

Arbeitsberichte des Instituts für Wirtschaftsinformatik
Herausgeber: Prof. Dr. J. Becker, Prof. Dr. H. L. Grob,
Prof. Dr. U. Müller-Funk, Prof. Dr. G. Vossen

Arbeitsbericht Nr. 51

**Entwicklung eines Data Warehouse
für das Produktionscontrolling:
Konzepte und Erfahrungen**

Stefan Eicker, Reinhard Jung,
Michael Nietsch, Robert Winter

Institut für Wirtschaftsinformatik der Westfälischen Wilhelms-Universität Münster,
Grevener Str. 91, 48159 Münster, Tel. (0251) 83-9750, Fax (0251) 83-9754

November 1996

Inhalt

1	Einführung	3
2	Grundlegende Fragen der Entwicklung eines Data Warehouse	6
3	Istanalyse	9
	3.1 Aufgaben der Istanalyse	9
	3.2 Projekterfahrungen	11
4	Entwicklung des Sollkonzepts	12
	4.1 Fachlich-betriebswirtschaftliches Sollkonzept	13
	4.2 Technisches Sollkonzept	15
5	Konzeptioneller Entwurf	16
	5.1 Uniforme Darstellung des Zeitbezugs	16
	5.2 Spezifikation der Ladevorgänge	17
	5.3 Detailentwurf für den Auswertungsbereich und Spezifikation von Aktualisierungsstrategien	18
6	Realisierung	19
	Literatur	21

Zusammenfassung

Aufgabe eines Data Warehouse ist die schnelle und flexible Bereitstellung entscheidungsrelevanter Daten. Es stellt damit - je nach Interpretation - eine Weiterentwicklung der Entscheidungsunterstützungssysteme oder eine auf Analyseaufgaben ausgerichtete Datenbank dar. Um seine Aufgabe erfüllen zu können, muß ein Data Warehouse heterogene Datenquellen zu einer stabilen, konsistenten Datenbasis zusammenführen, Detaildaten für die analytischen Auswertungen (vor-)verdichten sowie auch zeitraumbezogene Längsschnittanalysen unterstützen. Die Entwicklung eines Warehouse unterscheidet sich deshalb in vielen Punkten von der Entwicklung eines traditionellen, transaktionsorientierten Anwendungssystems. Der folgende Beitrag diskutiert, welche Aufgaben jeweils in den verschiedenen Phasen einer Data Warehouse-Entwicklung anfallen und wie diese Aufgaben durchgeführt werden können. Die vorgestellten Konzepte und Erfahrungen sind Ergebnisse eines Kooperationsprojekts zwischen dem Institut und einem großen Maschinenbauunternehmen.

1 Einführung

Die Unternehmen der Industriestaaten sehen sich einem immer stärkeren Wettbewerb ausgesetzt. Gründe für die Verschärfung des Wettbewerbs liegen in der Öffnung und Deregulierung der Weltmärkte, der Emanzipation der Kunden und in dem Markteintritt neuer Konkurrenten aus Schwellenländern. Als entscheidende Leistungskriterien eines Unternehmens werden deshalb die Schnelligkeit, Flexibilität und Effizienz seiner Reaktionen auf geänderte Kunden- und Marktanforderungen eingestuft¹⁾.

Die Ausrichtung eines Unternehmens auf die genannten Leistungskriterien erfordert Entscheidungen hoher Qualität auf allen organisatorischen Ebenen. Hierzu benötigen die Entscheidungsträger zu jeder Zeit den Zugriff auf die entscheidungsrelevanten Informationen. Bei der computergestützten Informationsverarbeitung standen jedoch bisher vor allem die Automatisierung und Rationalisierung der Massendatenverarbeitung im Vordergrund, d.h. die Unterstützung des operativen Tagesgeschäfts und dessen wertmäßiger Abrechnung. Eine Auswertung der anfallenden Daten unter dem Aspekt der Entscheidungsunterstützung findet i.a. nur unzureichend statt. Hinzu kommt, daß die operativen Systeme einen engen zeitlichen Bezug (z.B. der laufende Monat, das laufende Quartal oder das laufende Jahr) besitzen; für eine Reihe entscheidungsrelevanter Auswertungen wie zum Beispiel Trendanalysen müssen aber auch „historische Daten“ zur Verfügung stehen²⁾.

Benötigt wird ein Konzept, das es ermöglicht, auf der Basis der bestehenden Anwendungssystemlandschaft den Informationsbedarf aller Entscheidungsträger abzudecken, d.h. ihnen zu ermöglichen, Daten System- und Plattform-übergreifend schnell und flexibel auszuwerten. Unter dem Schlagwort "Data Warehouse" bzw. "Information Warehouse" hat deshalb der Ansatz, Endbenutzern die von ihnen benötigten Daten aus den operativen Datenbeständen zu extrahieren und als *dispositiven Datenbestand* für den lesenden Zugriff zur Verfügung zu stellen³⁾, eine Renaissance und Ausweitung erfahren: Durch ein *Data Warehouse* sollen operative Daten zu entscheidungsrelevanten Daten verdichtet sowie thematisch und zeitlich geordnet werden⁴⁾.

Unterschiede des Data Warehouse zum dispositiven Datenbestand liegen vor allem darin, daß

- das Warehouse für die verschiedenen Entscheidungsebenen Daten auf unterschiedlichen Verdichtungsstufen bereitstellen muß.

1) Vgl. Andersen Consulting (1996), S. 39

2) Vgl. z.B. Inmon (1990), S. 137 f.

3) Vgl. z.B. Rohrig (1987).

4) Vgl. z.B. Rother (1995), Celko et al. (1995), I/S Analyzer (1994), Kelly (1994), Poe (1994) und Inmon (1992).

- in einem Warehouse sehr große Datenmengen enthalten sind; im Laufe der Zeit können sich Datenbestände im Giga- oder sogar Terabytebereich ansammeln.
- die Basis des Data Warehouse in Gestalt der operativen Datenbestände sehr breit ist.
- insbesondere als Folge dieser breiten Basis erhebliche Anstrengungen erforderlich sind, um die Konsistenz der Data Warehouse-Daten herzustellen; man spricht auch vom "Scrubbing" der Daten.
- zeitpunktbezogene Daten für Längsschnittanalysen zu Zeitreihen bzw. versionierten Datenobjekten ergänzt werden müssen.

Ein Data Warehouse umfaßt neben der Datenbasis auch die Umgebung, die die Datenbasis einerseits mit Daten versorgt und andererseits ihre Daten verfügbar macht. Neben der Warehouse-Datenbasis gehören zu einem Data Warehouse (i.w.S.) somit auch (vgl. Abbildung 1):

- die Datenlieferanten (unternehmensinterne und -externe Datenquellen),
- Transformationsprozeduren und
- Frontend-Tools (Abfrage- bzw. Analysewerkzeuge, eventuell auch Executive Information Systems (EIS) oder Decision Support Systems (DSS)).

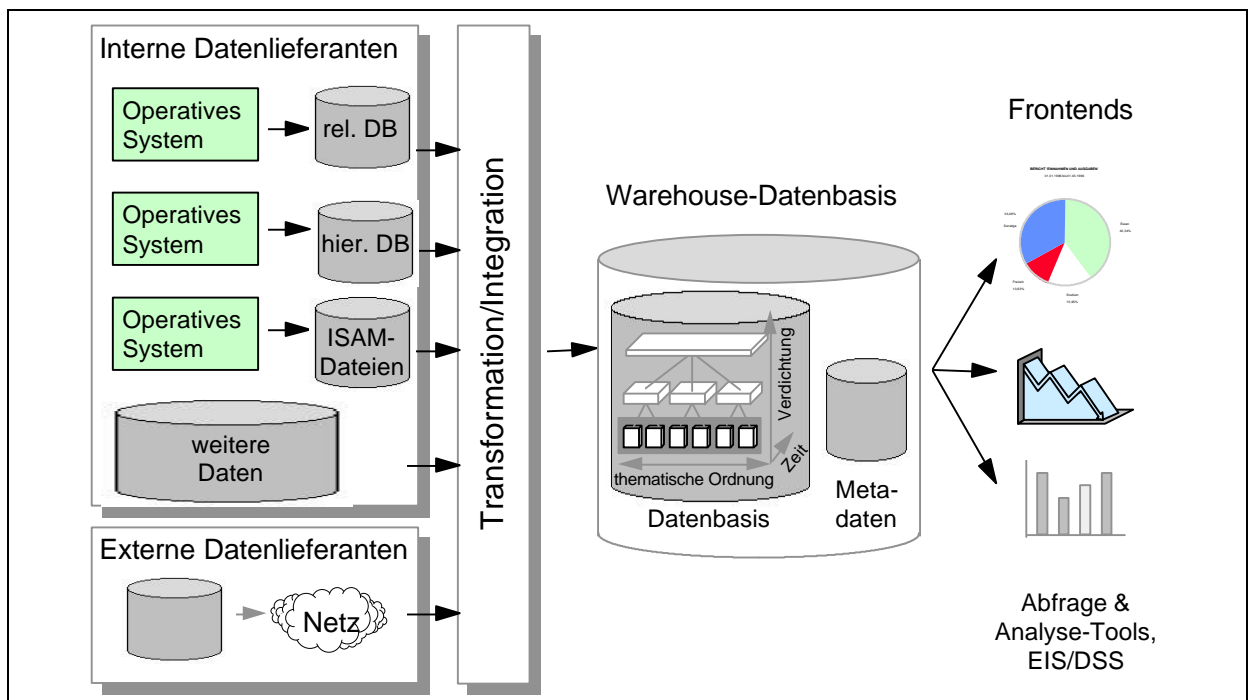


Abb. 1: Das Data-Warehouse (i.w.S.)

