

Anwendungsentwicklung mit der ABAP/4 Development Workbench

Die ABAP/4 Development Workbench ist eine vollständige, durch die R/3-Anwendungsentwicklung praxiserprobte Umgebung für die Realisierung von Client/Server-Lösungen. Kundenspezifische Neuentwicklungen sind mit den verfügbaren Werkzeugen genauso möglich wie die Erweiterung der von SAP ausgelieferten R/3-Anwendungen. Alle mit der ABAP/4 Development Workbench erstellten Anwendungen sind ohne weitere Anpassungen auf allen von SAP unterstützten Rechnertypen, Datenbanksystemen und grafischen Benutzeroberflächen ablauffähig.

**Kundenspezifische
Neuentwicklungen**

Mit der R/3-Entwicklungsumgebung können problemlos Client/Server-Anwendungen realisiert werden, ohne daß sich der Entwickler um Kommunikations- und Verteilungsaspekte kümmern muß. Alle entwickelten Anwendungen sind nicht an einen zentralen Rechner gebunden, sondern können auch dezentral unter Ausnutzung verschiedener Server ablaufen.

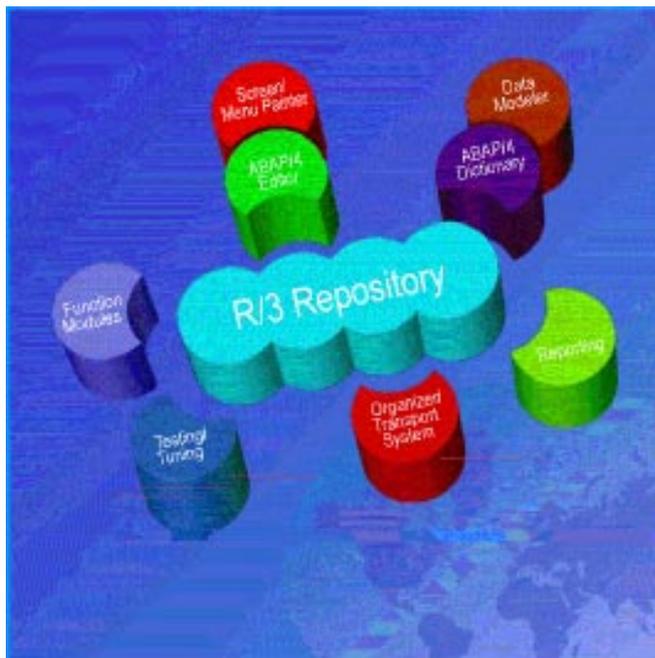


Abb. 2-1: Überblick ABAP/4 Development Workbench

Die ABAP/4 Development Workbench bietet den Zugriff auf die SAP-Entwicklungswerkzeuge, die den gesamten Software-Entwicklungszyklus abdecken. Anwendungsentwickler finden leistungsfähige Tools für die Programmierung betriebswirtschaftlicher Anwendungen in der 4GL ABAP/4, für den Datenbankzugriff, für die Netzkommunikation und für die Realisierung von grafischen Benutzeroberflächen. Metadaten der Anwendungsentwicklung werden mit Hilfe des ABAP/4 Dictionary beschrieben, verwaltet und aktiv in den Programmablauf eingebunden.

Sämtliche Entwicklungsobjekte der ABAP/4 Development Workbench sind im ABAP/4 Repository abgelegt. Zu diesen Entwicklungsobjekten gehören z.B. Dictionary-Daten, ABAP/4-Programme, Dynpros, Dokumentationen und Hilfetexte.

ABAP/4 Repository Die Navigation und Orientierung innerhalb der ABAP/4 Development Workbench wird komfortabel durch Objektlisten, Menüsteuerung und Mehrfenstertechnik unterstützt. Leistungsfähige Werkzeuge für Software-Wartung, Test und Tuning runden das Angebot der ABAP/4 Development Workbench ab.

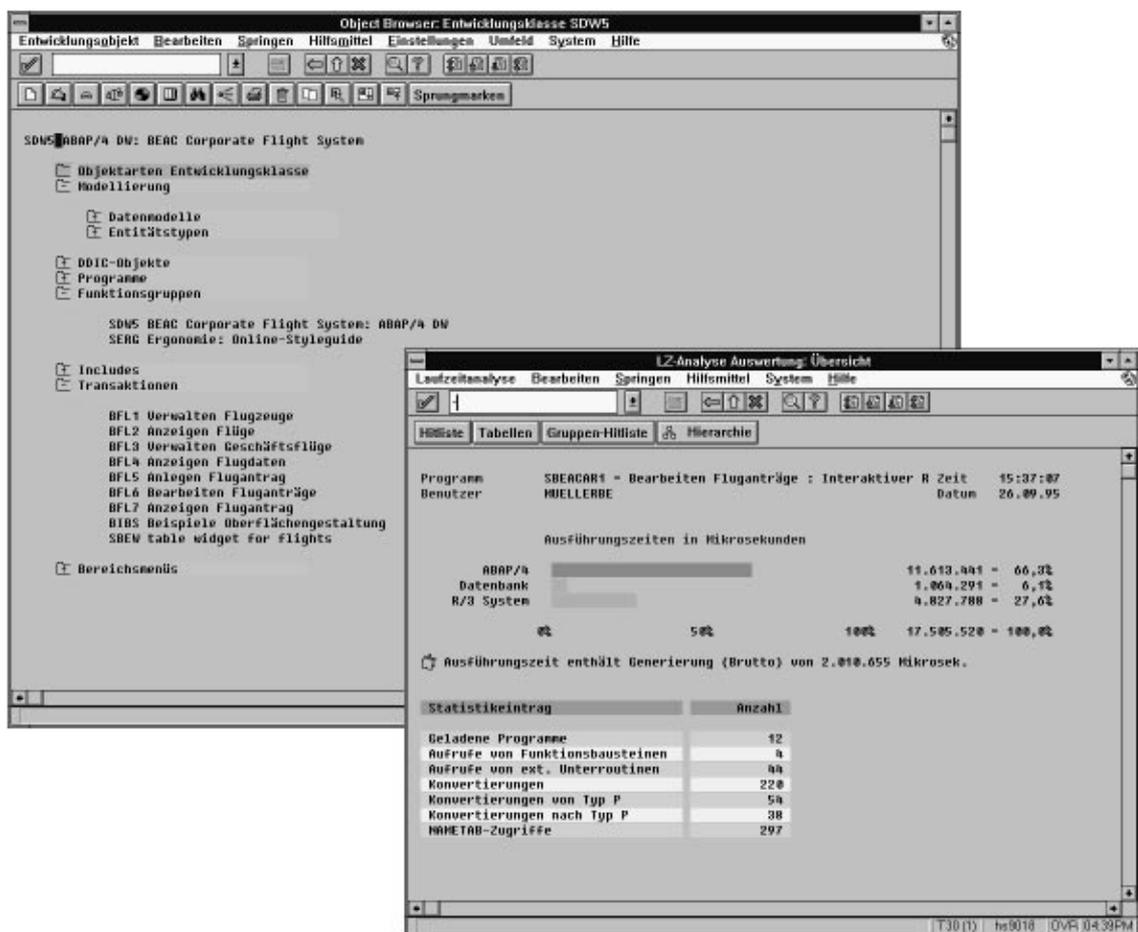


Abb. 2-2: Benutzeroberfläche der ABAP/4 Development Workbench

Vorgedachte betriebswirtschaftliche Funktionen

Von besonderer Bedeutung ist, daß zusammen mit den R/3-Entwicklungswerkzeugen eine umfangreiche Bibliothek vorgedachter betriebswirtschaftlicher Funktionen ausgeliefert wird, die leicht in individuell erstellte Programme übernommen werden können. Das entlastet den Programmierer von Routineaufgaben und verbessert so Produktivität und Qualität.

Das aktive ABAP/4 Dictionary

Software-Entwicklungsumgebungen benötigen Metadaten, d.h. Beschreibungen der innerhalb der erstellten Programme verwendeten Datenstrukturen. Zu diesen Metadaten gehören z.B. Tabellen- und Felddefinitionen, Festlegungen von Wertebereichen (Domänen) oder die Beschreibung von Beziehungen zwischen Tabellen. Im System R/3 werden diese Metadaten im ABAP/4 Dictionary beschrieben. Das ABAP/4 Dictionary selbst ist im ABAP/4 Repository abgelegt.

Metadaten im ABAP/4 Dictionary

Im ABAP/4 Dictionary werden sowohl die Strukturen der in der Datenbank gespeicherten Daten als auch interne Strukturen von Programmen sowie deren Schnittstellen beschrieben. Die Feldeigenschaften von relationalen Tabellen sind mit Hilfe eines zweistufigen Domänenkonzepts definiert. Beziehungen zwischen Tabellen werden durch Fremdschlüssel beschrieben. Externe Sichten auf Tabellen können durch Views definiert werden.

Zu den programminternen Strukturbeschreibungen gehört z.B. die Datentyp-Definition für Felder, die nur in einem Dynpro, nicht aber in einer Datenbanktabelle auftauchen, oder die Definition von Tabellen, die nur temporär in einem ABAP/4-Programm abgelegt werden. Internen Strukturen können z.B. für den Austausch von Daten zwischen Dynpros und ABAP/4-Programmen oder für die Definition von Schnittstellen, etwa bei der Parameterübergabe an Funktionsbausteine, verwendet werden. Alle automatischen Serviceleistungen, bei denen das ABAP/4 Dictionary mitwirkt, wie Online-Hilfe, Formatprüfungen etc., stehen auch für diese internen Strukturen zur Verfügung. Damit gehen die im ABAP/4 Dictionary enthaltenen Informationen weit über das hinaus, was der Datenbankkatalog des jeweils verwendeten Datenbank-Systems enthält. Um die Konsistenz und einfache Benutzung sicherzustellen, werden Tabellen- und View-Definitionen automatisch für alle unterstützten Datenbank-Systeme aus dem ABAP/4 Dictionary generiert.

Internen Strukturen

Das ABAP/4 Dictionary ist ein aktives Dictionary. Das bedeutet, daß sich Änderungen im ABAP/4 Dictionary automatisch und sofort auf die betroffenen Anwendungsprogramme auswirken können. Die R/3 Laufzeitumgebung sorgt dafür, daß stets auf die aktuellen Informationen des ABAP/4 Dictionary zugegriffen wird. Wenn z.B. ein neuer Wert in den Wertevorrat einer Domäne eingetragen wird, hat das zur Folge, daß in allen Dynpros, die ein Feld mit dieser Domäne enthalten, die Hilfefunktion sofort auch diesen Wert mit der Liste der möglichen Werte anzeigt. Neben dieser sofortigen Aktivierung bietet das ABAP/4 Dictionary dem Entwickler auch die Möglichkeit, Änderungen zunächst in einer nicht-aktiven Version zu sichern. Damit können z.B. Änderungen an mehreren voneinander abhängigen Objekten durchgeführt werden, die anschließend gemeinsam aktiviert werden sollen.

Aktives ABAP/4 Dictionary

Grundlage für die aktive Einbindung des ABAP/4 Dictionary in den Programmablauf ist die interpretative Arbeitsweise des Systems R/3.

