Aufbau eines Data Warehouse auf Grundlage betriebswirtschaftlicher Standardsoftware

Carsten Bange

Lehrstuhl für Betriebswirtschaftslehre und Wirtschaftsinformatik Prof. Dr. R. Thome
Universität Würzburg
Telefon: 0931 / 312451
eMail: bange@wiinf.uni-wuerzburg.de

Abstract

Die Integration verschiedener Datenbestände eines Unternehmens in einem homogenen und dauerhaft angelegten Data Warehouse ist inzwischen akzeptierte und weit verbreitete Vorgehensweise zum Aufbau entscheidungsorientierter Informationssysteme (EIS). Das Data Warehouse ermöglicht die umfassende Versorgung von Informationsempfängern mit Daten, die mehrdimensional und hierarchisch aufbereitet der Sichtweise von Anwendern in Unternehmen entsprechen.

Eine der wichtigsten Datenquellen für EIS sind die betrieblichen Transaktionssysteme, die Grundfunktionen wie Warenwirtschaft oder Rechnungswesen abwickeln. In diesem Bereich kommen häufig betriebswirtschaftliche Standardsoftwarepakete von Herstellern wie SAP AG, Baan oder Oracle zum Einsatz. Aufgrund ihres spezifischen Aufbaus bereitet die Anbindung dieser Transaktionssysteme an ein Data Warehouse jedoch oftmals besondere Probleme.

Dieser Beitrag zeigt die Herausforderungen des Extraktions-, Transformations- und Ladeprozesses von Daten eines SAP R/3-Systems in ein Data Warehouse auf und beschreibt eine mit SAS Institute Software realisierte Lösung. Im Kern der Anwendung stehen hierbei die Komponenten SAS/ACCESS Interface to R/3 für die Anbindung des SAP Systems und der SAS/Warehouse Administrator, der Aufbau und Management eines Data Warehouse zentral steuert.

Data Warehouse Systeme zur Entscheidungsunterstützung

Die betriebliche Leistungserstellung in Form von Produktion, Handel oder Dienstleistungen wird von einem weitgehend automatisiert abgewickelten Informationsfluss unterstützt und ermöglicht. Moderne Informationstechnik sorgt für eine integrierte und effiziente Erfassung, Speicherung, Verarbeitung und Verteilung von Daten im Unternehmen. Die Nutzung dieser Daten zur Unterstützung von Entscheidungen rückt seit ca. 35 Jahren immer wieder unter wechselnden Schlagworten in den Mittelpunkt der betrieblichen Informationsverarbeitung (Abb. 1).

Während frühere Konzepte wegen Unzulänglichkeit der Systeme oder übertriebener Versprechungen der Anbieter häufig scheiterten, hat sich in den 90er Jahren mit dem Data Warehouse-Ansatz eine erfolgreiche systematische und technologische Basis entwickelt.

Mit steigender Leistungsfähigkeit der Systeme wird auch den Anforderungen der Anwender zunehmend Rechnung getragen, was vor allem zu einer Akzeptanzsteigerung geführt hat, die Grundlage für eine weite Verbreitung von entscheidungsunterstützenden Systemen (EIS) ist. Die Hauptaufgabe eines EIS liegt in der Bereitstellung von Daten für Entscheidungen in der „richtigen“, also angemessenen und vom Anwender gewünschten Form. Das Problem bei der Einführung eines kompletten Informationssystems liegt hier sehr oft auf Seiten der Informationsbasis, in der alle momentan und zukünftig (vermutlich) relevanten Daten erfasst werden müssen. Nicht nur die unternehmensinternen, meist heterogenen Datenbestände (von Großrechnern über Netzwerke bis zu kleinen PC-Insellösungen) bereiten Probleme, die Grundlagen für spätere Auswertungen zu erstellen. Auch externe Datenbestände in den verschiedensten Formaten und Qualitäten (z. B. Markt- und Wettbewerbsdaten) erschweren diese Aufgabe. Das Ziel ist der Zugriff eines jeden Anwenders auf die Gesamtheit der zu

Verfügung stehenden Daten [ScBa99, S. 14]. Genau hier setzt das 1992 vorgestellte Konzept des Data Warehouse an, dessen geistiger Vater William H. INMON ist [Inmo92].

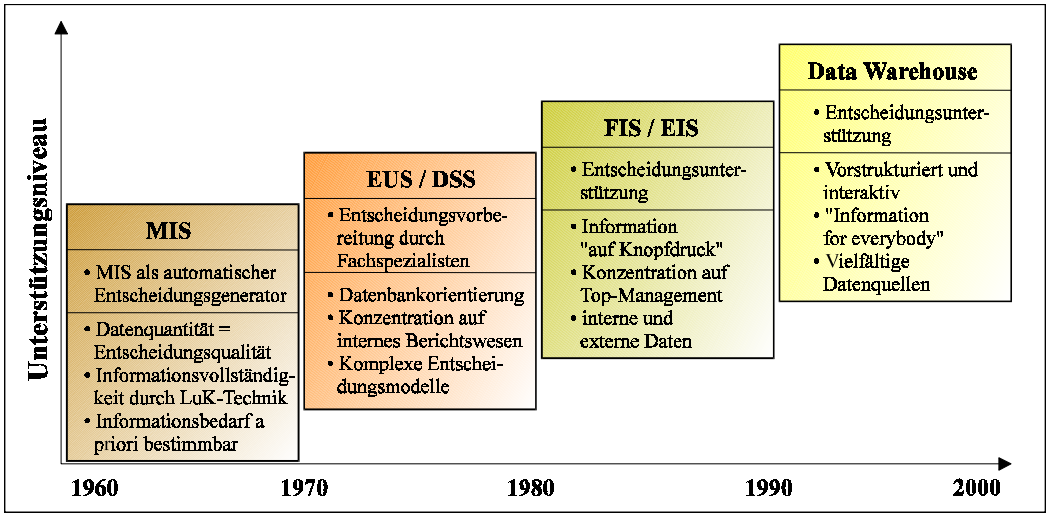

Abb. 1: Vom MIS zum Data Warehouse [BULL95, S. 19].

Data Warehouse Konzept

Inmon definierte als erster den Begriff des Data Warehouse als themenorientierte, integrierte, zeitbezogene und dauerhafte Sammlung von Informationen zur Entscheidungsunterstützung des Managements [Inmo92, S. 25]. Inmon stellt klar, dass der individuelle Informationsbedarf der Entscheidungsträger eines Unternehmens nicht durch anwendungsbezogen gesammelte Daten der operativen Systeme gedeckt werden kann, sondern durch ein Informationssystem, das verschiedene Anwendungen und Datenbestände einbezieht, die Zeit als bewertbare Bezugsgröße enthält und auch über einen längeren Zeitraum verfügbar ist. Diese eng eingegrenzte Sicht auf den Aufbau eines Datenlagers wurde mittlerweile von vielen Autoren um zusätzliche Bestandteile und Funktionen ergänzt. Als Data Warehouse System kann im weiteren Sinne neben der eigentlichen Datensammlung inklusive Verwaltung (engere Definition) die Anbindung, Extraktion und Transformation der Daten aus operativen und externen Datenbeständen sowie die Bereitstellung von Analyse- und Präsentationsmöglichkeiten mit entsprechenden Werkzeugen (Business Intelligence Tools) verstanden werden (Abb. 2) [Hans95, S. 8; Kell94, S. 55; Schu94, S. 209].

