungen³⁷ und öffentliche Institutionen haben ihre Software-Beschaffungen an die Bedingung geknüpft, dass ein offener Standard für Dokumente eingehalten wird. Bislang kamen dafür ausschließlich ODF und dazu verträgliche Anwendungen wie „StarOffice“/„OpenOffice“, „KDE Office“, „Softmaker Office“ oder „Abiword“ in Frage.

3.5 Ergebnisse der Ist-Analyse

Microsoft Lösungen sind immer noch sehr stark verbreitet, aber Lösungen mit offenen Formate werden bekannter und beliebter - auch für Geschäftskunden. Es gibt sehr viel Office-Pakete und einige ERP-Systeme. Das Dokumenten-Format (ODF) stellt eine Möglichkeit dar einen breiten Kundenkreis zu bedienen. Hält der Trend zum OpenOfficeSource an, so wird über kurz oder lang die SAP-Integration Lotus Notes, das Projekt "Atlantic" und diese mit dieser Diplomarbeit vorbereite Lösung, sowie ähnliche Lösungen, die Nase vorn haben.

³⁷ Zahlreiche Verwaltungsorgane unterschiedlicher Bundesstaaten vieler Staaten haben sich an ODF gebunden z.B. Argentinien, Belgien Brasilien, Deutschland, England, Finnland, Frankreich, Kroatien, Japan, Malaysia, Niederlande, Norwegen, Österreich, Schweden, Südafrika, Uruguay und andere.

4 Machbarkeits-Analyse

In diesem Kapitel ist mit Machbarkeit die technische Machbarkeit gemeint, d.h. es geht darum ob Ziele prinzipiell überhaupt erreichbar sind und ob die Ziele mit Hilfe des aktuellen Kenntnisstandes der Wissenschaften realisierbar sind.

Die Machbarkeitsanalyse kann auch als Teil der Ist-Analyse verstanden werden (was bereits existiert, ist auch machbar).

4.1 Fehlertoleranz bzw. Interpretationsfreudigkeit

„Fehlertolerante Systeme arbeiten auch dann noch korrekt, wenn während ihrer Ausführung eine bestimmte Anzahl von Fehlern auftreten.“ [FGdFiS01, S.3].

Eine der bekanntesten Definitionen von Fehlertoleranz stammt von Arora und Gouda [Cacfo93]. Nach Ihnen ist ein System fehlertolerant, wenn es die Fähigkeit besitzt Fehler (Abweichungen bezüglich einer Spezifikation) zu tolerieren. Außerdem müsse spezifiziert sein, welches Fehlverhalten noch akzeptabel sei und welches nicht [ebd.].

Auf bestimmte Fehlersituationen kann man toleranter reagieren, aber bezüglich der Daten werden in der Regel keine Fehler toleriert (Ausnahmen stellen z.B. die meisten Webbrowser dar).

Auch hängt die Bereitschaft zu einer bestimmten Fehlertoleranz von den Erwartungen des Nutzers ab. Geschäftsleute, Sachbearbeiter legen mehr Wert auf Daten-Genauigkeit als auf Genauigkeit bezüglich des Layouts oder der Farben.

4.2 Fehlertoleranz bezüglich der Daten

In der Regel ist Fehlertoleranz bezüglich der Daten-Genauigkeit und -Vollständigkeit nicht zulässig. Insbesondere bei Geschäftsdaten sind Ausnahmen kaum zu verantworten. Die Daten müssen entweder alle oder gar nicht angezeigt werden, weil anhand ihrer Darstellung eventuell wichtige Entscheidungen getroffen werden können. Andererseits können aufgrund von Daten, welche überhaupt nicht sichtbar sind, (weil z.B. deren Darstellung nicht für genau oder vollständig genug gehalten wird), auch keine guten Entscheidung getroffen werden. D.h. manchmal kann es zur Entscheidungsfindung besser sein, ein bisschen zu wissen als gar nichts.

Jedoch müssen auch juristische Folgen bedacht werden. Aus einer juristischen Perspektive gesehen, stellen die eventuellen Folgen einer Fehlentscheidung ein weitaus höheres Risiko dar, als die Folgen einer zu spät oder gar nicht getroffenen Entscheidung. Außerdem gerät ein Softwarehersteller immer in die Gefahr juristisch belangt zu werden, wenn sein Produkt Daten ungenau darstellt. Also muss – sollte man eine gewisse Toleranz bzgl. der Datengenauigkeit erlauben – ein deutliche Warnung auf eventuelle Fehler in der Darstellung unbedingt erfolgen!

Grundsätzlich besteht allgemeines Interesse an der Datenquellen-Vielfalt darstellbarer Information [VvDwa07, S.14]. Datenformate können auch fehlertolerant gestaltet sein. HTML ist beispielsweise aufwärtskompatibel aufgebaut, neue HTML Einheiten werden von Browsern, die sie nicht verstehen, ignoriert, ohne dass das Dokument sofort unlesbar wird. Je mehr man versucht dem Wunsch der Datenquellen-Vielfalt zu entsprechen, desto eher kommt man an eine Grenze, ab der die Daten dann nur noch fehlertolerant dargestellt werden können. In der Programmentwicklung ist die Gewährleistung von Aufwärtskompatibilität schwieriger als die von Abwärtskompatibilität, weil beim Erstellen einer aktuellen Version einer Anwendung noch nicht alle Formate und Strukturen

späterer Programm-Versionen dieser Anwendung bekannt sein können. Auch einige der, in dieser Arbeit gesetzten, Ziele lassen entspringen einem Wunsch nach Datenquellen-Vielfalt:

- (Ziel A) eine allgemeine Lösung ,
- (Ziel B.3.c) möglichst **viele unterschiedliche Datenquellen**
- (Wunsch ii) Datenquellen-Vielfalt

4.2.1 Nützlichkeit von unscharfer Information

Nach der in der Betriebswirtschaftslehre klassischen Definition von Wittmann (1959), wird Information als zweckorientiertes Wissen betrachtet. D.h. nach Green [WUWI04, S.136]: „Information wird nach Attributen, wie Zuverlässigkeit, Nützlichkeit, Vollständigkeit, Aktualität oder Kosten beurteilt.“

Je fehlertoleranter, beliebiger, kurz – salopp gesagt – interpretationsfreudiger ein System jedoch wird, desto eher mangelt es der einzelnen Informationsdarstellung an Zuverlässigkeit, Nützlichkeit und Vollständigkeit. Jedoch entschädigt die größere Interpretationsbreite (d. h. auch potentiell größere Vielfalt an zugänglicher Information) möglicherweise ein wenig für die Schwächen in Punkto Zuverlässigkeit, Nützlichkeit und Vollständigkeit (dies hängt ganz wesentlich von der Art des Anwendungsfalles ab).

Gut bekannt als fehlertolerante Interpreten sind die Web-Browser. So können beispielsweise ältere Webbrowser auch ein XHTML 1.0 Dokument darstellen, da ihre Fehlertoleranz die kleinen Unterschiede zwischen XHTML und HTML ausbügelt (irgendwie interpretiert oder großzügig übersieht) und XHTML einfach als fehlerbehaftetes HTML interpretiert. Dies geht im Falle des XHTML-Formats aber nur deshalb vergleichsweise einfach, weil (X)HTML bewusst möglichst abwärtskompatibel zu HTML konstruiert wurde³⁸.


In jedem Fall ist eine Darstellung von abwärtskompatibler Information in einem ältern Format nie Verlustfrei (es sei denn Features des neueren Formates wurden nicht genutzt).

4.2.2 Die Wissenstreppe – Alles bedarf einer Syntax



Abbildung 16: Wissenstreppe [WUWdW02, S.41]

³⁸ Übrigens: XHTML 2 soll nicht mehr abwärtskompatibel zu HTML sein [EGzQd08].

Auf der untersten Ebene dieser Abbildung ist das „Zeichen“ gesetzt. Streng genommen bedarf jedoch selbst solch ein Zeichen erst einer Syntax³⁹ (z.B. Datei-Format), bevor es zum Zeichen werden kann. Anschließend können Zeichen (noch erste Ebene), wiederum durch eine Syntax zu Daten verknüpft werden [vgl. auch luWli04, S.137].

4.2.3 Fehlertoleranz bis zur völligen Nutzlosigkeit?

Nur aufgrund von Annahmen, Vermutungen, Wissen beziehungsweise irgendwelcher Information über die Information (also Metainformation/Strukturinformation wie z.B. Syntax, Datenformat), kann überhaupt sinnvoll interpretiert werden. Die totale Fehlertoleranz (keine Strukturinformation verfügbar, vermutbar u. Ä.) führt also in die totale Beliebigkeit und dadurch auch in die völlige Nutzlosigkeit der Interpretation⁴⁰.

Demnach kann der Forderung, (sinnvoll) alle (Business-) Daten darzustellen, nicht entsprochen werden, es sei denn man könnte davon ausgehen, dass zu allen (Business-) Daten Struktur-/Meta-Information zugänglich oder ableitbar, konstruierbar bzw. vermutbar wären⁴¹.

Wie kann der Wunsch aber dennoch so weit wie möglich entsprochen werden? Wie kann man sich dennoch einer solch allgemeinen Forderung (Business-Daten darzustellen) nähern? Eine Antwort darauf wird im Folgenden versucht.

³⁹ Man könnte die Voraussetzungen bis auf die nötige Hardware des Computers, und noch darüber hinaus, zurückführen.

⁴⁰ Selbst die Kryptographie braucht Vorwissen, Annahmen (Strukturinformation), wie z.B. die Kenntnis von der Sprache, woraus z.B., im Falle deutscher Schrift, gefolgert werden kann, dass der häufigste Buchstabe wahrscheinlich das e ist (erste erfolgreiche Anwendung der Häufigkeitsanalyse von arabischen Gelehrten im 9. Jahrhundert).

⁴¹ Ein Beweis wäre schwierig: Für endliche Mengen ist unter bestimmten Umständen eine Untersuchung jedes einzelnen Elements möglich. Lange Zeit wurde auch die Aussage „Alle Schwäne sind weiß“ durch Beobachtung ausnahmslos bestätigt und wurde daher als empirisch bewiesen angesehen. Infolge der Entdeckung Australiens wurde diese Aussage jedoch widerlegt (Trauerschwan).

4.3 Business-Daten nach Office (B2O)

Bei Business-Daten handelt es sich meist um Zahlen, z.B. Statistiken, Bilanzen, Angebote. Lange Texte kommen eher selten vor. Und Binärdaten wie Filme, Applikationen oder auch Bilder kommen kaum als Business- Wert vor. Insofern scheinen Business-Daten ideal für den Import in die Tabellenkalkulation geeignet zu sein (Kalkulation hat mit Zahlen zu tun). Jedoch können Business-Daten viel mehr als Zahlen und Text sein.

Business-Daten können prinzipiell alle Arten von Daten sein [siehe ☆PDSSG08, S.2 und Kapitel 2 „Theoretische Grundlagen“]. Ebenso ist die Anzahl der möglichen Office-Pakete und damit auch der möglichen Zielformate unüberschaubar, jedenfalls nicht klar abgrenzbar. Man kann jedoch versuchen, die wichtigsten Quellen und Zielformate zu nennen:

- Auswahl von ERP-Quellen: Marktanteile ERP-Software
- Auswahl von Office-Paketen: Liste von Office-Paketen

Danach kann man versuchen, eine möglichst umfassende Menge von Einzellösungen in einer einzelnen Lösung zu vereinen [Stichwort Mashup⁴², □Mp08, S.79]. Im Rahmen dieser Diplomarbeit ist dieser Ansatz nicht umsetzbar (ist auch nicht gefordert), aber prinzipiell ist der Ansatz durchführbar. Formate konsolidieren und die rasante Verbreitung von Web-Services erleichtert das Umsetzen einer solchen Verbindung von Einzellösung. Teich spricht sogar (□DrWzS08, S.226) von einem kommenden Paradigmenwechsel und sagt wörtlich: „Individualität ist wieder gefragt“. Dies ermöglicht der Trend zum Web-Service (meist kleinen Funktionseinheiten), welcher oftmals in eine individuelle Einzellösung (oftmals als Mashup) eingeht. Insofern ist eine so allgemeine B2O-Lösung, wie in (Ziel A) gefordert, immerhin näherungsweise zu erreichen. Mehr als eine näherungsweise Umsetzung kann man jedoch nicht erwarten, da dafür die Ausgangsbedingungen nicht klar genug definiert werden konnten (siehe dazu z.B. die Definition des Begriffs „Business Daten“ im Kapitel 2.5).

Untere Abbildung stellt den von Irene Teich (und Herren Kolbensschlag und Reiners) genannten Wechsel zu Individuallösungen (aufgrund der Nutzung von Services) dar (□DrWzS08, S.226):

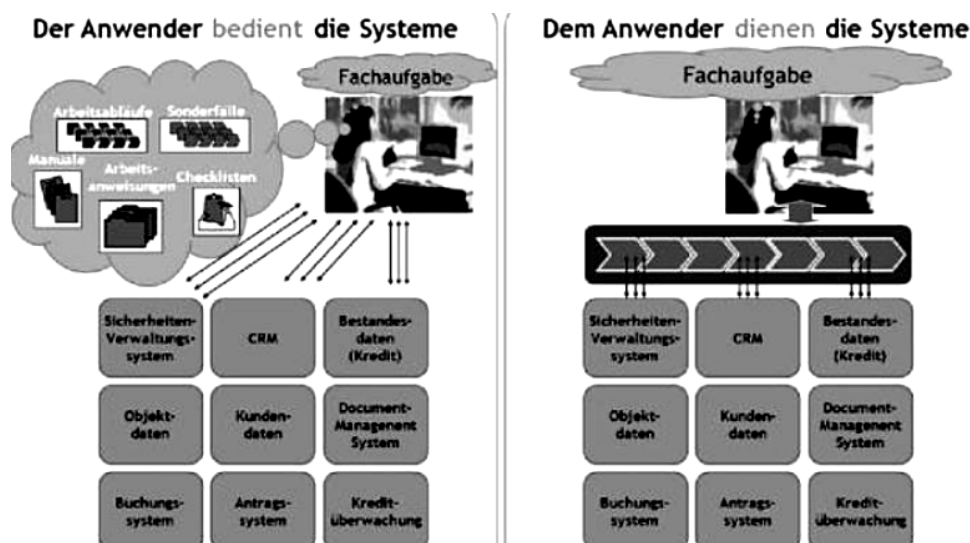


Abbildung 17: Paradigmenwechsel in der Ausprägung von IT-Systemen (□DrWzS08, S.226)

⁴² Mashup (engl. „Verknüpfung“) steht für die Erstellung neuer Inhalte durch die nahtlose (Re-)Kombination bereits bestehender Inhalte.

4.4 Business-Daten nach Symphony bzw. ODF (B2S)

Durch die Einschränkung des Datenziels und damit auch der Ausgabeformate auf ein Office-Paket (hier Symphony) ist die Zieldefinition klar genug definiert für eine praktische Umsetzung. Außerdem sind alle ODF- Ausgabeformate offen (hauptsächlich bestehend aus XML- Dateien) und zu Symphony existiert eine API. Die Definition der Quellen ist jedoch nach wie vor unzureichend und eine solch allgemeine Forderung immer noch nur näherungsweise möglich. Zwar ist damit solch eine praktische Umsetzung im Rahmen dieser Diplomarbeit nicht möglich (und auch nicht gefordert), jedoch ansonsten durchaus denkbar.

4.5 Daten aus dem SAP-System

Durch die Einschränkungen der Datenquelle auf Daten aus einem SAP System erhält man nun eine klar definierte Ausgangsmenge mit Ausgangsformaten und einer klaren Definition des Ziels, (nämlich dem Office-Paket IBM Lotus Symphony) mit seinen entsprechenden Formaten etc.

Zu jedem Modul aus dem SAP-System existiert eine Reihe BAPIs. BAPIs sind Businessobjekten mit den zugehörigen Operationen und Schnittstellen (siehe Kapitel 2.7.2 „Schnittstellen zu SAP-ERP-Systemen“).

5 Konzeption und Implementierung

Im Folgenden geht es (top-down) zuerst konkret von einem Anwendungsfall (der Einladung zur Jubiläumsfeier) mit klar definierter Quelle (Adress-Daten von SAP) und Ziel (ODS in Symphony) weiter zu allgemeineren Fällen (z.B. zu Fällen mit Einbeziehung noch anderer Quellen).

