

SAP TUTORIAL

Lara Lebedinski, Siegfried Zeilinger, Dr. Thomas Grechenig
INSO TU Wien

19. Februar 2006

Inhaltsverzeichnis

1	Grundsätzlicher Aufbau des SAP Systems	5
1.1	Die Grundlagen des SAP-Systems	5
1.2	Die Navigation	5
1.2.1	Die Anmeldung und SAP GUI Design	5
1.2.2	Auswahl von Funktionen im SAP System	6
1.2.3	Möglichkeiten zur Hilfe	7
1.2.4	Möglichkeiten zur Personalisierung der Benutzeroberfläche	7
1.3	Der Systemkern	7
1.3.1	Client-Server Architektur	7
1.3.2	Präsentations- und Datenbankschnittstelle	10
1.3.3	Prozesse des SAP Web Application Server	11
1.3.4	Die SAP-Transaktion	11
1.3.5	Verbuchungsverlauf	12
1.3.6	Berechtigungen	12
1.3.7	Die Begriffserklärung der Benutzerverwaltung	12
1.3.8	Konzept und Ablauf der Berechtigungsprüfungen	13
1.3.9	Erstellen und Pflegen von Benutzerstammsätzen	13
1.4	Die Workbench	14
1.4.1	Datenstruktur eines SAP-Systems	14
2	Module	15
2.1	Rechnungswesen	15
2.2	Logistik	16
2.3	Personalwirtschaft	17
3	Prozesse im Vertrieb	17
3.1	Unternehmensstruktur im Vertrieb	17
3.2	Prozesse im Vertrieb	19
3.2.1	Vorverkaufsaktivitäten	19
3.2.2	Auftragsbearbeitung	19
3.2.3	Beschaffung	20
3.2.4	Versand	20
3.2.5	Fakturierung	21
3.2.6	Belegfluss	22
3.2.7	Zahlung	22
4	ABAP	22
4.1	Der Ablauf eines ABAP Programms	23
4.2	Einführung in die ABAP Workbench	24
4.2.1	Die ABAP Workbench und das Repository	24
4.2.2	Analysieren eines vorhandenen Programms	26
4.3	ABAP-Anweisungen und Datendeklarationen	28
4.3.1	Die ABAP-Typen	28
4.3.2	Die Datenobjekte	29
4.4	Lesen von der Datenbank	32
4.4.1	Datenbanktabellen im R/3	32
4.4.2	Lesen von einer Datenbanktabelle	33
4.5	Die Liste	33

4.6	Selektionsbilder	34
4.6.1	Eigenschaften von Selektionsbildern	34
4.7	Softwarelogistik	35

Abbildungsverzeichnis

1	SAP Easy Access	6
2	Möglichkeiten zur Personalisierung	8
3	Verarbeitung einer Benutzeranforderung	9
4	Ablauf einer Datenbankabfrage	10
5	Die Datenstruktur eines SAP-Systems	15
6	Der Vertriebsbereich	19
7	Der Belegfluss	23
8	Das Zusammenspiel zwischen Serverebenen	24
9	Ausgewählte Werkzeuge der ABAP Workbench	25
10	Die Systemlandschaft	35

Sämtliche Abbildungen wurden den Kursunterlagen zu den SAP Standard-
schulungen SAPTEC, LO150 und BC400 entnommen. Das Copyright für diese
Abbildungen liegt bei der SAP AG, Walldorf, Deutschland. Eine Genehmigung
der SAP zur Nutzung im Zusammenhang mit diesem Tutorial liegt uns vor.

1 Grundsätzlicher Aufbau des SAP Systems

Die Abkürzung SAP stand ursprünglich für “Systemanalyse Programm-Entwicklung” und heute steht sie für “Systeme, Anwendungen, Produkte in der Datenverarbeitung”.

Das Hauptprodukt der SAP AG ist das Software-System R/3, mit dem in Unternehmen Daten verwaltet und Prozesse gesteuert werden. Das System ermöglicht eine komplette betriebswirtschaftliche Steuerung der Firma.¹

1.1 Die Grundlagen des SAP-Systems

Die grundlegenden Bestandteile des SAP Systems bilden:

- SAP NetWeaver
Stellt die technische Infrastruktur für alle SAP Lösungen dar. Der SAP Netweaver besteht aus vier Komponenten: People Integration, Information Integration, Process Integration und Application Platform. Der SAP Web Application Server stellt, zusammen mit der Datenbank, die Application Platform des SAP NetWeaver dar.
- mySAP Business Suite
Die mySAP Business Suite umfasst alle branchenübergreifende, SAP NetWeaver-basierten Lösungen der SAP.
- SAP Smart Business Solutions
Die SAP Smart Business Solutions sind SAP-Lösungen für Klein- und Mittelstandsunternehmen.
- SAP xApps
Die SAP xApps ermöglichen es, verschiedene Anwendungen miteinander zu verknüpfen, indem man über offene Schnittstellen auf bereits vorhandene Datenbestände und Funktionen zugreift.
- Branchenlösungen
Branchenlösungen bieten betriebswirtschaftliche Funktionen für unterschiedliche Branchen.

1.2 Die Navigation

1.2.1 Die Anmeldung und SAP GUI Design

Auf SAP-Systeme kann über unterschiedlich gestaltete Frontend-Programme zugegriffen werden, für fast alle SAP Lösugen stellt jedoch das SAP GUI (SAP Graphic User Interface) den Standardzugang dar. Das Programm SAP GUI verbindet den Frontend-Rechner mit SAP-Systemen.

Bei der Anmeldung an ein SAP-System wird der User nach dem Benutzernamen und Kennwort gefragt. Weiters muss man bei der Anmeldung den Mandaten angeben und man kann die Sprache wählen.

¹Dieses Kapitel basiert auf den Schulungsunterlagen “SAP NetWeaver: Grundlagen der Application Platform”.

Ein *Mandant* entspricht in der Regel der Abbildung eines Unternehmens im SAP System. Das heißt, dass in einem SAP-System mit mehreren Mandanten mehrere Unternehmen abgebildet und parallel tätig sein können. Von einem Mandanten aus kann man nur die Daten genau dieses Mandanten sehen und auf diese zugreifen².

Mehrfachanmeldungen werden ab SAP R/3 4.6 aus sicherheitsbedingten und auch lizenzrechtlichen Gründen protokolliert.

Die Abbildung (1)³ zeigt das Standardeinstiegsbild in SAP-Systeme. Im linken Bildbereich kann man eine Baumdarstellung der zur Verfügung stehenden Menüs des SAP-Systems, im rechten Bildbereich kann man ein Logo anzeigen lassen.

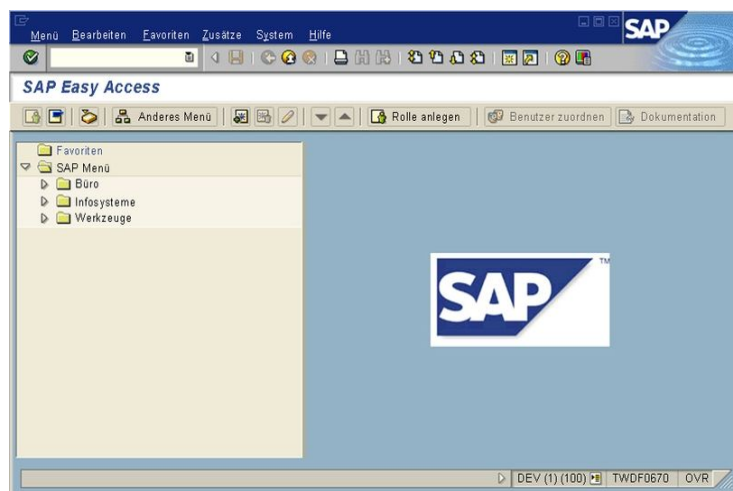


Abbildung 1: SAP Easy Access

1.2.2 Auswahl von Funktionen im SAP System

Nach der erfolgreichen Anmeldung stehen dem Benutzer im linken Bildbereich zwei untereinander angeordnete Übersichtsbäume zur Funktionswahl zur Verfügung:

- die benutzerdefinierte Favoritenliste
Die Favoritenliste beinhaltet Funktionen des SAP-Systems oder auch Links zu Internet-Inhalten oder auf Dateien des Frontend-Rechners. Die anfangs leere Favoritenliste ist von jedem Endanwender individuell editierbar und nur für diesen sichtbar. Favoriten können im SAP-Easy-Access-Bild über das Menü Favoriten bearbeitet werden.
- das rollenbasierte Benutzermenü oder das SAP-Menü
Das rollenbasierte Benutzermenü wird aufgrund der dem Benutzer zugeordneten Rolle(n) erstellt und auf das Frontend übertragen.

²siehe auch Definition Seite 11, Abschnitt 1.5.1

³entnommen aus "Application Platform Grundlagen"

In SAP-Systemen hat man 3 Möglichkeiten zur Navigation:

1. über die Eingabe von Transaktionscodes im Kommandofeld
2. über die Einträge der Menüs in der Menüleiste
3. über Einträge der Favoriten oder des Benutzer- oder SAP-Menüs

1.2.3 Möglichkeiten zur Hilfe

Mit der F1-Taste erhält man Erläuterungen zu Feldern, Menüs, Funktionen und Meldungen. Über die F1-Hilfe gelangt man auch zu technischen Informationen zum jeweiligen Feld.

Mit der F4-Taste erhält man Informationen zu möglichen Eingabewerten. Wenn Felder mit einem "Haken"-Symbol belegt sind, kann man in der jeweiligen Anwendung nur nach Eingabe eines zugelassenen Wertes fortfahren.

Die SAP-Bibliothek hilft beim Kennenlernen von Systemfunktionen. Sie bietet den Zugriff auf die Online-Dokumentation. Die dort abgelegten Informationen beschreiben nicht nur die Nutzung von Systemfunktionen, sondern erklären auch Konzepte der Systemarchitektur, geben Beispiele für mögliche sinnvolle Konfigurationen verschiedenster Prozesse und warnen vor möglichen Bedienungsfehlern und deren Folgen.

1.2.4 Möglichkeiten zur Personalisierung der Benutzeroberfläche

Im System stehen den Endanwendern vielfältige Personalisierungsmöglichkeiten zur Verfügung. Unter *Zusätze ->Einstellungen* kann man die Gestaltung des Einstiegsbilds beeinflussen, z.B.: über eine Ausschaltung des Bilds im rechten Teilbereich des Fensters oder über eine Zuschaltung der technischen Namen (Transaktionscodes) im Bild SAP Easy Access.

Unter *Anpassung des lokalen Layouts* findet man z.B.: unter *Optionen... ->Lokale Daten* die Möglichkeit, die Eingabehistorie zu verwalten. Die Eingabehistorie baut, wenn sie aktiviert ist, eine kleine Datenbank auf dem Frontend auf, die für die Eingabefelder in Transaktionen die letzten x Eingaben beinhaltet.

Unter den *Optionen...* findet man auch die Möglichkeit, die Geschwindigkeit der Quick Info einzustellen, oder Systemmeldungen als Dialogfenster anzeigen zu lassen (*Optionen... ->Nachrichten*). Es gibt hier noch viele weitere kleine Hilfen und Einstellungsmöglichkeiten, wie z.B.: die Wahl des Farbschemas des GUI-Erscheinungsbildes.

Wo die Möglichkeiten zur Personalisierung im SAP System zu finden sind, ist in der Abbildung (2)⁴ abgebildet.