Der ABAP/4-Prozessor arbeitet nicht direkt mit dem Original-Coding eines ABAP/4-Programms oder Dynpros, sondern interpretiert eine aus dem Coding generiertes Laufzeitobjekt, das aus einer Ereignis-, Verarbeitungs- und Datentabelle besteht. Dadurch wird sichergestellt, daß die aktive Einbindung der Dictionary-Informationen nicht zu Lasten der Effizienz geht. Laufzeitobjekte beziehen sich zwangsläufig auf den Stand des Dictionary zum Zeitpunkt der Generierung. Wenn beim Start eines Programms oder Dynpros durch Überprüfung des zu jedem Objekt gehörenden Zeitstempels festgestellt wird, daß seit der Generierung die zugehörigen Informationen im ABAP/4 Dictionary geändert wurden, so findet automatisch eine Nachgenerierung statt. Dabei wird der aktuelle Stand des Dictionary zugrundegelegt

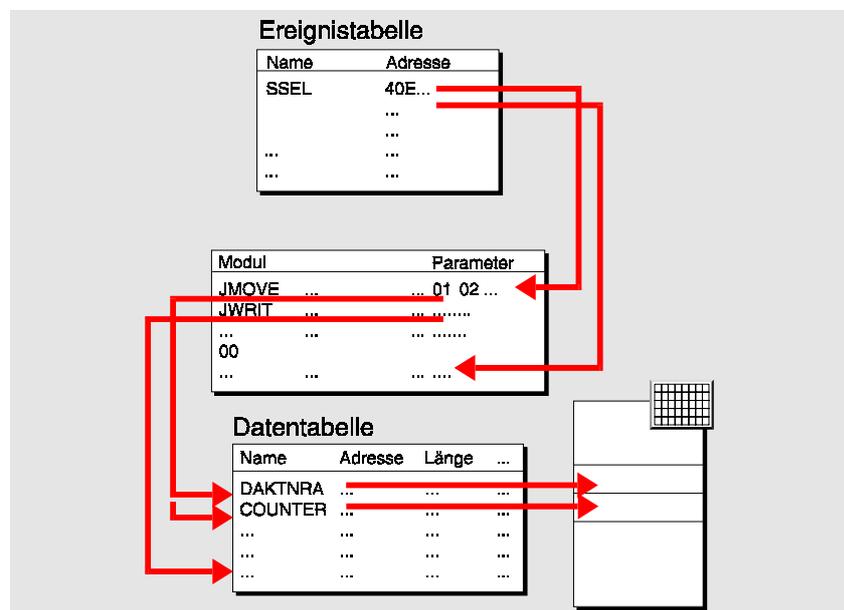


Abb. 8-3: Generiertes Laufzeitobjekt

Integriertes ABAP/4 Dictionary

Das ABAP/4 Dictionary ist vollständig in die ABAP/4 Development Workbench integriert: Im ABAP/4 Editor können Tabellendefinitionen des ABAP/4 Dictionary direkt angezeigt werden. In Anwendungsprogrammen werden die im ABAP/4 Dictionary definierten Tabellenstrukturen direkt verwendet. Interne Strukturen können sich auf Tabellen- oder Felddefinitionen des ABAP/4 Dictionary beziehen.

Tabellen und Felder

Das zentrale Konzept des ABAP/4 Dictionary ist die Tabelle, die aus Feldern besteht. Alle Eigenschaften von Feldern in den generierten Datenbanktabellen, in den Dynpros und in den ABAP/4-Programmen werden im ABAP/4 Dictionary definiert.

Eine Tabellendefinition benennt die Felder der Tabelle und spezifiziert für jedes Feld Eigenschaften, wie z.B. Format, Länge usw., sowie seine Zugehörigkeit zum Schlüssel. Darüber hinaus vermerkt sie, welche Beziehungen zu anderen Tabellen von der Tabelle ausgehen.

Tabellendefinition

Die Eigenschaften einzelner Tabellenfelder werden mit Hilfe eines zweistufigen Konzepts aus Domänen und Datenelementen beschrieben:

Domänen und Datenelemente

- ❑ Domänen spezifizieren die technischen Eigenschaften wie Datentyp, Länge, Wertebereich, Bildschirmdarstellung etc.
- ❑ Datenelemente beschreiben die betriebswirtschaftliche Bedeutung und geben für jedes Feld Schlüsselwörter, Überschriften, einen Dokumentationstext sowie eine Domäne vor. Die Dokumentationstexte werden unter anderem von der Online-Hilfe benutzt.

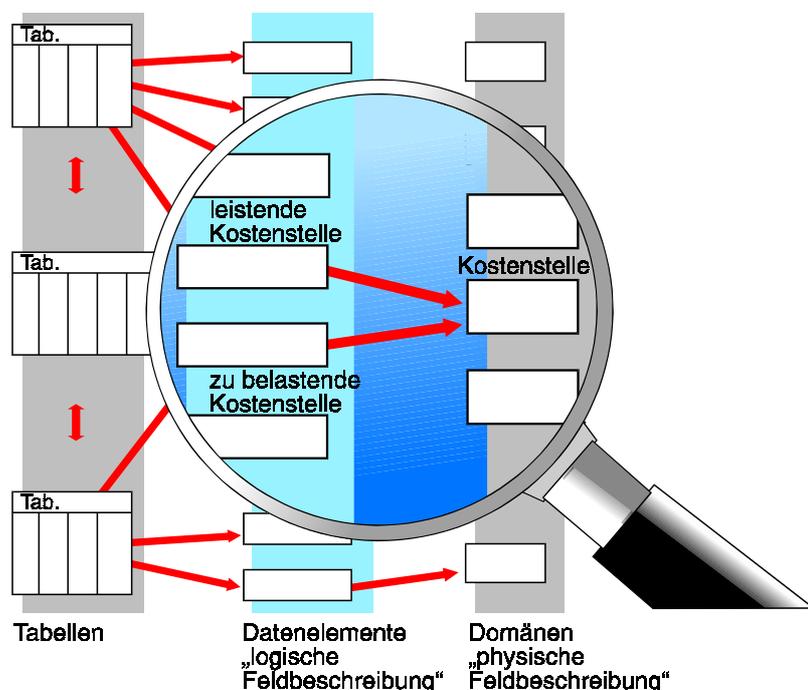


Abb. 8-4: Tabellen, Datenelemente und Domänen

Da Domänen und Datenelemente eigenständige Objekte im ABAP/4 Dictionary sind, kann eine Domäne in beliebig vielen Datenelementen vorkommen. Ein Datenelement kann wiederum mehrfach für Felder in Tabellendefinitionen verwendet werden. Darüber hinausgehende Möglichkeiten bietet das Include-Verfahren mit der Einbettung ganzer Datenstrukturen in Tabellen. Wird eine Struktur, die in andere Tabellen eingebettet ist, später verändert, so passen sich automatisch alle betroffenen Tabellen an. Insgesamt wird durch Domänen, Datenelemente und Include-Verfahren nicht nur ein hoher Wiederverwendungsgrad erreicht, sondern auch die automatische Konsistenzhaltung von Feldern mit gleicher Bedeutung garantiert.

Include Verfahren

Datenstrukturen, die bei Berechnungen innerhalb von Programmen oder bei

Strukturen

Datenübergabe zwischen Programmen benötigt werden, können ähnlich wie Tabellen zentral im ABAP/4 Dictionary definiert und gepflegt werden. Der einzige Unterschied ist, daß keine Datenbanktabellen aus ihnen generiert werden. In internen Strukturen können die gleichen Datenelemente und Domänen genutzt werden wie in Tabellen. Auch ein Include von Strukturen ist möglich. Damit wird auch bei umfangreichen Programmen, die intern Informationen aus mehreren Tabellen verknüpfen, ein hohes Maß an Konsistenz der Datendefinition erreicht.

Fremdschlüssel Mit Fremdschlüsseln werden Beziehungen zwischen Feldern definiert. Ein Fremdschlüssel definiert die Zuordnung von Feldern der abhängigen Tabelle zu den Schlüsselfeldern einer anderen, referierten Tabelle.

Materialien			Werke	
Mat.	Buchkreis	Werk	Buchkreis	Werk
44757	01	01	01	01
45857	01	01	01	02
89013	02	01	02	01
23005	02	03	02	03
33462	02	03	03	04
70382	03	04		
50748	02	02		

Abb. 8-5: Fremdschlüsselbeziehungen

Mit einem Fremdschlüssel wird sichergestellt, daß für die Felder der abhängigen Tabelle nur solche Werte zugelassen werden, die auch als Werte der Schlüsselfelder in der referierten Tabelle vorkommen. Referierte Tabellen werden deshalb auch als Prüftabellen bezeichnet. Die Prüfung findet schon sehr früh bei der Eingabe in die Bildschirmmasken statt und nicht erst beim Schreiben in die Datenbank. Auch in Strukturen, die in Bildschirmmasken erscheinen, können Fremdschlüssel für Prüfzwecke definiert werden.

Wertetabellen Um die Definition von Fremdschlüsseln zu erleichtern, gibt es im ABAP/4 Dictionary das Konzept der Wertetabellen für Domänen. Dieses ermöglicht ergänzend zur Wertebereichsbeschreibung durch Festwerte (z.B. 1,2,...,12) auch die Definition von dynamisch veränderlichen Wertebereichen. Die Domäne wird einem Schlüsselfeld der Wertetabelle zugeordnet. Die Daten dieser Tabelle bestimmen dann die möglichen Werte der Domäne.

Views Die im relationalen Modell übliche Methode, anwendungsbezogene Datensichten in Form von Views zur Verfügung zu stellen, kann auch im System R/3 genutzt werden. Im ABAP/4 Dictionary können Views als virtuelle Tabellen definiert werden, die keine eigenen Daten enthalten, sondern lediglich eine anwendungsorientierte Sicht auf andere, im ABAP/4 Dictionary verzeichnete Tabellen ermöglichen. Ein View enthält eine Auswahl der Felder einer Tabelle bzw. der Felder mehrerer, durch Fremdschlüssel verbundener Tabellen. Die Felder eines Views können andere Namen tragen als die der Originaltabellen. Durch die Angabe von Selektionskriterien können einzelne Zeilen der Originaltabelle für den View ausgeblendet werden.

Matchcodes

Matchcodes sind Indizes auf Originaldaten, mit denen interaktiv und schnell nach bestimmten Informationen gesucht werden kann. Matchcodes werden

mit Hilfe von Datenbank-Indizes, über Views oder als eigene, über Trigger fortgeschriebene Tabelleneinträge realisiert. Sie vereinigen Daten aus mehreren Originaltabellen, wobei gewisse Felder unterdrückt werden können. Für jeden Matchcode-Eintrag gibt es einen Verweis auf die Originaldaten. Im Unterschied zu Datenbank-Indizes können Matchcodes Felder mehrerer Tabellen beinhalten. Außerdem ist eine Einschränkung auf bestimmte Datensätze durch die Angabe von speziellen Selektionsbedingungen möglich. Matchcodes können auch von Fremdanwendungen genutzt werden.

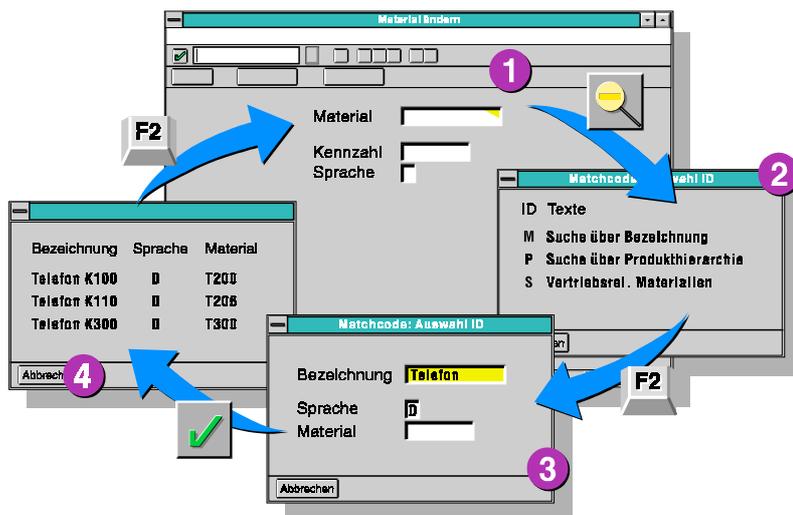


Abb. 8-6: Suche mit Hilfe von Matchcodes

Automatische Funktionen für Bildschirmfelder

Normalerweise beziehen sich alle Felder einer Bildschirmmaske auf Felder von Tabellen, die im ABAP/4 Dictionary definiert sind. Länge und Datentyp des Bildschirmfeldes werden ebenso aus dem Dictionary übernommen wie der Text, der die betriebswirtschaftliche Bedeutung des Feldes beschreibt.

