

Fachhochschule Köln
University of Applied Sciences Cologne
Abteilung Gummersbach
Fachbereich Informatik

Diplomarbeit
(Drei-Monats-Arbeit)

zur Erlangung
des Diplomgrades
Diplom-Informatiker (FH)
in der Fachrichtung Wirtschaftsinformatik

Integration CRM- und ERP-Systemen
am Beispiel von Clarify eFrontOffice und SAP

von Erika Bieltz

Inhaltsverzeichnis

1. Einführung.....	7
1.1. Motivation.....	7
1.2. Beispiel Auftragsabwicklung.....	8
1.2.1. Problemstellung.....	9
1.2.2. Zielsetzung.....	10
1.3. Aufgabenstellung und Vorgehensweise.....	11
I. Teil	
2. CRM – Customer Relationship Management	13
2.1. Definition.....	13
2.2. Aufgaben des CRM.....	15
2.2.1. Individualisierung.....	15
2.2.2. Profitabilität.....	16
2.3. Komponenten von CRM-Systemen	17
2.3.1. Operatives CRM.....	18
2.3.2. Analytisches CRM.....	18
2.3.3. Kommunikatives CRM.....	18
2.4. Funktionen eines CRM-Systems.....	19
2.4.1. Vertrieb.....	19
2.4.2. Service.....	22
2.4.3. Marketing.....	23
2.5. Clarify eFrontOffice.....	24
2.5.1. Module	24
3. ERP - Enterprise Resource Planning.....	28
3.1. Definition.....	28
3.2. Aufgaben von ERP	28
3.2.1. Abbildung der Wertschöpfungskette	28
3.3. Komponenten von ERP-Systemen.....	29

3.3.1.	<i>Finanz- und Rechnungswesen</i>	30
3.3.2.	<i>Personalwesen</i>	31
3.3.3.	<i>Logistik und Produktion</i>	31
3.4.	SAP R/3	32
3.4.1.	<i>Module</i>	33

II. Teil

4.	Integration	36
4.1.	Trennung von Anwendungsschwerpunkten.....	36
4.2.	Gemeinsame Geschäftsobjekte	37
4.3.	Systemübergreifende Geschäftsprozesse	37
4.4.	Integrationstechniken	40
4.4.1.	<i>Integrationsebenen</i>	40
4.4.2.	<i>Kommunikationsmodell</i>	42
4.5.	Grundlegender Aufbau einer Integrationsplattform.....	42
5.	Integration von Clarify und SAP	46
5.1.	Clarify	46
5.1.1.	<i>Architektur</i>	46
5.1.2.	<i>Integrationstechnologien</i>	48
5.1.3.	<i>Auswahl der Integrationstechnologie</i>	51
5.2.	SAP	52
5.2.1.	<i>Architektur</i>	52
5.2.2.	<i>Integrationstechnologien</i>	54
5.2.3.	<i>Auswahl der Integrationstechnologie</i>	56
5.3.	Technische Realisierung eines generischen Schnittstellenmoduls	57
5.3.1.	<i>Anforderungen</i>	57
5.3.2.	<i>Lösungskonzept</i>	58
5.3.3.	<i>Realisierung</i>	62

III. Teil

6. Fachliches Implementierungsbeispiel	78
6.1. Teil 1 – Durchführung in Clarify	79
6.1.1. Identifikation der BAPIs in SAP	79
6.1.2. BAPI-Aufruf von Oracle	85
6.1.3. Anpassung der Benutzerschnittstelle in Clarify.....	89
6.2. Teil 2 - Durchführung in SAP.....	92
6.2.1. Initiierung der Statusänderung aus SAP.....	93
6.2.2. Durchführung der Statusänderung in Oracle.....	95
7. Zusammenfassung.....	97
8. Literaturverzeichnis.....	99
9. Internetquellen	101
10. Glossar	104

Abbildungsverzeichnis

Abbildung 1: Prozessablauf ohne Integration.....	9
Abbildung 2: Prozessablauf mit Integration	11
Abbildung 3: Rentabilitätsauswirkungen durch Kundenloyalität.....	14
Abbildung 4: Gewinnanteil einzelner Kundengruppen	16
Abbildung 5: Komponenten einer CRM-Lösung	17
Abbildung 6: Darstellung des Ablaufs von Beispiel-Geschäftsprozesse.....	38
Abbildung 7: Ebenen der Integration.....	41
Abbildung 8: Aufbau einer Integrationsplattform	45
Abbildung 9: 2-tier-Architektur von Clarify.....	47
Abbildung 10: 3-tier-Architektur von Clarify.....	48
Abbildung 11: Die Client-Server-Architektur des SAP-Systems.....	52
Abbildung 12: Architektur-Typen von SAP R/3	53
Abbildung 13: Generisches Schnittstellenmodul - Funktionsweise	59
Abbildung 14: Embedded-SQL Beispiel	69
Abbildung 15: Funktionsablauf bei einer Client-Verbindung	74
Abbildung 16: Funktionsablauf bei einer Server-Verbindung.....	75
Abbildung 17: Geschäftsprozess „Auftragsverarbeitung“	78
Abbildung 18: SAP BAPI Browser	80
Abbildung 19: Clarify Form „Sales Order“	89
Abbildung 20: Clarify Form „Select Product“	90
Abbildung 21: Auftrags-Tabellen im Clarify-Datenmodell	95

Tabellenverzeichnis

Tabelle 1: Komponenten des SAPORAD.....	63
Tabelle 2: Parameter der Funktion BAPI_SALESORDER_CREATEFROMDAT1.....	81
Tabelle 3: Obligatorische Parameter – BAPI_SALESORDER_CREATEFROMDAT1.	82
Tabelle 4: Wrapper für BAPIs.....	83
Tabelle 5: RFC-Datentypen.....	85
Tabelle 6: Parameter der Procedure callFunctionModul.....	86
Tabelle 7: Parameter der Procedures createOrder_SAP und getConditions_SAP.....	86
Tabelle 8: Datenbank-Änderungen beim Umsetzen des Auftragsstatus.....	96

Abkürzungsverzeichnis

ABAP	Advanced Business Application Programming
ACD	Automatic Call Distribution
ALE	Application Link Enabling
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BAPI	Business Application Programming Interface
COM	Component Object Model
CRM	Customer Relationship Management
CTI	Computer Telephony Integration
EAI	Enterprise Application Integration
EDI	Elektronik Data Interchange
ERP	Enterprise Resource Planning
GUI	Graphical User Interface
IDOC	Intermediate Document
OLAP	Online Analytical Processing
OLE	Object Link Enabling
POMS	Persistent Object Management System
RDBMS	Relational Database Management System
RFC	Remote Function Call
SCM	Supply Chain Management
SDK	Software Development Kit
SQL	Structured Query Language

1. Einführung

1.1. Motivation

In der heutigen Marktwirtschaft mit dem intensiven Wettbewerb liegt ein zunehmend wichtiger Schlüssel für den unternehmerischen Erfolg in der Qualität der Kundenbeziehungen. Diese wiederum sind stark abhängig von der Fähigkeit des Unternehmens, individuelle Kundenprobleme zu lösen, anstatt nur Produkte oder Dienstleistungen anzubieten. Der Fokus muss somit weg von einer Produktorientierung hin zu einer langfristigen Kundenorientierung gehen.

Das Internet und andere virtuelle Kommunikationsportale ermöglichen auch den kleinsten Firmen die Teilnahme an globalen Märkten. Doch damit steigen zugleich die Erwartungen der Kunden: Ein umfassender Kundenservice rund um die Uhr ist heute eine selbstverständliche Forderung. Die Konkurrenz ist bloß einen Mausklick entfernt. Kundenservice darf keine lästige Pflicht sein. Er ist eine strategische Notwendigkeit.

„Customer Relationship Management“ (CRM) fasst sämtliche Aktivitäten zusammen, deren Ziel eine verbesserte und effiziente Kundenorientierung ist. Es umfasst die vielschichtigen Beziehungen eines Unternehmens mit seinen Kunden. Doch Kunden wollen nicht nur betreut werden, sie erwarten in allen Belangen bestmögliche Leistungen zum bestmöglichen Preis. Ein Unternehmen muss mit maßgeschneiderten Angeboten auf die individuellen Bedürfnisse seiner Kunden reagieren. Der "Tante-Emma-Laden" gleich um die Ecke kannte die Kunden und deren individuellen Gewohnheiten persönlich. Der Laden hat seine Kunden an sich gebunden, weil "Tante Emma" alles über ihre Vorlieben wusste. Auch ein Unternehmen verfügt über ein solches Gedächtnis. Mit ERP-Systemen wird die Wertschöpfungskette eines Unternehmens unterstützt und abgebildet. Sie verzeichnen jede Bestellung, jede Stornierung, jede Transaktion zwischen dem Kunden und dem Unternehmen und bieten somit einen riesigen Wissenspool. Um diese großen Datenmengen für die personalisierte Kommunikation mit den Kunden zu nutzen und durchgängige Prozesse bei der Kundeninteraktion zu gewährleisten, ist es notwendig, die ERP-Systeme eng mit den CRM-Anwendungen zu verzahnen.

Wie wichtig die Durchgängigkeit von Prozessen zwischen Frontend- und Backend für die Bearbeitung von Kundenprozessen ist, soll nachfolgend am Beispiel des Geschäftsprozesses für die „Auftragsabwicklung“ verdeutlicht werden.

1.2. Beispiel Auftragsabwicklung

In folgendem Szenario wird die Auftragsabwicklung in einem Call-Center dargestellt. Dem CRM-System wird hierbei die Rolle der Auftragserfassung und dem ERP-System die der Auftragsdurchführung zugeteilt. Anhand der einzelnen Schritte soll der Prozessablauf aufgezeigt werden:

1. Der Gesprächspartner eines Call-Center-Mitarbeiters am anderen Ende der Telefonleitung fragt, ob das Unternehmen ihm verschiedene Produkte zu einem bestimmten Termin liefern kann.
2. Der Call-Center-Mitarbeiter stellt fest, dass sein CRM-System den Anrufenden noch nicht als Kunden führt. Um ein Angebot zu erstellen muss er die Daten des Kunden im CRM-System erfassen; sie werden später auch von dem beteiligten ERP-System benötigt.
3. Der Call-Center-Mitarbeiter sucht die gewünschten Produkte in einem elektronischen Katalog und erstellt ein Angebot: Dazu gibt er die erforderlichen Produktdaten sowie vom Kunden gewünschte Stückzahlen und Liefertermine in das CRM-System ein.
4. Wiederum müssen diese Daten in die ERP-Systeme gelangen, die für die weitere Auftragsabwicklung zuständig sind.
5. Mit Hilfe der ERP-Systeme wird geprüft, ob die Produkte verfügbar sind und welche Preiskonditionen für einen Neukunden gelten. Aus einem der ERP-Systeme wird ersichtlich, dass ein Teil der gewünschten Produkte nicht zum angegebenen Termin geliefert werden kann.
6. Dies muss der Call-Center-Mitarbeiter dem Kunden mitteilen und mit ihm eine entsprechende Modifizierung des Auftrags besprechen.
7. Nachdem der Kunde mit der Änderung des Auftrags einverstanden ist kann der Call-Center-Mitarbeiter die Änderung ins CRM-System eingeben.
8. Auch im ERP-System müssen die Auftragsdaten geändert werden um eine richtige Durchführung zu gewährleisten.

1.2.1. Problemstellung

Bei der Durchführung des Prozesses für die Auftragsabwicklung muss der Call-Center-Mitarbeiter gleichzeitig mit zwei Systemen interagieren. Dies erfordert einen dauernden, ineffizienten Wechsel zwischen zwei Benutzeroberflächen.

Sowohl die Kundendaten, also auch die angeforderten Produktdaten müssen in beide Systeme eingepflegt werden. Das CRM-System benötigt die Daten, um Historien anzulegen und jederzeit Informationen zum Kunden bereitstellen zu können. Vom ERP-System werden die Daten für die Auftragsdurchführung benötigt. Informationen über Verfügbarkeit der gewünschten Produkte und Preiskonditionen muss der Call-Center-Mitarbeiter aus dem ERP-System entnehmen. Da in dem geschilderten Beispielsfall das gewünschte Produkt nicht zum angegebenen Termin lieferbar ist, muss der Call-Center-Mitarbeiter Alternativprodukte anbieten. Diese Informationen sind im ERP-System enthalten. Bei Bestätigung des Auftrages, muss die Änderung wieder in beide Systeme übernommen werden.

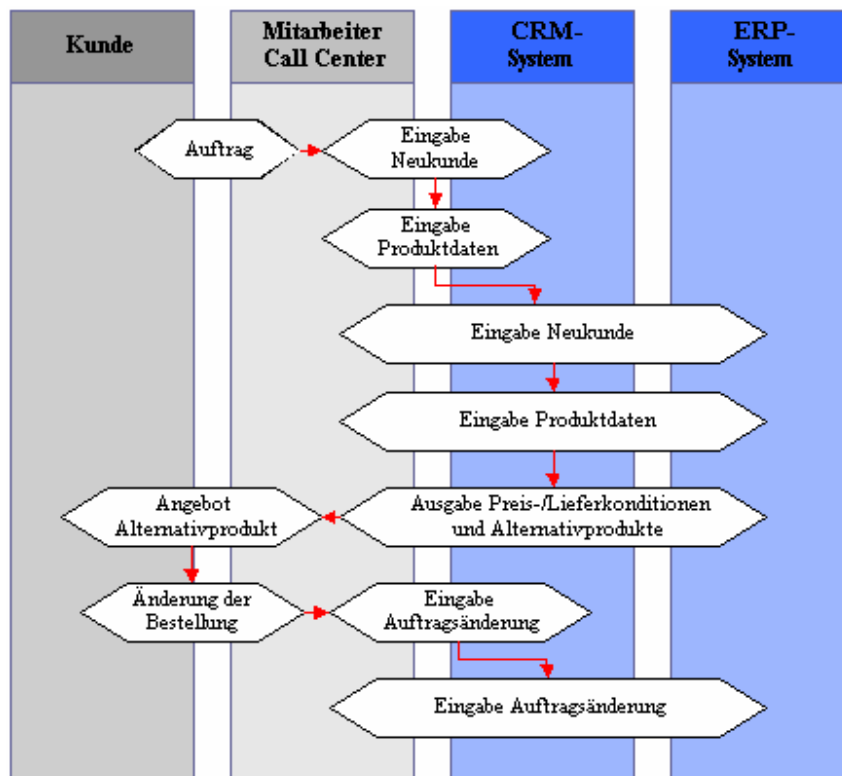


Abbildung 1: Prozessablauf ohne Integration

In Abbildung 1 wird der beschriebene Ablauf der Auftragsabwicklung ohne Integration der beteiligten System dargestellt.

Durch die mehrfache Dateneingabe entsteht nicht nur ein zusätzlicher Zeit- und Kostenaufwand, sondern auch ein Problem bei der Konsistenz der Daten. Durch die manuelle Eingabe kommt es zu Zeitverzögerungen und erhöhten Fehlerquoten, was zu inkonsistenten Datenbeständen führt.

Sind CRM und ERP-System nicht integriert, verlängert sich der Prozess der Auftragsabwicklung erheblich, was wiederum die Qualität des Services beeinträchtigt.

1.2.2. Zielsetzung

Das Ziel dieser Arbeit ist eine Integration des CRM-Systems Clarify eFrontOffice und des ERP-Systems SAP durchzuführen, um die aufgezeigten Probleme zu beheben. Bei einer integrierten Lösung erfolgt der größte Teil der Auftragsbearbeitung automatisch. Neue oder ggf. geänderte Kundendaten, sowie Produktdaten werden automatisch in die beteiligten ERP-Systeme übertragen. Binnen Sekunden sieht der Call-Center-Mitarbeiter das Ergebnis seiner Abfragen auf seinem Bildschirm. Dazu gehört auch, dass eines der Systeme automatisch ein Alternativ-Produkt anbietet, da der ursprünglich gewünschte Artikel nicht zum angegebenen Lieferdatum verfügbar ist. Erzeugt der Call-Center-Mitarbeiter nach letzten Rücksprachen mit dem Kunden von seinem CRM-Bildschirm aus eine Bestellung, wird wiederum automatisch eine Bestellung in dem ERP-System erzeugt.

Im Ergebnis stimmen die für die gesamte Auftragsabwicklung relevanten Daten im CRM- und ERP-System überein. Ohne Integration ist die Datenkonsistenz, wie sie für einen fehlerfreien Ablauf des beschriebenen Prozesses benötigt wird, nicht zu erreichen. Die Grafik aus Abbildung 2 zeigt einen optimierten Prozessablauf durch die Integration von CRM-System und ERP-System.

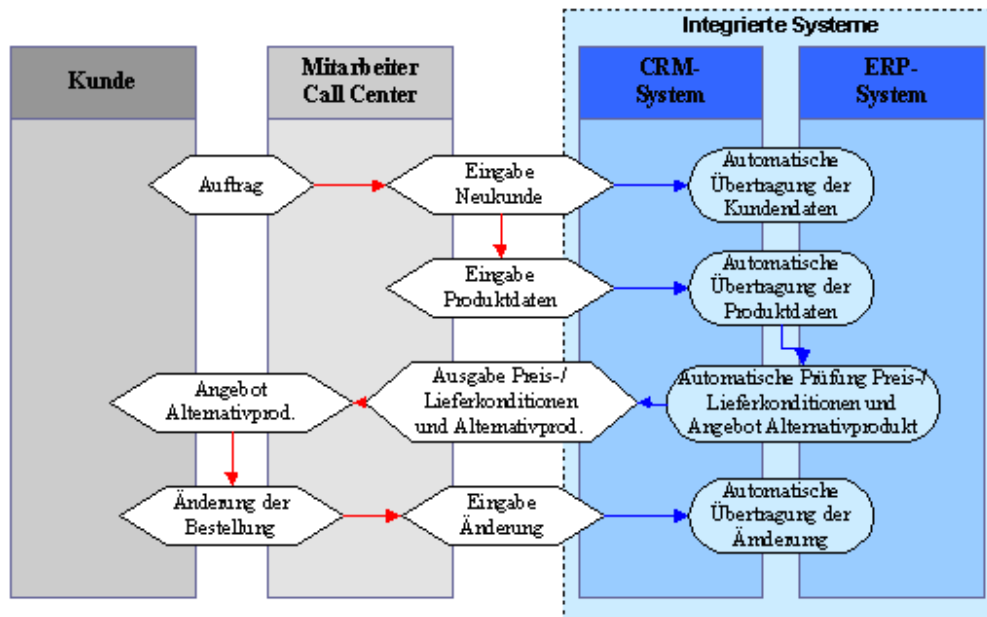


Abbildung 2: Prozessablauf mit Integration

Die Problemstellung bei dem Beispiel „Auftragsabwicklung“ (Kapitel 1.2.1) zeigt die Folgen einer fehlenden Integration: Eingabefehler bei mehrfacher Erfassung, verlängerte Prozesslaufzeiten und somit eine sinkende Servicequalität. Das Ziel (Kapitel 1.2.2) ist also, mehrfache Eingaben zu vermeiden, die Prozessdurchlaufzeiten zu optimieren und somit eine bessere Qualität des Services zu erreichen.

Nur eine Integration von CRM-Lösung und ERP-System kann eine permanente Aktualität aller relevanten Daten, die für eine Ausrichtung aller Aktivitäten des Unternehmens auf den Kunden zwingend notwendig ist, gewährleisten.

1.3. Aufgabenstellung und Vorgehensweise

Die Aufgabe dieser Diplomarbeit ist eine generische Lösung für die Integration von Customer Relationship Management und Enterprise Resource Management zu realisieren. Zur Durchführung der Integration werden die Kernprozesse der beiden Systeme identifiziert, um festzustellen, welche Aufgaben von welchem der beiden Systeme übernommen werden und inwieweit sich diese überschneiden. Die zu integrierenden Geschäftsprozesse und die Möglichkeiten der Integration werden dargestellt. Um die Reali-

sierbarkeit zu verdeutlichen, wird eine prototypische Implementierung der Integration eines Beispielprozesses am Beispiel der Systeme Clarify und SAP durchgeführt.

Die Durchführung der Diplomarbeit erfolgt in drei Hauptteilen. Im Nachfolgenden wird das Vorgehen näher beschrieben:

1. Teil – „CRM und ERP und die Systeme Clarify und SAP“

Im ersten Teil wird auf die konkreten Ausprägungen der beiden Anwendungsbereichen „Customer Relationship Management“ und „Enterprise Resource Management“ eingegangen. Beispielhaft werden hierzu die zwei weitverbreiteten Systeme Clarify und SAP vorgestellt. Für diese Systeme wird im 3. Teil eine Integration prototypisch realisiert.

2. Teil – „Integrationskonzept“

Im zweiten Teil wird ein Konzept für eine Integration von CRM und ERP erarbeitet. Das Konzept wird in zwei Teile aufgliedert: den inhaltlichen Teil und den technischen Teil. Im inhaltlichen Integrationskonzept wird festgestellt, welche Geschäftsprozesse systemübergreifend sind und wie die Durchgängigkeit der Prozesse gewährleistet werden kann. Der technische Teil beleuchtet die verschiedenen Integrationstechniken und erarbeitet und dabei eine generische Lösung, die auf alle Prozesse angewendet werden kann.

3. Teil – „Beispiel / Machbarkeitsbeleg“

Im dritten Teil wird durch Abbildung eines realen systemübergreifenden Kernprozesses eines Unternehmens die technische Integration am Beispiel der Systeme Clarify und SAP prototypisch realisiert.

2. CRM – Customer Relationship Management

2.1. Definition

„CRM ist eine kundenorientierte Unternehmens-Philosophie, die mit Hilfe moderner Informations- und Kommunikationssysteme versucht, auf lange Sicht profitable Kundenbeziehungen durch ganzheitliche Marketing-, Vertriebs- und Servicekonzepte aufzubauen.“ [WIL00]

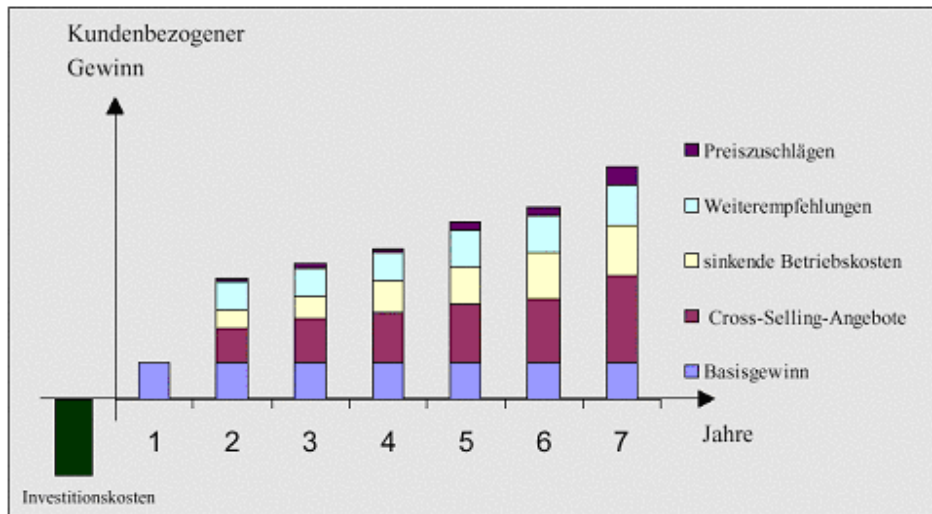
Das ist eine von vielen Definitionen für CRM. Obwohl die Bedeutung von CRM schon seit Jahren erkannt ist, gibt es keine allgemein gültige, einheitliche Definition von „Customer Relationship Management“. Generell versteht man darunter die Bearbeitung der Beziehung eines Unternehmens zu seinen Kunden. Hinter CRM muss eine kundenorientierte Unternehmenspolitik stehen. Die gesamte Unternehmensorganisation muss auf die Kundenbeziehungen ausgerichtet werden.

CRM lässt sich am besten anhand seiner drei Schlüsselbegriffe charakterisieren:

- Customer: Der Aufbau einer engen Beziehung zu bestehenden und potentiellen neuen Kunden steht im Fokus aller Unternehmensaktivität
- Relationship: Die individuelle Behandlung von bestehenden und potentiellen Kunden ist der Schlüssel zum zukünftigen Erfolg
- Management: Die Fähigkeit, alle Interaktionen mit bestehenden und potentiellen Kunden über alle organisatorischen Grenzen hinweg kontinuierlich zu koordinieren, muss entwickelt werden.

Customer Relationship Management umfasst die gesamte Interaktion eines Unternehmens mit bestehenden und zukünftigen Kunden während des gesamten Kaufentscheidungsprozesses und Besitzzyklus. Es muss eine kontinuierliche, im Idealfall lebenslange, Beziehung zum Kunden aufgebaut werden.

Das zentrale Ziel des CRM ist die Verbesserung der Kundenloyalität und das Verhindern der Abwanderung der Kunden. Welche Folgen die Verbesserung der Kundenloyalität auf die Rentabilität eines Unternehmens hat, wird in Abbildung 3 dargestellt.



Quelle: Becker, Jochen: Der Strategietrend im Marketing, Vahlen 2000, S. 123

Abbildung 3: Rentabilitätsauswirkungen durch Kundenloyalität

Im Zeitverlauf sind loyale Kunden profitabler. Zu dem eigentlichen Basisgewinn, den ein Unternehmen an einem Kunden verdient, kommen eine Reihe weiterer Gewinnpotenziale hinzu, wenn das Unternehmen den Kunden besser kennt. Eines davon ist die Möglichkeit von Cross-Selling-Angeboten. Einem Kunden der bisher bestimmte Leistungen bzw. Produkte eines Unternehmens bezogen hat, können gezielt auch andere Produkte und Leistungen, welche speziell an die Bedürfnisse des Kunden angepasst sind, angeboten werden. Ein weiterer zusätzlicher Erfolgsbeitrag loyaler Kunden liegt in den geringeren Betriebskosten insgesamt. Bei Kunden die das Unternehmen kennt und mit denen es öfter zu tun hat wird der Ablauf der Auftragsabwicklung mit der Zeit immer reibungsloser und reduziert dadurch den Aufwand des Unternehmen. Auch der Erfolgsbeitrag loyaler Kunden durch Weiterempfehlungen (Mund-zu-Mund-Propaganda) ist nicht zu unterschätzen. Ein zufriedener Kunde ist die kostengünstigste Werbung. Letztlich bringt auch die höhere Preisbereitschaft der loyalen Kunden weitere Gewinnmöglichkeiten. Kunden, die dem

Unternehmen schon lange treu sind und seiner Leistungsqualität vertrauen, sind auch bereit, einen höheren Preis zu zahlen.

2.2. Aufgaben des CRM

2.2.1. Individualisierung

Die Individualisierung der Kundenbeziehungen ist eine zentrale Forderung des CRM-Konzeptes. Dabei müssen nicht nur die Produkte und Dienstleistungen eines Unternehmens auf die Bedürfnisse des Kunden zugeschnitten sein, sondern auch die Ansprache an den Kunden. Um die Bedürfnisse seiner Kunden aber befriedigen zu können, muss das Unternehmen deren spezifische Wünsche kennen. Die Identifizierung der Kundenwünsche ist also eine zentrale Aufgabe des CRM. Das Wissen über den Kunden, sein Kaufverhalten und seine Bedürfnisse muss konsequent ausgebaut werden. Diese Identifizierung der Wünsche ist ein kontinuierlicher Prozess. Bei jeder Interaktion zwischen dem Anbieter und dem Kunden erfährt das Unternehmen etwas mehr über den Kunden. Alle diese Informationen müssen gesammelt und gebündelt werden. Diese Bündelung ist notwendig, da es vielfältige Möglichkeiten und Anlässe zur Kommunikation des Kunden mit dem Unternehmen gibt. Jede Interaktion mit dem Kunden muss dokumentiert werden, um allen relevanten Abteilungen zur Verfügung zu stehen. Nur so kann der jeweilige Mitarbeiter den Kunden adäquat bedienen. Die Beziehung zum Kunden wird nur eine echte 1:1-Beziehung, wenn das Unternehmen als Einheit auftritt.

Die Identifizierung der Wünsche muss konsequent zur Individualisierung und Anpassung des Marketing-Mixes genutzt werden. Das Unternehmen muss bereit sein, aus der Beziehung zu lernen und sein Angebot immer besser an die jeweiligen Kunden anzupassen.

Wenn das Unternehmen seine Kunden genau kennt, kann es seinen Kunden zum richtigen Zeitpunkt das richtige Angebot unterbreiten. Die Trefferquoten im Verkaufsabschluss erhöhen sich, und Streuverluste können systematisch reduziert werden. Das gewonnene Kundenwissen hilft nicht nur, bestehende Kunden an das Unternehmen zu binden, sondern hilft dem Unternehmen gleichzeitig, seine gesamten Aktivitäten besser auf die Bedürfnisse des Marktes auszurichten und so auch mehr neue Kunden zu akquirieren.