1.3 Der Systemkern

1.3.1 Client-Server Architektur

In Zusammenhang mit SAP-Systemen werden die Begriffe Client und Server zumeist aus softwareorientierter Sicht verwendet. In der softwareorientierten Sicht werden Client und Server jeweils auf Prozessebene (Service) definiert. Ein

⁴entnommen aus "Application Platform Grundlagen"

Service ist in diesem Zusammenhang ein Dienst, der von einer Softwarekomponente angeboten wird. Eine Solche Softwarekomponente kann aus einem Prozess (vgl. Workprozess) oder einer Gruppe von Prozessen (vgl. SAP Web Application Server) bestehen, und heißt dann Server für den jeweiligen Dienst. Softwarekomponenten die einen Dienst in Anspruch nehmen, werden entsprechend Clients genannt. Die Clients können wiederum gleichzeitig Server für bestimmte andere Dienste sein.

Um eine betriebswirtschaftliche Anwendungssoftware betreiben zu können, werden folgende Prozesse benötigt:

- Präsentationsprozesse - z.B.: zur Aufbereitung der Bildschirmbilder
- Applikationsprozesse - z.B.: zum Ausführen von Anwendungsprogrammen
- Datenbankprozesse - z.B.: zur Verwaltung und Organisation von Daten

Im Rahmen der Konfiguration eines SAP-Systems muss die Frage beantwortet werden, wie die benötigten Prozesse auf die zur Verfügung stehende Software verteilt werden. Je nach Anzahl der verwendeten Rechnerebenen spricht man von ein- oder mehrstufiger Konfiguration.

Bei einer einstufigen Konfiguration werden alle Verarbeitungsaufgaben (Datenbank-, Applikations- als auch Präsentationsprozesse) von einem Rechner erbracht. Zweistufige Konfigurationen werden typischerweise mit speziellen Präsentations-Servern realisiert, die ausschließlich für die Aufbereitung der graphischen Oberfläche zuständig sind. Bei einer dreistufigen Konfiguration werden jeweils eigene Rechner für die drei Schichten genutzt. Mit den Daten eines Datenbank-Servers können viele verschiedene Applikationsserver gleichzeitig arbeiten.

Die Bildschirmeingaben eines Benutzers werden vom SAP-Präsentationsprogramm SAP GUI entgegengenommen in ein geeignetes Format formatiert und an den Web Application Server (softwareorientierte Sicht) geschickt.

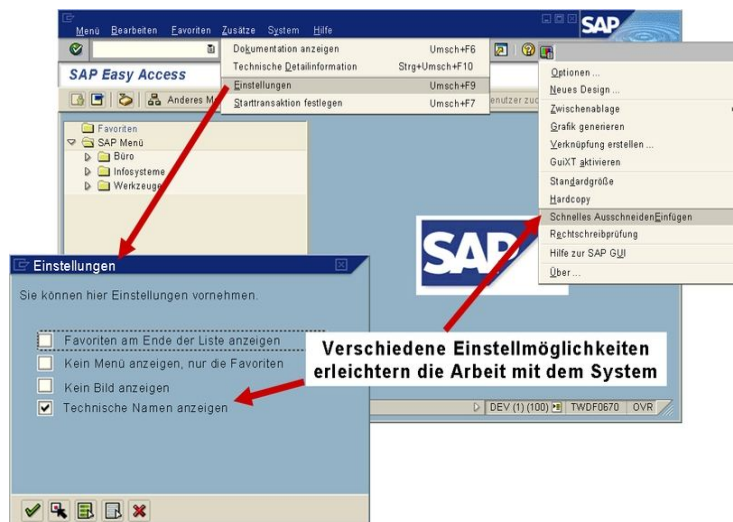


Abbildung 2: Möglichkeiten zur Personalisierung

Der zentrale Prozess des SAP Web Application Server ist der **Dispatcher**. Er verwaltet in Abstimmung mit dem jeweiligen Betriebssystem die Ressourcen für die in ABAP geschriebene Applikationen. Zu den Hauptaufgaben des Dispatchers gehört die Verteilung der Transaktionslast auf die Workprozesse, die Anbindung der Präsentationsebene und die Organisation von Kommunikationsvorgängen.

Die Verarbeitungsanforderungen werden zunächst in Request-Queues gespeichert, die anschließend nach dem Prinzip "first in - first out" abgearbeitet werden.

Der Dispatcher verteilt die Requests nacheinander auf freie Workprozesse. Im **Workprozess** findet die eigentliche Verarbeitung statt, wobei der Benutzer, der die Anforderungen über das SAP GUI eingegeben hat, nicht immer den gleichen Workprozess zugewiesen bekommt. Es gibt also keine feste Zuordnung von Workprozessen zu Benutzern. Zur Abarbeitung der Benutzeranforderungen ist es gegebenenfalls notwendig, Daten aus der Datenbank zu lesen bzw. in die Datenbank zu schreiben. Hierzu ist jeder Workprozess direkt mit der Datenbank verbunden.

Das gesamte Zusammenspiel der Präsentations-, Applikations- und Datenbankebene ist in der Abbildung (3)⁵ dargestellt.

Am Ende der Verarbeitung gelangt das Verarbeitungsergebnis des Workprozesses über den Dispatcher an das SAP GUI zurück. Das SAP GUI interpretiert die empfangenen Daten und erzeugt in Zusammenarbeit mit dem Betriebssystem des Frontend-Rechners das Ausgabebild für den Benutzer.

Die Puffer dienen zur schnelleren Bearbeitung der Benutzeranforderungen. Daten die oft gelesen, aber nur selten geändert werden, können als Kopie des Datenbankinhaltes im Shared Memory des Applikationsservers gehalten werden.

⁵entnommen aus "Aplication Platform Grundlagen"

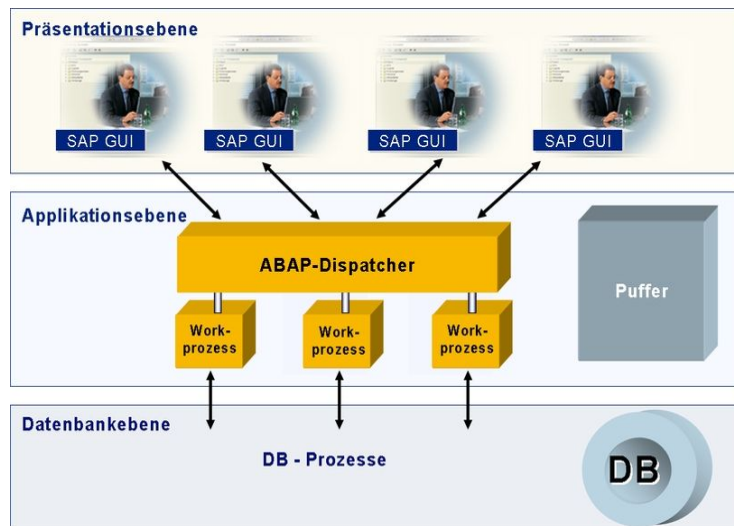


Abbildung 3: Verarbeitung einer Benutzeranforderung

1.3.2 Präsentations- und Datenbankschnittstelle

Mit Hilfe der Präsentationsschnittstelle, dem SAP GUI, kann der Anwender in Interaktion mit dem SAP-System treten und Daten eingeben bzw. sich anzeigen lassen.

Das SAP GUI realisiert die grafische Bedienoberfläche mit Hilfe der Mittel, die die jeweilige Präsentationsumgebung zur Verfügung stellt. Es orientiert sich primär am Windows-Style-Guide und wird auf verschiedenen Plattformen mit weitestgehend gleicher Funktionalität angeboten.

Für die Verwaltung großer Datenmengen werden heute zumeist Relationale Datenbankmanagementsysteme (RDBMS) verwendet. Diese speichern Daten und Beziehungen zwischen Daten in Form zweidimensionaler Tabellen. Auf Datenbankebene erfolgt die Definition von Daten, Tabellen und Tabellenbeziehungen im Datenbankkatalog (Data Dictionary) des RDBMS.

Innerhalb der SAP-Programmiersprache ABAP kann mittels SAP Open SQL (Structured Query Language) unabhängig vom jeweiligen RDBMS, auf die Anwendungsdaten der Datenbank zugegriffen werden. Die Datenbankschnittstelle, wie sie in der Abbildung (4)⁶ dargestellt ist, ist ein Bestandteil eines jeden Workprozesses des SAP Web Application Server. Sie sorgt für die Umsetzung der Open SQL-Anweisungen aus ABAP in entsprechende SQL-Anweisungen der speziell verwendeten Datenbank ("Native SQL"). Somit können ABAP Programme datenbankunabhängig programmiert werden.

Bei der Interpretation der Open SQL-Anweisungen führt die SAP-Datenbank-Schnittstelle eine Syntaxüberprüfung durch und sorgt automatisch für die optimale Ausnutzung der SAP Puffer, die lokal im Shared Memory eines jeden SAP Web Application Server liegen.

Weiterhin besteht in ABAP die Möglichkeit direkt "native" SQL-Befehle abzusetzen.

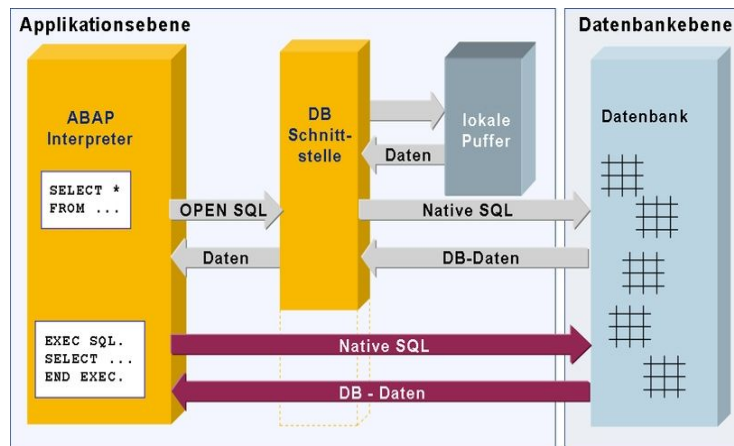


Abbildung 4: Ablauf einer Datenbankabfrage

⁶entnommen aus "Application Platform Grundlagen"

1.3.3 Prozesse des SAP Web Application Server

Das SAP-Laufzeitsystem besteht aus einer Menge parallel, kooperierender Prozesse. Zu diesen Prozessen gehört auf jedem Applikationsserver der Dispatcher sowie eine von den Ressourcen abhängige Anzahl von Workprozessen:

- *Dialog-Workprozesse* erledigen alle Aufträge zum Ausführen von Dialogschritten, die durch einen aktiven Benutzer ausgelöst werden. Jeder Dispatcher benötigt mindestens zwei Dialog-Workprozesse.
- *Spoolworkprozesse* geben sequentielle Datenströme an Drucker weiter. Pro SAP-System wird mindestens ein Spool-Workprozess benötigt, es ist mehr als einer pro Dispatcher möglich.
- *Verbuchungs-Workprozesse* führen Verbuchungsaufträge aus. Analog zu Spool-Workprozessen wird mindestens ein Verbuchungs-Workprozess pro SAP-System benötigt und es ist mehr als einer pro Dispatcher erlaubt.
- *Hintergrund-Workprozesse* verarbeiten Programme, die ohne Benutzerinteraktion ausgeführt werden sollen. Pro SAP-System benötigt man mindestens einen Hintergrund-Workprozess.
- Der *Enqueue-Workprozess* verwaltet die Sperrtabelle im Shared Memory. Die Sperrtabelle enthält die logische Datenbanksperren des SAP-Systems. Es ist nur ein Enqueue-Workprozess pro System nötig.

1.3.4 Die SAP-Transaktion

Transaktionen sind funktional zusammenhängende Verarbeitungseinheiten. Sie werden durch vier wesentliche Eigenschaften charakterisiert. Die Anfangsbuchstaben der englischen Buchstaben lassen sich zu einem Akronym **ACID** zusammensetzen.