Eine weitere Komponente des Data Warehouse bilden die Metadaten, die den Inhalt und die Struktur der im Data Warehouse gespeicherten Daten beschreiben.

Für die Entwicklung eines Data Warehouse ergeben sich durch seine speziellen Charakteristika besondere Rahmenbedingungen. Darüber, inwieweit die verfügbaren Vorgehensweisen und Methoden der Systemplanung und -entwicklung trotz dieser besonderen Rahmenbedingungen für die Data Warehouse-Realisierung nutzbar sind, welche Vorgehensweisen und Methoden sich am besten eignen und welche Erweiterungen oder Änderungen eventuell notwendig sind, liegen bisher kaum Erfahrungen oder wissenschaftliche Erkenntnisse vor. Spezielle Fragen, die sich beispielsweise ergeben, sind:

- Welche Vorgehensweise eignet sich am besten für den Entwurf des konzeptionellen Schemas des Warehouse; soll der Entwurf top-down von den Anforderungen oder bottom-up von den existierenden Daten ausgehen oder beide Ansätze miteinander verbinden⁵⁾?
- Welche zusätzlichen Beschreibungsmittel sind zur Spezifikation des konzeptionellen Schemas erforderlich⁶⁾?
- Wie können die Verbindungen des Warehouse zu seinen Datenquellen mit ihren statischen und dynamischen Aspekten beschrieben werden?
- Wie können die verschiedenen Datenbereiche - Bereiche zum Laden von Daten aus den Datenquellen, Bereiche unterschiedlicher Verdichtungsstufe, mit unterschiedlichem Zeitbezug, mit Daten unterschiedlicher Abstraktionsebene etc. - zu einer homogenen Data Warehouse-Struktur integriert werden⁷⁾?

Im folgenden soll zu diesen Fragen aus der Sicht eines konkreten Warehouse-Entwicklungsprojekts Stellung genommen werden; das Projekt wurde 1996 im Rahmen einer Kooperation zwischen dem Institut für Wirtschaftsinformatik der Westfälischen Wilhelms-Universität Münster und dem Produktionsbereich eines großen Maschinenbauunternehmens durchgeführt. In dem Projekt wurde für das Unternehmen, das - verteilt auf verschiedene Produktionsstandorte - komplexe Produkte mit hoher Variantenvielfalt fertigt, ein Data Warehouse zur Unterstützung des Produktionscontrollings entworfen und prototypisch realisiert. Beteiligt an dem Projekt waren auf Seiten des Unternehmens der Leiter des Produktionscontrollings sowie drei seiner Mitarbeiter. Teilnehmer des Projekts auf Seiten des Instituts für Wirtschaftsinformatik waren neben den Autoren dieses Beitrags sechs Studenten.

5) Vgl. auch Alur (1995), S. 22, und Parsaye (1995).

6) Vgl. Moriarty (1995).

7) Vgl. auch Maloney (1995), Stodder et al. (1995), Weldon (1995) und White (1995).

Da das Warehouse als Oracle-Datenbank realisiert werden sollte, wurde als CASE-Entwicklungsumgebung Designer/2000 von Oracle (Release 1.2) eingesetzt. Als Zielumgebung für die Datenbasis bzw. als Datenbankserver des Data Warehouse wurde ein RS/6000-Abteilungsrechner mit einer Oracle-Datenbank (Version 7.2) genutzt. Auswertungen wurden prototypisch auf PCs unter Verwendung von Microsoft Access (Version 2.0) realisiert; der Zugriff auf die Oracle-Datenbank folgte über ODBC.

2 Grundlegende Fragen der Entwicklung eines Data Warehouse

Bei der Entwicklung klassischer Datenbankanwendungen sind die zu unterstützenden Geschäftsprozesse bekannt. Im Gegensatz dazu muß ein Data Warehouse grundsätzlich auch Analysen unterstützen, die zum Entwicklungszeitpunkt noch unbekannt sind. Ein Ansatz, diese Problematik zu umgehen, besteht darin, die zu unterstützenden Analysen einzuschränken: Statt eine Verknüpfung beliebiger Daten zu ermöglichen, kann nur jeweils ein bestimmtes mehrdimensionales Informationsobjekt (sog. "Hypercube"⁸⁾) hinsichtlich einer beliebigen Untermenge seiner jeweiligen Bezugsobjekte verdichtet werden. Jeder Hypercube repräsentiert somit eine Klasse von Informationsobjekten, die gleichartige Bezugsobjekte haben (z.B. Umsätze mit den Bezugsobjekten Produkt, Kunde und Zeitraum).

Bei der Verwendung von Hypercubes setzt sich das konzeptionelle Warehouse-Schema aus einfachen Grundstrukturen zusammen. Diese werden auch als "Star Schemes"⁹⁾ bezeichnet, da der zu analysierende Objekttyp im konzeptionellen Schema sternförmig von Bezugsgrößen-Objekttypen umgeben und mit diesen über Referenzen verbunden ist. Für die Bezugsgrößen-Objekttypen können zusätzlich Abstraktionshierarchien definiert werden, um weitere Analysefunktionen anbieten zu können ("Drilldown"¹⁰⁾).

Data Warehouses sollen in den meisten Fällen nicht nur die Verdichtung ausgewählter Klassen von Informationsobjekten unterstützen. Vielmehr erfordert die Komplexität der meisten betrieblichen Datenstrukturen (z.B. im Fertigungsbereich) und die Variabilität entsprechender Analysen eine Datenbasis, die sich nicht auf Hypercubes beschränkt und die deshalb nicht ausschließlich durch Star Schemes modelliert werden kann.

Für die Realisierung von Data Warehouse-Systemen, die auf dem Hypercube-Ansatz basieren, stehen entsprechende Softwarewerkzeuge (z.B. Oracle Express Analyzer) zur Verfügung. Rei-

8) Vgl. Bulos (1996), S. 34, o.V. (1996), S. 42.

9) Vgl. Menninger (1996), S. 84.

10) Vgl. Meredith et al. (1996), S. 27-28.

chen - wie in dem hier beschriebenen Projekt - Star Schemas nicht zur Modellierung des Warehouse aus, schließt sich dagegen die Verwendung derartiger Werkzeuge aus.

Da die Grundstruktur des konzeptionellen Warehouse-Schemas und das Datenmodell für seine Darstellung nicht durch ein Werkzeug festgelegt wurden, mußten im Rahmen der Projektvorbereitung eine geeignete Struktur entworfen und ein geeignetes Datenmodell ausgewählt werden. Grundsätzlich ergibt sich die Struktur direkt aus den oben genannten Charakteristika eines Data Warehouse: Zum einen müssen, um aussagefähige Analysen zu ermöglichen, die aus operativen Systemen übernommenen Daten im Warehouse bereinigt und integriert sowie eventuell vorverdichtet werden.

Zum anderen werden von Entscheidungsträgern Informationen (z.B. Kennzahlen) benötigt, die sich nicht als Daten in operativen Systemen finden und somit im Warehouse berechnet werden müssen. Zu trennen sind einmalige und Standardauswertungen. Erstere sind im Schema nicht vorzusehen, sondern bei Bedarf mit Hilfe der Frontend-Werkzeuge durchzuführen. Standardauswertungen sollten dagegen im Schema Berücksichtigung finden. Diese Notwendigkeit ergibt sich aus dem 100%-Anspruch des konzeptionellen Schemas: Wann immer ein dauerhaft relevantes Informationsobjekt durch einen Begriff beschrieben werden kann (und nicht nur durch eine Formel zu seiner Ableitung¹¹⁾), ist es auch im konzeptionellen Schema abzubilden. Es ergeben sich somit drei Bereiche des konzeptionellen Warehouse-Schemas (vgl. Abbildung 2):

1. Der *Ladebereich* umfaßt alle Datenstrukturen, die aus operativen Systemen übernommen werden.
2. Der *Basisbereich* umfaßt Datenstrukturen, die eine konsistente, vorverdichtete Zusammenfassung der Lade-Datenstrukturen darstellen.
3. Der *Auswertungsbereich* umfaßt weiter verdichtete oder anderweitig aufbereitete Datenstrukturen.

Um die Konsistenz originärer und abgeleiteter Daten zu sichern, müssen die entsprechenden Ableitungsregeln in geeigneter Form repräsentiert werden, und Änderungen originärer Daten sind in abgeleitete Daten zu propagieren. Bei der Entwicklung eines Data Warehouse sind somit über die Struktur der Daten hinaus auch Verhaltensaspekte in Form von Aktualisierungsprozeduren einerseits zwischen Lade- und Basisdaten und andererseits zwischen Basis- und Auswertungsdaten zu berücksichtigen. Die Verknüpfungen zwischen dem Lade- und dem Basisbereich implizieren Aktualisierungsprozeduren, die Verknüpfungen zwischen dem Basis- und dem Auswertungsbereich dagegen Verdichtungs- und Verknüpfungsvorgänge. Aus den Be-

11) Vgl. Rauh et al. (1993).

ziehungen innerhalb der einzelnen Bereiche können Prozeduren zur dynamischen Integritätsicherung abgeleitet werden.