Bei der Ein- oder Ausgabe von Feldinhalten werden gegebenenfalls im Dictionary hinterlegte Konvertierungsroutinen zwischen interner und externer Darstellung ausgeführt. Betrags- und Mengenfelder können gemäß im Dictionary zugeordneter Referenzfelder, in denen die entsprechenden Währungs- oder Mengeneinheiten verzeichnet sind, dargestellt werden.

Zu jedem Bildschirmbild steht dem Anwender über die F1-Taste eine Felddokumentation zur Verfügung. Sie enthält die betriebswirtschaftliche Beschreibung aus dem zugeordneten Datenelement und die technische Felddefinition aus der zugehörigen Domäne.

Darstellung von
Bildschirmfeldern

Konvertierungsroutinen

Eingabehilfen

Über die F4-Taste kann der Anwender zu bestimmten Feldern eine Liste der möglichen Eingabewerte anzeigen und den gewünschten Wert durch einfachen Maus-Klick auswählen. Diese Hilfe steht immer dann zur Verfügung, wenn ein Matchcode, eine Prüftabelle oder Festwerte für ein Feld definiert wurde.

Prüfung von Bildschirmeingaben

Bildschirmeingaben werden automatisch gemäß dem im ABAP/4 Dictionary definierten Wertebereich des Feldes auf Gültigkeit überprüft. Insbesondere werden dabei Fremdschlüsseldefinitionen berücksichtigt. Die eingegebenen Werte werden in der Prüftabelle selektiert und nur dann akzeptiert, wenn sie dort vorhanden sind. Diese Vorgehensweise hat gegenüber einer späten Prüfung beim Schreiben in die Datenbank den entscheidenden Vorteil, daß Fehler so früh wie möglich erkannt werden und der Anwender die Eingabe unmittelbar korrigieren kann.

Generierung von Datenbankdefinitionen

Automatische Katalogänderungen

Das im ABAP/4 Dictionary definierte relationale Datenmodell ist unabhängig von den Datenbanksystemen spezieller Hersteller. Katalogdefinitionen und Katalogänderungen können für alle vom System R/3 unterstützten Datenbanksysteme automatisch generiert und durchgeführt werden. Die im Dictionary definierten Tabellen werden dabei 1:1 auf das unterliegende Datenbanksystem abgebildet.

Aus dem ABAP/4 Dictionary heraus können Datenbank-Tabellen, -Indizes und -Views komfortabel angelegt, gelöscht und modifiziert werden. Der Entwickler benötigt dazu keine datenbankspezifischen Kenntnisse. Sogar Parameter für die Speicherung von Tabellen in der Datenbank können aus den datenbankunabhängigen Kenngrößen der Tabelle abgeleitet werden.

ALTER TABLE

Zur Modifikation bestehender Tabellen werden vom System R/3 alle Möglichkeiten ausgenutzt, die das jeweilige Datenbanksystem zum ALTER TABLE bietet, z.B. INSERT FIELD, ADD FIELD etc. Ist ein ALTER TABLE nicht möglich, so werden automatisch Programme generiert, die die Daten der betroffenen Tabelle entladen und in die neue Struktur zurückspeichern.

Das Unternehmensdatenmodell

Die Tabellendefinitionen des ABAP/4 Dictionary orientieren sich an technischen Aspekten wie Effizienz der Entwicklung und Leistung der Anwendungsprogramme. Im Unterschied dazu berücksichtigt das Unternehmensdatenmodell (UDM) ausschließlich betriebswirtschaftliche Gesichtspunkte.

Entity-Relationship-Modell

Das Ziel der Erstellung von Datenmodellen ist es, die innerhalb der betrieblichen Leistungserstellung benötigten Daten formal in einem funktionsübergreifenden Zusammenhang abzubilden. Das Datenmodell der SAP stellt die unternehmensrelevanten Informationsobjekte sowie deren Beziehungen zueinander aus betriebswirtschaftlicher Sicht dar, und zwar in Form eines strukturierten Entity-Relationship-Modells. Entitätstypen sind Abbildungen

von Objekttypen der realen Welt, die eine betriebswirtschaftliche Bedeutung und eine Entsprechung im System R/3 haben. Die Beziehungen zwischen den Entitätstypen sind, wie die Fremdschlüsselbeziehungen, nach ihrem Typ und ihrer Kardinalität klassifiziert. Entitätstypen können durch Spezialisierung in Untertypen verfeinert werden.

Das Unternehmensdatenmodell der SAP ist in Teilmodelle für die einzelnen R/3-Anwendungen untergliedert. Datenmodelle können aus der ABAP/4 Development Workbench heraus sowohl textuell als auch grafisch dargestellt werden. Die grafische Darstellung dieser Modelle folgt immer einer einheitlichen Struktur. Dadurch wird die Orientierung erheblich vereinfacht. Als Ausgangspunkt für die grafische Darstellung können sowohl Teildatenmodelle als auch einzelne Entitätstypen gewählt werden. Die dynamische Veränderung des angezeigten UDM-Ausschnitts ist in jedem Fall möglich. Durch das Ein- und Ausblenden von Entitätstypen hat der Anwender die Möglichkeit, den aktuell benötigten Ausschnitt des Datenmodells individuell zusammenzustellen.

Grafische Darstellung

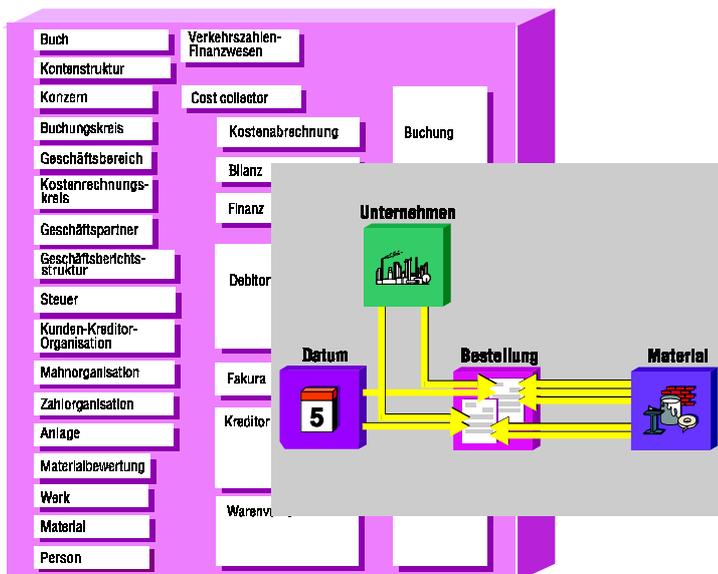


Abb. 8-7: Grafische Darstellung des UDMs

Zwischen dem SAP-Datenmodell und dem ABAP/4 Dictionary besteht ein enger Zusammenhang. Allerdings besteht die Möglichkeit, daß Entitätstypen sowohl auf mehrere Tabellen verteilt als auch mehrere Entitätstypen in einer Tabelle zusammengefaßt sind. Daher benutzt man im ABAP/4 Dictionary Views, um die Verbindung zwischen den Entitätstypen des Datenmodells und den Tabellen des ABAP/4 Dictionary exakt zu beschreiben. Die Felder der Tabellen entsprechen dabei den Attributen der Entitätstypen. Man kann sich über Views die Ausprägung von Entitätstypen anzeigen lassen und mit ihnen auf betriebswirtschaftlicher Ebene programmieren. Auch im SAP-Datenmodell wird das zweistufige Beschreibungskonzept mit Domänen und Datenelementen verwendet. Gleiche Domänen und Datenelemente im Datenmodell und im ABAP/4 Dictionary garantieren automatisch die Konsistenz zwischen beiden Sichtweisen.

Integration in das ABAP/4 Dictionary

Dialogprogrammierung mit Dynpro-Technik

Dynamisches Programm Jeder Dialog und jede Hintergrundverarbeitung in einem SAP-System wird gesteuert von Dynpros. Ein Dynpro (Dynamisches Programm) besteht aus einem Bildschirmbild und dessen Ablauflogik inklusive Prüfprozeduren und kontrolliert jeweils genau einen Dialogschritt. Dabei ist aber zu beachten, daß der Benutzer und das System unterschiedliche Auffassungen von der Einteilung des Dialogs in einzelne Schritte haben:

Aus der Sicht des Benutzers besteht ein Dialogschritt darin, daß ihm ein Bild gesendet wird, daß er in diesem Bild Daten einträgt und dann per Drucktaste oder Menü eine Verarbeitung auslöst.

Aus der Sicht des SAP-Systems beginnt ein Schritt mit dem Vorbereiten und Senden einer Bildschirmmaske; danach folgen die Entgegennahme des vom Benutzer ausgefüllten Bildschirms und die Analyse und Verarbeitung der im empfangenen Bild enthaltenen Eingabedaten. Dialogschritte für das System und für den Benutzer sind daher zeitlich gegeneinander versetzt. Die beiden Hälften des Dialogschritts aus Systemsicht heißen PBO (Process Before Output) und PAI (Process After Input). Ein Dynpro spezifiziert im wesentlichen dreierlei:

- Das Bildschirmlayout bestimmt die Position von Eingabe-, Ausgabe-, Auswahlfeldern usw, von grafischen Bedienelementen und Texten.
- Für alle Bildschirmfelder verzeichnet das Dynpro ihre Feldeigenschaften. Festgelegt werden hier u.a. Formate, Wertebereichsprüfungen und Besonderheiten der Darstellung (links-/ rechtsbündig, führende Nullen, Farbgebung usw).
- Die Ablauflogik legt fest, welche Verarbeitungen zu den Zeitpunkten PBO und PAI abzuwickeln sind.

Dynpros können mehrsprachig ausgelegt werden mit sprachabhängigen Dokumentations-/Hilfetexten, Fehlermeldungen und Bildschirmmasken.

Dynproketten und Modulpools

Dialogprogramm Jedes Dynpro bezieht sich auf genau ein ABAP/4-Dialogprogramm. Ein solches Dialogprogramm wird auch als Modulpool bezeichnet, da es sich in Dialogmodule gliedert. In der Ablauflogik eines Dynpros stehen auch Aufrufe von Modulen des dem Dynpro zugeordneten Modulpools.

Modulpool Alle Dynpros, die innerhalb einer Transaktion aufgerufen werden sollen, müssen sich auf einen gemeinsamen Modulpool beziehen. Die einzige Ausnahme von dieser Regel gilt dann, wenn innerhalb einer Transaktion Funktionsbausteine mit eigenen Dynpros aufgerufen werden. Es ist üblich, daß sich mehrere Transaktionen aus demselben Modulpool bedienen. Dabei dürfen Dynpros durchaus in mehreren Transaktionen verwendet werden; gegebenenfalls müssen dann die Module, welche die Ablauflogik aufrufen, durch spezielle Folgebildsteuerungen, spezielles Ausblenden von Feldern usw. dafür sorgen, daß das Dynpro in mehreren Dialogabläufen sinnvoll eingesetzt werden kann.

Die Dynpros zu einem Modulpool sind numeriert. Zu jedem Dynpro ist ein Standardfolgedynpro vermerkt, so daß sich eine statisch vorgegebene Dynprokette ergibt. Eine solche Kette endet, sobald als Folgedynpronummer eine 0 auftaucht; alternativ kann eine Kette aber auch in einen Zyklus laufen. Im Verlauf einer Transaktion können die gerufenen Dialogmodule - etwa als Reaktion auf gewisse Eingaben des Benutzers oder auf die Feststellung hin, daß ihm bestimmte Berechtigungen fehlen - erwirken, daß von der statischen Dynprokette abgewichen wird. Sowohl der Aufruf einer anderen Dynprokette, wobei nach deren Ende automatisch zur Aufrufstelle zurückgekehrt wird, als auch der endgültige Übergang zu einer anderen Dynprokette sind möglich.