- atomic - atomar
Eine Transaktion gelingt entweder vollständig oder bleibt ohne Auswirkung. Bei einem Ausfall eines transaktionsorientierten Systems muss also sichergestellt sein, dass niemals inkonsistente Teilergebnisse zurückbleiben.
- konsistent - consistent
Das System wird von einem betriebswirtschaftlich konsistenten, korrekten Zustand in einen (anderen) betriebswirtschaftlich konsistenten, korrekten Zustand überführt.
- isoliert - isolated
Die innerhalb einer Transaktion durchgeführten Änderungen können von anderen, gegebenenfalls parallel laufenden Transaktionen erst nach dem endgültigen Bestätigen ("Commit") gesehen werden (isoliert).
- dauerhaft - durable
Die Ergebnisse einer Transaktion sind dauerhaft, d.h. nach dem endgültigen Bestätigen fest in der Datenbank gespeichert.

Eine Datenbanktransaktion ist gemäß ACID-Prinzip eine nicht teilbare Folge von Datenbankoperationen, an deren Anfang und Ende ein konsistenter Datenbestand auf der Datenbank stehen muss. Anfang und Ende einer Datenbanktransaktion werden durch einen Commit-Befehl an das Datenbanksystem definiert. Während einer Datenbanktransaktion (d.h. zwischen zwei Commit Befehlen) sorgt das Datenbanksystem selbst für die Konsistenz des Datenbestands. Das Datenbanksystem übernimmt hierbei selbst die Aufgabe, den alten Zustand nach einem fehlerhaften Abbruch wieder herzustellen.

Betriebswirtschaftliche Transaktionen sind funktional zusammengehörige Verarbeitungseinheiten, die konsistente, betriebswirtschaftlich zulässige Datenbankänderungen durchführen. Typische Beispiele sind Soll- und Habenbuchungen, die nur gemeinsam sinnvoll sind; ebenso bedingen sich das Anlegen eines Auftrags und die Reservierung der betroffenen Materialien gegenseitig. Entsprechend wird eine SAP-Transaktion als unteilbarer Geschäftsprozess, der komplett oder überhaupt nicht durchgeführt werden soll, definiert. Eine SAP-Transaktion ist als Folge betriebswirtschaftlich konsistenter, logisch zusammenhängender Dialogschritte implementiert. Ein Benutzer-Dialogschritt wird jeweils durch ein Bildschirmbild repräsentiert.

1.3.5 Verbuchungsverlauf

In der Konfiguration eines SAP-Systems existieren sowohl Dialog-Work-prozesse als auch mindestens ein Verbuchungs-Workprozess. In den Verbuchungs-Workprozessen wird die Verbuchung, d.h. die Änderung der Einträge von Datenbanktabellen, durchgeführt.

Um Datenkonsistenz zu gewährleisten, muss die Verbuchung der Daten einer SAP-Transaktion entweder ganz oder überhaupt nicht ausgeführt werden.

Die asynchrone Verbuchung löst die Probleme, die durch unterschiedliche Bedeutung einer Transaktion auf Datenbankebene und SAP-Ebene verursacht werden. Durch das Bündeln der zu einer SAP-Transaktion gehörenden Verbuchungen in einer einzigen Datenbanktransaktion besteht die Möglichkeit eines vollständigen Rollbacks der zu dieser SAP-Transaktion gehörenden Daten.

1.3.6 Berechtigungen

Für die Anmeldung am SAP-System ist es notwendig, dass der Benutzer einen Benutzerstammsatz besitzt. Über den Benutzerstammsatz wird festgelegt, welche Rolle der Benutzer im SAP System hat, d.h. welche Aktivitäten der Benutzer im System ausführen darf. Benutzerstammsätze sind mandantenabhängig, dieses bedeutet, dass für jeden Mandaten im System eigene Benutzerstammsätze angelegt werden müssen.

1.3.7 Die Begriffserklärung der Benutzerverwaltung

Da die Begriffe der Benutzerverwaltung sehr ähnlich sind und leicht verwechselt werden können, werden sie im folgenden erklärt.

- **Benutzername**

Der Benutzername ist eine Zeichenfolge, mit der sich eine Person in einem Mandanten des SAP Systems authentifizieren kann. Unter diesem Benut-

zernamen finden die Systemaktivitäten statt, z.B.: Aufruf einer Transaktion.

- Benutzerstammsatz

Der Benutzerstammsatz setzt sich zusammen aus den Benutzernamen und den Benutzerdaten.

- Benutzerdaten

Die Benutzerdaten sind die Daten, die zu einem bestimmten Benutzernamen gehören.

- Berechtigung

Eine Berechtigungen ist die Erlaubnis zur Durchführung von Aktivitäten im SAP-System. Für verschiedene Aktionen sind unterschiedliche Berechtigungen notwendig.

- Berechtigungsprofil

Das Berechtigungsprofil ist eine Sammlung von Berechtigungen. Die SAP bietet eine große Anzahl an Einzel- und Sammelprofilen.

- Rolle

Die Rolle ist eine funktionale Beschreibung eines SAP-Users mit dem dazugehörigen Berechtigungsprofil und den Menüeinträgen des Benutzermenüs. Die Rollen werden im SAP System zentral administriert. Eine Rolle kann mehreren Benutzern zugewiesen werden, ein Benutzer kann mehrere Rollen haben.

1.3.8 Konzept und Ablauf der Berechtigungsprüfungen

Eine Rolle beinhaltet Berechtigungen für das SAP-System und Menüeinträge für das Benutzermenü der Benutzer. Die einzelnen Rollen werden vom Systemadministrator erstellt und verwaltet. Wenn einem Benutzer eine Rolle zugeordnet wird, dann wird diese Rolle in seinen Benutzerstammsatz eingetragen und er erhält die Berechtigungen dieser Rolle. Zusätzlich werden in seinem Benutzermenü die Transaktionen aufgelistet, zu denen er durch diese Rolle Berechtigungen hat. Einem Benutzer können mehrer Rollen zugeordnet werden.

Bei der Anmeldung des Benutzers am SAP-System werden die Berechtigungen aus den Rollen, die im Benutzerstammsatz hinterlegt sind, in den Benutzerkontext eingetragen. Der Benutzerkontext liegt im Shared Memory des Applikationsservers, auf dem der Benutzer angemeldet ist.

Wenn der Benutzer eine Transaktion aufruft, so wird vom Workprozess überprüft, ob er die Berechtigungen zum Start dieser Transaktion hat. Wenn der Benutzer keine Berechtigung hat, erhält er eine Fehlermeldung. Hat er die Berechtigung, so wird die Maske für diese Transaktion dargestellt.

1.3.9 Erstellen und Pflegen von Benutzerstammsätzen

Das Erstellen und Pflegen der Benutzerdaten findet man im Menü unter *Werkzeuge -> Administration -> Benutzerpflege -> Benutzer* oder man ruft die Transaktion SU01 auf.

1.4 Die Workbench

1.4.1 Datenstruktur eines SAP-Systems

SAP-Systeme haben eine spezifische Datenstruktur. Neben den betriebswirtschaftlichen Einstellungen (Customizing), die nur für bestimmte Mandanten des SAP-Systems relevant sind, enthält ein SAP-System immer auch anwendungsübergreifende Einstellungen und Objekte.

Im SAP-System unterscheidet man drei Datenkomponenten:

- Ein **Mandant** ist eine betriebswirtschaftliche, organisatorisch und datentechnisch abgeschlossene Einheit innerhalb des SAP-Systems. Ein Mandant zeichnet sich auch durch ein eigenes betriebswirtschaftliches Datenumfeld, eigene Stamm- und Bewegungs- und eigene Benutzerdaten. Diese Daten eines Mandanten werden als **mandantenabhängige Daten** bezeichnet.
- Das **Customizing** eines Mandanten wird bei der Einführung eines SAP-Systems festgelegt. Über Customizing-Einstellungen werden Organisationsstrukturen des Kunden, wie Vertriebswege, Buchungskreise, etc. festgelegt und die SAP-Transaktion so eingestellt, dass sie die kundenspezifischen Abläufe wiedergeben. Neben den mandantenabhängigen Customizing-Einstellungen, existieren auch jedoch Einstellungen, die - einmal durchgeführt - für alle Mandanten des SAP-Systems Bedeutung haben (z.B.: Einstellung des Feiertagskalenders).
- Das **Repository**, die zentrale Ablage für alle Entwicklungsobjekte der ABAP-Workbench, ist ebenfalls mandantenabhängig. Es beinhaltet alle Objekte des Dictionary (Tabellen, Datenelemente, Domänen, etc.) und darüber hinaus auch alle ABAP-Programme, Menüs, Bildschirmbilder - ("Screens"). Aufgrund ihrer Mandantenunabhängigkeit werden die aus einem Mandanten heraus entwickelten oder geänderten Repository-Objekte in allen anderen Mandanten des Systems in identischer Form verwendet.

In der Abbildung (5)⁷ kann man die Datenstruktur mit mandantenabhängigen, mandantenunabhängigen und Customizing Daten sehen.

Der Anpassungsprozess der SAP-Software an ein konkretes Unternehmen bezeichnet man als Customizing. Dieser Prozess umfasst mandantenabhängige als auch mandantenunabhängige Einstellungen.

Änderungen an Repository-Objekten sind nicht unbedingt notwendig, gegebenenfalls können jedoch einzelne Objekte angepasst werden. Änderungen und Erweiterungen des Repository sind in unterschiedlicher Form möglich:

- Erweiterungen des Repository durch **Eigenentwicklungen**. Im SAP-System ist es möglich, eigene Repository-Objekte wie Tabellen, Programme, Transaktionen, etc. anzulegen. Alle Entwicklungsarbeiten von Kundenseite geschehen typischerweise im so genannten Kundennamensraum, d.h. alle von Kunden angelegten Objekte haben Namen aus einem vorgegebenen Namensraum, die typischerweise mit den Buchstabe Y oder Z beginnen. Dies gilt für ABAP-Programme, Tabellen, etc.

⁷entnommen aus "Application Platform Grundlagen"

- Anpassen des Repository durch **Kundenerweiterungen**. Hier wird das Repository durch kundeneigene Objekte ergänzt.
- **Modifikationen** am SAP-Standard: Änderungen an SAP-Objekten, wie Programme und Tabellendefinitionen werden als Modifikationen bezeichnet. Das von SAP ausgelieferte Repository wird nicht nur ergänzt, sondern geändert. Diese Modifikationen sind beim nächsten System-Upgrade bzw. beim Einspielen von Support Packages mit dem dann neuen Repository abzugleichen.

2 Module

Wenn eine Firma sich für die Einführung des SAP Systems entscheidet, muss auch die Entscheidung getroffen werden, welche SAP Module erworben werden sollen.

Die SAP Module werden in drei Gruppen eingeteilt: Rechnungswesen, Logistik und Personalwirtschaft. Zusätzlich gibt es auch Branchenlösungen, die speziell auf die jeweilige Branche zugeschnitten sind. Im Folgenden werden die wichtigsten Module des SAP R/3 Systems aufgezählt und kurz erklärt. Dies ist also *keine* vollständige Liste der Module.⁸

2.1 Rechnungswesen

Finanzwesen (FI) Die Komponente Finanzwesen erfüllt die internationalen Anforderungen, die an ein externes Rechnungswesen einer Unternehmung gestellt werden, über einen offenen, integrierten Datenfluss und vereinfacht die finanzwirtschaftliche Entscheidungsfindung in hohem Maße. Im R/3-System finden zwischen den verschiedenen Anwendungen alle Buchungen in Echtzeit statt,

⁸Die Informationen für dieses Kapitel wurden aus <http://sap.help.com> entnommen.