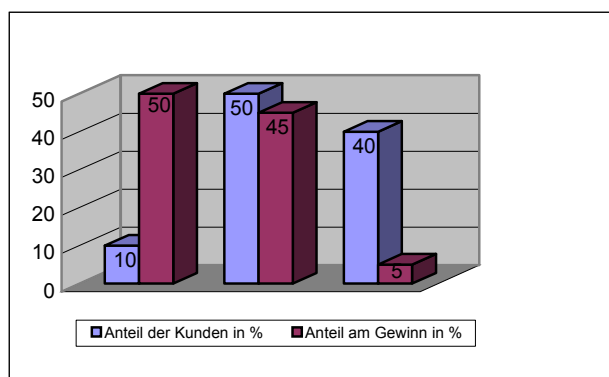
Durch gezielte Kundenforschung können Unternehmensaktivitäten auf die Wünsche und Bedürfnisse der Zielgruppe ausgerichtet werden.

2.2.2. Profitabilität

Eine weitere zentrale Zielsetzung des CRM liegt in der Fokussierung auf Kunden, die dem Unternehmen langfristig profitabel erscheinen. „Im Rahmen des CRM wird der Erhöhung des Marktanteils eine eher geringe Bedeutung beigemessen, dagegen tritt der Share of wallet eines Kunden zunehmend in den Vordergrund – also der Anteil der Kaufkraft des Kunden, die beim Unternehmen verbleibt.“[WIL00]

Die Identifizierung des Kundenwertes erlaubt es dem Unternehmen, seine Ressourcen auf die profitablen Kunden zu konzentrieren. Die Kunden können dann nicht nur entsprechend ihrer jeweiligen Bedürfnisse, sondern auch entsprechend ihres Wertes behandelt werden. Das Unternehmen kann so dem richtigen Kunden das oben erwähnte richtige Angebot unterbreiten.

zeigt, dass die meisten Unternehmen den größten Teil ihres Gewinns mit nur wenigen Kunden erwirtschaften. Da liegt es nahe, seine Anstrengungen auf die besten Kunden zu konzentrieren.



Quelle: [WIL00]

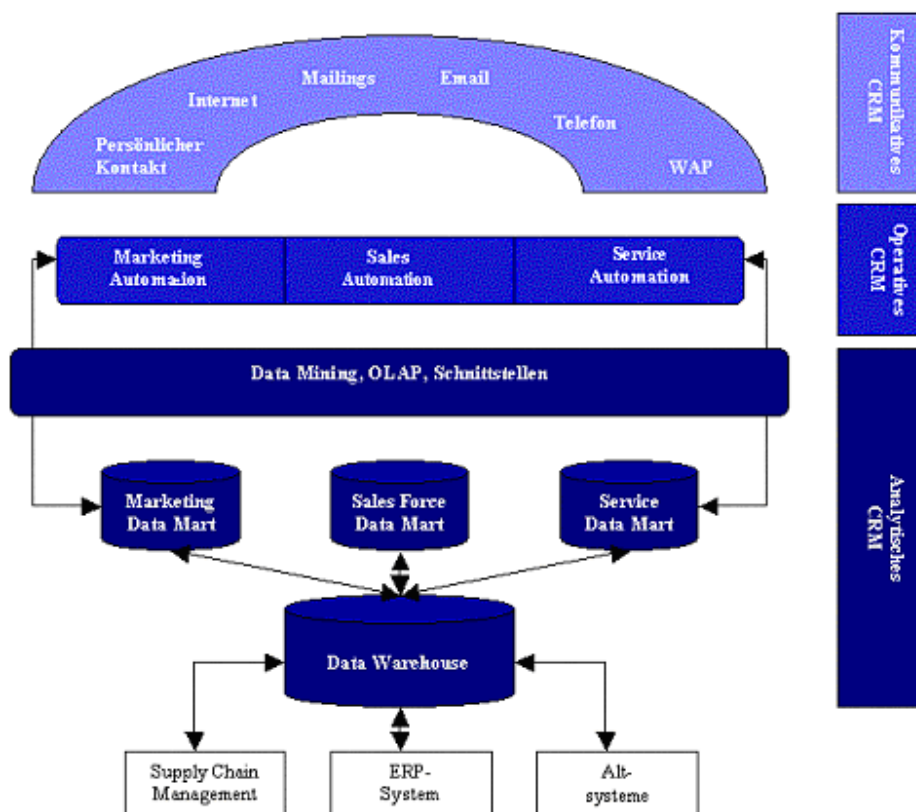
Abbildung 4: Gewinnanteil einzelner Kundengruppen

Der Kundenwert orientiert sich dabei aber nicht an dem zur Zeit getätigten Umsatz, sondern vor allem auch am zukünftigen Potenzial des Kunden. Je besser das Unternehmen den Kunden kennt, desto besser kann es auch dieses zukünftige Potenzial einschätzen. Beim CRM steht also der *Customer Lifetime Value* im Vordergrund. Der gesamte Kundenlebenszyklus wird mit einbezogen. CRM versucht nicht den kurzfristigen Umsatz je Kunde, sondern den Customer Lifetime Value zu erhöhen.

2.3. Komponenten von CRM-Systemen

Um die oben erläuterten Ziele zu erreichen und CRM im Unternehmen umsetzen zu können, sind umfassende Softwareprogramme notwendig. Diese Software bezeichnet man als CRM-Systeme. Mit diesen Software-Lösungen werden im Prinzip drei Aufgabenstellungen verfolgt: „Synchronisation und operative Unterstützung“ des Front Offices, des weiteren die „Zusammenführung und Auswertung heterogener Datenbestände“ und schließlich die „Einbindung diverser Kontaktkanäle“ [RAP00]. Zur Umsetzung dieser Aufgabenstellungen werden drei zentrale Aufgabenbereiche unterschieden, die sich gegenseitig nicht ersetzen, sondern vielmehr in engen Austauschbeziehungen zueinander stehen:

- Operatives CRM
- Analytisches CRM
- Kommunikatives CRM



Quelle: [WIL00]

Abbildung 5: Komponenten einer CRM-Lösung

2.3.1. Operatives CRM

Das operative CRM umfasst alle Anwendungen des Front-Office-Bereichs. Unter dem Begriff Front-Office werden die Bereiche eines Unternehmens zusammengefasst, die im direkten Kontakt mit dem Kunden stehen. „Im Front-Office-Bereich werden Lösungen zur Marketing-, Sales- und Service-Automation integriert, die den Dialog zwischen Kunden und Unternehmen weitestgehend unterstützen“ [WIL00]. Hierunter fallen sowohl die Versorgung mit sämtlichen Kundeninformationen, als auch Funktionen zur Rationalisierung einzelner Routineaufgaben. Um den Kunden gegenüber verlässliche Aussagen machen zu können, muss das operative CRM an vorhandene Back-Office-Lösungen (ERP- und Supply Chain Management Systeme aber auch bestehende Altsysteme) angebunden werden.

2.3.2. Analytisches CRM

Das analytische CRM beschäftigt sich mit der gezielten Analyse der im operativen CRM-Bereich generierten Daten. Die optimale Zusammenarbeit von Front- und Back-Office ist nur dann gewährleistet, wenn alle Unternehmensbereiche auf die gleichen Daten zugreifen können. Entscheidungsrelevante Daten aus mehreren Quellen müssen in einer einheitlichen Systemumgebung integriert sein, in einem sogenannten Data Warehouse. Das Data Warehouse bildet das Herzstück des CRM, da es alle Informationen über einen Kunden beinhaltet. Es stellt die Grundlage für das analytische CRM dar, das mit Hilfe von Analysetools wie OLAP oder Data Mining Auswertungen zu diesen Informationen erstellt. Die Ergebnisse der Analysen werden entweder in Data Marts (dezentrale, auf bestimmte Unternehmensbereiche fokussierte Datenpools) oder in das Datenbanksystem übertragen und dienen als Grundlage für eine „ganzheitliche Steuerung der Kundenkontakte über den gesamten Kundenlebenszyklus hinweg“ [WIL00].

2.3.3. Kommunikatives CRM

Das kommunikative CRM umfasst die gesamte Steuerung und Unterstützung, sowie die Synchronisation aller Kommunikationskanäle zum Kunden (Telephonie, Internet, Email, Mailings, Außendienst, etc.). Ziel ist es, eine möglichst direkte Interaktion zwischen

Kunden und Unternehmen zu ermöglichen, durch die Einbindung aller Kommunikationskanäle, über die der Kunde in Kontakt zu Unternehmen treten kann.

Als zentrale Kommunikationsschnittstelle zwischen Kunde und Unternehmen hat sich das Call Center entwickelt. Hier wird der größte Bereich der telefonischen Kommunikation zwischen dem Unternehmen und den Kunden abgewickelt. Das Call Center ist eine zeitgemäße Organisation zur Kundenbetreuung und –gewinnung. Die Zufriedenheit der Kunden mit der Gesamtleistung eines Unternehmens wird oft in Verbindung mit der Leistungsfähigkeit des Call-Centers gebracht, wobei die Reaktionsfähigkeit auf Anfragen und Prozesse hierbei eine wichtige Rolle spielt.

Zu einem weiteren interessanten und wichtigen Kommunikationskanal hat sich das Internet entwickelt. Der Kunde ist hierbei nicht mehr an Öffnungszeiten gebunden und kann die Leistungen des Unternehmens rund um die Uhr in Anspruch nehmen. Eine Studie der US-amerikanischen Yankee Group hat ergeben, dass demnächst jeder dritte Kundenkontakt über das Internet erfolgen wird [WIL00].

2.4. Funktionen eines CRM-Systems

Die Funktionen eines CRM-Systems können je nach Software-Anbieter sehr unterschiedlich sein. Die Anzahl und Ausprägung der verschiedenen Komponenten variiert zum Teil stark. Bei den meisten Software-Anbietern kann man verschiedene Module aus einem umfangreichen Funktionsangebot wählen. Die Lösungen werden bereits speziell für einzelne Branchen angeboten, da jede Branche unterschiedliche Anforderungen hat.

Folgenden Ausführungen geben einen Überblick über die wichtigsten, branchenübergreifenden Funktionen für die kundennahen Unternehmensbereiche Vertrieb, Service und Marketing. Dabei dürfen die einzelnen Bereiche nicht isoliert gesehen werden. Erst die Integration und der gemeinsame Zugriff auf eine Kundendatenbank gewährleistet die erfolgreiche Umsetzung von CRM und die Unterstützung bei einer ganzheitlichen Kundenorientierung.

2.4.1. Vertrieb

Hierunter fallen alle Funktionalitäten zur Erleichterung, Unterstützung und Automatisierung der Vertriebsaufgaben.

Um den Kunden im operativen Geschäft optimal beraten zu können, muss der Vertriebsmitarbeiter über den Kunden und die Produkte umfassend informiert sein. Informationen über den Kunden fangen mit den vollständigen Stammdaten zu einem Kunden an. Aber auch die Angabe von Kontaktpersonen im Unternehmen, sowie die Kontakt- und Auftragshistorie jedes Kunden müssen ebenfalls sofort ersichtlich sein. Dazu ist ein umfassendes **Kontaktmanagement** erforderlich. Zu jedem Stammsatz in der Kunden- sowie der Interessentendatenbank müssen dazu sämtliche Kontakte lückenlos erfasst und gepflegt werden. So entsteht im Laufe der Zeit eine umfassende Kontakthistorie. Anhand dieser kann ein Kunde entsprechend seinem Profil und Wert bedient werden. Auch die eigentliche Kontaktaufnahme wird - oft durch eine Einbindung in die Programmumgebungen von Microsoft Outlook oder Lotus Notes - vereinfacht und ein sofortiger Versand von Fax- oder e-Mail-Nachrichten ermöglicht.

Neben detaillierten Kundeninformationen sind auch umfassende Produktinformationen für eine gute Beratung unerlässlich. Eine **Marketing-Enzyklopädie** stellt dem Vertriebsmitarbeiter deshalb alle wichtigen Informationen über die Produkte in elektronischer Form zur Verfügung und liefert ihm wichtige Argumentationshilfen. Auch zusätzliche Informationen über Wettbewerber und deren Produkte können dem Verkäufer beim Dialog mit dem Kunden helfen. Informationen über häufige Kundenfragen und – einwände, und die besten Antworten beziehungsweise Gegenargumentationen helfen den Mitarbeitern zusätzlich bei ihren Verkaufsgesprächen.

Über eine **Verfügbarkeitsüberprüfung** kann der Mitarbeiter außerdem ersehen, ob die Waren verfügbar sind und dem Kunden so Auskunft über Lieferzeiten erteilen. Ein umfassendes **Angebots- und Auftragsmanagement** unterstützt den Vertriebsmitarbeiter bei der Erstellung von Angeboten, der Eingabe von Aufträgen und deren weiteren Verfolgung. Per Mausklick kann der Vertriebsmitarbeiter verschiedene Produktkonfigurationen auswählen und zusammenstellen. Die Preiskalkulation kann ebenfalls automatisiert werden. Die erstellten Angebote können systematisch verfolgt werden. Auch die Eingabe von Aufträgen wird vereinfacht. So wird zum Beispiel bei der Eingabe automatisch die Bonität des Kunden überprüft und individuelle Konditionen beachtet. Auch die weitere Verfolgung der Aufträge (**Order Tracking**) wird automatisiert. Der Vertriebsmitarbeiter

kann jederzeit den Status eines Auftrages abfragen und dem Kunden entsprechende Auskünfte erteilen.

Ein **Termin-Managementsystem** erinnert den Vertriebsmitarbeiter automatisch an ausstehende Aufgaben und Kundenkontakte und weist ihn beispielsweise auf einen vereinbarten Gesprächstermin hin. Für die Außendienstmitarbeiter sind solche Termin-Managementsysteme besonders wichtig. Auch alle anderen Funktionen des Vertriebs-Innendienstes stehen ihm übers Internet aktuell zur Verfügung. So kann er bei einem Verkaufsgespräch sofort nachschauen, ob ein bestimmtes Produkt verfügbar ist, er kann sofort Aufträge eingeben oder anfordern, dass ein Servicetechniker zum Kunden kommen soll. Zusätzlich stehen dem Außendienst spezielle Funktionen zur Unterstützung seiner Reiseaktivitäten und Rationalisierung von Routineaufgaben zur Verfügung, wie Routenplanung, Spesenerfassung, Reisekostenabrechnung.

Für den telefonischen Vertrieb, zum Beispiel über Call-Center, steht den Mitarbeitern zusätzlich eine **CTI-Technologie** (Computer Telephony Integration) zur Verfügung. Dabei wird die Telekommunikationsanlage mit den Computersystemen verbunden. Anrufe können dann automatisch an den nächsten freien Agenten weitergeleitet werden (ACD Automatic Call Distribution). Dem Agenten werden bei eingehenden Anrufen automatisch die dazugehörigen Kundeninformationen angezeigt. Umgekehrt können Kunden vom System auf Knopfdruck des Agenten automatisch angewählt werden (Power Dialing). Zusammengestellte Anruflisten können computergestützt abgearbeitet werden. Im Vertriebsbereich sind außerdem vielfältige Analysen möglich, welche die Mitarbeiter in ihrer Verkaufstätigkeit unterstützen. Als Beispiele hierfür sind die **Sales Cycle Analyse** zur Ermittlung des Wiederbeschaffungszeitpunkts oder die **Lost Order Analyse** zu nennen. Diese befasst sich mit der Frage, warum es bei einem bestimmten Kunden zu keinem Verkaufsabschluss gekommen ist. Auch Cross-Selling-Potentiale können systematisch aufgedeckt werden und ermöglichen eine maßgeschneiderte Kundenbetreuung. Eine Analyse der Vertriebstätigkeiten kann auch dem Management wichtige Informationen liefern. Umsatzentwicklungen lassen sich nach Vertriebsgebieten, dort wieder nach Produkten und weiter nach Kunden mehrdimensional auswerten. Es lassen sich Umsatztrends erkennen und Umsatzentwicklungen zuverlässiger prognostizieren. Die Effektivität der Vertriebsmitarbeiter und der Vertriebsaktivitäten lässt sich besser kontrollieren.

2.4.2. Service

Der Service hat sich in den letzten Jahren zu einem Differenzierungsfaktor im Wettbewerb entwickelt und ist ein starkes Werkzeug in der Kundenbindung. Die Zufriedenheit der Kunden wird heute geprägt durch eine verbesserte Betreuung und mehr fachliche Kompetenz bei der Beratung. Nur wer auch mit dem Kundenservice zufrieden ist, wird dem Unternehmen treu bleiben und weitere Produkte dort kaufen. Die Software-Funktionen in diesem Bereich sollen die Tätigkeit der Servicemitarbeiter erleichtern und steuern.

Ein umfassendes **Beschwerden-Management** ist für die Schaffung von Kundenloyalität besonders wichtig. Mit Beschwerden drückt ein Kunde aus, dass er mit der Leistung des Unternehmens unzufrieden ist. Das Unternehmen muss dieses als eine Chance zur Wiedergutmachung sehen. In solchen kritischen Phasen muss ein Kunde besonders zuvorkommend behandelt werden. Eine adäquate Reaktion auf die Beschwerde kann den Kunden wieder ans Unternehmen binden und in der Regel sogar stärker als vorher. Eingehenden Beschwerden müssen systematisch erfasst und bearbeitet werden. Sämtliche Aktionen zur Bearbeitung beziehungsweise zur Beseitigung des Problems müssen dokumentiert werden, damit bei wiederholenden Beschwerden darauf zurückgegriffen werden kann. Mit Hilfe von **Eskalationsmechanismen** kann gewährleistet werden, dass ein Problem innerhalb eines bestimmten Zeitraumes erfolgreich beseitigt wird und nicht in Vergessenheit gerät.

Ein Kunde wendet sich nicht nur bei Beschwerden, sondern auch beim Wunsch nach Beratung an das Unternehmen. Bei technischen Fragen kann der Service-Mitarbeiter durch einen **Help Desk** unterstützt werden. „Ein Help Desk ist ein wissensbasiertes Datenbankmanagementsystem, das für die Aufnahme von Störungsfällen und für die Beantwortung von Benutzeranfragen zuständig ist.“ [WIL00] Bei bekannten Problemen werden vorgefertigte Lösungsskripte angeboten, damit die Mitarbeiter die Fragen des Kunden direkt beantworten können. Handelt es sich um unbekannte oder komplexe Probleme, wird ein **Workflow** in Gang gesetzt, um die Anfrage an einen geeigneten Spezialisten weiterzuleiten.

Auch die Analyse der Beschwerden und Anfragen ist besonders wichtig. Das Unternehmen muss daraus lernen und die Qualität der Produkte und Dienstleistungen mit Hilfe

dieser Informationen ständig verbessern. So kann die Anzahl der Servicefälle, sowie die Bearbeitungszeit der übrig bleibenden Fälle reduziert werden.

2.4.3. Marketing

Hierunter fallen alle Funktionalitäten, die das Marketing im Unternehmen unterstützen und gezielt auf die Kunden ausrichten sollen. Dem **Kampagnen-Management** kommt dabei eine große Bedeutung zu. Es ermöglicht die Selektion von Zielgruppen, die Durchführung von Kommunikationsmaßnahmen und die kontinuierliche Verfolgung der Ergebnisse dieser Maßnahmen. Die Kunden können nach unterschiedlichen Merkmalen mehrdimensional segmentiert und klassifiziert werden. Die Marketingmitarbeiter erhalten durch die Auswertung der umfangreichen Kundeninformationen ein genaues Bild ihrer Zielgruppe und können ihre Maßnahmen effektiver auf deren Bedürfnisse ausrichten. Aus der umfangreichen Kundendatei können Kunden mit einem bestimmten Profil selektiert werden, um dann mit gezielten, auf diese Kundenprofile zugeschnittenen, Mailing-Aktionen bearbeitet zu werden. Die selektierten Adressen können leicht exportiert und an externe Marketing-Dienstleister übergeben oder mit Serienbrief-Hauptdokumenten der eigenen Textverarbeitungsprogramme verknüpft werden. Die Response-Erfassung wird durch die Software vereinfacht und sofort dem jeweiligen Kunden zugeordnet. So kann der Erfolg von Marketingmaßnahmen kontrolliert und gemessen werden. Die Marketingmitarbeiter können aus den Response-Analysen Erkenntnisse für ihre weiteren Maßnahmen gewinnen und so ihrer Arbeit immer effizienter und wirkungsvoller gestalten. Die Marketingmitarbeiter sind in der Regel für die Erstellung, Pflege und Aktualisierung der **Marketing-Enzyklopädie**, welche die Vertriebsmitarbeiter bei ihrer täglichen Arbeit unterstützt, verantwortlich. Besonders die Produktmanager stellen den Vertriebsmitarbeitern hier umfangreiches Informations- und Argumentationsmaterial zur Verfügung.

Im folgenden Kapitel wird das CRM-Standard-System Clarify eFrontOffice vorgestellt und es wird verdeutlicht wie die verschiedenen CRM-Aufgaben damit abgedeckt werden.

2.5. Clarify eFrontOffice

Clarify eFrontOffice, von Amdocs, ist ein voll integriertes CRM- und e-Business-Angebot, das die Kunden in die Lage versetzt, mit den Unternehmen zu ihren Bedingungen über das Medium ihrer Wahl in Verbindung zu treten. Es gewährleistet einen hohen Standard der Kundenbetreuung und sorgt so für einen reibungslosen Ablauf des Service- und Geschäftsbetriebes. Es ermöglicht den Unternehmen, Verkaufs-, Marketing- und Serviceinitiativen im eBusiness durch einen personalisierten Informationsaustausch zwischen Kunden und Geschäftspartnern über das Internet einzusetzen. Die Funktionalitäten von Clarify eFrontOffice werden auf einzelne Module aufgeteilt, die im nachfolgend vorgestellt werden.

2.5.1. Module

Clarify eFrontOffice bietet eine Reihe von Applikationen, die bei der Kundenbetreuung Unterstützung leisten. Entsprechend den verschiedenen Kommunikationskanälen, mit denen der Kunde in Kontakt zu Unternehmen treten kann, werden folgende Module angeboten, die darauf abgestimmt sind:

ClearCallCenter

ClearCallCenter ist eine Anwendung, die den Zugang zu Informationen und das Organisieren von Informationen während einer Interaktion mit einem Kunden bzw. potentiellen Kunden vereinfacht. ClearCallCenter unterstützt typische Arbeitsvorgänge eines Call Centers, um bestehende Kunden zu betreuen bzw. neue Kunden zu gewinnen. Hierbei handelt es sich beispielsweise um:

- Aufrufen aller relevanten Informationen bezüglich des Kunden (Stammdaten, Kontakt-Historien)
- Verwalten und Entgegennahme von Bestellungen
- Abrufen von Informationen über Produkte und Preise
- Bereitstellen von Lösungen zu Problemen
- Initiieren von Kampagnen und Einsicht in bestehende Kampagnen

ClearSales

Mit der Anwendung ClearSales kann ein kompletter Verkaufsprozess, vom Erstkontakt bis zum abgeschlossenen Vertrag, abgewickelt werden. Hierzu gehört die Verwaltung und Weiterverfolgung von potentiellen Geschäften, die Erzeugung von Angeboten und in letzter Instanz die Erstellung der Rahmenbedingungen für den Abschluss der Verträge. ClearSales ermöglicht auch die Verwaltung von Informationen über potentielle Kunden und Konkurrenzunternehmen sowie die Unterstützung der Marketingabteilung bei der Beurteilung der Marktsituation und der eigenen Marktposition, um so einen geeigneten Absatzplan zu erstellen.

ClearSupport

ClearSupport unterstützt die Kommunikation zwischen Unternehmen und Kunde und stellt ein komplettes Kundenanfragen- / Kundenproblem-Management-System zur Verfügung. Es ermöglicht die Bearbeitung eingehender Fragen und Probleme bezüglich Produkten, Produkterweiterungen und Serviceleistungen. Anfragen können verwaltet, bearbeitet, weitergeleitet und nachverfolgt werden, so dass eine schnelle Abwicklung gewährleistet ist.

ClearLogistics

ClearLogistics regelt das Angebot von Serviceleistungen vor Ort beim Kunden. Der Einsatz der Mitarbeitern kann durch Einsicht in die Terminpläne koordiniert und jedem Auftrag kann die beste, freie Ressource zugeordnet werden.

ClearLogistics unterstützt auch bei der Verwaltung und Durchführung von Änderungsaufträgen und Gewährleistungsanforderungen, wie z.B. Gutschriften, Umtäusche, Ersatzteillieferungen, etc. Reparaturprozesse können automatisiert werden durch Weiterleitung an zuständige Mitarbeiter und Benachrichtigung bei Überfälligkeit.

ClearContracts

ClearContracts bietet ein komplettes Vertrags-Management-System und ermöglicht die Erstellung, Verwaltung, Überwachung und Überarbeitung von Verträgen und Servicevereinbarungen.

ClearContract bietet die Möglichkeit, Vertriebs- und Verkaufsexperten schnell und flexibel Informationen über Serviceleistungen und Kalkulationen bis hin zu komplexen Vertragsentwürfen und Zertifikationen anzubieten. Durch die Automatisierung dieser Abläufe können in kürzester Zeit Angebote erstellt und Vertragsabschlüsse gefördert werden.

ClearQuality

ClearQuality ist ein Qualitätsmanagementsystem und unterstützt das Bearbeiten von Produktfehlern und Produkterweiterungen sowie das Verwalten des Informationsflusses, der dabei entsteht. ClearQuality wird primär im Bereich der Produktentwicklung sowie Qualitätssicherung eingesetzt.

Im Bereich Qualitätssicherung unterstützt ClearQuality das Aufzeichnen und das Protokollieren von Problemen, die z.B. durch Testläufe gefunden werden. Mit ClearQuality ist es möglich, die protokollierten Probleme weiterzuleiten, die dann an anderer Stelle bearbeitet werden. Die durch ClearSupport bereitgestellten Lösungen sind in diesem Workflow integriert und können zur Problembearbeitung herangezogen werden.

ClearHelpDesk

ClearHelpDesk ist eine integrierte Lösung, die verschiedene Möglichkeiten der Kontaktaufnahme unterstützt und den Mitarbeiter eines Unternehmens bei der Lösung von Kundenproblemen Hilfe leistet. Hierfür steht eine Wissensdatenbank zur Verfügung, mit der Probleme identifiziert und schnell gelöst werden können. Es besteht auch die Möglichkeit, Probleme an andere Mitarbeiter weiterzuleiten und nachzuverfolgen. Zusätzlich können Trend-Analysen gefahren werden, um häufige Probleme zu identifizieren und ein Reporting Tools gibt Auskunft über die Effizienz und Leistungsfähigkeit des Help-Desks.

Produkt Manager

Mit Hilfe des Produkt Managers können im Clarify-System Produkte und Serviceleistungen konfiguriert werden. Auf diese kann von den verschiedenen Clarify-Modulen während des Verkaufsprozesses zugegriffen werden.

Es können Stücklisten verwaltet werden und Produkte können zu Klassen oder Katalogen zusammengefasst werden. Der Produkt Manager bietet auch die Möglichkeit, Preislisten zu erstellen.

Policies and Customers

Policies and Customers dient der Benutzerverwaltung und bietet in dieser Eigenschaft die Möglichkeit, allgemeine Benutzerinformationen zu pflegen und den Benutzern Rollen im System zuzuordnen. Darüber hinaus können Geschäftsregeln definiert werden mit denen die Abläufe verschiedener Geschäftsprozesse festgelegt werden.

3. ERP - Enterprise Resource Planning

3.1. Definition

Unter Enterprise Resource Planning (ERP) versteht man betriebswirtschaftliche Standardsoftware für die integrierte Informationsverarbeitung, mit der unternehmensinterne Daten für Routineaufgaben in den verschiedenen Abteilungen der Unternehmen verarbeitet werden können.

ERP-Systeme sind Back-Office-Anwendungen, die eine effektive Planung und Steuerung aller Ressourcen eines Unternehmens gewährleisten sollen. Betriebswirtschaftliche Anwendungssysteme erfassen die Mengen und Werte des Güter-Leistungsflusses und des resultierenden Geldflusses. ERP-Systeme sind hierbei die operativen Systeme für die Geschäftsabwicklung.

Die Merkmale von ERP lassen sich wie folgt definieren:

- Koordination der Wertschöpfungskette eines Unternehmens über eine einheitliche, gemeinsame Datenbasis
- Überwindung der Grenzen zwischen den verschiedenen Abteilungen eines Unternehmens wie Produktion, Warenwirtschaft, Logistik, Vertrieb, Marketing, Personal, etc
- Möglichkeit zur unternehmensübergreifenden Integration von Marktpartnern (SCM, E-Business)

3.2. Aufgaben von ERP

3.2.1. *Abbildung der Wertschöpfungskette*

Durch den immer stärker werdenden Wettbewerb müssen die Unternehmen die Interaktion mit ihren Kunden verstärken. Um dem Kunden Produkte nach seinen Bedürfnissen schnell zu Verfügung stellen zu können, müssen die Unternehmen deshalb effizienter mit den Kunden und Lieferanten kommunizieren. Um diese Verbesserung in den Lieferzeiten und die Verkürzung in den Produktionszeiten zu erreichen, benötigen die Unternehmen ein effizientes Planungs- und Kontrollsystem. Es erfordert die Integration aller Prozesse innerhalb der Wertschöpfungskette um eine Synchronisation und Planung der Geschäfts-

prozesse zu ermöglichen. Ein ERP-System ist in diesem Sinn ein strategisches Werkzeug, welches einem Unternehmen ermöglicht, isolierte Funktionen in die Geschäftsprozesskette zu integrieren.

