Architektur von Data Warehouses und Business Intelligence Systemen

Bernhard Humm · Frank Wietek

Business Intelligence (BI) ist der Prozess der Umwandlung von Daten in Informationen und weiter in Wissen. Entscheidungen und Prognosen stützen sich auf dieses Wissen und schaffen dadurch Mehrwert für ein Unternehmen. Ein Data Warehouse (DW) bildet in vielen Fällen die technische Basis zur Implementierung einer BI-Lösung.

Dieser Artikel stellt DW-Konzepte und -Architekturen dar, welche die Erprobung in mehrjähriger Praxis bestanden haben und somit als etabliert gelten. Er erläutert gängige Begriffe und beantwortet konkrete Fragestellungen des Systementwurfs. Der Bogen spannt sich von der Modellierung der fachlichen Architektur über die technische Architektur und Systemarchitektur bis zu Produkten. Den Abschluss bilden aktuelle Markttrends.

Historie

Entscheidungsunterstützende (*dispositive/analytische*) Systeme haben eine lange Historie seit den 60er-Jahren und wurden im Verlauf der Jahrzehnte bei recht ähnlicher Funktionalität lediglich unterschiedlich betitelt: *Management Information Systems (MIS), Decision Support Systems (DSS), Executive Information Systems (EIS), Data Warehouses (DW)* und schließlich *Business Intelligence (BI)-Lösungen*. Abbildung 1 gibt eine Übersicht über jeweilige Kernideen beziehungsweise Schlagworte.

Im Laufe dieser Zeit wurden das Maß der Integration von Daten und das Niveau der Entscheidungsunterstützung permanent verbessert. Während die Ansätze aus den 60er-und 70er-Jahren noch weitgehend – gemessen an ihren Ansprüchen – scheiterten, bauten die Werkzeuge späterer Generationen jeweils auf ihren Vorgängern auf, lernten aus deren Fehlern und wurden zunehmend erfolgreicher.

Operative und dispositive Systeme: OLTP versus OLAP

Dienste, welche die Durchführung des operativen Geschäfts eines Unternehmens unterstützen, werden auch *OLTP (Online Transactional Processing)* genannt, dispositive Dienste auch *OLAP (Online Analytical Processing)*. OLAP und OLTP haben sehr unterschiedliche Charakteristiken, so dass sich eine Trennung in unterschiedliche Systeme anbietet. Tabelle 1 zeigt eine Gegenüberstellung.

Data Warehouse

Ein *Data Warehouse* ist ein dispositives System. Wir zitieren eine klassische Definition von Inmon [12]: Ein Data Warehouse ist eine themenorientierte, zeitorientierte, integrierte und unveränderliche Datensammlung, deren Daten sich für Managemententscheidungen auswerten lassen.
Insbesondere bedeuten:

- themenorientiert: alles über Kunden, Produkte etc.;
- zeitorientiert: periodische Ergänzung um aktuelle Daten, Verdichtung nach Zeitintervallen;
- integriert: Konsolidierung von Daten verschiedener operativer Systeme;
- unveränderlich: einmal gespeicherte Daten werden nicht mehr verändert.

DOI 10.1007/s00287-004-0450-5

Dr. B. Humm · Dr. F. Wietek
sd&m Research,
Carl-Wery-Straße 42, 81739 München
E-Mail: Bernhard.Humm@sdm.de
E-Mail: Frank.Wietek@sdm.de

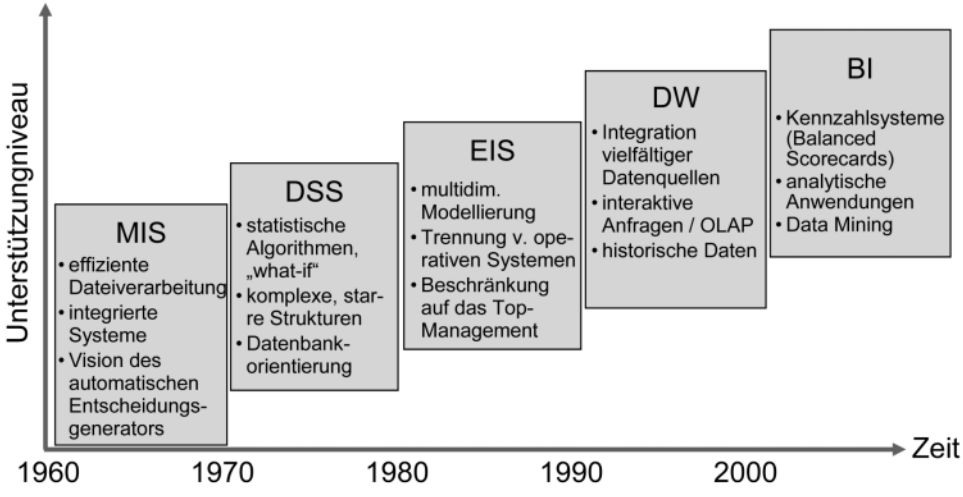

Abb. 1 Historie von entscheidungsunterstützenden Systemen

Aktuelle BI-Trends wie Real-Time Analytics, Planning/Commenting und das Zurückspielen analytischer Daten in operative Systeme weichen das Kriterium der Unveränderlichkeit auf. Vor diesem Hintergrund wird auch zunehmend von analytischen anstelle von dispositiven Systemen gesprochen.