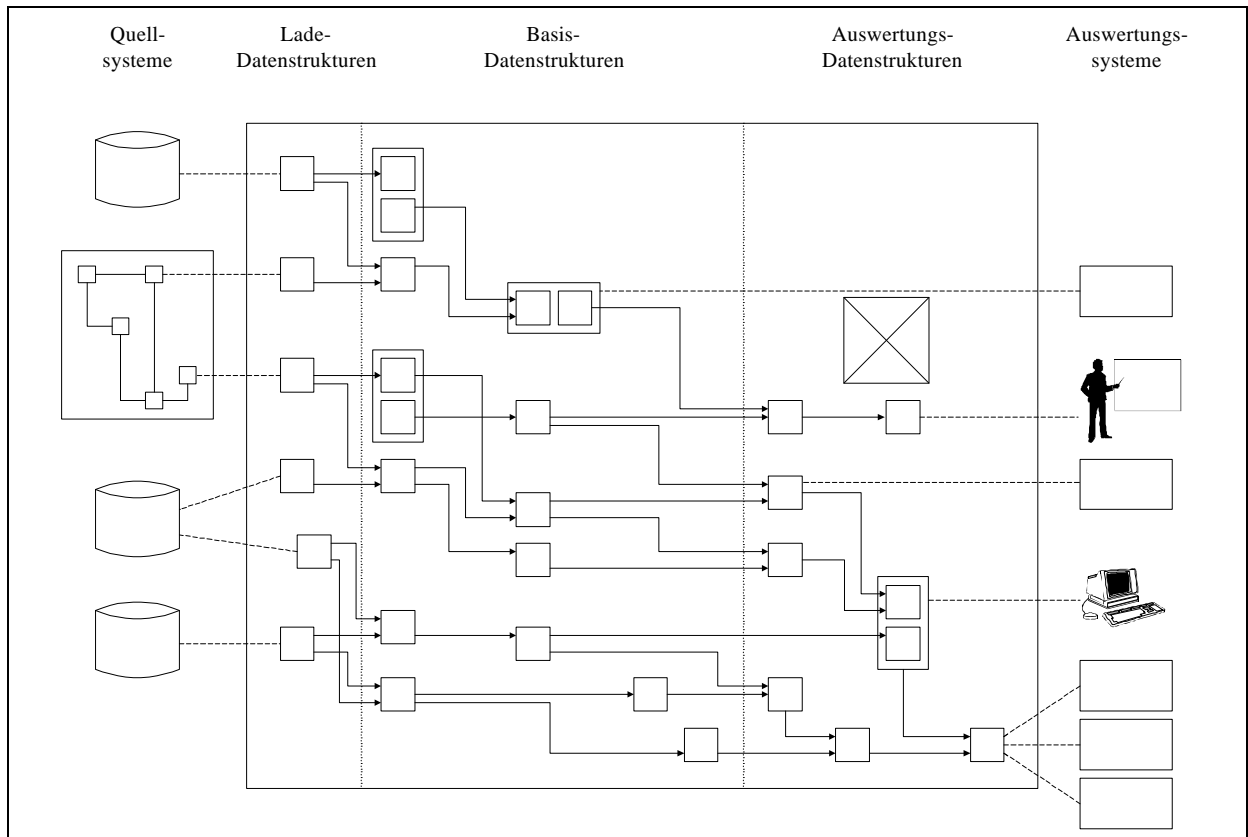


Abb. 2: Grundlegende Struktur des konzeptionellen Warehouse-Schemas

Um eine integrierte Entwicklung von Daten- und Funktionssicht zu gewährleisten, muß versucht werden, die Verhaltensaspekte möglichst eindeutig aus einer erweiterten Spezifikation der Datenstrukturen abzuleiten. Traditionelle Methoden der Datenmodellierung können die dynamischen Aspekte der Daten jedoch nur unzureichend erfassen, da zwar die zu jedem Zeitpunkt gültige Struktur der Daten abgebildet wird, nicht jedoch ihr Verhalten. Der immense Datendurchsatz im Data Warehouse sowie seine starke Einbindung in traditionell realisierte Administrations- und Auswertungssysteme (z.B. über SQL-Schnittstellen) verhindern jedoch eine naheliegende objektorientierte Realisierung.

Aus den genannten Gründen müssen die traditionellen Methoden der Datenmodellierung um zusätzliche Beschreibungsmittel erweitern werden, mit deren Hilfe das für die Spezifikation der Warehouse-Funktionalität notwendige Verhalten der Daten beschrieben werden kann. Objekttypen, Beziehungstypen, Attribute und Abstraktionsbeziehungen (Assoziation, Generalisierung, Aggregation) zwischen Objekttypen stellen die traditionellen Beschreibungsmittel der Datenmo-

dellierung dar. Als erweiterte Beschreibungsmittel zur Spezifikation des Verhaltens kommen insbesondere *Existenzabhängigkeiten*¹²⁾ und *Ableitungsregeln*¹³⁾ in Betracht.

Existenzabhängigkeiten implizieren die schemaweite Propagierung von Instanzlöschungen und -einfügungen, Ableitungsregeln implizieren dagegen die schemaweite Propagierung von Attributwertänderungen. Stehen die durch die Existenzabhängigkeit verknüpften Objekttypen in einer Assoziations- oder Generalisierungsbeziehung zueinander, müssen Löschungen durch abhängige Löschungen und (unter bestimmten Bedingungen) Einfügungen durch abhängige Einfügungen propagiert werden. Stehen die Objekttypen dagegen in einer Assoziationsbeziehung zueinander, ist die Propagierung durch abhängige Änderungen der Regelfall. Nur unter bestimmten Bedingungen werden hierbei Löschungen durch abhängige Löschungen und Einfügungen durch abhängige Einfügungen propagiert. Unabhängig von der zugrunde liegenden Abstraktionsart sind Änderungen grundsätzlich durch abhängige Änderungen zu propagieren¹⁴⁾

Existenzabhängigkeiten und Ableitungsregeln stellen somit wesentliche Darstellungsmittel für ein Warehouse-Schema dar. Sie lassen sich sogar formal derart spezifizieren, daß aus ihnen automatisch entsprechende Propagierungstrigger einer aktiven Datenbank generiert werden können¹⁵⁾.

3 Istanalyse

3.1 Aufgaben der Istanalyse

In der Phase "Istanalyse" wird der Istzustand des Problembereichs, für den ein (Anwendungs-)System entwickelt werden soll, erhoben, aufbereitet und kritisch analysiert. Im Rahmen einer Data Warehouse-Entwicklung bildet ein Unternehmensbereich oder sogar das gesamte Unternehmen einschließlich seiner Anwendungssystemlandschaft den zu betrachtenden Problembereich. Die Komplexität eines solchen Problembereichs erfordert eine starke Zielorientierung bei der Systemabgrenzung, -erhebung und -beschreibung. Außerdem treten Schwachstellenanalysen zur Vorbereitung der Zieldefinition in den Hintergrund, da ein Data Warehouse nicht auf eine konkrete Problemlösung abzielt.

12) Vgl. Scheuermann et al. (1980).

13) Vgl. Hammer et al. (1981).

14) Vgl. Winter (1994).

15) Vgl. Winter (1994).

Aus der Struktur eines Data Warehouse-Systems ergeben sich zwei Kernfragen, die in der Istanalyse zu beantworten sind:

1. Welche manuellen und computerunterstützten Auswertungen bzw. Analysen werden in dem Unternehmensbereich bzw. in dem Unternehmen zur Zeit ~~stellt~~ erstellt?
2. Welche Daten/Datenquellen werden in welcher Form für die bestehenden Auswertungen genutzt und welche weiteren Daten/Datenquellen können für Auswertungen genutzt ~~wen~~ werden?

Die erste Frage bezieht sich auf den Datenauswertungsteil des Warehouse-Systems, die zweite auf den Datenextraktionsteil.

Die Beantwortung der zweiten Frage beinhaltet auch, daß die Interoperabilität der zu den Daten bzw. Datenquellen gehörenden Anwendungssysteme im Detail untersucht wird und eine aktuelle Dokumentation der Datenquellen auf der logischen und darauf aufbauend eine integrierte Dokumentation auf der konzeptionellen Ebene erstellt wird. Insbesondere sind die zeitlichen Bezüge der Daten zu ermitteln und zu dokumentieren, da dieser Zeitbezug ein wesentliches Auswertungskriterium - d.h. eine Auswertungsdimension - darstellt.

Zu beachten ist, daß aus technischen oder organisatorischen Gründen ein direkter Zugriff auf die Datenbanken bzw. Dateien der Anwendungssysteme häufig nicht möglich oder nicht erwünscht ist. In diesem Fall müssen bestehende Schnittstellen zwischen den Systemen zur Versorgung des Data Warehouse genutzt werden.

Der für die Beantwortung der beiden Kernfragen erforderliche Aufwand wird wesentlich durch die Anzahl, die Struktur und die (vorhandene) Dokumentation einerseits der Datenquellen und andererseits der Auswertungen bestimmt. Die historisch gewachsenen Anwendungssystemlandschaften und die damit verbundene Heterogenität, Inkonsistenz und Undokumentiertheit der Daten und Auswertungen erfordern im allgemeinen ein aufwendiges iteratives Vorgehen. Es besteht darin, daß die Projektmitglieder die entsprechenden Personen befragen, deren Antworten verarbeiten und wiederum dann durch Rückfragen die offenen Punkte klären. Dadurch erschließt sich der Problembereich mit den zugehörigen Datenquellen und Auswertungen sukzessiv von einer abstrakten bis zu einer sehr detaillierten Ebene.