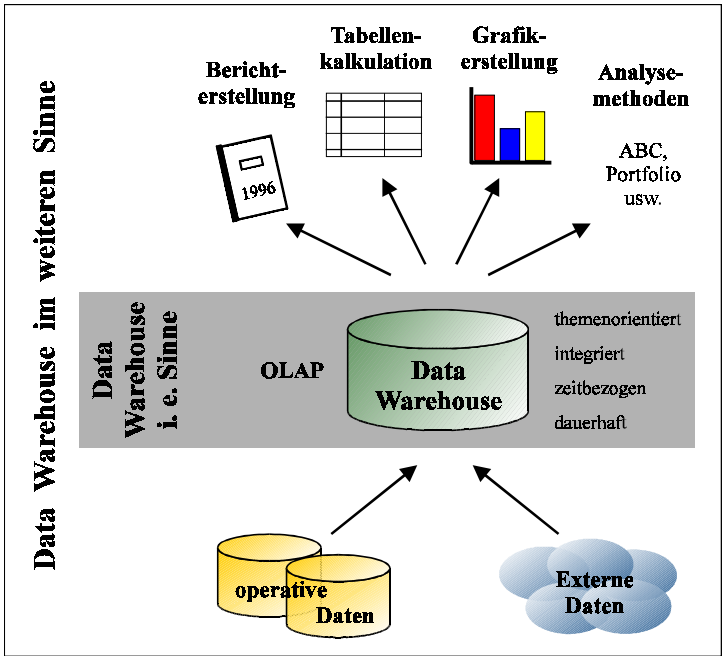

Abb. 2: Abgrenzungen des Data Warehouse-Begriffs [ScBa99, S. 16]

Die Verbindung der bereits vorhandenen Datenbestände in einer Form, dass aus ihnen heraus neue Auswertungen und Erkenntnisse gezogen werden können, erfolgt in einer eigenen Data Warehouse-Datenbank (DWDB). Die unterschiedlichen operativen Datenquellen werden extrahiert und ggf. transformiert, aggregiert und dann in der DWDB zusammengefügt. Die DWDB enthält folglich nur redundante Daten, die streng von den Daten der operativen Systeme getrennt sein sollten. Wichtig ist hierbei die Erstellung eines globalen Datenmodells, das auf der Analyse aller Unternehmensstrukturen basiert. Tabelle 1 zeigt die unterschiedlichen Merkmale der verschiedenen Datenbasen auf.

Tabelle 1: Unterschiedliche Eigenschaften der Datenbasis

Merkmale der Datenbasis	**Operative Informationssysteme**	**Data Warehouse**
Datenstruktur ist ...	anwendungsbezogen	themenbezogen
Dauer der Datenhaltung ist ...	ca. 90 Tage	mehrere Jahre
Granularität der Daten ist ...	detailliert	verdichtet & detailliert
Aktualisierung erfolgt ...	zeitnah	periodisch
Daten nach Änderung ...	i. d. R. überschrieben	fortgeschrieben

Eng mit dem Data Warehouse Konzept ist der Begriff des On-Line Analytical Processing (OLAP) verknüpft. E. F. CODD, der Erfinder des relationalen Datenbankmodells, prägte den Begriff 1993 in einem Aufsatz, in dem er die unterschiedlichen Anforderungen der Abarbeitung analytischer Anfragen (OLAP) der Abarbeitung von Transaktionen (On-Line Transaction Processing - OLTP) gegenüber stellte [Codd93]. OLAP beinhaltet im wesentlichen die konzeptionelle Basis für Lösungen zur Unterstützung einer dynamischen Datenanalyse in Unternehmen. Das Grundprinzip von OLAP basiert auf der Betrachtung von Daten aus verschiedenen Blickwinkeln (Dimensionen), die eine schnelle und flexible Analyse ermöglichen, so dass der Umgang mit großen Datenmengen vereinfacht wird. Auswirkungen ergeben sich bei der Wahl der eingesetzten Data Warehouse Datenbank und erforderlichen Maßnahmen zur Sicherstellung einer hohen Antwortgeschwindigkeit auch bei komplexen Abfragen. Für eine detailliertere Betrachtung von OLAP-Konzepten sei auf die einschlägige Literatur wie beispielsweise [Oehl00] verwiesen.

Aufbau eines Data Warehouse

Im Kern eines Data Warehouse Systems steht die Data Warehouse Datenbank. Zur Bereitstellung von OLAP-Funktionen in Berichts- und Analysewerkzeugen werden üblicherweise als Data Warehouse-Datenbank entweder multidimensionale Datenbankmanagementsysteme (MDBMS) oder relationale Datenbankmanagementsysteme (RDBMS) mit modifizierten Datenmodellen und Zusatzkomponenten zur Abfragebeschleunigung eingesetzt. MDBMS weisen dabei Vorteile hinsichtlich der Abfragegeschwindigkeit auf, RDBMS hinsichtlich des verwaltbaren Datenvolumens und der Administrierbarkeit des Systems z.B. bezüglich Sicherheitsaspekten.
Als zentrales Datenlager wird das Data Warehouse aus unternehmensinternen und -externen Datenquellen befüllt. Dieser Prozess und spezialisierte Werkzeuge für seine Unterstützung werden nach den Hauptaufgaben Extraktion, Transformation und Laden in der Regel als ETL-Prozess bzw. ETL-Werkzeuge bezeichnet. Die Komplexität hierbei liegt vor allem in der zentralen Idee des Data Warehouse, eine homogene und konsistente Datenbasis zur Entscheidungsunterstützung aus zahlreichen internen und externen Quellen aufzubauen. Die erwünschte Konsistenz und Homogenität liegt aber in den operativen Datenbanken nicht überall in gleicher Güte und vor allem nicht anwendungsübergreifend vor, sondern muss im Zuge des Data Warehouse-Aufbaus erst geschaffen werden [ScBa99, S.28-33]. In den Quellsystemen verstecken sich dabei syntaktische und semantische Fehler, die im Prozess der Extraktion und Transformation der Daten aus den Vorsystemen korrigiert werden müssen. Die Datentransformation als Kernprozess lässt sich grundsätzlich in die folgenden vier Schritte untergliedern (vgl. Abb. 3) [KeFi98, S. 63f]:

- **Filterung:** Im Rahmen der Filterung erfolgt zunächst die Anbindung der unterschiedlichen internen und externen Datenquellen über entsprechende Schnittstellen wie z. B. ODBC. Im Zuge der Extraktion erfolgt die Selektion der Basisdaten in eine sogenannte Staging Area. Die temporäre Zwischenspeicherung ist notwendig, um die Rohdaten von systematischen Fehlern zu bereinigen, die verschiedenen Datentypen zu harmonisieren und ggf. zu verdichten bzw. anzureichern, ehe sie ins Data Warehouse gelangen. Die Behebung systematischer Fehler, z. B. fehlende oder nicht-interpretierbare Steuerzeichen, Umwandlung unterschiedlicher Zeichensätze etc. erfolgt dabei schon im Zuge der Extraktion mit Hilfe von Mapping-Tabellen. Dazu werden in die Extraktionsprozeduren Regeln zur Fehlerbereinigung integriert. Werden während der Extraktion Anomalien festgestellt, die nicht mit Hilfe einer bekannten Regel zu lösen sind, werden diese in einer Logdatei dokumentiert und müssen dann manuell bearbeitet werden. In aller Regel nehmen diese Fehler im Laufe der Aktualisierungszyklen jedoch stark ab, da die Extraktionsergebnisse auch zu Qualitätsverbesserungsmaßnahmen in den operativen Informationssystemen führen.

Nach der Filterung erfolgt die entscheidungsorientierte Transformation der selektierten Rohdaten, bestehend aus den Teilprozessen Harmonisierung, Verdichtung und Anreicherung. An dieser Stelle erfolgen Schritte zur homogenen Darstellung von Zeit- und Währungsdaten, Beseitigung von Attributs- und Schlüsseldisharmonien, Aggregation und Berechnung von Kennzahlen sowie die Konsolidierung verschiedener Teilmengen des Datenbestandes.

- **Harmonisierung**: Der Schritt der Harmonisierung umfasst die themenbezogene Gruppierung der Daten, z.B. nach Kunden, Produkten oder Organisationseinheiten. Hierzu gehört neben der Zusammenfassung der Daten aus den unterschiedlichen Datenquellen auch die einheitliche Kodierung von Attributen und die Abstimmung der Schlüsselbeziehungen.