Data Warehouses stellen typischerweise OLAP-Funktionalität zur Verfügung. Zur Beschreibung der interaktiven Analyse haben sich folgende Begriffe für Operationen etabliert:

- *Drill-Down* und *Roll-up*: schrittweise Verfeinerung bzw. Verdichtung von Analyseergebnissen, zum Beispiel von Jahres- über Monats- zu Tagesauswertungen. Die Verdichtung von Analyseergebnissen nennt man auch *Aggregation*. Diese wird in SQL mittels *Grouping Sets* und dem *Cube Operator* umgesetzt.
- *Slice-and-Dice*: Navigation in einem multidimensionalen Datenraum durch Fokussierung auf einzelne Aspekte, zum Beispiel Verteilung der Umsätze für ein bestimmtes Produkt auf unterschiedliche Regionen und Zeiträume.
- *Drill-Through*: direkter Zugriff aus analytischen Systemen auf operative Basisdaten, zum Beispiel auf einzelne Verträge.

Ein Bindeglied zwischen operativem Geschäft und dispositivem Einsatz bilden vordefinierte Standardberichte, die in der Regel einen großen Teil der DW-Nutzung ausmachen.

Business Intelligence

Business Intelligence umfasst ein breites Spektrum an Anwendungen und Technologien zur entscheidungsorientierten Sammlung, Aufbereitung und Darstellung geschäftsrelevanter Informationen. Es bezeichnet den analytischen Prozess, der Unternehmens- und Wettbewerbsdaten in handlungsgerichtetes Wissen über die Fähigkeiten, Positionen, Handlungen und Ziele der betrachteten internen oder externen Handlungsfelder (Akteure und Prozesse) transformiert [9]. BI ist der Oberbegriff für DW/OLAP, *Data Mining* und *Analytical Applications*. Data Mining umfasst – als Teilbereich des Knowledge Discovery in Databases – (teil)automatisierte Techniken zum Auffinden von Strukturen in großen Datenmengen, zum Beispiel die Kundensegmentierung nach Nutzungsprofilen [8]. Analytical Applications umfassen Anwendungen zur Planung, Simulation und zur Berechnung komplexer Kennzahlsysteme.

Die folgenden Abschnitte beschreiben die Modellierung (fachliche Architektur) und das Design (technische Architektur und Systemarchitektur) von Data Warehouses.

Business Intelligence

Entscheidungsunterstützende Systeme haben eine Historie von über 40 Jahren. Sie werden heute unter dem Schlagwort *Business Intelligence (BI)* zusammengefasst. *Data Warehousing (DW)* ist die wichtigste BI-Technologie, für die etablierte Modellierungstechniken, Architekturen und reife Produkte existieren. Dieser Artikel gibt eine einführende Übersicht in die Architektur von Data Warehouses und Business-Intelligence-Systemen auf der Basis umfangreicher Projekterfahrungen.

	Operativ (OLTP)	Dispositiv (OLAP)
Dienst	Unterstützung des operativen Geschäfts	Unterstützung von Analysen und Entscheidungen
Daten	aktuell, detailliert	historisch, verdichtet und aufbereitet
Operationen	Anlegen, Lesen, Ändern, Löschen: satzorientiert, vordefiniert, wenige Daten je Transaktion	multidimensionale Abfragen: ad-hoc, viele Daten je Anfrage; zusätzlich Aktualisierung periodisch im Hintergrund

Tabelle 1

Fachliche Architektur

Sprachen und Techniken zur Modellierung betrieblicher Informationssysteme wie *Entity/Relationship (E/R) Modellierung* [5] oder *Object Oriented Analysis* (*OOA;* [6]) sind nicht nur etabliert, sondern auch umfassend standardisiert, zum Beispiel für die OOA durch *UML* [23, 24]. Solche Standards für die fachliche Modellierung von Data Warehouses existieren noch nicht – Ansätze gibt es in Form von Notationen wie ADAPT [4] sowie Metadaten- und Schnittstellenstandards wie dem Common Warehouse Metamodel der OMG [22] oder XML for Analysis (XML/A; [25]). Wir geben daher zu den nachfolgenden Begriffsdefinitionen auch gängige Synonyme an (vgl. auch z. B. [1, 13, 14, 15, 16, 17]).

Measures, Fakten, Dimensionen und fachliche Sterne

Die *Measure* (auch *Maßzahl* oder *Kennzahl* genannt) ist die kleinste Informationseinheit des DW und bildet die Basis für alle Auswertungen. Measures sind stets numerisch und können in der Regel aggregiert werden (Summe, Mittelwert etc.). Komplexe Measures können aus anderen Measures oder (evtl. nicht aggregierbaren) Basisinformationen abgeleitet sein. Wir unterscheiden Measures und *Measure-Ausprägungen.*

Beispiel: Measure `Umsatz [EUR]`, Ausprägung `570`.

Fakt-Beschreibungen dienen zur Anzeige von Zusatzinformationen zu einzelnen Measures. Fakt-Beschreibungen werden nicht aggregiert und müssen daher nicht numerisch sein.

Ein *Fakt* fasst eine Gruppe zusammengehörender Ausprägungen von Measures und Fakt-Beschreibungen zusammen. Jeder Fakt ist eindeutig einer Ausprägung jeder Dimensions-Basis (s. unten) zugeordnet. Häufig werden die Begriffe Fakt und Measure synonym verwendet.

BI technology

Systems that support business decisions have been established within the last 40 years. Today, they are subsumed under the heading *business intelligence (BI). Data warehousing (DW)* is the most important BI technology. Over the years, architectures and methodologies for building DW have been established and, today, mature products are available. This article provides an introductory overview of the architecture of DW and BI systems based on extensive project experience.