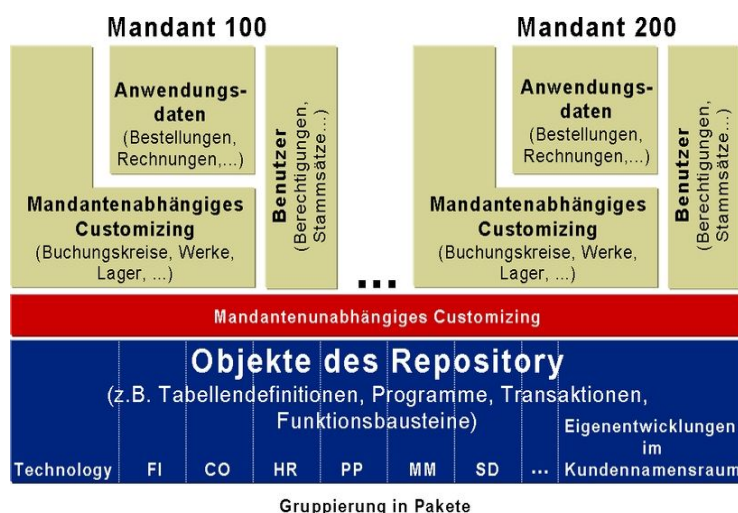


Abbildung 5: Die Datenstruktur eines SAP-Systems

z.B. zieht eine Buchung im Nebenbuch "Kreditoren" automatisch eine Gegenbuchung im Hauptbuch nach sich.

Controlling (CO) Das Controlling stellt Informationen für Entscheidungen des Managements bereit. Es dient der Koordination, Überwachung und Optimierung aller ablaufenden Prozesse innerhalb eines Unternehmens. Dazu werden der Verbrauch an Produktionsfaktoren sowie die vom Unternehmen erbrachten Leistungen erfasst.

Projektsysteme (PS) Innerhalb des Projektsystems können Projekte abgebildet und verwaltet werden. Das Projektsystem bildet dabei sowohl die technische als auch die kaufmännische Sicht auf das Projekt ab. Großangelegte Projekte, wie z.B. die Planung einer Messe, erfordern eine genaue Planung der zahlreichen Detailaktivitäten. Der Projektverantwortliche muss die termin-, kosten- und leistungsgerechte Durchführung des Projekts sicherstellen und die notwendigen Ressourcen und Finanzmittel optimal bereitstellen.

2.2 Logistik

Produktionsplanung und Steuerung (PP) Das PP-System deckt sämtliche betriebswirtschaftlichen Funktionen im Bereich der Produktionsplanung und -steuerung ab. Die Funktionen der Produktionsplanung und -steuerung bieten eine komplexe Lösung, um sowohl das Produktionsprogramm (Art und Menge der Erzeugnisse) als auch die Fertigung zu planen. Die unmittelbare Vorbereitung der Fertigung schließt die Beschaffung, Lagerung und den Transport von Materialien und Zwischenprodukten ein.

Vertrieb (SD) Das Vertriebssystem Sales and Distribution stellt eine branchenneutrale Gesamtlösung für die Aufgaben des Verkaufs, des Versands und der Fakturierung dar. SD ist als integrierter Baustein des R/3-Systems in den gesamten betriebswirtschaftlichen Ablauf integriert und setzt sich aus folgenden zentralen Komponenten zusammen: Vertriebsunterstützung, Verkauf, Versand, Transport, Außenhandel und Fakturierung. Dieses Modul wird im Kapitel 3 genauer beschrieben.

Materialwirtschaft (MM) Die Komponente Materialwirtschaft besteht im wesentlichen aus den Teilkomponenten Einkauf, Bestandsführung, Lagerverwaltung und Rechnungsprüfung. Mit dem Einkaufssystem können Bedarfe an Materialien oder Dienstleistungen, weitgehend automatisch in Bestellungen umgesetzt werden. Der Einkäufer muss nur in Ausnahmefällen eingreifen. Mit der Bestandsführung erfolgt die Abbildung der physischen Bestände durch die Echtzeit-Erfassung aller bestandsverändernden Vorgänge und die daraus resultierenden Bestandsfortschreibungen.

Qualitätsmanagement (QM) Die Anwendungskomponente Qualitätsmanagement unterstützt die Aufgaben der Qualitätsplanung, Qualitätsprüfung und Qualitätslenkung. Mit den Funktionen des Moduls Qualitätsmanagement kann

man die wesentlichen Elemente eines QM-Systems, z.B. nach ISO 9000, verwirklichen. Andere im R/3-System integrierte Module (z.B. MM, SD, PP) ergänzen diese Funktionen.

Instandhaltung (PM) Die Komponente beinhaltet die Funktionen für die Instandhaltung; d.h. die Inspektion, die Wartung und die Instandsetzung. Die Instandhaltungsabwicklung umfasst mehrere Stufen, die jedoch nicht alle zwingend und in vollem Umfang durchlaufen werden müssen.

2.3 Personalwirtschaft

Das System Human Resources besteht im wesentlichen aus den Komponenten Personalmanagement, Personalzeitwirtschaft und Personalabrechnung. Aus diesen Komponenten können die Daten in FI und in CO übergeben werden. Leistungsdaten können bei Rückmeldungen online in PP und CO gebucht werden.

3 Prozesse im Vertrieb

Dieses Kapitel beschäftigt sich mit dem Modul Vertrieb. Es beschreibt die Prozesse, die im Vertrieb ablaufen, und die Daten die in Prozessen verarbeitet werden.⁹

3.1 Unternehmensstruktur im Vertrieb

Die Definition der Organisationsstruktur im R/3-System ist ein wichtiger Schritt. Dafür ist eine genaue Analyse der Geschäftsprozesse im Unternehmen erforderlich.

Für die Abbildung der Struktur einer Unternehmensorganisation stehen dem Benutzer im R/3-System Organisationseinheiten zur Verfügung. Eine Organisationseinheit kann eine organisatorische und/oder eine juristische Gegebenheit in einem Unternehmen darstellen. Das Vertriebsmodul nutzt eine Reihe von Organisationseinheiten, die ausschließlich zur Abbildung des Vertriebs vorgesehen sind, wie z.B. Verkaufsorganisation, Vertriebsweg und Sparte.

Im Modul Vertrieb unterscheidet man zwischen folgenden Organisationseinheiten:

- Buchungskreis
- Vertriebsbereich
 - Verkaufsorganisation
 - Vertriebsweg
 - Sparte
- Werk
- Lagerort
- Versandstelle

⁹Die Informationen für dieses Kapitel wurden aus "Prozesse im Vertrieb" entnommen.

Buchungskreis Der Buchungskreis ist die kleinste organisatorische Einheit des externen Rechnungswesens. Der Buchungskreis ist eine rechtlich selbstständige und bilanzierende Einheit. Ein Buchungskreis kann z.B. ein Unternehmen innerhalb einer Konzerns oder eine Tochtergesellschaft sein.

Verkaufsorganisation Die Verkaufsorganisation ist eine organisatorische Einheit der Logistik, die das Unternehmen nach den Erfordernissen des Vertriebs gliedert. Eine Verkaufsorganisation ist verantwortlich für den Vertrieb von Materialien und Dienstleistungen. Mit Hilfe der Verkaufsorganisation kann man eine regionale, nationale oder internationale Gliederung des Marktes berücksichtigen. Eine Verkaufsorganisation muss **eindeutig** einem Buchungskreis zugeordnet werden. Für eine Verkaufsorganisation pflegt man eigene Stammdaten.

Vetriebsweg Der Vertriebsweg kennzeichnet den Weg, auf dem verkaufsfähige Materialien oder Dienstleistungen zum Kunden gelangen. Verkaufsorganisationen können mehrere Vertriebswege zugeordnet werden. Beispiele für Vertriebswege sind Großhandel, Einzelhandel, Internethandel, ...

Sparte Mit der Sparte kann man Materialien und Dienstleistungen gruppieren. Einer Verkaufsorganisation können mehrere Sparten zugeordnet werden, für die sie zuständig ist. Eine Sparte kann z.B. eine bestimmte Produktgruppe abbilden.

Vetriebsbereich Der Vertriebsbereich ist, wie man in der Abbildung (6)¹⁰ sehen kann, eine Kombination von Verkaufsorganisation, Vertriebsweg und Sparte. Er legt fest über welchen Vertriebsweg eine Verkaufsorganisation Produkte einer Sparte vertreiben kann. Jeder Vertriebsbeleg ist genau einem Vertriebsbereich zugeordnet. Ein Vertriebsbereich kann nur zu einem Buchungskreis gehören. Bei der Bearbeitung von Vertriebsbelegen werden abhängig vom Vertriebsbereich auf verschiedene Stammdaten zugegriffen werden. Dazu gehören z.B. Kundenstammdaten, Materialstammdaten, Preise und Rabatte.

Werk, Lagerort Das Werk ist ein Ort mit Materialbestand. Ein Werk kann z.B. eine Produktionsstätte im R/3-System abbilden.

Werk und Lagerort sind organisatorische Einheiten, die von allen logistischen Bereichen des R/3-Systems genutzt werden können:

- Die Materialwirtschaft konzentriert sich hauptsächlich auf den Materialfluss. Aus Sicht der Materialwirtschaft ist ein Werk in erster Linie ein Ort mit Materialbestand.
- In der Produktion kann ein Werk eine Fertigungseinrichtung abbilden.
- Im Vertrieb stellt das Werk denjenigen Ort dar, von dem Materialien und Dienstleistungen ausgeliefert werden und entspricht somit einem Verteilerzentrum.

Im Vertrieb spielt das Werk eine zentrale Rolle

¹⁰entnommen aus "Prozesse im Vertrieb"

- Um das Vertriebsmodul nutzen zu können muss mindestens ein Werk angelegt werden.
- Ein Werk muss eindeutig einem Buchungskreis zugeordnet werden.

Versandstelle Der Versand ist ein integrierter Baustein der Vertriebsabwicklung. Die Versandstelle ist die höchste organisatorische Einheit des Versands und ist für die Steuerung der Versandaktivitäten verantwortlich. Die organisatorische Zuordnung der Versandstelle im R/3-System erfolgt über das Werk. Die Versandstelle kann z.B. eine Lkw-Laderampe, oder Poststelle oder ein Güterbahnhof sein.

3.2 Prozesse im Vertrieb

Eine effektive Abwicklung von Kundenaufträgen bindet alle Vertriebsaktivitäten in eine Kette eng integrierter Prozesse ein. Der R/3-Vertriebsprozess beginnt mit dem Aufbau und der Pflege von Kundenbeziehungen und endet mit der Fakturierung der gelieferten Waren oder erbrachten Leistungen an den Kunden.

3.2.1 Vorverkaufsaktivitäten

Unter Vorverkaufsaktivitäten versteht man den Aufbau und die Pflege von Kundenbeziehungen. Die Vorverkaufsaktivitäten umfassen das Erfassen und Verfolgen von Kundenkontakten, Mailing-Aktionen, das Beantworten von telefonischen Kundenfragen, ...

3.2.2 Auftragsbearbeitung

Die Mitarbeiter der Abteilung Kundenservice nehmen schriftliche und telefonische Aufträge entgegen. Ein Kundenauftrag ist ein elektronischer Beleg, in dem die vom Kunden bestellten Waren oder Dienstleistungen erfasst werden. Terminaufträge enthalten normalerweise:

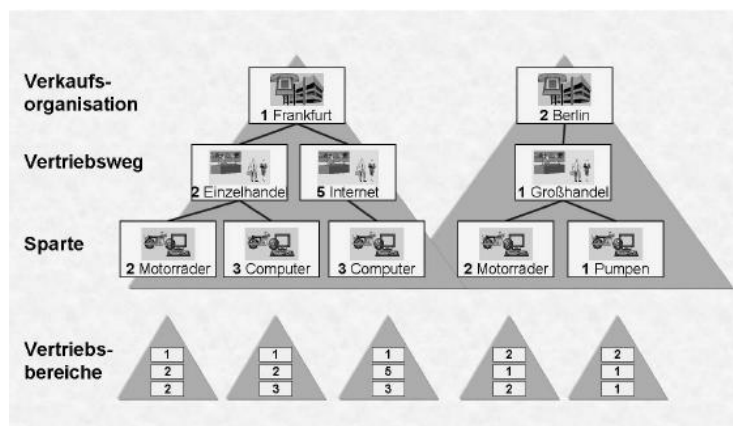


Abbildung 6: Der Vertriebsbereich

- Informationen über den Kunden und über die Ware
- Preisinformationen zu allen Positionen
- Informationen zu Lieferterminen und Liefermengen
- Informationen für die Versandabwicklung
- Informationen für die Fakturierung

Auftrag anlegen Ein Verkaufsbeleg wird immer innerhalb eines Vertriebsbereichs angelegt. Ein Auftrag bezieht sich auf einen oder mehrere Angebote.