Es ist somit die Hauptaufgabe von ERP, die Verwaltung und Abbildung der Wertschöpfungskette zu gewährleisten und Geschäftsprozesse aufeinander abzustimmen und zu automatisieren, um die betriebliche Effizienz zu steigern. Durch die zahlreichen Insellösungen sind die Unternehmen mit den Problemen inkompatibler Informationssysteme und uneinheitlicher Betriebspraktiken konfrontiert. Die mit ERP verbundene Vision ist, alle Informationen in einem Unternehmen über ein einziges, unternehmensweites System zusammenzufassen und einen nahtlosen Informationsfluss und den Austausch von Informationen zwischen den verschiedenen Abteilungen und Funktionen in einem Unternehmen zu ermöglichen. Dadurch wird ein einheitlicher, globaler Überblick über das Unternehmen gewährleistet. Dies ist die Voraussetzung für eine bessere Kontrolle des gesamten Unternehmens durch das Management einer effizienteren Entscheidungsfindung auf Grundlage aller Informationen und nicht nur einzelner Fragmente.

3.3. Komponenten von ERP-Systemen

Die ERP-Systeme sind je nach Anbieter verschieden aufgebaut und versuchen meistens alle Funktionsbereiche eines Unternehmens von der Beschaffung über Lagerhaltung bis hin zur Auftragsabwicklung abzudecken. Die einzelnen Funktionen sind in Modulen zusammengefasst. Die Module können einzeln von Unternehmen bezogen werden oder als integrierte Lösung. Auch bei ERP-Systemen gibt es branchenbezogene Lösungen, die auf die spezielle Anforderung der jeweiligen Branche zugeschnitten sind. Je nach Branche gibt es verschiedene Zusammenstellungen der Komponenten. Im Folgenden werden die wichtigsten Funktionsbereiche, die von ERP-Systemen abgedeckt werden, vorgestellt:

- Finanz- und Rechnungswesen
- Personalwesen
- Logistik und Produktion

3.3.1. *Finanz- und Rechnungswesen*

In einer sich schnell ändernden Geschäftswelt sind transparente und kontrollierbare Geschäftsprozesse unverzichtbare Voraussetzungen für den Erfolg des Unternehmens. Das wichtigste Steuerungs- und Kontrollinstrument eines Unternehmens ist das Rechnungswesen. Nicht nur die gegenwärtigen finanziellen Aktivitäten können damit laufend überwacht, sondern auch die Vergangenheitsdaten können analysiert werden, um zukünftig Geschäftsprozesse zu optimieren. Mit Hilfe des Rechnungswesens kann das komplette Finanzmanagement bewältigt werden und zu einer gesteigerten Rentabilität beitragen. Es leistet sowohl Hilfestellung bei Prognosen als auch bei der strategischen Planung.

Das Buchhaltungssystem erlaubt die Zusammenarbeit von Unternehmenseinheiten mit Shared Services und Geschäftspartnern zur Abwicklung von Finanz- und Zahlungsprozessen, die durch Ein- und Verkaufsprozesse ausgelöst werden. Mit Hilfe der **Kreditoren- und Debitorenbuchhaltung** werden einfache, normale Transaktionen automatisch abgewickelt, so dass der Benutzer sich auf Abweichungen konzentrieren kann. Es werden sowohl Kunden- als auch Lieferantenrechnungen verarbeitet, basierend auf Rechnungsdaten, die mit den integrierten Bereichen Beschaffung und Auftragsverarbeitung abgeglichen sind. Die Zahlungsfunktion des Systems erfasst sämtliche Rechnungen im Unternehmen. Vorschlagslisten für die Bezahlung von Lieferantenrechnungen werden erstellt; ebenso werden Kundenrechnungen auf Fälligkeit überwacht. Zusätzlich liefert das System umfangreiches statistisches Datenmaterial über die Zahlungsmuster der Kunden und Kreditrahmenberechnung und -überwachung.

Ein **Berichtsgenerator** dient der mengen- und wertmäßigen Erfassung von rechnungswesensrelevanten Geschäftsvorfällen für ein konsistentes, abgestimmtes und revisionsfähiges externes Berichtswesen und zur Information der Geschäftsführung sowie als Datenbasis für analytische Anwendungen.

Mit Hilfe von **Analysen** werden wertschöpfende Potenziale im Unternehmen identifiziert und Prognosen erstellt, mit denen eine effiziente Geschäftsplanung und funktionsübergreifende Optimierung der wirtschaftlichen Leistungsfähigkeit sowie die optimale Nutzung von Sachanlagen und immateriellen Vermögenswerten unterstützt werden.

3.3.2. *Personalwesen*

Die gesamte Dynamik der Personalwirtschaft hat sich verändert. Personalabteilungen werden mit neuen Aufgaben konfrontiert. Die Attraktivität des Unternehmens muss gesteigert werden, um für Mitarbeiterbindung und Motivation sorgen. So können die persönlichen Ziele der Mitarbeiter mit der Unternehmensstrategie in Einklang gebracht werden. Das Personalwesen leistet somit einen wesentlichen Beitrag zum Unternehmenserfolg.

Die **Personalbeschaffung** leistet Hilfe bei der strategischen Planung von personellen Ressourcen und unterstützt beim Einstellungsprozess. Durch die Erfassung der Personalanforderungen im gesamten Unternehmen nach definierten Kriterien wird die Qualität der Anforderungsspezifikationen vereinheitlicht und gesichert. Die **Personalentwicklung** verwaltet sämtliche Informationen über Beurteilungsgespräche und die Entwicklungsplanung. Es können Entwicklungspläne erstellt werden mit Maßnahmen zur Mitarbeiterentwicklung. Das kann etwa Jobrotation, Weiterbildung und Mentorenbetreuung beinhalten, sowie die Verantwortlichkeitsdefinitionen und Termine der jeweiligen Maßnahmen. Schlüsselpositionen in einem Unternehmen erfordern ständige Planung im Hinblick auf potentielle Ersatzkräfte und Reserven - um die Stabilität zu erhöhen und die negativen Auswirkungen von Kündigungen und Pensionierungen zu reduzieren.

Mit der **Personalweiterbildung** können Kurse verwaltet und geplant werden. Sie basiert auf Weiterbildungsanforderungen, die vom Unternehmen festgelegt wurden.

Weitere Funktionen des Personalwesens sind die **Personalzeitverwaltung**, für ein effizientes Erfassen, Verwalten und Analysieren von Arbeitszeiten, die **Personalabrechnung**, zur Unterstützung der Lohn- und Gehaltsabrechnungen und die **Reisekostenabrechnung**, für die Verwaltung aller mit Geschäftsreisen verbundenen Kosten.

Durch das ständige Erfassen und Abbildung der Qualifikationen, Erfahrungen, Kompetenzbereiche und absolvierter Trainings seiner Mitarbeiter kann die Gesamtkompetenz eines Unternehmens transparent gemacht werden.

3.3.3. *Logistik und Produktion*

Nähe und Zusammenarbeit zwischen Lieferanten, Distributoren und Kunden sind heute entscheidende Wettbewerbsfaktoren. Sobald der Gesamtwarenfluss wächst, steigt der

Bedarf nach Schnelligkeit, Präzision, Wirtschaftlichkeit und Flexibilität im ganzen Logistiknetz. Nur die Unternehmen werden Erfolg haben, die das richtige Produkt zur richtigen Zeit und zum richtigen Preis an den richtigen Ort liefern. Das Logistiknetz zieht sich von der Beschaffung der Rohstoffe oder Produkte beim Lieferanten bis hin zu Auftragsauslieferung zum Kunden.

In Fertigungsunternehmen wird mit der **Materialbedarfsplanung** Produktionspläne erstellt und Einkaufsanforderungen kalkuliert. Basierend auf dem Auftragseingang und dem Bedarf wird ein klares Bild der Ressourcenanforderung gezeichnet. Bei der **Beschaffung** wird der gesamte Einkaufsprozess verwaltet, von der Angebotseinholung, über die Bestellung und Bestellüberwachung bis hin zum Wareneingang, der Wareneingangskontrolle und dem Abgleich mit der Lieferantenrechnung.

In der **Produktion** werden Produktionspläne erstellt, die aufgrund von Prognosen, basierend auf Verkaufszahlen, berechnet werden. Anschließend wird Ausführung der Pläne überwacht, wobei unterschiedliche Produktionsmodelle unterstützt werden. Mit Hilfe der **Materialwirtschaft** wird ein Balance zwischen Liefergenauigkeit und niedrigem Kapitaleinsatz beim Bestand hergestellt. Aufgrund von Analysen der Umschlaghäufigkeit und Verbrauch werden Bestellvorschläge für den Einkauf und die Produktion generiert.

Die **Auftragsabwicklung** verwaltet Aufträge, Lieferpläne, die Entnahme und Bereitstellung in Industrie- und Handelsunternehmen. Vom Auftragseingang bis zur Auftragsauslieferung wird mit Standardabläufen eine einfache Auftragsverarbeitung gewährleistet. Hilfreich sind dabei Funktionen wie das Abfragen des Auftragsstatus, das Feststellen der Liefertermine oder die Suche nach Alternativ-Produkten.

3.4. SAP R/3

Das weltweit am stärksten verbreitete ERP-System ist SAP R/3, es zählt als Standard für betriebswirtschaftliche Anwendungssoftware. SAP ist eine branchenneutrale Lösung, welche alle betriebswirtschaftlichen Bereiche unterstützt, verbindet und integriert. Neben der konsequenten Integration ist die umfangreiche Parametrisierung, die eine flexible Anpassung an firmenspezifische Anforderungen erlaubt, ein besonderes Merkmal von SAP. Das System R/3 wird von SAP seit 1992 für Client-/Server-Umgebungen angeboten und ist mittelfristig als R/2-Nachfolger anzusehen. Das "R" von den Produktlinien R/2

und R/3 steht für "Realtime" und hebt die sofortige Verbuchung und Aktualisierung von Daten hervor. Nachfolgend werden die Komponenten des R/3-Systems vorgestellt.

3.4.1. Module

SAP R/3 integriert in seinem Funktionsumfang eine Vielzahl betriebswirtschaftlicher Teilbereiche. Die betriebswirtschaftliche Funktionalität von R/3 wird in einer modularen Struktur realisiert. Jedes der Module lässt sich seinerseits in eine Menge sogenannter Komponenten zerlegen. Die SAP AG fasst die Menge der betriebswirtschaftlichen Module wiederum in den drei Kategorien Rechnungswesen, Logistik und Personal zusammen.

Module des Rechnungswesens

CO Controlling

Das Controlling stellt Instrumente für die Planung, Steuerung und Kontrolle der unternehmensweiten Kontrollinstrumente zur Verfügung. Es wird unterteilt in:

- Gemeinkosten-Controlling – für Kostenarten- und Kostenstellenrechnung und Soll-/Ist-Vergleiche
- Produktkosten-Controlling – für die Kostenträgerrechnung
- Ergebnis und Vertriebscontrolling – für die Beurteilung von Marktsegmenten nach Produkten, Kunden, Aufträgen oder Profit-Center

FI Finanzwesen

Finanzbuchhaltung ist für die Konsolidierung und das Finanzcontrolling zuständig. Außerdem wird es für die Finanzmittelüberwachung verwendet.

IM Investitionsmanagement

Das Investitionsmanagement gewährleistet eine integrierte Verwaltung und Abwicklung von Investitionsprojekten. Es ermöglicht dabei die Planung, Abrechnung und Wirtschaftlichkeitskontrolle der Investitionen.

TR Treasury

Dieses Modul ist eine Komplettlösung für ein effizientes Finanz-Management. Mit Hilfe von Treasury kann eine unternehmensweite Liquiditätsplanung durchgeführt werden, Finanzanlagen und Kredite können verwaltet werden und ein umfassendes Risikomanagement wird gewährleistet.

EC Unternehmenscontrolling

Mit dem Unternehmenscontrolling ist eine kontinuierliche Planung und Überwachung der Erfolgsfaktoren und Leistungskennzahlen des Unternehmens möglich. Es bietet eine schnelle Generierung unternehmensweit konsistenter Geschäftspläne und stellt eine hohe Verdichtung von Managementinformationen zur Verfügung.

Module der Logistik

PM Instandhaltung

Die Instandhaltung gewährleistet die Verwaltung und Strukturierung eigener Anlagen. Mit Hilfe eines Instandhaltungsinformationssystems wird zusätzlich die Ressourcen- und Wartungsplanung ermöglicht.

MM Material-Management

Die Funktionen des Material-Managements umfassen den Einkauf, die Bestandsführung und Disposition, die Lagerverwaltung und die Rechnungsprüfung. Diese wird zusätzlich unterstützt durch eine automatische Lieferantenbeurteilung und einen durchgängigen Workflow.

PP Produktionsplanung und -steuerung

Die Planung der Produktion, des Materialbedarfs und der Kapazitäten wird mit Hilfe der Produktionsplanung vollzogen. Auf dieser Grundlage werden Fertigungsaufträge erstellt. Zusätzlich wird die Verwaltung von Stücklisten und Arbeitsplätzen gewährleistet.

QM Qualitäts-Management

Das Qualitäts-Management ist für die Erfassung und Verwaltung der für die Qualitätssicherung relevanten Abläufe entlang der gesamten logistischen Kette vom Wareneingang

bis zur Fertigungs- und Endprüfung zuständig. Es koordiniert die Prüfabwicklung und Prüfmittelverwaltung und stößt Korrekturmaßnahmen an.

PS Projekt-Management

Das Projekt-Management gewährleistet die Koordination und Steuerung aller Phasen eines Projekts im direkten Zusammenwirken mit Einkauf und Controlling vom Angebot über Design und Genehmigung bis zum Ressourcen-Management und der Abrechnung

SD Vertrieb

Das Vertriebs-Modul unterstützt Funktionen für die Anfrage- und Angebotsabwicklung, die Kundenauftragsabwicklung, für Fakturierung und Versand.

Es bietet zusätzlich Funktionen zur Preisfindung, Variantenkonfiguration mit direkter Verbindung zur Ergebnisrechnung und Produktion

Modul Personalwesen

HR Human Resources

Dieses Modul setzt sich aus zwei Bestandteilen zusammen:

- HR-PD Personalplanung und –entwicklung – für Organisationsmanagement, Personalbeschaffung, Personalentwicklung, Personaleinsatz und Veranstaltungsmanagement
- HR-PA Personaladministration und - abrechnung – für Personalzeitwirtschaft, Leistungslohn, Personalabrechnung und Reisekosten

4. Integration

4.1. Trennung von Anwendungsschwerpunkten

Aus der Beschreibung der Grundlagen zu CRM und ERP in den vorangegangenen Kapiteln wurde die Vielzahl der Überschneidungen im Angebot von Funktionalitäten deutlich. So können z. B. Teile der Auftragsabwicklung von beiden Systemen durchgeführt werden, und für die Produktkonfiguration werden auch von beiden Systemen Funktionalitäten angeboten. CRM –Systeme bieten immer mehr auch ERP-Funktionalitäten an, wie z. B. für Logistik, oder Qualitätsmanagement; genauso versuchen ERP-Anbieter das Kundenbeziehungs-Management mit Funktionalitäten zu unterstützen.

Die Entscheidung, welches System für welche Aufgaben eingesetzt werden soll, wird von verschiedenen Faktoren beeinflusst, wie z. B. historische Faktoren, organisatorische Aspekte, technologische Anforderungen, Kosten etc. Ein Unternehmen kann z. B. ein CRM-System als führendes System für Kundenverwaltung und Auftragsabwicklung einsetzen und nur einzelne ERP-Module zusätzlich für Abrechnung oder Personalverwaltung verwenden. Genauso kann auch das ERP-System als führendes System eingesetzt werden und von dem CRM-System wird nur die „Call-Center“- Funktionalität genutzt.

CRM- und ERP-Systemen bieten zwar eine Vielzahl von gleichen Funktionalitäten an, diese sind aber auf einen unterschiedlichen Focus ausgerichtet. Bei CRM liegt der Focus auf der Kundenorientierung, wobei ERP eher auf die Wertschöpfungskette eines Unternehmens ausgerichtet ist. Die Kernkompetenzen der Systeme liegen in unterschiedlichen Bereichen. Um diese voll ausnutzen zu können, sollten die Systeme auch bei einem integrierten Einsatz ihre Eigenständigkeit bewahren.

Die typische Rollenaufteilung sieht meistens so aus, dass das CRM-System das Frontoffice darstellt, also jegliche Kommunikation und Interaktion mit Kunden, Interessenten oder Geschäftspartnern durchführt und das ERP-System die Rolle des Backoffices übernimmt, also für die Leistungserstellung und Abrechnung zuständig ist.

4.2. Gemeinsame Geschäftsobjekte

Da beide Systeme, sowohl das CRM-System als auch das ERP-System unabhängig voneinander eingesetzt werden können, gibt es bestimmte Objekte, die in beiden Systemen vorhanden sind.

Beide Systeme beinhalten beispielsweise Informationen über Kunden, um eine getrennte Abwicklung des operativen Geschäfts zu gewährleisten. Im CRM-System müssen Kundendaten zur Akquisition, zum Vertragsabschluss, zur Kundenbindung und zu Auswertungszwecken zur Verfügung stehen. Um z.B. gelieferte Leistungen in Rechnung stellen zu können, muss aber auch das ERP-System die Kunden kennen.

Ein weiteres Beispiel für gemeinsame Geschäftsobjekte sind Verträge. Sie werden beim Verkauf von Lieferungen und Leistungen angelegt und enthalten Informationen über die verkauften Produkte. In der Kommunikation mit dem Kunden ist es wichtig zu wissen, welche Verträge er bereits bei dem Unternehmen abgeschlossen hat und welche Produkte er bereits besitzt. Vertragsinformationen im CRM-System unterstützen Mitarbeiter beim Verkaufsgespräch, bieten Möglichkeiten für Cross- und Up-Selling und dienen zusätzlich zu Auswertungszwecken. Das ERP-System benötigt die Vertragsinformationen für die Durchführung der darin enthaltenen Aufträge. Das System muss wissen, welche Leistungen zu welchen Konditionen vereinbart wurden. Für die Abrechnung der Verträge müssen diesen auch Informationen über Zahlungskonditionen entnommen werden.

Ein weiteres Beispiel ist das Objekt „Produkte“. Auch hier kann eine doppelte Haltung sinnvoll sein. Es können noch eine Reihe weiterer gemeinsamer Geschäftsobjekte aufgezählt werden. Die Entscheidung, welche Objekte in beiden Systemen genutzt werden, wird aufgrund der Aufgabenverteilung der Systeme getroffen. Der Nachteil gemeinsamer Geschäftsobjekte ist, dass eine doppelte Haltung von Daten auch eine doppelte Pflege erfordert. Um die Konsistenz der Daten zu gewährleisten, muss eine ständige Synchronisation zwischen den Systemen erfolgen.

4.3. Systemübergreifende Geschäftsprozesse

Die Steuerung der betrieblichen Wertschöpfungskette erfordert eine ganzheitliche systemübergreifende Sichtweise. Die Geschäftsprozesse eines Unternehmens sind eine Abfolge von mehreren Funktionen, die auf ein bestimmtes Ziel ausgerichtet sind. Die ver-

schiedenen Aufgaben in einem Unternehmen werden von mehreren Systemen durchgeführt. Bei der Durchführung eines Geschäftsprozesses kann es sein, dass dieser die Grenzen eines Systems überschreitet und beim Durchführen der einzelnen Funktionen zwischen den Systemen wechseln muss, um zu einem Abschluss zu kommen. Am Beispiel von folgenden Geschäftsprozessen wird der Ablauf, welcher sich zwischen CRM- und ERP-System abspielt, dargestellt.

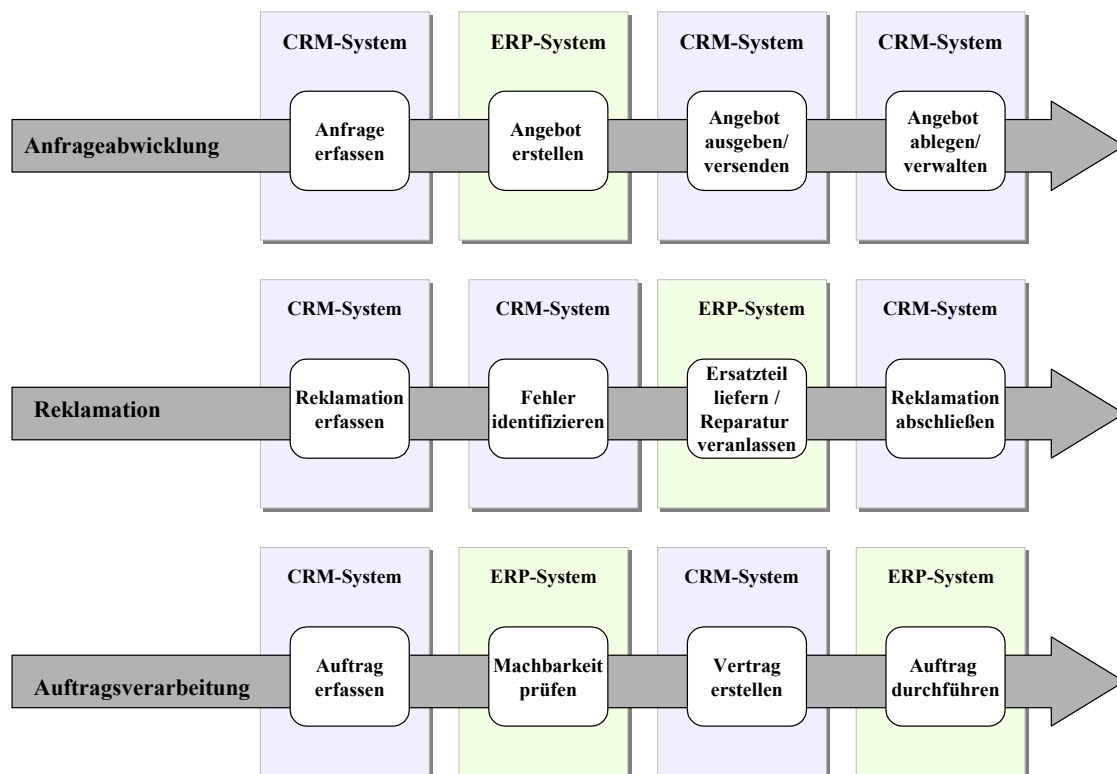


Abbildung 6: Darstellung des Ablaufs von Beispiel-Geschäftsprozesse

Anfrageabwicklung

Die Angebotsverwaltung wird vom CRM-System übernommen, hier werden die Funktionen für das Erfassen, Versenden und Ablegen der Angebote durchgeführt. Die Verwaltung der Produkte und alle Informationen bzgl. Verfügbarkeit, Lieferzeit und Preise sind im ERP-System verfügbar, deswegen wird hier das Erstellen der Angebote übernommen. Bei dem Ablauf des Geschäftsprozesses für die Anfrageabwicklung muss nach dem Erfassen der Anfrage im CRM-System ins ERP-System gewechselt werden, um dort die Angebotserstellung durchzuführen. Danach muss wieder ein Wechsel zwischen den Sys-

temen erfolgen, da das Weiterverarbeiten der Angebote und die Verwaltung im CRM-System durchgeführt wird.

Reklamationsprozess

Jede Interaktion mit dem Kunden wird von dem CRM-System übernommen. So ist es auch im Fall von Beschwerden. Diese werden im CRM-System erfasst und mit Hilfe von Wissensdatenbanken wird versucht, das Problem für die Beschwerde zu identifizieren und eine Lösung zu finden. Handelt es sich dabei um eine mangelhafte Leistung, müssen entweder Ersatzteile geliefert werden oder eine andere Form der Gewährleistung muss erfolgen. Alle diese Aufgaben sind Teil des ERP-Systems. Für die Problem-Lösung muss der Geschäftsprozess für die Reklamationsbearbeitung im ERP-System weitergeführt werden. Die Aufgabe der Nachverfolgung und des Abschlusses einer Reklamation wird wiederum vom CRM-System übernommen. Um den Reklamationsprozess abzuschließen, muss also ein nochmaliger Wechsel zwischen den Systemen erfolgen.

Auftragsabwicklung

Auch bei dem Ablauf des Prozesses für die Auftragsabwicklung muss zwischen den Systemen hin und her gewechselt werden. Die Erfassung des Auftrages erfolgt im CRM-System. Die Realisierbarkeit des Auftrages wird im ERP-System geprüft, weil hier Informationen über Bonität des Kunden, Verfügbarkeit und Lieferzeit der Produkte vorhanden sind. Die Erfassung und Erstellung der Verträge übernimmt das CRM-System und der nächste Schritt – die Durchführung der vereinbarten Leistung – liegt wiederum im ERP-System.

Anhand der drei beschriebenen Beispiel-Prozessen lässt sich Folgendes erkennen: Damit ein optimaler und effizienter Ablauf der Geschäftsprozesse erfolgen kann, muss ein reibungsloser Wechsel zwischen den Systemen für die Durchführung der benötigten Funktionen gewährleistet sein.

4.4. Integrationstechniken

Wie aus den Kapiteln 4.1, 4.2 und 4.3 hervorgeht, sind bei allen Integrationsszenarien drei fundamentale Integrationsprobleme involviert:

- das Synchronisieren von Daten zwischen den Systemen,
- das Isolieren von Anwendungen und ihrer Geschäftslogik voneinander
- und das Automatisieren von Geschäftsprozessen.

Aus den verschiedenen Technologien, muss die adäquate Lösung für die Integrationsprobleme ausgewählt werden.

CRM- und ERP-Systeme sind in der Regel als Client-Server Architektur konzipiert. Daraus ergeben sich drei Ebenen, auf denen die Integration mit anderen Systemen ansetzen kann. Im Folgenden werden die verschiedenen Integrationsebenen und das dafür verwendete Kommunikationsmodell dargestellt.

4.4.1. Integrationsebenen

Die drei Ebenen, auf denen die Integration zwischen Systemen realisiert werden kann sind:

- Präsentationsebene
- Funktionsebene
- Datenbankebene

In Abbildung 7 werden die drei Integrationsebenen grafisch dargestellt, wobei hier die Präsentationsebene dem User Interface Level entspricht, die Funktionsebene dem Method Level und die Datenbankebene dem Data Level.

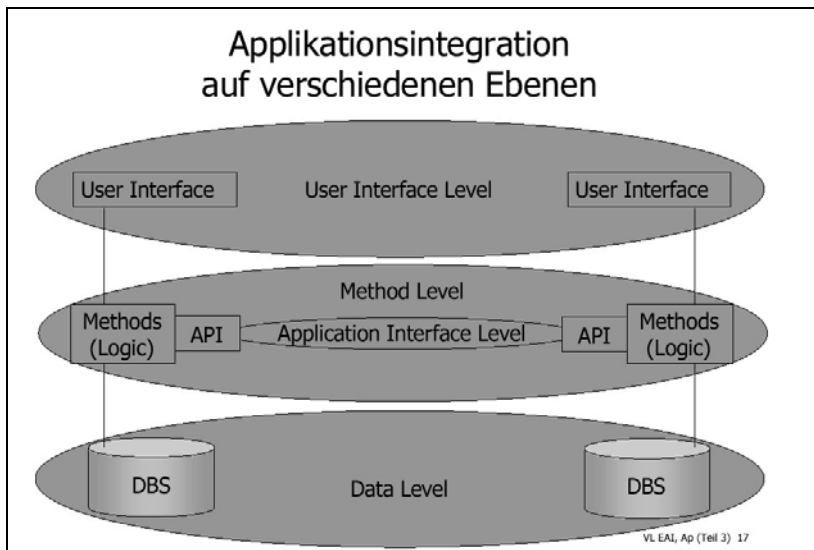


Abbildung 7: Ebenen der Integration

Integration auf Präsentationsebene

Bei der Integration auf der Präsentationsebene wird den Benutzeroberflächen ein oder mehrerer Systeme eine neu entwickelte Oberfläche vorgeschaltet. Ziel ist es die verschiedenen Systeme von einer einzigen Benutzerfläche aus bedienen zu lassen und eine Prüfung der Daten und des Ablaufs zwischen den Systemen durchzuführen.

Der Nachteil dieser Integration ist, dass der Entwickler nur die Ausgaben der einbezogenen Systeme verarbeiten und nicht auf deren Logik zurückgreifen kann.

Integration auf Funktionsebene

Die Integration auf Funktionsebene ist eine Zusammenführung über Schnittstellen (APIs), die von den jeweiligen Anwendungen bereitgestellt werden. Über APIs (Applikation Programming Interface) stellen die Systeme eine Reihe von Funktionen und Prozeduren zur Verfügung, die außerhalb der Anwendungen aufgerufen werden können.

Der Vorteil dieser Integration liegt in der Wiederverwendung von bestehender Geschäftslogik, ohne dabei in innere Abläufe eines Systems eingreifen zu müssen. Bei der Integration auf Funktionsebene wird eine steuernde Instanz benötigt, die den richtigen Ablauf bei der Durchführung der Funktionen eines Geschäftsprozesses gewährleistet.