Beispiel eines Fakts aus einem Auftrags-DW:
`AuftragsNr = "4711"` (Fakt-Beschreibung),
`AnzahlVerkäufe = 10` (Measure),
`Umsatz [EUR] = 570` (Measure),
`Tag = 3.5.2003` (Dimension Zeit),
`BusinessUnit = "Niederlassung Rhein-Main"` (Dimension Region)

Eine *Dimension* ist ein Filter- und Auswertungs-Kriterium für Measures. Anders ausgedrückt: Dimensionen spannen einen mehrdimensionalen Fakten-Raum auf und bilden somit das Koordinatensystem zur Navigation durch die Daten. Eine Dimension kann aus mehreren *Dimensions-Elementen* bestehen, die hierarchisch, das heißt in einem Baum oder allgemein in einem gerichteten azyklischen Graphen, angeordnet sind. Diese Struktur ist die *Hierarchie der Dimension*. Wir unterscheiden wieder zwischen Dimensions-Elementen und ihren *Ausprägungen*.

Oftmals ist die Dimensions-Hierarchie linear, das heißt die Dimensions-Elemente bilden eine Liste von feinster Einteilung (der *Dimensions-Basis* mit *Basiswerten* als Ausprägungen) bis zu gröbster Einteilung. Dimensions-Elemente können verschiedene *Beschreibungen/Dimensionsattribute* haben,

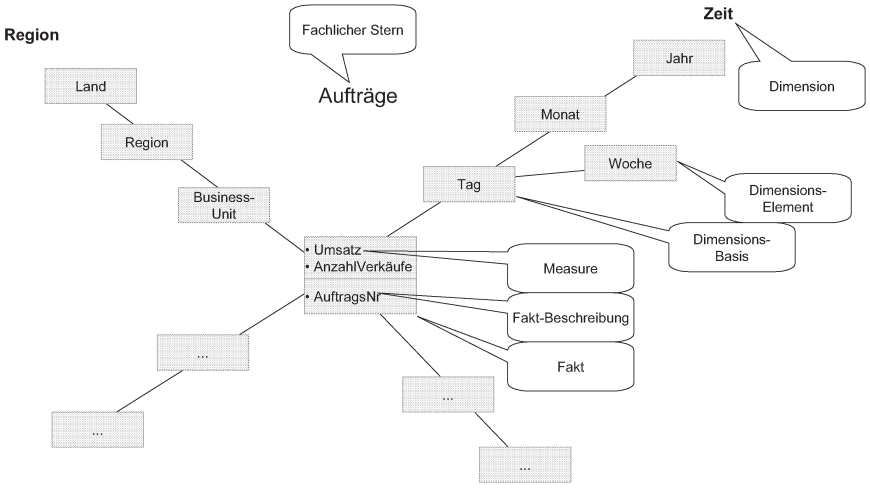

Abb. 2 Fachlicher Stern „Aufträge“

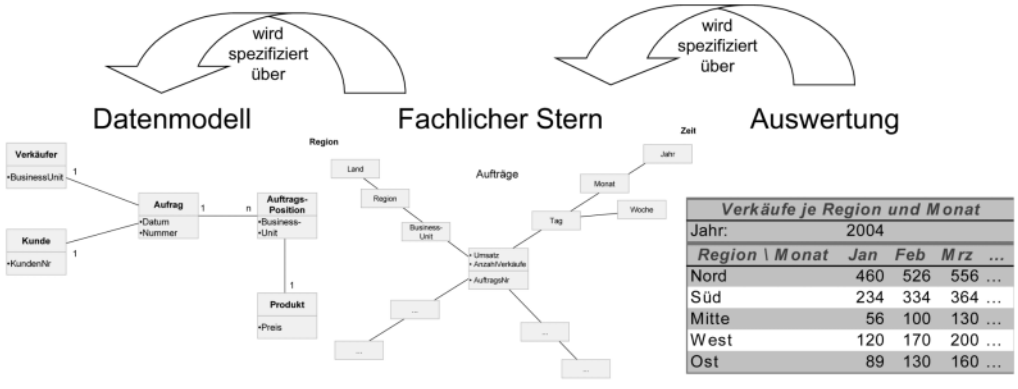

Abb. 3 Übersicht Modellierung

zum Beispiel Kurz- und Langtexte in verschiedenen Sprachen.

Beispiel: In nahezu jedem DW gibt es mindestens eine Dimension Zeit, zum Beispiel mit Dimensions-Basis „Tag“ (Ausprägung 03.10.2004) und weiteren Dimensions-Elementen „Monat“ (Oktober), “Jahr“ (2004). Häufig existieren auch mehrere Zeit-Dimensionen, zum Beispiel für das Auftrags- und das Lieferdatum.

Ein *fachlicher Stern (fachlicher Würfel/Cube, multidimensionales Datenmodell bzw. -schema)* ist die Definition von Measures, Fakt-Beschreibungen und zugehörigen Dimensionen. Hierbei können einzelne Dimensionen, Measures und Beschreibungen auch in verschiedenen Sternen wieder verwendet werden.

Abbildung 2 zeigt den fachlichen Stern „Aufträge“ für ein Auftrags-DW. Der Stern „Aufträge“ umfasst die Measures „Umsatz“ und „Anzahl Verkäufe“, die Fakt-Beschreibung „AuftragsNr“ und die Dimensionen „Zeit“ und „Region“.

Auswertungen

Fachliche Sterne sind nur Mittel zum Zweck – für den Anwender bringen die Auswertungen der Daten fachlichen Nutzen. Abbildung 3 zeigt, wie Auswertungen über Modelle spezifiziert werden, und komplettiert damit das Bild der fachlichen Architektur.