Die Beantwortung der beiden Kernfragen kann zunächst parallel erfolgen. Methodisch sind die beiden zugehörigen Teilaufgaben der Istanalyse in das Reverse Engineering einzuordnen. Dieses beschäftigt sich mit der sukzessiven Nachdokumentation von Anwendungssystemen und Daten

auf den verschiedenen Abstraktionsebenen¹⁶⁾. Als Ergebnis der Teilaufgaben ergeben sich zwei konzeptionelle Datenschemata; das eine wird aus den Auswertungen, das andere aus der logischen Struktur der Datenquellen abgeleitet.

Unterschiedliche Datenquellen bzw. Datenbasen unterschiedlicher Anwendungssysteme können grundsätzlich getrennt voneinander betrachtet werden. Auf diese Weise kann die Analyse der Datenquellen durch den Entwurf entsprechender Teilschemata vereinfacht sowie auch weiter parallelisiert werden. Die Teilschemata sind anschließend zu dem konzeptionellen Gesamtschema der Datenquellen zu integrieren.

Im letzten Schritt der Istanalyse sind die beiden entstandenen Schemata zu integrieren. Das sich ergebende Gesamtschema und die dokumentierten logischen Verbindungen zwischen den Komponenten des Schemas und den Datenquellen bildet das Ergebnis der Istanalyse.

Bei der Durchführung der beiden Schemaintegrationsprozesse kann jeweils auf Methoden aus dem Bereich der View-/Schemaintegration zurückgegriffen werden¹⁷⁾. Wenn die (DV-technischen) Verbindungen zwischen den bestehenden Auswertungen und den zugehörigen Datenquellen bekannt sind, kann die Integration des Schemas der Auswertungen und des Gesamtschemas der Datenquellen dadurch vereinfacht werden, daß die Verbindungen auf der Ebene des DV-Konzepts in Verbindungen zwischen den Schemata umgesetzt werden.

Vielfach sollen die bestehenden Auswertungen auch nach der Einführung eines Data Warehouse in derselben Form zur Verfügung stehen. In diesem Fall muß neben der konzeptionellen Struktur der Auswertungen auch ihr Layout in das Sollkonzept eingehen. Für die Istanalyse bedeutet dies, daß das Layout der Auswertungen ebenfalls zu analysieren und in geeigneter Form zu dokumentieren ist.

3.2 Projekterfahrungen

Im Projekt wurde zu Beginn der Istanalyse und quasi als Einstiegspunkt für weitere Analysetätigkeiten von den betroffenen Fachabteilungen jeweils eine Dokumentation ihrer Systeme sowie ihrer Auswertungen erstellt. Zum Einsatz kommt im Produktionsbereich des Unternehmens eine Reihe unterschiedlicher Anwendungssysteme: Das Spektrum reicht vom SAP R/2 Modul "RV" über ein ereignisgesteuertes System für die hierarchische Produktionsprogrammplanung sowie Fertigungs- und Montageleitstände bis hin zu BDE(Betriebsdatenerfassung)-Terminals. Der

16) Vgl. z.B. Baumöl et al. (1996).

17) Vgl. z.B. Mannino et al. (1988), Batini et al. (1986), Navathe et al. (1986).

„Datenverkehr“ zwischen den verschiedenen Systemen wird hauptsächlich nachts durch Transfer entsprechender Dateien durchgeführt.

Die (Erst-)Dokumentation stellte - wie sich im Laufe der Istanalyse ergab - bereits eine weitestgehend vollständige Auflistung der relevanten Datenquellen und Auswertungen dar. Allerdings war diese Dokumentation sehr unstrukturiert und heterogen. Insbesondere fehlten völlig Informationen über die logischen Beziehungen zwischen den verschiedenen Datenquellen. Außerdem erwiesen sich viele Informationen als nicht mehr aktuell; eine Reihe von Dokumenten war bereits mehr als fünf Jahre nicht aktualisiert worden.

In dem iterativen Analyseprozeß, welcher von Interviews in Fachabteilungen getragen wurde, entstand sukzessiv ein konzeptionelles Schema einerseits der Auswertungen, andererseits der Datenquellen. Hierbei mußte der Weg von den Datenquellen hin zu den Auswertungen völlig neu erschlossen werden. Erschwert wurde diese Aufgabe auf der Auswertungsseite durch die eher anwender- und weniger entwicklerorientierte Dokumentation, die lediglich lokale Anwendungssichten auf die Teilsysteme beschrieb.

Abbildung 3 zeigt einen kleinen Ausschnitt des sich ergebenden Gesamtschemas des Ist-Modells; insgesamt umfaßt das Schema mehr als 120 Objekttypen und weit über 300 Beziehungstypen. In einem Rechteck liegende Rechtecke stellen Spezialisierungen des Objekttyps dar, der durch das äußere Rechteck repräsentiert wird.

Um eine ausreichende Lesbarkeit des Schemas zu erreichen, wurden die Anordnungsregeln des SER-Modells¹⁸⁾ genutzt. Darüber hinaus kennzeichnen Farbcodes die Herkunft der Objekttypen; beispielsweise wurden die Objekttypen des PPS-Systems grün und die Objekttypen der Fertigungs- und Montageleitstände blau eingezeichnet.

Nicht abbildbare Informationen, wie zum Beispiel die Datenherkunft der Objekttypen (in Gestalt ihrer Datenquelle) und ihrer Attribute (in Gestalt des zugehörigen Datenfelds), wurden in zuvor festgelegter Form in die „Description“ der zugehörigen Komponente des Schemas eingetragen.

4 Entwicklung des Sollkonzepts

Gegenstand der Phase *Sollkonzept* ist, aufbauend auf der Istanalyse, die Zieldefinition des zu entwickelnden Anwendungssystems. Für ein Data Warehouse sind einerseits die fachlich-

18) Vgl. Sinz (1987).

betriebswirtschaftlichen Anforderungen, andererseits die technischen Anforderungen festzulegen.

4.1 Fachlich-betriebswirtschaftliches Sollkonzept

Produktionscontrolling als fachlich-betriebswirtschaftliche Domäne

Die fachlich-betriebswirtschaftliche Zieldefinition ergibt sich aus dem Einsatzgebiet des Data Warehouse; im vorliegenden Fall handelt es sich um das Produktionscontrolling. Die im folgenden dargestellten Überlegungen sind aber generell auch auf andere Controllingbereiche übertragbar.

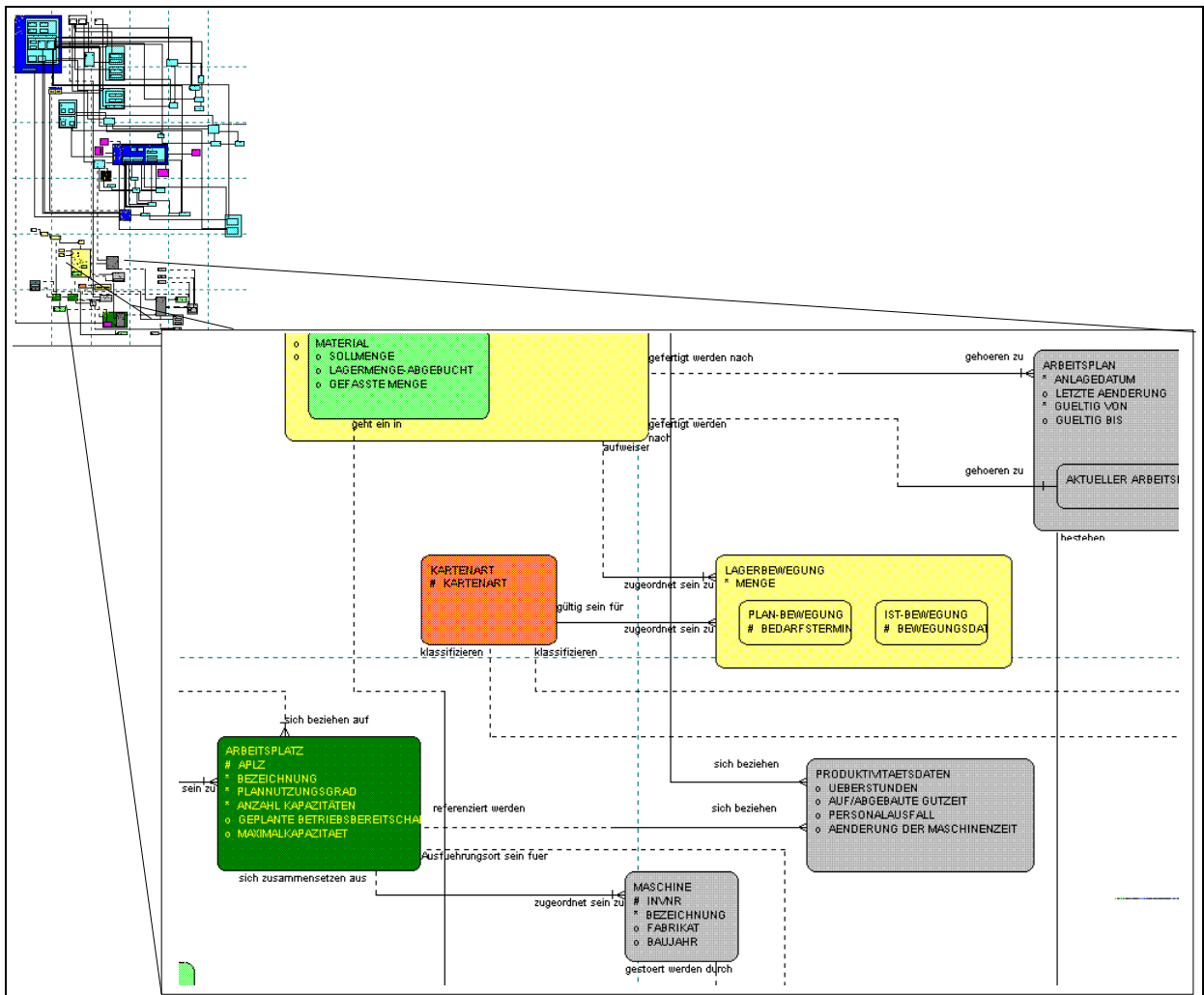


Abb. 3: Ausschnitt aus dem Ist-Modell

Heinen definiert "Controlling" aufgabenbezogen als die "Schaffung und laufende Anwendung eines institutionalisierten Planungs- und Kontrollsystems für die Gesamtunternehmung"¹⁹⁾. Das