Struktur des Verkaufsbelegs Ein Verkaufsbeleg ist in drei Ebenen Kopf, Positionen und Einteilung gegliedert.

- Kopf des Verkaufsbelegs
Die Daten des Belegkopfes gelten für den gesamten Beleg.
- Positionen
Jede Position im Verkaufsbeleg enthält eigene Daten. Dazu gehören z.B. Angaben zum Material. Jeder Verkaufsbeleg kann mehrere Positionen haben.
- Einteilung der Belegpositionen
Einteilungen enthalten Liefermengen und Liefertermine. Sie gehören eindeutig zu einer Position. Jede Position, welche im weiteren Vertriebsprozess eine Auslieferung nach sich ziehen soll, muss mindestens eine Einteilung besitzen.

3.2.3 Beschaffung

In diesem Prozessschritt des Vertriebs wird die Verfügbarkeit der bestellten Ware geprüft und der Bedarf an die Disposition übergeben. Die Disposition organisiert und überwacht den eigentlichen Beschaffungsprozess. Die Art und Weise, wie ein Material für einen Kundenauftrag beschafft wird, kann unterschiedlich sein. Beispielsweise kann die Beschaffung aus dem verfügbaren Bestand erfolgen, durch Nachschub sichergestellt werden, ...

3.2.4 Versand

Mit dem Anlegen des Lieferbelegs beginnt die Versandabwicklung im Vertrieb. Der Lieferbeleg steuert, unterstützt und überwacht sämtliche Teilprozesse der Versandabwicklung, wie z.B. das Kommissionieren, Verpacken, ... Beim Anlegen des Lieferbelegs werden Daten, wie die Materialien und die bestätigten Mengen, aus dem Kundenauftrag übernommen. Beim Anlegen des Transportauftrags werden Daten aus der Auslieferung in den Transportauftrag übernommen und von der Lagerverwaltung weiter verarbeitet.

Auslieferung anlegen Eine Auslieferung wird innerhalb einer Versandstelle für Aufträge angelegt, die zur Auslieferung fällig sind. Die relevanten Daten werden aus dem Auftrag in die Auslieferung kopiert. Man kann aus einem Auftrag eine oder mehrere Auslieferungen anlegen. Umgekehrt kann man Positionen aus mehreren Aufträgen in eine Auslieferung zusammenführen.

Struktur des Lieferbelegs Ein Lieferbeleg hat zwei Ebenen: Kopf und Positionen.

- Kopf des Lieferbelegs
Die Daten des Belegkopfes gelten für den gesamten Beleg, dazu gehören z.B. Daten zum Warenempfänger.
- Position des Lieferbelegs
Jede Position im Lieferbeleg enthält eigene Daten, dazu gehören z.B. Angaben zum Material, zu lieferfähigen Mengen, ...

Kommissionieren Man legt einen Transportauftrag für eine Auslieferung an. Ein Transportauftrag wird für eine Lagernummer erstellt. Das R/3-System kann mehrere Auslieferungen in eine Gruppe von Transportaufträgen zusammenfassen, sofern diese Auslieferungen die gleiche Lagernummer haben.

Warenausgang buchen Beim Buchen des Warenausganges werden automatisch:

- die Mengen in der Bestandsführung und die Lieferbedarfe der Disposition aktualisiert
- die Wertänderung in den Bestandskonten der Materialbuchhaltung verbucht
- weitere Belege für das Rechnungswesen vom System erzeugt, z.B. für das Controlling
- der Status in allen zugehörigen Vertriebsbelegen aktualisiert

3.2.5 Fakturierung

Beim Anlegen der Faktura werden Daten aus dem Kundenauftrag und der Auslieferung in die Faktura übernommen. Die Faktura hat mehrere wichtige Funktionen. Einerseits ist sie der Vertriebsbeleg, der den Benutzer bei der Erstellung von Rechnungen unterstützt; andererseits dient sie als Datenquelle für das Finanzwesen, um den User bei der Zahlungsüberwachung und -abwicklung zu unterstützen. Wenn man eine Faktura anlegt, werden üblicherweise die Sachkonten automatisch eingetragen. Dabei führt das R/3-System eine Sollbuchung auf dem Forderungskonto des Kunden und eine Habenbuchung auf dem Erlöskonto.

Faktura anlegen Man kann eine Rechnung für eine einzelne Auslieferung oder einen einzelnen Kundenauftrag anlegen. Man kann Rechnungen mit Hilfe von Selektionskriterien wie Kunde, Fakturadatum und Empfangsland zusammenfassen.

Ein Fakturabeleg ist in die beiden Ebenen Kopf und Position strukturiert.

Beim Sichern eines Fakturabelegs erzeugt das System automatisch alle erforderlichen Belege für das Rechnungswesen. Dabei führt das R/3-System in der Buchhaltung eine Sollbuchung auf dem Forderungskonto und eine Habenbuchung auf dem Erlöskonto aus. Weiterhin werden beim Buchen des Fakturabelegs:

- der Status in allen zugehörigen Verkaufs-, Liefer- und Fakturabelgen aktualisiert.
- die Vertriebsstatistik im Vertriebsinformationssystem fortgeschrieben
- das Kreditkonto des Kunden fortgeschrieben

3.2.6 Belegfluss

Die Belege innerhalb eines Vertriebsprozesses sind über den Belegfluss miteinander verknüpft. Man kann sich den Belegfluss als Liste der miteinander verknüpften Belege anzeigen lassen. Aus dieser Liste können die betroffenen Belege angezeigt werden oder Statusübersichten zu den Belegen aufgerufen werden. Der Belegfluss wird auf der Ebene des Belegkopfes und der Belegpositionen fortgeschrieben.

3.2.7 Zahlung

Die Zahlung ist ein Prozess, welcher zum Anwendungsmodul Finanzwesen gehört. Die Zahlung unterstützt das Buchen von Zahlungen aufgrund von Rechnungen und die Klärung von Abweichungen.

In der Abbildung (7)¹¹ werden die Teilprozesse der Kundenauftragsabwicklung des R/3-Systems miteinander in Beziehung gebracht. Die Reihenfolge der Blöcke von oben nach unten stellt den üblichen zeitlichen Ablauf im Vertriebsprozess dar. Die Abschnitte auf der rechten und linken Seite der Grafik stellen die wichtigen Schnittstellen zwischen dem Vertrieb und dem Vertriebsinformationssystem (Data Warehouse von SAP), der Materialwirtschaft und der Produktionsplanung dar.

4 ABAP

Die Sprache ABAP (Advanced Business Application Programming) ist die eigene Programmiersprache der SAP. Ein Großteil der betriebswirtschaftlichen Anwendungen eines SAP-Systems ist in ABAP geschrieben. Ein ABAP-Programm besteht aus einzelnen Anweisungen. Jede Anweisung beginnt mit einem Schlüsselwort und endet mit einem Punkt.¹²

¹¹entnommen aus "Prozesse im Vertrieb"

¹²Dieses Kapitel basiert auf den ABAP-Schulungsunterlagen.

Aus ABAP Programmen heraus ist es über spezielle Befehle oder Schlüsselwörter möglich, Selektionsbilder zu erzeugen, Listen zu drucken oder auf Tabelleninhalte zuzugreifen.

Der Zugriff auf die Datenbank erfolgt aus ABAP heraus im Allgemeinen über Open-SQL-Befehle. Open SQL besteht aus einer Menge von ABAP-Anweisungen, die Operationen auf der zentralen Datenbank des SAP-Systems durchführen. Das Ergebnis dieser Operationen und etwaige Fehlermeldungen sind vom verwendeten Datenbanksystem unabhängig. Das macht die Programmentwicklung unabhängig vom jeweils verwendeten Datenbanktyp.

Ausgewählte charakteristische Eigenschaften der Sprache ABAP sind:

- Mehrsprachigkeit (durch getrenntes Speichern von Textelementen wie Listenüberschriften, Texten zu Eingabefeldern, ...)
- Möglichkeit der einfachen und effektiven Entwicklung grafischer Benutzungsoberflächen (mit Hilfe des Screen Painter)
- Möglichkeit der objektorientierten Programmierung ("ABAP Objects")
- Plattformunabhängigkeit (durch Verwendung von Open SQL und Nutzung der Datenbankschnittstelle)
- effizienter Zugriff auf Datenstrukturen (Tabellen, Datenelemente)

4.1 Der Ablauf eines ABAP Programms

Wenn man den Ablauf des ABAP Programms von der Client/Server Sicht betrachtet¹³, so findet auf der Präsentationsserver-Schicht die Darstellung des

¹³siehe Kapitel 1.3.1

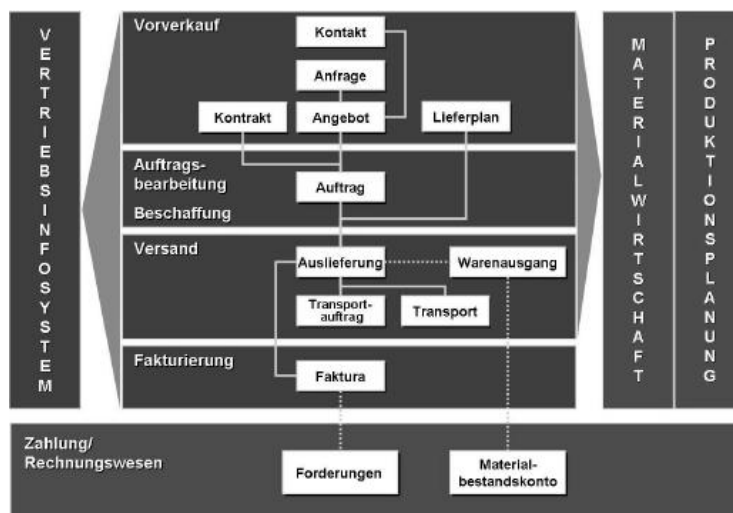


Abbildung 7: Der Belegfluss

Bildschirmbildes statt. Das ABAP-Laufzeitsystem und somit auch das ABAP-Programm sind auf dem Applikationsserver. Die unterste Schicht ist die Datenbank, die abhängig vom jeweiligen Programm für die Ausführung notwendig sein kann.

Den Anwender interessiert sein betriebswirtschaftlicher Ablauf und wie er die zugehörigen Daten eingeben kann oder zur Ausgabe bekommt. Weniger interessant ist, ob dazu ein Programm abläuft oder implizit mehrere Programme gerufen werden oder ob es technische Unterschiede zwischen den Bildschirmbildern gibt. Für den Anwender stellt also das R/3-System mit Applikations-Servern und Datenbank eine Black Box dar.