Eine Auswertung besteht aus Beschriftungen, Filtern, Gruppierungs-Ebenen und Kennzahlen:

- *Beschriftungen* können Zeilen, Spalten und die gesamte Auswertung erläutern, zum Beispiel der Titel „Verkäufe je Region und Monat“.
- *Filter* (d. h. Möglichkeiten zur Parametrisierung) schränken die dargestellten Kennzahlen ein. Im Beispiel: „Jahr = 2004“.
- *Gruppierungs-Ebenen* strukturieren die Daten des Reports. Sie bilden die Zeilen und Spalten einer Auswertung. Im Beispiel: „Region“, „Monat“.
- *Kennzahlen* definieren die eigentlichen Daten des Reports – sie bilden dessen Zellen.

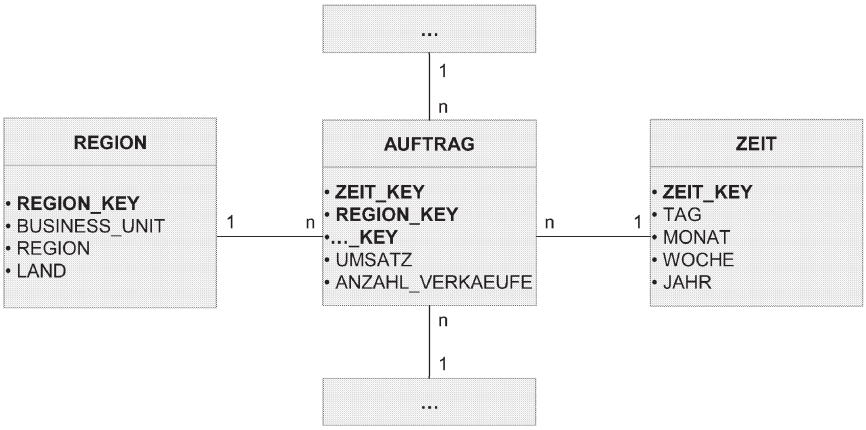

Abb. 4 Beispiel für ein Stern-Schema

Eine Auswertung wird auf der Basis von einem oder mehreren fachlichen Sternen spezifiziert – im Beispiel auf dem fachlichen Stern „Aufträge".

Die fachlichen Sterne selbst werden – ausgehend von Informationsanforderungen der Endanwender – auf der Basis von Datenmodellen (zum Beispiel E/R- oder OOA-Modellen) der operativen Systeme spezifiziert. Die Spezifikation kann beispielsweise in Pseudocode wie folgt aussehen:

- Dimensionsbasis:
  Tag = Auftrag.Datum
- Measure:
  AnzahlVerkäufe = sum(Auftrag.AuftragsPosition.Anzahl)
- Fakt-Beschreibung:
  AuftragsNr = Auftrag.Nummer

Mit dieser Spezifikation schließt sich der Kreis, und die Auswertungen sind präzise auf der Basis der Semantik der operativen Systeme definiert.

Theorie und Praxis

Das Verständnis der theoretischen Grundlagen zur Modellierung der fachlichen DW-Architektur hilft in der Spezifikationsphase eines DW-Projekts. Dennoch geht es leider in der Projektpraxis selten so einfach und strukturiert vor. Nachfolgend einige Beispiele:

- *Fehlende Datenmodelle*: Häufig existieren keine Datenmodelle operativer Systeme oder sie liegen in einer schlechten Qualität (unvollständig, fehlende Semantik, etc.) vor. In der Projektpraxis muss man sich häufig auf die Suche machen nach den Basisdaten für Auswertungen. Manchmal ist es dabei sogar Ziel führend, auf schon existierende Report-Strukturen der operativen Systeme direkt aufzusetzen.
- *Inkonsistente Datenmodelle*: Datenmodelle operativer Systeme sind häufig untereinander semantisch inkonsistent. Noch viel gravierender ist, dass das Verständnis unterschiedlicher Fachbereiche von identischen Begriffen häufig weit auseinander geht bzw. fachliches Verständnis und implementierte Datenmodelle nicht zueinander passen. Hier helfen nur umfangreiche Abstimmungen und gegebenenfalls der Entwurf unterschiedlicher Sichten für unterschiedliche Nutzerkreise auf Basis eines integrierten Datenmodells und klarer Begriffsabgrenzungen.
- *Measure vs. Dimension*: Die Entscheidung, ob ein Datum als Measure oder als Dimension abgebildet wird, ist eine Design-Entscheidung, die viel Erfahrung erfordert.
  Beispiel: Plan- und Ist-Umsätze können mit zwei Measures „PlanUmsatz" und „IstUmsatz" modelliert werden. Alternativ können ein Measure „Umsatz" und eine Dimension „Umsatztyp" mit den Ausprägungen „Plan", „Ist" – und evtl. weiteren Ausprägungen – verwendet werden. Die erste Variante ist natürlicher und performanter zu implementieren, die zweite Variante ist jedoch flexibler und einfacher um weitere Ausprägungen zu erweitern.
- *Fakten- vs. Dimensionsbeschreibungen*: Gruppen von Faktenbeschreibungen können auch als Attribute einer eigenen Dimension modelliert werden. Beispiel: Kommen in obigem Beispiel zur „AuftragsNr" noch Beschreibungen wie „Auftragstitel" und „Bearbeitungsinstruktionen" hinzu, könnte man hieraus eine eigene Auftragsdimension erzeugen bzw. Aufträge als Dimensions-Basis einer bereits bestehenden Kostenstellendimension einfügen.
- *Parallelhierarchien*: Parallelhierarchien treten auf, wenn Dimensions-Elemente Verzweigungen haben. Ein klassisches Beispiel findet sich im fachlichen Stern Aufträge: Während Monate eindeutig einem Jahr zuzuordnen sind, können Wochen nicht ohne weiteres Jahren zugeordnet werden. Eine Möglichkeit ist es, Wochen als Kalenderwochen eindeutig Jahren zuzuweisen. Man erhält dann in der Modellierung von Dimensionselementen einen gerichteten azyklischen Graphen. Dies führt zu einer komplexen Modellierung mit verschiedenen Jahresdefinitionen. Alternativ kann man auf die Aggregation von Wochen zu Jahren verzichten. Häufig führt die Verwendung von Parallelhierarchien zu Problemen

bei der technischen Umsetzung und der Nutzung.