5.1 Szenario „Einladung zur Jubiläumsfeier“

Ein **Sachbearbeiter** aus der Personalabteilung benötigt die **Adressliste** aller Kunden für die Einladung zur Jubiläumsfeier. Er möchte die Adressliste in Form einer Tabellenkalkulation per **E-Mail-Anhang** an eine **Werbeagentur**, welche letztlich für den Versand zuständig ist, versenden. Diesen Sachverhalt stellt folgendes Schaubild dar:

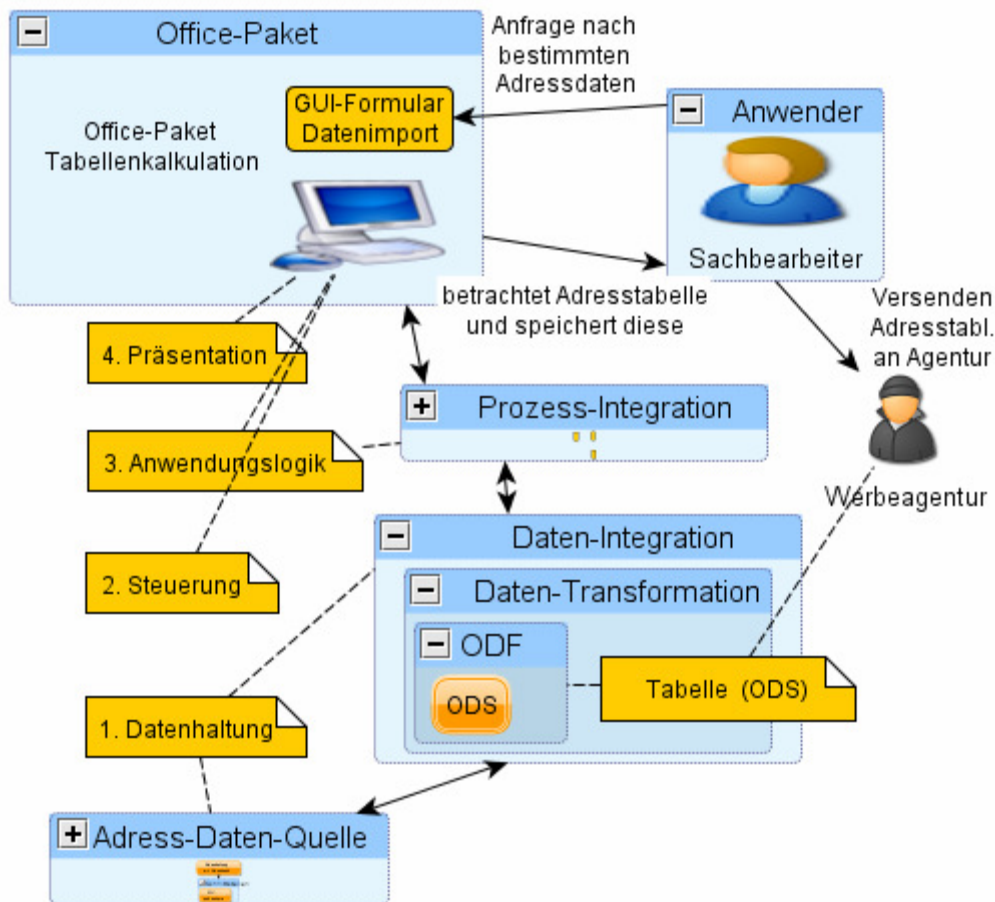


Abbildung 18: Szenario Einladung zur Jubiläumsfeier

Das Ergebnis seiner Abfrage in Symphony, welche er dann anschließend speichert und per E-Mail versendet, könnte folgendermaßen aussehen:

	A	B	C	D	E	F
2	0000001001	LAMPEN	Lampen-Markt GmbH	DE	DE	Frankfu
3	0000001002	OMEGA	Omega Soft-Hardware	DE	DE	Nuernbu
4	0000001003		IDES AG Tochtergesel	DE	DE	Mannhe
5	0000001004		US-Partner	US	US	JEWET
6	0000001005	RFID	CPG Europa	DE	DE	Offenba
7	0000001007	RFID	Pharma AG	DE	DE	Frankfu
8	0000001010	BECKER	Becker Berlin (Versand)	DE	DE	Berlin
9	0000001012	AM	Autohaus Franzl GmbH	DE	DE	Muenct
10	0000001020	BECKER	Becker Berlin (Lagerun)	DE	DE	Berlin

Abbildung 19: Kunden-Adressliste in Symphony-Tabelle

In obigem Fall (Screenshot) wurden Daten aus einem SAP-ERP-System gelesen. Dazu wurde das BAPI_CUSTOMER_GETLIST verwendet. Mit diesem können Daten aus dem „Business Object“ 'Customer' des SAP-Moduls Finanzen ausgelesen werden.

5.2 Implementierung SAP-Daten in Symphony-Tabelle

Grundsätzlich kann man auf zwei unterschiedlichen Wegen ein Dokument erzeugen und in einer Office-Applikationen darstellen:

- Entweder man erzeugt das Dokument erst komplett und öffnet es dann anschließend in der Office-Applikation (B2ODF→O).
- Oder man öffnet ein leeres Dokument (zum Beispiel unsichtbar) innerhalb der Office-Applikation und erzeugt dessen Inhalt mit Instrumenten der Office-Applikation bzw. der Office API (O→B2ODF).

Der zweite Weg nutzt von Anfang an die Instrumente der Officeapplikation (API) und ermöglicht eventuell (falls gewünscht) bereits während der Generierung einen Blick auf den momentanen Generierungsstand.

Der erste Weg ermöglicht eine klarere Trennung von Generierung und Darstellung und damit eine größere Unabhängigkeit (geringere Verstrickung) von Bibliotheken. Eine klare Trennung ist allgemein auch aus der Perspektive der Verwertbarkeit (objektorientiert) oder auch hinsichtlich der Modularisierung von Vorteil.

5.2.1 SAP nach ODF ohne Symphony-API (B2ODF)

Standardmäßig speichert Symphony Dokumente im „OpenDocument Format“ (ODF), ein Format, das auch Microsoft in Zukunft unterstützen wird. Wie Golem berichtet am 20.06.2008 [MODhg08], will Microsoft nun sogar am ODF-Format mitarbeiten und ihr eigenes Microsoft OOXML-Format (Office Open XML) zurückstellen (ODF habe gewonnen).

Die ODF-Formate sind nicht proprietär und im Wesentlichen gezippte XML-Dateien. Dadurch lassen sie sich leicht mit unterschiedlichen XML-Tools und herkömmlichen Datei-Packern bearbeiten. Somit lassen sich unabhängige Konverter erstellen (unabhängig von einem Office-Paket bzw. von der Präsentationsschicht). Im Folgenden wird mittels einer ODF-Datei (Textdokument) dieser Sachverhalt dargestellt.

Wichtiger Hinweis: Trotzdem ist für die meisten Anwendungsfälle (auch im Fall „Einladung zur Jubiläumsfeier“) ein GUI⁴³, z.B. für die Eingabe/Auswahl der zu importierenden/integrierenden Daten, erforderlich. Dieser Aspekt wird jedoch erst später näher betrachtet, nachdem die Symphony-API beschrieben wurde.

5.2.1.1 ODT ausgepackt

ODF-Dokumente (wie auch OpenOffice-Dokumente) bestehen zum größten Teil aus XML-Dateien, was man leicht nachprüfen kann, wenn man diese einmal mit einem herkömmlichen Entpacker auspackt⁴⁴ [JliXD05, S.290].

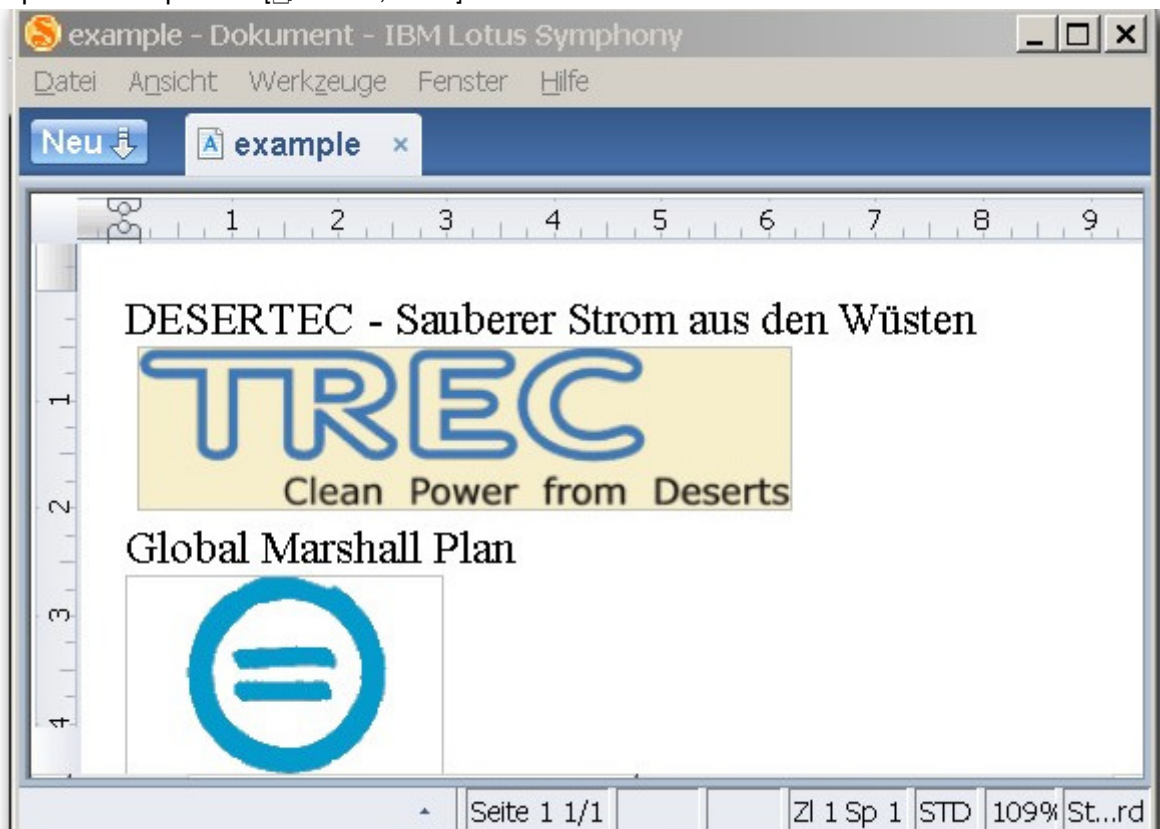


Abbildung 20: ODT Screenshot (Die beiden Bilder sind aus <http://www.clubofrome.de/aktiv/>)

⁴³ Eine grafische Benutzeroberfläche bzw. Graphical User Interface (GUI) ist eine Softwarekomponente, die einem Computerbenutzer die Interaktion mit der Maschine über grafische, metaphorhafte Elemente erlaubt.

⁴⁴ Auch das OpenOffice-Format (swx) kann auf die gleiche Art entpackt werden [JliXD05, S.290].

Im Dateisystem stellt sich das ODT-Dokument als einzelne Datei dar. Im Folgenden eine mögliche Darstellung⁴⁵:

Name	Größe	Typ
..		Dateiordner
example.odt	16 kB	OpenDocument - Text

Abbildung 21: ODT-Dokument im Dateisystem

Ändert man nun die Dateierdung von „odt“ einfach in „zip“ und entpackt diese, erhält man einen neuen Ordner:

Name	Größe	Typ
..		Dateiordner
example		Dateiordner
example.zip	16 kB	ZIP-komprimierter Ordner

Abbildung 22: ODT-Dokument mit Endung .zip

In der ausgepackten ODT-Datei finden sich dann einige XML-Dateien und (mindestens) ein Miniaturbild im PNG-Format:

Name +	Größe	Typ	Dateityp	Dateien ▾	Beschreibung																											
..		Dateiordner																														
META-INF		Dateiordner																														
Pictures		Dateiordner																														
Thumbnails		Dateiordner																														
content.xml	5 kB	XML-Datei																														
meta.xml	1 kB	XML-Datei																														
mimetype	1 kB	Datei																														
settings.xml	6 kB	XML-Datei																														
styles.xml	7 kB	XML-Datei																														
			<table border="1"> <thead> <tr> <th>Dateityp</th> <th>Dateien ▾</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>Datendateien</td> <td>5</td> <td>Dateien die verschied..</td> </tr> <tr> <td>.xml</td> <td>5</td> <td>XML-Datei</td> </tr> <tr> <td>Sonstige Dateien</td> <td>1</td> <td>Unbekannte Dateitype</td> </tr> <tr> <td>(Keine)</td> <td>1</td> <td>Keine Dateierdung</td> </tr> <tr> <td>Grafik-Dateien</td> <td>1</td> <td>Dateien, die Bilder, Gr..</td> </tr> <tr> <td>.png</td> <td>2</td> <td>IrfanView PNG File</td> </tr> <tr> <td>Container-Dateien</td> <td>1</td> <td>Komprimierte Archive .</td> </tr> <tr> <td>.zip</td> <td>1</td> <td>ZIP-komprimierter Or..</td> </tr> </tbody> </table>	Dateityp	Dateien ▾	Beschreibung	Datendateien	5	Dateien die verschied..	.xml	5	XML-Datei	Sonstige Dateien	1	Unbekannte Dateitype	(Keine)	1	Keine Dateierdung	Grafik-Dateien	1	Dateien, die Bilder, Gr..	.png	2	IrfanView PNG File	Container-Dateien	1	Komprimierte Archive .	.zip	1	ZIP-komprimierter Or..		
Dateityp	Dateien ▾	Beschreibung																														
Datendateien	5	Dateien die verschied..																														
.xml	5	XML-Datei																														
Sonstige Dateien	1	Unbekannte Dateitype																														
(Keine)	1	Keine Dateierdung																														
Grafik-Dateien	1	Dateien, die Bilder, Gr..																														
.png	2	IrfanView PNG File																														
Container-Dateien	1	Komprimierte Archive .																														
.zip	1	ZIP-komprimierter Or..																														

Abbildung 23: ODT ausgepackt

Nach dem Auspacken erhält man folgende XML-Dateien:

- content.xml
- styles.xml
- meta.xml
- settings.xml
- META-INF/manifest.xml

⁴⁵ die Darstellung variiert unwesentlich je nach verwendetem Datei-Browser-Tool

Der Anfang der zugehörigen `content.xml` sieht (auszugsweise) so aus:

```
<?xml version="1.0" encoding="UTF-8" ?>
<office:document-content xmlns:office="urn:oasis:names:tc:opendocument:xmlns:office:1.0"
.....
- <text:p text:style-name="Default_20_Text">
  DESERTEC - Sauberer Strom aus den Wüsten
  <text:line-break />
- <draw:frame draw:style-name="fr2" draw:name="Grafik2" text:anchor-type="as-char"
  svg:width="6.096cm" svg:height="1.524cm" draw:z-index="0">
  <draw:image xlink:href="Pictures/100000000000012C0000004B14DADCB1.gif"
  xlink:type="simple" xlink:show="embed" xlink:actuate="onLoad" />
</draw:frame>
<text:line-break />
Global Marshall Plan
```

Abbildung 24: Quelltext-Auszug content.xml

Für die Bearbeitung von XML-Dateien gibt es eine Vielzahl von bewährten, meist kostenlosen, Tools und Bibliotheken, für fast jede Programmiersprache.

5.2.1.2 ODF- Dokument- Generierung anhand Vorlage (und Daten)

Das folgende Diagramm beschreibt den Ablauf der Erstellung eines ODF- Dokuments (ODS, ODT, ODP) anhand einer Vorlage und Daten. Zu unterem Diagramm findet sich im Anschluss ein Kommentar.

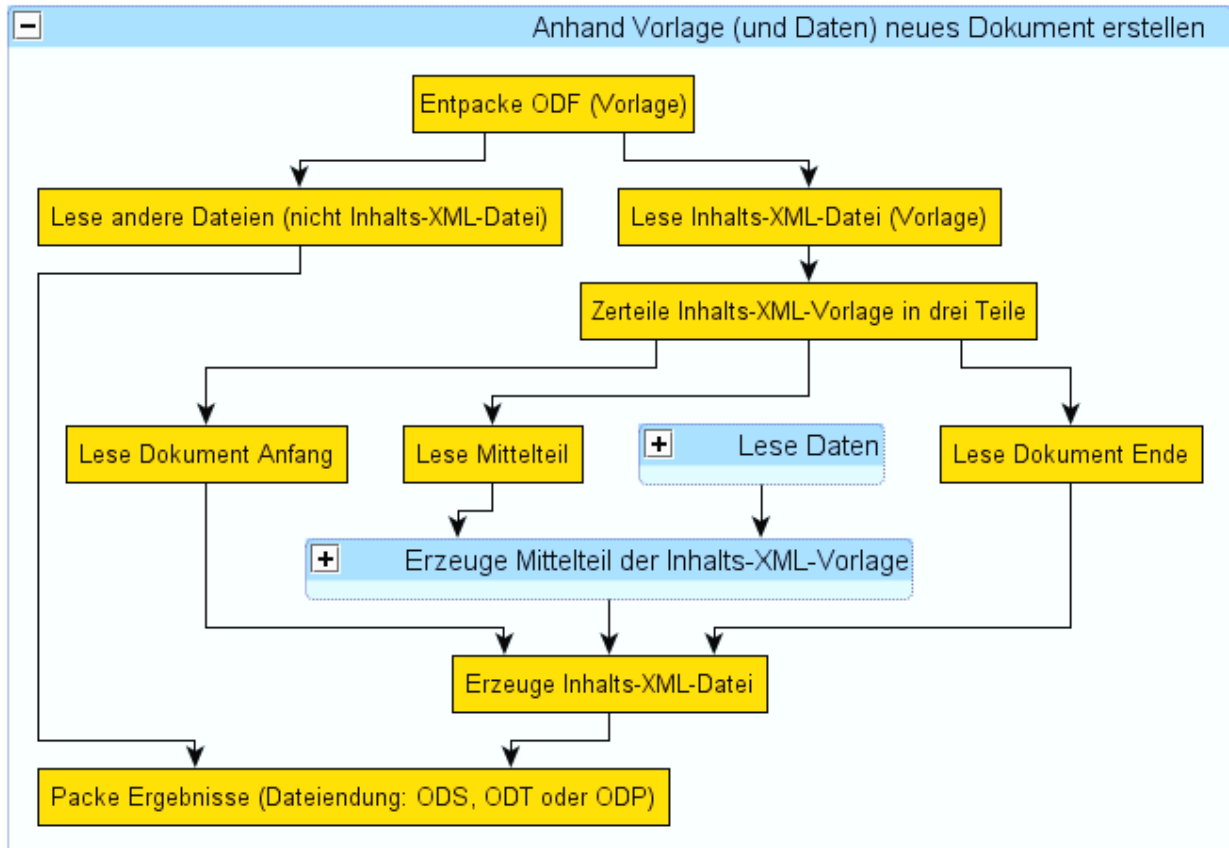


Abbildung 25: Anhand Vorlage (und Daten) neues Dokument erstellen

Das Diagramm beschreibt den Ablauf der Erstellung eines ODF- Dokuments anhand einer Vorlage und Daten. Dabei werden, der Übersichtlichkeit wegen, Fehlerbehandlungen, Plausibilitätsprüfung und Ähnliches in dieser Ablaufdarstellung weggelassen.

ODF-Dokumente zeigen sich nach dem Entpacken ihr Inneres, d.h. eine Reihe von XML-Dokumenten, Ordnern und binären Dateien. ODF ist also nicht proprietär und einfach zugänglich. Die dem Diagramm zu Grunde liegende Idee, nämlich eine ODF Vorlage zu verwenden und in dieser nur einen kleinen Teil (z.B. einen welcher die Daten repräsentiert) auszutauschen, liegt daher nahe.