19) Heinen (1991), S. 66.

Produktionscontrolling hat im Rahmen dieses Systems die Aufgabe, die Wirtschaftlichkeit des Produktionsprozesses sicherzustellen. Ein Data Warehouse, das im Produktionscontrolling eingesetzt werden soll, ist folglich ein Hilfsmittel des Planungs- und Kontroll(sub)systems zur Sicherung der Wirtschaftlichkeit der Produktion. Darüber hinaus muß es dazu in der Lage sein, verdichtete Daten aus dem Produktionsbereich für eine unternehmensweite Abstimmung (durch das übergeordnete Planungs- und Kontrollsystem) bereitzustellen.

Wie andere Controllingbereiche auch²⁰⁾, nutzt das Produktionscontrolling sowohl zur bereichsinternen Informationsversorgung wie auch zur Berichterstattung an übergeordnete Stellen sogenannte Kennzahlen. Kennzahlen sind Zahlen, die quantitativ erfaßbare Sachverhalte in konzentrierter Form darstellen. Für das Produktionscontrolling sind folgende Kennzahlenkategorien von Bedeutung:

- Kennzahlen über Produktionskosten (z.B. Betriebsunterbrechungskosten, Einzellohnkosten, Einzelmaterialkosten, Ausschußquote, Materialabfallquote);
- Kennzahlen über Kapazitätsauslastungen (z.B. technische bzw. personelle Kapazitätsauslastung, Überstundenquote);
- Kennzahlen über Zeiten und Termine (z.B. Durchlaufzeiten, Terminüber- und -unterschreitungen).

Die einzelnen Kennzahlen können arbeitsplatz-, kostenstellen- oder bereichsbezogen gebildet werden. Weitere Bezugsobjekte der Kennzahlenanalyse können (z.B. bei der Ermittlung von Einzelkosten) Produktvarianten und Zwischenprodukte sein. Im Rahmen der Plankostenrechnung werden zusätzlich kostenstellenbezogene Soll-/Ist-Abweichungen ermittelt und auf ihre Ursachen (z.B. Mengenabweichung, Preisabweichung) zurückgeführt.

Festlegung und Validierung des fachlich-betriebswirtschaftlichen Sollkonzepts

Der Umfang des Sollkonzepts für ein Data Warehouse wird nicht nur durch generelle Projektrestriktionen (z.B. Budget, Zeitrahmen) beschränkt, sondern ist auch an den Umfang, die Genauigkeit und die Aktualität der nutzbaren Daten gebunden. Unter Umständen können nicht alle wünschenswerten Funktionen unterstützt werden, weil die jeweils notwendigen Daten nicht zur Verfügung stehen. Für die betrachtete Domäne kann deshalb im allgemeinen nicht der gesamte Umfang an Auswertungen bzw. Kennzahlenberechnungen realisiert werden.

20) Vgl. z.B. Reichmann (1995).

In der Istanalyse des Projekts stellte sich beispielsweise heraus, daß Lohnscheine im Unternehmen teilweise nicht direkt nach der Fertigstellung eines Auftrags, sondern verspätet abgegeben werden. Entsprechend macht es keinen Sinn, von dieser Rückmeldung abgeleitete schichtgenaue Kennzahlen ins Sollkonzept aufzunehmen.

4.2 Technisches Sollkonzept

Das technische Sollkonzept des Data Warehouse ergibt sich aus der Klärung zweier zentraler Fragen:

- Welches Datenbankmanagementsystem (DBMS) soll genutzt werden?
- Welche(s) Auswertungswerkzeug(e) soll(en) verwendet werden?

Zur Speicherung von und zum Zugriff auf Warehouse-Daten können multidimensionale oder relationale DBMS genutzt werden: *Multidimensionale DBMS* wurden speziell für die Speicherung Hypercube-artiger Datenstrukturen entwickelt und können die interaktive Auswertung (z.B. Verdichtung) entsprechender Datenstrukturen hocheffizient durchführen. Allerdings ist die Umsetzung der Input-Daten in die Hypercube-artigen Datenstrukturen sehr aufwendig. Außerdem handelt es sich bei multidimensionalen DBMS um proprietäre und spezialisierte Softwaresysteme. Der Zugriff auf die in ihnen gespeicherten Daten ist meist nur über die jeweilige zugehörige Auswertungssoftware möglich, die die Daten ausschließlich in der durch die Hypercube-Struktur vorgegebenen Weise auswerten kann.

Daten eines Data Warehouse können auch mit Hilfe *relationaler DBMS* gespeichert werden. Allerdings können relationale DBMS spezielle Warehouse-Auswertungen nur vergleichsweise ineffizient durchführen, da sie auf die Unterstützung der konventionellen Anwendungssysteme ausgerichtet sind. Auf der anderen Seite erlaubt die Nutzung eines relationalen DBMS eine "offene" Realisierung mit Hilfe marktgängiger, weitgehend standardisierter Softwarewerkzeuge. Darüber hinaus bemühen sich die Anbieter relationaler DBMS, Warehouse-Anwendungen durch größere Hauptspeicher-Adreßräume, Bitlisten-Indizes, effizientere Implementierung von Joins großer Tabellen usw. besser zu unterstützen.

Mehrdimensionale DBMS sollten aus den genannten Gründen nur dort Verwendung finden, wo ihre Vorzüge durch Beschränkung auf Hypercube-Auswertungen voll zum Tragen kommen. In allen anderen Fällen (und damit auch im hier beschriebenen Anwendungsfall) ist die Nutzung relationaler DBMS vorteilhaft bzw. können überhaupt nur relationale DBMS genutzt werden.

Ähnliche Kriterien wie die Auswahl des DBMS bestimmen die Auswahl der Auswertungswerkzeuge: Falls ausschließlich festgelegte, standardisierte Auswertungen durchge-

führt werden sollen, kommt die Verwendung entsprechender proprietärer Werkzeuge (z.B. Oracle Express, STG MetaCube Explorer) in Betracht, die jeweils für ein bestimmtes multidimensionales DBMS angeboten werden (Oracle Express Server bzw. STG MetaCube). In allen anderen Fällen muß auf allgemein verwendbare Endbenutzerwerkzeuge wie z.B. Microsoft Excel bzw. Access oder Oracle Data Query zurückgegriffen werden. Diese Werkzeuge erfordern zwar aufgrund ihrer fehlenden Spezialisierung einen vergleichsweise höheren Aufwand zur Lösung von Warehouse-Problemstellungen. Die Werkzeuge sind jedoch flexibel und interoperabel (z.B. durch ODBC).

Im hier beschriebenen Projekt wurde mit dem Oracle Server ein relationales DBMS benutzt, auf das über ODBC mit Microsoft Access zugegriffen werden konnte.

5 Konzeptioneller Entwurf

Ein Teil des konzeptionellen Entwurfs fällt bei der hier beschriebenen Vorgehensweise zur Realisierung eines Data Warehouse bereits in die Istanalyse: Die dort entwickelten konzeptionellen Schemata können in der Entwurfsphase direkt übernommen und an das Sollkonzept angepaßt werden. Die Anpassung erfolgt primär im Hinblick auf die uniforme Darstellung des Zeitbezugs, die Spezifikation der Ladevorgänge (Inflow) sowie den Detailentwurf für den Auswertungsbereich und die Spezifikation von Aktualisierungsstrategien.

5.1 Uniforme Darstellung des Zeitbezugs

Die meisten Objekttypen haben einen zeitlichen Bezug, der im Schema modelliert werden muß. Beispielsweise bezieht sich eine geplante Produktionsmenge auf einen Zeitraum, eine Fertigmeldung aus der Betriebsdatenerfassung auf einen Zeitpunkt. Der Zeitbezug eines Objekttyps wurde im Projekt mittels Referenzierung des Objekttyps "Zeitpunkt" oder "Zeitraum" abgeleitet.