- *Dimensionen variabler Tiefe*: Diese sind notwendig, wenn die Dimensionselemente nicht statisch festgelegt werden können.
- *Beispiel*: Im operativen System wird ein Klassifkationsschema für Aufträge in Form eines Baums gepflegt. Die Außendienstmitarbeiter klassifizieren jeden gewonnenen Auftrag durch Auswahl und Zuordnung eines Baumelements. Im DW sollen Aufträge nach deren Klassifikation gefiltert und gruppiert werden. Modellierungstechnisch ist dies einfach umzusetzen mit einem Dimensionselement „Kategorie" und der Einführung einer Rekursionskante von „Kategorie" zu „Kategorie". Eine performante technische Umsetzung ist jedoch schwierig.

Technische Architektur

Star-Schema und Snowflake-Schema

Wird ein *relationales Datenbankmanagementsystem (DBMS)* für die Datenhaltung des DW verwendet, so werden Fakten und Dimensionen klassischerweise nach einem *Star Schema* gespeichert. Abbildung 4 zeigt dies am Beispiel.

Für jeden fachlichen Stern wird eine *Faktentabelle* angelegt. Diese enthält als Attribute die Measures, die Fakt-Beschreibungen und Fremdschlüssel auf *Dimensionstabellen* für jede Dimension. Diese enthalten denormalisiert jeweils alle Dimensionselemente sowie weitere Dimensionsbeschreibungen als Attribute.

In einem *Snowflake-Schema* werden die Dimensionstabellen durch Aufgliederung nach Hierarchiestufen (Dimensionselementen) normalisiert. Die Normalisierung im Snowflake-Schema geht stark auf Kosten des performanten Zugriffs und der einfachen Darstellbarkeit durch DW-Frontend-Werkzeuge. Trotzdem hat ein Snowflake-Schema mit beschränkter Hierarchietiefe und in geeigneter Kapselung vor dem Endanwender in speziellen Anwendungsfällen, etwa im Rahmen der Versionierung, durchaus seine Berechtigung.

Relational (ROLAP) versus multidimensional (MOLAP)

Außer relationalen DBMS werden auch multidimensionale DBMS zur Datenhaltung von DW eingesetzt. Zur Unterscheidung werden die Begriffe *MOLAP (Multidimensional OLAP), ROLAP (Relational OLAP)* und *HOLAP (Hybrid OLAP)* verwendet.

- *MOLAP* -Lösungen implementieren die multidimensionale Sicht auf Analysedaten physisch, indem die Fakten inklusive Zwischensummen und abgeleiteten Kennzahlen sequentialisiert in *multidimensionalen Datenbanken (MDDB)* gespeichert werden. Zusammengehörende Fakten werden auch als *Datenwürfel (Cubes)* bezeichnet.
- *ROLAP* -Lösungen implementieren fachliche Sterne in relationalen DBMS, üblicherweise nach dem Star-Schema, seltener nach dem Snowflake-Schema oder anderen Schemata. Der Zugriff auf die Daten erfolgt entweder direkt über die Anfragesprache des DBMS oder über eine ROLAP-Engine, die die angefragten Ergebnisdaten multidimensional aufbereitet und bei Maßnahmen zum Performance-Tuning unterstützt.
- *HOLAP* bezeichnet die für den Benutzer mehr oder weniger transparente Kombinationen von ROLAP- und MOLAP-Technologien.

MOLAP bietet ein breites Funktionsspektrum für multidimensionale Operationen mit intuitiven Analysesprachen. Derzeit unterstützen Produkte jedoch nur ein begrenztes Datenvolumen. MOLAP ist zu empfehlen, wenn spezielle Funktionalität und hohe Performanz komplexer analytischer Auswertungen auf nicht zu großen Datenbasen gefordert, aber der Aufwand für das Würfel-Update, also den Aufbau der multidimensionalen Strukturen aus den Basisdaten, unkritisch ist.

ROLAP bietet hohe Performance, Stabilität und Betriebssicherheit auch für große Datenmengen. Nur für komplexe multidimensionale Anfragen ist der Befehlsumfang teilweise noch eingeschränkt. Die möglichst transparente Bereitstellung und Nutzung von Voraggregierungen zur Performancesteigerung – bis vor einigen Jahren noch ein großer Pluspunkt von MOLAP – wird inzwischen auch von den meisten ROLAP-Systemen gut unterstützt. Insgesamt ist ROLAP in der Regel die einfachere, kompaktere, preisgünstigere und flexiblere Lösung.

Die großen ROLAP-Anbieter sind mit der Integration von MOLAP-Technologien zu HOLAP-Lösungen mehr oder weniger weit fortgeschritten und können darüber einerseits die Nachteile heterogener Produktlandschaften (zum Beispiel Zusatzkosten für Administration und SW-Lizenzen) überwinden und andererseits die Stärken beider Welten vereinen.