- **Verdichtung**: Liegen die Daten in bereinigter und konsistenter Form vor, werden sie im Anschluss daran verdichtet. Da an ein Data Warehouse gestellte Abfragen vielfach aggregierter Natur sind, bietet es sich aus Performance-Überlegungen an, wahrscheinliche Aggregationen bereits im voraus zu berechnen und somit zeitintensive Rechenvorgänge während der Abfrage zu vermeiden. Typische Aggregate sind beispielsweise entlang der Zeitdimension (Monat, Quartal, Jahr) zu finden.
- **Anreicherung**: Im letzten Schritt können die aufbereiteten Daten noch durch betriebswirtschaftliche Kennzahlen angereichert werden. Hierzu gehören beispielsweise Kenngrößen wie Plan/Ist-Abweichungen oder der Deckungsbeitrag, die sich aus den vorhandenen Daten berechnen lassen. Bei der Entscheidung über die Vorausberechnung und Verdichtung dieser Daten spielen wiederum Performance-Überlegungen eine wichtige Rolle.

Nach der technisch ausgerichteten Extraktion steht bei der Transformation primär der betriebswirtschaftlich logische Aspekt im Mittelpunkt. Die Transformation der Daten erfolgt dabei in der Staging Area, in der Daten umgerechnet, zerlegt und verdichtet werden, ehe sie im abschließenden Ladeprozess an die richtige Stelle ins Data Warehouse weitergereicht werden. Im gesamten ETL-Prozess fallen Metadaten an, die detaillierte Informationen über Datenquellen, -bewegungen und –ziele sowie über strukturelle und inhaltliche Veränderungen der Daten während des Prozesses speichern.

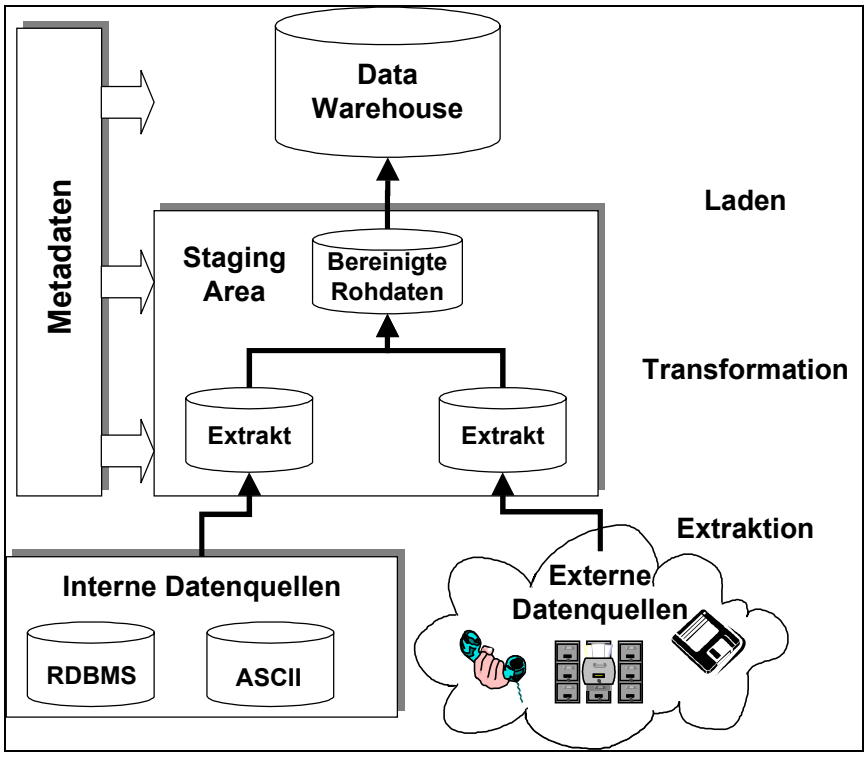

Abb. 3: Extraktions-, Transformations- und Ladeprozess

Anbindung von Standardanwendungssoftware an ein Data Warehouse

Die wichtigsten Datenlieferanten für das Data Warehouse sind in der Regel die Transaktionssysteme des Unternehmens, die als Anwendungssoftware häufig auf betriebswirtschaftlicher Standardsoftware basieren. Die am weitesten verbreitete

Standardanwendungssoftware ist dabei das R/3-System der SAP AG, dessen Anbindung an ein Data Warehouse exemplarisch untersucht werden soll.

Betriebswirtschaftliche Standardsoftware

Der Softwareeinsatz in Organisationen lässt sich unterscheiden in System- und Anwendungssoftware, die beide individuell erstellt sein können, oder in Form von Standardsoftware von einem Softwarelieferanten angeboten werden (Abb. 4). Während Individualsoftware generell den Vorteil der maßgeschneiderten Anpassung an die betrieblichen Erfordernisse bietet, ergeben sich jedoch auch Nachteile durch den hohen Erstellungs- und Wartungsaufwand. Letzterer wirkt sich besonders nachteilig in dynamischen Umgebungen aus, wo ständig neue Anforderungen abgebildet werden müssen, die von internen Veränderungen des Tätigkeitsfeldes, Verbesserungswünschen oder Sachzwängen aber auch durch externe Rahmenbedingungen wie Verordnungen oder Gesetze induziert werden.

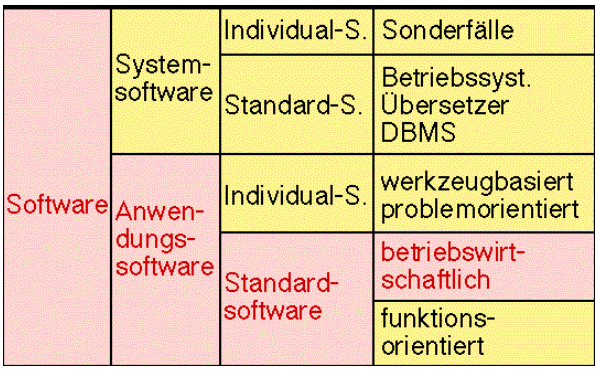

Abb.4: Softwarekategorien [ThHu96, S. 34]

Die betriebswirtschaftliche Anwendungssoftware hat zur Aufgabe, den bei der betrieblichen Leistungserstellung anfallenden Informationsfluss zu steuern und Daten zu integrieren, erzeugen, speichern und weiterzugeben. Beispiele sind Warenwirtschaftssysteme, Programme zur Rechnungslegung oder die Datenerfassung in der Produktion. Anfang 1999 setzten 54% der deutschen Unternehmen für diese Aufgaben Standardanwendungssoftware ein, wobei von den verbleibenden Unternehmen 11% eine Einführung planten [Meta99].
Standardanwendungssoftware im betriebswirtschaftlichen Bereich wird in der Regel als Softwarebibliothek angeboten. Softwarebibliotheken bestehen aus verschiedenen Funktionsmodulen, die je nach Bedarf ausgewählt und angepasst werden können. Besondere Vorteile ergeben sich durch die horizontale und vertikale Datenintegration, die durch die Weitergabe und Aufnahme von Daten anderer Funktionsmodule und der Benutzung einer gemeinsamen Datenbasis erzielt wird. Weiterhin steht mit der Softwarebibliothek ein umfassender betriebswirtschaftlicher Wissensschatz zur Verfügung, der im eigenen Unternehmen eingesetzt werden kann. Letztlich wirken sich die allgemeinen Vorteile von Standardsoftware aus, da das einzelne Unternehmen vom Softwareanbieter neue Versionen beziehen kann, die dem technischen Fortschritt angepasst sind und Weiterentwicklungen oder neue Funktionen bereitstellen.
Softwarebibliotheken sollen generell folgenden Anforderungen erfüllen [ThHu94, S. 45]:

- Einsatz einer allgemein verbreiteten und auf unterschiedlichen Plattformen einsetzbaren Programmiersprache
- Funktionsumfang mit einem Abdeckungsgrad von mindestens 80% der in jedem beliebigen, konkreten Unternehmen benötigten betriebswirtschaftlichen Funktionen
- Koordinierter Zugriff aller Module auf eine gemeinsame, laufend aktualisierte Datenbank

- Anpassbarkeit der einzelnen Programmmodule an betriebliche Anforderungen durch Parametereinstellung oder andere Adaptionsverfahren
- Dynamische Adaptionsfähigkeit zur Änderung von Parametereinstellungen unter Konsistenzerhaltung des lebendigen Anwendungssystems
- Dynamische Austauschbarkeit einzelner Module
- Konsistente Datenverarbeitung auch bei asynchroner Anpassung bzw. Weiterentwicklung der Module

Die am weitesten verbreitete betriebswirtschaftliche Standardsoftwarebibliothek ist R/3 der SAP AG, die Anfang 1999 in Deutschland einen Marktanteil von 58 % erzielen konnte [Meta99]. Weitere Anbieter sind beispielsweise Baan, Oracle, Peoplesoft und Navision.