Dynproketten

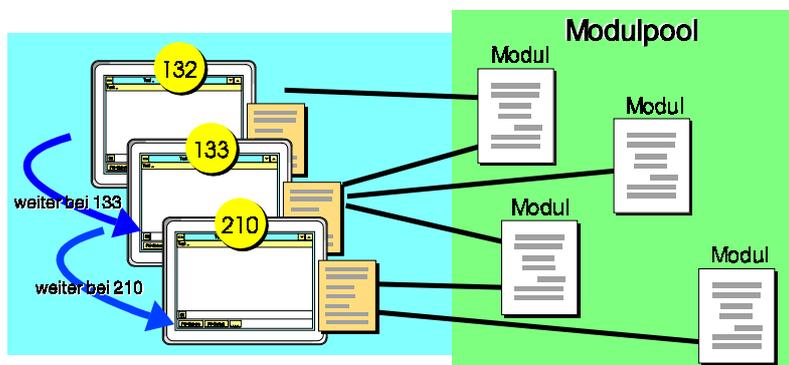


Abb.8-8: Dynproketten und Dialogmodule

Die Dialogmodule können ihrerseits neue Transaktionen aufrufen. Auch hier sind wahlweise die Rückkehr beim Ende der gerufenen Transaktion oder das endgültige Verlassen der alten Transaktion möglich. Für die gerufene Transaktion wird ein eigener, interner Modus neu eröffnet. Dem Benutzer bleiben solche internen Übergänge zwischen Transaktionen in der Regel verborgen.

Feldeigenschaften und Standardfunktionen

Bei gleicher Namensgebung für ein Feld im ABAP/4 Dictionary und im Dynpro besteht automatisch ein Bezug zwischen den beiden Feldern. Dies führt dazu, daß ein großer Teil der Feldeigenschaften im Dynpro aus dem ABAP/4 Dictionary übernommen wird. Die Feldeigenschaften, sowie eventuell Datenelement und Domäne des zugeordneten Dictionary-Feldes, bilden die Grundlage für die Standardfunktionen, die das Dynpro im Dialog leistet:

- ❑ Der Dynpro-Prozessor führt automatische Formatprüfungen für die Bildschirmfelder anhand der entsprechenden Angaben im Dynpro durch.
- ❑ Für automatische Wertebereichsprüfungen legt der Dynpro-Prozessor entweder die Prüftabelle des Dictionary-Feldes oder den festen Wertebereich in seiner Domäne zugrunde.

- ❑ Online-Hilfe, inklusive des Nachschlagens im Glossar, wickelt der Dynpro-Prozessor ebenfalls eigenständig ab. Er verwendet dabei u.a. die Hilfetexte zu den Datenelementen der Felder.
- ❑ Der Dynpro-Prozessor konvertiert die Benutzereingaben in interne Repräsentationen und stellt sie für das Dialogprogramm bereit. Umgekehrt erledigt er auch die Aufbereitung auszugebender Daten unter Berücksichtigung der entsprechenden Feldeigenschaften im Dynpro.
- ❑ Spezielle Feldeigenschaften können den Dynpro-Prozessor anweisen, Vorschlagswerte für ein Feld aus einem bestimmten Benutzerparameter zu holen bzw. im Falle von Eingaben in das Feld diese in einen bestimmten Benutzerparameter zu übernehmen.
- ❑ Die Matchcode-Suche ist eine weitere Standardfunktion des Dynpro-Prozessors. Für Bildschirmfelder, die die Eingabehilfe durch Matchcodes unterstützen, muß dieses als Feldeigenschaft im Dynpro verzeichnet sein.

Ablauflogik

PBO/ PAI Die Ablauflogik eines Dynpros zerfällt in zwei Hälften, von denen die eine bei PBO (Process Before Output), die andere bei PAI (Process After Input) interpretiert wird. Beide Hälften definieren je eine sequentielle Abfolge von Befehlen.

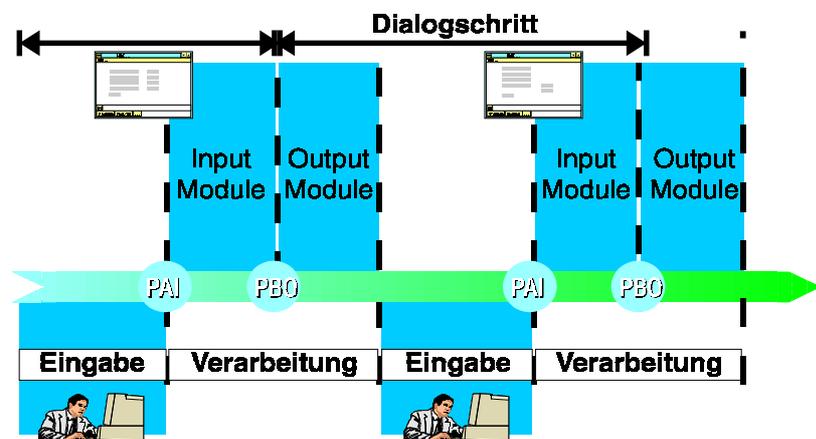


Abb.8-9: Dialogschritte aus Benutzer- und Systemsicht

Die Befehle in der Ablauflogik sind syntaktisch an die Konstrukte der Sprache ABAP/4 angelehnt und können folgendes veranlassen:

- ❑ den Aufruf eines ABAP/4-Dialogmoduls.
- ❑ eine Prüfung von Feldinhalten gegen eine Tabelle oder einen festen Wertebereich.
- ❑ eine Berechtigungsprüfung im Stammsatz des Benutzers.

Dialogmodule, die zum Zeitpunkt PBO aufgerufen werden, sollen die Bildschirmmaske kontextabhängig vorbereiten, z.B. durch Setzen von Feldinhalten oder durch Ausblenden nicht benötigter Felder im Bildschirmlayout. Dialogmodule, die zum Zeitpunkt PAI aufgerufen werden, sollen die Eingaben des Benutzers überprüfen und geeignete Dialogschritte bzw. die Verbuchung veranlassen.

Fehlerdialoge

Eine weitere wesentliche Aufgabe des Dynpro-Prozessors ist die autonome Durchführung von Fehlerdialogen. Fehler können festgestellt werden:

- bei den automatischen Prüfungen, die der Dynpro-Prozessor aufgrund der Feldeigenschaften durchführt,
- bei Prüfungen, die die Befehle in der Ablauflogik verlangen, und
- bei den komplexeren Prüfungen, welche die Dialogmodule vornehmen.

Dementsprechend ist die anzuzeigende Fehlermeldung für den Benutzer entweder eine Standardmeldung oder eine Meldung, die in der Ablauflogik bzw. im Dialogmodul ausgewählt wird. Der Dynpro-Prozessor blendet die Fehlermeldung in das empfangene Bildschirmbild ein und sendet es an den Benutzer zurück. Dabei sorgt er dafür, daß:

Kontextabhängige Meldungen

- die Meldung gegebenenfalls kontextabhängig angepaßt wird durch das Einsetzen von aktuellen Feldinhalten für Platzhalter im Meldungstext, und daß
- nur diejenigen Felder eingabebereit gemacht werden, für die eine Korrektur ihres Inhaltes zur Behebung des Fehlers beitragen kann. Welche das sind, entscheidet er anhand der logischen Klammerungen und der Feldabhängigkeiten in der Ablauflogik.

Nach der Korrektur eines Fehlers durch den Benutzer durchläuft der Dynpro-Prozessor in der Regel nicht alle PAI-Befehle erneut, sondern nur solche, die von den tatsächlich vorgenommenen Änderungen der Eingaben betroffen sein könnten. Auch hier dienen die logischen Verknüpfungen von Feldern und Befehlen bzw. die Blockbildung von Befehlen in der Ablauflogik als Anhaltspunkte.

Anwendungsprogrammierung mit ABAP/4

Advanced Business Application Programming

Um Anwendungsentwicklern optimale Arbeitsbedingungen zu schaffen, konzipierte die SAP in den frühen achtziger Jahren die Programmiersprache ABAP/4 (Advanced Business Application Programming). Die Sprache wird ständig weiterentwickelt, insbesondere orientiert an den Kundenanforderungen im betriebswirtschaftlichen Umfeld.

Die SAP verwendet heute für alle Programmentwicklungen im Anwendungsbereich ausschließlich ABAP/4. ABAP/4 zählt zu den 4GL-Sprachen (4th Generation Languages). Das heißt im einzelnen:

- ABAP/4 bildet eine integrierte Architektur zusammen mit weiteren Komponenten wie dem R/3 Repository und den einzelnen Entwicklungswerkzeugen.
- ABAP/4 ermöglicht eine Programmentwicklung ohne spezifische Kenntnisse des technischen Umfeldes (z.B. Datenbank, Betriebssystem, Netzwerk, Client/Server-Kommunikation).
- ABAP/4 unterstützt strukturierte Programmierung. Alle üblichen Kontrollstrukturen und Konzepte für die Modularisierung von Programmsystemen sind in der Sprache enthalten.
- ABAP/4 ist eine portabel kompilierende Sprache. Programme werden in eine interne Repräsentation transformiert, die für die Interpretation zur Laufzeit optimal ist. Die Verbindung zwischen einem ABAP/4-Programm und den von ihm verwendeten externen Daten wird durch das ABAP/4 Dictionary realisiert. Änderungen in der Organisation der Daten verlangen daher kein Neudesign des vorhandenen Codings, sondern bewirken lediglich eine automatische Nachgenerierung des Laufzeitobjekts unter der Kontrolle des ABAP/4 Dictionary.
- ABAP/4 ist vom Sprachumfang her speziell auf den Einsatz im Rahmen betrieblicher Informationssysteme zugeschnitten.

Software-Entwicklung mit Prototyping

ABAP/4 ermöglicht eine durch Prototyping unterstützte Programmentwicklung. Der Entwickler erstellt zunächst eine vorläufige Version des Programms und der Bedieneroberfläche, und fügt dann fehlende Funktionalität hinzu. Der interpretative Charakter der Sprache erleichtert es dabei, Zwischenversionen zu erstellen, die bereits lauffähig sind. Diese können dann umgehend, idealerweise schon unter Einbeziehung der Endanwender, beurteilt und getestet werden. Zwischenversionen können vollständig in die Endversion einbezogen werden (kein "Throw Away"-Prototyping).

Aus ABAP/4-Programmen heraus ist jeder Zugriff auf externe Kanäle des Systems R/3 möglich, sei es CPI-C, RFC, Spool, sequentielle Datenbestände, Telekom-Dienste o.ä.

Struktur der Sprache ABAP/4

Die ABAP/4-Sprachelemente sind in vier Kategorien unterteilbar:

- ❑ **Deklarative Sprachelemente** beschreiben den Aufbau der Daten, die in einem Programm verarbeitet werden (z.B. TABLES, DATA).
- ❑ **Operationale Sprachelemente** ermöglichen elementare Datenmanipulationen (z.B. ADD, MOVE).
- ❑ **Steuerungs-Sprachelemente** realisieren Kontrollstrukturen wie Schleifen, Verzweigungen, Unterprogramme etc. (z.B. DO, IF, CASE, PERFORM).
- ❑ **Zeitpunkt-Sprachelemente** verknüpfen Programmteile mit bestimmten Ereignissen, die bei der Abarbeitung eines Programms eintreten können (z.B. TOP-OF-PAGE, AT USER-COMMAND).

Mehrsprachigkeit

Um die Mehrsprachigkeit des Systems zu gewährleisten, können einem Programm mehrsprachige Texte, die z.B. auf Reportlisten ausgegeben werden sollen, als Textelemente zugeordnet werden. Sie können nachträglich verändert werden, ohne daß ein Eingriff in das Coding notwendig wäre. Im Coding selbst taucht lediglich die Nummer des Textelements auf. Texte werden jeweils in der Sprache ausgegeben, die mit dem Sprachkennzeichen bei der Anmeldung eines Benutzers gewählt wurde.

**Sprachabhängige
Textpools**

Von zunehmender Bedeutung ist die Unterstützung piktographischer Sprachen mit ihrer umfangreichen Zeichenmenge wie z.B. Japanisch und Chinesisch. SAP trägt diesem Umstand durch die Berücksichtigung des Zwei-Byte-Code-Systems (DBCS - Double Byte Character Set) Rechnung. Die ABAP/4 Development Workbench wurde entsprechend erweitert.

Double Byte Character Set

ABAP/4 wurde z.B. um den Datentyp W (Wide Character) und eine Reihe weiterer Funktionen für die String-Behandlung erweitert, damit Double-Byte-Zeichen in einem Zeichenfeld gemischt mit normalen Ein-Byte-Zeichen existieren können.