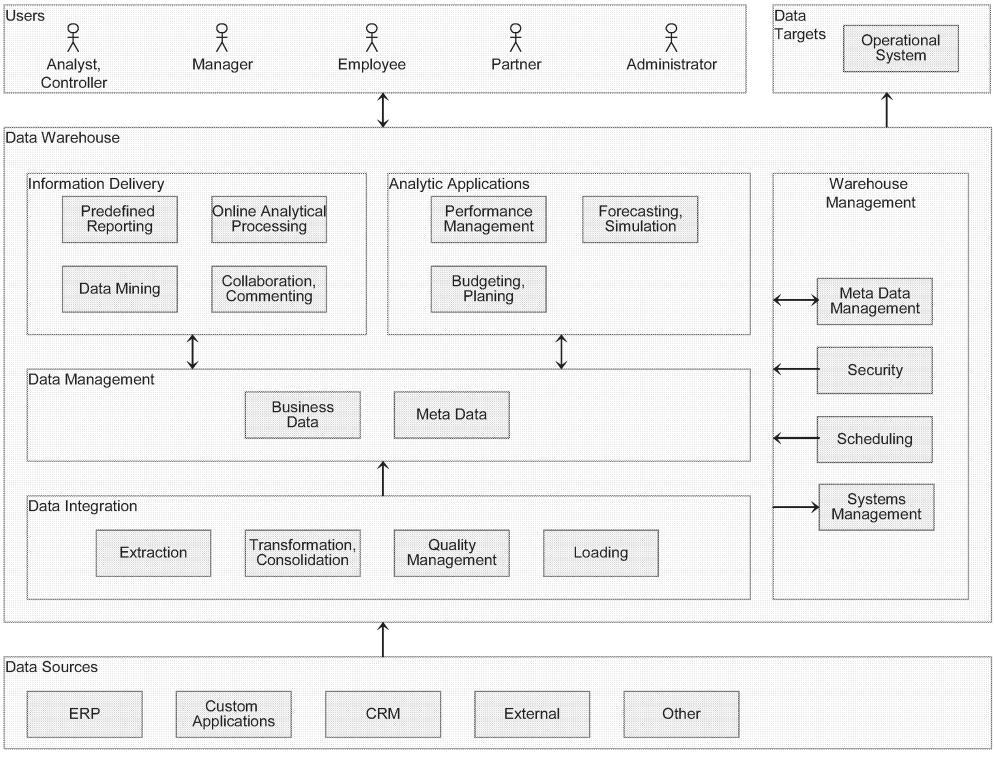

Abb. 5 BI-Referenzarchitektur

Systemarchitektur

Die BI-Referenzarchitektur

Erfolgreich in die fachliche und technische Anwendungslandschaft eines Unternehmens integrierte BI-Lösungen zeichnen sich durch eine klare Struktur aus. Unabhängig von den verwendeten Produkten weisen BI-Architekturen Gemeinsamkeiten auf, die wir in einer BI-Referenzarchitektur (Abb. 5) destilliert haben (vgl. auch [1]).

Die Referenzarchitektur ist modular und serviceorientiert aufgebaut. Services können von unterschiedlichen Produkten oder Individuallösungen abgedeckt werden. Nach der Referenzarchitektur unterteilen wir ein DW in die drei aufeinander aufbauenden Bereiche *Data Integration*, *Data Management* und *Information Delivery/Analytic Applications*. Daneben stehen Querschnittsfunktionen zum *Warehouse Management*.

Bereich „Data Integration"

Operative Systeme eines Unternehmens bilden die *Datenquellen (Data Sources)* für das DW. Beispiele sind:

- *Enterprise Resource Planning Systeme (ERP)*, z. B. SAP R/3,
- *Custom Appliations*, z. B. das System für den Zahlungsverkehr eines Finanzdienstleisters,
- *Customer Relationship Management Systeme (CRM)*, zum Beispiel von Siebel,
- *Externe Systeme*, zum Beispiel zur Bereitstellung von Währungskursen,
- *Weitere Anwendungen*, zum Beispiel Microsoft Excel.

Data Integration stellt die Schnittstelle zu den Quellsystemen dar. Die Daten werden aus den Quellsystemen extrahiert, transformiert, qualitätsgesichert und der DW-Datenhaltung zur Speicherung übergeben. Den Prozess der Datenintegration bezeichnet man auch als *ETL (Extraction, Transformation, and Loading)*.

Man spricht von einer *Staging Area* (einem Arbeitsbereich) als dem temporären Bereich in einer Datenbank oder dem Filesystem, in dem die Vorverarbeitung der zu ladenden Daten stattfindet, bevor sie in das Ziel-DW geladen werden. Ein derartiges Staging wird vor allem aufgrund der Komplexität von ETL-Prozessen erforderlich.

Mitunter fließen die Ergebnisse des Transformationsprozesses nicht direkt in das DW, sondern zunächst in ein *Operational Data Store (ODS*; [11]). Diese Datenbank bietet eine integrierte, konsolidierte, datenzentrierte Sicht auf die Quelldaten, beschränkt sich jedoch auf den jeweils aktuellen Stand und ist typischerweise sowohl fachlich wie auch technisch relational modelliert. Sowohl die

Historisierung als auch die Umstrukturierung der Daten in eine denormalisierte, multidimensionale Sicht erfolgen in der Regel erst beim Laden in das DW. Ein ODS wird meist häufiger aktualisiert als das DW und kann daher zur Unterstützung operativ taktischer Entscheidungen auf Basis eines integrierten Datenbestands verwendet werden.

ETL wird heute durch ausgereifte Werkzeuge unterstützt. Die Zwischenablage in einer Staging Area verliert aufgrund der damit verbundenen höheren Laufzeit des Prozesses immer mehr an Bedeutung.