5.2.2 SAP nach ODF mit Symphony-API (O2B2ODF)

Ein anderer Weg zu einem Office-Dokument ist der indirekte Weg über eine Office-API. Viele Office-Pakete bieten solch eine Schnittstelle an (neben Symphony beispielsweise auch Microsoft mit seinem VBA). Bezüglich der Darstellung eines Office-Dokumentes in der zugehörigen Applikation ist der Weg über die Office-API allerdings der direktere. Außerdem ist im Falle von proprietären Formaten (wie den meisten Microsoft-Formaten) der Weg über eine proprietäre API des Herstellers einfacher zu gehen.

5.2.2.1 Die Symphony-API

Das Office-Paket IBM Lotus Symphony enthält lizenzkostenfreie Editoren für das offene, ISO zertifizierte „OpenDocument Format“ (ODF) und kann aber auch mit den Microsoft Office Formaten umgehen. Lotus Symphony läuft auf mehreren Betriebssystemen. IBM Lotus Symphony baut im Kern auf OpenOffice auf. IBM bietet eine einfach zu verwendende API und dazu eine gute API-Dokumentation. Die OpenOffice-API soll wesentlich schwieriger zu verstehen sein, als die Symphony-API (welche in Lotus Notes 8 bereits enthalten ist).

Die Symphony-API ist ein klassisches Beispiel einer Message-orientierten Middleware. Das Wichtigste an einer so genannten „Message oriented Middleware“ ist die Entkopplung von Sender und Empfänger. Ein Sender erzeugt Nachrichten und übergibt sie der Middleware und die Middleware entscheidet darüber, was mit den Nachrichten passiert. Der Empfänger reagiert auf die ihm aufgetragenen Aufgaben [WbSSb03, S.113].

„Lotus Symphony“-Dokumente können mithilfe der Symphony-API und des „Symphony Developer Toolkit“ mit anderen Daten integriert werden, z. B. mit IBM Lotus Notes-Anwendungen oder sonstigen wichtigen Anwendungen. Als Entwicklungstechnologien werden viele Kombinationen unterstützt: Java mit Eclipse, LotusScript, OpenOffice und .NET (die .NET-Unterstützung ist erst für ein zukünftiges Release geplant). Typischerweise wird Lotus Symphony mithilfe der Symphony-APIs und des „Developer Toolkits“ durch Eclipse-Plug-ins erweitert oder mit Hilfe LotusScript (sollten die Produktivitätswerkzeuge innerhalb von Notes 8 verwendet werden). Auch UNO-API können verwendet werden.

Mit der Symphony-API lassen sich

- GUI-Elemente, beispielsweise in einem seitlichen Teil-Bereich (Shelf View), der Symphony-Oberfläche hinzufügen.
- in der laufenden Anwendung, Dokument öffnen und ändern.

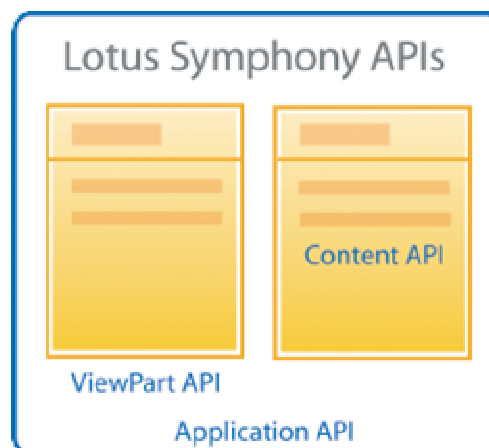


Abbildung 26: Symphony API zur Dokumentverarbeitung und ViewPart-Erstellung

Der mit dieser Diplomarbeit erstellte Prototyp ist Eclipse-basierend (Java mit Eclipse) und mithilfe dem „Symphony SDK“ erstellt worden.

Tutorials und Beispiele zur Verwendung der Symphony SDK finden sich auf den Projekt-Web-Seiten. Beispielsweise ist der „IBM Lotus Symphony Toolkit Guide“ unter

<http://symphony.lotus.com/software/lotus/symphony/developers.nsf/home#3> zu finden. Ebenso sind dort ein Dokument mit dem Titel „IBM® Lotus® Symphony™ Developers Tutorial: Building A Simple Document Workflow Plug-in“ und eines mit dem Titel „IBM® Lotus® Symphony™ Developer’s Guide“ als PDF verfügbar.

5.2.2.2 Symphony API als wesentliche Integrationskomponente

Verwendet man die Symphony API für die Generierung eines Symphony-Dokumentes zur Laufzeit, so gehört die Symphony API zu den wesentlichen Integrations-Komponenten, was folgende Abbildung darstellt. Zuvor konnte ohne deren Verwendung ein ODF-Dokument (hier im Beispiel ein ODS-Dokument) erstellt werden. Hier spielt nun aber die Symphony API auch in der Phase der Daten-Transformation eine wesentliche Rolle (nicht nur zur Zeit der Prozess-Integration oder Generierung von GUI-Elementen). Ob noch eine Mittelschicht (zusätzliche Daten-Transformation) z.B. in Form eines Web-Services eingezogen wird oder nicht, ist von diesem Sachverhalt unabhängig.

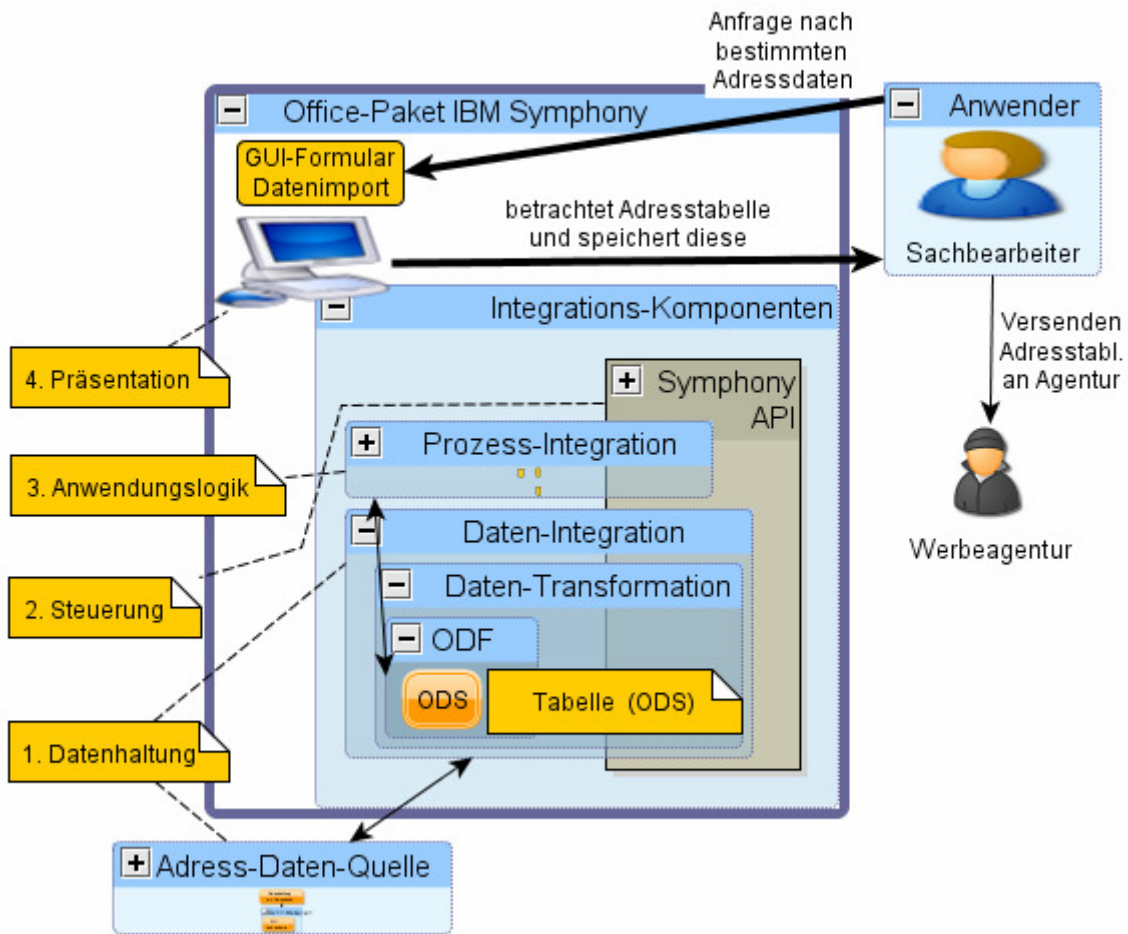


Abbildung 27: Architektur ODS mit Sym. API

5.3 Aufbau der Implementierung des Prototyp

5.3.1 Sicht auf die Java-Paket-Struktur

Die Quelltexte (der Java Programm Code des Symphony Plugins) wurden in Pakete unterteilt⁴⁶. Auf der ersten Ebene finden sich die Pakete Data, SAP und UI (für User Interface).

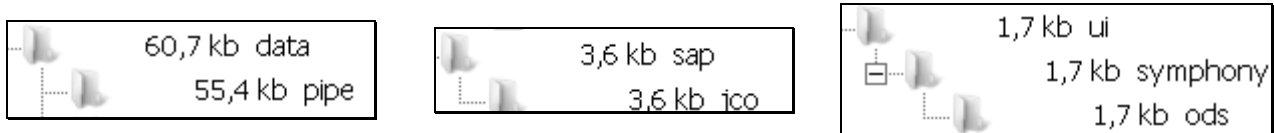


Abbildung 28: Paketstruktur des Prototyps

Im Paket `sap.jco` befinden sich kleine Klassen und eine größere, die `SAPAccess`-Klasse. Folgendes Klassendiagramm zeigt einige repräsentative Methoden.

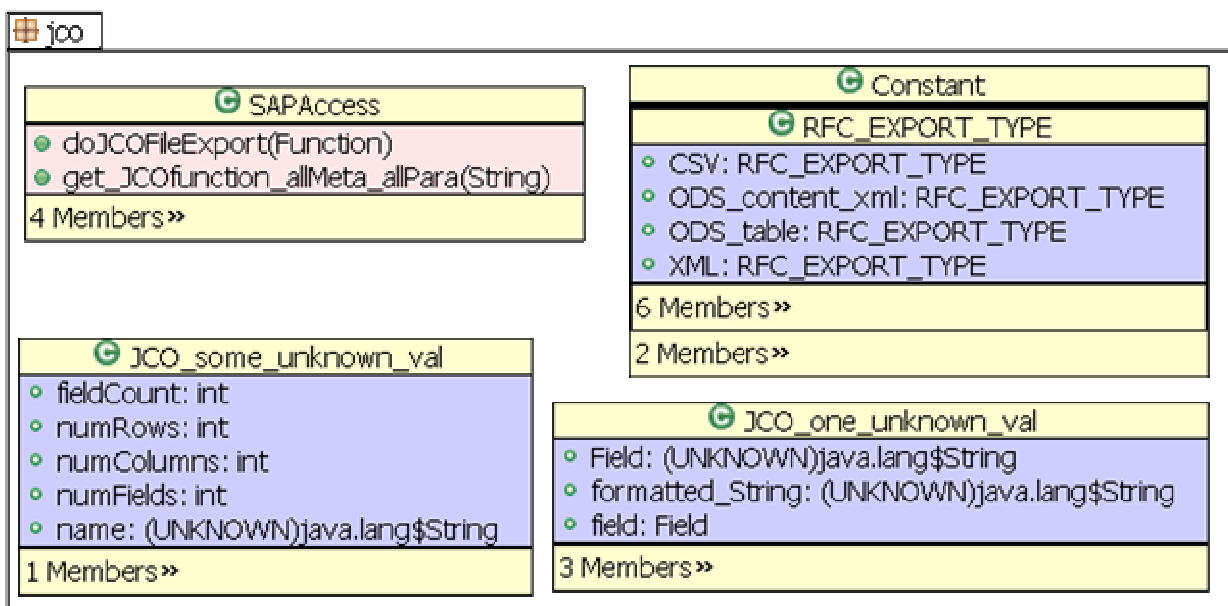


Abbildung 29: Paket `com.ibm.isicc.sap.jco`

Auszugsweise hier eine Klasse, die Klasse `JCO_some_unknown_val`:

```
package com.ibm.isicc.sap.jco;
public class JCO_some_unknown_val {
    public int fieldCount;
    public int numRows;
    public int numColumns;
    public int numFields;
    public String name;
}
```

Über GUI-Elemente in der ShelfView (seitlicher Fensterbereich innerhalb Symphony) startet der Anwender unter anderem eine Generierung bzw. fordert SAP-Daten als ODS-Dokument an. Unteres Klassendiagramm zeigt die Klassen im Paket `com.ibm.isicc.ui.symphony` mit einer Auswahl einiger repräsentativer Methoden.

⁴⁶ Pakete in Java (Java Packages) bestehen aus mehreren Klassendefinitionen und eventuell weiteren Paketen.

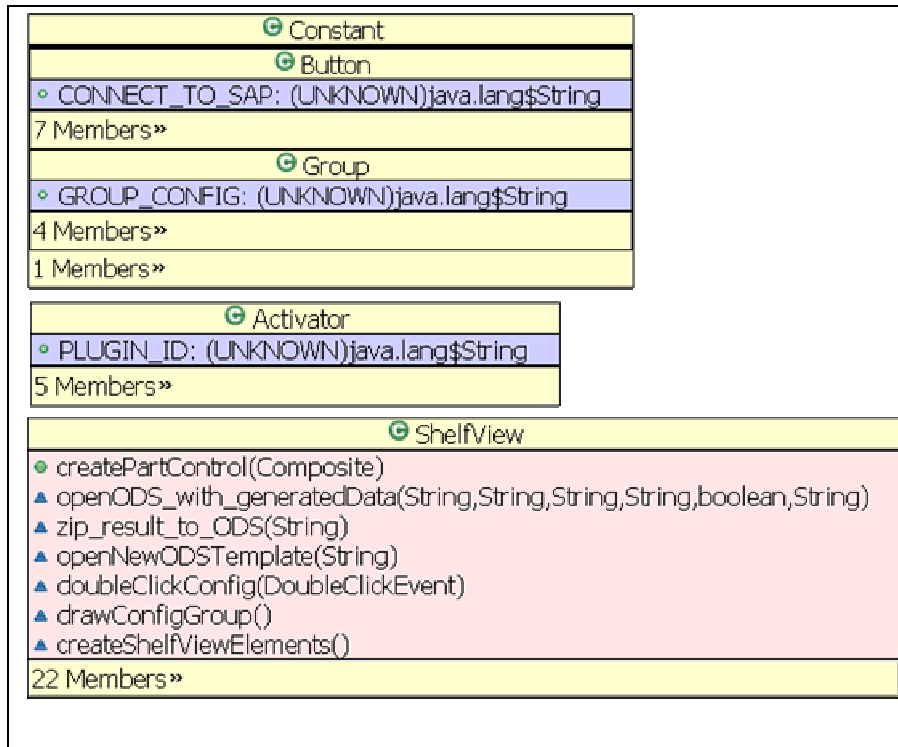


Abbildung 30: Paket com.ibm.isicc.ui.symphony

Der interne Ablauf einer Generierung im aktuellen Prototyp

Der Kern der Generierung läuft wie unten dargestellt ab.

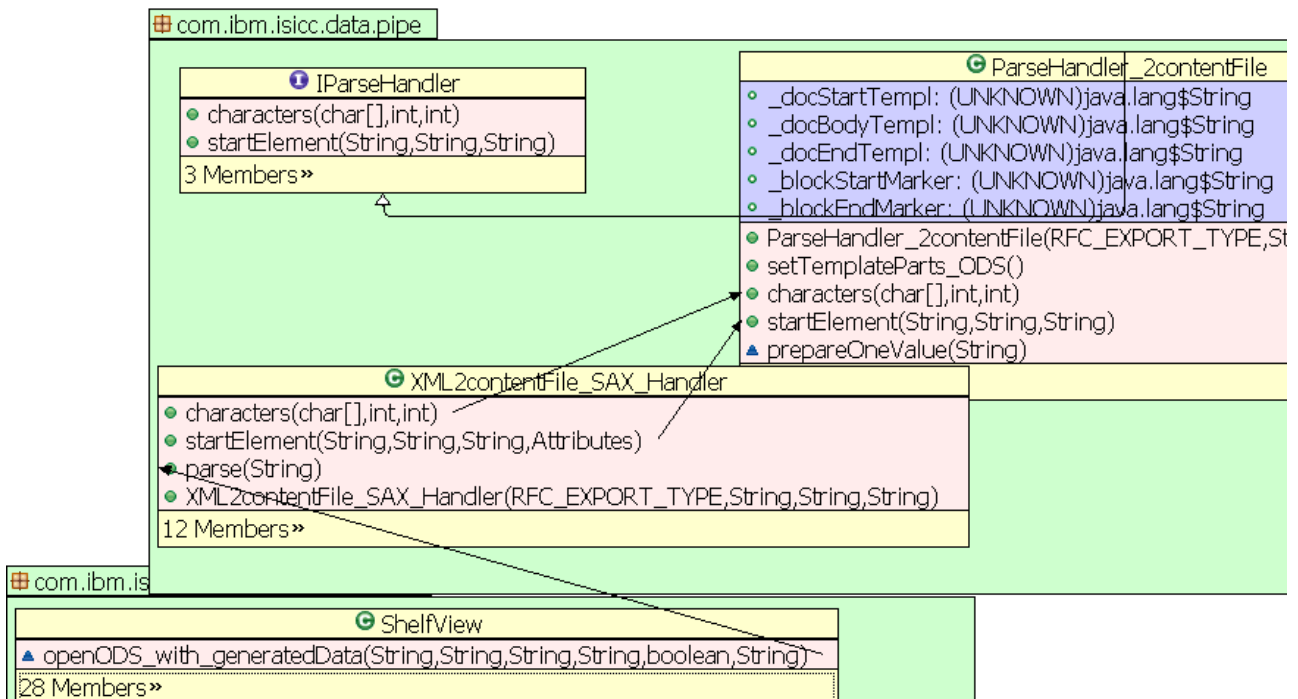


Abbildung 31: Ablauf einer Generierung im Prototyp

Über GUI-Elemente in der ShelfView steuert der Anwender und läuft eine Generierung aus. Der Vorgang zur Erstellung einer Inhalts-Datei (contentFile), dem Inhaltsteil (Body-Bereich) z.B. einer content.xml, ist oben abgebildet. Der Ablauf ist weitgehend generisch, da der Exporttyp (jeweils der erste Parameter), wie z.B. ODS oder CSV, mit angegeben werden kann. Das

funktioniert u. a. deshalb, da die `ParseHandler_2contentFile` die jeweilige Vorlage immer in zerlegt Form (Start, Body, End, ElementStart usw.) vorhält und je nach Bedarf zusammenbaut. Im Falle von ODF (nicht von CSV) als Export-Format muss die Vorlage wieder mit der Generierung zu einer neuen ODF-Datei zusammengeführt und gezippt werden. Der `XML2contentFile_SAX_Handler` ist nicht der einzige Handler derzeit (er wird verwendet wenn eine lokale XML-Datei in ein anderes Format übertragen werden soll). An seiner Position könnte auch die Klasse `Rfc2contentFile` stehen, welche intern den `ParseHandler_2contentFile` verwendet (`pH = new ParseHandler_2contentFile(targetFormat, targetValAdress, tblName, itemName)`).