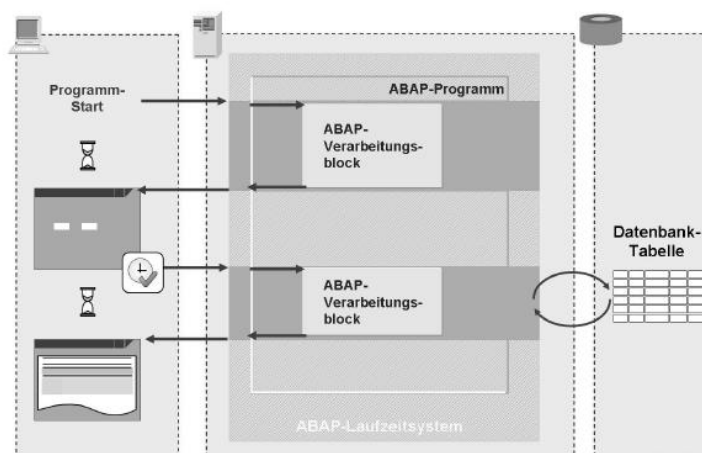


Abbildung 8: Das Zusammenspiel zwischen Serverebenen

Führt der Anwender eine Benutzeraktion aus (z.B. die Entertaste, eine Funktionstaste, eine Menüfunktion oder eine Drucktaste), so wechselt die Kontrolle vom Präsentations-Server an den Applikationsserver, und gewisse Teile des ABAP-Programms werden verarbeitet. Wird aus dem ABAP-Programm ein weiterer Benutzerdialog angestoßen, so wird ein Bildschirmbild zum Präsentationsserver gesendet und die Kontrolle wechselt zum Präsentationsserver. Dieses Wechselspiel zwischen unterschiedlichen Serverebenen kann man der Abbildung (8)¹⁴ entnehmen.

4.2 Einführung in die ABAP Workbench

4.2.1 Die ABAP Workbench und das Repository

Die ABAP Workbench ist die integrierte grafische Entwicklungsumgebung des SAP-Systems. Sie unterstützt unter anderem die Entwicklung, das Testen und die Verwaltung von in ABAP geschriebenen Anwendungen.

Mit der ABAP Workbench werden die Anwendungsprogramme geschrieben. Die Workbench ist eine grafische Programmierumgebung. In der Workbench können Programmierwerkzeuge aufgerufen werden. Eine ABAP Anwendung ist z.B. eine Transaktion oder ein Report.

¹⁴entnommen aus "ABAP Schulung"

Im SAP Easy Access findet man die Werkzeuge unter *Werkzeuge -> ABAP Workbench -> Entwicklung*. Von dort aus hat man unter anderem Zugriff auf

- den ABAP Editor (Transaktion SE38) zum Schreiben von ABAP Programmen
- das ABAP Dictionary (Transaktion SE11) zur Definition und Beschreibung von Tabellen, Datenelementen, Sperrelementen usw.
- den Screen Painter (Transaktion SE51, im Unterverzeichnis *Oberfläche*) zur Erstellung interaktiver Bedienoberfläche.
- den Function Builder (Transaktion SE37) zur Erstellung und Verwaltung von Funktionsbausteinen (dies sind Stücke gekapselten ABAP-Codings mit definierten Ein- und Ausgabeschnittstelle).

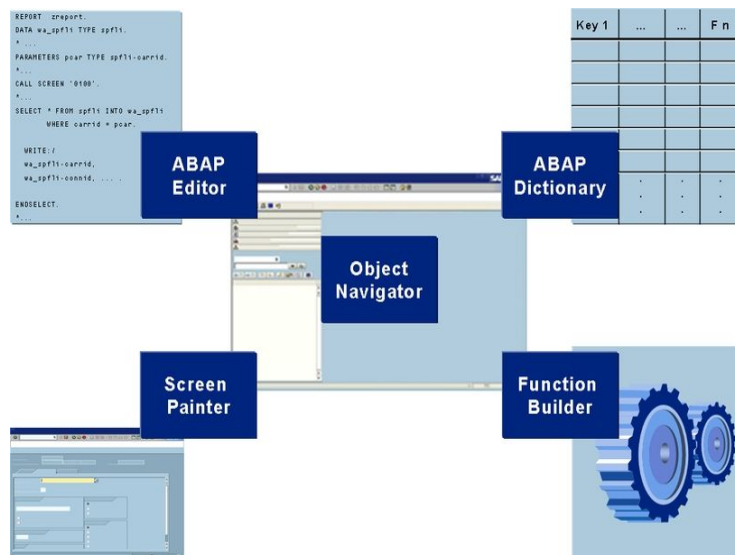


Abbildung 9: Ausgewählte Werkzeuge der ABAP Workbench

Die Werkzeuge der ABAP-Workbench sind in der Abbildung (9)¹⁵ abgebildet.

Bei der Arbeit in der Workbench stoßt man auf Entwicklungsobjekte und Pakete:

- Entwicklungsobjekte sind Objekte, die mit Hilfe der ABAP Workbench bearbeitet werden können, z.B. Reports oder Transaktionen.
- Ein Paket fasst logisch zusammenhängende Entwicklungsobjekte zusammen, beispielsweise alle Objekte einer Anwendung

¹⁵entnommen aus "Application Platform Grundlagen"

Um die Entwicklungsprozesse in der integrierten Umgebung der ABAP Workbench zu organisieren, bietet SAP den Object Navigator (Transaktion SE80, Menüpfad *Werkzeuge -> ABAP Workbench -> Übersicht -> Object Navigator*) an. Hiermit erhält man einen übersichtlichen und einheitlichen Zugang zu Repository-Objekten. Anstatt mit Werkzeugen und Paketen zu arbeiten, arbeitet man im Object Navigator mit Objekten, und die Workbench ruft für jedes Objekt das entsprechende Werkzeug auf.

Die Datenbank enthält sowohl Anwendungs- und Customizing-Tabellen, die in der Regel mandantenabhängig sind, als auch das Repository. Das Repository enthält alle Entwicklungsobjekte (Programme, Definitionen von Datenbanktabelle und zentrale Typen). Die Repository-Objekte sind mandantenunabhängig, sie sind also in allen Mandanten sichtbar und verwendbar. Das Repository ist nach Anwendungen untergliedert. Innerhalb einer Anwendung existieren zur logischen Unterteilung mehrere Entwicklungsklassen.

Die Struktur des Repository findet man unter der Anwendungshierarchie (*Werkzeuge -> ABAP Workbench -> Übersicht -> Anwendungshierarchie* oder Transaktion SE81). Über das Repository Infosystem (*Werkzeuge -> ABAP Workbench -> Übersicht -> Infosystem* oder Transaktion SE84) kann man gezielt nach Repository-Objekten suchen. Es stehen für verschiedene Repository-Objekte geeignete Suchkriterien zur Verfügung.

4.2.2 Analysieren eines vorhandenen Programms

Funktionsumfang ermitteln: Programm ausführen Es gibt verschiedene Möglichkeiten, ein Programm zu starten. Die erste Möglichkeit ist aus ‘SAP Easy Access’; wenn man ein Programm oft verwendet kann man es in seine Favoritenliste aufnehmen. Wählt man einen Eintrag im Baum aus, so kann man das zugehörige Programm über das Kontextmenü starten. Über *Zusätze -> Einstellungen* und Ankreuzen von ‘Technische Namen anzeigen’ kann man die technischen Transaktionscodes im Baum anzeigen lassen. Der Transaktionscode ist auch die zweite Art ein Programm auszuführen. Ein Programm kann man auch über den Menüeintrag im Object Navigator starten.

Über Ausführen des Programms kann man den Funktionsumfang ermitteln. Zu einem beliebigen Bildschirmbild kann man Informationen über den Programmnamen und die Bildnummer des Bildschirmbildes erhalten über den Menüpfad *System -> Status*. Zu einem beliebigen Feld des Bildschirmbildes kann man Informationen über den Feldnamen und Feldtyp erhalten über *F1 -> Technische Info*.

Statische Analyse: Objektliste Einen Überblick über die Programmobjekte erhält man über die Objektliste des Programms im Navigations-Bereich des Object Navigators. Um die Objektliste zu einem Programm anzuzeigen, muss man im oberen Eingabefeld ‘Programm’ wählen und im unteren Eingabefeld den Programmnamen eingeben. Über die Ikone mit der Brille oder die Enter-Taste erhält man die Objektliste. In dem Hierarchybaum erscheinen nur diejenigen Objektarten, zu denen Objekte vorhanden sind. Zu jedem Objekt der Objektliste kann man eine Umfeldermittlung durchführen. Dafür muss man ‘Verwendungsnachweis’ im Kontextmenü des gewünschten Objektes wählen. Man bekommt eine Liste mit allen Stellen an denen das Objekt verwendet wird.

Über Doppelklick kann man zu einem Eintrag der Liste zur Anzeige der Verwendungsstelle navigieren.

Dynamische Analyse: Debugging Modus Es gibt mehrere Möglichkeiten, ein Programm im Debugging Modus zu starten, ohne das Programm zu verändern:

- Über das Kontextmenü zu einem Programmnamen in der Objektliste des Programms: *Ausführen ->Debugging*
- Über einen Haltepunkt im Editor: Man kann gezielt eine Programmstelle auswählen, an der das Laufzeitsystem in den Debuggingmodus umschalten soll. Dafür muss man die gewünschte Zeile markieren und die Ikone mit dem STOP-Zeichen wählen. Anschließend muss das Programm über *Ausführen ->Direkt* oder F8 gestartet werden.

Startet man ein Programm im Debugging Modus, so kann man über die Ikone "Einzelschritt" zeilenweise das Programm ausführen.

Zur Laufzeit kann auch in den Debugging Modus umgeschaltet werden (System ->Hilfsmittel ->Debugging ABAP). Die zweite Möglichkeit in den Debugging Modus zu kommen, ist die Eingabe von /h in das Befehlsfeld der Symbolleiste und Enter.

Die allgemeine ABAP-Syntax Für die allgemeine ABAP-Syntax gelten folgende Regeln:

- Ein ABAP Programm besteht aus einzelnen Sätzen (Anweisungen).
- Jeder Satz muss mit einem Punkt abgeschlossen werden.
- Das erste Wort innerhalb eines Satzes ist ein Schlüsselwort.
- Sätze dürfen eingerückt werden.
- Sätze dürfen sich über mehrere Zeilen erstrecken.
- Mehrere Sätze in einer Zeile sind möglich.
- Aufeinanderfolgende Sätze mit identischem Anfang können zu einem Kettenatz zusammengefasst werden. Dabei muss der identische Satzanfang mit einem Doppelpunkt abgeschlossen werden. Die geketteten Satzteile müssen durch ein Komma getrennt werden. Vor und hinter dem Trennzeichen (Doppelpunkt, Komma, Punkt) können Leerzeichen stehen.

Das Beispielprogramm

```
REPORT erster_report.
```

```
WRITE 'Mein erster ABAP Report.'
```

enthält zwei Anweisungen, jede in einer Zeile. Die Schlüsselwörter sind REPORT und WRITE. In diesem Fall enthält die Liste die Zeile "Mein erster ABAP Report."

Zu einer ABAP-Anweisung kann man sich die Schlüsselwort-Dokumentation anzeigen lassen. Wenn der Cursor auf einem Schlüsselwort steht, bekommt man mit F1 die Dokumentation. Die andere Möglichkeit ist über die Informations Ikone die Schlüsselwort-Dokumentation für den entsprechenden ABAP-Begriff zu bekommen. Mit dem Doppelklick auf einen Tabellennamen oder Feldnamen im Editor bekommt man detaillierte Informationen zu den Einzelobjekten.

Mit einem Stern (*) in der ersten Spalte kann eine ganze Zeile als Kommentar gekennzeichnet werden. Will man nur den Rest einer Zeile als Kommentar kennzeichnen, erreicht man das mit Anführungszeichen (").

Wenn man ein Programm (oder allgemeiner: ein Entwicklungsobjekt) anlegt oder ändert und dann sichert, wird zunächst immer nur eine inaktive Version im Repository abgelegt. So ist es möglich, weiter zu entwickeln, ohne den aktuellen Systemzustand zu ändern.

Damit ein Repository-Objekt systemweit verfügbar ist, muss es zusätzlich noch "aktiviert" werden. Diese somit erzeugte aktive Fassung des Programmes wird herangezogen, wenn zum Beispiel ein Anwender das Programm ausführen möchte.