Besondere Bedeutung kommt dem *Quality Management* zu. Daten aus operativen Systemen haben in der Praxis immer unterschiedliche, häufig auch sehr schlechte Qualität. Meist ist mit der Einführung eines DW auch eine umfangreiche manuelle Datenpflege durch Fachbereiche notwendig. Systemtechnisch muss das DW mit ungültigen, inkonsistenten, unvollständigen oder fehlenden Daten adäquat umgehen. Hierzu dienen Verfahren zu Data Profiling, Parsing, Standardisierung und Matching/Data Cleansing sowie zur Verifizierung von Datensätzen [2].

Bereich „Data Management"

Das *Data Management* bildet den Kern des DW. In ihm sind die Geschäftsdaten, also die Inhalte fachlicher Sterne, sowie Metadaten des DW gespeichert. Die Speicherung der Geschäftsdaten im DW-Kern erfolgt üblicherweise auf Basis eines fachlich multidimensionalen Modells.

Zusätzlich können optional Geschäftsdaten in (ebenfalls multidimensional modellierten) *Data Marts* gespeichert werden. Data Marts schneiden bestimmte fachliche Ausschnitte aus dem Gesamtdatenbestand heraus und stellen diese für effiziente Auswertungen bereit. Sie können als separate Datenbanken/Schemata oder aber auch als Views auf den DW-Kern implementiert werden. Häufiger als der DW-Kern werden Data Marts auch technisch multidimensional abgelegt (beispielsweise als MOLAP-Würfel, die den Endnutzern zur Verfügung gestellt werden). Gerade in großen DW-Projekten wird oft ein großer DW-Kern entwickelt, aus dem in einzelnen Projektstufen fachlich abgegrenzte Data Marts abgeleitet bzw. geladen werden *(Hub-and-Spoke-Architektur)*.

Bereiche „Information Delivery" und „Analytical Applications"

Die *Information-Delivery-Services* extrahieren Informationen aus den Daten und stellen sie den Anwendern in vielfältiger Form zu Verfügung:

- *Predefined Reporting*: Ausführung vordefinierter, in der Regel parametrisierter Standardberichte entweder auf Anfrage oder regelmäßig im Batch;
- *Online Analytical Processing (OLAP)*: Ad-hoc-Formulierung von Anfragen, Navigation durch den multidimensionalen Datenbestand mittels Slice & Dice, Drill-down und Roll-up;
- *Data Mining*: ungerichtete, teilweise automatisierte Untersuchung und Analyse großer Datenmengen, um wichtige Muster, Trends, Beziehungen und Regeln zu entdecken;
- *Collaboration/Commenting*: Unterstützung der Kooperation/Kommunikation von Anwendern bei der Datenanalyse, unter Anderem durch Speicherung von Annotationen im DW und vereinfachten Austausch von Analyseergebnissen.

Auch *Analytical Applications* erweitern die Funktionalität klassischer DW. Sie stellen Informationen in anwendungsspezifischen Zusammenhängen dar:

- *Business Performance Management*: Leistungsmessung der internen und externen Prozesse durch definierte Metriken (z. B. *Balanced Scorecards*) mit dem Ziel der Optimierung von Geschäftsprozessen;
- *Forecasting/Simulation*: Fortschreibung aktueller Kennzahlen in die Zukunft auf Basis flexibler Vorhersagemodelle;
- *Budgeting/Planning*: interaktiver Vergleich und Bewertung von Planzahlen, *What-If-* und *Break-Even-Analysen.*

In der Regel erfolgt der Datenfluss unidirektional aus dem DW zu den analytischen Anwendungen. In Ausnahmefällen (Commenting, Planning) werden jedoch auch Daten in das DW zurück geschrieben.

Von zunehmender Bedeutung ist auch der Rückfluss aus dem DW in die operativen Systeme zur Unterstützung taktischer Entscheidungen im täglichen Geschäft (*Data Targets*), zum Beispiel der Rückfluss von Analyseergebnissen zur Kundensegmentierung aus dem analytischen CRM in das operative CRM zur Durchführung von Kampagnen.

Bereich „Warehouse Management"

Unter *Warehouse Management* verstehen wir die für den Aufbau, Pflege und den Betrieb eines DW notwendigen Querschnittsfunktionen:

- *Meta Data Management*: Verwaltung aller Metadaten des DW, insbesondere Metadaten-Austausch zwischen den verschiedenen DW-Komponenten bzw. Bereitstellung einer gemeinsam genutzten, homogenen Metadatenbasis;
- *Security*: Dienste zur Authentifizierung (Benutzerkennung), Autorisierung (Zugriffskontrolle auf möglichst feingranularer Ebene) und evtl. Verschlüsselung – hierbei möglichst Unterstützung eines *Single Sign-on;*
- *Scheduling*: Steuerung von DW-Prozessen, insbesondere ETL oder die regelmäßige Generierung und Verteilung von Berichten;
- *Systems Management*: Werkzeuge für den Betrieb des DW, zum Beispiel zum Performance- und Auslastungs-Monitoring sowie zur Archivierung und Datensicherung.