Stringbehandlung

Elementare betriebswirtschaftliche Datentypen und Operationen

Die Datentypen und Operationen von ABAP/4 orientieren sich an den Bedürfnissen betriebswirtschaftlicher Anwendungssysteme. Beispielsweise wird neben der reinen Arithmetik auch die Datumsrechnung speziell unterstützt. Außerdem enthält ABAP/4 umfangreiche Funktionen für die Verarbeitung textueller Daten.

Für Datums- und Zeitangaben gibt es eigene Datentypen. In gemischten Ausdrücken werden durch automatische Konvertierungen Zahlen als Längen von Zeitintervallen interpretiert. Dadurch liefert z.B. die Addition einer Zahl n zu einem Datum das um n Kalendertage verschobene Datum; die Subtraktion zweier Zeitangaben liefert ihre Differenz in Sekunden.

Die üblichen Operationen auf Zeichenketten bzw. Tabellen, deren Zeilen Zeichenketten sind, gehören ebenfalls zum Sprachumfang von ABAP/4, wie z.B. das Suchen und Ersetzen einer vorgegebenen Zeichenkette, das Eliminieren überflüssiger Leerzeichen, das Verschieben einer Zeichenkette, das Überprüfen, ob Zeichen aus einer Zeichenkette in einer anderen vorkommen, usw.

Tabellenverarbeitung und SQL

Da die Tabelle die wesentlichste Datenstruktur im System R/3 ist, bietet ABAP/4 reichhaltige Funktionen für die Handhabung tabellarischer Daten.

Open SQL Die in ABAP/4 integrierten Open SQL-Sprachmittel unterstützen lesende und verändernde Zugriffe auf die im ABAP/4 Dictionary definierten logischen Tabellen. Die entscheidenden Vorteile der Integration sind:

- die leistungssteigernden Puffermechanismen der R/3-Datenbankschnittstelle,
- die uneingeschränkte Portabilität der erstellten Anwendungen auf alle unterstützten Datenbanksysteme,
- und der Syntax-Check des ABAP/4-Programm-Editors überprüft die im ABAP/4-Coding eingebetteten Open SQL-Statements direkt.

Native SQL Für Native SQL-Statements, die auch proprietäre Erweiterungen der jeweiligen Datenbank nutzen und uninterpretiert an die Datenbank weitergeleitet werden, gelten die genannten Vorteile nicht. Programmierfehler in Native SQL-Anweisungen werden dadurch erst als Laufzeitfehler erkannt. Native SQL hat zusätzlich den Nachteil, nicht vollständig portabel zwischen verschiedenen Datenbanken zu sein.

Portable Datenbankoperationen Mit den in ABAP/4 integrierten Open SQL-Sprachelementen können sowohl einzelne Zeilen als auch Mengen von Zeilen gelesen, geändert und gelöscht werden. Der relationale Join kann durch View-Definitionen im ABAP/4 Dictionary realisiert werden. Die Datenbankschnittstelle setzt Open SQL-Anweisungen aus den ABAP/4-Programmen in Anweisungen für die relationale Datenbank um. Das ABAP/4-Programm braucht daher weder Besonderheiten der Speicherung noch Inkompatibilitäten zwischen verschiedenen Datenbanksystemen zu berücksichtigen.

Die Datenbankschnittstelle übergibt gelesene Daten aus ABAP/4 Dictionary-Tabellen in speziellen Arbeitsbereichen an das ABAP/4-Programm; geänderte Daten für die Datenbank entnimmt sie aus denselben Bereichen. Bei der Deklaration solcher Arbeitsbereiche unterstützt die Integration des ABAP/4 Dictionary in ABAP/4 den Softwareentwickler. Arbeitsbereiche für Dictionary-Objekte werden automatisch in der richtigen Struktur angelegt. Bei der Deklaration weiterer, programminterner Arbeitsbereiche kann der Entwickler Tabellenstrukturen aus dem ABAP/4 Dictionary als Teilstrukturen aufnehmen. Auch bei der Deklaration einzelner Felder kann er fordern, daß sie in ihren Eigenschaften den im Dictionary verzeichneten Feldern entsprechen.

Datenbankschnittstelle

Im ABAP/4-Programm kann der Entwickler weitere Tabellen anlegen, die jeweils nur für die Dauer eines Programmlaufs existieren. ABAP/4 bietet vielfältige Operationen zur Bearbeitung solcher internen Tabellen, wie z.B. das Suchen, Anfügen, Einfügen und Löschen von Zeilen, das Durchlaufen einer Tabelle in einer Schleife und das Sortieren nach beliebigen Feldinhalten.

Interne Tabellen

Interne Tabellen werden vom System bei Bedarf dynamisch erweitert. Der Vorteil für den Entwickler: Er muß nicht im voraus wissen, wieviel Speicher seine internen Tabellen maximal benötigen. Das garantiert eine ökonomische Speicherbelegung und hilft, das Laufzeitverhalten zu optimieren.

Dynamische Tabellenerweiterungen

Modularisierung und Verteilung

In ABAP/4 können Unterroutinen definiert und aufgerufen werden. Auch der Aufruf von Unterroutinen in anderen Programmen ist möglich. Sowohl der Name der zu rufenden Routine als auch der Name des Programms, in dem sie steht, können beim Aufruf dynamisch bestimmt werden. Unterroutinen dürfen lokale Felder enthalten, die der Außenwelt nicht zugänglich sind. Parameter können wahlweise "by value" oder "by reference" an Unterroutinen übergeben werden.

ABAP/4-Unterroutinen

Funktionsbausteine

Neben den einfachen Unterroutinen dienen vor allem Funktionsbausteine als Mittel zur Modularisierung von ABAP/4-Programmen. Funktionsbausteine zeichnen sich durch die Möglichkeit zur Wiederverwendung und durch die Datenkapselung aus. Sie sind ein Basiselement für die Anwendungsintegration im System R/3.

Modularisierung von ABAP/4-Programmen

Im wesentlichen haben Funktionsbausteine die folgenden Eigenschaften:

- Funktionsbausteine besitzen eine klar abgegrenzte Datenschnittstelle.
- Die Datenschnittstellen der Funktionsbausteine sind aufwärtskompatibel erweiterbar.

- ❑ Funktionsbausteine können in einer eigenen Testumgebung inklusive Regressionstest validiert werden.
- ❑ Funktionsbausteine werden in einer Funktionsbibliothek komfortabel verwaltet.
- ❑ Es gibt ein Konzept zur Ausnahmebehandlung innerhalb von Funktionsbausteinen.
- ❑ Funktionsbausteine können über Systemgrenzen hinweg aufgerufen werden (Remote Function Call, RFC).

Export-, Import- und Tabellenparametern

In der Datenschnittstelle eines Funktionsbausteins werden seine Parameter aufgelistet, getrennt nach Export-, Import- und Tabellenparametern. Über Tabellenparameter können Referenzen auf komplette Tabellen übergeben werden. Für Import- und Export-Parameter kann in der Datenschnittstelle ein Referenzfeld aus dem ABAP/4 Dictionary spezifiziert, für Import-Parameter ein Vorgabewert festgelegt werden. Der Aufrufer entscheidet, aus welchen Feldern Import-Parameter mit Werten versorgt und in welchen Feldern Werte aus Export-Parametern zurückgeliefert werden sollen. Er kann Import-Parameter unbelegt lassen, für die Vorgabewerte existieren.

Spezielle Testumgebung

Über die Datenschnittstelle hinaus kann der Funktionsbaustein keinen Kontakt zu seinem Aufrufer unterhalten. Wegen dieser streng begrenzten Schnittstelle besteht die Möglichkeit, einen Baustein sinnvoll auf die Erfüllung seiner Funktion zu testen, ohne ihn in ein Rahmenprogramm einzubinden. In einer speziellen Testumgebung hat der Programmierer daher die Möglichkeit, Funktionsbausteine zu testen, indem er Import-Parameter vorgibt, den Baustein startet und anschließend Export-Parameter und Zeitbedarf abliest. Testdaten können vom System R/3 abgespeichert und später automatisch validiert werden.

Ausnahmebehandlung

Eine weitere Besonderheit von Funktionsbausteinen ist das Konzept der Ausnahmebehandlung. Eine Ausnahme, die im Baustein ausgelöst wird, kann den Programmabbruch oder eine Meldung an den Benutzer mit entsprechender Beeinflussung des Dialogfortgangs bewirken. Beides kann aber der Aufrufer des Bausteins verhindern, indem er explizit wünscht, selbst auf gewisse Ausnahmen reagieren zu dürfen.

Anwendungsbausteine

Funktionsbausteine werden in einer eigenen Bibliothek verwaltet, die mit komfortablen Suchfunktionen für Software-Entwickler ausgestattet ist. Eine reichhaltige Grundausstattung dieser Funktionsbibliothek ist im R/3-Standard bereits enthalten. Insbesondere sind viele Anwendungsbausteine für systemweit nutzbare Anwendungsfunktionen in Form von Funktionsbausteinen realisiert.

Remote Function Call RFC

Funktionsbausteine bilden die wesentliche Grundlage für die verteilte Verarbeitung von ABAP/4-Programmen. Der Aufruf erfolgt über Remote Function Call. Für das aufrufende Programm ist die Verteilung auf mehrere Systeme vollständig transparent. Der Zielrechner eines RFC kann über eine Tabelle unabhängig vom jeweiligen Programm festgelegt werden.

Funktionserweiterung über Customer Exits

Die Erweiterung von Standardfunktionen des Systems R/3 durch kundenindividuelle Ausprägungen vorgefertigter Customer Exits ist ein Verfahren, das direkte Eingriffe in die gegebene Logik verhindert.

Zielsetzung der Technologie ist, das Ablaufverhalten und die Funktionalität der Standardsoftware einfach und robust zu ändern, wenn einzelne Funktionen der Anwendungen für besondere Geschäftsprozesse durch Customizing-Funktionen nicht anpaßbar sind. Unkontrollierbare und unkoordinierte Eingriffe in die Standardsoftware werden vermieden. Dies ist ein erheblicher Beitrag zur Minimierung des Pflegeaufwands und demzufolge der Folgekosten von kundenspezifischen Anpassungen der Standardsoftware.

**Minimierung des
Pflegeaufwands**

Die wichtigsten Aspekte dieser Technologie sind:

- Alle potentiellen Kundenanforderungen, die nicht Bestandteil der Standardlösungen sind, werden als vordefinierte Schnittstellen in die Standardsoftware implementiert. Diese Schnittstellen sind nach Bedarf für das Einbinden kundenspezifischer Logik nutzbar.
- Die Technologie garantiert eine eindeutige Trennung zwischen SAP-Logik und Kunden-Logik.
- Der Anspung der Exits ist in der Standardsoftware implementiert, jedoch solange inaktiv, bis er explizit aktiviert wird.
- Customer Exits sind dem Kundensystem zugeordnet. Die SAP garantiert, daß sie bei Korrekturen und Releasewechsel nicht überschrieben werden.
- Aufwärtskompatibilität ist gewährleistet. Demzufolge bleibt der Anspung eines Exits aus der Standardsoftware sowie die Gültigkeit der Aufrufschnittstelle auch in zukünftigen Releaseständen erhalten.

Einige Beispiele für implementierte Exits: Function Exits (CALL CUSTOMER_FUNCTION), Menüerweiterungen, Dynproerweiterungen (CALL CUSTOMER-SUBSCREEN), Texterweiterungen zur Datenelement-Dokumentation und kundenseitige Textänderungen von Schlüsselworttexten und Kurztexten für Datenelemente.

Entwicklungswerkzeuge der ABAP/4 Development Workbench

Werkzeugintegration Die ABAP/4 Development Workbench enthält alle für die Erstellung und Pflege von ABAP/4-Programmen erforderlichen Werkzeuge. Sie gestatten es dem Entwickler, über intuitiv bedienbare Dialogschritte Entwicklungsobjekte zu erstellen und diese zu vollständigen Anwendungen zusammenzusetzen.