Im Data-Paket findet sich für die Datentransferlösung ohne die Symphony API, welche die Funktionalität zum Entpacken von Dateien von ZIP- beziehungsweise ODF-Dateien benötigt (und umgekehrt). Folgendes Klassen-Diagramm zeigt die wichtigsten Elemente dieses Pakets.



Abbildung 32: Paket com.ibm.isicc.data

Die zur eigentlichen Daten-Transformation wichtigen Klassen (Paket `data.pipe`) und einige repräsentative Methoden sind in folgendem Klassendiagramm dargestellt.

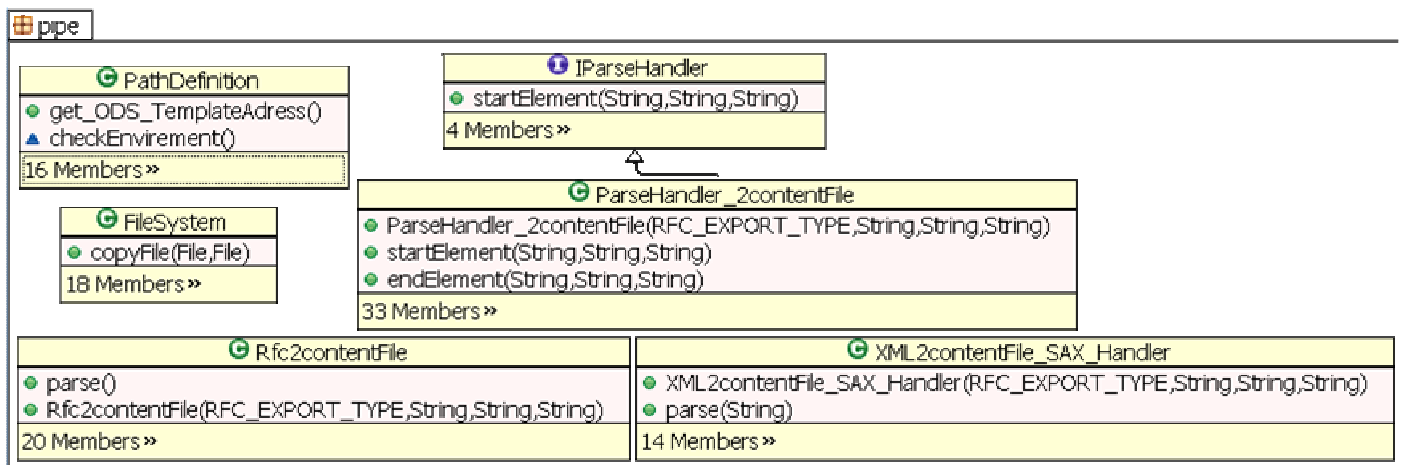


Abbildung 33: Paket com.ibm.isicc.data.pipe

5.3.2 Trennung von Verarbeitung und Design im Prototyp

Die Präsentationsschicht wird durch Code im Paket `com.ibm.isicc.ui.symphony` (und anderen wie der Symphony API) repräsentiert. Dies ist ein erster Schritt der Trennung von Verarbeitung und Design.

Außerdem sieht der Prototyp auch eine Trennung von Logik und Design während der Generierungsphase des Dokuments vor. Im Java-Code findet sich keine Layout-Information⁴⁷. Diese wird zur Laufzeit aus einer der verfügbaren ausgelesenen Vorlagen ausgelesen. Dieser

⁴⁷ Am Ende des Kapitels wird auf eine kleine Einschränkung/Ausnahme hingewiesen.

Prototyp stellt derzeit nur eine Vorlage (`ODS-template.ods`) zur Verfügung, ist aber intern so angelegt, das er beliebig viele Vorlagen verwenden kann.

Unteres Bild zeigt die Lage der Vorlagen (derzeit nur eine) im Dateisystem:

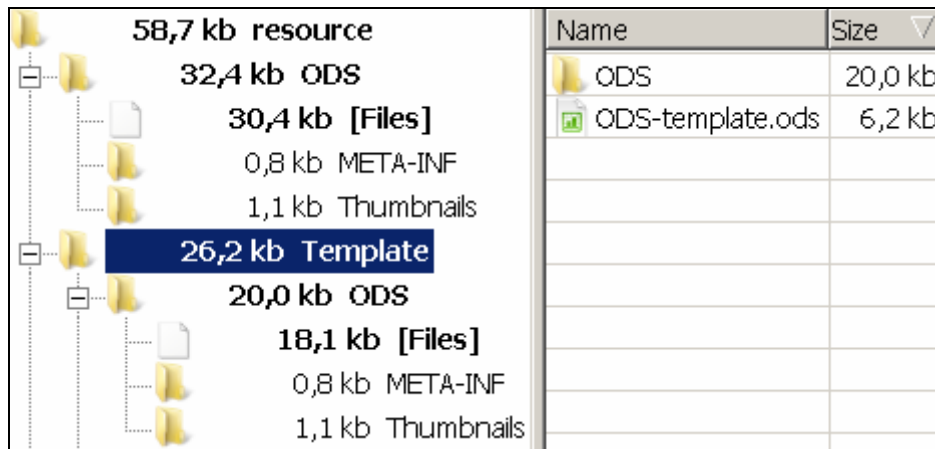


Abbildung 34: Lage der Templates im Dateisystem

Bei der Auswahl der Vorlage (Template) spielt die Klasse `PathDefinition` im Paket `com.ibm.isicc.data.pipe` eine wesentliche Rolle. Diese ist unten auszugsweise dargestellt.

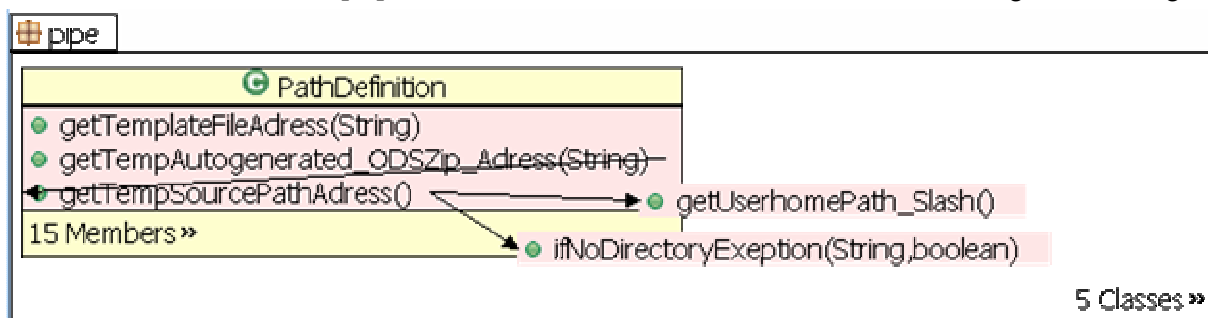


Abbildung 35: Klasse `PathDefinition` u. a. zur Abfrage der Template-Adresse

Um den Vorlagen-Code in seine Bestandteile aufzuteilen müssen layout-spezifische Trenncodes bekannt sein. Für die Speicherung und Nutzung dieser Trenncodes (Seperatoren) ist die Klasse `ParseHandler_2contentFile` zuständig, welche auszugsweise unten abgebildet ist.

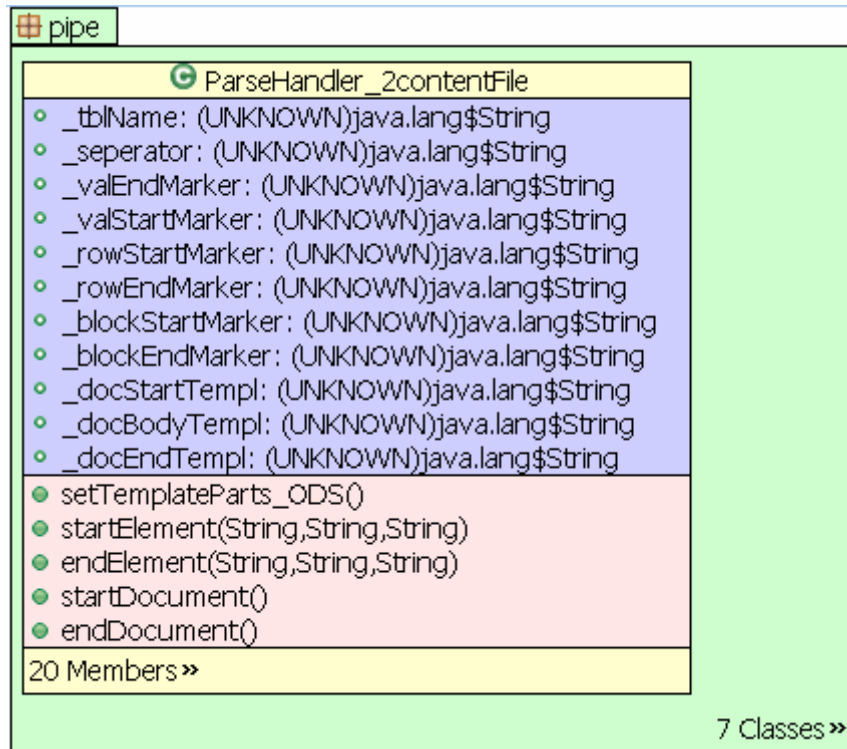


Abbildung 36: layoutspezifische Trenncodes in Klasse ParseHandler_2contentFile

Die Trenncodes liegen (leider) im Java-Code hardcodiert vor. Dies verdeutlicht folgender Quelltextauszug, welcher verdeutlicht wo, im Falle von CSV⁴⁸-Dateien, die Zuweisung des Trenncodes vorgenommen wird.

```
package com.ibm.isicc.data.pipe;
...
public class ParseHandler_2contentFile implements IParseHandler{
...
    public ParseHandler_2contentFile(RFC_EXPORT_TYPE targetFormat,
        String targetValAdress, String tblName, String itemName) {
...
        if (targetFormat.compareTo(RFC_EXPORT_TYPE.CSV) == 0) {
            _seperator = ",";
        }
    }
}
```

Quelltext 1: CSV-Trenncode hardcodiert im Java-Quelltext

Besser wäre es, mittels einer Metasprache für die Template- Verarbeitung, innerhalb der Vorlage selbst die Bereiche zur Trennung zu markieren, wie es beispielsweise im Projekt „DoSqlWeb“⁴⁹ ausnahmslos der Fall ist.

⁴⁸ CSV bedeutet Character Separated Values oder Comma Separated Values. In CSV-Dateien liegen die Werte meist durch Kommas (oft auch Semikolon oder Tabulator) getrennt vor.

⁴⁹ DoSqlWeb hieß zwischenzeitlich DOPE.PHP. Es ist seit dem 7.11.07 als OpenSource verfügbar über http://sourceforge.net/project/showfiles.php?group_id=209572

5.4 Handlungsempfehlungen

Einerseits können Daten von ERP-SAP mittels verschiedener Schnittstellen abgefragt werden und andererseits kann auch die Verarbeitung (beziehungsweise Integration) auf unterschiedliche Art und Weise erfolgen (zum Beispiel Unterscheidung nach Direktheit oder Nutzung der Office API). Im Folgenden nun ein Vorschlag mit Angabe einiger Vorteile.

5.4.1 Vorteile bei SAP nach ODF ohne Symphony-API

Erzeugt man das Dokument komplett und öffnet es erst anschließend in der Office-Applikation, ermöglicht dies eine klarere Trennung von Generierungs-Aktivität und Darstellungs-Aktivität. Eine klare Trennung ist normalerweise auch aus der Perspektive der Wiederverwendbarkeit von Vorteil.

Vorteile dieser Lösung:

- Diese Weise der Dokumentgenerierung lässt sich leicht generalisieren (mit kleinen Änderungen ist sie auch für viele andere Dokumenttypen anwendbar).
- Die indirekte Lösung eröffnet einen zusätzlichen Zugang zu den Quelldaten (Container-Format).
- Insbesondere XML- Dateien werden sehr vielfältig durch Bibliotheken, Parser etc. unterstützt.
- Erstellung und Nutzung von ODF-Vorlagen ist einfach.
- Design/Layout der Vorlagen können ohne Programmierkenntnisse verändert werden.
- weniger Bibliotheken werden benötigt (z.B. die Symphony- API wird nicht gebraucht).
- Nahezu mit allen, vielleicht allen Programmiersprachen und Scriptsprachen könnte dieser Ansatz relativ leicht implementiert werden, da er wenig Funktionalitäten erfordert (hauptsächlich entpacken, packen und parsen).
- Für nahezu jedes Betriebssystem lässt sich obiger Ansatz umsetzen, da die Wahl der Programier-/Script-Sprache nahezu frei ist (s. o.).
- Ein Batch- Modus (Massenverarbeitung) lässt sich leicht umsetzen.
- Der Ansatz kann leicht über eine GUI gesteuert werden (z.B. über „IBM Lotus Symphony“ verwendet werden).

5.4.2 Vorteile einer zusätzlichen Daten-Transformationsschicht

Je klarer die einzelnen Funktionalitäten, z.B. in Form von Komponenten geordnet sind, desto einfacher ist es einzelne Bausteine wiederzuverwenden oder auszutauschen.

Insbesondere dort, wo man sich als Nutzer von Daten noch andere Komponenten (als die eine in der konkreten Aufgabe gestellten) vorstellen kann, scheint ein zusätzlicher Schritt sinnvoll zu sein. Untere Abbildung zeigt Schritte der Datenwandlung an. Dort ist nach den Rohrdaten (z.B. proprietäre Business-Daten) u. a. ein Service zu finden, welcher die Daten in ein offenes Format wandelt. Dies ermöglicht einer größeren Zahl von Anwendungen (mit Zugriffsrechten) die Daten zu nutzen. Man bedenke bitte, dass auch immer Dienste selbst wieder Nutzer von Diensten sein können (dieser Sachverhalt findet sich auch in dieser Abbildung angedeutet).

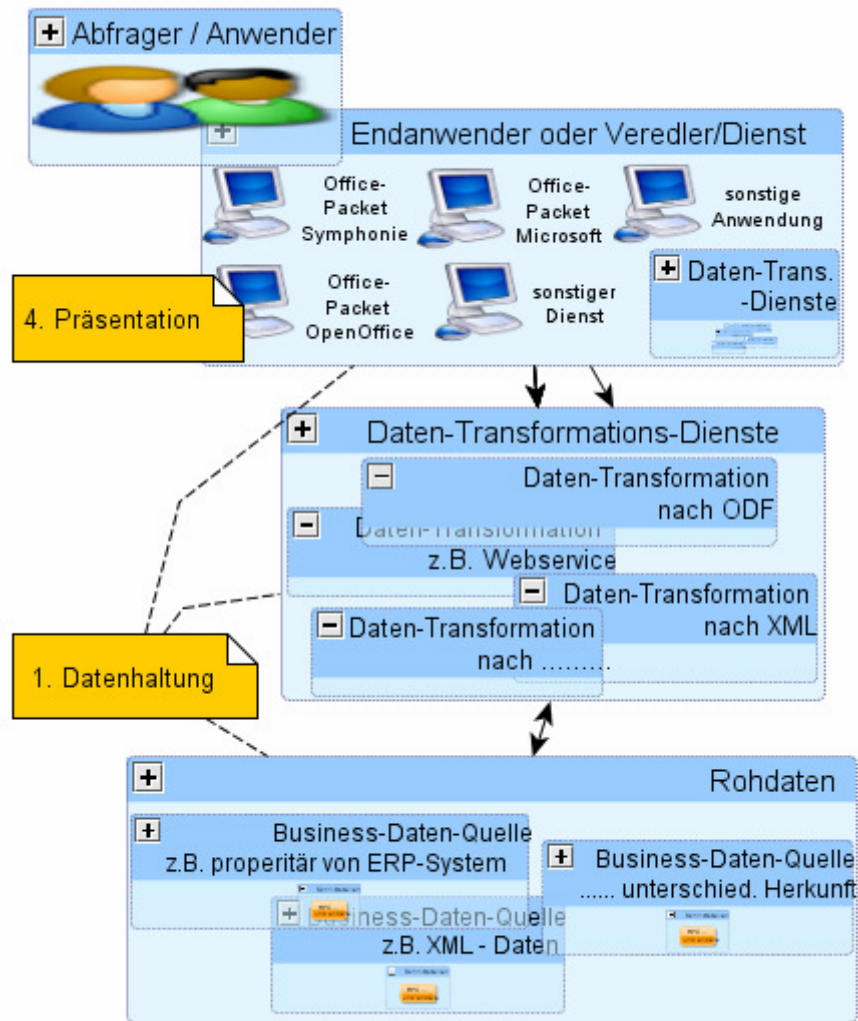


Abbildung 37: Zwischenschichten bzgl. Datenfluss

Bei Verwendung eines direkten Wegs, z.B. im Falle der Nutzung von RFC ohne zusätzliche Transformations-Schicht, kann man die Daten bereits während des Pollings (zyklischer Abfragen) mit Layoutinformationen anreichern (z.B. als ODF). Dagegen erlaubt der indirekte Weg zusätzliche Zwischenschichten.