Ergänzung von Standardsoftware durch EIS

Moderne Standardanwendungssysteme wie R/3 bauen auf einer integrierten (relationalen) Datenbasis auf und bieten den Anwendern eine Fülle bereits integrierter Berichte. Dies wirft die Frage auf, warum viele Unternehmen dennoch nicht auf den Einsatz zusätzlicher Werkzeuge zur Entscheidungsunterstützung verzichten. Die wichtigsten Gründe für den Einsatz von Data Warehouse- und Business Intelligence-Lösungen als Ergänzung zu Standardanwendungssoftware sind:

- **Systembelastung**: Analytische Anfragen und Berichtsgenerierungen können operative Systeme so stark belasten, dass es zu einer spürbaren Beeinträchtigung der Transaktionskapazität kommt. Neben unkalkulierbar hohen Rechenzeiten durch komplexe Operationen sind auch die Zeitpunkte der Belastung nicht vorhersagbar, was dem Ziel einer möglichst gleichmäßigen Systembeanspruchung der operativen Systeme widerspricht.
- **Flexibilität**: Werden zusätzliche Informationen benötigt, die in den Standard-Berichten nicht enthalten sind, erfordert dies die individuelle Entwicklung neuer Berichte mit Hilfe einer Programmiersprache. Ad-Hoc-Anforderungen können daher nicht bewältigt werden.
- **Benutzerführung**: Die Berichtsmöglichkeiten von Standardanwendungssoftware können nicht allen Benutzern entscheidungsorientierter Informationssysteme angeboten werden. Insbesondere sollten Anwender, die nicht täglich mit dem System arbeiten, mit intuitiveren Oberflächen arbeiten können.
- **Funktionalität**: Entscheidungsunterstützende Systeme bieten zahlreiche Funktionen, die in reinen Reporting Lösungen nicht verfügbar sind. Beispiele sind flexible Navigationsinstrumente, betriebswirtschaftliche Analysemethoden wie Prognosen, Abweichungsanalysen etc. oder komplexere Data Mining Verfahren.
- **Zusätzliche Datenquellen**: Schätzungen zufolge können in Unternehmen durchschnittlich nur 60-80% der entscheidungsrelevanten Daten aus operativen Transaktionssystemen wie R/3 gewonnen werden. Besondere Stärke des Data Warehouse ist die Integration von Daten aus verschiedensten internen und externen Quellen in einem einheitlichen Modell.

Das Thema Data Warehouse wird auch von der SAP selbst aufgegriffen, um das On-Line Transaction Processing (OLTP) System R/3 um ein On-Line Analytical Processing (OLAP) System zu ergänzen. Kernelement der Strategie ist das SAP Business Information Warehouse (SAP BW), das insbesondere vorkonfigurierte Modelle wie beispielsweise Marktsegmentanalyse, Finanzmanagement etc. bereitstellt. Diese Modelle unterstützen die multidimensionale Analyse und erlauben eine schnelle Einführung und Verwendbarkeit des Systems, da die Schnittstellen und Extraktionsmechanismen aus dem operativen R/3 System bereits definiert sind. Weichen die Analyseanforderungen der Anwender jedoch von den

vorgegebenen Modellen ab, müssen im Einzelfall Anpassungen vorgenommen werden. Insbesondere die Integration von SAP-fremden Daten erfolgt nur über den Einsatz von ETL-Werkzeugen von Drittanbietern. Zur Sicherstellung einer hohen Performance verwendet das SAP BW einen dedizierten Server. Die durch die OLAP-Engine aufgebauten InfoCubes bieten die Möglichkeit, bestimmte Aggregationen vorzuhalten, die das Laufzeitverhalten typischer Anfragen enorm verkürzen.

SAP R/3 als Datenquelle

Die besonderen Herausforderungen bei der Anbindung von Standardanwendungssoftware werden am Beispiel des am weitesten verbreiteten Systems R/3 der SAP AG deutlich. Sie liegen vor allem in der für operative Anwendungszwecke optimierten Systemarchitektur, die einen Datenzugriff erschwert.

Systemarchitektur

Das SAP R/3-System basiert auf einer 3 Tier-Client/Server-Architektur, die sich in Datenbank-, Applikations- und Präsentationsserver aufteilt (Abb. 5).

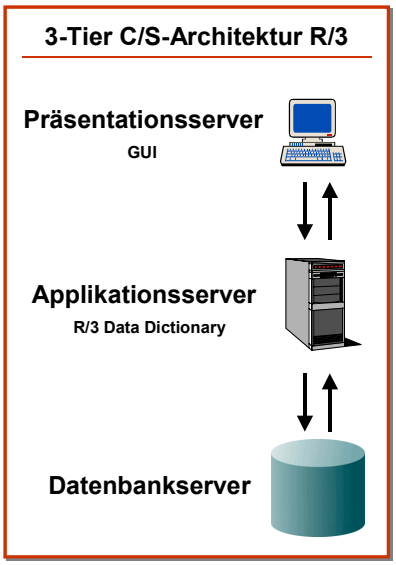

Abb. 5: Systemarchitektur des SAP R/3-Systems

Diese Architektur gewährleistet eine hohe Flexibilität und Skalierbarkeit bei der Wahl der Betriebs- und Datenbanksysteme. Dem Datenbankserver obliegt dabei die Verwaltung aller Datenbestände im SAP-System. Allerdings werden nicht alle Daten so in der Datenbank abgespeichert, dass sie via Standard Query Language (SQL) abrufbar sind. Neben den transparenten Tabellen existieren auch sogenannte Pool- und Cluster-Tabellen, die nicht immer exakt den physikalischen Tabellen entsprechen, wie sie im Datenbank-Server abgelegt sind. Oft werden mehrere logische SAP-Tabellen physikalisch in einer Tabelle komprimiert und nicht streng normalisiert abgespeichert. Dies erfolgt vor allem aus Platz- und teilweise Performance-Gründen und betrifft auch Bereiche, die beim Aufbau eines Data Warehouse betroffen sind. Beim direkten Zugriff auf Pool- und Clustertabellen können somit Informationsverluste auftreten, die einen direkten Zugriff auf den Datenbankserver vereiteln. Noch weitergehende Restriktionen gelten für die intransparenten Tabellen, die beispielsweise für das Human Resources (HR)-Modul benutzt werden. Daten werden aus Sicherheitsgründen verschlüsselt abgelegt und können nur durch besondere ABAP/4-Routinen wieder extrahiert werden. Weitere Schwierigkeit bereiten Hierarchien wie Kostenstellen oder Produktgruppen, die erst zur Laufzeit aufgebaut werden. Je nach Anforderungen der Analysewerkzeuge

müssen diese hierarchischen Strukturen stark umgeformt werden, um sinnvolle Datenstrukturen im Data Warehouse zu ermöglichen.