Um den Überblick über die Einzelteile einer Anwendung zu erleichtern, stellt die ABAP/4 Development Workbench die Entwicklungsobjekte als strukturierte Liste dar. Der Entwickler benutzt nur ein einziges Werkzeug, das diese Objektlisten verwaltet und allgemeine Funktionen wie Anzeigen, Anlegen, Kopieren, Testen und Verwendungsnachweis anbietet. Bei Auswahl eines Objektes und der gewünschten Funktion wird automatisch in das zur Bearbeitung geeignete Werkzeug verzweigt.

Performance-Analyse Neben dem Programmierer stehen innerhalb der ABAP/4 Development Workbench separate Werkzeuge für die Pflege von Dokumentation und Textelementen zu einem Programm sowie die ABAP/4-Funktionsbibliothek zur Verfügung. Die Fehlersuche in ABAP/4-Programmen wird durch einen komfortablen Debugger unterstützt. Außerdem kann das Input-/Output-Verhalten von Programmen anhand von Ablaufprotokollen überprüft werden. Verfahren zur detaillierten Performance-Analyse helfen bei der Optimierung des Laufzeitverhaltens.

Screen Painter

Der Screen Painter ist das Werkzeug in der Development Workbench, mit dem Dynpros angelegt, modifiziert, angezeigt und gelöscht werden. Entsprechend dem dreiteiligen Aufbau der Dynpros unterstützt der Screen Painter im wesentlichen drei Tätigkeitsfelder:

- das Anordnen von Feldbezeichnungen und Schablonen unter Verwendung des Fullscreen-Editors,
- das Festlegen der Feldeigenschaften für die im Layout eingetragenen Bildschirmfelder (z.B. grafische Darstellung, Darstellung mit führenden Nullen) und
- das Erfassen der Ablauflogik.

Außerdem können im Screen Painter allgemeine Angaben über das Dynpro erfaßt werden, wie z.B. die Sprache, in der es abgefaßt ist, die Nummer des Folgedynpros bei statisch definierten Bildfolgen und die Art der Darstellung (z.B. Dialogfenster, Selektionsbild). Hilfsmittel für die vereinfachte Erstellung fremdsprachiger Versionen zu einem Dynpro, welches in einer beliebigen Sprache bereits vorliegt, sind vorhanden.

Zur Ergänzung des Dialoges im primären Dialogfenster sind sekundäre (modale) Dialogfenster einsetzbar. Dieser Fenstertyp dient zur Vereinfachung komplexer Dialoge. Sekundäre Dialogfenster eröffnen beispielsweise ergänzende Arbeitsschritte oder erlauben Fehlerbehandlungen. Der Einsatz von parallelen (amodalen) Fenstern ist ebenfalls geplant. Eine beispielhafte Anwendung dieser Technik ist die Darstellung einer Liste in einem Fenster und parallel dazu die Anzeige und das Bearbeiten eines aus der Liste gewählten Dokumentes im anderen Fenster.

Dialogfenster

Zur Bildschirmgestaltung sind im Screen-Painter ein alphanumerischer und ein grafischer Editor integriert.

Grafischer Editor

Im grafischen Editor ordnet man die Ein- und Ausgabefelder und die zugehörigen Feldbezeichnungen auf dem Bildschirm an. Selbstverständlich können mit Hilfe des Grafik-Editors auch grafische Bedienelemente in die Bildschirmmasken eingebunden werden. Diese Elemente können vom Benutzer der erstellten Anwendung entweder mit Hilfe der Maus oder über die Tastatur angesprochen werden. Folgende Elemente stehen u.a. zur Verfügung:

Grafische Bedienelemente

- Pushbutton:** Ein Pushbutton stellt eine Drucktaste auf einem Dynpro dar. Beim Mausklick auf einen Pushbutton wird eine definierte Funktion der Anwendung ausgeführt.
- Radiobutton:** Radiobuttons sind Auswahlfelder, die die Selektion von genau einer Ausprägung aus einer Menge vorgegebener Alternativen ermöglicht.
- Checkbox:** Checkboxes sind Ankreuzfelder, die die Auswahl von mehreren Ausprägungen aus einer Menge vorgegebener Alternativen ermöglicht.
- Listbox:** Die Arbeit mit umfangreichen Listen wird mit Hilfe der Listboxen ergonomisch gestaltet.
- Rahmen:** Rahmen mit integriertem Überschriftsfeld fassen logisch zusammengehörige Felder optisch zusammen. Bei der Darstellung von Ein-/Ausgabefeldern variabler Größe (z.B. Listenausgaben) paßt sich der Rahmen automatisch an.

Parallel zum grafischen Editor kann sich der Entwickler eine beliebige Dictionary-Tabelle anzeigen lassen, um Felder aus dieser Tabelle in den Arbeitsbereich eines Fensters zu übernehmen. Die Übernahme mehrerer Tabellenfelder ist mit Cut&Paste möglich. In der parallel dargestellten Feldliste werden Name und alle Attribute (grafischer Typ, Format, Ein-/Ausgabefeld etc.) des aktiven Dynprofeldes angezeigt. Diese Anzeige wird bei einem Feldwechsel kontextsensitiv aktualisiert. Dadurch lassen sich in parallelen Fenstern alle Objekte anzeigen und verarbeiten, die zur Erstellung einer Bildschirmmaske notwendig sind. Änderungen von Feldbezeichnungen und Ein-/Ausgabedefinitionen, die im ABAP/4 Dictionary hinterlegt sind, führen zu einer automatischen Aktualisierung des Dynpros.

Cut & Paste

Menu Painter

Zentrale Arbeitsoberfläche Da in der Regel einheitliche Menüs über mehrere Bildschirmbilder hinweg verwendet werden, ist eine übergreifende Menüdefinition sinnvoll. Mit dem Menu Painter werden die vom Programm realisierten Funktionen bestimmten Menüs, Funktions- und Drucktasten zugeordnet. Auf einer zentralen Arbeitsoberfläche, die den Aufbau der Menüs und Drucktasten einer Anwendung darstellt, kann der Entwickler die entsprechenden Definitionen vornehmen.

Einmal erstellte Menüs können komplett kopiert werden, um eine rationelle Entwicklung zu gewährleisten. Die Menüfunktionen können aktiviert oder deaktiviert werden. Bei einer Deaktivierung erscheint zwar die Funktion farbschwach im Menü, kann aber nicht ausgewählt werden. Ein Hauptmenü (Aktionsmenü) kann aus bis zu drei Stufen und jede Stufe aus bis zu 15 Einträgen bestehen. Die Pflegesprache wird beim Anlegen eines Programms vergeben. Sie besitzt auch Gültigkeit für die Textelemente, den Screen Painter und den Menu Painter.

Ergonomische Oberflächengestaltung

Dem Entwickler stehen für Definitionen in den Menüs und für die Drucktasten Vorschläge gemäß SAP-Style-Guide zur Verfügung. Durch Verwendung dieser Vorschläge wird die einheitliche Bedienung aller mit der ABAP/4 Development Workbench erstellten Oberflächen unterstützt. Die Vorschläge können bei der Oberflächengestaltung übernommen und modifiziert oder einfach ausgeschaltet werden, um eigene Anforderungen zu realisieren. Zur Überprüfung der definierten Einträge ist es möglich, eine Prüfliste zu erzeugen, die dem Entwickler Fehler und Verstöße gegen die Standards der SAP aufzeigt.

Editor

Sonderfunktionen Der Editor für das Erfassen und Bearbeiten von ABAP/4-Programmen bietet neben den normalen Textoperationen wie Einfügen, Suchen, Ersetzen usw. mehrere Sonderfunktionen für die Programmentwicklung:

- systemweite Navigation und Anzeige von Entwicklungsobjekten (Tabellendefinitionen, Funktionsbausteine etc.).
- Verwendungsnachweis von Entwicklungsobjekten.
- Starten von ABAP/4-Reports.
- Auflösen von Verweisen auf Include-Programme.
- Anzeigen von Informationen über Syntaxregeln und erlaubte Schlüsselwörter.
- Prüfen der syntaktischen Korrektheit eines Programmtextes.

- Einfügen von vorgefertigten Programmteilen, etwa bei Schleifen, bedingte Anweisungen, Aufruf von Funktionsbausteinen mit Import- und Exportparametern und bei SELECT-Anweisungen.
- Strukturieren des Quellcodes mit dem Pretty Printer.
- Vergleichen zweier ABAP/4-Programme mit Hilfe des "Split Screen" - Editors.

Die ABAP/4 Development Workbench bietet komfortable Navigationsmöglichkeiten, welche die Programmentwicklung und Programmwartung drastisch vereinfachen. Ausgangspunkt der Navigation sind die Objektlisten. Durch Auswählen mit der Maus kann von jeder Verwendungsstelle eines Entwicklungsobjektes direkt zur Definitionsstelle verzweigt werden. Auch eine kaskadenartige Navigation über mehrere Definitionsebenen wird unterstützt. So ist es beispielsweise möglich,

Navigation

- innerhalb des Programm-Editors von einem Unterprogrammaufruf zur zugehörigen Unterprogrammdefinition zu springen
- aus einem ABAP/4-Programm über eine Tabellendeklaration zur Tabellendefinition im ABAP/4 Dictionary zu verzweigen
- von einem Tabellenfeld im ABAP/4 Dictionary auf das dazugehörige Datenelement und von dort auf die Domäne zu verzweigen
- ausgehend von der Ablaufsteuerung eines Dynpro direkt in den Programm-Editor zu den zugehörige ABAP/4-Modulen zu springen

Bei der Navigation werden alle Ausgangspositionen in einem Stapelspeicher abgelegt. Dadurch wird auch bei mehrstufiger Navigation sichergestellt, daß jede Ausgangsposition, gegebenenfalls in mehreren Schritten, wieder erreichbar ist.

Verwendungsnachweis

Zu jedem Zeitpunkt kann für alle Entwicklungsobjekte ein systemweit aktueller Verwendungsnachweis abgerufen werden. Beispielsweise alle Aufrufstellen eines Funktionsbausteins oder alle Programme, die eine bestimmte, im ABAP/4 Dictionary definierte Tabelle verwenden. Die generierten Listen können als Ausgangspunkt für weitergehende Navigationen benutzt werden.

Debugger

Mit dem Debugger der ABAP/4 Development Workbench kann sowohl nach Fehlern in ABAP/4-Programmen als auch in Dynpros gesucht werden. Mit Hilfe des "Remote Debugging" ist bei RFCs sogar die Fehlersuche über Systemgrenzen hinweg ermöglicht. Beim Debugging von Dialogprogrammen kann der Benutzer beliebig zwischen Dynpro- und Modulpool-Ebene hin und her wechseln.

Remote Debugging

Der Debugger kann den Programmlauf wahlweise nach jeder Anweisung oder aber an bestimmten Stellen des Programms unterbrechen. Außerdem besteht die Möglichkeit, nach Laufzeitfehlern oder aus der Prozeßüberwachung direkt in den Debugger einzusteigen.

Unterbrechungsstellen können im Programm fest vorgegeben sein; im Debugger können interaktiv weitere hinzugefügt und von Fall zu Fall verändert werden. Die zusätzlichen Unterbrechungsstellen kann der Benutzer statisch durch die Angabe einer Programmzeile oder dynamisch durch die Angabe einer Stop-Bedingung - z.B. "Der Wert der Variablen x ist kleiner (bzw. größer, gleich) y" - charakterisieren.

Nach jeder Unterbrechung gibt der Debugger Auskunft über den gegenwärtigen Stand der Verarbeitung. Man kann sich informieren über die Inhalte von Feldern, Arbeitsbereichen und internen Tabellen, den bisher erstellten Teil der Liste, die aktuelle Aufrufschachtelung u.a. Bevor man das Programm weiterlaufen läßt, sind die Inhalte von Feldern und internen Tabellen beliebig veränderbar.