Integration auf Datenbankebene

Die Integration auf Datenbankebene ermöglicht es, Daten zwischen den Datenbanken der einzelnen Systeme zu bewegen. Sie umgeht dabei Präsentations- und funktionale Schicht und greift direkt auf die Daten zu, die ein zu integrierendes System erzeugt und verwaltet. Aufgaben zur Pflege der Datenbestände, wie z.B. Replikation und Synchronisierung von Daten zwischen den einzelnen Systemen, können schnell realisiert werden.

Diese Integrationsform ist die flexibelste, da alle Daten der jeweiligen Systeme im Zugriff stehen, erfordert aber ein genaues Kenntnis des Datenmodells, da es durch die fehlende Geschäftslogik zu Fehlinterpretationen kommen kann.

4.4.2. Kommunikationsmodell

Das Kommunikationsmodell regelt die Art der Kommunikation zwischen den integrierten Systemen. Man unterscheidet zwischen synchroner und asynchroner Kommunikation.

Synchrone Kommunikation

Bei der synchronen Kommunikation erfolgt der Kommunikationsvorgang in einem zwischen Sender und Empfänger abgesprochenen Verfahren. Nach dem Verschicken einer Nachricht wartet der Sender auf eine Antwort des Empfängers. Sender und Empfänger sind somit voneinander abhängig. Synchrone Kommunikation wird eingesetzt, wenn ein Sender für seine weitere Verarbeitung auf die Antwort des Empfängers angewiesen ist.

Asynchrone Kommunikation

Sender und Empfänger arbeiten bei der asynchronen Kommunikation relativ unabhängig voneinander. Weder muss der Empfänger auf die Nachricht des Senders antworten, noch wartet der Sender auf eine Antwort des Empfängers. Zum Einsatz kommt die asynchrone Kommunikation bei reiner Übertragung von Informationen.

4.5. Grundlegender Aufbau einer Integrationsplattform

Die Integration ist die Grundlage für eine optimale Prozessverzahnung und Ablaufkontrolle. Sie muss die Durchführung von konsistenten Transaktionen für den Ablauf von

Geschäftsprozessen zwischen den Systemen gewährleisten. Dies wird durch eine reibungslose Kommunikation mit folgenden Komponenten ermöglicht:

- Initiation – wodurch wird die Kommunikation angestoßen?
- Analysis – was soll gemacht werden?
- Rules – wie soll es gemacht werden?
- Transport – auf welchem Weg?
- Commitment – wurde der Vorgang erfolgreich durchgeführt?

Initiation

Der Aufbau einer Kommunikation zwischen den Systemen muss durch bestimmte Ereignisse initiiert werden. Die Ereignisse werden entweder durch Benutzereingaben oder durch Hintergrundprozesse ausgelöst. Ereignisse sind entweder Datenänderungen oder Aufruf von Funktionen während eines Prozessablaufs. Das Ergebnis ist der Aufbau einer Kommunikation, um Daten zwischen den Systemen auszutauschen oder um auf Funktionen des anderen Systems zuzugreifen.

Analysis

Um die weitere Vorgehensweise zu bestimmen, müssen die Ereignisse analysiert werden. Bei Datenänderungen muss festgestellt werden:

- um was für Daten es sich handelt,
- ob die Änderung für das andere System relevant ist,
- ob nur einzelne Daten oder das gesamte Objekt (z.B. Kunde, Auftrag) an dem sich die Daten geändert haben, übertragen werden sollen,
- welche zusätzlichen Informationen zu dem Objekt noch benötigt werden.

Bei Funktionsaufrufen muss geprüft werden:

- welche Aufgaben durch die Funktion durchgeführt werden sollen,
- ob diese Aufgaben in einem anderen System durchzuführen sind,
- welche Informationen das System für die Durchführung noch benötigt,
- welche Ergebnisse erwartet werden.

Rules

Die Durchführung von Aufgaben erfordert bestimmte Geschäftsregeln, die eingehalten werden müssen. Der Ablauf eines Prozesses muss gesteuert werden, um eine bestimmte Reihenfolge bei der Durchführung der einzelnen Schritte zu gewährleisten. Es wird festgelegt, auf welches Ereignis welche Funktion durchgeführt werden soll. Zusätzlich werden Regeln für die Art der Durchführung von Funktionen festgelegt:

- der Zeitpunkt der Durchführung,
- auf welche Funktion im anderen System für die Durchführung zugegriffen wird,
- welche Parameter übergeben werden,
- wie mit den Ergebnissen verfahren wird,
- was beim Auftreten von Fehlern gemacht wird.

Transport

Um von einem System Funktionen des anderen Systems aufrufen zu können, wird eine Kommunikationsverbindung zwischen den Systemen hergestellt. Format und Protokoll für den Transport werden bestimmt. Transformation auf beiden Seiten sorgt für die Umwandlung in die richtigen Formate. Für den eigentlichen Datentransfer wird das Transport-Medium bestimmt und Adressierung und Routing werden festgelegt.

Commitment

Die Rücksendung einer Bestätigung signalisiert dem aufrufenden System den erfolgreichen Abschluss der ausgelösten Transaktionen. Im Fehlerfall wird eine Fehlermeldung generiert und zurückgesendet und alle offenen Transaktionen werden rückgängig gemacht, damit die Konsistenz der Systeme erhalten bleibt und der Benutzer die Möglichkeit hat den Vorgang nach Beheben des Fehlers zu wiederholen.

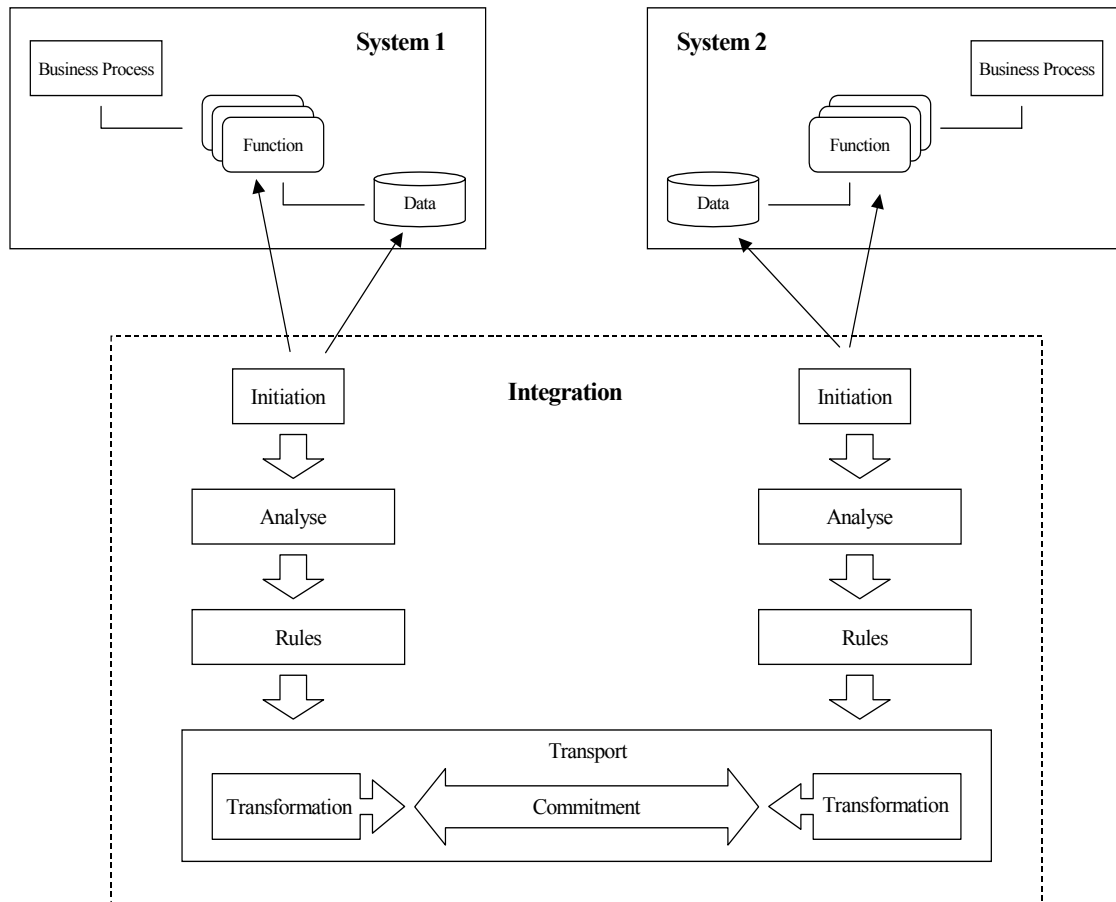


Abbildung 8: Aufbau einer Integrationsplattform

5. Integration von Clarify und SAP

In diesem Kapitel soll ein Integrationskonzept erarbeitet werden, dass durchgängige Prozesse zwischen den Systemen Clarify und SAP gewährleistet. Dabei bestehen bei jedem Prozess dieselben Anforderungen an die Schnittstellen der Systeme:

- Funktionsaufrufe an das andere System zu senden und
- Funktionsaufrufe vom anderen System entgegenzunehmen.

Zur Einführung werden die Architekturen der beiden Systeme dargestellt und die angebotenen Integrationstechnologien vorgestellt, um aufgrund der Anforderungen die geeignete Technologie auszuwählen.

Für die Kommunikation der Systeme über die ausgewählten Schnittstellen muss ein Lösungskonzept erarbeitet werden, aufgrund dessen die Integration realisiert werden kann.

5.1. Clarify

5.1.1. Architektur

Clarify eFrontOffice ist eine Client/Server-Applikation. Der Server beinhaltet eine relationale Datenbank, welche die gesamten Kundendaten und andere Informationen bzgl. Aktivitäten, Kontaktpersonen, etc. enthält. Der Zugang erfolgt über den Client, der die Schnittstelle zum Benutzer darstellt, wobei dieser die Wahl hat zwischen:

- Desktop Client – Fat-Client für Mitarbeiter im Unternehmen, die für den Kunden telefonisch zur Verfügung stehen
- Web Client – Thin-Client, hat die selben Funktionalitäten wie der Fat-Client, kann aber nur bei einer 3-tier-Architektur verwendet werden
- Traveler Client – für Außendienstmitarbeiter, die den Kunden im persönlichen Gespräch beraten

Das Clarify-System lässt sich sowohl als 2-tier-Architektur als auch als 3-tier- bzw. n-tier-Architektur aufbauen.

In Abbildung 9 wird die 2-tier-Architektur eines Clarify-Systems dargestellt:

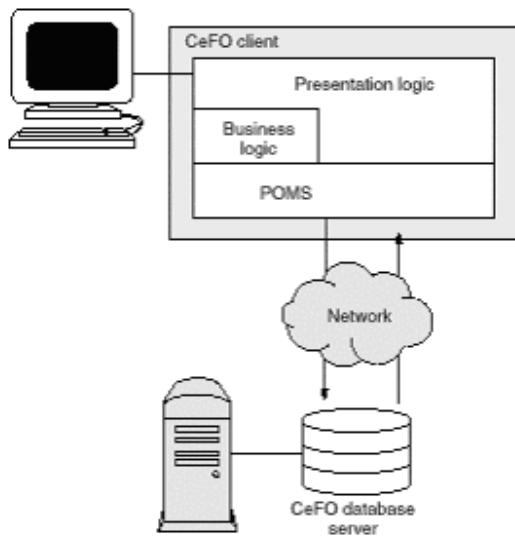


Abbildung 9: 2-tier-Architektur von Clarify

In einer 2-tier-Architektur läuft der Client auf einer Workstation und greift über das Netzwerk direkt auf die Datenbank zu. In dieser Konfiguration sind die gesamten Anwendungsdienste auf der Client-Seite. Diese umfassen die Präsentationslogik, die Business Logik und POMS (Persistent Object Management System). POMS ist ein Datenbank-Verbindungsdienst und wird für die Kommunikation zwischen Client und Server verwendet. Er wandelt den vom Client generierten Meta-Code in SQL-Code um und leitet ihn an die Datenbank weiter. Das Ergebnis wird dann an die aufrufende Client-Anwendung zurückgegeben. Die Business Logik ist für den kompletten Ablauf der Funktionen verantwortlich. Sie enthält Regeln für den Workflow und die Zustellung von Nachrichten. Die Präsentationslogik wird durch eine grafische Benutzeroberfläche dargestellt, in der die Anwendungsdaten aufbereitet und angezeigt werden.

Der Unterschied zwischen 2-tier und 3-tier besteht darin, dass bei der 3-tier-Architektur der Client nicht direkt auf die Datenbank zugreift, sondern über einen Applikations-Server. Dieser kann auf einer oder mehreren Maschinen laufen und enthält die gesamte Business Logik. Er verwendet auch POMS für den Zugriff auf die Datenbank. Die Kommunikation zwischen dem Client und dem Applikations-Server wird von Tuxedo übernommen, einer Middleware von der Firma BEA Systems. Zusätzlich zur Präsentationslogik enthält der Client noch Dienste für entfernte Funktionsaufrufe und ein Stub-POMS

für die Weiterleitung von Datenbankabfragen an die POMS-Komponente des Applikations-Servers. Die Komponenten der 3-tier Architektur sind noch einmal in Abbildung 10 dargestellt:

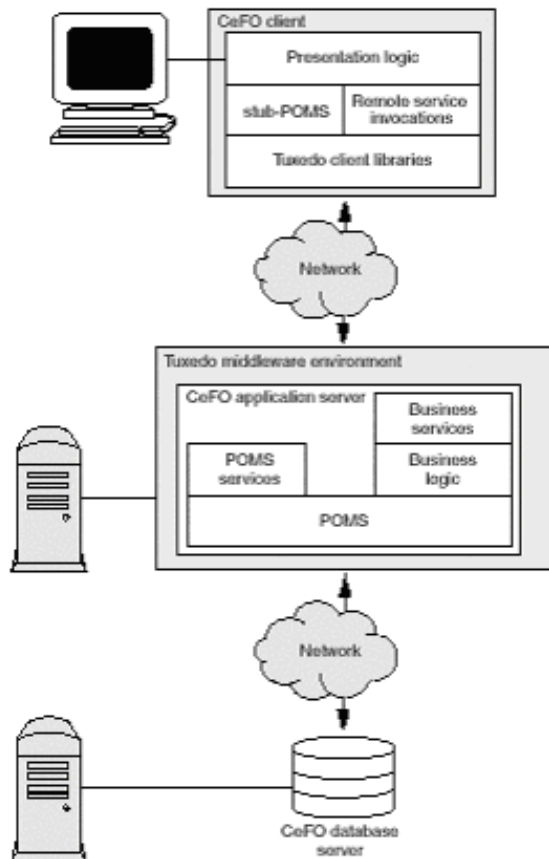


Abbildung 10: 3-tier-Architektur von Clarify

5.1.2. Integrationstechnologien

Clarify bietet eine Reihe von Techniken für die Realisierung von Schnittstellen zu Fremdsystemen. Diese werden unterteilt in:

- Client-basierte Techniken werden vom Benutzer oder von Client Anwendungen initiiert und ermöglichen das Darstellen der Daten von Fremdsystemen. Clarify unterstützt hier die Integration über ActiveX Controls.
- Applikations-Server-basierte Techniken ermöglichen die Realisierung von Schnittstellen auf Basis der 3-tier-Architektur. Hierzu gehört die Integration über

die von Clarify verwendete Middleware Tuxedo oder andere Enterprise Applikation Integration Anwendungen.

- Server-basierte Techniken werden auf dem Server angewendet und ermöglichen Updates auf die Datenbank. Dazu gehören: Clarify Data Exchange, Clarify API, Clear Basic und die Integration über die Technologie der verwendeten Datenbank.

Im Folgenden werden die einzelnen Techniken dargestellt.

AktiveX Controls

Auf der Windows Plattform kann der Clarify Client ActiveX Objekte einbinden. Sie werden verwendet, um die Steuerelemente der Benutzeroberfläche zu unterstützen. Des Weiteren besteht mit der Verwendung von ActiveX die Möglichkeit, Clarify in eine verteilte Objektarchitektur einzubinden – ActiveX kann benutzt werden, um auf die Methoden und Eigenschaften von COM-Objekten zuzugreifen, bzw. andere Anwendungen zu integrieren

TUXEDO

Clarify verwendet TUXEDO nicht nur, um Remote POMS innerhalb seiner eigenen Architektur zu realisieren, sondern bietet darüber hinaus die Möglichkeit, Fremdsysteme unter Verwendung von TUXEDO-Requests einzubinden, insofern diese ihrerseits TUXEDO-Requests verarbeiten können. Die TUXEDO-Requests an die Fremdsysteme werden in einem sogenannten Wrapper Service gebündelt, welcher im Applikation Server arbeitet.

Clarify Data Exchange

Data Exchange (dataex) ist eine Batch-Schnittstelle mit der Daten im ASCII-Format in die Clarify-Datenbank importiert oder exportiert werden können. Der Nachteil dieser Schnittstelle ist, dass keine vorhandenen Daten verändert oder gelöscht werden. Des Weiteren eignet sich Data Exchange aufgrund der Performance nicht für die Verarbeitung von Massendaten. Die Empfehlung des Herstellers besagt hierzu, diese Schnittstelle nur bis zu einer Größenordnung von 5.000 Datensätzen zu verwenden.

Clarify API

Das Clarify Applikation Programming Interface (Clarify Api) stellt auf dem Clarify Server eine Bibliothek von C-Funktionen zur Verfügung, die für die Integration von Clarify Funktionalität in andere Anwendungen genutzt werden kann.

Das Clarify API erlaubt den Zugriff auf jegliche Objekte der Clarify-Datenbank. Die Verwendung dieses Funktionsumfangs setzt fundierte Kenntnisse des Datenmodells voraus. Zur Anwendung dieser Funktionen wird ein Mindestumfang an Informationen benötigt, um die Konsistenz der Clarify-Datenbank nicht zu verletzen. Nachteilig ist, dass nur vorhandene Funktionen genutzt werden können und das API nicht durch weitere Funktionen erweitert werden kann.

Clear Basic

Clear Basic, eine Version von Visual Basic, ist die von Clarify verwendete Programmiersprache. Clear Basic Code kann sowohl auf dem Client also auch auf dem Server ausgeführt werden. Auf dem Server wird Clear Basic nicht zur Darstellung, sondern zum Zugriff auf die Datenbank verwendet. Die Ausführung von ClearBasic Code auf dem Server wird primär für die Realisierung von Schnittstellen zu anderen Systemen verwendet.

Datenbank-Technologie

Die Integration von Clarify mit anderen System kann auch unter Verwendung der Technologie des Datenbankherstellers, auf dessen Datenbank das Clarify-System aufgesetzt wird, durchgeführt werden. So bietet z. B. Oracle, dessen Datenbank am weitesten verbreitet ist, beim Einsatz mit Clarify die Möglichkeit der Integration über Stored Procedures. Funktionen und Prozeduren werden dabei in der Oracle-eigenen Programmiersprache PL/SQL implementiert und als Stored Procedures, die direkt in der Datenbank laufen, abgelegt. Über diese kann dann von außen zugegriffen werden.

5.1.3. Auswahl der Integrationstechnologie

Auf Grundlage der Anforderungen werden im Folgenden die aufgeführten Integrations-technologien hinsichtlich ihrer Verwendbarkeit betrachtet und es werden die kritischen Punkte beleuchtet, die bei ihrem Einsatz berücksichtigt werden müssen.

Die Integration über die Präsentationsebene (Client-basierte Technologie) hat den Nachteil, dass eine Vielzahl von Verbindungen zwischen den beiden Systemen aufgebaut werden muss. Über jeden Client würde direkt auf die Funktionen des anderen Systems zugegriffen und es gäbe keine Zentralisierung der Logik, was sich nachteilig auf die Performance auswirken würde.

Bei der Integration über Enterprise Applikation Integration, in diesem Fall TUXEDO, werden die Anforderungen erfüllt, Funktionsaufrufen abzuschicken und entgegenzunehmen. Diese Art von Integration kann aber nur für 3-tier-Architekturen realisiert werden und lässt sich nicht für 2-tier-Architekturen verwenden, die bei der Nutzung von Clarify sehr weit verbreitet sind.

Die Verwendung von Data Exchange eignet sich nur für den Batch-Input im asynchronen Modus. Es besteht keine Möglichkeit, Funktionsaufrufe von anderen Systemen entgegenzunehmen oder an andere Systeme zu versenden.

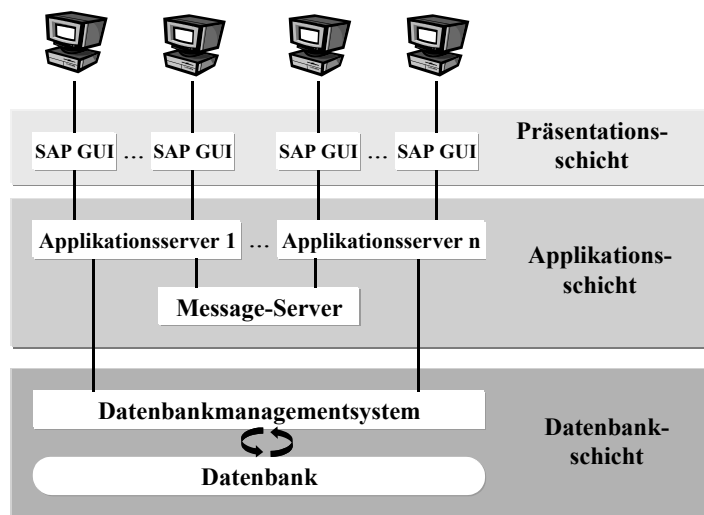
Das Applikation Programming Interface (API) erfüllt zwar die gestellten Anforderungen, ist aber nicht flexibel genug, da nur vorhandene Funktionen verwendet werden können, aber keine Erweiterungen möglich sind.

Es empfiehlt sich also, die Integration über die Technologie des Datenbankherstellers durchzuführen. Da Oracle der am weitesten verbreitete Hersteller ist, wird diese Technologie als Grundlage für die Integration ausgewählt. Ein zusätzlicher Vorteil bei der Integration über die Datenbank ist, dass andere Systeme die auch auf die Datenbank aufsetzen, mit integriert werden können.

5.2. SAP

5.2.1. Architektur

Die SAP R/3-Software-Architektur basiert auf dem Client-Server-Prinzip. Die Systemprozesse sind auf mehreren Servern verteilbar und bestehen aus einer Präsentationsschicht, einer Applikationsschicht und einer Datenbankschicht. Jede einzelne dieser Schichten wird durch eine Softwarekomponente repräsentiert. Das Gesamtsystem kann auf mehrere Rechner verteilt werden. Sowohl in vertikaler Richtung als auch in horizontaler Richtung sind verschiedene Szenarien möglich. In Abbildung 11 sind die einzelnen Schichten des R/3-Systems dargestellt.



[KEL00]

Abbildung 11: Die Client-Server-Architektur des SAP-Systems

Präsentationsschicht

Die Präsentationsschicht dient der Realisierung der graphischen Bedienoberfläche als Schnittstelle zum Benutzer. Mit Hilfe des SAPGUI (SAP-eigenes Graphical User Interface) werden Benutzerinteraktionen entgegengenommen und an die Applikationsschicht weitergeleitet. Die Verarbeitungsergebnisse werden dann auf dem Bildschirm des Benutzers dargestellt. Die Präsentationsschicht enthält keinerlei Anwendungslogik. Diese ist vollständig in der Applikationsschicht abgelegt.

Applikationsschicht

Die Applikationsschicht ist für den kompletten Ablauf der Funktionen verantwortlich. Die Softwarekomponenten der Applikationsschicht bestehen aus einem oder mehreren Applikations-Servern und einem Message-Server, der für die Kommunikation zwischen den Applikations-Servern zuständig ist. Die Applikations-Server stellen Dienste für den Betrieb des SAP-Systems zur Verfügung. Theoretisch würde ein Applikations-Server genügen, in der Praxis werden die Dienste meistens auf mehrere Applikations-Server verteilt [KEL00].

Datenbankschicht

In der Datenbankschicht werden alle vom R/3-System benötigten Daten in einer zentralen relationalen Datenbank gespeichert und verwaltet. Dabei sind nicht nur Anwendungsdaten, sondern auch alle Verwaltungsdaten, Customizing-Einstellungen, Quelltexte etc. enthalten. Die Datenbankschicht setzt sich aus dem RDBMS (Relationales Datenbank-Managementssystem) und der eigentlichen Datenbank zusammen [KEL00].

Die logische Aufteilung in Präsentation, Anwendung und Datenbank findet sich auch in der physischen Struktur wieder. Das R/3 lässt sich jedoch sowohl als zentrales System als auch als zwei- oder dreistufige Architektur aufbauen (Abbildung 12) [VIC01], wobei das einstufige Serverkonzept in der Praxis nicht verwendet wird.

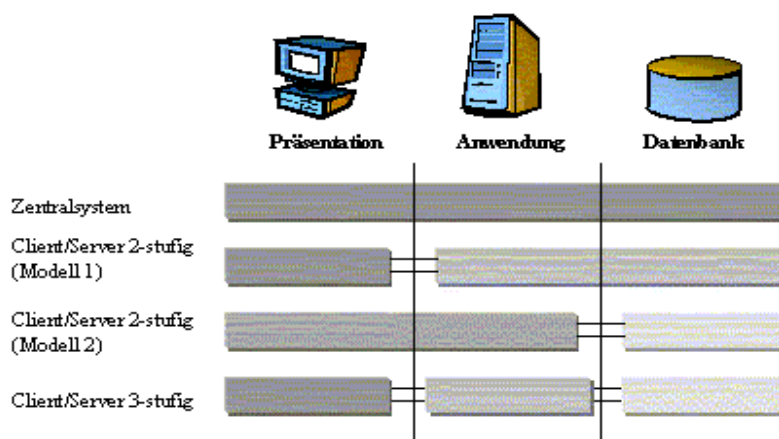


Abbildung 12: Architektur-Typen von SAP R/3

- *Zentrale R/3-Konfiguration*: alle Dienste werden von einem Rechner geleistet.
- *Zweistufige R/3-Konfiguration*: Anwendung und Datenbank werden auf einem Server zusammengefasst, die Terminals werden von einem separaten Rechner, dem Präsentationsserver, verwaltet.
- *Alternative, zweistufige R/3-Konfiguration*: die Datenbankdienste laufen auf einem separaten Rechner, während die Anwendungs- und Präsentationsdienste auf einem Server gruppiert sind.
- *Drei- und mehrstufige R/3-Konfiguration*: für die Präsentations-, Anwendungs- und Datenbankdienste werden eigenständige Rechner verwendet.

5.2.2. *Integrationstechnologien*

Um eine Verbindung zwischen SAP R/3 und einem externen System herzustellen, existieren eine Reihe von Technologien. Diese werden im Nachfolgenden dargestellt, um die Technologie auswählen zu können, die den gestellten Anforderungen entspricht. Genau wie bei Clarify können die Integrationstechnologien nach den Architekturebenen klassifiziert werden:

- Integration über die Präsentationsebene – Hierfür wird die Technologie SAP Automation verwendet.
- Integration über die Anwendungsebene – Die dazugehörige Technologie ist die Integration über BAPIs, bzw. Funktionsbausteine
- Integration über die Datenbankebene – Hierzu gehört Batch Input und ALE.

Nachfolgend werden die einzelnen Technologien beschrieben.

SAP Automation

SAP Automation bietet eine GUI-Schnittstelle, mit der Daten, welche zwischen dem SAPGUI und dem Applikations-Server ausgetauscht werden, für externe Anwendungen zugänglich gemacht werden. Es besteht aus mehreren Komponenten, die das Component Object Model (COM) unterstützen. Damit erhalten alle COM konformen Anwendungen Zugriff auf SAP Systeme über OLE Automation. Bei den Komponenten handelt es sich teils um OLE Server, teils um ActiveX Controls.

Mit der SAP Automation GUI-Schnittstelle kann dem Standard-SAPGUI eine andere Benutzeroberfläche vorgeschaltet werden und es ermöglicht externen Anwendungen, Interaktionen zwischen dem Endbenutzer und dem SAPGUI zu überwachen und festzuhalten. Mit SAP Automation kann aber nicht auf die Geschäfts- und Transaktionslogik der R/3-Applikationen zugegriffen werden.

Funktionsbausteine / BAPI

Mit Hilfe von RFC (Remote Function Call), dem von SAP genutzten Protokoll für entfernte Kommunikation, können externe Anwendungen auf die Funktionalität der R/3-Applikationen zugreifen. Dabei gibt es folgende Möglichkeiten:

- **Direktaufrufe von Funktionsbausteinen**
Funktionsbausteine sind Prozeduren, die speziell für den externen Aufruf aus anderen Applikationen vorgesehen sind. Sie sind die mit Abstand am häufigsten verwendete Methode, um anwendungsübergreifende wiederverwendbare Softwarekomponenten für ein SAP-System abzubilden. Alle zum RFC freigegebenen Funktionsbausteine können von anderen SAP-Systemen und Fremdsystemen verwendet werden.
- **Verwendung von BAPIs zum Zugriff auf Geschäftsobjekte**
Das R/3-System beinhaltet eine Reihe von vordefinierten Geschäftsobjekten, zur objekt-orientierten Darstellung der realen Welt. Das können zum Beispiel Aufträge, Kundenkonten, Mitarbeiter etc. sein, die als Klassen im R/3-System abgebildet sind. Die von SAP zur Verfügung gestellten BAPIs (Business Application Programming Interface) sind Schnittstellen, über die andere Anwendungen auf die Geschäftsobjekte zugreifen können. Sie sind ein Set von Methoden, um Geschäftsobjekte zu bearbeiten. BAPIs sind als Funktionsgruppen realisiert, die einzelnen Methoden werden dabei als Funktionsbausteine abgebildet, auf die über RFC zugegriffen werden kann.