Als Problem ergibt sich, daß Zeitangaben nicht immer sekundengenau spezifiziert sind. In der betrieblichen Realität sind beispielsweise häufig Zeitangaben wie Kalenderwoche und -tag anzutreffen. Im Projekt wurde das Problem dadurch gelöst, daß Zeitangaben geeignet transformiert wurden. So wurde etwa eine Kalenderwoche in den Zeitraum von "der ersten Sekunde der ersten Stunde des ersten Tages bis zur letzten Sekunde der letzten Stunde des letzten Tages" (der Kalenderwoche) transformiert. Um die Information zu erhalten, daß es sich um einen grob spezifizierten Zeitraum mit einem bestimmten Namen handelt, wurde der zur Zuordnung verwendete Beziehungstyp entsprechend benannt (im Beispiel: "Kalenderwoche").

5.2 Spezifikation der Ladevorgänge

Der Inflow des Data Warehouse, d.h. das Laden von Daten operativer Systeme in das Warehouse, wird durch spezielle Importprozeduren realisiert. Im Ladebereich des Data Warehouse werden relevante Transferdateien unmittelbar in entsprechende Relationen umgesetzt. Weitere Relationen entstehen durch die Replikation von Quelldaten aus relationalen DBMS im Warehouse. Die Ladedaten werden bei Aktualisierungen vollständig überschrieben.

Um in allgemeiner Form festlegen zu können, wie die Basisdaten eines Data Warehouse durch Daten aus dem Ladebereich zu aktualisieren sind, können Ladevorgänge entsprechend dem in Abbildung 4 dargestellten Entscheidungsbaum klassifiziert werden: Ladedaten umfassen entweder "neue" Bewegungsdaten ("net-change") oder den gesamten aktuellen Datenumfang ("regenerativ", "Vollumfang"). Beispielsweise liegen Lagerbestände in einem Unternehmen wöchentlich in Form von Lagerzugängen bzw. -abgängen der letzten Woche, in einem anderen Unternehmen als aktueller Gesamtbestand des Lagers vor.

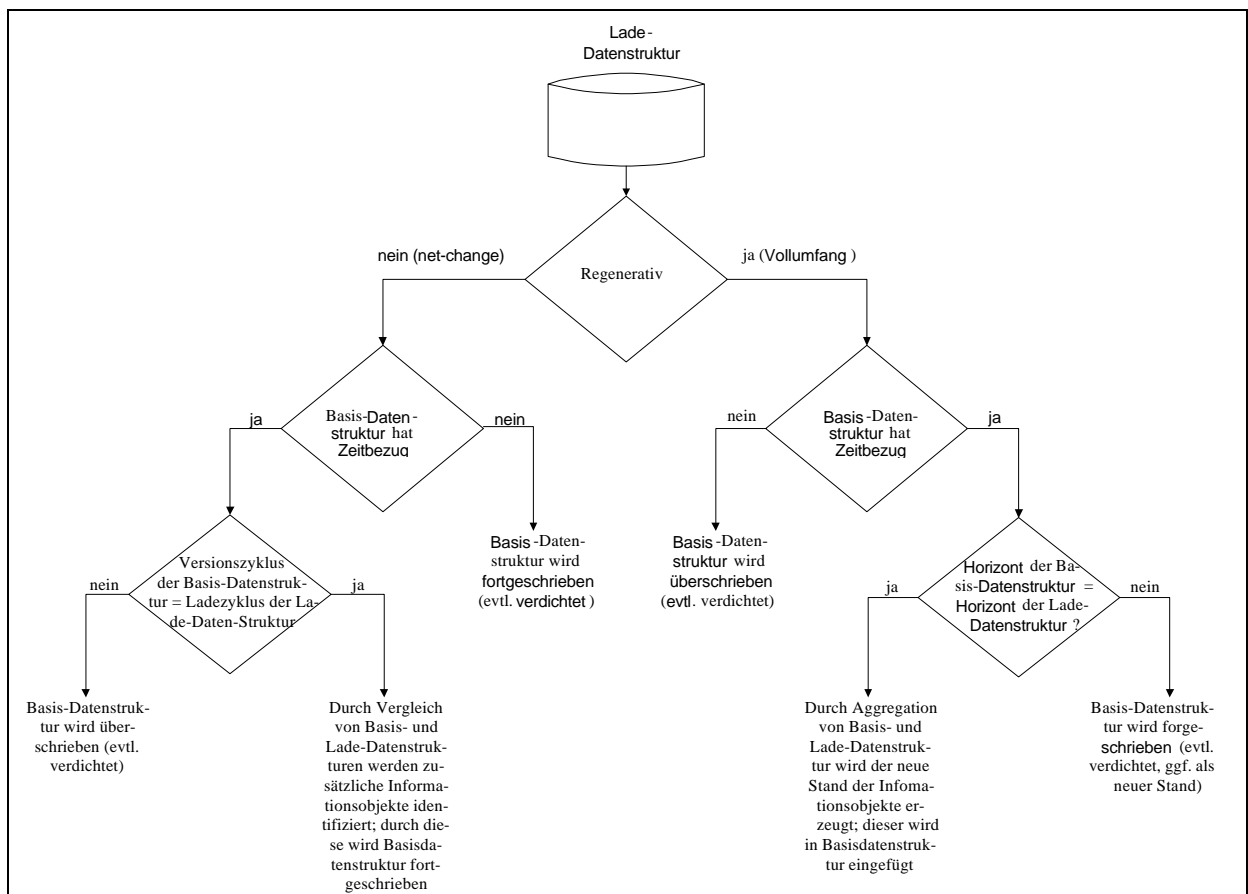


Abb. 4: Entscheidungsbaum zur Ableitung der Klasse des Ladevorgangs

Basisdaten können hinsichtlich ihres Zeitbezugs klassifiziert werden. Beispielsweise bezieht sich ein Lagerbestand entweder auf einen bestimmten Zeitpunkt, oder er hat als "aktueller Lagerbestand" keinen expliziten Zeitbezug. Für Bewegungsdaten mit Zeitbezug ist zu prüfen, ob Versi-

ons- und Ladezyklus übereinstimmen. Zum Beispiel können Lagerzugänge bzw. -abgänge zwar täglich aus dem Materialwirtschaftssystem übernommen, werden aber vielleicht im Data Warehouse nur mit Wochenbezug archiviert.

Für regenerative Daten mit Zeitbezug ist zu prüfen, ob die Zeitbezüge für Lade- und Basisdaten übereinstimmen. Etwa richtet sich die Form der Archivierung (Überschreiben oder Ergänzen) des Fertigmeldungsbestands eines bestimmten Tages danach, ob im Warehouse Fertigmeldungen mit Tages- oder mit Wochenbezug gespeichert werden.

5.3 Detailentwurf für den Auswertungsbereich und Spezifikation von Aktualisierungsstrategien

Eine wesentliche Entwurfsentscheidung beinhaltet die Frage, ob Objekttypen des Schemas in physische Tabellen oder in Sichten umzusetzen sind. Abgeleitete Daten sollten als Sichten implementiert werden, da die Konsistenzsicherung wesentlich einfacher zu gestalten ist. Allerdings muß für abgeleitete Daten immer dann eine physische Tabelle angelegt werden, wenn sie originale Attribute besitzen (z.B. Zeitangaben). In solchen Fällen kann, wie auch bei zu erwartendem häufigen Zugriff, eine Implementierung in Form einer abgeleiteten Tabelle erfolgen. Bei geringem Aktualitätsbedarf bietet sich ein Snapshot an, bei Online-Aktualitätsbedarf dagegen eine materialisierte Sicht, deren Inhalte durch integritätserhaltende Datenbanktriggern²¹⁾ konsistent gehalten werden.

Sofern im Sollkonzept für bestimmte Kennzahlen Längsschnittanalysen spezifiziert wurden, sind die zu ihrer Berechnung verwendeten Informationsobjekte daraufhin zu untersuchen, ob und für welchen Zeitraum historische Daten vorgehalten werden müssen. Darüber hinaus können auch für Kennzahlen - dem Data Warehouse-Konzept folgend - Vorverdichtungen festgelegt werden. Wenn beispielsweise die Gesamtdurchlaufzeit der verschiedenen Produktvarianten in vielen Auswertungen benötigt wird, sollte (auch aus Laufzeitgründen) eine entsprechende Vorverdichtung der Durchlaufzeiten in den einzelnen Kostenstellen als Datenobjekt modelliert werden.

Schließlich muß im Rahmen des Entwurfs festgelegt werden, ob eine Aktualisierung von Ladedaten über die Basisdaten vollständig in alle Auswertungsdaten propagiert werden soll ("Push") oder ob erst bei einem Zugriff auf Auswertungsdaten eine Aktualisierung erfolgen soll ("Pull"). Zur Auswahl einer Aktualisierungsstrategie können zwei Kriterien herangezogen werden:

21) Vgl. Winter (1994).

- *Zugriffshäufigkeit*: Bei häufigem Zugriff sollten alle Daten entlang des Propagierungspfads²²⁾ "Push"-aktualisiert werden; ansonsten reicht die "Pull"-Strategie aus.
- *Ermittlungsaufwand*: Wenn ein Datenobjekt durch komplexe und zeitaufwendige Datenbankoperationen (Gruppierung, komplexe Joins) ermittelt wird, sollte es nach der "Pull"-Strategie aktualisiert werden; ansonsten kann die "Push"-Strategie verfolgt werden.