Performance-Messung

Interaktive Analyse des Verbrauchs

Die Entwicklungswerkzeuge zur Performance-Messung ermitteln den genauen Verbrauch an Rechnerleistung von Modularisierungseinheiten wie Unterprogrammen, Funktionsbausteinen usw., sowie für Datenbankzugriffe und Operationen mit internen Tabellen in einem ABAP/4-Programm. Die gewonnenen Daten können als Hitlisten auf unterschiedlichen Verdichtungsebenen übersichtlich aufbereitet werden: Absoluter oder prozentualer Verbrauch, Sortierung der Verbraucher nach Verbrauchswerten, interaktive Aufschlüsselung des Gesamtverbrauchs einer Routine in den Verbrauch ihrer Unterroutinen usw.

Computer Aided Test Tool

Systematische Tests

Das CATT-Werkzeug (Abk.: Computer Aided Test Tool) unterstützt das Beschreiben, Zusammenfassen und Automatisieren von wiederholbaren betriebswirtschaftlichen Vorgängen für Testabläufe. Das gilt für einzelne Funktionen und für umfangreiche Abläufe. Eingabedialoge werden mit Datenübernahme über die Hintergrundverarbeitung automatisch simuliert. Der Test ist aber auch jederzeit manuell in Einzelschritten durchführbar. Tests werden vereinheitlicht und systematisiert, wobei sie beliebig oft wiederholbar sind. Der Testablauf und die dazu gehörenden Parameter-einstellungen werden durch das CATT-Werkzeug jederzeit protokolliert. CATT steht zur Zeit nur SAP-intern zur Verfügung, eine generelle Freigabe ist geplant.

Query und Reporting

Reports sind Programme, die einen Teil des Datenbankinhalts lesen und aufbereiten. Reports können direkt mit Hilfe von SELECT-Anweisungen auf die Datenbank zugreifen oder auch den komfortablen Weg über logische Datenbanken nutzen.

Logische Datenbanken sind spezielle ABAP/4-Programme, die einen hierarchisch strukturierten Ausschnitt des Datenbankinhalts für Report bereitstellen. Reports werten den Inhalt der logischen Datenbank unter dem Blickwinkel einer speziellen Fragestellung aus. Sie liefern ein Ergebnis in Form einer Reportliste, die der Benutzer am Bildschirm betrachten, interaktiv weiterverarbeiten und ausdrucken kann.

Reports werden auch verwendet, um abgeleitete Datenbestände, wie z.B. Strategische Indizes, periodisch zu aktualisieren. In Strategischen Indizes werden vornehmlich die Ergebnisse von Auswertungen abgespeichert, die einen sehr hohen Rechenaufwand haben, deren Werte aber nicht unbedingt tagesaktuell sein müssen, wie z.B. die Liste der 100 umsatzstärksten Kunden.

Logische Datenbanken

Eine logische Datenbank ist ein ABAP/4-Programm, das Reports mit einer hierarchisch organisierten Menge von Tabellenzeilen versorgen soll; üblicherweise verwendet man den Begriff "logische Datenbank" sowohl für das Programm selbst als auch für die von ihm gelieferte Datenmenge. Jede logische Datenbank vereinigt in sich die Inhalte gewisser Tabellen. Die Zeilen dieser Tabellen kann sie, eine nach der anderen, an einen Report liefern. Die Reihenfolge, in der die einzelnen Zeilen abgeliefert werden, wird normalerweise bestimmt von hierarchischen Fremdschlüsselbeziehungen zwischen den Tabellen.

Ein Report arbeitet mit maximal einer logischen Datenbank. Eine logische Datenbank kann in der Regel von mehreren Reports gemeinschaftlich verwendet werden.

Nach jedem Datentransport erhält der Report Gelegenheit, die transportierte Zeile zu verarbeiten, indem - abhängig davon, aus welcher Tabelle die Zeile stammt - ein spezieller Programmteil aktiviert wird. Man nennt diesen Vorgang, der den wesentlichen Teil eines jeden Reportlaufs ausmacht, das Durchlaufen der logischen Datenbank.

ABAP/4 enthält Sprachmittel, mit denen das Durchlaufen der logischen Datenbank von einem Report aus beeinflusst werden kann. Beispielsweise kann verlangt werden, daß die von der zuletzt gelieferten Zeile hierarchisch abhängigen Daten nicht an den Report gegeben werden. Außerdem kann jederzeit dynamisch überprüft werden, ob die gegenwärtig bearbeiteten Daten den vom Benutzer gemachten Selektionseingaben genügen.

Das Konzept, Programmteile für Datenbankzugriffe aus ABAP/4-Reports zu separieren und sie statt dessen als logische Datenbanken zu implementieren,

Logische Datenbanken

Strategische Indizes

Fremdschlüsselbeziehungen

ABAP/4-Systemfunktionen

Gleichartige Zugriffspfade

hat erhebliche Vorteile. Weil jede logische Datenbank mehrere Reports bedient, müssen gleichartige Zugriffspfade nur einmal codiert werden. Dies betrifft nicht nur die Anweisungen für den Datenzugriff selbst, sondern auch das Coding für Berechtigungsprüfungen sowie für das Navigieren in Datenbeständen anhand von Fremdschlüsseln.

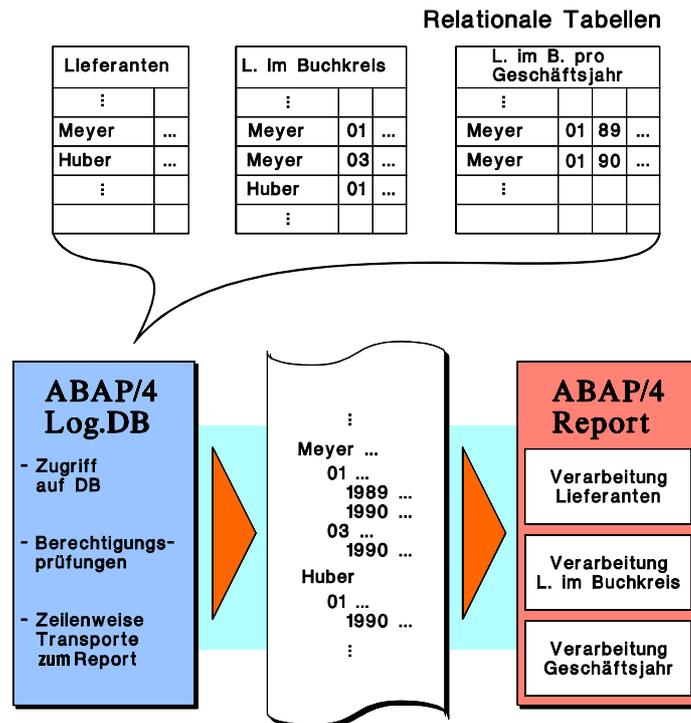


Abb. 8-10: Steuerung eines Reports durch seine logische Datenbank

Das Starten von Reports

ABAP/4-Reports werden aus einem speziellen Menü heraus gestartet. Im allgemeinen wird der Benutzer nach Aufruf eines Reports zunächst aufgefordert, durch Angabe von Selektionen die Menge der Daten, die der Report berücksichtigen soll, in verschiedener Hinsicht einzugrenzen. Hier-zu dient ein eigenes Dynpro, das mit jedem Report verbunden ist, das sogenannte Selektionsbild.

Selektionen können z.B. einfache Parameter sein, die aus einem einzelnen Wert bestehen. Die Selektionsfelder erlauben aber auch komplexere Eingaben, in denen Intervalle, Vergleiche, gesuchte Texte, mehrere alternative Bedingungen usw. vorkommen können. Der Benutzer hat außerdem die Möglichkeit, umfangreiche oder häufig vorkommende Selektionseingaben zu speichern und sie dann mehrfach zum Aufrufen des Reports zu verwenden.

Unter den Selektionen eines Reports sind in der Regel einige, mit denen die Menge der Tabellenzeilen, die seine logische Datenbank ihm übergibt, weiter eingeschränkt werden kann. Die hier eingegebenen Intervalle heißen Datenbankabgrenzungen. Zu jeder logischen Datenbank gehört ein fester Satz von Selektionen für Datenbankabgrenzungen. Aus diesem festen Satz und den speziellen Selektionen und Parametern, die im Report selbst definiert sind, wird das Selektionsbild eines Reports automatisch generiert.

Selektionsbild

Sowohl im Selektionsbild eines Reports als auch später während der Online-Anzeige seiner Liste kann der Benutzer per Funktionstaste auf ein Menü von ABAP/4-Systemfunktionen verzweigen. In diesem Menü kann er z.B. anordnen, daß die Reportliste ausgedruckt wird oder daß der Report im Debugging-Modus weiterlaufen soll.

Reports können auch aus anderen ABAP/4-Programmen heraus aufgerufen werden. Dies ist auch die Grundlage für das interaktive Reporting.

Listanzeige und interaktives Reporting

Der ABAP/4-Report schreibt die von ihm aufbereiteten Daten in eine Reportliste. Das Anzeigen der Reportliste am Bildschirm ist ein Service, den ein spezieller List-Prozessor automatisch für das Reportprogramm leistet. Er füllt dazu ein sogenanntes List-Dynpro mit einem Teil der Liste und gliedert es in den Dialogablauf ein. Der Benutzer variiert den angezeigten Teil der Liste interaktiv mit Hilfe einer Blätterleiste oder durch Kommando-dos in der Befehlszeile.

List-Prozessor

Der Report kann einzelne Felder in der Liste optisch hervorheben. Außerdem kann er eingabebereite Felder in die Liste einfügen und dadurch z.B. dem Benutzer erlauben, Informationen in der Liste zu ergänzen, bevor er sie ausdrucken läßt.

Ferner sind im Report weitere Verarbeitungen definierbar, die der Benutzer während der Anzeige der Liste durch Funktionstasten oder Eingaben in der Kommandozeile anstoßen kann. Man spricht dann von interaktivem Reporting.

Bei den weiteren Verarbeitungen, die im interaktiven Reporting zum Tragen kommen, kann es sich z.B. um Aufrufe eines anderen Reports oder einer Transaktion handeln. Durch den wechselseitigen Aufruf von Reports erhält der Benutzer die Möglichkeit, aus dem Betrachten einer Reportliste in einen anderen Report zu verzweigen und dabei Daten aus der Liste als Parameter für den neuen Report mitzunehmen.

Interaktives Reporting

Üblich ist im interaktiven Reporting auch, daß ein einziger Report mehrere Listen dynamisch erzeugt. Der Report kann z.B. zunächst eine komprimierte Version der aufbereiteten Daten in einer Grundliste anbieten. Der Benutzer wählt einzelne Datensätze in der Grundliste aus, und derselbe Report liefert ihm in einer Folgeleiste detailliertere Informationen zu den ausgewählten Daten. Dieser Vorgang kann sich schrittweise wiederholen.

Window-Techniken im Reportprogramm

Ein ABAP/4-Report kann bis zu 10 aufeinander aufbauende Listen gleichzeitig verwalten, die der Benutzer interaktiv erzeugt und wieder aufgibt. Durch die Verwendung von Window-Techniken im Reportprogramm können die Listen am Bildschirm so angeordnet werden, daß jede Folgeliste die vorherigen nur teilweise überdeckt.

Listobjekte für Microsoft Excel

Die Einbindung eines speziellen Funktionsbausteins in ABAP/4-Programme erlaubt es R/3-Anwendungen, Listobjekte zu bilden und nach Microsoft Excel zu exportieren (siehe Kap. 4: Integration von Desktop-Anwendungen). Der Funktionsbaustein sorgt nicht nur dafür, daß ein Listobjekt zum Präsentationsrechner des Benutzers übertragen wird, sondern startet auf Wunsch auch automatisch Excel. Dem Listobjekt können in der Excel-Anwendung Informationen hinzugefügt werden, ohne die Konsistenz des Objektes zu zerstören. Dadurch ist es möglich, geänderte Listobjekte wieder in das System R/3 zu importieren.

ABAP/4-Query

Die Anwendung ABAP/4-Query dient zur Erzeugung von ABAP/4-Reports, die im R/3-Standard nicht enthalten sind. Sie ist speziell für solche Anwender gedacht ist, die keine Programmierkenntnisse in ABAP/4 besitzen.