4.3 ABAP-Anweisungen und Datendeklarationen

4.3.1 Die ABAP-Typen

Die Typen beschreiben Eigenschaften von Ein- und Ausgabefeldern von Bildschirmobjekten, Datenobjekten und Schnittstellen-Parametern.

Die Typen können programmlokal oder zentral definiert werden. Programmlokale Typen werden verwendet wenn nur technische oder keine semantischen Eigenschaften benötigt werden und wenn die Typen nur programmlokal verwendet werden. Die zentralen Typen (=ABAP-Dictionary-Typen) werden benötigt, wenn die Typen auch außerhalb des Programms gebraucht werden, oder wenn auch semantische Informationen benötigt werden (z.B. auf Bildschirmbildern mit Ein-/ und Ausgabefeldern).

Im ABAP-Dictionary werden sowohl globale Typen definiert als auch die Beschreibung von Datenbanktabellen verwaltet. Für globale Typen gibt es folgende Möglichkeiten:

- Elementare Typen heißen Datenelemente. Sie beinhalten die vollständige Beschreibung der technischen Eigenschaften eines elementaren Feldes (Technischer Typ, Länge, Anzahl der Nachkommastellen) und semantische Information (Feldbezeichner, Felddokumentation, Suchhilfe).
- Strukturtypen
- Strukturtypen heißen im Regelfall Strukturen. Sie können aus elementaren Komponenten bestehen. Eine Komponente kann aber auch einen strukturierten Typen haben. Dadurch sind beliebige geschachtelte Typen konstruierbar.
- Tabellentypen sind Typen für interne Tabellen.

Über das Repository Infosystem kann man nach ABAP-Dictionary-Typen suchen.

Im Regelfall wird ein Datenobjekt mit einem vollständigen programmlokalen oder einem vollständigen globalen Typ typisiert. Der Doppelklick auf den Namen des lokalen Typs führt den Benutzer zur Definitionsstelle des Typen. Globale Typen müssen im Dictionary gesucht werden.

Vollständig eingebaute ABAP-Typen können direkt zur Typisierung von Variablen verwendet werden. In diesem Fall bleibt Doppelklick auf den Typ wirkungslos. Informationen über eingebaute ABAP-Typen erhält man über das Schlüsselwort-Dokumenatation zu TYPES oder DATA.

Die folgenden eingebauten ABAP-Typen sind vollständig:

- **d** - Datum (**D**aten: YYYYMMDD)
- **t** - Zeitpunkt (Time: HHMMSS)
- **i** - ganze Zahl (**I**nteger)
- **f** - Gleitpunktzahl (**F**loating Point)
- **string** - Zeichenfolge (String, Länge variabel)
- **xstring** - Bytefolge (he**X**adecimal-String, Länge variabel)

Bei folgenden eingebauten Typen muss die Länge ergänzt werden:

- **c** - Zeichen (**C**haracter)
- **n** - Numerischer Text (**N**umerical Character)
- **x** - Byte (he**X**adecimal)
- **p** - gepackte Zahl (**P**acked numbers = Binary Coded Decimals), hier ist zusätzlich die Anzahl an Nachkommastellen anzugeben.

4.3.2 Die Datenobjekte

Elementare Datenobjekte Datenobjekte kann man einem Typ zuordnen, indem man sich auf einen eingebauten ABAP-Typ, einen benutzerdefinierten Typ oder einen ABAP-Dictionary-Typ bezieht.

Bsp.: DATA: <varname> TYPE < type>.

Die zweite Möglichkeit ist, die Deklaration LIKE zu verwenden. Bezieht sich eine Variable **v2** mit LIKE auf die Variable **v1** (DATA v2 LIKE v1), so erbt sie den Typ von **v2**.

Bsp.: DATA: <varname> LIKE < data object>.

Ein elementares Datenobjekt kann folgendermaßen definiert werden:

- DATA: <type name> TYPE <predefined ABAP Type>.

falls man ein Datenobjekt der Typen **d**, **t**, **i**, **f**, **string** oder **xstring** benötigt. Diese eingebauten ABAP Typen sind vollständig.

- TYPES <type name> (<length>) TYPE <predefined ABAP Type>.

DATA <do name> TYPE <type name>.

falls man ein Datenobjekt der Typen **c**, **n**, **p** oder **x** benötigt. Für diese Typen muss die Länge festgelegt werden. Für den Typ **p** kann zusätzlich noch die Anzahl der Nachkommastellen mit DECIMALS <nn> definiert werden.

- DATA: <do name> TYPE <global Type>.

falls ein passender globaler Typ im Dictionary definiert ist.

Für die Fallunterscheidung stellt ABAP die IF und CASE-Abfragen zur Verfügung.

CASE <data object 1>.

WHEN <data object 2>.

<Anweisung 1>.

WHEN <data object 3>.

<Anweisung 2>.

...

WHEN OTHERS.

<Anweisung *x*>.

ENDCASE.

Von den angeführten Anweisungsfolgen wird genau eine verarbeitet. Die Angabe von WHEN OTHERS ist optional.

IF <logical expression 1>.

<Anweisung 1>.

ELSEIF <logical expression 2>.

<Anweisung 2>.

...

ELSE.

<Anweisung 3>.

ENDIF.

Die Anweisungen ELSE und ELSEIF sind optional. Wenn der logische Ausdruck erfüllt ist, wird die anschließende Anweisungsfolge ausgeführt. Ist der logische Ausdruck nicht erfüllt, so wird der ELSE- bzw. ELSEIF-Teil verarbeitet. Ist keine ELSE-bzw. keine weitere ELSEIF-Anweisung vorhanden, so wird die Programmverarbeitung hinter ENDIF fortgesetzt.

Strukturen In ABAP ist es möglich, strukturierte Datenobjekte (kurz Strukturen genannt) zu definieren. So können zusammengehörige Variablen in einem Objekt zusammengefasst werden. Strukturen sind schachtelbar, so dass Komponenten auch Strukturen oder Tabellen sein können.

Es gibt zwei verschiedene Arten von Strukturen in einem ABAP-Programm:

- Über

```
DATA <name> TYPE <structure_type>.
```

definierte Strukturen dienen als Zielfelder für Datenbankzugriffe oder für programmlokale Berechnungen. Die Strukturtypen können im ABAP Dictionary oder programmlokal deklariert sein.

- Über

```
TABLES <ABAP-Dictionary-Structures>.
```

wird eine Struktur definiert, die technisch in einem eigenen Bereich verwaltet wird. Seit Release 4.0 benötigt man TABLES-Strukturen nur noch als Schnittstellen zum Dynpro.

Einem Strukturtyp werden einzelne Komponenten zugeordnet. Zu jeder Komponente müssen der Komponentename, der Komponententyp und ein Kurztext erfasst werden. Der Komponentename ist frei wählbar und es existieren keine festen Namenskonventionen. In der Regel wird für den Komponententyp ein Datenelement zur Typisierung verwendet. Der Kurztext beschreibt schließlich die Komponente. Beim Anlegen einer Struktur im Dictionary ist zu beachten, dass der Name der Struktur im Kundennamensraum liegt.

Die Strukturtypen können auch programmlokal über die TYPE-Anweisung definiert werden. Die Komponenten werden durch

```
BEGIN OF <structure type name>  
...  
END OF <structure type name>.
```

Eine Struktur kann man dann über

```
DATA <structure name> TYPE <structure type name>.
```

anlegen.

Die Felder der Struktur werden angesprochen durch <Structure>-<Fieldname>.

Interne Tabellen Interne Tabellen sind Datenobjekte, mit denen man Datenmengen einer festen Struktur aber mehrerer Zeilen im Arbeitsspeicher halten kann. Die einzelnen Komponenten einer Zeile bezeichnet man daher auch als Spalten interner Tabelle. Interne Tabellen sind auch dynamische Datenobjekte, die beliebig viele Zeilen eines festen Typs enthalten können. Die Zeilentypen von internen Tabellen sind beliebige ABAP-Datentypen. Sie können somit elementar, strukturiert oder selbst interne Tabellen sein.

Um einen Tabellentyp vollständig zu spezifizieren, müssen folgende Informationen angegeben sein:

- Zeilentyp: Die Informationen zu den gewünschten Spalten, deren Namen und Typen wird gewährleistet, indem ein Strukturtyp als Zeilentyp angegeben wird.
- Schlüssel: Zu einem vollständigen spezifizierten Schlüssel gehören auch Informationen, wie z.B.: Welche Spalten sollen Schlüsselspalten sein?...
- Tabellenart: Es gibt drei Tabellenarten: Standardtabellen, sortierte Tabellen und Hash Tabellen. Den Haupteinfluss auf die Wahl der Tabellenart hat die geschätzte Zugriffsart.

Die Zugriffsart definiert, wie das Laufzeitsystem auf einzelne Tabellenzeilen zugreift. ABAP kennt zwei unterschiedliche Zugriffsarten, den Indexzugriff und den Schlüsselzugriff. Beim Indexzugriff wird über den Index der Datensätze, der vom System gepflegt wird, zugegriffen. Beim Schlüsselzugriff wird über einen Suchbegriff zugegriffen, der typischerweise aus dem Tabellenschlüssel oder dem generischen Tabellenschlüssel besteht.

Für interne Tabellen stehen folgende Operationen zur Verfügung: APPEND, INSERT, READ, MODIFY, DELETE, COLLECT, LOOP ... ENDLOOP, SORT, CLEAR, REFRESH, FREE. Informationen zu den angeführten ABAP-Anweisungen kann man der Schlüsselwortdokumentation zu den angegebenen ABAP-Schlüsselwörtern entnehmen.

4.4 Lesen von der Datenbank

4.4.1 Datenbanktabellen im R/3

Die Datenbanktabellen werden im R/3 Dictionary über das ABAP-Dictionary verwaltet. Dort können die aktuellen Informationen zu allen technischen Eigenschaften einer Tabelle nachgeschaut werden.

Eine *Tabelle* ist eine zweidimensionale Matrix aus Spalten (Feldern) und Zeilen (Einträgen). Sie hat einen Namen und Eigenschaften wie z.B. den Tabellentyp. Jede Tabelle des ABAP Dictionary hat einen Primärschlüssel. Dies ist eine Spaltenkombination, die jede Zeile in dieser Tabelle eindeutig identifiziert.

Ein Feld (d.h. eine Spalte einer Tabelle) hat einen Namen und Eigenschaften, z.B. kann es Primärschlüsselfeld sein. Das Feld ist kein eigenständiges Objekt, sondern tabellenabhängig und kann nur innerhalb einer Tabelle gepflegt werden. Ein Tabellenfeld kann mit Hilfe einer Tabelle gepflegt werden:

- Die Domäne dient zur technischen Definition des Tabellenfeldes. In ihr werden z.B. Feldlänge, Feldtyp, Ausgabeigenschaften und eine mögliche Werteeinschränkung mit Hilfe von Festwerten definiert.
- Das Datenelement dient ursprünglich der Beschreibung der semantischen Eigenschaften eines Feldes im Kontext der Tabelle. Dies sind Eigenschaften, die nur dort und nicht allgemein von Bedeutung sind. Im Datenelement muss z.B. eine Kurzbeschreibung des Tabellenfeldes hinterlegt werden, die beim Aufruf der F1-Hilfe auf dem Bildschirmbild angezeigt wird.

Das zweistufige Domänenkonzept (bestehend aus den Ebenen Datenelement und Domäne) erlaubt die Definition und Pflege technischer Feldeigenschaften auf Domänenebene. Eine Domäne kann ihre Feldeigenschaften an beliebig viele

Felder weitergeben, wodurch bei Änderungen der so beschriebenen Feldeigenschaften lediglich die Domäne, nicht aber die einzelnen Felder, explizit geändert werden müssen.

Tabellen, Datenelemente und Domänen werden zentral im ABAP Dictionary verwaltet.