5.4.3 Überlegungen zur Wahl der Kommunikationsform mit SAP

Synchrone Kommunikation mit Präferenz auf Geschwindigkeit und Aktualität

Bezüglich Geschwindigkeit und Aktualität der Daten ist die synchrone Art der Kommunikation gegenüber der asynchronen Kommunikation von Vorteil. Im Falle von SAP als Datenquelle basiert die Kommunikation auf RFC- oder BAPI-Aufrufen, welche in Echtzeit ablaufen. Der mit dieser Arbeit erstellte Prototyp kommuniziert synchron mit einem SAP-ERP-System.

„Möchte man ein SAP-System synchron mit einem Fremdsystem koppeln, dann ist die RFC-Technologie die bewährte Lösung“, schreibt Jeske [SfJeK05, S.107] und bezeichnet RFC als

„Technologie“ und als „Mechanismus“⁵⁰ um „auf einem entfernten System eine Funktion auszuführen.“

Jedoch ist synchrone Kommunikation nicht so zuverlässig wie die IDoc⁵¹-basierte asynchrone Datenübertragung. Bei Synchroner Kommunikation ist das Risiko gegeben, dass der Wartende die Kommunikation als fehlgeschlagen betrachtet, wobei der andere Sender weiterhin die Anfrage bearbeitet. Dies spricht für die Verwendung von asynchronen Typen der Kommunikation zwischen dem SAP ERP und IBM RPM, falls für die Übertragung der Business Objekte entsprechende IDoc's von SAP zur Verfügung gestellt werden, was nicht immer der Falls ist.

⁵⁰ Jeske [SfJeK05, S.107] bezeichnet RFC nicht nur als „Technologie“ und „Mechanismus“, sondern auch als „Protokoll“ u. „Schnittstelle“. Nekolar nennt RFC ebenfalls „Protokoll“, auch „Prozeduraufruf“ [pEuR02, S.55].

⁵¹ Intermediate Document

5.5 Datentransformation via Web-Service

Was ein Web-Service im Allgemeine ist wurde bereits im Kapitel 2.2 „Web-Service“ beschrieben. Charakteristisch für einen Web-Service sind Programmierbarkeit, Selbstbeschreibung und Kapselung.

Der normale Ablauf:

- 1) Ein Anbieter stellt einen Dienst mittels eines Web-Service bereit, welcher von externen Anwendungen angesprochen werden kann.
- 2) Der Dienst wird öffentlich gemacht, z.B. über den Verzeichnisdienst UDDI⁵² oder auch ebXML.

Durch die „Web-Services Description Language“ (WSDL), einer XML-Spezifikation, wird den Konsumenten, d.h. den angebotenen Anwendungen, bekannt, welche Funktionen der Dienst bereitstellt. Zur Kommunikation wird das „Simple Object Access Protocol“ (SOAP⁵³), ein XML-basiertes Protokoll, verwendet.

Verwendet werden kann der Dienst jetzt von unterschiedlichsten Anwendungen (egal ob .NET, PHP oder Java). Im Folgenden wird nur Java verwendet.

Dieser Zusammenhang zwischen Anbieter, Verzeichnis und Nutzer ist im Dreieckmodell (SOA-Triangle) dargestellt.

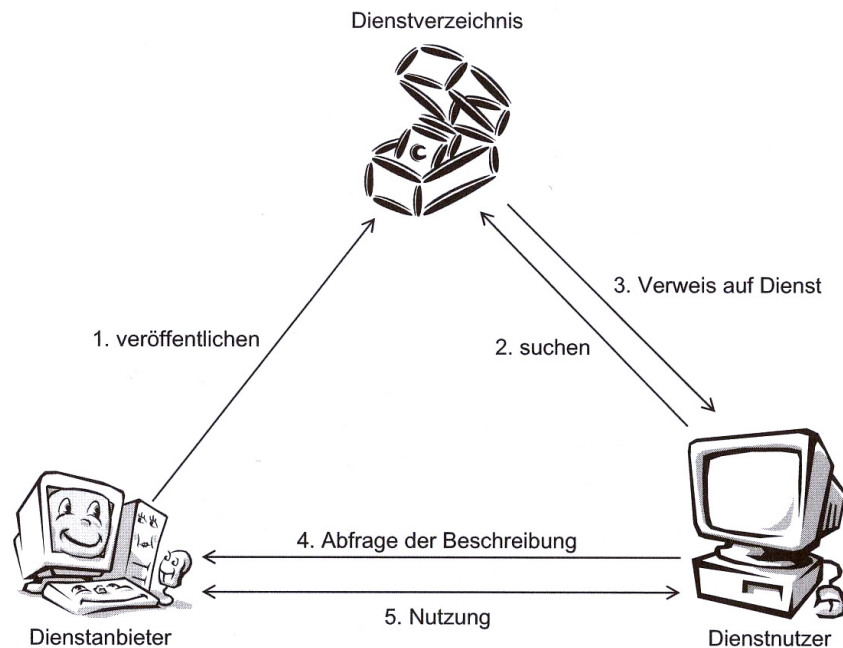


Abbildung 38: SOA-Triangle [Dostal, Jeckle, Melzer 2005, Abbildung 2.2]

⁵² UDDI ist logisch ein Service, physisch aber auf mehreren sog. Nodes (Websites) verteilt, welche z.B. von Sun, SAP, IBM, Microsoft und anderen betrieben werden.

⁵³ SOAP ermöglicht beliebige Daten zu transportieren. Eine SOAP-Nachricht besteht aus den Komponenten: SOAP-Envelope (Umschlag für die eigentliche Nachricht), SOAP-Header (enthält Informationen wie beispielsweise eine Authentifizierung), SOAP-Body (enthält die Daten).

Um eine beliebige Klasse als Web-Service zu exportieren, müssen lediglich einige Java-Annotations verwendet werden, die mit Java 5 eingeführt wurden. Und der Import von folgenden Bibliotheken ist notwendig: `javax.jws.Web-Service`, `javax.jws.soap.SOAPBinding` und `javax.jws.soap.SOAPBinding.Style`.

Beispiel:

```
package com.ibm.isicc.data.pipe;

import javax.jws.Web-Service;
import javax.jws.soap.SOAPBinding;
import javax.jws.soap.SOAPBinding.Style;

@Web-Service
@SOAPBinding(style=Style.RPC)

public class Rfc20ds {
    public int getTheDouble(int one) {
        return one + one;
    }
}
```

Die Java-Annotations `@SOAPBinding(style=Style.RPC)` legt fest, dass diese Klasse per SOAP-Protokoll gebunden werden soll und der Austausch prozedurorientiert⁵⁴ stattfindet.

Quelltext 2: Java-Klasse als Web-Service exportieren

Diese Klasse muss nun noch veröffentlicht werden. Dies kann über einen Server geschehen, wie es im folgenden Beispiel der Fall ist:

```
package com.ibm.isicc.Web-Service.server;

import javax.xml.ws.Endpoint;
import com.ibm.isicc.data.pipe.Rfc20ds;

public class Rfc20dsServer {
    public static void main (String args[]) {
        Rfc20ds r2oDienst = new Rfc20ds();
        Endpoint endpoint =
            Endpoint.publish("http://localhost:8080/r2o", r2oDienst);
    }
}
```

Quelltext 3: Server zur Veröffentlichung des Service

Oben wurde als Server das eigene System (der local host) gewählt.

Fügt man der obigen Adresse „?wsdl“ an, so dass sie `http://localhost:8080/calculator?wsdl` lautet, erhält man eine XML-Datei, genauer eine WSDL-Datei (Web-Service Description Language), welche den Dienst vollständig beschreibt. WSDL ist eine plattform-, programmiersprachen- und protokollunabhängige Beschreibungssprache für Web-Services für den Austausch von Nachrichten auf Basis von XML.

⁵⁴ RPC = Remote Procedure Call

Die WSDL-Datei dazu sieht folgendermaßen aus:

```
<definitions targetNamespace="http://test/" name="Rfc2OdsService">
  <types/>
  - <message name="getTheDouble">
    <part name="arg0" type="xsd:int"/>
  </message>
  - <message name="getTheDoubleResponse">
    <part name="return" type="xsd:int"/>
  </message>
  - <portType name="Rfc2Ods">
    - <operation name="getTheDouble" parameterOrder="arg0">
      <input message="tns:getTheDouble"/>
      <output message="tns:getTheDoubleResponse"/>
    </operation>
  </portType>
  - <binding name="Rfc2OdsPortBinding" type="tns:Rfc2Ods">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"/>
    - <operation name="getTheDouble">
      <soap:operation soapAction="" />
      - <input>
        <soap:body use="literal" namespace="http://test/" />
      </input>
      - <output>
        <soap:body use="literal" namespace="http://test/" />
      </output>
    </operation>
  </binding>
  - <service name="Rfc2OdsService">
    - <port name="Rfc2OdsPort" binding="tns:Rfc2OdsPortBinding">
      <soap:address location="http://localhost:8080/r2o"/>
    </port>
  </service>
</definitions>
```

Quelltext 4: WSDL Rfc2Ods

Aus dieser WSDL-Datei lässt sich ein Client generieren, indem auf der Shell (z.B. DOS-Fenster) folgender Befehl ausgeführt wird:

```
wsimport -keep http://localhost:8080/r2o?wsdl
```

Danach sind folgende vier generierte Dateien zu finden. In der ersten Zeile ist das Interface, danach der Service, zu finden:

```
→ Interface: Rfc2Ods.java,          Rfc2Ods.class,
→ Service:   Rfc2OdsServer.java,    Rfc2OdsServer.class
```

Hier zunächst ein Quelltext-Auszug aus der generierten Interface-Klasse:

```
package com.ibm.isicc.data.pipe;
import javax.jws.WebMethod;
/** * This class was generated by the JAX-WS RI.
@WebService(name = "Rfc2Ods", targetNamespace = "http
@SOAPBinding(style = SOAPBinding.Style.RPC)
public interface Rfc2Ods {
    * @param arg0
    @WebMethod
    @WebResult(partName = "return")
    public int getTheDouble(
        @WebParam(name = "arg0", partName = "arg0")
        int arg0);
}
```

Quelltext 5: Auszug der Interface-Klasse Rfc2Ods

Die Quelltexte der generierten Dateien ändert man am besten nicht, sondern erstellt typischerweise einen Client, welcher die Interface- und Service-Klasse verwendet, so als ob es eine lokale Klasse wäre.

```
public static void main(String args[]) {
    Rfc2OdsService service = new Rfc2OdsService ();
    Calculator r2o = service.getRfc2OdsPort ();
    System.out.println("Double: " + r2o.getTheDouble(5));
}
```

Quelltext 6: Client der Interface- und Service-Klasse verwendet

Damit Java die Klassen auch findet, müssen diese wie es auch mit allen anderen Klassen üblich ist, zuvor importiert werden.

Man kann also mithilfe eines Web-Services Funktionalitäten auslagern, jedoch weiterhin so arbeiten, als wären sie (bezüglich der Darstellung im Quelltext) lokal vorhanden. Außerdem erhält man zusätzliche Flexibilität, weil einem frei steht in welcher Sprache man die Implementierung umsetzen möchte und hat nebenbei damit eine zusätzliche Schnittstelle zu den Daten geschaffen.

Im Falle des mit dieser Diplomarbeit erstellten Prototyps bietet es sich an, die Datentransformation von Rohdaten in ein offenes Format (z.B. RFC nach ODF) komplett als Web-Service auszulagern, um damit größere Flexibilität bei der Wahl der Implementierungssprache und der Nutzungsmöglichkeiten zu gewinnen.

Ein guter Ansatz wäre es auch (um die Layout-Informationen aus der Netzwerkübertragung zu erhalten), die Konvertierung in die entsprechende ODF-Datei (im Beispiel ODS) erst lokal vorzunehmen und vom Web-Service lediglich die Daten, allerdings in einem offenen und verbreiteten Format, zu erhalten. Dabei würde der Web-Service in erster Linie als Normierungsinstrument wirken. Das wirft die Frage nach dem Format und dessen Struktur auf.

5.6 Diagramm mit Fokus auf den Konfigurationsablauf

Gewünscht, aber im Prototyp nicht implementiert, sind zwei verschiedene Zugänge für zwei verschiedene Symphony-Anwender. Einen Superuser bzw. Konfigurator, welcher die Szenarien festlegt und einen Normalanwender (Sachbearbeiter), welcher die angelegten Szenarien abrufen. Letztlich wird durch diese Schritte ein, so im Use-Case-Diagramm⁵⁵ genanntes, SubSet definiert. Der "Actor" bzw. Symphony-Anwender weiß nicht, was RFC, BAPI usw. ist. Diese technischen Begriffe dürfen beim Normal-Anwender nicht mehr auftauchen.

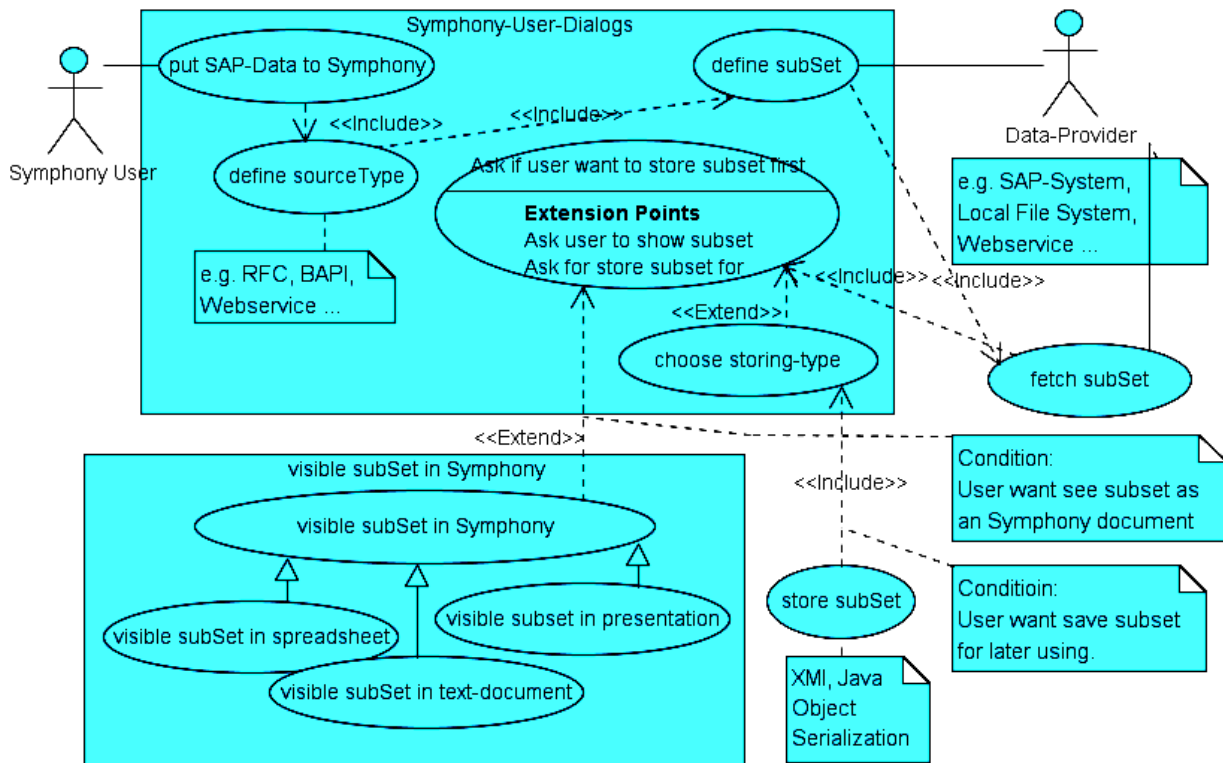


Abbildung 39: Diagramm mit Focus auf den Konfigurations-Ablauf

Erklärungen zu diesem Use-Case-Diagramm:

Hauptakteure:

- Officebenutzer
- Superuser (macht Voreinstellungen, damit Office-Benutzer nur sprechende Szenarien-Namen zur Auswahl hat).
- Data-Provider (beispielsweise SAP System, Web-Service, Local Dateisystem etc.)

Benutzerdialoge:

In dem dargestellten Szenario befinden sich (idealisiert) nur wenige Anwenderdialoge. Der Office-Benutzer fordert, über ein Menü aus seiner Tabellenkalkulation, Daten aus einem Business System (beispielsweise SAP System) an. Diese Datenanforderung vollzieht sich in Schritten, welche ein sogenannter Import-Wizard koordiniert.

⁵⁵ Use-Case-Diagramme „werden zur Darstellung funktionaler Anforderungen aus Sicht der Anwender („Actors“) verwendet“ [MvIE04, S.90]. Wobei die Akteure nicht nur Benutzer, sondern auch „Nachbarsysteme“ [DUKfd05, S.35] sein können, wie hier im Beispiel das SAP-System.

Zuerst legt der Superuser den Schnittstellentyp der Quelle selbst fest, was die Menge der möglichen Daten entscheidend definiert⁵⁶, und ergänzt – falls erforderlich – fehlende Zugangsdaten (Benutzername, Passwort usw.). Danach wird der Superuser aufgefordert eine Menge namentlich zu nennen (beispielsweise Namen einer RFC-Funktion) und anschließend eine Teilmenge festzulegen.

Dieser letztere Schritt, also das Festlegen der Teilmenge, geschieht erst nach einer erfolgreichen Abfrage von Metadaten, wie zum Beispiel Inputparametern für die RFC- Schnittstelle. Letztlich muss noch festgelegt werden, was mit den Daten geschehen soll. Dies bestimmt der Office-Benutzer. Hierzu sind mehrere Verarbeitungsweisen denkbar, wie beispielsweise:

- Speichern der Abfragemenge (Ja/Nein)
mit anschließender Eingabe des Speichertyps.
- Anzeigen der Abfragemenge (Ja/Nein)
mit anschließender Abfrage von Darstellungsfunktionen

⁵⁶ Es sind wesentlich mehr Daten per RFC verfügbar als über die BABI-Schnittstelle.

Ehrenwörtliche Erklärung

Ich versichere,

1. Dass ich die Kapitel der Diplomarbeit, für die ich als Verfasser genannt werde, selbständig verfasst habe,
2. dass ich keine anderen, als die angegebenen Quellen und Hilfsmittel benutzt habe,
3. dass ich diese Arbeit bei keinem anderen Prüfungsverfahren vorgelegt habe.