Batch Input

Batch Input wird verwendet, um große Mengen von Daten von einem Fremdsystem an das R/3-System zu übertragen. Diese Schnittstelle eignet sich für den einmaligen Import von Massendaten oder für periodische Datentransfers aus externen Systemen. Die Daten

werden von den Fremdsystemen in Form von Dateien an SAP übergeben. Mit Hilfe von Batch-Input-Mappen werden die Daten konvertiert und ins R/3-System übernommen.

ALE

ALE (Applikation Link Enabling) ist ein Mechanismus zur Synchronisation verteilt gehaltener Datenbestände. Es umfasst einen betriebswirtschaftlich kontrollierten Datenaustausch zwischen verteilten Anwendungen bei konsistenter Datenhaltung. Der asynchrone Datenaustausch erfolgt über ein von SAP definiertes Datenformat, das IDoc. IDoc ist der SAP-eigene Standard für Geschäftsdokumente und basiert auf dem in der Praxis weitverbreiteten EDI-Konzept. EDI ist ein Standard für den Austausch von Geschäftsdokumenten zwischen Unternehmen.

5.2.3. Auswahl der Integrationstechnologie

Aufgrund der Anforderungen, die an die Prozessintegration gestellt werden, muss anhand der Verwendbarkeit die geeignete Schnittstelle zum SAP-System ausgewählt werden.

Genau wie bei Clarify ist auch bei SAP eine Integration über die Präsentationsebene, aufgrund der fehlenden Zentralisierung der Logik nicht zu empfehlen.

Die Technologien Batch Input und ALE eignen sich nur für asynchrone Integration. Synchrone Funktionsaufrufe werden dabei nicht unterstützt.

Die Integration über die Datenbank-Technologie, die bei Clarify ausgewählt wurde, wäre theoretisch auch bei SAP möglich. Aber durch den noch immer vorhandenen Einsatz von Pool- und Clustertabellen bei der Verwaltung von SAP-Daten, wird eine Integration über die Datenbank in der Praxis erheblich erschwert.

Bei SAP fällt die Auswahl auf die Integration mit BAPI und Funktionsbausteinen. Die Anforderungen der Prozessintegration, Zugriffe auf Funktionalitäten des R/3-Systems zu gewährleisten, werden durch die Bereitstellung der BAPIs voll erfüllt. Die vorhandenen Funktionalitäten können mit Hilfe von Funktionsbausteinen jederzeit erweitert werden.

5.3. Technische Realisierung eines generischen Schnittstellenmoduls

Aus der Darstellung der Integrationstechnologien (Kapitel 5.1.2 und 5.2.2), welche von den Systemen Clarify und SAP angeboten werden, geht hervor, dass den Anforderungen für die Integration der beiden Systeme zur Gewährleistung durchgängiger Geschäftsprozesse folgende Schnittstellen entsprechen:

- auf der Seite von Clarify:
die Integration über die Datenbank-Technologie - wegen der weiten Verbreitung wird hier Oracle als Grundlage für die technische Realisierung ausgewählt
- auf der Seite von SAP:
die Integration über BAPIs und Funktionsbausteine.

Um eine Verbindung zwischen diesen Schnittstellen der Systeme herzustellen, ist eine bidirektionale Schnittstellenkomponente erforderlich, welche die Kommunikation zwischen den Systemen gewährleistet. Nach der Spezifikation der Anforderungen und dem Erarbeiten des Lösungskonzepts, wird in diesem Kapitel die Realisierung dieser Schnittstellenkomponente vollzogen.

5.3.1. Anforderungen

Für die nahtlose Prozessintegration zwischen Clarify und SAP soll ein generisches Schnittstellenmodul zur Verfügung gestellt werden, mit dessen Hilfe die Systeme über die genannten Schnittstellen jederzeit Funktionsaufrufe im anderen System initiieren können. Um dies zu gewährleisten, muss die Schnittstellenkomponente folgende Anforderungen erfüllen:

- **Anmeldung und Verbindungsaufbau**
Um von einem System aus Nachrichten an ein anderes System zu versenden, muss eine Kommunikationsverbindung zwischen den Systemen hergestellt werden. Diese kann je nach Anforderung der Systeme dauerhaft, also permanent bestehen, oder nur kurzzeitig für die Abwicklung einer Transaktion aufgebaut werden. Das Schnittstellenmodul ist für den Aufbau und die Bereitstellung dieser Verbindung zuständig. Es muss die Anforderungen der Systeme, in diesem Fall Oracle und SAP, kennen und wissen, wie es sich bei den Systemen anmelden, bzw. identifizieren soll damit diese Verbindung zustande kommt.

- Transformation

Der Aufruf von Funktionen erfolgt über bestimmte Funktionssignaturen, die von den Systemen vorgegeben werden. Da die Funktionssignaturen in den Systemen verschieden sind, ist es die Aufgabe des Schnittstellenmoduls, Funktionsaufrufe, die von Oracle initiiert werden in das SAP-Format und umgekehrt die Aufrufe aus dem SAP-System in das Oracle-Format zu transformieren. Nicht nur die Funktionssignaturen unterscheiden sich, sondern auch die Datentypen werden von den Systemen verschieden abgebildet. Für die Parameterübergabe bei dem Funktionsaufruf, müssen die Datentypen der Systeme aufeinander gemappt werden, um von den Systemen verarbeitet werden zu können. Für das Mapping der Datentypen ist auch das Schnittstellenmodul zuständig.

- Transaktionssicherheit

Transaktionen müssen immer komplett abgeschlossen werden. Tritt während einer Transaktion ein Fehler auf, müssen alle durch die Transaktion verursachte Änderungen rückgängig gemacht werden. Die Transaktionssicherheit wird von den Systemen selber übernommen. Die Schnittstellenkomponente muss gewährleisten, dass die Transaktionskonzepte der Systeme eingehalten werden und die Meldungen zwischen den Systemen weitergeleitet und im entsprechenden Format bereitgestellt werden.

- Administration

Um einen reibungslosen Ablauf sicherzustellen, müssen alle getätigten Aktionen festgehalten werden. So kann im Falle eines Fehlers dieser direkt nachvollzogen und behoben werden kann. Hierfür müssen Log-Files erstellt werden, in denen alle Aktivitäten, z.B. Herstellen der Verbinden, Senden von Aufrufen, Empfangen von Ergebnissen etc., mit der Angabe des Zeitpunktes protokolliert werden.

5.3.2. Lösungskonzept

Das Lösungskonzept bezieht sich auf die generische Schnittstellenkomponente, für den reibungslosen Ablauf von Geschäftsprozessen und den Austausch von Funktionalitäten zwischen den Systemen ermöglichen soll. Prinzipiell ist die Aufgabe der Schnittstellenkomponente, alle Aufrufe von vorhandenen Funktions- bzw. Prozedurmodulen weiterzu-

leiten, unabhängig von den fachlichen Anforderungen. Die Funktionen für die Abwicklung der Geschäftsprozesse werden im SAP- oder Oracle-System bereitgestellt. Hierauf wird im dritten Teil „Fachliches Implementierungsbeispiel“ näher eingegangen.

Grundsätzlich soll die Schnittstellenkomponente folgende zwei Funktionsschwerpunkte abbilden:

- Eine Oracle-Datenbank soll über die Schnittstellenkomponente einen beliebigen Funktionsbaustein bzw. BAPI in einem beliebigen SAP-System aufrufen können, wobei der Funktionsbaustein für die Oracle-Datenbank wie eine normale Stored Procedure aussieht.
- Ein SAP-System soll über die Schnittstellenkomponente eine beliebige Stored Procedure in einer beliebigen Oracle-Datenbank aufrufen, wobei die Stored Procedure für das SAP-System wie ein normaler Funktionsbaustein aussieht.

In Abbildung 13 wird die Funktionsweise der generischen Schnittstellenkomponente bildlich dargestellt.

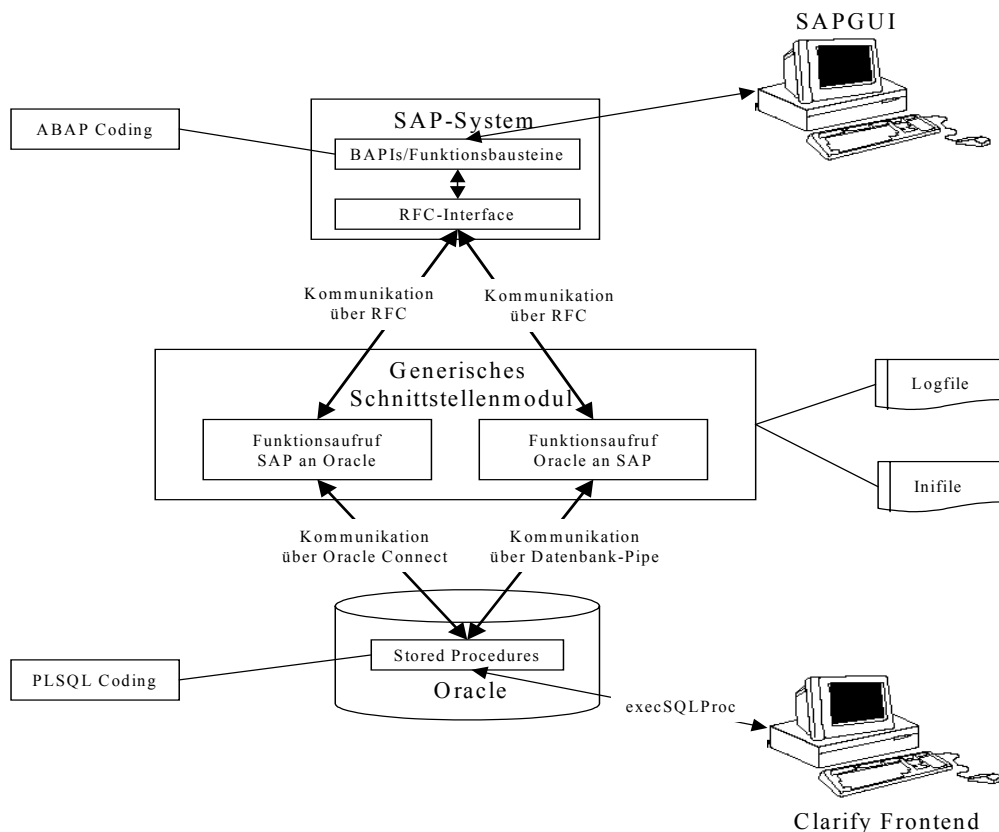


Abbildung 13: Generisches Schnittstellenmodul - Funktionsweise

Aufruf eines SAP-Funktionsbausteins durch Oracle

Für den Empfang und die Weiterleitung der Nachrichten von Oracle wird durch die Schnittstellenkomponente eine permanente Verbindung zwischen dem Oracle- und SAP-System aufgebaut. Die Verbindung mit SAP wird als Anmeldung eines RFC-Users realisiert, die Verbindung mit Oracle über eine Datenbank-Pipe.

Die Schnittstellenkomponente übernimmt hier die Rolle eines Clients, welcher an der geöffneten Datenbank-Pipe horcht und auf eine entsprechende Nachricht wartet. Das Versenden einer Nachricht wird von Clarify durch den Aufruf einer Oracle-Prozedur initiiert, wobei folgende Signatur verwendet wird:

```
MYSQLPROC.EXECUTEPROC
  "PROCNAME",
  INPUTLIST,
  RETURNVAL
```

Damit der Aufruf vom SAP-System verstanden wird, muss dieser als RFC-Aufruf ankommen:

```
CALL FUNCTION 'PROCNAME'
  DESTINATION 'SAPSYSTEM'
  EXPORTING
    RETVAL_1,
    RETVAL_n
  IMPORTING
    INPUT_1,
    INPUT_n
  TABLES
    ITAB_INPUT_RETVAL
```

Der Funktionsaufruf wird konvertiert und an die Datenbank-Pipe verschickt. Nach dem Versenden der Nachricht fällt die aktuelle Oracle-Session ihrerseits in den „Horchmodus“ und wartet auf eine Rückantwort, welche wieder in das Oracle-Format konvertiert wird und gegebenenfalls an die Benutzeroberfläche in Clarify weitergereicht wird. Falls in irgendeinem der beschriebenen Schritte ein Fehler auftritt, wird eine Exception an die aufrufende Stored Procedure hochgereicht. Tritt innerhalb der SAP-Verarbeitung eine

SAP-eigene Exception auf, wird die komplette Fehlerbeschreibung zum Aufrufer übergeben.

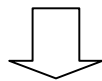
Sollte eines der beiden Systeme während der Laufzeit ausfallen oder die Netzwerkverbindung abbrechen, so wird eine konfigurierbare Anzahl von Versuchen lang ein Reconnect zu dem entsprechenden System versucht. Sämtliche Aktionen werden in einer Protokolldatei mitgeschrieben.

Aufruf einer Oracle Stored Procedure durch SAP

Anders als bei einem Aufruf aus Oracle muss hier keine direkte Verbindung zu den zu integrierenden Systemen aufgebaut werden. Die Schnittstellenkomponente übernimmt hier die Rolle eines Servers, der RFC-Aufrufe entgegennehmen kann.

In einem aufrufenden SAP-System müssen alle von diesem SAP-System nutzbaren Remote-Systeme vermerkt sein. Ähnlich wie ein echtes SAP-Remote-System wird die Schnittstellenkomponente hier bekannt gemacht. Bei einem Remote-Aufruf eines SAP-Funktionsbausteins wird immer das entsprechende Zielsystem angegeben. Die Schnittstellenkomponente fängt den RFC-Aufruf auf und wandelt ihn in Oracle-Format um:

```
CALL FUNCTION 'PROCNAME'  
  DESTINATION 'SAPSYSTEM'  
  IMPORTING  
    INPUT_1,  
    INPUT_n  
  EXPORTING  
    RETVAL_1,  
    RETVAL_n  
  TABLES  
    ITAB_INPUT_RETVAL
```



```
callSQLProcedure('ProcName',  
  aInputParam,  
  aOutputParam,  
  aSAPTableList,  
  aSAPElementList,  
  aSAPTableValues,  
  aSAPException);
```

Der Funktionsaufruf erfolgt über eine direkte Client-Verbindung zur Oracle-Datenbank. Die Rückgabewerte werden entgegengenommen und vom Oracle-Format in das SAP-Format umgewandelt und an das aufrufende SAP-System zurückgeschickt.

Falls in irgendeinem der beschriebenen Schritte ein Fehler auftritt, wird eine Exception im SAP-Format erzeugt und an das aufrufende SAP-System hochgereicht. Tritt innerhalb der aufzurufenden Stored Procedure eine Oracle-Exception auf, so wird die komplette Fehlerbeschreibung zum SAP-System übergeben. Auch hier werden alle getätigten Aktionen in einer Protokolldatei mitgeschrieben.

5.3.3. Realisierung

Für die Realisierung der generischen Schnittstellenkomponente wird das von SAP zur Verfügung gestellte RFC API (Remote Function Call Application Programming Interface) genutzt, mit dem Fremdsysteme auf Funktionen des SAP-Systems zugreifen können. Das RFC API wird in Form von C-Routinen bereitgestellt, die als RFC library auf verschiedenen externen Plattformen, wie Windows, Unix und OS/2 eingefügt werden können, um die Nutzung von RFC-Funktionalität zwischen dem SAP-System und C-Programmen den entsprechenden Plattformen zu gewährleisten. Für jede unterstützte Plattform wird ein RFC SDK (Software Development Kit) bereitgestellt, welches sich aus folgenden Dateien zusammensetzt:

- saprfc.h – enthält die Header der RFC-Funktionen zum Zugriff auf das SAP-System und die Deklaration der benötigten Datentypen
- sapitab.h – enthält die Header der Funktionen, welche zum Manipulieren interner Tabellen in SAP verwendet werden
- librfc – plattformspezifische Bibliothek

Um die Funktionalität des RFC APIs nutzen zu können wird das Schnittstellenmodul, welches die Kommunikation zwischen SAP und Oracle gewährleisten soll, als C-Programm realisiert. Im nachfolgenden Text wird das Programm SAPORAD (SAP ORAcle Daemon) genannt.

Der SAPORAD kann als Hintergrundprozess (Daemon) gestartet werden, wobei entweder eine Client-Instanz oder eine Server-Instanz erzeugt werden kann. Der Client wird verwendet, um die von Oracle kommenden Aufrufe an das SAP-System zu versenden. Der Server nimmt Aufrufe des SAP-Systems entgegen und leitet sie an Oracle weiter. Der SAPORAD besteht aus folgenden Komponenten:

Tabelle 1: Komponenten des SAPORAD

Datei	beinhaltet Funktionen für:
Sapora_rfc.h / sapora_rfc.c	Abwicklung der SAP-seitigen Kommunikation über das RFC APIs
Sapora_oraapi.h / sapora_oraapi.c	Abwicklung der Kommunikation mit Oracle
Sapora_trap.h / sapora_trap.c	Protokollierung der durchgeführten Aktivitäten
Sapora_ini.h / sapora_ini.c	Nutzung einer Initialisierungs-Datei
Sapora_signal.h / sapora_signal.c	Abwicklung von Benutzer-Signalen
saporad.h / saporad.c	Start des SAPORADs als Client oder als Server und Ablaufregelung der Kommunikation
saporad.sql	Abwicklung der Kommunikation mit der Schnittstellenkomponente

Für das Versenden und das Empfangen von Aufrufen müssen auch auf Oracle-Seite Funktionen bereitgestellt werden. Diese werden in dem SQL-Package saporad.sql zusammengefasst und in der Sprache PL/SQL realisiert.

Der SAPORAD verwendet für die Kommunikation mit dem SAP-System folgende Funktionen des RFC APIs:

- beim Aufruf als Client:

- RfcOpen - baut eine RFC-Verbindung zum SAP-System auf
- RfcInstallStructure - definiert die Struktur für Tabellen-Parameter
- RfcCall - ruft einen Funktionsbaustein im SAP-System auf
- RfcReceive - erhält die Rückgabewerte des aufgerufenen Funktionsbausteins
- RfcClose - beendet die RFC-Verbindung zum SAP-System

- beim Aufruf als Server:

- RfcAcceptExt - wird genutzt, um sich am SAP-System zu registrieren und eingehende Aufrufe des SAP-Systems zu akzeptieren, bzw. entgegenzunehmen
- RfcGetName - liefert den Namen des Funktionsbausteins eines eingehenden Aufrufs
- RfcGetData - liefert die zugehörigen Parameter zu dem aufgerufenen Funktionsbausteins
- RfcSendData - sendet Rückgabewerte zurück an das SAP-System

Im Folgenden werden die Funktionen der einzelnen Komponenten des SAPORADs beschrieben.

sapora_rfc

Der SAPORAD verwendet sapora_rfc für die Kommunikation mit dem SAP-System über das RFC API. Dazu gehört der Auf- und Abbau der Kommunikationsverbindung und das Versenden bzw. Empfangen von Funktionsaufrufen und deren Ergebnissen, wobei hierfür die benötigten Parameter in dem vom RFC API vorgegebenen Format bereitgestellt werden müssen. Die wichtigsten Funktionen des sapora_rfc werden nachfolgend beschrieben:

- connect2SAP

Die Funktion connect2SAP stellt mit Hilfe der Funktion RfcOpen des RFC APIs eine Verbindung zum SAP-System her. Für die Herstellung der Verbindung muss die Adresse des Ziel-Systems und die für die Anmeldung benötigten Daten angegeben werden. Diese werden anhand einer Initialisierungsdatei als Übergabeparameter an die Funktion connect2SAP übergeben. Es werden folgende Werte aus der Initialisierungsdatei benötigt:

- HOSTNAME - Hostname des Zielsystems
- SYSNR - Nummer des Zielsystems
- DESTINATION - Name des Empfängers
- CLIENT - Name des Clients
- SAPUSER - Name des Benutzers

- SAPUSERPWD – Passwort des Benutzers
- LANGUAGE – Sprache mit der sich am System angemeldet wird
- TRACE – Parameter, der festlegt, ob protokolliert werden soll

Diese Werte werden der Datei entnommen und an die Funktion RfcOpen übergeben, welche die Verbindung zum SAP-System herstellt.

- callFunctionModul

Die Funktion callFunctionModul benutzt RfcCall (RFC APIs) um einen Funktionsbaustein im SAP-System aufzurufen. Für den Aufruf benötigt RfcCall den Namen des Funktionsbausteins und die von dem Funktionsbaustein erwarteten Eingabeparameter. Dieses können sowohl Einzelwerte z.B. eine Kundennummer, als auch Tabellen sein z.B. die Positionen eines Auftrages. Damit die Parameter in SAP richtig konvertiert werden, müssen sie in den im RFC API vordefinierten Datentypen übergeben werden. Es handelt sich dabei um Arrays, in denen zusätzlich zu den Werten der Parameter auch Information für die Konvertierung abgelegt werden können. Einzelwerte werden als RFC_PARAMETER übergeben, welcher folgende Bestandteile enthält:

- | | | |
|------|---|---|
| name | - | Bezeichnung des Parameters |
| nlen | - | Länge des Parameternamens |
| type | - | Datentyp des Parameters (im RFC API vorgegebene Datentypen) |
| leng | - | Feldlänge in Bytes |
| addr | - | Wert des Parameters |

Tabellen-Parameter müssen dem Datentypen RFC_TABLE entsprechen, welcher wie folgt aufgebaut ist:

- | | | |
|----------|---|--|
| name | - | Name der Tabelle |
| nlen | - | Länge des Tabellennamens |
| type | - | Datentyp der Tabellenzeilen – wird von InstallStructure zurückgeliefert |
| leng | - | Zeilenlänge in Bytes |
| ithandle | - | Adresse der Tabellenstruktur in SAP; Interne Tabelle muss mit ItCreate (sapitab.h) angelegt werden; ithandle ist der Rückgabewert von ItCreate |
| itmode | - | Art des Empfangs der Tabelle 'call by reference' oder 'call by value' (wird nur bei RfcGetData verwendet) |
| newitab | - | interne Verwendung in SAP |

Bei Verwendung von Tabellen muss vor dem Aufruf von RfcCall die Struktur der Tabellen im SAP-System bekannt gemacht werden. Dies geschieht mit der Funktion InstallStructure des RFC APIs. Dabei wird der Name der Tabelle, die Liste der Elemente und die Anzahl der Elemente übergeben. Die Elementliste beinhaltet zu jedem Element Informationen entsprechend des Datentyps RFC_TYPE_ELEMENT:

name	-	Bezeichnung des Elementes
type	-	Datentyp des Elements(im RFC API vorgegebene Datentypen)
leng	-	Feldlänge in Bytes
decimals	-	Dezimalzahlen
value	-	Wert des Elementes

Nach Aufruf von RfcCall werden mit RfcReceive die Rückgabewerte des in SAP aufgerufenen Funktionsmoduls entgegengenommen. Rückgabewerte können auch Einzelwerte oder Tabellen sein. Sie entsprechen auch den Datentypen RFC_PARAMETER und RFC_TABLE. Die Tabellen-Parameter müssen denen von RfcCall entsprechen. Die erhaltenen Rückgabewerte werden mit Hilfe der Funktionen packChar2Msg, packInt2Msg und packDate2Msg (saporad_oraapi) an Oracle geschickt.

- packInitMsg, packExitMsg, packOutputMsg, packTableMsg
Die genannten Funktionen übertragen die von SAP erhaltenen Rückgabewerte zurück an Oracle. Mit packInitMsg und packExitMsg wird der Anfang und das Ende einer Übertragung an Oracle gemeldet und mit packOutputMsg und packTableMsg werden die Einzel- und Tabellen-Parameter gesendet. Die Übertragung erfolgt mit Hilfe der Funktionen packChar2Msg, packInt2Msg und packDate2Msg aus saporad_oraapi, mit denen die einzelnen Werte in Oracle-Datentypen umgewandelt und an die Datenbank-Pipe übergeben werden.
- doRfc
Mit der Funktion doRfc werden Aufrufe von Oracle empfangen. Die einzelnen Werte werden mit Hilfe der Funktionen unpackMsg2int und unpackMsg2char aus saporad_oraapi von der Datenbankpipe entgegengenommen. Für die Weiterleitung des Auf-

rufs an SAP wird die Funktion callFunctionModul aufgerufen, die auch gleichzeitig die empfangenen Rückgabewerte zurück an Oracle sendet.

- disconnectFromSap

Die Funktion disconnectFromSap beendet durch Aufruf der Funktion RfcClose (RFC API) die Verbindung zum SAP-System.

- getParamDef

Die Funktion getParamDef erhält als Übergabeparameter eine Datei, welche Informationen über die Parameter eines aus SAP kommenden Funktionsaufrufs enthält. Die Angaben zu den Parametern werden ausgelesen und es werden für die Import-, Export- und Tabellen-Parameter Arrays entsprechend den Datentypen RFC_PARAMETER und RFC_TABLE erzeugt. Die Adresse dieser Arrays wird als Rückgabewert zurückgeliefert, um die Parameter mit den Werten aus SAP belegen zu können.

- ProcessRfc

ProcessRfc wird für den Aufruf von Procedures in Oracle verwendet. Die Funktion erhält als Eingabeparameter den Namen der Procedure, die in Oracle aufgerufen werden soll und eine Initialisierungsdatei, welche Informationen darüber enthält, welche Parameter für den Aufruf benötigt werden. Diese werden mit der Funktion getParamDef aus der Datei entnommen und mit RfcGetData (RFC API) werden die Werte der Import-Parameter aus SAP geholt. Der Aufruf der Procedure in Oracle erfolgt mit callSQLProcedure (saporad_oraapi). Die Rückgabewerte werden mit receiveSqlScalarResults und receiveSqlTableResults (saporad_oraapi) empfangen und mit RfcSendData (RFC API) an SAP übertragen.

saporad_oraapi

Mit saporad_oraapi wird die Kommunikation mit Oracle abgewickelt. Diese wird über eine Datenbankpipe durchgeführt. Hierfür wird von Oracle das Package DBMS_PIPE zur Verfügung gestellt. Es ermöglicht zwei oder mehreren Sessions miteinander zu kommu-

nizieren. Die wichtigsten Funktionen des DBMS_PIPE-Packages, welche für die Kommunikation zwischen dem SAPORAD und Oracle verwendet werden sind:

- PACK_MESSAGE - bildet eine Nachricht in einem lokalen Puffer
- SEND_MESSAGE - versendet die Nachricht an die Datenbankpipe; diese wird erzeugt, falls noch nicht vorhanden
- RECEIVE_MESSAGE - holt die Nachricht aus der Datenbankpipe und legt sie in einen lokalen Puffer ab
- NEXT_ITEM_TYPE - liefert den Datentyp des nächsten Wertes im Puffer
- UNPACK_MESSAGE - holt die einzelnen Werte aus dem Puffer
- PURGE - leert den Inhalt der Datenbankpipe
- RESET_BUFFER - leert den Inhalt des Puffers

Der SAPORAD verwendet Embedded SQL, um auf die Funktionen des DBMS_PIPE-Packages zuzugreifen. Embedded SQL sind SQL-Anweisungen, die in C-Programme eingefügt werden. Für die Übersetzung wird der Precompiler proc von Oracle benötigt, welcher die SQL-Anweisungen in C-Sprache umwandelt.

Im Folgenden werden die wichtigsten Funktionen des `saporad_oraapi` näher beschrieben.