6 Realisierung

Die Unterschiede zwischen einer Data Warehouse-Realisierung und der Realisierung eines konventionellen Anwendungssystems resultieren hauptsächlich daraus, daß die Warehouse-Datenbank verschiedene Ableitungsstufen von Daten umfaßt und auf regelmäßige, umfangreiche Aktualisierungen aus operativen Systemen angewiesen ist. Mit jeder Datenaktualisierung ist eine Vielzahl von Propagierungen (z.B. Überschreiben bestehender Informationen, Fortschreibung von Historien, Ergänzung neuer Informationen) über mehrere Stufen hinweg verbunden.

Zur Realisierung der dynamischen Konsistenzsicherung bieten sich Triggermechanismen an, wie sie von marktgängigen Datenbanksystemen bereitgestellt werden: Jede Lade-Datenstruktur wird mit Datenbanktriggern versehen. Die Trigger propagieren entsprechend der Existenzabhängigkeiten und Ableitungsregeln des konzeptionellen Schemas Einfügungen, Änderungen und Löschungen von Instanzen konsistent in die Basisdaten. Analog werden Basis-Datenstrukturen mit Datenbanktriggern versehen, die Transaktionen konsistent in die Auswertungsdaten propagieren.

Die Vielzahl und Komplexität der in einem Data Warehouse zu spezifizierenden und zu implementierenden Datenstrukturen legt die Nutzung einer leistungsfähigen CASE-Umgebung nahe. Im hier beschriebenen Projekt wurde die CASE-Umgebung Designer/2000 von Oracle genutzt.

In Designer/2000 können Existenzabhängigkeiten auf Seiten des abhängigen Entitätstyps grundsätzlich durch Aufnahme der Beziehungstypen zu dem/den jeweiligen unabhängigen Entitätstyp(en) in den Primärschlüssel abgebildet werden. Ableitungsregeln werden als spezielle Eigenschaften (abgeleiteter) Attribute erfaßt. Allerdings kann durch die Eigenschaften die implizierte Semantik der Informationsableitung nicht vollständig erfaßt werden:

22) Vgl. Scheuermann et al. (1980).

- Generalisierungsbeziehungen sind immer disjunkt und vollständig, so daß die drei anderen Fälle (disjunkt/partiell, nicht disjunkt/vollständig und nicht disjunkt/partiell) nicht abgebildet und damit auch die zugehörigen Propagierungen nicht generiert werden können.
- Aggregationsbeziehungen (z.B. Verdichtungen) können überhaupt nicht abgebildet werden. Dieser Mangel hat für die Modellierung eines Data Warehouse entscheidende Bedeutung, da eine seiner Hauptfunktionen in der Vorverdichtung von Ladedaten und der Bereitstellung verdichteter abgeleiteter Daten liegt.
- Assoziationsbeziehungen (z.B. Verknüpfungen) führen beim abgeleiteten Entitätstyp immer zu einer Verkettung aller Primärschlüssel der verknüpften Entitätstypen. Das bedeutet, daß z.B. Equi-Joins, deren Joinbedingung sich auf ein Schlüsselattribut bezieht (d.h. der Standardfall abgeleiteter Daten), nicht konzeptionell abgebildet werden können. Ebenso können die Verknüpfungsbedingungen von Joins nicht direkt spezifiziert werden. Als Folge können die jeweiligen Propagierungen nicht vollständig generiert werden.

Damit die mit den graphischen Werkzeugen der CASE-Umgebung nicht modellierbare Semantik im Repository erfaßt werden konnte, wurden verschiedene Konventionen festgelegt. Sie spezifizieren z.B., mit welchen Schlüsselwörtern, in welcher Form und in welchem Repository-Objekt Joinbedingungen, Beschreibungen von Aggregationsregeln oder zusätzliche Eigenschaften von Generalisierungsbeziehungen abzuspeichern sind. Die Zusatzinformationen werden von speziellen, selbstentwickelten Generatoren dazu benutzt, in Ergänzung der von der CASE-Umgebung generierten Datenbankobjekte (Indizes und referentielle Integritätsbedingungen) Snapshotdefinitionen, Sichtendefinitionen und insbesondere Datenbanktrigger zu erzeugen.

Auf Grundlage des in Abschnitt 5.2 beschriebenen Entscheidungsbaums können im Prinzip auch Importprozeduren durch Repository-Ergänzungen spezifiziert und durch dedizierte Generatoren automatisch erzeugt werden. Im Vergleich zu "datenbankinternen" Propagierungen sind Importprozeduren jedoch bisher weit weniger ausführlich untersucht worden; eine abgeschlossene Systematik des implizierten Verhaltens liegt noch nicht vor. Im Projekt wurden deshalb die Importprozeduren traditionell entworfen und implementiert.

Die im Sollkonzept vorgesehenen Standardauswertungen konnten mit Hilfe von Microsoft Access schnell und mit vergleichsweise geringem Aufwand realisiert werden. Für individuelle Auswertungen können die Endbenutzer Oracle Data Query nutzen.

Literatur

- Alur, N. (1995): Missing Links in Data Warehouse, in: *Database Programming & Design* 9, Jg. 8, S. 21-23.
- Andersen Consulting (1996): Globales Informationsmanagement dank Data Warehouse und C/S-Middleware, in: *Datenbank Fokus* 2, S. 39-43.
- Batini, C./Lenzerini, M./Navathe, S.B. (1986): A Comparative Analysis of Methodologies for Database Integration, in: *ACM Computing Surveys* 4, Jg. 18, S. 323-364.
- Baumöl, U./Borchers, J./Eicker, S./Hildebrand, K./Jung, R./Lehner, F. (1996): Einordnung und Terminologie des Software Reengineering, in: *Informatik-Spektrum* 4, Jg. 19, S. 191-195.
- Bulos, D. (1996): A New Dimension, in: *Database Programming & Design* 6, Jg. 9, S. 33-37.
- Celko, J./McDonald, J. (1995): Don't Warehouse Dirty Data, in: *Datamation* 15, Jg. 41, S. 42-52.
- Hammer, M./McLeod, D. (1981): Database Description with SDM - A Semantic Database Model, in: *ACM Transactions on Database Systems* 2, Jg. 6, S. 351-386.
- Heinen, E. (1991): Industriebetriebslehre. Entscheidungen im Industriebetrieb, 9. Aufl., Wiesbaden.
- Inmon, W.H. (1992): Building the Data Warehouse, Wellesley, MA.
- Inmon, W.H. (1990): Using ORACLE to Build Decision Support Systems, London u.a..
- I/S Analyzer (1994): How to Build and to Design Data Warehouse to Improve Information Access, in: *I/S Analyzer* 11, Jg. 32.
- Kelly, S. (1994): Data Warehousing, New York.
- Maloney, T.F. (1995): Brave New Architecture, in: *Database Programming and Design* 11, Jg. 8, S. 22-27.
- Menninger, D. (1996): Designing a Database for OLAP, in: *Oracle Magazine*, Jg. 10, Nr. 2, S. 83-86.
- Moriarty, T. (1995): Modeling Data Warehouses, in: *Database Programming and Design* 8, Jg. 8, S. 61-63.
- Mannino, M.V./Navathe, S.B./Effelsberg, W. (1988): A Rule-Based Approach For Merging Generalization Hierarchies, in: *Information Systems* 3, Jg. 13, S. 257-272.
- Meredith, M.E./Khader, A. (1996): Designing Large Warehouses, in: *Database Programming & Design* 6, Jg. 9, S. 25-30.
- Navathe, S.B./Elmasri, R./Larson, J. (1986): Integration User Views in Database Design, in: *IEEE Computer* 1, Jg. 19, S. 50-62.
- o.V. (1996): Overlapping Dimensions, in: *Oracle Magazine*, Jg. 10, Nr. 2, S.42.

- Parsaye, K. (1995): The Sandwich Paradigm, in *Database Programming & Design* 4, Jg. 8, S. 50-55.
- Poe, V. (1994): Clear, Careful, and Realistic: Guidelines for Warehouse Development, in: *Database Programming & Design* 9, Jg. 7, S. 60-64.
- Rauh, O./Stickel E. (1993): Searching for Compositions in ER Schemes, in: *Proc. of the 12th Int. Conference on Entity Relationship Approach*, Arlington, S. 7586.
- Reichmann, Th. (1995): Controlling mit Kennzahlen und Managementberichten. 4. Aufl., München.
- Rohrig, N. (1987): Copy- und Extrakt-Management, in *Information Management* 3, Jg. 2, S. 49-55.
- Rother, G. (1995): Data Warehouse: Die neue Art, Daten zu lagern, in: *Diebold Management Report* 8/9, S. 3-7.
- Scheuermann, P./Schiffner, G./Weber, H. (1980): Abstraction Capabilities and Invariant Properties Modeling within the Entity-Relationship-Approach, in: Chen, P.P.-S. (Hrsg.): *Entity-Relationship Approach to Systems Analysis and Design*, Amsterdam, S. 121-140.
- Sinz, E.J. (1987): Datenmodellierung betrieblicher Probleme und ihre Unterstützung durch ein wissensbasiertes Entwicklungssystem. Habilitationsschrift, Universität Regensburg.
- Stodder, D./Pucky, A. (1995): Software AG's Data Warehouse Push, in: *Database Programming & Design* 7, Jg. 8, S. 68-69.
- Weldon, J.-L. (1995): Managing Multidimensional Data: Harnessing The Power, in: *Database Programming and Design* 8, Jg. 8, S. 24-33.
- White, C. (1995): The Key to a Data Warehouse, in: *Database Programming & Design* 2, Jg. 8, S. 23 - 25.
- Winter, R. (1994): Formalised Conceptual Models as a Foundation for Information Systems Development, in: Loucopoulos, P. (Hrsg.) *Entity-Relationship Approach - ER'94*, Berlin u.a., S. 437-455.