Heidelberg, im November 2005

Sebastian Lauffer

Zusammenfassung und Ausblick

Diese Diplomarbeit bietet einen Beitrag zur Integration von Information in Office-Pakete. Es wurden nicht alle zu anfangs gesetzten Ziele⁵⁷ erreicht. Es konnte anhand eines lauffähigen Prototyps gezeigt werden dass eine Integration von SAP-ERP-Daten nach IBM Lotus Symphony möglich ist.

Zur Bewertung des Ergebnisses verwende ich nun die mit Kürzel versehenen selbstauferlegten Fein-Ziele (wie am Anfang der Diplomarbeit definiert).

Zu (Ziel A) Überlegung und Diskussion einer allgemeine Lösungen für die Einbindung von Business-Daten in Office Produkte. Ich konnte schnell feststellen das für eine ganz allgemeine Lösung - beziehungsweise nahezu allgemeine Lösung - die Ausgangslage einerseits begrifflich zu unscharf und andererseits (derzeit noch) zu heterogenen ist. Beispielsweise ist eine Konsolidierung zu offenen Standards zwar feststellbar (ODF, XML, Web-Services), wird aber noch nicht von allen unterstützt. Auch die Umstellung auf Web-Services in Unternehmen wird noch einige Jahre dauern [♣PUUeW08].

Zu (Ziel B) Überlegung und Diskussion einer Lösung, Erstellung einer Architektur (B.1), eines Lösungsansatzes (B.2) und Prototyp (B.3) usw. Der Prototyp wurde wie gefordert Eclipse-basierend und mit dem Symphony SDK erstellt. Konzeptionell erlaubt die Art seiner Implementierung Erweiterungen und das Wiederverwenden einzelner Java-Klassen oder auch Java-Pakete. Die Daten wurden wie gefordert aus SAP mittels RFC/BAPI via JCo abgeholt und als Datei gespeichert. Erfolgreiche Tests wurden mit folgendem Speicherformaten durchgeführt: CSV und ODS.

Der Prototyp ist intern so modular umgesetzt, dass prinzipiell auch andere Formate zur Speicherung verwendet werden können. Diese Flexibilität wurde aus Zeitgründen jedoch nach außen, das heißt im grafischen User Interface in der Symphony Applikationen, noch nicht sichtbar.

Leider wurde der Wunsch, Daten aus SAP via Web-Service und sonstigem Web-Services lesen zu können im Prototyp nicht umgesetzt und nur in den theoretischen Grundlagen und im Kapitel Konzeption davon gesprochen. Ein Web-Service könnte auch als Normierungsinstanz eingesetzt werden um Daten unterschiedlicher ERP-Systeme in einem einheitlichen Format zu fassen.

Erleichtern würde eine Verallgemeinerung für andere SAP Lösungen auch die Tatsache, dass alle SAP-Kernel die gleichen Schnittstellen anbieten. Allerdings benutzen andere SAP-Lösungen auch inhaltlich anders aufgebaute SAP-Datenobjekte (SAP-Businessobjekte). Aber, technisch-konzeptionell gesehen, lässt sich diese Lösung auf andere SAP-Systeme, wie „Suplay Chain Managment“-Systeme, übertragen.

⁵⁷ siehe Kapitel 1 „Ziel / Aufgabenformulierung“

Ausblick

Es wurde vom ISICC zu Beginn der Arbeit in Aussicht gestellt, den Prototyp eventuell zur Weiterentwicklung an das IBM Labor zu übergeben. Gut vorstellbar, dass dort der Web-Service-Ansatz weiter verfolgt wird und an der Kopplung zu unterschiedlichen SAP-Systemen und später vielleicht sogar am Anschluss unterschiedlicher ERP-Systeme gearbeitet wird.

Nach einer Meldung, der PR-COM Gesellschaft für strategische Kommunikation mbH [♠PUUeW08], ergab eine kürzlich von Progress in Auftrag gegebene Umfrage bei mehr als 500 Unternehmen, dass 65 Prozent der Unternehmen SOA als die "Architektur der Zukunft" sehen.

SOA erleichtert den Schritt zu Individuallösungen, durch die relativ einfache Zusammenstellung neuer Lösungen.

Zukünftig wird es wieder mehr Individuallösungen geben. Die breite Nutzung von offenen Formaten und Web-Services oder Mashups machen dies möglich. Auch Teich [□DrWzS08, S.226] sagt dies ganz deutlich: „Individualität ist wieder gefragt. (...) Individualität ist ein Erfolgsfaktor der Zukunft — besser sein als die Mitbewerber. Doch dieser Erfolg stellt sich erst dann ein, wenn sich Softwarelösungen nach dem Anwender richten und nicht umgekehrt.“ Untere Abbildung veranschaulicht dies.

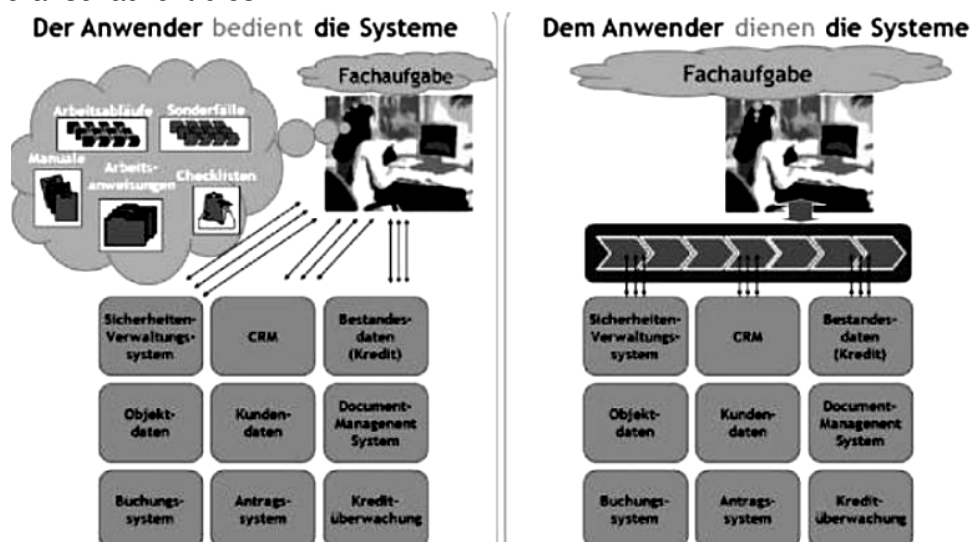


Abbildung 17: Paradigmenwechsel in der Ausprägung von IT-Systemen [□DrWzS08, S.226]

Da SOA ist etwas, was aktiv getan werden muss, nicht etwas, das einfach gekauft werden kann. SOA verlangt oftmals erhebliche Investition. Unternehmen, die bei der Umsetzung Fehler machen, stehen mitunter am Ende plötzlich instabilen und unflexibleren Systemen gegenüber. Und die Umstrukturierung von Spaghetti-Code⁵⁸ in kleine geschlossene Funktionseinheiten, wie es Web-Services erfordern, sowie die Klärung einiger Fragen, z.B. sicherheitsrelevanter Art, wird einige Unternehmen noch Jahre kosten. So ist es auch nicht verwunderlich, dass in der oben genannten Umfrage [♠PUUeW08] die 66 Prozent (der befragten 500 Unternehmen), welche SOA für die kommende "de-facto Architektur" halten, nur 41 Prozent damit rechnen, dass dies bereits in den nächsten drei Jahren der Fall sein wird und nur 52 Prozent meinen, dass dies bis zu fünf Jahre dauern könnte.

⁵⁸ Spaghetti-Code ist ein bildhafter Ausdruck für ein Programm, in dem, durch unbedingte Sprünge (z.B. zahlreiche verschachtelte Sprunganweisungen), der Programmablauf schwer nachzuvollziehen ist.

Abstract

This diploma thesis deals with the integration of business data into Office packages.

The trend toward OpenOffice standards such as XML, ODF or Web-Services, improves the chance of large, almost universal, integration solutions.

The ODF (OpenDocument format) is an ISO / IEC standard for Office document exchange.

The market for the ERP to Office integration solutions is highly competitive and mainly split between Microsoft and IBM.

The latter sets more value on OpenOffice technologies with its project "Atlantic" and thus comes also much cheaper on the market than Microsoft with its solution called "Duet".

Part of this diploma thesis is a prototype which represents data from SAP in one of the newest Office packages (IBM Lotus Symphony).

Thereby different solutions are discussed and partially implemented.

Multi-component solutions bring additional flexibility and modularization possibilities. This will allow the connection of other Office packages and systems and an ideal place for further deployments is created where the raw data, for example, Data from different ERP systems can be transformed into a single format.

With each data transformation the task of implementing the data types correctly into the new format emerges, therefore the choice of the formats is particularly important. If you adhere as early as possible to OpenOffice standards, you may save later a step of data transformation. Each of the solutions discussed requires, as the last or the only step of data transformation, the transformation into an ODF format. It will be found out that this last transformation step can be carried out in two ways. On the one hand, in the case of ODF it is particularly easy to carry out the transformations independent of the Symphony API (because ODF consists mainly of zipped XML files), and only then to present the document in the application. On the other hand, it is possible using the API Symphony to produce the ODF document. The first option allows for a clearer separation of generation and presentation and is therefore preferred.

Quellenverzeichnis

Bücher

☐ABDWE06:

Analyzing Business Data With Excel

Von Gerald Knight

Veröffentlicht von O'Reilly, 2006

ISBN 0596100736, 9780596100735

248 Seiten

☐ASSfP07:

Advanced SQL SQL für Praxis und Studium: SQL für Praxis und Studium

Von Daniel Warner

Veröffentlicht von Franzis Verlag, 2007

ISBN 3772371701, 9783772371707

533 Seiten

☐AuDPv00:

Algorithmen und Datenstrukturen. Pascal-version: mit 30 Tabellen, 69 Übungen

Von Niklaus Wirth

Veröffentlicht von Vieweg+Teubner Verlag, 2000

ISBN 3519222507, 9783519222507

320 Seiten

☐BEsAE07:

Basiswissen ERP-systeme: Auswahl, Einführung& Einsatz betriebswirtschaftl. Standardsoftware

Von Martin Hesseler, Marcus Görtz

Veröffentlicht von W3I GmbH, 2007

ISBN 3937137386, 9783937137384

406 Seiten

☐CaCFo93:

Closure and Convergence: A Foundation of Fault-Tolerant Computing

Von Anish Arora und Mohamed Gouda Veröffentlicht von Zeitschrift IEEE Press Piscataway, NJ, USA, 1993

Band 19, Ausgabe 11 (November 1993)

<http://www2.computer.org/portal/web/csdl/doi/10.1109/32.256850>

ISSN:0098-5589

Seiten: 1015-1027

☐CfIKu06:

Controlling für Industrieunternehmen: Kompakt und IT-unterstützt. Mit Fallstudie

Von Jürgen Bauer, Egbert Hayessen

Veröffentlicht von Vieweg+Teubner Verlag, 2006

ISBN 3834800678, 9783834800671

240 Seiten

 COSKu05:

Chefsache Open-Source: Kostenvorteile und Unabhängigkeit durch Open-Source

Von Theo Saleck

Veröffentlicht von Vieweg+Teubner Verlag, 2005

ISBN 3528058730, 9783528058739

234 Seiten

 CSmEK02:

CRM-Systeme mit EAI: Konzeption, Implementierung und Evaluation; mit 8 Tabellen

Von Matthias Meyer

Mitwirkende Personen Matthias Meyer

Veröffentlicht von Vieweg+Teubner Verlag, 2002

ISBN 3528057955, 9783528057954

250 Seiten

 DDIRV08:

Die Dritte Industrielle Revolution

Veröffentlicht von Bundesministerium für Umwelt, Naturschutz und Reaktorsicherheit (BMU),
September 2008

Referat Öffentlichkeitsarbeit 11055 Berlin - www.bmu.de

(kostenlos und aus 100 % Altpapier)

148 Seiten

 DeFzs02:

Digital erfolgreich: Fallstudien zu strategischen E-business-konzepten

Von Petra Schubert, Dorian Selz, Patrick Haertsch

Veröffentlicht von Springer, 2002

ISBN 3540435328, 9783540435327

303 Seiten

 DgUDu07:


Der gläserne User- Datensicherheit und Anonymität im Internet- Traum oder Realität

Von Sandro Birke

Veröffentlicht von GRIN Verlag, 2007

ISBN 3638683753, 9783638683753

88 Seiten

 DHzE01:

Dokumentenmanagementsysteme: Handbuch zur Einführung

Von Harald Klingelhöller

Veröffentlicht von Springer, 2001

ISBN 3540412506, 9783540412502

254 Seiten

 DIMuL01:

Detailliertere Informationen

Mengen und Längen: Mengen und Längen

Von Horst Schinköthe

Veröffentlicht von BoD – Books on Demand, 2001
ISBN 3831107017, 9783831107018
124 Seiten

☞DMDM06:
Diskrete Mathematik: Diskrete Mathematik
Von Karl-Heinz Zimmermann
Veröffentlicht von BoD – Books on Demand, 2006
ISBN 3833455292, 9783833455292
412 Seiten

☞DMGuZ99:
Dokumenten-Management: Grundlagen und Zukunft
Von Ulrich Kampffmeyer, Barbara Merkel
Veröffentlicht von PROJECT CONSULT GmbH, 1999
ISBN 3980675602, 9783980675604
320 Seiten

☞DrWzS08:
Der richtige Weg zur Softwareauswahl: Pflichtenheft, Compliance, Erfolgskontrolle
Von Irene Teich, Walter Kolbensschlag, Wilfried Reiners
Veröffentlicht von Springer, 2008
ISBN 3540712615, 9783540712619
240 Seiten

☞DSkPS08:
Die Softwareindustrie: Ökonomische Prinzipien, Strategien, Perspektiven
Von Peter Buxmann, Heiner Diefenbach, Thomas Hess
Veröffentlicht von Springer, 2008
ISBN 3540718281, 9783540718284
216 Seiten

☞DuDvA08:
Datenbereinigung und Datenübernahme von Altsystem nach mySAP ERP 05 bei einem internationalen Automobilzulieferer
Von Oliver Jung, Dipl. Wirtschaftsinformatiker (BA) Oliver Jung
Veröffentlicht von BoD – Books on Demand, 2008
ISBN 3638905721, 9783638905725
64 Seiten

☞DUkfd05:
Die UML-kurzreferenz 2.0 für die Praxis: Kurz, bündig, ballastfrei
Von Bernd Oestereich
Veröffentlicht von Oldenbourg Wissenschaftsverlag, 2005
ISBN 3486577883, 9783486577884
182 Seiten

☞DuXKA02:

Datenbanken und XML: Konzepte, Anwendungen, Systeme
Von Wassilios Kazakos, Andreas Schmidt, Peter Tomczyk
Mitwirkende Personen P Lockemann
Veröffentlicht von Springer, 2002
ISBN 354041956X, 9783540419563
352 Seiten

EGEEi04:
EDV-Grundwissen: Eine Einführung in Theorie und Praxis der modernen EDV
Von Manfred Precht, Nikolaus Meier, Dieter Tremel
Veröffentlicht von Pearson Education Deutschland, 2004
ISBN 3827321298, 9783827321299
624 Seiten

EidI06:
Einführung in die Informatik
Von Heinz-Peter Gumm, Manfred Sommer
Veröffentlicht von Oldenbourg Wissenschaftsverlag, 2006
ISBN 3486581155, 9783486581157
894 Seiten

EidIA06:
Einführung in die Internetökonomie: Arbeiten und investieren in einer modernen Wirtschaft
Von Harald Meisner
Veröffentlicht von LIT Verlag Berlin-Hamburg-Münster, 2006
ISBN 3825874125, 9783825874124
200 Seiten

EidW07:
Einführung in die Wirtschaftsinformatik
Von Michael Bächle, Arthur Kolb
Veröffentlicht von Oldenbourg Wissenschaftsverlag, 2007
ISBN 3486583638, 9783486583632
131 Seiten

ERPuS04:
Enterprise Resource Planning und Supply Chain Management: Architektur und Funktionen
Von Norbert Gronau
Veröffentlicht von Oldenbourg Wissenschaftsverlag, 2004
ISBN 3486272659, 9783486272659
315 Seiten

ESWCT08:
Einfach - So! Word & Co: Textverarbeitung mit Ashampoo TextMaker, OpenOffice Writer und
Microsoft Word
Von Monika Pross
Veröffentlicht von BoD – Books on Demand, 2008
ISBN 3837026566, 9783837026566

316 Seiten

☞EU07:

Elektronische Unternehmenskommunikation
Von Frank Martin H
Veröffentlicht von Deutscher Fachverlag, 2007
ISBN 3866410786, 9783866410787
609 Seiten

☞EVSvm08:

ERP Value: Signifikante Vorteile mit ERP-systemen
herausgegeben von Olaf Jacob
Mitwirkende Personen Roland Abele, Olaf Jacob, Heiko Eberle, Dirk Hammermann, Jürgen
Hawig, Gerhard Kaminski, Frank Lutz, Olaf Passenheim, Frank Roth, Johannes Stephany, D.
Doeffinger, V Foerster, M Miller, L Rattmann, O Toufar
Veröffentlicht von Springer, 2008
ISBN 3540744398, 9783540744399
250 Seiten

☞EvWSi07:

Einsatz von Web-Services im E-commerce
Von Martin Schädler
Veröffentlicht von GRIN Verlag, 2007
ISBN 3638652238, 9783638652230
88 Seiten

☞FaWSo07:

Founders at Work: Stories of Startups' Early Days
Von Jessica Livingston
Veröffentlicht von Apress, 2007
ISBN 1590597141, 9781590597149
456 Seiten

☞FvISU07:

Führen von IT-Service-Unternehmen Zukunft erfolgreich gestalten: Zukunft erfolgreich gestalten
Von Kay P Hradilak
Veröffentlicht von Vieweg+Teubner Verlag, 2007
ISBN 3834802603, 9783834802606
163 Seiten

☞GiTuP05

Geoinformatik in Theorie und Praxis:
In Theorie und Praxis
Von Norbert de Lange
Veröffentlicht von Springer, 2005
ISBN 3540282912, 9783540282914
454 Seiten