Datenzugriff

Die flexible Architektur des SAP-Systems erlaubt den Zugriff mehrerer Applikations-Server auf eine Datenbank. Jedes R/3-Modul basiert auf der Programmiersprache ABAP/4 (Advanced Business Application Programming 4GL) und wird über einen Applikations-Server gestartet. ABAP/4-Programme greifen aber über das R/3 Data Dictionary, das Teil des Applikationsservers ist, auf die Daten zu. Für die Funktionen ist es dabei unerheblich, ob der Zugriff auf transparente (relationale) oder Pool- und Clustertabellen erfolgt.
Für die Datenextraktion ergeben sich nun drei Möglichkeiten für einen Zugriff auf die Datenbasis von R/3:

1. **Entwicklung von ABAP/4 Programmen**. Abgesehen von der Knappheit an ABAP/4-Programmierern ist die Eigenentwicklung der Datenextraktions- und Transformationsroutinen zeitraubend und teuer – sowohl bei der Erstellung als auch bei der Wartung. Ständig wechselnde Anforderungen sind in „lebenden“ Data Warehouse-Lösungen Normalität.
2. **Direkter Zugriff auf die Datenbank**. SQL-basierte Reporting-Werkzeuge können direkt oder über ODBC auf den Datenbankserver zugreifen und Tabellen auslesen. Hauptnachteil dieser Vorgehensweise ist die Tatsache, dass wichtige Tabellen in Form von Pool- und Clustertabellen vorliegen, die nicht durch SQL gelesen werden können. Weiterhin wird aufgrund der angestrebten Datenbankunabhängigkeit von R/3 ein Teil der Datenbankverwaltung wie das Sperren von Datensätzen bei Änderungsläufen im Applikations-Server überwacht. Auch wenn moderne Datenbankmanagementsysteme den gleichzeitigen Zugriff verschiedener Anwendungen beherrschen, können trotzdem Inkonsistenzen entstehen, wenn Datensätze verarbeitet werden, die noch in offenen Transaktionen involviert sind.
3. **Extraktion über den Applikationsserver**. Extraktionswerkzeuge generieren ABAP/4 Code, der auf den Applikationsserver übertragen und ausgeführt wird oder steuern den Aufruf von Funktionsmodulen im Applikationsserver. Der Zugriff erfolgt dabei in der Regel über Remote Function Calls (RFC), mit denen die Extraktionswerkzeuge die Prozesse auf dem Applikationsserver anstoßen. So können sämtliche Tabellen ausgelesen werden, ohne dass der Anwender selbst ABAP/4 Programme schreiben muss. Die Integration des R/3 Data Dictionary bietet den weiteren Vorteil, dass der Benutzer unabhängig von Änderungen an der R/3 Datenstruktur z. B. bei Releasewechsel wird.

Dieser grobe Überblick über die Architektur des R/3-Systems und den davon abhängigen Anbindungsmöglichkeiten zeigt, dass der vermeintlich leichte Zugriff auf die operative Datenbasis eines Transaktionssystems nicht ganz unproblematisch ist. Gerade der entscheidende Schritt, das Mapping der R/3-Daten in ein Data Warehouse stellt viele Unternehmen vor immense Probleme.

Data Warehouse-Aufbau mit SAS Institute Software

SAS bietet mit dem SAS/Warehouse Administrator eine leistungsfähige Steuerungskomponente für den gesamten Extraktions-, Transformations- und Ladeprozeß eines Data Warehouse an. Zur Anbindung der operativen Systeme wird das Modul SAS/ACCESS benutzt, das über Schnittstellen zu einer Vielzahl von Datenquellen inklusive einem Interface zu SAP R/3 verfügt.

Die hier vorgestellten Erfahrungen mit den Werkzeugen von SAS Institute basieren auf einem ausführlichen Marktvergleich von ETL- und Data Warehouse Lösungen des Lehrstuhls für Betriebswirtschaftslehre und Wirtschaftsinformatik der Universität Würzburg, der ständig aktualisiert im Business Application Research Center (www.barc.de) verfügbar ist.

Anbindung des SAP R/3-Systems mit SAS/ACCESS

Die im vorhergehenden Kapitel beschriebenen Möglichkeiten der Datenextraktion über Datenbank- oder Applikationsserver werden durch das SAS/ACCESS Modul realisiert.
Soll eine direkte Extraktion von Daten aus transparenten Tabellen der relationalen Datenbank des Datenbankservers vorgenommen werden, so stellt SAS/ACCESS in Verbindung mit SAS/Connect entsprechende Schnittstellen bereit, die eine problemlose Überführung der Daten in das SAS System ermöglichen. Alle gängigen relationalen Datenbanken, die als Datenbankserver für SAP R/3 dienen können (Oracle, IBM DB2, Informix, Sybase, SQL Server, etc.) werden auf unterschiedlichsten Plattformen angesprochen.
Deutlich komplexer, bei realistischen Anforderungen jedoch nicht vermeidbar, ist der Zugriff auf das R/3-System über den Applikationsserver. Hierfür bietet SAS Institute das SAS/ACCESS Interface für R/3 an.

Systemarchitektur

Die benötigten Komponenten zur Anbindung von R/3 mit SAS sind ein RFC-Server und das SAS ACCESS Interface to R/3 (Abb. 6).
Für die Verbindung zum R/3-System werden Remote Function Calls (RFC) benutzt, die eine dynamische und bilaterale Kommunikation zwischen ETL-Werkzeug und SAP-System zulassen. Im SAP R/3 Applikationsserver werden zwei ABAP/4-Funktionsmodule angesprochen, die über einen RFC-Server mit dem SAS/ACCESS Interface für R/3 im SAS System kommunizieren. Der gesondert installierte SAS RFC-Server kann auf dem SAP Applikationsserver oder einem anderen Rechner liegen und dient als Schnittstelle zwischen SAP-System und Warehouse zur Abwicklung der Datenströme zwischen den Systemen. Die Datenübertragung zwischen RFC-Server und SAS System wird über das TCP/IP-Protokoll abgewickelt. Innerhalb des R/3-Systems benutzt SAS eine Datenübertragung auf dem SAP-eigenen CPI-C Protokoll, was zu höheren Datenübertragungsraten beim Aufruf der ABAP/4-Datenbankabfragen innerhalb des Applikationsservers führen soll.