- connect2Oracle

Mit `connect2Oracle` wird die Verbindung zu der Datenbank aufgebaut. Für den Aufbau der Verbindung werden folgende Parameter benötigt, die der Initialisierungsdatei entnommen werden:

- | | | |
|-----------|---|---------------------------|
| DBSERVER | - | Name der Datenbankinstanz |
| DBUSER | - | Name des Benutzers |
| DBUSERPWD | - | Passwort des Benutzers |

Die Herstellung der Verbindung erfolgt mit dem Embedded-SQL-Befehl :

```
EXEC SQL CONNECT <DBUSER>/<DBUSERPWD>@<DBSERVER>
```

Zur Verdeutlichung der Verwendung von Embedded SQL wird hier als Beispiel die Funktion `connect2Oracle` näher dargestellt:

```

void connect2Oracle(IniFile* pIniFile, db_error_t *db_error)
{
    /* auftretende Variablen im Deklarationsblock bekannt machen */

    EXEC SQL BEGIN DECLARE SECTION;
        char uid[CONNECT_STRING_SIZE];
    EXEC SQL END DECLARE SECTION;

    char dbserver[DBSERVER_SIZE];
    char dbuser[DBUSER_SIZE];
    char dbuserpwd[DBUSERPWD_SIZE];

    /* benötigte Parameter werden aus dem IniFile entnommen und zu dem
       Connect-String uid zusammengesetzt: DBUSER/DBUSERPWD@DBSERVER */

    uid[0] = 0;

    assignIniValue(pIniFile, "DBUSER", dbuser, DBUSER_SIZE);
    strncat(uid, dbuser, CONNECT_STRING_SIZE);
    strncat(uid, "/", CONNECT_STRING_SIZE);

    assignIniValue(pIniFile, "DBUSERPWD", dbuserpwd, DBUSERPWD_SIZE);
    strncat(uid, dbuserpwd, CONNECT_STRING_SIZE);
    strncat(uid, "@", CONNECT_STRING_SIZE);

    assignIniValue(pIniFile, "DBSERVER", dbserver, DBSERVER_SIZE);
    strncat(uid, dbserver, CONNECT_STRING_SIZE);

    /* Aufruf des Befehls CONNECT mit Übergabe des Strings uid */

    EXEC SQL CONNECT :uid;

    return;

    /* Fehlerbehandlung */

    sql_error_label:
        sql_error(ERR_ROLLBACK, db_error);
}

```

Abbildung 14: Embedded-SQL Beispiel

Jede Embedded-SQL-Anweisung beginnt mit dem Präfix „EXEC SQL“. Die in den Anweisungen verwendeten C-Variablen müssen dem Precompiler proc im Deklarationssteil bekannt gemacht werden. Die Typumwandlung erfolgt automatisch. Die Übergabe von Daten an SQL erfolgt mit **:uid**, wobei der Wert der C-Variablen uid übergeben wird. Nach jeder SQL-Anweisung sollte eine Fehlerabfrage erfolgen. In `sapora_oraapi` wurde die `WHENEVER`-Anweisung verwendet, mit der nach jeder SQL-Anweisung eine Fehlerabfrage gemacht wird:

```
EXEC SQL WHENEVER SQLERROR GOTO sql_error_label
```

Im Falle eines Fehlers springt der Compiler zu der Textmarke `sql_error_label` und ruft die Funktion `sql_error` auf, welche die Transaktion rückgängig macht und die Fehlermeldung an die Variablen `db_error` übergibt.

- RetryOracleConnect

`RetryOracleConnect` stellt durch Aufruf der Funktion `connect2Oracle` die Verbindung zur Datenbank her. Bei Fehlschlagen des Verbindungsaufbaus wird der Versuch mehrere Male wiederholt. Die Anzahl der Versuche und die Zeit zwischen den Versuchen für einen erneuten Verbindungsaufbau werden anhand von Parametern in der Initialisierungsdatei festgelegt.

- resetPipe

Mit der Funktion `resetPipe` wird im Falle eines Fehlers der Inhalt der Datenbankpipe gelöscht, damit nach Beheben des Fehlers neu gestartet werden kann. Hierfür werden die Funktionen `DBMS_PIPE.PURGE` und `DBMS_PIPE.RESET_BUFFER` verwendet.

- receiveMsg

`receiveMsg` ruft die Funktion `RECEIVE_MESSAGE` des `DBMS_PIPE`-Packages auf, um eine Nachricht von einer Datenbankpipe zu empfangen. Nach dem Empfang der Nachricht wird diese aus der Datenbankpipe gelöscht, jede Nachricht kann also nur einmal abgerufen werden. Um Zugang zu den einzelnen Werten der Nachricht zu erhalten müssen mehrere Aufrufe der Funktion `UNPACK_MESSAGE` erfolgen.

- unpackMsg2char, unpackMsg2int

Je nachdem, in welchen Datentyp der ankommende Wert umgewandelt werden soll, werden die Funktionen `unpackMsg2char` oder `unpackMsg2int` aufgerufen, welche mit `DBMS_PIPE.UNPACK_MESSAGE` die einzelnen Wert der empfangenen Nachricht entnehmen und die Datentypumwandlung durchführen. Mit `unpackMsg2char` werden die Werte in Strings umgewandelt und mit `unpackMsg2int` in Integer-Werte.

- packChar2Msg, packInt2Msg, packDate2Msg

Mit den genannten Funktionen werden einzelne Werte durch Aufruf von DBMS_PIPE.PACK_MESSAGE zu einer Nachricht zusammengefügt und in einen lokalen Puffer abgelegt. Die Werte werden vorher in die von Oracle erwarteten Datentypen umgewandelt:

- VARCHAR – String (Zeichenkette)
- INT – numerischer Wert
- DATE – Datums-Wert

Entsprechend den erwarteten Datentypen wird bei VARCHAR die Funktion packChar2Msg, bei INT die Funktion packInt2Msg und bei DATE die Funktion packDate2Msg aufgerufen. Bei DBMS_PIPE.PACK_MESSAGE handelt es sich um eine überladene Funktion, welche Parameter in allen drei genannten Datentypen erhalten kann.

- sendMsg

Mit der Funktion sendMsg wird der lokale Zwischenspeicher, an den die einzelnen Elemente mit DBMS_PIPE.PACK_MESSAGE übergeben wurden, zu einer Nachricht zusammengefasst und über Datenbankpipe versendet. Dabei muss darauf geachtet werden, dass die von Oracle vorgegebene Größe einer Nachricht nicht überschritten wird. Das Versenden der Nachricht erfolgt mit DBMS_PIPE.SEND_MESSAGE.

- callSQLProcedure

Die Funktion callSQLProcedure wird verwendet, wenn der SAPORAD als Server fungiert und Funktionsaufrufe von SAP entgegennimmt, um diese an Oracle weiterleitet. Die Weiterleitung an Oracle erfolgt nicht über die Datenbankpipe, sondern als Client-Verbindung über Embedded SQL. Die Funktion callSqlProcedure erhält als Parameter den Namen der aufzurufenden Funktion und die benötigten Eingabeparameter als RFC_PARAMETER-Array. Über Embedded SQL wird der Inhalt der Arrays in globalen Rekords auf Oracle-Seite übertragen. Mit der Procedure saporad.callPlsqlProcedure werden die Werte aus den Rekords entnommen und der Funktionsaufruf wird durchgeführt..

- receiveSQLScalarResults, receiveSQLTableResults

Die genannten Funktionen werden für den Empfang der Rückgabewerte aus Oracle verwendet. Die Rückgabewerte werden auf Oracle-Seite genau wie die Eingabe-Parameter in globalen Records gespeichert. Über Embedded SQL wird der Inhalt der Records in die RFC-Parametertypen RFC_PARAMETER und RFC_TABLE übertragen.

- disconnectFromOracle

Die Funktion beendet die Verbindung zu Oracle durch Ausführen des Befehls:
EXEC SQL COMMIT RELEASE;

sapora_trap

Die Komponente sapora_trap wird für die Protokollierung der getätigten Aktivitäten verwendet. Die wichtigste Funktion ist die Funktion log_trap. Sie wird nach jeder Aktion, die protokolliert werden soll, und auch beim Auftreten eines Fehlers aufgerufen und erhält als Eingabe-Parameter die Beschreibung des Fehlers oder der getätigten Aktion. Die Meldung wird dann mit Angabe des Zeitpunkts als Notiz, Warnung oder Fehler in die Protokoll-Datei geschrieben.

sapora_inifile

sapora_inifile enthält die Funktionen für die Benutzung einer Initialisierungsdatei. Mit dem SAPORAD können mehrere Instanzen erzeugt werden, so dass mehrere Kommunikationsverbindungen aufgebaut werden können. Die Informationen zu den Systemen, zwischen denen die Verbindung hergestellt werden soll, wie Name des Systems, IP-Adresse, Anmelde-Parameter etc, werden in einer Initialisierungs-Datei gespeichert. Die Datei wird beim Erzeugen einer Instanz übergeben, damit die für den Verbindungsaufbau benötigten Werte entnommen werden können. Dies wird mit folgenden Funktionen durchgeführt:

- installInifile

Mit `installInifile` werden die Daten aus der Initialisierungs-Datei gelesen und als Liste in C abgelegt. Die Liste enthält den Namen des Parameters und den Wert.

- deinstallInifile

Mit der Funktion `deinstallInifile` wird die Liste, die aus den Daten der Initialisierungsdatei erzeugt wurde, wieder gelöscht.

- assignIniValue

Die Funktion `assignIniValue` sucht anhand des Parameternamens einen Wert in der Liste und weist ihn einer Variablen zu.

sapora_signals

`sapora_signals` nutzt das Unix-Signal-Handling, um Benutzer-Eingaben an das laufende Programm weiterzuleiten. Das kann z. B. der Abbruch des Programms sein.

saporad

Die Komponente `saporad` stellt das eigentliche Programm dar. Es wird mit folgendem Befehl gestartet:

```
saporad [-s|-c] [-d] [-i inifile] [-f logfile].
```

Mit den Optionen kann folgendes bestimmt werden:

- s der SAPORAD soll als Server gestartet werden
- c der SAPORAD soll als Client gestartet werden
- d der SAPORAD soll als Daemon (Hintergrundprozess) laufen
- i inifile die Datei inifile soll als Initialisierungsdatei genommen werden. Bei Weglassen dieser Option wird `saporad.ini` als Default-Initialisierungsdatei genommen
- f logfile die Datei logfile soll als Protokoll-Datei genommen werden. Bei Weglassen dieser Option wird `saporad.log` als Default-Protokoll-Datei genommen.

Der Ablauf beim Aufruf der benötigten Funktionen sieht, je nachdem ob Programm als Client oder als Server gestartet wird, wie folgt aus:

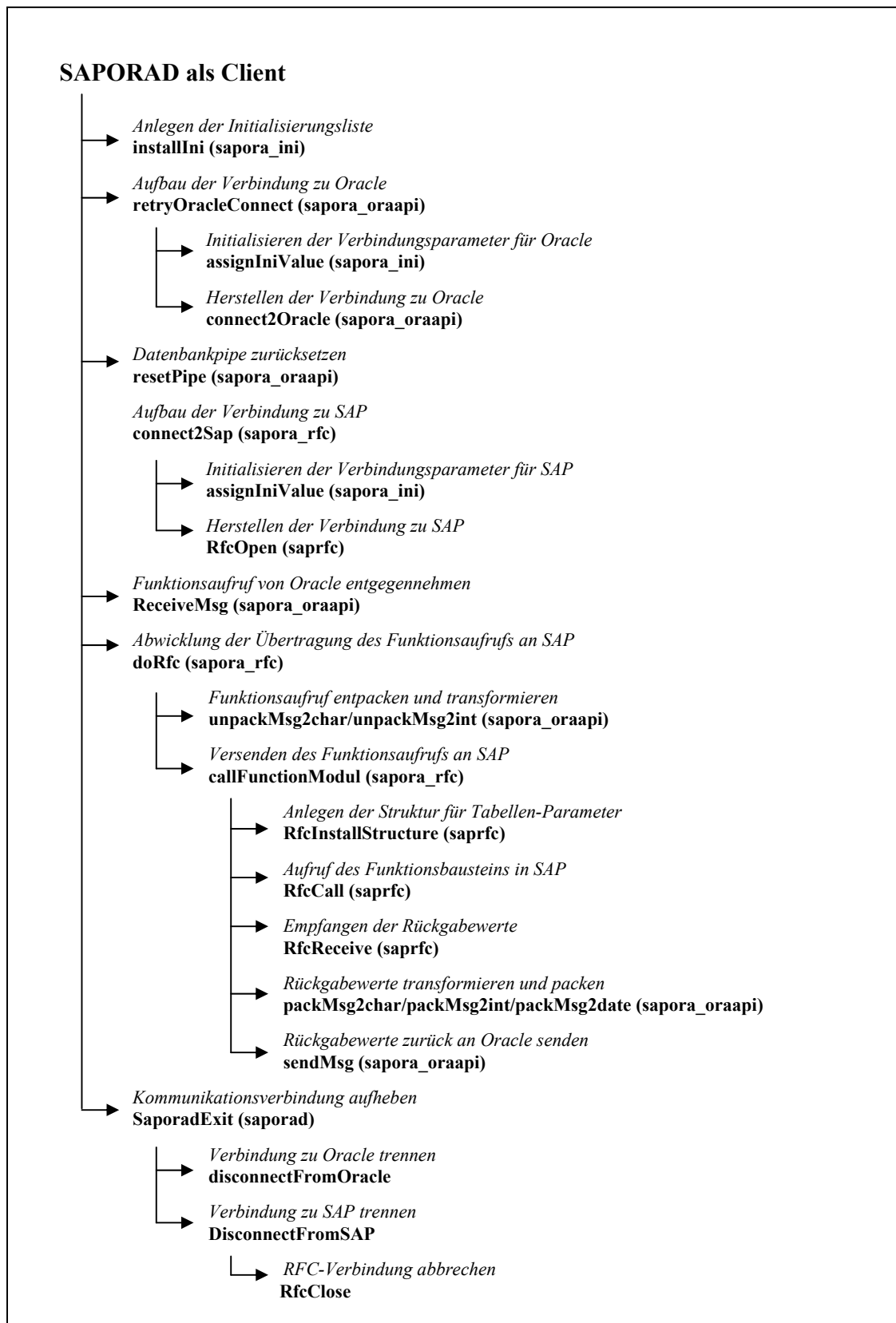


Abbildung 15: Funktionsablauf bei einer Client-Verbindung

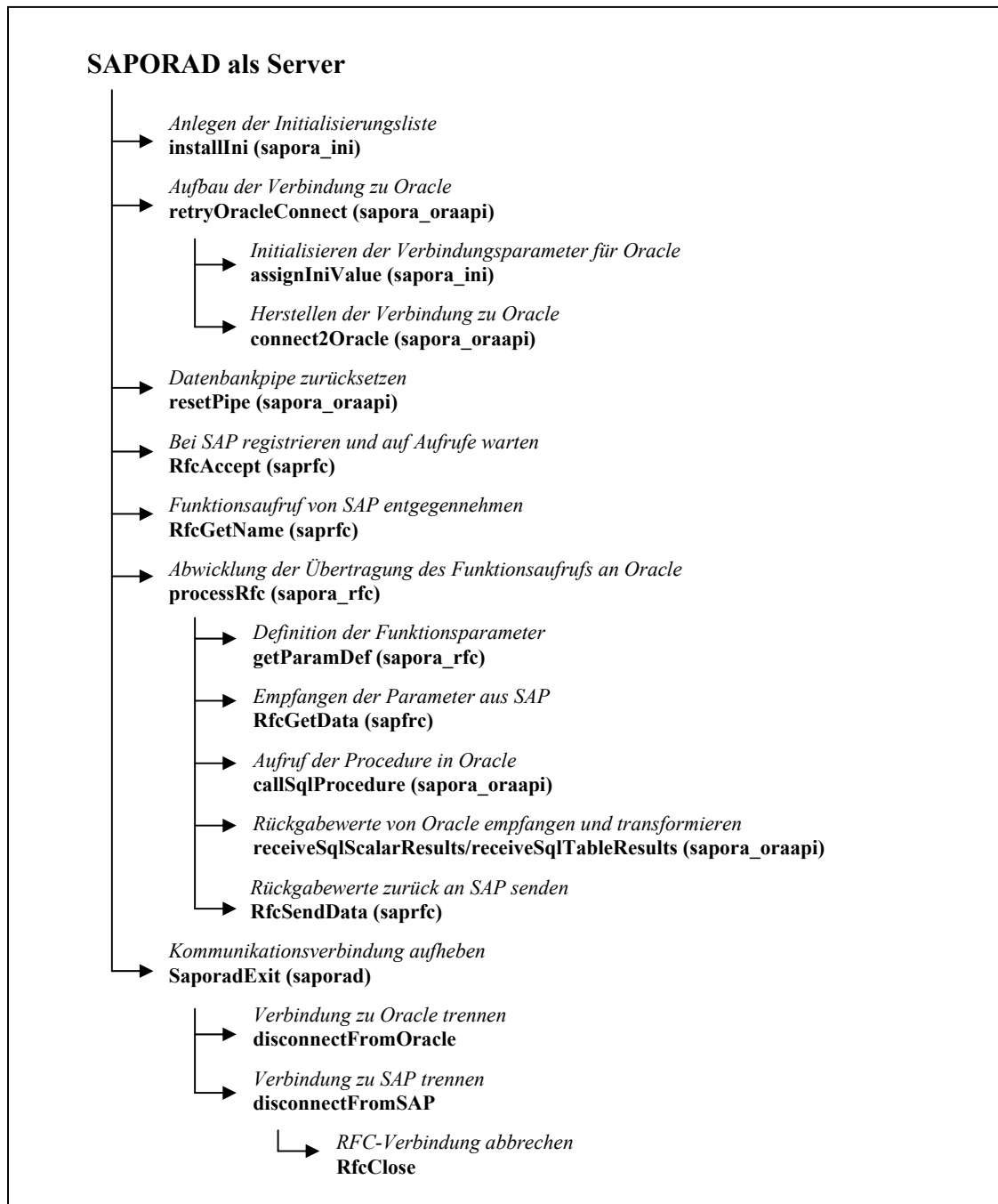


Abbildung 16: Funktionsablauf bei einer Server-Verbindung

Der SAPORAD gewährleistet eine permanente Verbindung zwischen SAP und Oracle und bietet deswegen die Option, als Hintergrundprozess zu laufen. Funktionsaufrufe können jederzeit zwischen den Systemen ausgetauscht werden. Die Funktion SaporadExit

zum Aufheben der Verbindung wird nur im Falle eines Fehlers oder durch explizite Benutzeraufforderung aufgerufen.

saporad.sql

Die Komponente saporad.sql ist ein PL/SQL-Package und enthält Funktionen für das Versenden und Empfangen von Funktionsaufrufen und deren Ergebnisse auf Oracle-Seite. Die beiden Hauptfunktionen callFunctionModul und callPISqlProcedure, welche diese Aufgaben übernehmen, sind nachfolgend beschrieben.

- CallFunctionModul

Die Procedure callFunctionModul wird für das Versenden von Funktionsaufrufen an SAP verwendet. Die Daten für die Funktionsaufrufe werden an die Datenbankpipe übergeben und von der Komponente saporad_oraapi empfangen. Die von SAP zurückgelieferten Rückgabewerte werden von saporad_oraapi an dieselbe Datenbankpipe übergeben und können dann auf Oracle-Seite wieder entnommen werden. Um eine Übereinstimmung beim Versand und Empfang zu gewährleisten, wurde ein bestimmtes Format für die Ausgangs- bzw. Eingangsnachrichten festgelegt:

- Ausgangsnachricht:
 - Typ der Nachricht (Funktionsaufruf)
 - Name der Datenbankpipe (eindeutige Session)
 - Name des Funktionsbausteins
 - Parameterliste:
 - Import-Parameter
 - Export-Parameter
 - Tabellen-Parameter
 - Exceptions
- Eingangsnachricht:
 - Parameterliste:
 - Export-Parameter
 - Tabellen-Parameter
 - Exceptions

Die Parameter müssen den RFC-Datentypen RFC_PARAMETER, RFC_TABLE und RFC_TYPE_ELEMENT entsprechen und alle Informationen enthalten um in

SAP transformiert werden zu können. Hierfür wurden entsprechend die globalen Record-Typen `t_SAPParamTable`, `t_SAPTableList` und `t_SAPElementList` definiert.

Für jeden Funktionsbaustein der in SAP aufgerufen werden soll, muss in Oracle eine Procedure vorhanden sein, welche die erwarteten Parameter mit den benötigten Informationen belegt und an `callFunctionModul` übergibt. Die Rückgabewerte werden dann von `callFunctionModul` an diese Prozedur zurückgeliefert. Für das Extrahieren der erhaltenen Rückgabewerte werden von `saporad.rfc` folgende Funktionen bereitgestellt:

- `assignSap2ora` - liefert anhand des Parameterfeldes den entsprechenden Wert aus der Export-Parameter-Liste
 - `convSap2ora` - wandelt ABAP-Datentypen in Oracle-Datentypen um
 - `parseTableField` - liefert anhand des Feldnamens den Wert aus der Tabellen-Parameter-Liste
-
- `callPLSQLProcedure`

Die Funktion `callPLSQLProcedure` wird über Embedded SQL aufgerufen, um bestimmte Procedure-Aufrufe in Oracle zu tätigen. Sie erhält als Eingabe-Parameter den Namen der aufzurufenden Procedure. Die für den Aufruf der Procedure benötigten Parameter werden auch über Embedded SQL in einem globalen Rekord gespeichert, welcher sich aus dem Namen, dem Datentyp und dem Wert des Parameters zusammensetzt. Die Werte werden dem Rekord entnommen, in Oracle-Datentypen umgewandelt und beim Aufruf der Procedure übergeben. Die Rückgabewerte der Procedure müssen in SAP-Datentypen transformiert werden und wiederum in globalen Records abgelegt werden, von wo sie über Embedded SQL entnommen werden.

6. Fachliches Implementierungsbeispiel

Für die in Kapitel 5 erarbeitete und entwickelte Integrationslösung, wird in diesem Kapitel als Machbarkeitsbeleg die prototypische Realisierung eines Geschäftsprozesses durchgeführt. Hierfür wird als Beispiel der Geschäftsprozess der „Auftragsverarbeitung“ ausgewählt, welcher alle Funktionen von der Auftragserfassung über das CRM-System Clarify bis zur Auftragsdurchführung im ERP-System SAP beinhaltet. Folgende Grafik zeigt, welche Funktionen in diesem Beispiel von den beiden Systemen übernommen werden:

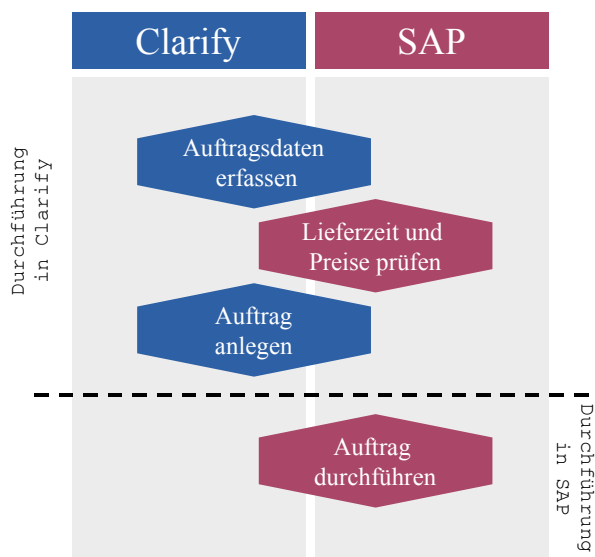


Abbildung 17: Geschäftsprozess „Auftragsverarbeitung“

Der Geschäftsprozess wird dabei in zwei Teile aufgeteilt:

- im ersten Teil erfolgt die Erfassung der Daten bis hin zum Anlegen des Auftrags über das Call Center Modul vom Clarify
- der zweite Teil, die Durchführung de Auftrages bzw. die Auslieferung an den Kunden, wird über das Logistik Modul von SAP vollzogen.

Der Zugriff auf bestimmte Funktionen des anderen Systems und die automatische Datenübertragung müssen für die durchgängige Abwicklung der beiden Teile des Prozesses gewährleistet sein.

Zur Durchführung des ersten Teils müssen folgende Funktionen des SAP-Systems in Clarify verfügbar sein:

- das Feststellen der Preise und Verfügbarkeit der bestellten Produkte um dem Kunden eine direkte Auskunft über die Machbarkeit des Auftrages erteilen zu können
- das Übertragen der Auftragsdaten und das Anlegen des Auftrages im SAP-System, um eine Durchführung zu gewährleisten.

Im zweiten Teil des Prozesses muss nach der Durchführung des Auftrages in SAP eine entsprechende Meldung an das Clarify System gesendet werden, um bei Kundenanfragen Auskunft über Auftragsstatus erteilen zu können.

Im Folgenden wird die Realisierung der beiden Teile des Geschäftsprozesses „Auftragsverarbeitung“ dargestellt.

6.1. Teil 1 – Durchführung in Clarify

Die Realisierung des ersten Teils wird in folgenden Schritten durchgeführt:

1. Identifikation der BAPIs in SAP und gegebenenfalls Anpassung an die von Clarify benötigten Funktionalitäten
2. Durchführung der BAPI-Aufrufe auf Oracle-Seite
3. Anpassung der Benutzerschnittstelle in Clarify für den Zugriff auf die SAP-Funktionen

Die einzelnen Aufgaben werden nachfolgend detailliert beschrieben.

6.1.1. Identifikation der BAPIs in SAP

BAPIs (Business Application Programming Interfaces) sind offene, betriebswirtschaftlich orientierte Schnittstellen für den externen Zugriff auf die Geschäftsprozesse und Daten des R/3-Systems. BAPIs werden als Methoden auf SAP-Business-Objekte definiert. Im BAPI Browser werden alle SAP-Business-Objekte und deren BAPIs dargestellt. Alle Business-Objekte, für die BAPIs implementiert wurden, werden in der R/3-Anwendungshierarchie angezeigt (siehe Abbildung 18) und die für den BAPI-Aufruf benötigten Schnittstelleninformationen sowie die Dokumentation wird bereitgestellt.

Nachdem ein BAPI von SAP eingeführt und freigegeben wurde, bleiben seine Schnittstellendefinition und Parameter langfristig stabil. Jede Weiterentwicklung an einem freigegebenen BAPI wird syntaktisch und inhaltlich abwärtskompatibel vorgenommen. Dies bedeutet, dass Anwendungen, die auf BAPIs aus einem bestimmten R/3-Release programmiert wurden, durch Änderungen an diesen BAPIs in einem Nachfolge-Release nicht beeinflusst werden.

Jedes BAPI, das eine Instanz eines Objekts erstellt oder die Daten eines Objekts verändert, hinterlässt einen konsistenten Datenbankzustand. Alle Datenbankänderungen werden vollständig oder überhaupt nicht ausgeführt.

Für die Entwicklung neuer BAPIs steht der Business Object Builder zur Verfügung. Zusätzlich können mit dem Funktion Builder objekt-unabhängige Funktionsbausteine entwickelt werden.

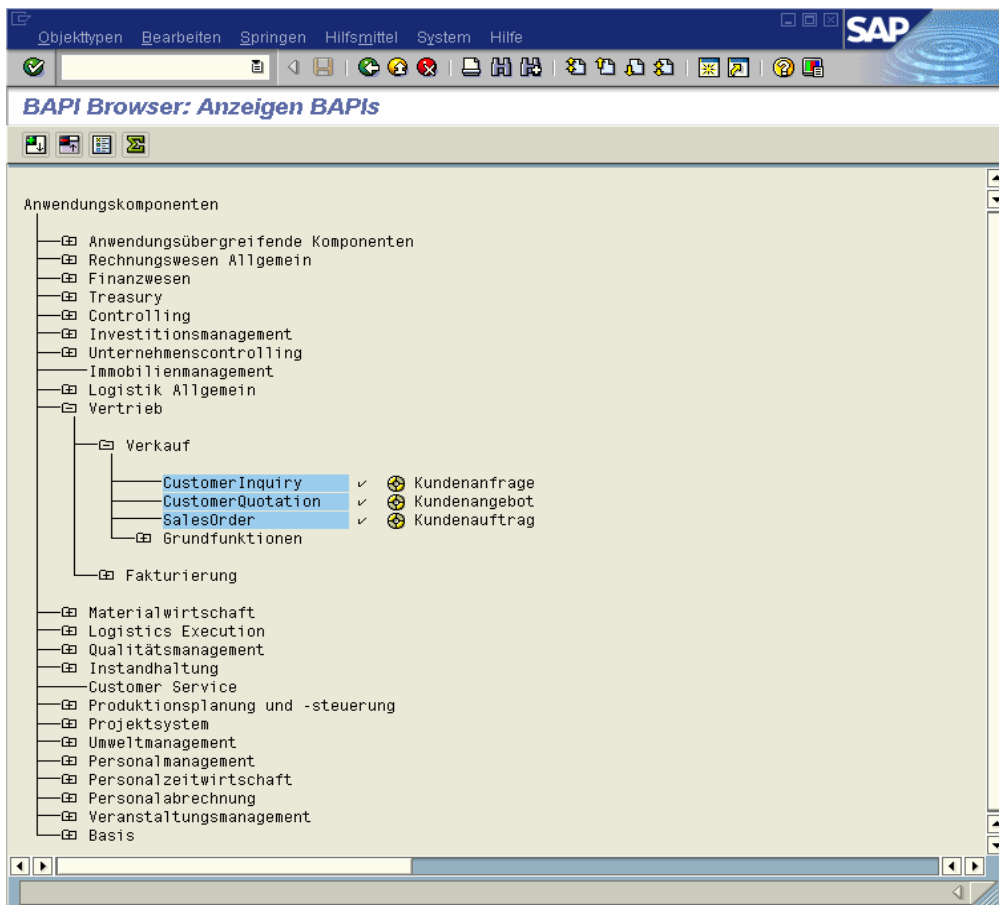


Abbildung 18: SAP BAPI Browser

Für den Geschäftsprozess „Auftragsbearbeitung“ werden BAPIs des Objekts „SalesOrder“ verwendet. Für die von Clarify benötigten Funktionalitäten werden dabei folgende BAPIs dieses Objektes ausgewählt:

- BAPI_SALESORDER_CREATEFROMDAT1

Mit diesem Bapi werden in SAP neue Kundenaufträge angelegt. In Tabelle 2 werden die Parameter der Methode dargestellt.