 GJtLa04:

Grundkurs Java-technologien: Lernen anhand lauffähiger Beispiele- Konzepte einfach erklärt- die komplette Einführung in J2SE und J2EE

Von Erwin Merker

Veröffentlicht von Vieweg+Teubner Verlag, 2004

ISBN 3528058986, 9783528058982

402 Seiten

 GWAME04:

Grundkurs Wirtschaftsinformatik (Abts/Mülder): Eine kompakte und praxisorientierte Einführung

Von Dietmar Abts, Wilhelm Mülder

Veröffentlicht von Vieweg+Teubner Verlag, 2004

ISBN 3528455039, 9783528455033

467 Seiten

 GWEku04:

Grundkurs Wirtschaftsinformatik, Eine kompakte und praxisorientierte Einführung

Von Dietmar Abts, Wilhelm Mülder

Veröffentlicht von Vieweg, Braunschweig/Wiesbaden, 2004

ISBN 3528355034

400 Seiten

 HoRoV08:

Handbook of Research on Virtual Workplaces and the New Nature of Business Practices

Von Pavel Zemliansky, Kirk St Amant

Mitwirkende Personen Pavel Zemliansky, Kirk St Amant

Veröffentlicht von Idea Group Inc (IGI), 2008

ISBN 1599048930, 9781599048932

733 Seiten

 IIPuK02:

Integrierte Informationsverarbeitung 2. Planungs- und Kontrollsysteme in der Industrie

Von Peter Mertens, Joachim Griese

Veröffentlicht Von Gabler Verlag; Auflage: 9. Aufl. 2002

ISBN 340919102X, 9783409191029

270 Seiten

 IM06:

Industrielles Management

Von Karl-Werner Hansmann

Veröffentlicht von Oldenbourg Wissenschaftsverlag, 2006

ISBN 3486580582, 9783486580587

428 Seiten

 luWli04:

Individualisierung und Wissensarbeit: Individualisierungsprozess in Unternehmen und ihre Auswirkungen am Beispiel der Personalorganisation

Von Sandra M. Green

Veröffentlicht von DUV, 2004
ISBN 3824407833, 9783824407835
265 Seiten

☐IWMKz04:
Interorganisatorische Wissensnetzwerke: Mit Kooperationen zum Erfolg
herausgegeben von Rolf Caspers, Roland Berger Strategy Consultant, Nils Bickhoff, Thomas Bieger
Veröffentlicht von Springer, 2004
ISBN 3540201823, 9783540201823
353 Seiten

☐IzFRu08:
Internetworking zwischen X.25, Frame Relay und ATM
Von Claus Strobel
Veröffentlicht von GRIN Verlag, 2008
ISBN 3638891224, 9783638891226
288 Seiten

☐JliLi04:
Jetzt lerne ich Linux im Büro: Office-aufgaben einfach und sicher unter Linux meistern
Von Michael Kofler
Veröffentlicht von Pearson Education Deutschland, 2004
ISBN 382726782X, 9783827267825
429 Seiten

☐JliXD05:
Jetzt lerne ich XML: Der einfache Einstieg in den führenden Dokumenten- und Web-standard
Von Günter Born
Veröffentlicht von Pearson Education Deutschland, 2005
ISBN 3827268540, 9783827268549
442 Seiten

☐LA08
Lineare Algebra
Von Siegfried Bosch
Veröffentlicht von Springer, 2008
ISBN 3540764372, 9783540764373
297 Seiten

☐LNDAL04:
Lotus Notes/ Domino Administration: Lotus Groupware verwalten, Version 5 bis 6.5
Von Nadin Ebel
Veröffentlicht von Pearson Education Deutschland, 2004
ISBN 3827322006, 9783827322005
1272 Seiten

☐MDA07:

Model Driven Architecture

Von Dennis Marc Busch, Diplom-Wirtschaftsinformatiker Univ. Dennis Marc Busch

Veröffentlicht von GRIN Verlag, 2007

ISBN 363871702X, 9783638717021

56 Seiten

mHTGu03:

mySAP HR – Technische Grundlagen und Programmierung Von Ewald Brochhausen, Jürgen Kieleisch, Jürgen Schnering, Jens Staeck Veröffentlicht von Galileo Pres GmbH, 2003

ISBN 3898425487, 9783898425483

473 Seiten

OkDPd08:

Online-kommunikation: Die Psychologie der neuen Medien für die Berufspraxis

Von Annette Kielholz

Veröffentlicht von Springer, 2008

ISBN 3540763287, 9783540763284

321 Seiten

Mp08:

Mashups programmieren

Von Denny Carl, Jörn Clausen, Denny Carl, Marco Hassler, Anatol Zund

Veröffentlicht von O'Reilly Germany, 2008

ISBN 3897217589, 9783897217584

280 Seiten

MvIEm04:

Modellierung von Informationssystemen: Ein methodischer Leitfaden zur Projektabwicklung

Von Adrian Specker

Veröffentlicht von vdf Hochschulverlag AG, 2004

ISBN 3728129844, 9783728129840

433 Seiten

PEMEu06:

Profikurs Eclipse 3: Mit Eclipse 3.2 und Plugins professionell Java-anwendungen entwickeln- von UML bis JUnit

Von Gottfried Wolmeringer, Thorsten Klein

Veröffentlicht von Vieweg+Teubner Verlag, 2006

ISBN 3834800074, 9783834800077

217 Seiten

pEuR02:

E-procurement: Euphorie und Realität

Von Alexander-Philip Nekolar

Veröffentlicht von Springer, 2002

ISBN 3540439803, 9783540439806

163 Seiten

PIIAI08:

Plunkett's Infotech Industry Almanac 2008: InfoTech Industry Market Research, Statistics, Trends and Leading Companies

Von Jack W. Plunkett

Mitwirkende Personen Jack W. Plunkett

Veröffentlicht von Plunkett Research, Ltd., 2008

ISBN 1593921047, 9781593921040

686 Seiten

SaITU04:

Sas/access 9.1 Interface To R/3: User's Guide

Von SAS Institute

Veröffentlicht von SAS Publishing, 2004

ISBN 1590472039, 9781590472033

112 Seiten

SAMMC02:

Strategisches Account Management: Mit CRM den Kundenwert steigern

Von Reinhold Rapp, Kaj Storbacka, Kari Kaario

Veröffentlicht von Gabler Verlag, 2002

ISBN 340911775X, 9783409117753

174 Seiten

SfJeK05

SAP für Java-entwickler: Konzepte, Schnittstellen, Technologien

Von Till Jeske

Veröffentlicht von Springer, 2005

ISBN 3540237879, 9783540237877

278 Seiten

SI02:

Starthilfe Informatik

Von Hans-Jürgen Appelrath, Dietrich Boles, Ingo Wegener, Volker Claus

Veröffentlicht von Vieweg+Teubner Verlag, 2002

ISBN 3519102412, 9783519102410

158 Seiten

SPPSa06:

SAP R/3 ®- Praxishandbuch Projektmanagement: SAP R/3 ® als Werkzeug für professionelles Projektmanagement

Von Holger Gubbels

Veröffentlicht von Vieweg+Teubner Verlag, 2006

ISBN 3834800651, 9783834800657

318 Seiten

SRMSO05:

Supplier Relationship Management: Strategie, Organisation und It des modernen Beschaffungsmanagements

Von Wieland Appelfeller, Wolfgang Buchholz
Veröffentlicht von Gabler Verlag, 2005
ISBN 3409126872, 9783409126878
349 Seiten

☐SvSCS07:
Steuerung von Supply Chains: Strategien- Methoden- Beispiele
herausgegeben von Peter Klaus, Franz Staberhofer, Markus Rothböck
Mitwirkende Personen Peter Klaus
Veröffentlicht von Gabler Verlag, 2007
ISBN 3834905100, 9783834905109
244 Seiten

☐TuKid02:
Telematik- und Kommunikationssysteme in der vernetzten Wirtschaft
Von Günter Müller, Torsten Eymann, Michael Kreutzer
Veröffentlicht von Oldenbourg Wissenschaftsverlag, 2002
ISBN 3486258885, 9783486258882
500 Seiten

☐VPZLA05:
Von Prozessmodellen Zu Lauffähigen Anwendungen
Aris in der Praxis
Von August-Wilhelm Scheer, Wolfram Jost, Karl Wagner
Mitwirkende Personen August-Wilhelm Scheer, Wolfram Jost, Karl Wagner
Veröffentlicht von Springer, 2005
ISBN 3540234578, 9783540234579
247 Seiten

☐VvDwa07:
Vergleich von Data-warehouse-anwendungen anhand der Anforderungen mittelst. Unternehmen
Von Michael Bresik, Bachelor of Science in Computer Science Michael Bresik
Veröffentlicht von GRIN Verlag, 2007
ISBN 3638878619, 9783638878616
96 Seiten

☐WNfSu06:
Wirtschaftsinformatik: Nachschlagewerk für Studium und Praxis
Von Wolfgang Lassmann, Jens Schwarzer
Mitwirkende Personen Wolfgang Lassmann
Veröffentlicht von Gabler Verlag, 2006
ISBN 3409127259, 9783409127257
596 Seiten

☐WPOJd06:
Weblogs, Podcasting & Online-Journalismus: der umfassende Einstieg in WordPress,
journalistisches Know-how anschaulich vermittelt, Suchmaschinenoptimierung, RSS
Von Moritz Sauer

Veröffentlicht von O'Reilly Germany, 2006
ISBN 389721458X, 9783897214583
299 Seiten

☐Wtcs04:
Web-technologien in e-commerce-systemen
Von Claus Strobel
Veröffentlicht von Oldenbourg Wissenschaftsverlag, 2004
ISBN 3486274341, 9783486274349
335 Seiten

☐WTuTi07:
Web 2.0: Trends und Technologien im Kontext der net Economy
Von Tobias Kollmann
Mitwirkende Personen Tobias Kollmann
Veröffentlicht von DUV, 2007
ISBN 3835008366, 9783835008366
278 Seiten

☐WbSSb03:
Web-basierte Systemintegration: So überführen sie bestehende Anwendungssysteme in eine moderne Webarchitektur;[mit online-service zum Buch]
Von Harry Marsh Sneed, Stephan H. Sneed
Veröffentlicht von Vieweg+Teubner Verlag, 2003
ISBN 3528058374, 9783528058371

☐WUWdW02:
Wissensorientierte Unternehmensführung. Wertschöpfung durch Wissen (Taschenbuch)
Von Klaus North
Veröffentlicht von Dr. Th. Gabler Verlag; Auflage: 2., aktualis. u. erw. A., Oktober 2002
ISBN 3409230297, 9783409230292
290 Seiten

☐XuSXi08:
XQuery und SQL/XML in DB2-Datenbanken
Von Alfred Moos
Veröffentlicht von Vieweg+Teubner Verlag, September 2008
ISBN 3834803901, 9783834803900
333 Seiten

☐ZAlmi07:
Zwischenbetriebliche Anwendungsintegration: IT-management in Unternehmensnetzwerken
Von Florian Stadlbauer
Veröffentlicht von Springer, 2007
ISBN 3835007513, 9783835007512
260 Seiten

Internetquellen

🏠AgZ08

Auf gute Zusammenarbeit

<http://www.e3cms.de/index.php?id=1886>

Copyright © 2008 B4Bmedia.net AG

Abgerufen am: Oktober, 2008

🏠BDvIL08

Bereits 100.000 Downloads von IBM Lotus Symphony in der ersten Woche

http://www-05.ibm.com/de/pressroom/presseinfos/2007/09/26_3.html

Copyright © 2008 IBM Deutschland GmbH

Abgerufen am: Oktober, 2008

🏠CSAaa

CAS Software AG arbeitet an ERP-Kopplung für Pia

http://www.computerwoche.de/knowledge_center/crm/1878098/

Autor(en): Frank Niemann, COMPUTERWOCHE-Redakteur.

Erstellt am 6. November 2008

Copyright © IDG BUSINESS MEDIA GMBH München

Abgerufen am: November, 2008

🏠DwRfa07

Die wahren Risiken für assetintensive Unternehmen

Dateiformat: PDF

IBM Configuration Management-Lösungen. White Paper. April 2007

<ftp://ftp.software.ibm.com/software/emea/de/tivoli/TIW10322-DEDE-00.pdf>

Copyright © IBM Corporation 2007

Abgerufen am: Oktober, 2008

🏠EGzQd08

EfA 2006 – Teil 2: Grundsätzliches zur Qualität des Code

<http://www.einfach-fuer-alle.de/artikel/efa-relaunch/2006/teil2/>

Autor: Tomas Caspers (tc)

Creative Commons Public License

Abgerufen am: Oktober, 2008

🏠Erp08

Enterprise resource planning

http://en.wikipedia.org/wiki/Enterprise_Resource_Planning

Diese S. wurde zuletzt am 7 November 2008, um 13:01 Uhr geändert.

Der Text steht unter der GNU-Lizenz für freie Dokumentation.

Wikipedia® ist eine eingetragene Marke der Wikimedia Foundation Inc.

Abgerufen am: Oktober, 2008

🏠FGdFiS01

Formale Grundlagen der Fehlertoleranz in verteilten Systemen

<http://tuprints.ulb.tu-darmstadt.de/162/>

(2001) Dissertation thesis, TU Darmstadt.

Autor: Gärtner, Felix Christoph

Abgerufen am: Oktober, 2008

🏠GoICT08

Glossary of Integrated Circuit Terminology - P

Nach <http://www.icknowledge.com/glossary/p.html>

"Process Integration - the act of integrating a set of process steps to create a process flow."

Copyright © 2000 - 2004 IC Knowledge

Abgerufen am: Oktober, 2008

🏠PUUeW08

Progress-Umfrage: Unternehmen erwarten Wettbewerbsvorteile durch SOA erst auf lange Sicht

Veröffentlicht am 23. April 2008

von Progress Software AG, Bernstrasse 388, 8953 Dietikon-Zürich

www.progress.com/progress_software/worldwide_sites/ch/uber_uns/press/docs/2008/230408.pdf

Abgerufen am: Oktober, 2008

🏠ISICC08

IBM SAP International Competence Center (ISICC) Walldorf

<http://www-05.ibm.com/de/worktogether/customer/isicc.html>

Copyright © 2008 IBM SAP International Competence Center (ISICC) Walldorf

Abgerufen am: Oktober, 2008

🏠KfMW08

15 Konkurrenten für Microsoft Word

<http://software.magnus.de/office-buero/artikel/textverarbeitungen.html>

Veröffentlicht am 19.05.2008 von Tom Rathert

Abgerufen am: Oktober, 2008

🏠Ofdml08

Office document markup languages

http://en.wikipedia.org/wiki/List_of_document_markup_languages

Diese S. wurde zuletzt am 15. Oktober 2008 geändert.

Der Text steht unter der GNU-Lizenz für freie Dokumentation.

Wikipedia® ist eine eingetragene Marke der Wikimedia Foundation Inc.

Abgerufen am: Oktober, 2008

🏠Offic08

Office-Paket

<http://de.wikipedia.org/wiki/Office>

Diese S. wurde zuletzt am 14. Oktober 2008 um 16:14 Uhr geändert.

Der Text steht unter der GNU-Lizenz für freie Dokumentation.

Wikipedia® ist eine eingetragene Marke der Wikimedia Foundation Inc.

Abgerufen am: Oktober, 2008

🏠OOKI8i

OpenOffice-Konkurrenz: IBM Lotus Symphony - Beta 3 der OpenOffice-Source-Suite jetzt auch auf Deutsch

<http://www.netzwelt.de/news/76985-OpenOffice-konkurrenz-ibm-lotus-symphony.html>

Copyright © 2008 by netzwelt.de

Abgerufen am: September, 2008

LoEsp08

List of ERP software packages

http://en.wikipedia.org/wiki/List_of_ERP_software_packages

Diese S. wurde zuletzt am 20. November 2008 geändert.

Der Text steht unter der GNU-Lizenz für freie Dokumentation.

Wikipedia® ist eine eingetragene Marke der Wikimedia Foundation Inc.

Abgerufen am: Oktober, 2008

LNaDP08

Lotus Notes and Domino Product and Roadmap Update

[www.detdominopros.org/hosting/dnp/dnpmeeti.nsf/.../\\$FILE/Detroit%20Notes%20Professionals%20032608.pdf](http://www.detdominopros.org/hosting/dnp/dnpmeeti.nsf/.../$FILE/Detroit%20Notes%20Professionals%20032608.pdf)

Copyright © IBM Corporation 2008. All Rights Reserved.

Abgerufen am: September, 2008

MODhg08

Microsoft: OpenDocument hat gewonnen

<http://www.golem.de/0806/60546.html>

Copyright © 1997 - 2008 Golem.de. Alle Rechte vorbehalten.

Abgerufen am: September, 2008

OOoWS08

OWL Ontology of Web-Services Architecture Concepts

<http://www.w3.org/2004/02/wsa/>

Copyright © 2002-2004 W3C® (MIT, ERCIM, Keio)

Abgerufen am: September, 2008

PDSSG08

Power, D. J. Decision Support Systems Glossary.

DSS Resources, World Wide Web,

<http://DSSResources.COM/glossary/>, 1999.

Bzw. <http://services.eliteral.com/glossary/decision-support-systems-glossary.php>

Abgerufen am: September, 2008

DSodI08

Die Schnittstelle oder das Interface

<http://de.wikipedia.org/wiki/Schnittstelle>

Diese S. wurde zuletzt am 13 November 2008 geändert.

Der Text steht unter der GNU-Lizenz für freie Dokumentation.

Wikipedia® ist eine eingetragene Marke der Wikimedia Foundation Inc.