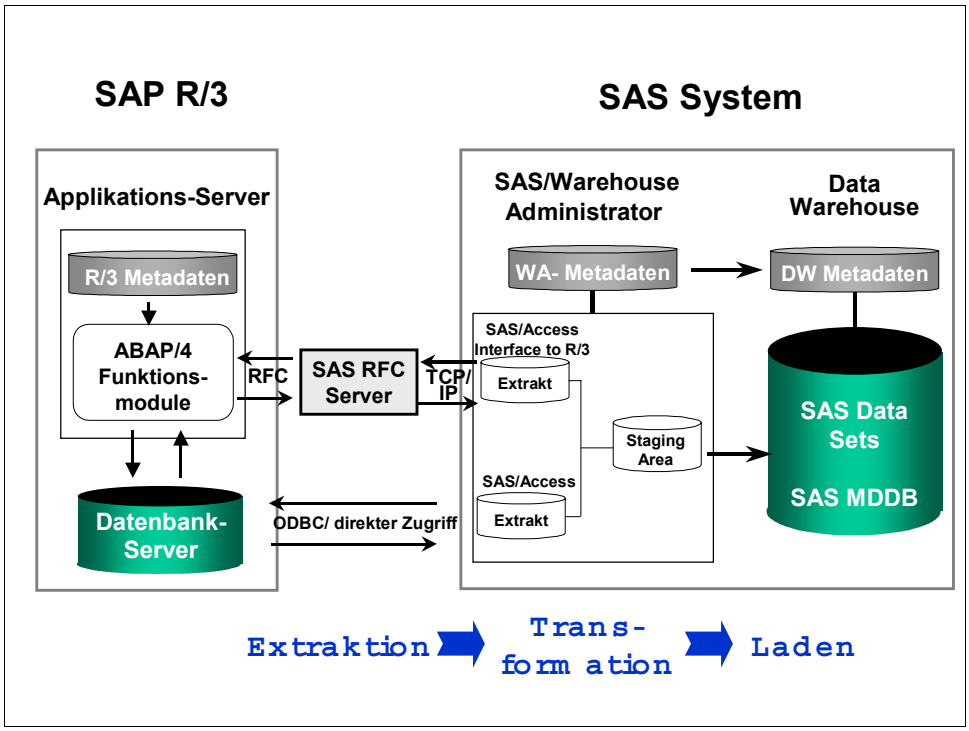

Abb. 6: Anbindung eines SAP R/3 Systems an SAS

Datenextraktion

Zur Extraktion von Daten aus dem R/3 System übergibt das SAS/ACCESS Interface dem RFC-Server die relevanten Parameter der Abfrage wie Tabelle, Spalten und "where-Bedingung". Es wird also kein ABAP/4-Code generiert. Vielmehr handelt es sich bei den mitgelieferten Modulen um parametrisierbare Gerüste, die mit den entsprechenden Parametern befüllt werden. Der RFC-Server reicht die Parameter an das ABAP/4-Funktionsmodul und an das ABAP/4-Reportmodul im SAP R/3-System weiter. Das ABAP/4-Funktionsmodul ist dabei der Funktionsbaustein, der die Kommunikation innerhalb des SAP R/3-Systems steuert, wobei das ABAP/4-Reportmodul die eigentlichen Abfragen an die SAP R/3-Datenbasis vornimmt. Das SAP R/3-System arbeitet die Parameter ab und liefert die Daten auf dem gleichen Weg zurück, so dass sie über den RFC-Server in das SAS System gelangen.

Datenselektion

Der erste Schritt bei einer SAP-Anbindung ist die Integration der Metadaten des R/3-Systems um eine Selektion der zu extrahierenden Daten zu ermöglichen. Daraufhin erlaubt der Data Model Explorer das Durchsuchen des R/3-Datenmodells nach den relevanten Quelldaten-beständen für die Extraktion.

Metadatenintegration

Über die Metadaten des R/3-Repositories können auf Tabellen- und Feldbeschreibungen zugegriffen und Änderungen des R/3-Datenmodells z. B. bei Releasewechsel oder Erweiterungen nachvollzogen werden. Im Repository des R/3-Systems findet sich in allen Systemsprachen sowohl ein kurzer Beschreibungstext zu den Tabellen, als auch zu den einzelnen Tabellenspalten. Angegeben ist eine Inhalts- bzw. Funktionsbeschreibung der Tabelle und der jeweiligen Tabellenspalte.
Zur Integration der Metadaten nimmt SAS einen kompletten Export des R/3-Repositories in das SAS System vor. Das zu Forschungszwecken extrahierte Repository eines produktiven R/3-Systems Release 4.0B umfasste trotz Beschränkung auf die Tabellen der Systemsprache

Deutsch dabei über 300 MB. Bei diesem Umfang des R/3-Repository nimmt der Vorgang auch bei einer schnellen Netzwerkverbindung mehrere Stunden in Anspruch und stellt natürlich nur ein statisches Abbild (Schnappschuss) der Metadaten dar. Vorteile ergeben sich jedoch in der Folge aus den schnelleren Zugriffsmöglichkeiten der lokal gehaltenen Daten. Bei vielen anderen Lösungen werden die Metadaten im R/3-System direkt durchsucht und erst bei Bedarf, d. h. bei Identifizierung durch den Benutzer als Datenquelle, in das ETL-Tool überführt. Diese Werkzeuge greifen bei jeder Suchanfrage direkt auf das aktuelle R/3-Repository zu, wobei sich vergleichsweise höhere Antwortzeiten ergeben können.
Nach dem vollständigen Export der R/3-Metadaten wird die Datenselektion durch Suchfunktionen und dem Data Model Explorer unterstützt. Die Komplexität dieses Prozesses sollte nicht unterschätzt werden: Im Testsystem waren Metadaten enthalten von ca.
13.000 transparenten Tabellen,
2.000 Pool- und Clustertabellen,
12.000 Views und
25.000 intransparenten Tabellen.
Auch wenn aus diesen insgesamt ca. 52.000 Tabellen mit 840.000 Feldern die Systemtabellen und andere nicht benutzte ausgeblendet werden, verbleibt doch ein riesiger Datenbestand, in dem die Kennzahlen und Dimensionen des Data Warehouse Modells gesucht werden müssen.
Neben der reinen Datenfülle bereitet das undurchsichtige Datenschema und die nicht-sprechenden Bezeichnungen der Tabellen und Spalten mit 4- oder 5-Ziffern-Kürzeln wie VBAK oder T001 zusätzliche Schwierigkeiten. Die Unterstützung eines Werkzeuges bei der Datenselektion ist daher enorm wichtig, um eine effiziente Identifizierung der Datenquellen zu erlauben.

Navigationsunterstützung im Datenmodell

Neben einer Suche nach Begriffen in den Metadaten kann mit dem SAS Data Model Explorer bei der Tabellensuche auch ein Einstieg über die modulare Struktur von R/3 erfolgen. Hier bietet eine Zuordnung von bestimmten Funktionen innerhalb eines R/3-Moduls zu Tabellennamen eine gute Möglichkeit, gesuchte Daten schneller zu identifizieren. Hilfreich ist hier die graphische Übersicht des R/3-Datenschemas in einer Baumstruktur mit einer Angabe der jeweils angesprochenen Tabellen (Abb. 7). Ist ein gesuchtes Datenfeld in einer Tabelle identifiziert worden, ist es zum Verständnis des R/3-Datenschemas weiterhin wichtig zu wissen, mit welchen Tabellen die gefundene verknüpft ist und wo das Datenfeld noch auftaucht. Hier hilft die Verfolgung von Tabellenschlüsseln und eine Anzeige von gleichnamigen Datenfeldern in anderen Tabellen.