Tabelle 2: Parameter der Funktion BAPI_SALESORDER_CREATEFROMDAT1

Parameter-Typ	Parameterbezeichnung	Typreferenz	Parameterbeschreibung
Import-Parameter	ORDER_HEADER_IN	Obligatorisch	enthält Daten des Auftragskopfes
	WITHOUT_COMMIT	Optional	steuert das Commit Work
Export-Parameter	SALESDOCUMENT	Optional	liefert die Auftragsnummer
	SOLD_TO_PARTY	Optional	Auftraggeber
	SHIP_TO_PARTY	Optional	Warenempfänger
	BILLING_PARTY	Optional	Rechnungsempfänger
	RETURN	Optional	liefert Fehlertext bei Fehler
Tabellen-Parameter	ORDER_ITEMS_IN	Obligatorisch	Positionseingabedaten
	ORDER_PARTNERS	Obligatorisch	Daten der Geschäftspartners
	ORDER_ITEMS_OUT	Optional	Positionsausgabedaten
	ORDER_CFGS_REF	Optional	Konfiguration: Referenzdaten
	ORDER_CFGS_INST	Optional	Konfiguration: Instanzen
	ORDER_CFGS_PART_OF	Optional	Konfiguration: Part-Of Angaben
	ORDER_CFGS_VALUE	Optional	Konfiguration: Merkmalswerte
	ORDER_CCARD	Optional	Kreditkartendaten
ORDER_CFGS_BLOB	Optional	Interne Konfigurationsdaten	

Als Input-Parameter müssen mindestens die Auftragskopfdaten über die Struktur ORDER_HEADER_IN, die Partnerdaten über die Tabelle ORDER_PARTNERS und die Positionsdaten über die Tabelle ORDER_ITEMS_IN übergeben werden.

Bei konfigurierbaren Positionen können die Konfigurationsdaten in den Tabellen ORDER_CFGS_REF, ORDER_CFGS_INST, ORDER_CFGS_PART_OF, ORDER_CFGS_VALUE und ORDER_CFGS_BLOB übergeben werden. Über die Struktur ORDER_CCARD können Kreditkarteninformationen übergeben werden.

Nach erfolgreichem Anlegen des Auftrages werden zum einen die Auftragsnummer im Feld SALESDOCUMENT, zum anderen Detaildaten über die beteiligten Partner über die Strukturen SOLD_TO_PARTY, SHIP_TO_PARTY und BILLING_PARTY zurückgeliefert. Eventuell auftretende Fehler werden über den Parameter RETURN

bekanntgegeben. Informationen über Preise und Verfügbarkeit der angeforderten Funktionen werden über die Tabelle ORDER_ITEMS_OUT zurückgeliefert. Nachfolgend werden noch mal die Muss-Eingaben zusammengefasst:

Tabelle 3: Obligatorische Parameter – BAPI_SALESORDER_CREATEFROMDAT1

Struktur/Tabelle	Feld	Beschreibung
ORDER_HEADER_IN	DOC_TYPE	Verkaufsbelegart
	SALES_ORG	Verkaufsorganisation
	DISTR_CHAN	Vertriebsweg
	DIVISION	Sparte
ORDER_PARTNERS	PARTN_ROLE	Rolle des Geschäftspartners
	PARTN_NUM	Debitorennummer
ORDER_ITEMS_IN	MATERIAL	Produktnummer
	REQ_QTY	Bestellmenge

- **BAPI_SALESORDER_SIMULATE**

Diese Methode hat die gleiche Schnittstellendefinition wie die Funktion BAPI_SALESORDER_CREATEFROMDAT1. Der funktionale Unterschied liegt darin, dass der Kundenauftrag nicht verbucht sondern nur simuliert wird.

Dieses BAPI wird dafür verwendet Preisinformationen und Verfügbarkeit für angeforderte Produkte zu ermitteln. Diese Daten werden über die Tabelle ORDER_ITEMS_OUT zurückgeliefert.

Bei den Parametern der beschriebenen BAPIs handelt es sich hauptsächlich um Record- oder Tabellentypen, die sich aus mehreren einzelnen Elementen zusammensetzen. Für jedes Element müssen Informationen, entsprechend dem vom RFC API vordefinierten Datentypen RFC_TYPE_ELEMENT (Kapitel 5.3.3 – Beschreibung der Funktion „call-FunctionModul), übergeben werden, unabhängig davon ob es beim Aufruf der Funktion belegt wird oder nicht. Zusätzlich müssen für Record- und Tabellen-Parameter Strukturen in SAP angelegt werden um diese bekannt zu machen. Die meisten Parameter der verwendeten BAPIs sind optional und werden nicht alle von Clarify benötigt. Um die von Oracle aus zu tätigen BAPI-Aufrufe zu optimieren, werden für die verwendete BAPIs Funktionsbaustein als Wrapper angelegt, welche nur die Parameter enthalten, die tatsächlich von Clarify benötigt werden. Die Wrapper-Funktion ruft das entsprechende BAPI

auf, indem er die erhaltenen Werte in die Parameter des BAPIs überführt. In Tabelle 4 werden die Wrapper mit den benötigten Parametern dargestellt:

Tabelle 4: Wrapper für BAPIs

Z_BAPI_SALESORDER_CREATE			
Import-Parameter	PART_NUM	Debitorennummer	
	ORDERNO	Auftragsnummer des Kunden	
Export-Parameter	SALESDOCUMENT	Auftragsnummer des angelegten Auftrages aus SAP	
	RETURN	Fehlertext bei Fehler	
Tabellen-Parameter	ORDER_ITEMS_IN	MATERIAL	Produktnummer
		REQ_QTY	Angeforderte Stückzahl
Z_BAPI_SALESORDER_SIMULATE			
Import-Parameter	PART_NUM	Kundennummer	
	MATERIAL	Produkt-Nummer des angeforderten Produkts	
	QUANTITY	Angeforderte Stückzahl	
Export-Parameter	PRICE	Produktpreis entsprechend den Kundenkonditionen	
	DLV_DATE	Lieferdatum für die angeforderte Menge	
	RETURN	Fehlertext bei Fehler	

Da nicht alle benötigten Muss-Eingaben der Methoden BAPI_SALESORDER_CREATEFROMDAT1 und BAPI_SALESORDER_SIMULATE in Clarify verwendet werden, werden diese in den Wrapper-Funktionen mit festen Werten belegt:

- ORDER_HEADER_IN:
 - DOC_TYPE (Verkaufsbelegart): SO (Sofortauftrag)
 - SALES_ORG (Verkaufsorganisation): 1000 (Deutschland)
 - DISTR_CHAN (Vertriebsweg): 10 (Endkundenverkauf)
 - DIVISION (Sparte): 00 (Spartenübergreifend)
- ORDER_PARTNERS:
 - PARTN_ROLE (Partnerrolle): AG (Auftraggeber)

Nachfolgend wird die Wrapper-Funktion Z_BAPI_SALESORDER_SIMULATE als Beispiel dargestellt:

```

FUNCTION Z_BAPI_SALESORDER_CREATE.

*-----
*""Lokale Schnittstelle:
*   IMPORTING
*       VALUE(PART_NUM) LIKE  BAPIPARTNR-PARTN_NUMB
*       VALUE(ORDERNO) LIKE  BAPISDHEAD-PURCH_NO
*   EXPORTING
*       VALUE(SALESDOCUMENT) LIKE  BAPIVBELN-VBELN
*       VALUE(RET_MSG) LIKE  BAPIRETURN-MESSAGE
*   TABLES
*       ORDER_ITEMS_IN STRUCTURE  ZORDERITEMSIN
*-----

* Deklaration der verwendeten Variablen

DATA: HEADER LIKE BAPISDHEAD,
      PARTNER TYPE TABLE OF BAPIPARTNR WITH HEADER LINE,
      Z_BAPIITEMIN TYPE TABLE OF BAPIITEMIN WITH HEADER LINE,
      Z_BAPIITEMEX TYPE TABLE OF BAPIITEMEX WITH HEADER LINE.
      Z_RETURN LIKE BAPIRETURN.

* Belegen der Variablen mit Festwerten und den übergebenen Parameter-
Werten

HEADER-DOC_TYPE = 'SO'.
HEADER-SALES_ORG = '1000'.
HEADER-DISTR_CHAN = '10'.
HEADER-PURCH_NO = ORDERNO.

PARTNER-PARTN_ROLE = 'AG'.
PARTNER-PARTN_NUMB = PART_NUM.
APPEND PARTNER.

LOOP AT ORDER_ITEMS_IN.
    Z_BAPIITEMIN-MATERIAL = ORDER_ITEMS_IN-MATERIAL.
    Z_BAPIITEMIN-REQ_QTY = ORDER_ITEMS_IN-REQ_QTY.
    APPEND Z_BAPIITEMIN.
ENDLOOP.

* Aufruf des BAPIS

CALL FUNCTION 'BAPI_SALESORDER_CREATEFROMDATA'
    EXPORTING
        ORDER_HEADER_IN = HEADER
    IMPORTING
        SALESDOCUMENT = SALESDOCUMENT
        RETURN = Z_RETURN
    TABLES
        ORDER_ITEMS_IN = Z_BAPIITEMIN
        ORDER_PARTNERS = PARTNER.

* Übertragen des Fehlertextes

RET_MSG = Z_RETURN-MESSAGE.

ENDFUNCTION.

```

Die Funktion Z_BAPI_SALESORDER_CREATE erhält als Übergabe-Parameter Kundennummer und Auftragsnummer des Kunden und die Tabelle mit den Auftragspositio-

nen, mit Angabe der Produktnummer und der angeforderten Stückzahl. Diese Werte werden in Variablen eingelesen und beim Aufruf des BAPIs BAPI_SALESORDER_CREATEFROMDAT1 übergeben. Die zurückerhaltene Auftragsnummer wird als Rückgabewert an Oracle zurückgeliefert. Im Falle eines Fehlers wird der Fehlertext mit dem Rückgabe-Parameter RET_MSG zurückgegeben.

6.1.2. BAPI-Aufruf von Oracle

Für den Aufruf von BAPIs aus Oracle werden die Funktionen der Komponente saporad.sql des SAPORADs genutzt. Für jedes verwendete BAPI wird eine Stored Procedure in Oracle angelegt, welche die benötigten Parameter-Daten bereitstellt und die Abwicklung des BAPI-Aufrufs steuert. Die Informationen zu den einzelnen Parametern werden entsprechend den im RFC API vordefinierten Parameter-Datentypen übergeben. Hierfür werden im Package saporad.sql folgende Tabellentypen bereitgestellt, die den Datentypen des RFC APIs entsprechen:

t_SapParamTable	-	RFC_PARAMETER
t_SapTableList	-	RFC_TABLE
t_SapElementList	-	RFC_TYPE_ELEMENT
t_SapException	-	RFC_TYPE_EXCEPTION

Die Informationen bzgl. Name, Datentyp, Feldlänge der Parameter werden aus Definition der BAPIs aus SAP übernommen. Für die Angabe der Datentypen werden im saporad.pck Konstanten für RFC-Datentypen bereitgestellt, welche auf die des RFC APIs abgestimmt sind. In Tabelle 5 werden die einzelnen Datentypen dargestellt.

Tabelle 5: RFC-Datentypen

RFC-Typ	ABAP Typ	Beschreibung	Größe	Konstante in saporad.sql	Umwandlung in Oracle-Typ
RFC_CHAR	C	String mit fester Länge	1-255	kt_C	VARCHAR2
RFC_NUM	N	String, enthält nur Ziffern	1-255	kt_N	NUMBER
RFC_BYTE	X	Bytfolge	1-255	kt_X	VARCHAR2
RFC_BCD	P	Binärcodierte Dezimalzahl		kt_P	NUMBER
RFC_INT1	INT1	1-byte unsigned integer	1	kt_INT1	NUMBER
RFC_INT2	INT2	2-byte signed integer	2	kt_INT2	NUMBER
RFC_INT	I	4-byte signed integer	4	kt_INT	NUMBER
RFC_FLOAT	F	8-byte Gleitkommazahl	8	kt_FLOAT	NUMBER
RFC_DATE	D	Datum	8	kt_DATE	DATE
RFC_TIME	T	Zeit	6	kt_T	DATE

Nachdem die Parameterlisten mit den benötigten Informationen zu den einzelnen Parametern belegt sind, werden sie zusammen mit dem Namen des in SAP aufzurufenden BAPIs an die Procedure callFunctionModul des SAPORADs übergeben. Folgende Parameter werden von der Procedure callFunctionModul erwartet:

Tabelle 6: Parameter der Procedure callFunctionModul

Parametername	Art	Datentyp	Beschreibung
SFunctionModule	IN	VARCHAR2	Name des BAPIs
aInputParams	IN	t_SAPPParamTable	BAPI-IMPORT-Parameter
aOutputParams	IN OUT	t_SAPPParamTable	BAPI-EXPORT-Parameter
aTables	IN OUT	t_SAPTableList	BAPI-TABLE-Parameter (Tabelleninformationen, Elementinformation und Tabelleninhalt)
aTableElements	IN OUT	t_SAPElementList	
aTableData	IN	t_SAPTableDataList	
aSapException	IN OUT	t_SAPEException	BAPI-EXCEPTION-Parameter

Die Rückgabewerte aus SAP werden in den Parametern aOutputParams und aTableData zurückgeliefert. Auftretenden Fehler werden in dem Parameter aSapException zurückgegeben. Die einzelnen Werte werden mit den Funktionen assignSap2Ora, parseTableField und convSap2Ora des SAPORADs entnommen in Oracle-Datentypen convertiert.

Für den Aufruf der in diesem Beispiel verwendeten BAPIs werden folgende zwei Procedures angelegt:

- createOrder_SAP für den Aufruf von Z_BAPI_SALESORDER_CREATE
- getConditions_SAP für den Aufruf von Z_BAPI_SALESORDER_SIMULATE

Der Aufruf der Procedures erfolgt aus Clarify, durch Übergabe folgender Parameter:

Tabelle 7: Parameter der Procedures createOrder_SAP und getConditions_SAP

createOrder_SAP		
EingabeParameter	custObjid	Clarify-Kundennummer
	orderNoCust	Auftragsnummer des Kunden
Ausgabe-Parameter	orderNoSap	Auftragsnummer aus SAP
	returnCode	Fehlercode
	returnText	Fehlertext
getConditions_SAP		
EingabeParameter	material	Produktnummer
	quantity	Angeforderte Stückzahl
Ausgabe-Parameter	price	Preis des Produktes
	dlv_date	Lieferdatum für das angeforderte Produkt
	returnCode	Fehlercode
	ReturnText	Fehlertext

Nachfolgend wird als Beispiel die Procedure createOrder_SAP dargestellt:

```

PROCEDURE createOrder_SAP(custObjid      IN NUMBER,
                          orderNoCust    IN VARCHAR2,
                          orderNoSap     OUT NUMBER,
                          return_code    OUT INTEGER,
                          return_text    OUT VARCHAR2) IS

  InputParam      saporad.t_SAPParamTable;
  OutputParam     saporad.t_SAPParamTable;
  SapTableList    saporad.t_SAPTableList;
  SapTableData    saporad.t_SAPTableDataList;
  SapTableValues  saporad.t_SAPTableValues;
  SapElementList  saporad.t_SAPElementList;
  SapException    saporad.t_SAPEException;
  DebitNoSap      table_site.external_id%TYPE;
  RetMsgSap       VARCHAR(220);
  CURSOR c_orderItems IS
    SELECT material,
           req_qty
    FROM x_orderitems_temp
    WHERE order_no = orderNoCust;
  rec_orderItem c_orderItems%ROWTYPE;
  j BINARY_INTEGER;

  BEGIN

    SELECT external_id INTO debitNoSap
    FROM table_site
    WHERE objid = custObjid;

    InputParam(1).name      := 'PART_NUM';
    InputParam(1).type      := saporad.kT_C;
    InputParam(1).len       := 10;
    InputParam(1).value     := LPAD(debitNoSap,10,'0');

    InputParam(2).name      := 'ORDERNO';
    InputParam(2).type      := saporad.kT_C;
    InputParam(2).len       := 20;
    InputParam(2).value     := LPAD(orderNoCust,10,'0');

    OutputParam(1).name     := 'SALESDOCUMENT';
    OutputParam(1).type     := saporad.kT_C;
    OutputParam(1).len      := 10;

    OutputParam(2).name     := 'RET_MSG';
    OutputParam(2).type     := saporad.kT_C;
    OutputParam(2).len      := 220;

    SAPTableList(1).name    := 'ORDER_ITEMS_IN';
    SAPTableList(1).elementcount := 2;

    SAPElementList(1).tablename := 'ORDER_ITEMS_IN';
    SAPElementList(1).name      := 'MATERIAL';
    SAPElementList(1).type      := saporad.kT_C;
    SAPElementList(1).len       := 18;
    SAPElementList(1).decimals  := 0;

    SAPElementList(2).tablename := 'ORDER_ITEMS_IN';
    SAPElementList(2).name      := 'REQ_QTY';
    SAPElementList(2).type      := saporad.kT_NUM;
    SAPElementList(2).len       := 13;
    SAPElementList(2).decimals  := 0;

```



```

j := 0;
OPEN c_orderItems;
LOOP
  FETCH c_orderItems INTO rec_orderItem;
  EXIT WHEN c_orderItems%NOTFOUND;
  aSAPTableData(j).tablename := 'ORDER_ITEMS_IN';
  aSAPTableData(j).value := RPAD(rec_orderItem.material,18,' ')
                          ||LPAD(rec_orderItem.req_qty,13,'0');
  j := j + 1;
END LOOP;
CLOSE c_orderItems;

IF saporad.callFunctionModule('Z_BAPI_SALESORDER_SIMULATE1', InputParam,
                             OutputParam, SAPTableList,
                             SAPElementList, SAPTableData,
                             SAPException) THEN
  saporad.AssignSAP2ORA(aOutputParam, 'SALESDOCUMENT', orderNoSap);
  saporad.AssignSAP2ORA(aOutputParam, 'RET_MSG', orderNoSap);

  IF RET_MSG <> ' ' THEN
    return_text := (SAPException.name||' '||SAPException.text);
    return_code := ude_internal_error;
    RETURN;
  ENDIF;

  IF SAPException.name <> ' ' THEN
    return_text := (SAPException.name||' '||SAPException.text);
    return_code := ude_internal_error;
    RETURN;
  END IF;
END IF;

EXCEPTION
  WHEN saporad.conversion_error THEN
    return_code := saporad.conversion_error;
    return_text := SQLERRM;
  WHEN sending_error THEN
    return_code := saporad.sending_error;
    return_text := SQLERRM;
  WHEN receiving_error THEN
    return_code := saporad.receiving_error;
    return_text := SQLERRM;
  WHEN message_corrupted THEN
    return_code := ude_internal_error;
    return_text := SQLERRM;

END createOrder_SAP;

```

Die Procedure createOrder_SAP erhält als Eingabe-Parameter die Clarify-Kundennummer und die Auftragsnummer des Kunden. Die SAP-Kundennummer, die in diesem Beispiel in dem Feld external_id der Kundentabelle table_site abgelegt ist, wird mit Hilfe der Clarify-Kundennummer selektiert und als IMPORT-Parameter PART_NUM von Z_BAPI_SALESORDER_CREATE in der Liste InputParam gespeichert. Auch die Auftragsnummer des Kunden wird in der Liste InputParam als ORDERNO abgelegt. Die Positionen des Auftrages werden mit Hilfe eines Cursors aus

der Tabelle x_orderitems_temp mit Hilfe der Auftragsnummer entnommen und als Tabellen-Parameter ORDER_ITEM_IN übergeben. Mit callFunctionModul wird die Funktion BAPI_SALESORDER_CREATE in SAP aufgerufen mit welcher der Auftrag in SAP angelegt wird. Die Auftragsnummer des in SAP angelegten Auftrags wird in dem EXPORT-Parameter SALESDOCUMENT zurückgeliefert und an Clarify zurückgegeben. Die von SAP in dem EXPORT-Parameter RET_MSG oder in dem EXCEPTION-Parameter zurückgelieferten Fehler und auch Fehler die in Oracle auftreten werden in den Parametern returnCode und returnText nach Clarify weitergereicht.

6.1.3. Anpassung der Benutzerschnittstelle in Clarify

Von den grafischen Benutzeroberfläche in Clarify wird der erste Teil des hier als Beispiel verwendete Geschäftsprozess der „Auftragsverarbeitung“ von dem Benutzer abgewickelt. Die Aufgaben des Benutzers liegen dabei in der Eingabe der Auftragsdaten, der Prüfung der Preise und Verfügbarkeit der angeforderten Produkte und letztendlich des Anlegen des Auftrages.

The screenshot shows the 'Sales Order-05' form in Clarify. The form is divided into several sections:

- Header Fields:** ID/Title (05, EB-003), Condition (Quoted), Expire Date (8/23/2002), Status (New Order), Currency (EURO), Price List (EURO Std Price List), Ship Via (Please Specify), FOB (Origin).
- Contact Information:** Contact (Hans Meyer), Account, Sales Rep (bier).
- Shipping Information:** Ship to (Hans Meyer Maar), Address (Maarweg).
- Line Items Table:**

Line	Part No	Revision	Description	Qty	Price	Delivery Da	Status	Comments
1	Analog		Analoger Telefonanschl	1	€20	??/? ?/?/?	Open	
- Summary Fields:** Gross (€20), Net Line Price (€20), Fixed Adjustment (€0), Subtotal (€20), Shipping (€0), Handling (€0), Tax (€0), Total (€20).
- Buttons:** Copy Order, New Order, Add Order, Save Order, Payments, Done.

Abbildung 19: Clarify Form „Sales Order“

Das Erfassen der Auftragsdaten erfolgt in Clarify über die Maske Sales Order (siehe Abbildung 19), wobei der Benutzer die Möglichkeit hat die Daten zum Kunden, Auftragskonditionen und die einzelnen Positionen des Auftrages einzugeben. Für das Anlegen der Auftragspositionen wird über einen Rechts-Click auf die Tabelle „Line Items“ die Maske Select Product geöffnet. Durch Auswahl eines der angezeigten Produkte und Eingabe der gewünschten Stückzahl, wird durch Klicken des Buttons „Submit“ das Lieferdatum und der Preis des Produktes angezeigt. Ist der Auftraggeber mit den Liefer- und Preisbedingungen einverstanden, wird das Produkt durch Klicken des Buttons „Use“ als Auftragsposition in die Tabelle „Line Items“ der Maske Sales Order übernommen. Nachdem alle gewünschten Auftragspositionen angelegt wurden kann die Maske Select Product geschlossen werden. Durch Klicken des Buttons „Save Order“ wird der Auftrag angelegt.

Part Num	Rev	Description	Domain	Model No	Line	Family	Part Status	Rev. Status	Sppt Prog?
▶ Analog		Analoger Telef	Product		All	All	Active	Active	No
DSL		DSL Flatrate	Product		All	All	Active	Active	No
Einzelplatzz ISDN		Internet Einzel ISDN Telefona	Product		All	All	Active	Active	No

Abbildung 20: Clarify Form „Select Product“

Der beschriebene Ablauf wurde durch das Anpassen der Benutzeroberfläche auf die Schnittstelle zu SAP möglich. Für die Feststellung der Preise und Lieferzeit wurde die Maske „Select Product“ um folgende Objekte erweitert:

- Feld Quantity
- Feld Delivery-Date
- Feld Price
- Button Submit.

Zusätzlich wurde aus der Produkt-Tabelle die Spalte Price entfernt, in der die in Clarify vorhandenen Produktpreise dargestellt wurden. Die Standardfunktionalität von Clarify, die durch den Clear-Basic-Code zu den Clarify-Forms festgelegt ist, wurde nur geringfügig verändert. Es wurde folgende Funktionalität hinzugefügt:

- durch Klicken des Buttons Submit wird der Aufruf der Oracle Procedure getConditions_SAP initiiert. Voraussetzung hierfür ist die Auswahl eines Produktes in der Produkt-Tabelle und die Eingabe einer Stückzahl. Die Produktnummer des ausgewählten Produktes (Feld „Part Num“) und die eingegebene Stückzahl werden als Parameter an die Procedure getConditions_SAP übergeben. Nach der erfolgreichen Ausführung der Procedure werden die Rückgabewerte für Preis und Lieferdatum in den Feldern „Price“ und „Delivery-Date“ dargestellt. Bei Rückgabe eines Fehlers wird dieser in einer Message-Box an den Benutzer weitergegeben.
- Die Funktionalität der Buttons „Use“ und „Use/Done“ wurde so verändert, daß sie nur dann betätigt werden können, wenn die Felder „Price“ und „Delivery-Date“ belegt sind, also eine Abfrage in SAP getätigt wurde. Sobald ein anderes Produkt in der Produkt-Tabelle angewählt wird, werden diese Felder wieder geleert und eine neue Abfrage muss getätigt werden.

Die Maske „Sales Order“ wurde nur durch das Hinzufügen der Spalte „Delivery Date“ in die Tabelle „Line Items“ verändert. Im Clear-Basic-Code der Maske „Sales Order“ wurden folgende Änderungen durchgeführt:

- Die Spalten „Qty“, „Price“ und „Delivery Date“ werden mit den Daten der neuen Felder aus der Maske „Select Product“ gefüllt.
- Durch Klicken des Buttons „Save Order“ wird zuerst der Aufruf der Oracle-Procedure createOrder_SAP initiiert. Erst nach erfolgreicher Durchführung wird die Standardfunktionalität der Maske, also das Speichern des Auftrages in Clarify, vollzogen. Da

die Übergabe von zusammengesetzten Datentypen in Clarify nicht möglich ist, werden die Auftragspositionen temporär in der dafür angelegten Tabelle `x_orderitems_temp` gespeichert. Dabei wird die Auftragsnummer des Kunden aus dem Feld „Title“ und die Produktnummer und Stückzahl aus den Spalten „Part Num“ und „Qty“ der Tabelle „Line Items“ entnommen und in der Tabelle `x_orderitems_temp` abgelegt. Mit Hilfe der an die Procedure übergebenen Auftragsnummer des Kunden können die gespeicherten Positionen in der Tabelle eindeutig identifiziert werden. Nach der Durchführung der Procedure werden die Auftragspositionen wieder aus der Tabelle entfernt. Zusätzlich zu der Auftragsnummer erhält die Procedure `createOrder_SAP` die Clarify-Kundennummer als Übergabe-Parameter, mit der die SAP-Kundennummer aus der Datenbank selektiert werden kann, die in diesem Beispiel in der Kundentabelle gespeichert ist. Die von der Procedure zurückgelieferte Auftragsnummer aus SAP wird in der Clarify-Datenbank im Feld `po_number` der Tabelle `table_contract` (siehe Abbildung 21) gespeichert. Tritt bei der Durchführung der Procedure ein Fehler auf wird diese als Message-Box an den Kunden weitergegeben und das Speichern des Auftrags in Clarify wird auch nicht durchgeführt. Nach Beheben des Fehlers kann der Vorgang vom Benutzer wiederholt werden.

6.2. Teil 2 - Durchführung in SAP

Der zweite Teil des Geschäftsprozesses „Auftragsverarbeitung“, die Durchführung des Auftrages bzw. Auslieferung an den Kunden wird in diesem Beispiel im SAP-System vollzogen. Die Anforderung hierbei besteht darin, nach erfolgreichem Abschluss des Auftrages in SAP, auch in Clarify den Status der Auftrages auf „Fulfilled“ zu setzen. Die Realisierung erfolgt in zwei Schritten:

1. Initiierung der Statusänderung aus SAP
2. Durchführung der Statusänderung in Oracle

Nachfolgend wird die Realisierung der einzelnen Schritte beschrieben.

6.2.1. Initiierung der Statusänderung aus SAP

Der von Clarify aus angelegte Auftrag kann in SAP über die Transaktion VA03 angezeigt werden. Durch Auswahl der Menue-Felder *Logistik -> Vertrieb -> Verkauf* im Einstiegsbild von SAP gelangt der Benutzer in das SD-Modul (Sales & Distribution). Durch Wählen von *Auftrag -> Anzeigen* wird die Transaktion VA03 aufgerufen. Der Auftrag kann in SAP über die SAP-Auftragsnummer, die Auftragsnummer des Kunden, oder durch Anzeige aller Aufträge zum Kunden identifiziert werden. In der Maske „Auftrag anzeigen“ werden die Daten des Auftrages dargestellt.