Abgerufen am: November, 2008

SiwMf07

SAP ist weltweit Marktführer für ERP, CRM und SCM

http://www.sap.com/germany/about/press/archive/press_show.epx?ID=3644

Copyright © SAP AG

Abgerufen am: November, 2008

🏠SGINM08

SAP-GESCHÄFTSBERICHT 2007 INNOVATION NACH MASS

www.sap.com/germany/about/investor/pdf/GB2007_DE.pdf

Copyright © 2008 SAP AG, Dietmar-Hopp-Allee 16, 69190 Walldorf, Deutschland

Abgerufen am: Oktober, 2008

🏠SMISO07

SAP, Microsoft, Infor, Sage und Oracle sind die Top-Five im ERP-Geschäft

<http://www.computerwoche.de/index.cfm?pid=334&pk=596904>

Copyright © IDG BUSINESS MEDIA GMBH München

Abgerufen am: November, 2008

🏠SSWi08

SRH Studienschwerpunkt Wirtschaftsinformatik

<http://web.fh-heidelberg.de/de/fh-heidelberg/4705.html>

Abgerufen am: Oktober, 2008

🏠WSAww04

David Booth et al.: Web-Service Architecture W3C Working Group Note 11 February 2004

<http://www.w3.org/TR/ws-arch/>

Abgerufen am: Oktober, 2008

🏠WRfSO08

Worldwide Revenue von Standalone Open-Source Software Will Grow 26% to Reach \$5.8 Billion by 2011, IDC Research Indicates

<http://www.idc.com/getdoc.jsp?containerId=prUS20711507>

Abgerufen am: Oktober, 2008

Anmerkung:

„Grundsätzlich darf man bei der Informationssuche nie vergessen, dass gerade das Internet eine hohe Anfälligkeit hat für falsche oder manipulierte Informationen.“ [🏠OkDPd08, S.281].

Aber Sauer [🏠WPOJd06, S.134] betont: „Auch wenn in vielen Büchern die Internetrecherche immer noch als unsicher gilt, wächst mit jedem Tag das digitale globale Gedächtnis. Nachrichtenarchive wachsen zu einem immer großartigen Informations-Pool für eine fundierte Recherche heran.“

Abbildungsverzeichnis

Abbildung 1: Vereinfachte Architektur Darstellung	12
Abbildung 2: Architektur Business-Daten nach Office (B2O).....	13
Abbildung 3: Architektur Business-Daten nach Symphony (B2S)	14
Abbildung 4: Architektur Daten von SAP-ERP nach Symphony	15
Abbildung 5: Punkt-zu-Punkt-Integration.....	17
Abbildung 6: Hub-and-Spoke-Integration	17
Abbildung 7: Serviceorientierte Integration	17
Abbildung 8: ERP-Konzept [GWAME04, S.165].....	21
Abbildung 9: Schichten einer Client/Server-Architektur [vgl. WnfSu06, S.149].....	21
Abbildung 10: ERP-Architektur "Zentrale Präsentation"	22
Abbildung 11: ERP-Architektur "Dezentrale Präsentation"	22
Abbildung 12: ERP-Architektur "Dezentrale Verarbeitung mit zentraler Speicherung"	22
Abbildung 13: Allgemeine Struktur eines Business-Objekts [mHTGu03].....	28
Abbildung 14: Typische Synchrone SAP-Kommunikation	28
Abbildung 15: Trends bei Office-Paketen.....	30
Abbildung 16: Wissenstreppe [WUWdW02, S.41].....	37
Abbildung 17: Paradigmenwechsel in der Ausprägung von IT-Systemen [DrWzS08, S.226]	39
Abbildung 18: Szenario Einladung zur Jubiläumsfeier	41
Abbildung 19: Kunden-Adressliste in Symphony-Tabelle.....	42
Abbildung 20: ODT Screenshot (Die beiden Bilder sind aus http://www.clubofrome.de/aktiv/)	43
Abbildung 21: ODT-Dokument im Dateisystem.....	44
Abbildung 22: ODT-Dokument mit Endung .zip.....	44
Abbildung 23: ODT ausgepackt	44
Abbildung 24: Quelltext-Auszug content.xml.....	45
Abbildung 25: Anhand Vorlage (und Daten) neues Dokument erstellen.....	46
Abbildung 26: Symphony API zur Dokumentverarbeitung und ViewPart-Erstellung	47
Abbildung 27: Architektur ODS mit Sym. API.....	49
Abbildung 28: Paketstruktur des Prototyps	50
Abbildung 29: Paket com.ibm.isicc.sap.jco	50
Abbildung 30: Paket com.ibm.isicc.ui.symphony.....	51
Abbildung 31: Ablauf einer Generierung im Prototyp	51
Abbildung 32: Paket com.ibm.isicc.data.....	52
Abbildung 33: Paket com.ibm.isicc.data.pipe	52
Abbildung 34: Lage der Templates im Dateisystem	53
Abbildung 35: Klasse PathDefinition u. a. zur Abfrage der Template-Adresse	53
Abbildung 36: layoutspezifische Trenncodes in Klasse ParseHandler_2contentFile	54
Abbildung 37: Zwischenschichten bzg. Datenfluss	56
Abbildung 38: SOA-Triangle [Dostal, Jeckle, Melzer 2005, Abbildung 2.2]	58
Abbildung 39: Diagramm mit Focus auf den Konfigurations-Ablauf.....	62

Quelltextverzeichnis

Quelltext 1: CSV-Trenncode hardcodiert im Java-Quelltext	54
Quelltext 2: Java-Klasse als Web-Service exportieren.....	59
Quelltext 3: Server zur Veröffentlichung des Service	59
Quelltext 4: WSDL Rfc2Ods	60
Quelltext 5: Auszug der Interface-Klasse Rfc2Ods	61
Quelltext 6: Client der Interface- und Service-Klasse verwendet.....	61

Tabellenverzeichnis

Tabelle 1: Drei Arten der Integration	17
Tabelle 2: Mögliche Verteilung von ERP-Aufgaben zw. Client und Server.....	22
Tabelle 3: Liste von Office-Paketen	88

Anhang

Marktanteile ERP-Software

Marktanteile 2005 (laut Gartner Dataquest) der fünf weltweit größten Anbieter von ERP-Software:

Marktanteile 2005 [SiwMf07]			
#	Anbieter	Umsatz (Millionen \$)	Marktanteil (%)
1	SAP	4726	28,7
2	Oracle	1674	10,2
3	Sage	1221	7,4
4	Microsoft	616	3,7
5	SSA Global Technologies (von Infor Global Solutions übernommen)	464	2,8

Die Experton Group AG hat eine Liste der umsatzstärksten ERP-Anbieter in Deutschland angefertigt. Die größten Anbieter in Deutschland Jahr 2006 sind:

ERP-Marktanteile (Lizenzen und Wartungsgebühren) [SMISO07]		Deutschland	2006
#	Anbieter		Marktanteil (%)
1	SAP		56
2	Microsoft Dynamics		5
3	Infor		4
4	Sage (einschl. bäurer)		4
5	Oracle (einschl. PeopleSoft u. J.D. Edwards)		3
6	SoftM		2

Liste von ERP Software-Systemen

Freie ERP Software und Open-Source ERP Software nach [LoEsp08]:

- Adempiere, ein Java-basiertes ERP-System
- Compiere, ein Java-basiertes ERP-System
- Dolibarr, ein PHP-basiertes ERP-System
- ERP5, ein Python-basiertes ERP-System
- GNU Enterprise
- GRR (software), ein PHP/MySQL-basiertes, freies ERP-system
- JFire
- Kualifoundation
- LedgerSMB
- OFBiz
- OpenOfficeBlueLab
- OpenOfficebravo, ein Java-basiertes ERP-System
- OpenOfficeERP (Tiny ERP)
- OpenOfficetaps (ein Java-basiertes ERP-System)
- OrangeHRM
- Postbooks von XTuple
- SQL-Ledger
- Stoq
- WebERP

Proprietary ERP software nach [LoEsp08]:

- 1C:Enterprise von 1C Company
- 24SevenOffice Start, Premium, Professional and Custom von 24SevenOffice
- abas ERP von ABAS Software
- Accpac von The Sage Group
- Agresso Business World von Unit 4 Agresso
- AMS Advantage von CGI Group (American Management Systems)
- BatchMaster ERP von BatchMaster Software
- Comprehensive Patient Administrator
- Consona Corporation - Intuitive; Made2manage; AXIS; Cimnet; Encompix; DTR
- Epicor Enterprise von Epicor
- ERP Adage (aka Adage) von Infor Global Solutions
- ERP LN (aka Baan) von Infor Global Solutions
- ERP LX (aka BPCS) von Infor Global Solutions
- ERP SL (aka SyteLine) von Infor Global Solutions
- ERP SX.Enterprise (aka SX.Enterprise) von Infor Global Solutions
- ERP VE (aka Visual Enterprise) von Infor Global Solutions
- ERP XA (aka MAPICS) von Infor Global Solutions
- Global Shop Solutions One-System ERP Solutions
- HansaWorld products
- IFS Applications von Industrial and Financial Systems
- JD Edwards EnterpriseOne & JD Edwards World von Oracle
- kVASy4 von SIV.AG

-
- Lawson M3 von Lawson Software
 - Maximo (MRO) von IBM
 - MFG/PRO von QAD Inc
 - Microsoft Dynamics AX (formerly Axapta) von Microsoft
 - Microsoft Dynamics GP (formerly Great Plains) von Microsoft
 - Microsoft Dynamics NAV (formerly Navision) von Microsoft
 - Microsoft Dynamics SL (formerly Solomon) von Microsoft
 - Momentum von CGI Group
 - Movex von Intenia
 - NetERP von NetSuite Inc.
 - OpenOfficeda QX von OpenOfficeda
 - OpenOfficeMFG von xTuple
 - Oracle e-Business Suite von Oracle
 - Paradigm von Consona Corporation
 - PeopleSoft von Oracle
 - Ramco e.Applications von Ramco Systems
 - Ramco Ondemand erp von Ramco Systems
 - MAS 90, MAS 200 and MAS 500 von The Sage Group
 - SAGE ERP X3 von The Sage Group
 - SAP Business One von SAP
 - SAP R/3 von SAP
 - SYSPRO von Syspro
 - SYS-APPS von Exclusive Technologies
 - mySAP von SAP
 - Vantage von Epicor
 - Visual Enterprise von Infor Global Solutions

Liste von Office-Paketen

Tabelle 3: Liste von Office-Paketen

Alle Angaben in dieser Tabelle ohne Gewähr und ohne Anspruch auf Vollständigkeit (Stand: 25. September 2008)

some supported (import ...) formats

Short name	long name	Offline-Version	Online-Ver.	API available	CSV	ODF	Types	MS-HTML	OS
Ability	Ability Office	± 50 €		prob. No	1	0	1	0	
AbiWord	AbiWord		0 €	Prob. yes		1			UNIX, Win., QNX, Mac, BeOS
ajaxWrite	ajaxWrite		0 €	prob. No	0	0	1	0	Web-Browser
Ashampoo	Ashampoo Office	± 60 €		prob. No			1		Web-Browser
Buzzword			0 €	prob. No	0	0		0	Web-Browser
Easy	Easy Office	it depends		prob. No					
ELO Digital	ELO Digital Office	± 300 €		No	0	0	1	0	Win.
Financ. Of.	Lex. Financial Of.	± 400 €		prob. No					Win.
FlySuite		± 30 €	0 €	prob. No			1		Win., Web-Browser
GNOME	GNOME Office	0 €		No		1	1		Linux (Gnome)
KOffice		0 €		No	1	1	1	1	Linux (KDE)
Mobile	Mobile Office					1			Symbian OS
MS Office	Microsoft Office ...	± 400 €	± 400 €		1	1	1	1	Win., Web-Brow.
NeoOffice		0 €				1			Mac OS X
OpenOffice Off.	Sun OpenOffice	0 €			1	1	1	1	Win., Linux, Solaris, Mac OS
Papyrus	Papyrus Office	± 100 €		No	1	0	1		Win., Mac OS
Scribus	Scribus	0 €				1			Linux, Unix, MacOS, OS/2, eCS, Win.
Siag Office	Siag Office	0 €		Yes	1	0	0	0	Unix, MacOS X, OpenOfficeBSD
SoftMaker	SoftMaker Office	± 70 €			1	1	1	1	Win., Linux, FreeBSD, Pocket, Zaurus
SOT	SOT Office 2002	0 €		prob. No	0	0	0	0	Linux, (Windows)
StarOffice	Sun StarOffice	± 90 €		prob. No	1	1	1	1	
Symphony	IBM Lotus Symphony	0 €		Yes	1	1	1	1	Windows, Linux
Text & Tab.	Google Text & Tabellen		0 €	prob. No	1	1	1	1	
TextEdit	Apples TextEdit					1			Mac OS
USB-Office	bhv USB-Office	± 15 €		prob. No					Win.
WordPerfect	Corel WordPerfect® O.	± 200 €			1	1	1	1	Win.
Zoho	Zoho Sheet/Writer/ ...	± 0 €	± 0 €	Yes	1	1	1	1	Web-Browser

Liste der Office-Dokument-Beschreibungs-Sprachen

- Microsoft Office 2003 XML Format
- Office Open XML (OOXML) - OpenOffice standard format für Office-Dokumente.
- OpenDocument (ODF) - OpenOffice standard format für Office-Dokumente.
- OpenOffice.org XML – Vorgänger des OpenDocument Formates.
- ReportML – Report-Formatierungs-Sprache. (derzeit kein teil von Office Open XML)
- Uniform Office Format (UOF) - OpenOffice format für Office-Dokumente, ist abgestimmt mit OpenDocument.

[Odfml08]

Verwendete Tools - Empfehlungen

Eclipse als Java-Entwicklungsplattform

Der mit dieser Arbeit erstellte Prototyp nutzt als Entwicklungsplattform. Vielen ist Eclipse nur als IDE bekannt, obwohl Eclipse eine vom Betriebssystem unabhängige Plattform ist, auf der neben Entwicklungsumgebungen auch Client-Applikationen betrieben werden können.

Mehr zu eclipse auf <http://www.eclipse.org>

FreeCommander – Alternativer Dateimanager unter Windows.

Bedanken und empfehlen möchte ich mich auch für den kostenlosen Dateimanager FreeCommander bzw. für die hervorragende Arbeit des Entwicklers Marek Jasinski. Er hat zwei nebeneinander angeordnete Ordnerfenster, besitzt eine integrierte DOS-Eingabezeile. Funktionen lassen sich bequem durch Hotkeys starten, Dateinamen und Pfade lassen sich in die Zwischenablage kopieren, Dateien können wie im Windows-Explorer oder wie im Norton Commander selektiert werden, das Programm nutzerspezifisch anpassen, lässt sich auf eine CD oder USB-Stick kopieren.

Fullscreen-Plug-in für Eclipse

Für mehr Platz bei der Arbeit mit Eclipse als IDE sorgt das Eclipse Full Screen Plugins. Eclipse bietet eine Vielzahl von Sichten an, die dem Entwickler beim Programmieren helfen, mit dem Eclipse Full Screen Plug-in kann zwischendurch die Taskleiste sowie die Fenstertitelzeile ausgeblendet werden. Es steht kostenlos zur Verfügung. Bezugsquelle: <http://code.google.com/p/eclipse-fullscreen/>

Das Plug-in ist derzeit leider nur in einer Version für Windows zur Verfügung!

Goto File- Plug-in für Eclipse

Java-Entwickler die mit Eclipse arbeiten können viel Ihrer Zeit sparen, die sie bei dem Navigieren in verschiedenen Ansichten benötigt haben um eine Datei zu finden, wenn Sie Goto File verwenden. Den Dateinamen oder zumindest Teile des Dateinamens der gesuchten Datei weiss man. Und dieses Wissen reicht auch schon um mit der Schnellsuche von Goto File fündig zu werden. Mausbedienung ist nicht erforderlich. Es steht kostenlos zur Verfügung.

Bezugsquelle: <http://www.muermann.org/gotofile/>

MouseFeed- Plug-in für Eclipse

MouseFeed ist eine Shortcut-Erinnerung für Eclipse-Nutzer. Es soll helfen die Keyboard Shortcuts schnell zu erlernen. Klickt der User auf ein GUI-Element, erscheint ein Pop-up-Fenster, welches an den entsprechenden Shortcut erinnert. Es steht kostenlos zur Verfügung. Bezugsquelle: <http://www.mousefeed.com/>

Relo - Relationship based Exploration

Relo hilft Entwickler Code besser zu verstehen. Relo erstellt automatisch Diagramme auf Grundlage von UML. Relo- Diagramme beziehen sich nur auf einen kleinen überschaubaren Teil des Codes und enthalten keine irrelevanten Informationen, so dass ein Entwickler, sich auf wichtige Code Beziehungen konzentrieren kann. Relo kann auch zur Navigation im Code verwendet werden. Es steht kostenlos zur Verfügung. Die Bezugsadresse ist <http://relo.csail.mit.edu>

Relo wurde bereits in folgenden wissenschaftlichen Publikationen erwähnt:

- Vineet Sinha, David Karger, Rob Miller, "Relo: Helping Users Manage Context during Interactive Exploratory Visualization of Large Codebases", Visual Languages and Human-Centric Computing (VL/HCC 2006). Sep. 4-8, 2006, Brighton, United Kingdom. 8 pages.

[<http://relo.csail.mit.edu/documentation/relo-vlhcc06.pdf>]

- Vineet Sinha, David Karger, Rob Miller, "Relo: Helping Users Manage Context during Interactive Exploratory Visualization of Large Codebases", OOPSLA'05 Eclipse Technology eXchange (ETX) Workshop. Oct. 16-20, 2005, San Diego, California, USA. 5 pages.

[<http://relo.csail.mit.edu/documentation/relo-etx05.pdf>]

QuickRex-Plug-in für Eclipse

QuickREx ermöglicht über einen weiteren View in Eclipse, das Testen von regulären Ausdrücken. Dabei werden die Regex-Patterns on the fly ausgeführt, so dass jede Änderung sofort das entsprechende Ergebnis zeigt. Bezugsquelle: <http://www.bastian-bergerhoff.com/eclipse/features/> bzw. <http://www.bastian-bergerhoff.com/eclipse/features/web/QuickREx/toc.html>

yEd - Java Graph Editor

Mithilfe des Java Grapheneditor können Zeichnungen generiert werden und automatische Layouts für unterschiedliche Diagramme und Netzwerke erstellt werden. Der Editor wurde mit Swing-Komponenten erstellt, dadurch kann er auf jeder Plattform laufen, die eine passende Runtime Environment zur Verfügung stellt. yEd verwendet die yFiles Java-Klassenbibliothek, die der Analyse, der automatischen Layouterstellung und Visualisierung von Diagrammen, Netzwerken und Graphen dient.