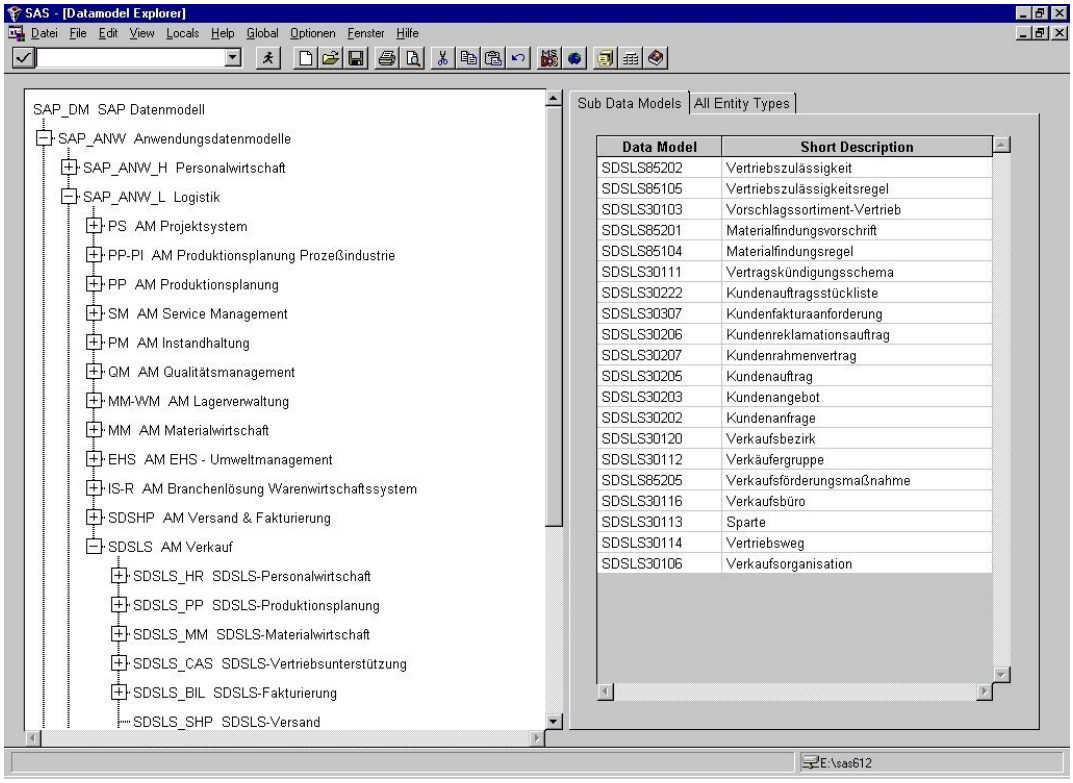

Abb. 7: Data Model Explorer des SAS/ACCESS Interface to R/3

Prozesssteuerung mit dem SAS/Warehouse Administrator

Die Anbindung des SAP R/3-Systems ermöglicht die Extraktion der gewünschten Datenfelder in das SAS System. Innerhalb des SAS/Warehouse Administrators können nun die identifizierten Datenquellen mit den Data Warehouse Zieltabellen im SAS System verbunden werden. Bevor ein Data Warehouse mit relationalen SAS Data Sets oder in multidimensionalen Strukturen (SAS/MDDB) aufgebaut werden kann, ist in der Regel eine Transformation der Daten nötig. Die oben beschriebenen Schritte Filterung, Harmonisierung, Verdichtung und Anreicherung werden im SAS/Warehouse Administrator definiert und überwacht.

Transformation und Laden

Die meisten Transformationen lassen sich im SAS/Warehouse Administrator mit Hilfe von grafischen Objekten, Wizards und vorgegebenen Funktionen definieren. Müssen komplexere Transformationen vorgenommen werden, so steht im SAS-System mit den SAS Data Steps eine mächtige Programmiersprache zur Verfügung. Über diese können zusätzlich noch externe Unterprogramme aufgerufen werden, welche auch in Cobol, Assembler, PL/1, Fortran oder C geschrieben sein können. Zur Veranschaulichung des Transformationsprozesses dient der Prozesseditor, der die Überführung der einzelnen Quelltabellen mit den dazugehörigen Abbildungs- und Transformationsregeln in die Zieltabellen grafisch darstellt (Abb. 8). Die angezeigten Elemente lassen sich per Anklicken auf weitere Einzelheiten und Eigenschaften hin untersuchen und modifizieren. Insgesamt ergibt sich eine übersichtliche Darstellung des Datenflusses mit seinen Abhängigkeiten und vorgenommenen Transformationen zwischen Datenquellen in Vorsystemen und den Strukturen im Data Warehouse.

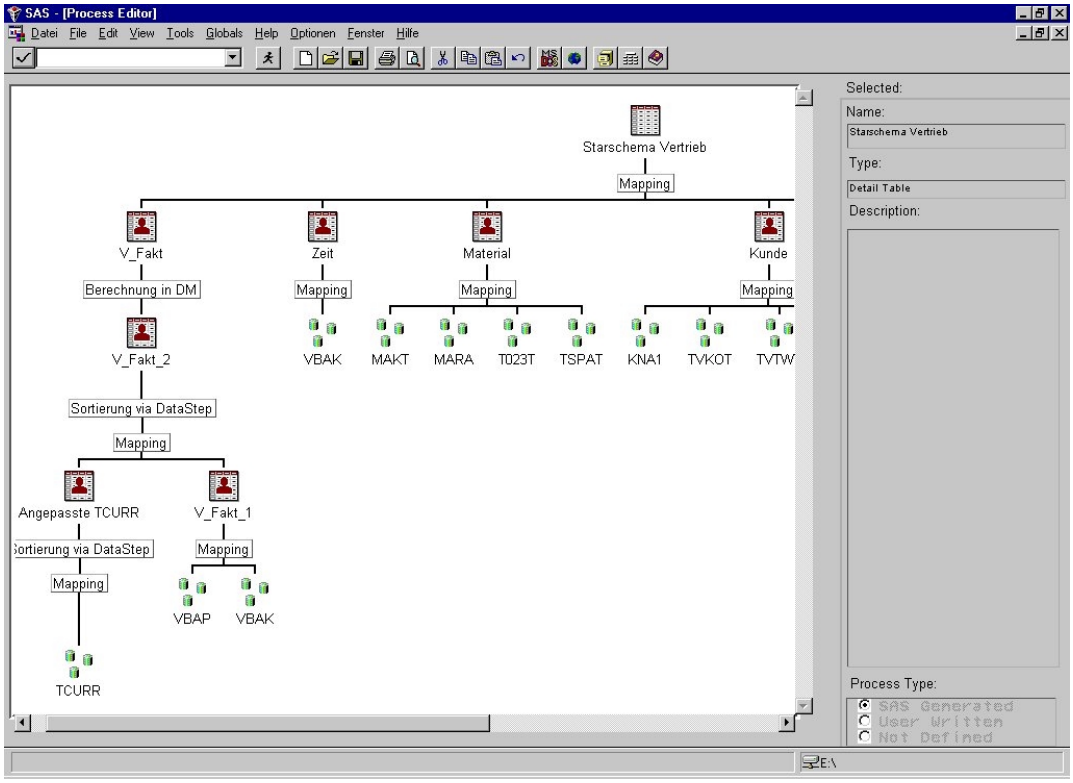

Abb. 8: Datenfluss im Prozesseditor des SAS/Warehouse Administrators

Die Administration und Ausführung aller Prozesse lassen sich im SAS/Warehouse Administrator überwachen und steuern. Der Ladeprozess kann manuell angestoßen werden oder über einen Scheduler, der verschiedene Jobs auf Unix und Windows NT zu bestimmten Zeitpunkten oder auch in Zeitintervallen ausführt (Abb. 9). Alle ausgeführten Prozesse werden außerdem in eine Log-Datei geschrieben, die detaillierte Status- und Fehlermeldungen wiedergibt. Zusätzlich lassen sich über das Changed Data Capture Support-Modul der SAS Software Loginformationen wie Update, Insert und Delete aus den Logdateien der Datenquellen lesen, um bei einem Delta Update (ausschließliches Nachladen der Veränderungen) zur Verfügung zu stehen. Über eine Browse-Funktion lassen sich über den SAS/Warehouse Administrator die in den Zieltabellen gespeicherten Daten abfragen.

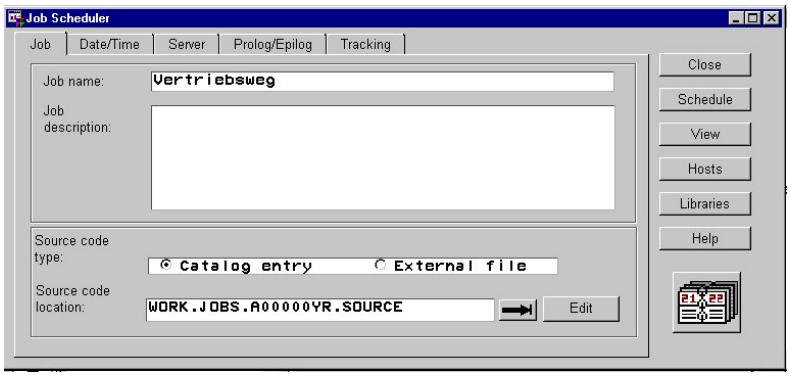

Abb. 9: SAS Scheduler

Metadaten

Innerhalb des SAS/Warehouse Administrators werden alle technischen und betriebswirtschaftlichen Metadaten gesammelt und in der SAS-eigenen Datenbank gespeichert. Die Struktur des Data Warehouse wird grafisch und die in den Tabellen enthaltenen Informationen werden tabellarisch veranschaulicht. Sowohl die verwendeten

Tabellen aus den Datenquellen als auch die Zieltabellen sind in einem Schema ersichtlich (Abb. 10). Für die SAP R/3-Daten wird zusätzlich ein eigenes Metadaten Repository bereitgestellt, welches neben den einzelnen Tabellen auch das Datenmodell von SAP R/3 wiedergibt, das mit dem Data Model Explorer angezeigt und durchsucht werden kann (s.o.).