4.4.2 Lesen von einer Datenbanktabelle

Die Anweisungen von Open SQL sind eine vollständig in ABAP integrierte Untermenge von Standard SQL. Sie erlauben dem ABAP-Programmierer einen einheitlichen Zugriff auf Daten, unabhängig vom installierten Datenbanksystem. Die Datenbankschnittstelle setzt Open SQL in datenbankabhängiges Standard SQL um.

Die SELECT Anweisung hat die folgende Form:

```
SELECT <ergebnis> FROM <tabelle>
      INTO <ziel>
      WHERE <bedingungen>
      ENDSELECT.
```

Mit der Open SQL-Anweisung SELECT programmiert man lesende Datenbankzugriffe. Hinter der SELECT-Klausel verbirgt sich eine aufwendige Logik, die es erlaubt, auf viele verschiedene Arten auf Datenbanktabellen zuzugreifen. Die SELECT-Klausel beschreibt, ob das Ergebnis der Selektion mehrere Zeile oder einen einzelnen Datensatz (SELECT SINGLE) umfassen soll, welche Spalten das Ergebnis haben soll und ob im Ergebnis identische Zeilen vorkommen dürfen. Die INTO-Klausel bestimmt die internen Datenobjekte, in die die selektierten Daten gestellt werden sollen. Die FROM-Klausel benennt die Quelle (Datenbanktabelle oder View), von der die Daten selektiert werden sollen. Die WHERE-Klausel gibt Bedingungen an, die das Ergebnis der Selektion erfüllen muss. Sie bestimmt also die Zeilen der Ergebnistabelle.

Statt der INTO-Klausel kann auch die Klausel INTO CORRESPONDING FIELDS <table > verwendet werden. In diesem Fall werden die Daten in die namensgleichen Felder einer Struktur gestellt.

4.5 Die Liste

Die Liste ist darauf zugeschnitten, Daten in übersichtlicher Form, meist tabellarisch, auszugeben. Über die Standardoberfläche einer Liste stehen dem Benutzer eine Reihe von operationen bezüglich Liste zur Verfügung: Zurück, Beenden, Abbrechen, Drucken, Suchen in der Liste, Sichern und Liste als E-Mail senden.

Für die Liste kann man eine Listenüberschrift und bis zu vier Zeilen Spaltenüberschriften definieren. Die Überschriften können entweder über den Editor definiert werden. Die andere Möglichkeit ist, wenn man das Programm **sichert** und **aktiviert** und anschließend über **Starten** des Programms die Liste erzeugt, kann man über den Menüpfad System -> Liste -> Listen- und Spaltenüberschriften bestimmen. Ist keine Listenüberschrift definiert, so wird stattdessen der Programmtitel verwendet.

Der folgende Report ist ein kleines Beispiel einer Liste:

```

REPORT liste_von_winf_studenten.
DATA wa_students TYPE students.
SELECT id name surname birthday
FROM students
INTO CORRESPONDING FIELDS OF wa_students
WHERE faculty = 'Wirtschaftsinformatik'
WRITE: / wa_student-id,
wa_student-name,
wa_student-surname,
wa_student-birthday.
ENDSELECT.

```

Aus der students Tabelle werden die Studenten ausgewählt die Wirtschaftsinformatik studieren und die Matrikelnummer, der Name, der Nachname und das Geburtsdatum werden ausgegeben.

4.6 Selektionsbilder

Die Selektionsbilder dienen der Eingabe von Auswahlkriterien für den weiteren Programmverlauf. Erstellt man zum Beispiel eine Liste aus Daten einer sehr großen Datenbanktabelle, so kann man über ein Selektionsbild die Werte einschränken.

4.6.1 Eigenschaften von Selektionsbildern

Die Selektionsbilder sind darauf zugeschnitten, dem Anwender eine Eingabemaske für Wertemengen zur Verfügung zu stellen, die zur Einschränkung der Datenmenge genutzt wird, die von der Datenbank gelesen werden muss. Der Anwender hat folgende Möglichkeiten:

- Eingabe auf Einzelfeldern
- komplexe Eingabe: Intervalle, Operationen, Muster
- Sichern von ausgefüllten Selektionsbildern als Varianten
- Werthilfen bzw. Suchhilfen über die Funktionstaste F4 oder Bestätigung der Kombo-Box

Ist ein Eingabefeld mit einem Datenelement typisiert, so stehen semantische Informationen wie Feldbezeichner, Eingabehilfe und Werthilfe zur Verfügung. Wenn ein Eingabefeld mit einem Strukturfeld typisiert ist, das ein Datenelement referenziert, kann man auch die Feldbezeichner, Eingabehilfe und Werthilfe benutzen.

Standardmäßig erscheinen die Feldnamen als Text vor den Eingabefeldern eines Selektionsbilds. Sie können aber auch durch Selektionstexte ersetzt werden.

In einem ausführbaren Programm genügt es, eine PARAMETERS-Anweisung zu programmieren, um das Standardselektionsbild zu erhalten. Durch die Anweisung

PARAMETERS <name > TYPE <type>.

bzw.

PARAMETERS <name > LIKE <data objet>.

wird ein elementares Eingabefeld auf dem Selektionsbild erzeugt und ein typgleiches Datenobjekt <name> deklariert. Gibt der Anwender einen Wert ein und wählt "Ausführen", dann wird der Eingabewert in das programminterne Datenobjekt <name> gestellt. Dabei sind nur typkonforme Eingaben zugelassen.

4.7 Softwarelogistik

Der Entwicklungszyklus von SAP-Entwicklungsprojekten kann mit einer Dreisystemlandschaft abgebildet werden(siehe Abbildung (10)¹⁶). Das System besteht aus den Komponenten Entwicklungssystem, Konsolidierungssystem und Produktivsystem.

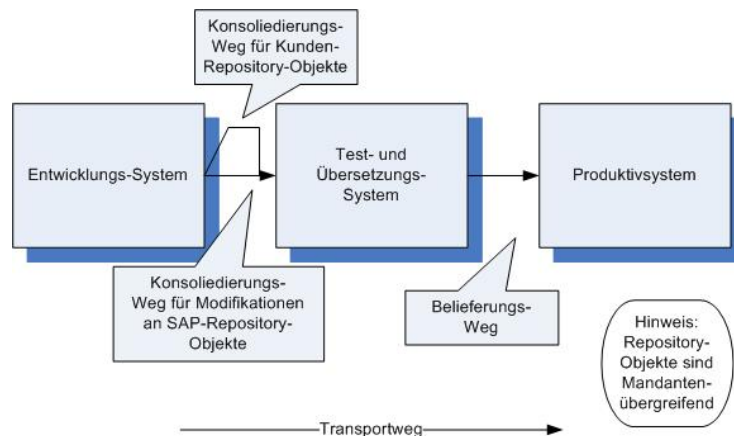


Abbildung 10: Die Systemlandschaft

Der Entwicklungszyklus läuft folgendermaßen ab:

- Alle neuen und geänderten Programme werden einem Änderungsauftrag zugeordnet. Unmittelbar vor Einleitung der Testphase im Konsolidierungssystem wird dieser Auftrag freigegeben und in das Konsolidierungssystem transportiert.
- Im Konsolidierungssystem werden Tests durchgeführt. Identifizierte Fehler werden an die Entwicklung weitergeleitet. Diese führt notwendige Korrekturen durch und ordnet die korrigierten Programme einem neuen Änderungsauftrag zu. Dieser Auftrag wird freigegeben und in das Konsolidierungssystem transportiert. Falls nötig, muss dieser Teilzyklus mehrfach wiederholt werden.

¹⁶entnommen aus "ABAP Grundlagen"

- Sobald der Test erfolgreich abgeschlossen ist, werden alle Aufträge in das Produktivsystem transportiert.

Die aktuelle System-Konfiguration kann im Transport Management System nachgeschaut werden (Menüpfad: *Werkzeuge -> Administration -> Transport -> Transport Management System* bzw. Transaktion STMS).

Index

- Änderung des Repository, 14
- Arbeiten am Customizing, 14
- Erweiterung des Repository, 14

- ABAP, 22
- ABAP Dictionary, 25
- ABAP Editor, 25
- ABAP Programm, Ablauf, 23
- ABAP Workbench, 24
- ABAP, Eigenschaften, 23
- ABAP-Syntax, 27
- ABAP-Typen, 28
- ACID, 11
- Anmeldung, 5
- Anmeldung am SAP-System, 12, 13
- Applikationsprozesse, 8
- asynchrone Verbuchung, 12
- atomar, 11
- atomic, 11
- Auftrag anlegen, 20
- Auftragsbearbeitung, 19
- Auslieferung anlegen, 21

- Belegfluss, 22
- Benutzerstammsatz, erstellen, 13
- Benutzerdaten, 13
- Benutzermenü, 6
- Benutzername, 12
- Benutzerstammsatz, 13
- Benutzerstammsatz, pflegen, 13
- Berechtigungen, 12, 13
- Berechtigungsprüfung, 13
- Berechtigungsprofil, 13
- Beschaffung, 20
- Buchungskreis, 18

- Client, 7
- Client-Server Architektur, 7
- consistent, 11
- Customizing, 14

- Datenbankprozesse, 8
- Datenbankschnittstelle, 10
- Datenbanktabellen, 32
- Datenelement, 32
- Datenobjekte, 29
- Datenstruktur, 14
- dauerhaft, 11

- Debugging Modus, 27
- Dialog-Workprozesse, 11
- Dispatcher, 9
- Domäne, 32
- Domänenkonzept, 32
- durable, 11

- Eigenentwicklung, 14
- Eigenschaften von Selektionsbildern, 34
- Eingabehistorie, 7
- elementare Datenobjekte, 29
- elementare Typen, 28
- Enqueue-Workprozess, 11

- Faktura anlegen, 22
- Fakturierung, 21
- Favoriten, 6
- Favoritenliste, 6
- Feld, 32
- Function Builder, 25

- Hilfe, 7
- Hintergrund-Workprozesse, 11

- interne Tabellen, 31
- isolated, 11
- isoliert, 11

- Kommissionieren, 21
- Konfiguration des SAP-Systems, 8
- konstistent, 11
- Kundenerweiterungen, 15

- Lagerort, 18
- Lesen von einer Datenbanktabelle, 33
- Liste, 33

- Mandant, 6, 14
- Modifikationen, 15
- mySAP Business Suite, 5

- Navigation, 7

- Object Navigator, 26
- Objektliste, 26
- Open SQL, 10
- Organisationseinheit, 17

- PARAMETERS, 34

Personalisierung der Benutzeroberfläche, 7
Präsentationsprozesse, 8
Präsentationsschnittstelle, 10
programmlokale Typen, 28
Puffer, 9

RDBMS, 10
Relationale Datenbankmanagementsysteme, 10
Repository, 14, 26
Rolle, 13

SAP Branchenlösungen, 5
SAP GUI, 5
SAP NetWeaver, 5
SAP Smart Business Solutions, 5
SAP xApps, 5
SAP-Bibliothek, 7
SAP-Transaktion, 11
Screen Painter, 25
SELECT, 33
Selektionsbilder, 34
Server, 7
Sparte, 18
Spoolworkprozesse, 11
Struktur des Lieferbelegs, 21
Struktur des Verkaufsbelegs, 20
Strukturen, 31

Tabelle, 32
Tabellentypen, 28
Terminauftrag, 19

Verbuchungs-Workprozess, 12
Verbuchungs-Workprozesse, 11
Verbuchungsverlauf, 12
Verkaufsorganisation, 18
Versand, 20
Versandstelle, 19
Vertriebsbereich, 18
Vertriebsbereich, 18
Vertriebsweg, 18
Vorverkaufsaktivitäten, 19

Warenausgang buchen, 21
Werk, 18
Workprozess, 9

Zahlung, 22
zentrale Typen, 28