Automatisch generierte Reports

ABAP/4-Query erlaubt es, einfache Reports zu definieren, ohne zu programmieren. Der Benutzer muß lediglich die Anforderungen an die Liste, die er benötigt, formulieren. Ein Report, der eine solche Liste liefern kann, wird dann automatisch generiert. Das Ergebnis der Generierung, eine sogenannte Query, ist ein normaler ABAP/4-Report, der insbesondere auch mit den Mitteln der ABAP/4 Development Workbench modifiziert und erweitert werden kann.

Der Funktionsumfang, den eine Query bieten kann, entspricht natürlich nicht dem eines ABAP/4-Reports im allgemeinen. Der Bereich des klassischen Reporting, wo Ranglisten und Statistiken zu erstellen sind, kann aber vollständig mit Queries abgedeckt werden.

Beim Anlegen einer Query muß sich der Benutzer als erstes für ein Sachgebiet entscheiden. Das Sachgebiet bestimmt, auf welche Tabellen bzw. auf welche Felder dieser Tabellen die Query Bezug nehmen kann. Die Sachgebiete von ABAP/4-Query sind in der Regel Untermengen von logischen Datenbanken.

Die Felder, die zum Sachgebiet gehören, werden dem Benutzer als Auswahl-liste angeboten. Durch Ankreuzen derjenigen Felder, die tatsächlich für die Liste relevant sein sollen, schränkt er diese Vorauswahl weiter ein. Über das Sachgebiet kann er zusätzlich weitere Felder einführen, die ebenfalls in der Liste auftauchen sollen. In einem solchen Fall muß er durch einfache ABAP/4-Statements angeben, wie die Inhalte dieser Felder aus den Inhalten der Sachgebietsfelder zu errechnen sind.

Auswahllisten

Bevor der von ABAP/4-Query erzeugte Report gestartet werden kann, muß der Benutzer noch das Listen-Layout spezifizieren. Speziell muß festgelegt werden, in welcher Position und in welcher Reihenfolge die ausgewählten Felder ausgegeben werden sollen und nach welchen Feldinhalten die Tabellenzeilen der Liste sortiert sein sollen. Als Ergänzung oder auch als Alternative zur Auflistung der einzelnen Zeilen kann die Query Zwischen- oder Gesamtsummen von Feldinhalten ausgeben. Auch nach solchen Zwischensummen kann sortiert werden, so daß auf einfache Weise Rang-listen entstehen. Zu den Zwischensummen können die Anzahl summierter Werte, der Durchschnittswert und der prozentuale Anteil der Zwischen- an der Gesamtsumme automatisch errechnet und in Form von Statistiken der Liste angefügt werden. Die Liste einer einzigen Query kann bis zu 99 solcher Statistiken enthalten.

Listen-Layout

Alle diese Forderungen an die Liste formuliert der Benutzer allein durch die Auswahl von Ausgabeoptionen oder Positionsangaben zu den Feldern, die er zuvor als relevant für die Liste ausgewählt hat. Sogar Texte, die als Erläuterungen auf der Liste auftauchen sollen, kann ABAP/4-Query - anhand von Informationen aus dem ABAP/4 Dictionary - automatisch bestimmen. Der Benutzer hat aber natürlich die Möglichkeit, statt dessen eigene Ausgabetexte für die Liste in die Query einzubringen.

Entwicklungsorganisation

- ABAP/4 Development Organizer** Die Entwicklung und Wartung großer Softwaresysteme erfordert leistungsfähige Organisationswerkzeuge. Diese müssen sowohl den Anforderungen einer zentralen als auch einer dezentralen Entwicklung Rechnung tragen. Die Arbeit einzelner Entwickler muß genauso unterstützt werden wie die Teamarbeit von Projektgruppen. Dabei darf es keine Rolle spielen, ob die Entwickler einer Projektgruppe auf einem oder verschiedenen Rechnern arbeiten. Der ABAP/4 Development Organizer als integrierter Bestandteil der Workbench erfüllt diese Anforderungen sowohl für kleine als auch für größte Entwicklungsvorhaben.
- Change Management** Entwicklungsprojekte werden in verschiedene Aufträge aufgeteilt, an denen einzelne Entwickler bzw. Projektgruppen mitarbeiten. In den Aufträgen werden alle geänderten Entwicklungsobjekte automatisch erfaßt. Sie sind dann für diese Aufträge zur ausschließlichen Verwendung reserviert. Die Entwicklung oder Pflege der Entwicklungsobjekte ist bis zur Freigabe des Auftrags für andere Entwickler gesperrt; die entsprechenden Objekte können nur angezeigt werden. Dadurch können die Aufträge unabhängig voneinander freigegeben werden.
- Entwicklungsclassen** Jedem Entwicklungsobjekt wird der verantwortliche Entwickler und eine Entwicklungsclass zugeordnet, die das Sachgebiet des Objekts kennzeichnet. Dadurch lassen sich sofort Ansprechpartner zu jedem Objekt finden. Die Entwicklungsclass wird auch als Strukturierungs- und Einstiegshilfe in der gesamten ABAP/4 Development Workbench benutzt.
- Versionsverwaltung** Mit der Freigabe eines Auftrags wird automatisch eine Versionshaltung aller Objekte vorgenommen, die sowohl Vergleiche als auch einen Zugriff auf frühere Versionen erlaubt. Dadurch können die freigegebenen Zustände vor und nach einem Auftrag oder Projekt dokumentiert und sogar wiederhergestellt werden.
- Revisionssicherheit** Zu jedem Auftrag wird durch die Entwickler eine gegliederte Dokumentation über die Projektziele, den Status und Besonderheiten angelegt. Weiterhin wird automatisch ein Protokoll über alle geänderten Entwicklungsobjekte erstellt. Die Protokollierung zusammen mit Dokumentation und Versionierung garantiert einen lückenlosen Änderungsnachweis für die Revision in Ein- und Mehrsystem-Konfigurationen.

Entwicklungsobjekte werden normalerweise nicht im Produktivsystem, sondern je nach Umfang in einem oder mehreren Entwicklungs- oder Testsystemen entwickelt und gewartet. Um die Konsistenz der Objekte zu gewährleisten, besitzt jedes Objekt einen definierten Originalstandort und Freigabebeweg im Systemverbund, der jedoch jederzeit verlagert werden kann. Änderungen sind prinzipiell nur am Originalobjekt möglich, um ungewollte Parallelentwicklung zu vermeiden. Beim Einsatz mehrerer Entwicklungssysteme können die Entwickler ihre Objekte im lokalen Entwicklungssystem nach einem Check-Out isoliert entwickeln und testen. Um einen Integrationstest von Objekten aus mehreren lokalen Systemen durchführen zu können, werden die einzelnen Objekte in ein gemeinsames Integrationssystem transportiert und dort getestet.

Eng an den ABAP/4 Development Organizer gekoppelt ist das Transportsystem, das den Transport von Entwicklungsobjekten und Customizing-Einstellungen durchführt. Transporte können anhand der Aufträge durchgeführt werden, d.h. die Menge der zu transportierenden Objekte nach einer Freigabe ergibt sich automatisch.

Transportsystem

Für jeden Transport wird automatisch ein Transportprotokoll angelegt. Sollte sich z.B. ein Produktivsystem nach einem Import aus dem Testsystem fehlerhaft verhalten, so kann sofort festgestellt werden, welche Objekte transportiert wurden, wer der Auftraggeber war und aus welchem Grund der Transport durchgeführt wurde. Um die Konsistenz eines Transports schon im Vorfeld prüfen zu können, besteht die Möglichkeit, den Transport in das Zielsystem zu simulieren. Der Benutzer kann dadurch schon frühzeitig mögliche Auswirkungen seines Transportauftrags auf das Zielsystem abschätzen.

Durch die enge Koppelung des Transportsystems an den ABAP/4 Development Organizer werden alle eingerichteten Transportwege automatisch benutzt, außerdem wird unbeabsichtigtes Überschreiben von Objekten verhindert.

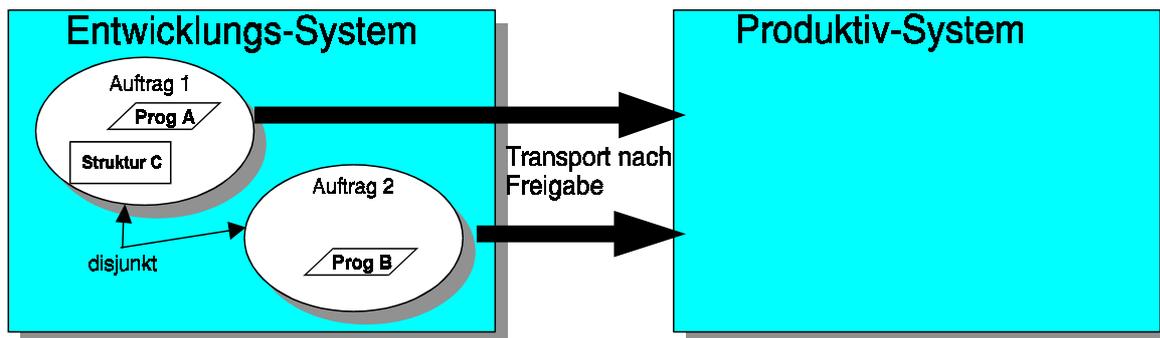


Abb. 8-11: Typische Konfiguration einer Entwicklungsumgebung mit dem ABAP/4 Development Organizer

Schnelle Releasewechsel

Der ABAP/4 Development Organizer gewährleistet kurze Stillstandszeiten von Kundeninstallationen beim Releasewechsel. Kundenspezifische, vom Standard abweichende Funktionen und Datenstrukturen können problemlos übernommen werden, da sie vollständig erfasst sind und Versionen sowohl des SAP-Standard als auch des Kunden existieren. Der Abgleich kann dann werkzeuggestützt durchgeführt werden.

Ergänzend zum ABAP/4 Development Organizer und dem Transportsystem steht ein Informationssystem für das Suchen, Anzeigen, Bearbeiten und Analysieren von Aufträgen und Transporten zur Verfügung.

Informationssystem

Unter anderem werden folgende Anzeigefunktionen angeboten:

- Aufträge und Transporte mit der jeweiligen Entwicklungs- oder Transporthistorie, sortiert und selektiert nach Entwicklern oder Sachgebieten.
- Offene Aufträge und Transporte, die noch nicht freigegeben bzw. transportiert wurden.
- Aufträge und Transporte, die bestimmte Entwicklungsobjekte betreffen.
- Entwicklungsobjekte, selektiert und sortiert nach Sachgebieten, verantwortlichen Entwicklern, Änderungsdatum und weiteren Kriterien.

ABAP/4 Repository Information System

Im ABAP/4 Repository sind alle Entwicklungsobjekte der ABAP/4 Development Workbench abgelegt. Zum ABAP/4 Repository gibt es eine Sammlung von Reports, mit denen die im ABAP/4 Repository enthaltenen Informationen besonders komfortabel ausgewertet werden können. Dieses Repository Information System umfaßt unter anderem Reports:

- zur Auflistung von Programmen, Tabellen, Feldern, Datenelementen und Domänen nach bestimmten Kriterien, wie z.B. dem Datum der letzten Änderung oder vorgegebenen Objekteigenschaften.
- zum Nachweis der Verwendung von Tabellen und ihren Feldern in Dynpros und ABAP/4-Programmen.
- zur Darstellung von Fremdschlüsselbeziehungen.

Navigation zwischen Reports

Wird das Infosystem im Dialog eingesetzt, so ist fast jede Navigation zwischen den Reports möglich, die semantisch sinnvoll ist, wie z.B. der Übergang:

- aus einer Liste von Datenelementen zu der Liste aller Tabellenfelder, die sich auf diese Datenelemente beziehen,
- aus der Liste von Tabellenfeldern auf die Liste ihrer Verwendungen in Programmen,
- aus dieser Liste wiederum auf eine Liste der beteiligten Programme etc.

Vor jedem Übergang können gewisse Einträge in der angezeigten Liste angekreuzt und als Selektionseingaben für den nächsten Report übernommen werden. Außerdem kann der Benutzer aus dem Betrachten einer Liste direkt auf die Pflege der angezeigten Objekte verzweigen.