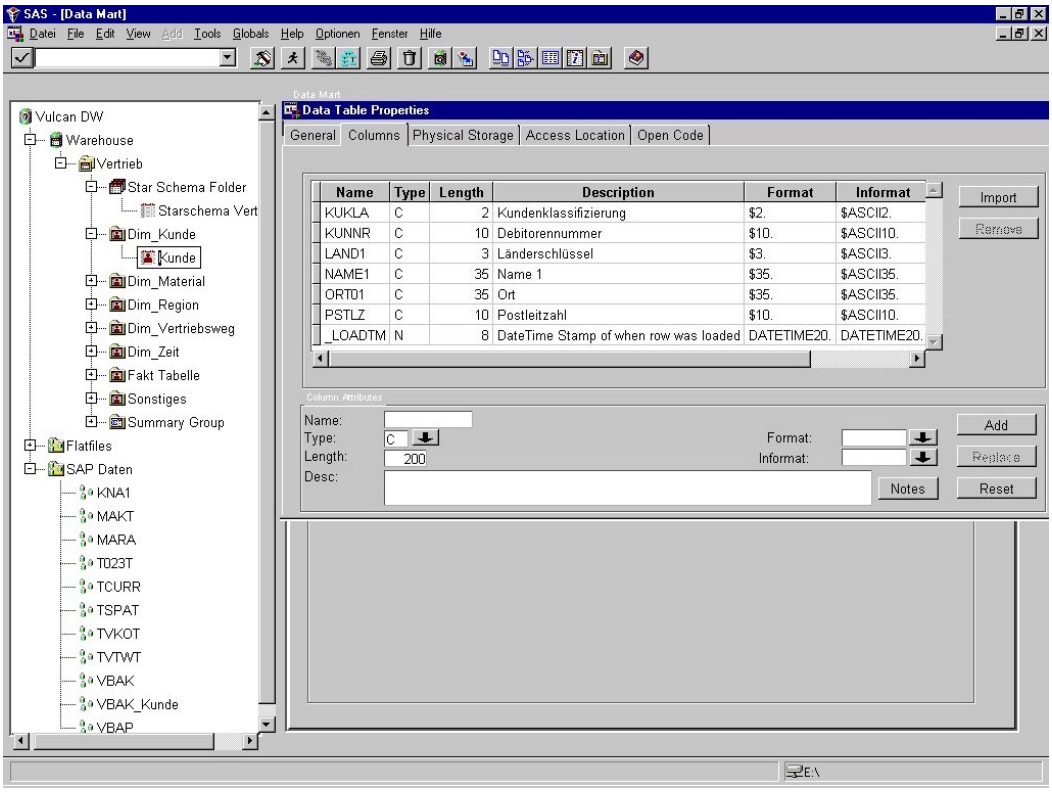

Abb. 10: Quell- und Zieldaten im SAS/Warehouse Administrator

Einsatz von vorgefertigten Datenmodellen

Vorgefertigte Data Warehouse-Datenmodelle (Templates) für abgegrenzte betriebliche Funktionsbereiche oder Aufgaben wie Marketing/Vertrieb, Controlling, Produktion/Logistik oder Einkauf (Procurement) bieten bedeutende Vorteile in Form von drastisch gesenkten Einführungszeiten des Data Warehouse. Ein Template besteht einerseits aus einem spezifischen Datenmodell und andererseits aus vordefinierten SAP R/3-Extraktions- und Transformationsprozessen. Da auf ein bewährtes Datenmodell zurückgegriffen werden kann, das bereits mit den R/3-Tabellen verknüpft ist, müssen nur Anpassungen an die unternehmensspezifischen Eigenheiten vorgenommen werden. Geht man davon aus, dass eine Vielzahl der Analysebedarfe von Unternehmen übereinstimmen, so können durch den Einsatz von Templates erhebliche Zeit- und Kostenvorteile erzielt werden [BaSc99]. SAS bietet hier vorgefertigte Modelle für verschiedene betriebliche Funktionsbereiche wie Vertrieb oder Einkauf an.

Fazit

Die Integration der Daten aus verschiedenen operativen Systemen und anderen Datenquellen in ein Data Warehouse ermöglicht die umfassende Verteilung von Informationen innerhalb der Organisation und die zielgerichtete Unterstützung des Managements beim Treffen von Entscheidungen. Die größte Menge der einzubindenden Daten liegt in den Transaktionssystemen jedes Unternehmens vor, die häufig auf betriebswirtschaftlicher Standardsoftware basieren. Das am weitesten verbreitete Softwareprodukt R/3 der SAP AG

stellt bei der Extraktion der Daten aus dem System besondere Anforderungen, die nur durch spezielle Werkzeuge erfüllt werden können. Das SAS ACCESS Interface to R/3 ermöglicht die einfache und transparente Integration von SAP R/3-Datenquellen in ein Data Warehouse, wobei der SAS/Warehouse Administrator eine leistungsfähige Unterstützung für den gesamten Prozess der Extraktion, Transformation und des abschließenden Ladens der Daten in die Zieltabellen des Data Warehouse bietet.

Literatur

[BaSc99] Bange, C.; Schinzer, H.: Vorkonfigurierte Data Marts - Ready to Run. In: is report 3 (1999) 9, S. 44-49.

[Bull95] Bullinger, H.-J. et al.: Produktivitätsfaktor Information: Data Warehouse, Data Minig und Führungsinformationen im betrieblichen Einsatz. In: Bullinger, H.-J. (Hrsg.) IAO-Forum: Data Warehouse und seine Anwendungen. Data Mining, OLAP, und Führungsinformationen im betrieblichen Einsatz. IRB, Stuttgart 1995, S. 11-30.

[Codd93] Codd, E. F. et al.: Providing OLAP to User-Analysts. In: An IT Mandate, Whitepaper, Codd & Associates, o. O. 1993.

[Hans95] Hansen, W. R.: Das Data Warehouse. Lösung zur Selbstbedienung der Anwender. In: Bullinger, H.-J. (Hrsg.) IAO-Forum: Data Warehouse und seine Anwendungen. Data Mining, OLAP, und Führungsinformationen im betrieblichen Einsatz. IRB, Stuttgart 1995, S. 33-48.

[Inmo92] Inmon, W. H.: Building the Data Warehouse. QED Technical Publishing Group, Wellesley 1992.

[Kell94] Kelly, S.: Data Warehousing. The route to mass customisation. Wiley, Chichester 1994

[KeFi98] Kemper, H.-G.; Finger, R.: Datentransformation im Data Warehouse – Konzeptionelle Überlegungen zur Filterung, Harmonisierung, Verdichtung und Anreicherung operativer Datenbestände. In: Chamoni, P.; Gluchowski, P. (Hrsg.): Analytische Informationssysteme. Springer, Berlin 1998, S. 61-77.

[Meta99] Untersuchung des Marktes für ERM-Software durch die META Group, München, Februar 1999.

[Oehl99] Oehler, K.: OLAP. Grundlagen, Modellierung und betriebswirtschaftliche Lösungen. Hanser, München etc. 2000.

[ScBa99] Schinzer, H.; Bange, C.; Mertens, H.: Data Warehouse und Data Mining. Marktführende Produkte im Vergleich. Vahlen, München 1999.

[Schu94] Schur, S. G.: The Database Factory. Active Database for Enterprise Computing. Wiley, New York 1994.

[ThHu96] Thome, R.; Hufgard, A.: Continuous System Engineering. Vogel, Würzburg 1996.