Terminauftrag 5891 anzeigen: Übersicht

Belegfluß Auftraggeber Aufträge Kundenstamblatt Statusübersicht

Terminauftrag: 5891 Nettowert: 0,00 DEM
 Auftraggeber: 1111 P.S.G. GmbH / Augsburgstr. 345 / D-90431 Nuernberg
 Warenempfänger: 1111 P.S.G. GmbH / Augsburgstr. 345 / D-90431 Nuernberg
 Bestellnummer: 12345 Bestelldatum: 10.07.2002

Verkauf Positionsübersicht Besteller Beschaffung Versand Absagegrund

Wunschlieferdat: T 11.07.2002 AuslieferWerk:
 Komplettlief.: Gesamtgewicht: 560 K6
 Liefersperre: Volumen: 1,500 M3
 Fakturasperre: Preisdatum: 10.07.2002
 Zahlungskarte: Gültig bis:
 Zahlungsbed: ZB01 14 Tage 3%, 30/2%, ... Incoterms: CIP Nürnberg
 Auftragsgrund: Lieferzeit:
 Vertr.bereich: 1000 / 10 / 00 Deutschl. Frankfurt, Endkundenverkauf, Spartenübergreifend

Pos	Material	Auftragsmenge	ME	E	Bezeichnung	1. Datum	Werk	Ptyp
10	P-103	2	ST	<input checked="" type="checkbox"/>	Pumpe Chromstahl IDESNORM...	11.07.2002	1000	TAN

Verfügbarkeit Konditionen Einteilungen Konfig.

VA03 ksap OVR

Das Ausliefern des Auftrages an den Kunden wird durch Wählen der Menue-Felder *Verkaufsbeleg -> Beliefern* eingeleitet. Dies ist erst möglich wenn alle benötigten Daten vorhanden sind und das Lieferdatum erreicht wurde. In diesem Beispiel wird nur der Fall einer Komplettlieferung abgehandelt. Teillieferung werden nicht berücksichtigt.

Die Änderung des Auftragsstatus in Clarify soll durch eine Oracle Stored Procedure durchgeführt werden. Um dies zu initiieren ist es nicht nötig die Benutzermaske in SAP, das sogenannte Dynpro (Dynamisches Programm), zu verändern. Der Aufruf der Procedure wird in eins der dahinterliegenden ABAP-Programme eingebunden, die ausgeführt werden, wenn der Benutzer die Belieferung des Auftrages auswählt.

Für die Erweiterung von Standard-Programmen werden in SAP sogenannte User-Exits zur Verfügung gestellt. User-Exits sind „leere Modifikationshülsen“ die im Standard vorgedacht sind um mit kundenspezifischer Logik ausgestaltet werden zu können. Durch die Verwendung von User-Exits bleibt die Aufwärtskompatibilität gewährleistet, d.h. der Ansporn eines Exits aus der Standardsoftware sowie die Gültigkeit des User-Exits bleiben auch in zukünftigen Release-Ständen erhalten.

Für den Aufruf der Oracle Procedure für die Statusänderung in Clarify wird der User-Exit USEREXIT_SET_STATUS_VBUK verwendet, der in der Funktion RV_XVBUK_MAINTAIN angesprochen wird. Die Funktion RV_XVBUK_MAINTAIN ist für das Pflegen des Auftragsstatus in SAP zuständig, wobei der genannte User-Exit vor der Speicherung der Statusänderung in der Datenbank aufgerufen wird. Zu diesem Zeitpunkt ist der Rekord VBUK mit den neuen Statusdaten zu dem Auftrag gefüllt und muss nur noch an die Datenbank übergeben werden. In dem User-Exit wird der Lieferstatus, das Feld LFSTK, abgefragt. Enthält dieser den Wert 'c' bedeutet das der Auftrag erhält den Status „geliefert“. Ist das der Fall wird Procedure „SET_ORDERSTATUS_CLARIFY“ als RFC-Funktion aufgerufen, die den Status in Clarify auf „Fulfilled“ umsetzt. Bei einer RFC-Funktion wird die DESTINATION, also der Empfänger angegeben. Hier wird die Destination ORA verwendet, welche die Adresse des SAPORAD enthält. Dieser leitet den Aufruf der Procedure an Oracle weiter und liefert die erhaltenen Rückgabewerte an SAP zurück. Die Procedure erhält als Übergabewert die Belegnummer des Auftrages dessen Status umgesetzt werden soll, die der Tabelle VBUK aus dem Feld VBELN entnommen wird. Als Rückgabewert wird im Falle eines Fehlers die Fehlerbeschreibung aus Oracle zurückgegeben. Dieser wird als Message an den Benutzer weitergeleitet um die Behebung des Fehlers zu gewährleisten. Nachfolgend wird der User-Exit dargestellt.

```

FORM USEREXIT_SET_STATUS_VBUK.

*" Deklaration der verwendeten Variablen
DATA: Z_ERROR_TXT(100) TYPE C.

*" Prüfen ob der Lieferstatus gesetzt ist
IF VBUK-LFSTK EQ 'c'

*" Aufruf der RFC-Funktion
CALL FUNCTION 'SET_ORDERSTATUS_CLARIFY' DESTINATION ORA
EXPORTING
PART_NUM = VBUK-VBELN
IMPORTING
ERROR_TXT = Z_ERROR_TXT.

*" Ausgabe der zurückgelieferten Fehlerbeschreibung, falls vorhanden
IF Z_ERROR_TXT <> ' '.
MESSAGE c001(clarify) WITH Z_ERROR_TXT.
ENDIF.
ENDIF.

END FORM.

```

6.2.2. Durchführung der Statusänderung in Oracle

Für das Ändern des Auftragsstatus wird in Oracle die Procedure `set_orderStatus` angelegt. Um die Funktionalität des Clarify-Standards weiterhin zu gewährleisten, muss die Procedure die selben Aktivitäten durchführen, die beim Ändern des Auftragsstatus von der Benutzermaske aus in Clarify, erfolgen. Die Auftragsdaten sind in der Clarify-Datenbank in den Tabellen `table_contract`, `table_contr_schedule`, `table_contr_itm` und `table_condition` abgelegt. Im folgenden Ausschnitt des Clarify-Datenmodells werden die einzelnen Tabellen mit den in diesem Beispiel betroffenen Feldern dargestellt:

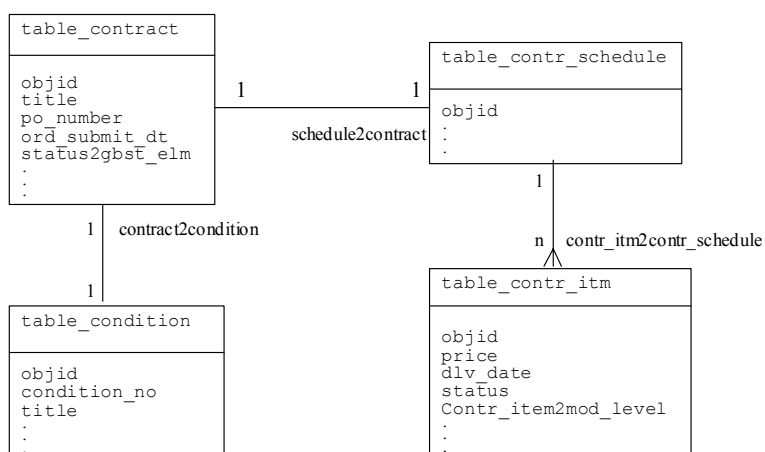


Abbildung 21: Auftrags-Tabellen im Clarify-Datenmodell

Die Tabellen table_contract und table_contr_schedule enthalten Information zum Auftrag, table_condition die Auftragskonditionen und table_contr_itm die Informationen zu den einzelnen Positionen des Auftrags.

Für das Umsetzen des Auftragsstatus auf „Fulfilled“ müssen folgende Felder der Auftragsstabelle verändert werden:

Tabelle 8: Datenbank-Änderungen beim Umsetzen des Auftragsstatus

Tabelle	Feld	Änderung
table_contract	ord_submit_dt	wird auf das aktuelle Datum gesetzt
	status2gbst_elm	verweis auf den Wert “Fulfilled” der Tabelle gbst_elm
table_contr_itm	status	erhält den Wert “Fulfilled”
table_condition	condition_no	erhält die Konditionsnummer “2048” (“Submitted“)
	title	erhält den Wert “Submitted”

Die Procedure set_orderStatus erhält als Eingabeparameter die SAP Auftragsnummer die in diesem Beispiel im Feld po_number der Tabelle table_contract abgelegt ist. Anhand dieser kann der Auftrag in der Datenbank identifiziert werden und die Änderung des Status durchgeführt werden. Im Falle eines Fehlers wird der Fehlertext an SAP zurückgegeben.

Damit die Procedure in Oracle aufgerufen werden kann, muss sie beim SAPORAD bekannt gemacht werden. Dies wird in der Procedure callPLSQLProcedure durchgeführt.

```

PROCEDURE callPLSQLProcedure (sProcedureName IN VARCHAR2) IS
BEGIN
  IF sProcedureName = 'SET_ORDER_STATUS' THEN
    salesorder.set_orderStatus( gaImporting(1).value,
                               gaExporting(1).value);
  END IF;
END;

```

callPLSQLProcedure erhält den Namen der Procedure als Eingabeparameter und ruft diese auf, indem sie die Parameter aus den globalen Rekords gaImporting und gaExporting übergibt (siehe Kapitel 5.3.3 – Beschreibung von callSQLProcedure).

7. Zusammenfassung

Die Entwicklung der Absatzmärkte und der technologische Fortschritt haben dazu geführt, dass Kundenbeziehungen im Wettbewerb zwischen den Unternehmen immer mehr im Vordergrund stehen. Der Kundenservice hat sich zu einem bedeutenden Differenzierungsfaktor im Wettbewerb entwickelt. Die Pflege der bestehenden Kundenbeziehungen und eine systematische Neukundengewinnung sind ein entscheidender Erfolgsfaktor. Bedingt durch den Wettbewerb sind die Anforderungen der Kunden gestiegen. Neben der Bereitstellung der Produkte und Dienstleistungen erwarten die Kunden eine personalisierte Kommunikation mit den Unternehmen und eine rasche und effiziente Bearbeitung der jeweiligen Anfragen. CRM-Systeme unterstützen die Unternehmen dabei, diesen Ansprüchen gerecht zu werden und helfen den Unternehmen ihre Geschäftsprozesse auf den Kunden auszurichten.

Geschäftsprozesse dürfen aber nicht an Systemgrenzen enden, sondern müssen applikationsübergreifend abgebildet sein. Aus diesem Grund streben immer mehr Unternehmen nach einer integrierten Lösung von CRM- und ERP-Systemen, die den durchgängigen reibungslosen Prozess von der Kundeninteraktion bis zum Fulfillment gewährleistet. Um die Bearbeitung von Kundenprozessen möglichst kostensparend und effizient zu unterstützen, muss ein Höchstmaß an Datenkonsistenz und die Durchgängigkeit der Prozesse zwischen Frontend- und Backend-System über alle Kanäle hinweg gewährleistet sein. Das Ziel dieser Diplomarbeit war eine Lösung zur erarbeiten für die Prozessintegration zwischen CRM- und ERP-Systemen zu erarbeiteten.

Aufbauend auf den Grundlagen von CRM und ERP wurden die beiden Systeme Clarify eFrontoffice und SAP R/3 vorgestellt, für welche die Integration vollzogen wurde. Aus den verschiedenen Integrationstechnologien, die von den beiden Systemen angeboten werden, wurden die für die Prozessintegration geeigneten Schnittstellen ausgewählt, wobei die Hauptanforderung darin bestand die Abfolge der Funktionen eines Geschäftsprozesses durchgängig zu gestalten, also von einem System aus steuern zu können. Dies erfordert den Zugriff eines Systems auf Funktionalitäten des jeweiligen anderen Systems. Als geeignete Schnittstellen wurde auf der Seite von SAP die Integration über BAPIs (Business Applikation Programming Interfaces) bzw. Funktionsbausteinen ausgewählt

und auf der Seite von Clarify die Integration über die Datenbank-Technologie, wobei hier die Technologie des Herstellers Oracle verwendet wurde. Für den Zugriff auf die Funktionalitäten des anderen Systems wurde eine generische Lösung erarbeitet, die es ermöglicht von Oracle Funktionen des SAP-Systems (BAPIs) und von SAP die Funktionen von Oracle (Stored Procedures) aufzurufen.

In Kapitel 6 wurde als Machbarkeitsbeleg beispielhaft der Geschäftsprozess „Auftragsverarbeitung“ mit der erarbeiteten Lösung integriert. Das Beispiel zeigt, dass durch den ermöglichten Zugriff auf die Funktionen des anderen Systems die Durchgängigkeit des Geschäftsprozesses gewährleistet wurde. Der Benutzer kann die dafür erforderlichen Aktivitäten von einem System aus tätigen, Informationen aus dem anderen System werden direkt auf der Benutzeroberfläche des verwendeten Systems dargestellt und Änderungen an Daten werden automatisch übertragen. Dadurch wird die Konsistenz und Aktualität der Systeme gewährleistet und Kundenwünsche können schnell und effizient bearbeitet werden. Durch die Generik der Schnittstelle kann entsprechend dem Beispiel der „Auftragsverarbeitung“ auch jeder andere Geschäftsprozess abgebildet werden. Die Aufwärtskompatibilität der Schnittstellen wird durch die jeweiligen Hersteller gewährleistet. Die Auswahl der Integration über die Datenbank-Technologie auf der Seite von Clarify, ermöglicht es entsprechend auch andere Systeme, welche auf der Datenbank aufbauen, zu integrieren.

Fazit: Durch Integration von CRM und ERP entsteht eine durchgängige Geschäftsprozesskette, welche die Schranke zwischen den internen Geschäftsvorgängen und der Interaktion mit den Kunden überwindet. Nur so ist gewährleistet, dass das CRM- und das ERP-System wie zwei Zahnräder ineinander greifen– und der Kunde wirklich ins Zentrum der Unternehmensstrategie rückt.

8. Literaturverzeichnis

- [CLA00] Clarify Architecture
White Paper
Clarify Inc., 2000
- [CLA01] Clarify Documentation
Nortel Networks, 2001
- [ESA01] Berna Esam
Grundlegende Applikationen und Daten, - ERP, SCM und CRM
Seminararbeit, Betriebswirtschaftliches Institut der Universität Stuttgart
- [FER01] Otto K. Ferstl, Stephan Mantel, Martin Schissler, Elmar J. Sinz
Unterstützung von Kopplungsarchitekturen durch SAP R/3
Bayerischer Forschungsverbund Wirtschaftsinformatik, Würzburg 2001
- [GUE00] Helmuth Gumbel
Integrating eBusiness with SAP R/3
White Paper, November 2000
- [JAB02] Prof. Dr. Ing. Stefan Jablonski
Eine prozessbasierte, ganzheitliche Methode für EAI
Prodato Integration Technologie GmbH, 2002
- [KEL00] Dr. Horst Keller, Sascha Krüger
ABAP Objects – Einführung in die SAP-Programmierung
Galileo Press GmbH, Bonn 2000

- [RAP00] Reinhold Rapp
Customer Relationship Management. Das neue Konzept zur Revolutionierung der Kundenbeziehungen
Campus Fachbuch, August 2000
- [SAP01] mySAP CRM aus betriebswirtschaftlicher Sicht
White Paper
SAP Deutschland AG, 2001
- [SAP96] ABAP/4 Development Workbench
Remote Communications
SAP Documentation
- [VIC01] Prof. Frank Victor
SAP R/3 – Vorlesungsskript zum Wahlpflichtfach
Fachhochschule Köln, Wintersemester 2001
- [WIL00] Prof. Klaus Wilde, Hajo Hippner
absatzwirtschaft CRM 2000 – Studie der Universität Eichstätt
Verlagsgruppe Handelsblatt, 2000

9. Internetquellen

- [URL01] ResQ Enterprise: Integrated CRM/ERP for the Masses
<http://www.opensolutionsinc.com> - Stand: März 2002
- [URL02] Firms Link CRM, ERP To Streamline Order Process
<http://www.crmdaily.com/perl/story/12418.html> - Stand: März 2002
- [URL03] ERP und CRM: Zwei Welten wachsen zusammen
http://www.softlab.at/solutions/offering/crm_erp.shtml
Stand: März 2002
- [URL04] Integrating ERP and CRM
http://eai.ebizq.net/erp/earls_1.html - Stand: März 2002
- [URL05] Online retailers bring ERP and CRM systems to the fore
http://eai.ebizq.net/erp/lawton_1.html - Stand: März 2002
- [URL06] Integrated Solutions - Beware The ERP/CRM Split
http://www.integratedsolutionsmag.com/articles/2001_12/011207.htm
Stand: April 2002
- [URL07] Das Versprechen von ERP und CRM
<http://www.impress.com/news/news2.html> - Stand: April 2002
- [URL08] Integrating CRM, ERP and E-Commerce
http://b2b.ebizq.net/e_commerce/thomas_1.html - Stand: April 2002
- [URL09] ERP
<http://student.ifs.tuwien.ac.at/~bn02/inhalt-erp.html> - Stand: Mai 2002

- [URL10] ERP: Drei Buchstaben für Ihr Unternehmen
<http://www.e-business.de/texte/5319.asp> - Stand: Mai 2002
- [URL11] ERP-/ ERM-Systeme
<http://www.competence-site.de/standardsoftware.nsf!open>
Stand: Mai 2002
- [URL12] CAS-/ CRM-Systeme
<http://www.competence-site.de/crm.nsf!open> - Stand: Mai 2002
- [URL13] ERP Forum
<http://www.erp-forum.de> - Stand: Mai 2002
- [URL14] Einführung von SAP15
<http://www.fh-wedel.16de/~gla/sap/tutorial/allg/index.html>
Stand: Juni 200
- [URL15] Einführung in die Geschäftsprozeßmodellierung
<http://www.empirica.com/geschaeftsproz/> - Stand: Juni 2002
- [URL16] Übersicht Service-Engineering
<http://www.business-wissen.de/index.php?bsID=12> - Stand: Juni 2002
- [URL17] Oracle Architektur
<http://www.oracle.com/appsnet/technology/ia/main.html>
Stand: Juni 2002
- [URL18] SAP Help Portal
<http://help.sap.com/> - Stand: Juni 2002 - Stand: Juni 2002

- [URL19] Oracle Online Documentation
http://download-west.oracle.com/docs/cd/A87861_01/NT817EE/index.htm
Stand: Juni 2002
- [URL20] Entwicklung integrierter Systeme
<http://www.informatik.uni-halle.de/dbs/Lehre/Integr/> - Stand: Juni 2002
- [URL21] Embedded SQL
<http://www-lehre.informatik.uni-osnabrueck.de/~db/kap8/kap8b.html> -
Stand: Juli 2002
- [URL22] An Introduction to PL/SQL
<http://www.dulcian.com/papers/Introduction%20to%20PLSQL.htm>
Stand: Juli 2002
- [URL23] C Standard Library
<http://www.infosys.utas.edu.au/info/documentation/C/CStdLib.html#strtol> -
Stand: Juli 2002

10. Glossar

Automatic Call Distribution (ACD)

Die rechnergestützte automatische Verteilung eingehender Anrufe in einem Call Center. Eine ACD Anlage kann Teil der Telefonanlage oder ein eigenständiges System sein. Sie stellt die eingehenden Anrufe in eine Warteschleife und weist sie nach vorprogrammierten Schlüsseln freien Arbeitsplätzen zu.

Back Office

Im Englischen in etwa für "Büro im Hintergrund". Die Mitarbeiter eines zentralen Back Office einer Firma können die Kollegen an der Front (Front Office), d.h. beim Kunden, den Rücken stärken. Das Back Office versorgt sie mit relevanten Informationen und Analysen. Es kann Tätigkeiten zur Entlastung übernehmen bzw. delegieren und stellt sein geballtes Know-how zur Verfügung.

Beschwerde-Management

Beschwerdemanagement umfasst die Planung, Durchführung und Kontrolle aller Maßnahmen, die ein Unternehmen mit Kundenbeschwerden ergreift. Die Ziele des Beschwerdemanagements liegen darin, Kundenzufriedenheit wieder herzustellen, die negativen Auswirkungen von Kundenunzufriedenheit auf das Unternehmen zu minimieren und die in Beschwerden enthaltenen Hinweise auf betriebliche Schwächen und Marktchancen zu identifizieren und zu nutzen.

Call Center

Unternehmensabteilungen oder eigenständige Firmen, die einen serviceorientierten telefonischen Dialog des Unternehmens mit den Kunden beziehungsweise Interessenten gewährleisten. Hauptaufgaben des Call Centers sind die Bearbeitung eingehender Anfragen, Reklamationen, Beschwerden und Aufträge (Inbound Aufgaben). Im Outbound Call Center werden Kunden aktiv vom Unternehmen aus betreut beziehungsweise Verkaufsgespräche geführt.

Cross Selling

Über die Cross Selling Rate wird ermittelt, wie viele Kunden, die das Produkt A gekauft haben, auch für das Produkt B in Frage kommen könnten. Es wird also ermittelt, inwieweit es möglich ist, einem Kunden weitere Produkte des Sortiments zu verkaufen.

CTI - Computer Telephone Integration

CTI ist die Verknüpfung von Telefonanlage und EDV. Die Telefonanlage ist direkt mit der Kundendatenbank des Unternehmens verbunden. Jeder Call Center Agent erhält so direkt auf seinem Bildschirm alle kundenrelevanten Daten, um zum Beispiel Bestellungen aufzunehmen und direkt weiterzuleiten. Wenn alle Kundenbetreuer beschäftigt sind, speichert die Anlage eingehende Anrufe oder leitet sie je nach Bedarf an ein anderes Call Center weiter.

Customer Relationship Management

Unter Customer Relationship Management (CRM) verstehen wir die ganzheitliche Bearbeitung der Beziehung eines Unternehmens zu seinen Kunden. Kommunikations-, Distributions- und Angebotspolitik sind nicht weiterhin losgelöst voneinander zu betrachten, sondern integriert an den Kundenbedürfnissen auszurichten. Ziel des CRM ist es, wirtschaftlich orientierte Entscheidungskriterien und Kriterien der Kundenzufriedenheit zur Verfügung zu stellen, um eine Optimierung des Kundenportfolios im Hinblick auf den Unternehmenswert zu erreichen.

Customer Lifetime Value

Die Verfolgung des Customer Lifetime Value als Zielgröße im Marketing stellt ein Konzept dar, sich nicht nur an dem kurzfristigen, in einer Periode mit einem Kunden erzielbaren Erfolg zu orientieren, sondern sich an dem langfristigen Wert der Kundenbeziehung mit all seinen Ein- und Auszahlungsströmen zu orientieren. Der Customer Lifetime Value ist in der Regel um so höher, je höher die Kundenzufriedenheit ist.

Data Mining

Data Mining ist eine Kombination verschiedener multivarianter statistischer Verfahren einschließlich neuronaler Netze, die den Marketing-Experten helfen, Trends schon früh zu erkennen, zum Beispiel Abwanderungstendenzen in einer bestimmten Kundengruppe. Data-Mining ist ein komplexer Prozeß, bei dem zunächst aus einer Vielzahl von Daten die Variablen definiert werden müssen, die für die simultane Auswertung in Frage kommen. Entscheidend für das Resultat einer Anfrage ist die Wahl des geeigneten statistischen Modells, das zur Datenauswertung herangezogen wird.

Data Warehouse

Sammlung aller relevanten Daten im sogenannten »Warenhaus«, damit diese dann per Data Mining oder mit anderen Analysetools (z. B. OLAP) analysiert und ausgewertet werden können.

E-Business

Die Abgrenzung von E-Commerce und E-Business ist alles andere als klar, nicht selten werden beide Begriffe synonym verwendet. E-Business stellt den umfassenderen Begriff dar und bedeutet die Digitalisierung sämtlicher Geschäftsprozesse. Nicht nur einzelne Geschäftsbereiche wie Vertrieb oder Marketing werden ins Netz verlagert, sondern der gesamte Unternehmensablauf wird vom Netz geprägt.

EDI – Electronic Data Interchange

EDI wird in der Regel bei enger Kooperation zwischen Kunden und Herstellern eingesetzt. EDI umfasst den Datenfluss zwischen den Partnern. Vornehmlich handelt es sich um Bestell- und Fakturierungsdaten. Lösungen sind heute häufig Web-basiert.

ERP - Enterprise Ressources Planning

Systeme zur internen Unternehmensplanung, -steuerung, -abwicklung und -kontrolle. Wird auch häufig als Back-Office - Anwendung bezeichnet.

Front Office

Unter Front Office-Anwendungen versteht man Einrichtungen und Applikationen in den Bereichen Marketing, Vertrieb und Service, die dem direkten Kundenkontakt dienen, beispielsweise Customer-Relationship-Management-Systeme.

Helpdesk

Wissensbasiertes Datenbanksystem, das ein Call Center bei der Beantwortung eingehender Fragen unterstützt. Weiter dient es zur Verfolgung und Analyse häufig auftretender Fragen und Problemen von Kunden.

Kampagnen-Management

Das Kampagnen-Management umfasst alle organisatorischen Aktivitäten, die zur Durchführung einer Marketing- und Verkaufsaktion (Kampagne) notwendig sind, Zusammenfassung meist mehrstufiger Verkaufsprozesse.

Kontakt-Management

Das Kontakt-Management umfasst alle Aktivitäten und Informationen, die bei einem Kontakt mit Kunden und Interessenten entstehen. Die Speicherung und Verwaltung erfolgt in einer Kundendatenbank. Kontaktmanagementsysteme sind Standard-Softwarepakete, welche sich neben der Adressen- und Terminverwaltung auf die Erfassung und Verwaltung der Kontakte konzentrieren.

Kundenbindung

Kundenbindung stellt eine Basis für die Erhaltung und Steigerung des langfristigen Unternehmenswertes dar. Sie beschreiben, inwieweit Kunden der oder den Marken des Unternehmens treu bleiben und Wiederholungskäufe tätigen.

Kundenzufriedenheit

In der Kundenzufriedenheit drückt sich aus, wie zufrieden der Kunde mit dem bei einem Unternehmen nachgefragten Produkt oder der Dienstleistung ist. Sie stellt eine notwendige, aber nicht hinreichende Bedingung für Kundenbindung dar.

Marketing Enzyklopädie

Eine Marketing Enzyklopädie ist ein elektronisches Wissensarchiv und Informationssystem für den Vertrieb und Service. Sie beinhaltet Informationen in verschiedenen Datenformaten (Text, Audi, Video) wie z.B. Produktinformationen, Verkaufspräsentationen oder Informationen über Marketingaktivitäten.

Marketing-Mix

Im Marketing-Mix ordnen sich die zur Marktbearbeitung zur Verfügung stehenden Instrumente in die vier Bereiche Distributions-, Kommunikations-, Preis- und Produktpolitik ein. Teilweise werden auch Produkt- und Preispolitik unter dem Begriff Angebotspolitik zusammengefaßt. Der Marketing-Mix eines Unternehmens beschreibt die konkrete Ausgestaltung der vom Unternehmen zur Marktbearbeitung ausgewählten Instrumente.

Neukundengewinnung

Die Neukundengewinnung ist wichtig für das Wachstum des Unternehmens. Dabei ist zu beachten, dass die "Alt"-Kunden nicht vernachlässigt werden. Die Ansprache der Neukunden und die Produktgestaltung darf sich allerdings nicht zu weit von den Bedürfnissen der "Alt"-Kunden entfernen, da ansonsten auf der einen Seite Kunden zugewonnen, aber auf der anderen Seite auch verloren werden.

Olap

Bei OLAP handelt es sich um eine Top-down-Methode, mit der man nach zuvor definierten Strukturen oder Dimensionen, etwa Zeitraum, Sparte oder Kundengruppe, in den Daten navigieren und sie "live on screen" auswerten kann. Ein typisches Einsatzfeld sind Bewertungen in verschiedenen Controlling-Bereichen, etwa die Kosten-Nutzen-Analyse konkreter Vorgehensweisen bei der Neukundengewinnung.

Personalisierung

Unter Personalisierung versteht man das Anzeigen von Inhalten abgestimmt auf den jeweiligen Benutzer, basierend auf den über den Benutzer gespeicherten Informationen.

Diese über den Benutzer gespeicherten Informationen können dabei z.B. Transaktions-Daten (Kaufhistorie, Benutzerverhalten auf dem Website, etc.), psychographische oder soziodemographische Informationen sein. Eng verbunden mit Personalisierung ist der Begriff Mass-Customization, der jedoch nicht nur das Maßschneidern von Web-Inhalten, sondern auch Produkte und Service auf den Konsumenten berücksichtigt.

Supply Chain Management

Lösungen zur Unterstützung des Einkaufs und des Beschaffungsmanagements. Bildet die Prozesskette eines Unternehmens in Richtung Lieferant ab.

Wissensdatenbank

In Wissensdatenbanken werden beispielsweise Fehler und deren Lösungsmöglichkeiten gespeichert, auf die bei Bedarf zugegriffen werden kann. Typische Anwendungen sind der Service und der Kundendienst.

Workflow Management

Ablauforganisation von Vorgängen und Geschäftsprozessen. Workflowmanagement-Systeme haben die Aufgabe, Büroprozesse zu unterstützen, d.h. die Bearbeitungsfunktionen aufzurufen und die jeweiligen Arbeitsschritte zu kontrollieren, so dass ein automatischer Arbeitsablauf gemäß den Prozesserfordernissen gestaltet wird. Sie erhöhen die Effizienz der Prozessbearbeitung und gewinnen deshalb an Bedeutung z.B. in der Reklamationsbearbeitung und an der Schnittstelle zwischen Vertriebsinnen- und Außendienst.