Arbeitsberichte des Instituts für Wirtschaftsinformatik

- Nr. 1 Bolte, Ch., Kurbel, K., Moazzami, M., Pietsch, W.: Erfahrungen bei der Entwicklung eines Informationssystems auf RDBMS- und 4GL-Basis; Februar 1991.
- Nr. 2 Kurbel, K.: Das technologische Umfeld der Informationsverarbeitung - Ein subjektiver 'State of the Art'-Report über Hardware, Software und Paradigmen; März 1991.
- Nr. 3 Kurbel, K.: CA-Techniken und CIM; Mai 1991.
- Nr. 4 Nietsch, M., Nietsch, T., Rautenstrauch, C., Rinschede, M., Siedentopf, J.: Anforderungen mittelständischer Industriebetriebe an einen elektronischen Leitstand - Ergebnisse einer Untersuchung bei zwölf Unternehmen; Juli 1991.
- Nr. 5 Becker, J., Prischmann, M.: Konnektionistische Modelle - Grundlagen und Konzepte; September 1991.
- Nr. 6 Grob, H. L.: Ein produktivitätsorientierter Ansatz zur Evaluierung von Beratungserfolgen; September 1991.
- Nr. 7 Becker, J.: CIM und Logistik; Oktober 1991.
- Nr. 8 Burgholz, M., Kurbel, K., Nietsch, Th., Rautenstrauch, C.: Erfahrungen bei der Entwicklung und Portierung eines elektronischen Leitstands; Januar 1992.
- Nr. 9 Becker, J., Prischmann, M.: Anwendung konnektionistischer Systeme; Februar 1992.
- Nr. 10 Becker, J.: Computer Integrated Manufacturing aus Sicht der Betriebswirtschaftslehre und der Wirtschaftsinformatik; April 1992.
- Nr. 11 Kurbel, K., Dornhoff, P.: A System for Case-Based Effort Estimation for Software-Development Projects; Juli 1992.
- Nr. 12 Dornhoff, P.: Aufwandsplanung zur Unterstützung des Managements von Softwareentwicklungsprojekten; August 1992.
- Nr. 13 Eicker, S., Schnieder, T.: Reengineering; August 1992.
- Nr. 14 Erkelenz, F.: KVD2 - Ein integriertes wissensbasiertes Modul zur Bemessung von Krankenhausverweildauern - Problemstellung, Konzeption und Realisierung; Dezember 1992.
- Nr. 15 Horster, B., Schneider, B., Siedentopf, J.: Kriterien zur Auswahl konnektionistischer Verfahren für betriebliche Probleme; März 1993.
- Nr. 16 Jung, R.: Wirtschaftlichkeitsfaktoren beim integrationsorientierten Reengineering: Verteilungsarchitektur und Integrationsschritte aus ökonomischer Sicht; Juli 1993.
- Nr. 17 Miller, C., Weiland, R.: Der Übergang von proprietären zu offenen Systemen aus Sicht der Transaktionskostentheorie; Juli 1993.
- Nr. 18 Becker, J., Rosemann, M.: Design for Logistics - Ein Beispiel für die logistikgerechte Gestaltung des Computer Integrated Manufacturing; Juli 1993.
- Nr. 19 Becker, J., Rosemann, M.: Informationswirtschaftliche Integrationsschwerpunkte innerhalb der logistischen Subsysteme - Ein Beitrag zu einem produktionsübergreifenden Verständnis von CIM; Juli 1993.
- Nr. 20 Becker, J.: Neue Verfahren der entwurfs- und konstruktionsbegleitenden Kalkulation und ihre Grenzen in der praktischen Anwendung; Juli 1993.

- Nr. 21 Becker, K., Prischmann, M.: VESKONN - Prototypische Umsetzung eines modularen Konzepts zur Konstruktionsunterstützung mit konnektionistischen Methoden; November 1993
- Nr. 22 Schneider, B.: Neuronale Netze für betriebliche Anwendungen: Anwendungspotentiale und existierende Systeme; November 1993.
- Nr. 23 Nietsch, T., Rautenstrauch, C., Rehfeldt, M., Rosemann, M., Turowski, K.: Ansätze für die Verbesserung von PPS-Systemen durch Fuzzy-Logik; Dezember 1993.
- Nr. 24 Nietsch, M., Rinschede, M., Rautenstrauch, C.: Werkzeuggestützte Individualisierung des objektorientierten Leitstands ooL; Dezember 1993.
- Nr. 25 Meckenstock, A., Unland, R., Zimmer, D.: Flexible Unterstützung kooperativer Entwurfsumgebungen durch einen Transaktions-Baukasten; Dezember 1993.
- Nr. 26 Grob, H. L.: Computer Assisted Learning (CAL) durch Berechnungsexperimente; Januar 1994.
- Nr. 27 Kirn, St., Unland, R. (Hrsg.): Tagungsband zum Workshop "Unterstützung Organisatorischer Prozesse durch CSCW". In Kooperation mit GI-Fachausschuß 5.5 "Betriebliche Kommunikations- und Informationssysteme" und Arbeitskreis 5.5.1 "Computer Supported Cooperative Work", Westfälische Wilhelms-Universität Münster, 4.-5. November 1993
- Nr. 28 Kirn, St., Unland, R.: Zur Verbundintelligenz integrierter Mensch-Computer-Teams: Ein organisationstheoretischer Ansatz; März 1994.
- Nr. 29 Kirn, St., Unland, R.: Workflow Management mit kooperativen Softwaresystemen: State of the Art und Problemabriß; März 1994.
- Nr. 30 Unland, R.: Optimistic Concurrency Control Revisited; März 1994.
- Nr. 31 Unland, R.: Semantics-Based Locking: From Isolation to Cooperation; März 1994.
- Nr. 32 Meckenstock, A., Unland, R., Zimmer, D.: Controlling Cooperation and Recovery in Nested Transactions; März 1994.
- Nr. 33 Kurbel, K., Schnieder, T.: Integration Issues of Information Engineering Based I-CASE Tools; September 1994.
- Nr. 34 Unland, R.: TOPAZ: A Tool Kit for the Construction of Application Specific Transaction; November 1994.
- Nr. 35 Unland, R.: Organizational Intelligence and Negotiation Based DAI Systems - Theoretical Foundations and Experimental Results; November 1994.
- Nr. 36 Unland, R., Kirn, St., Wanka, U., O'Hare, G.M.P., Abbas, S.: AEGIS: AGENT ORIENTED ORGANISATIONS; Februar 1995.
- Nr. 37 Jung, R., Rimpler, A., Schnieder, T., Teubner, A.: Eine empirische Untersuchung von Kosteneinflußfaktoren bei integrationsorientierten Reengineering-Projekten; März 1995.
- Nr. 38 Kirn, St.: Organisatorische Flexibilität durch Workflow-Management-Systeme?; Juli 1995.
- Nr. 39 Kirn, St.: Cooperative Knowledge Processing: The Key Technology for Future Organizations; Juli 1995.
- Nr. 40 Kirn, St.: Organisational Intelligence and Distributed AI; Juli 1995.
- Nr. 41 Fischer, K., Kirn, St., Weinhard, Ch. (Hrsg.): Organisationsaspekte in Multiagentensystemen; September 1995.

- Nr. 42 Grob, H. L., Lange, W.: Zum Wandel des Berufsbildes bei Wirtschaftsinformatikern, Eine empirische Analyse auf der Basis von Stellenanzeigen, Oktober 1995.
- Nr. 43 Abu-Alwan, I., Schlagheck, B., Unland, R.: Evaluierung des objektorientierten Datenbankmanagementsystems ObjectStore, Dezember 1995.
- Nr. 44 Winter, R., Using Formalized Invariant Properties of an Extended Conceptual Model to Generate Reusable Consistency Control for Information Systems; Dezember 1995.
- Nr. 45 Winter, R., Design and Implementation of Derivation Rules in Information Systems; Februar 1996.
- Nr. 46 Becker, J.: Eine Architektur für Handelsinformationssysteme; März 1996.
- Nr. 47 Becker, J., Rosemann, M. (Hrsg.): Workflowmanagement - State-of-the-Art aus Sicht von Theorie und Praxis, Proceedings zum Workshop vom 10. April 1996; April 1996.
- Nr. 48 Rosemann, M., zur Mühlen, M.: Der Lösungsbeitrag von Metadatenmodellen beim Vergleich von Workflowmanagementsystemen; Juni 1996.
- Nr. 49 Rosemann, M., Denecke, Th., Püttmann, M.: Konzeption und prototypische Realisierung eines Informationssystems für das Prozeßmonitoring und -controlling; September 1996.
- Nr. 50 Rehfeldt, M., Skall, M., Turowski, K., v. Uthmann, C.: Workflow-basierte Geschäftsprozeßregelung als Konzept für das Management von Produktentwicklungs-Prozessen; November 1996.
- Nr. 51 Eicker, S., Jung, R., Nietsch, M., Winter, R.: Entwicklung eines Data Warehouse für das Produktionscontrolling: Konzepte und Erfahrungen; November 1996.