Von den Anforderungen zur konkreten Architektur: Kernfragen des DW-Design

Die BI-Referenzarchitektur stellt eine Leitlinie für die Entwicklung konkreter DW-Architekturen dar. Sie ermöglicht es, in einer frühen Projektphase schnell und effizient die notwendigen Services zu bestimmen, den Umfang des Projekts zu umreißen und die grundlegenden Fragestellungen zur Absicherung eines erfolgreichen Projekts zu klären. Beispiele sind:

- Welche Anwendergruppen sollen das System nutzen? Für Analysten/Controller eignen sich OLAP und Analytic Applications. Für Manager ist eher Predefined Reporting adäquat;
- Wie viele Anwender bedienen das System und wie sind die Performance-Anforderungen? Bei hohen Lastanforderungen ist derzeit ROLAP der MOLAP-Technologie vorzuziehen;
- Wo arbeiten die Anwender? Arbeiten die Anwender innerhalb eines Unternehmensnetzwerks, so können lokale Desktop-Anwendungen installiert werden. Andernfalls und auch bei großen Nutzerzahlen ist ein Browser-basierter Zugang notwendig;
- Wie ist die Heterogenität und Qualität der Quelldaten? Wie heterogen sind die Anwendergruppen mit ihren bisherigen Begriffswelten und Kennzahlsystemen? In jedem Fall müssen Maßnahmen zur Konsolidierung und zum Quality Management aufgesetzt werden, sowohl systemtechnisch, aber vor allem auch organisatorisch.
- Auf welcher Architekturebene soll das zukünftige Berichtswesen/die Datenanalyse fachlich bzw. technisch aufsetzen: verbessertes Production Reporting auf den Quellsystemen, konsolidierte Sicht auf ein ODS, multidimensionale Analyse auf dem gesamten DW-Bestand oder einzelnen Ausschnitten (Data Marts)? Entsprechend sind Verarbeitungsprozesse und Komponentenschnittstellen zu gestalten.
- Ist ein Rückspielen von Analyseergebnissen in operative Systeme (Data Targets) geplant? Dann ist eine saubere Trennung von operativen und dispositiven Daten besonders wichtig. Falls sich durch die Einbindung der DW-Analysen in das Tagesgeschäft eine erhöhte Kritikalität der DW-Implementierung ergibt, sind entsprechende Maßnahmen zur Sicherstellung von Verfügbarkeit, Datenaktualität und Ausfallsicherheit vorzusehen.
- Welche Information Delivery Dienste und Analytic Applications sind geplant oder für die Zukunft angedacht? Danach ist die Produktauswahl weitsichtig auszurichten.
- Sind der Drill-Through auf operative Daten und Real-time Analytics geplant bzw. für die Zukunft anvisiert? Dann ist die gesamte Systemarchitektur der Data Integration frühzeitig darauf auszurichten. Eine nachträgliche grundlegende Umgestaltung ist schwierig und aufwändig.

Noch wichtiger als derartige fachlich-technische Aspekte sind oft organisatorische Fragen zum Projektvorgehen und DW-Betrieb – von Umfang und stetiger Weiterentwicklung einer DW-Lösung („think big – start small") über Einbettung in die IT-Gesamtstrategie, Sponsoring im Management, Klärung von Verantwortlichkeiten und Abstimmung der Erwartungshaltungen bis zur Einbindung der Fachabteilungen in die Planung. Dies ist nicht Gegenstand dieses Artikels – s. aber dazu z. B. [2] oder [10].

DW-Archetypen

Aus der Summe betrachteter Einzelaspekte einer DW-Implementierung ergeben sich typische Szenarien mit ähnlichen Charakteristika. Das Bewusstsein um die fachliche Einordnung eines DW-Projekts erleichtert den Rückgriff auf bewährte Modellierungs- und Designansätze. In der Regel treten mehrere dieser so genannten *Archetypen* in Kombination in einer konkreten DW-Implementierung auf:

- *Transaktionsorientiert*: Der Klassiker unter den DW, der sich eng am theoretischen Modell mittels Star Schemas implementieren lässt. Die aus Buchungs- oder ähnlichen Systemen importierten Quelldaten umfassen vordefinierte Fakten wie Mengen oder Umsätze. Typische Dimensionen sind Stammdaten zu Kostenstellen und -arten, Zeitperioden, Produkten etc., über die die Fakten in der Regel mittels Addition aggregiert werden können.
- *Workflow-orientiert*: Die Quelldaten des DW stammen aus Workflow-Engines oder ähnlichen prozessunterstützenden Systemen. Diese liefern Status-/ Ereignis-Records und deren

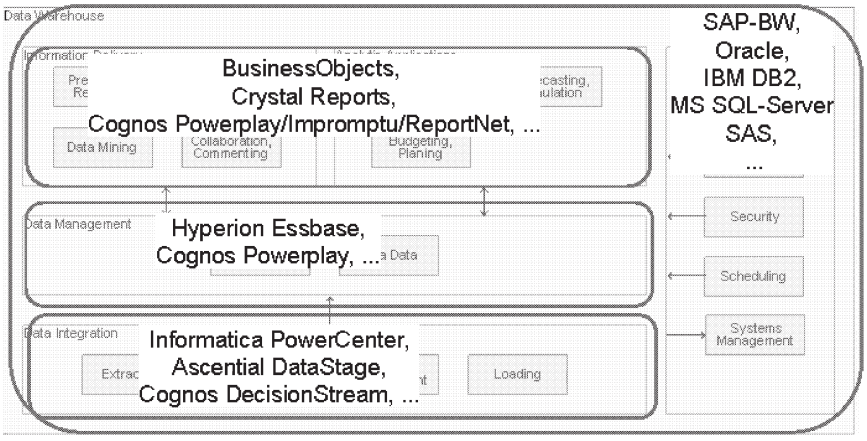

Abb. 6 Exemplarische BI-Produktlandkarte

Zeitstempel, die sich auf bestimmte Vorgänge beziehen, z. B. die Abwicklung eines Auftrags. Aus den Statusübergängen werden Measures abgeleitet, zum Beispiel die Anzahl offener Aufträge.

- *Stammdatenorientiert*: DW-Lösungen können auch (fast) ganz ohne Measures auskommen. Sollen nur Stammdaten (Kunden, Verträge, Produkte etc.) verwaltet und für DW-Auswertungen bereitgestellt werden, ist es sinnvoll, die Daten von der normalisierten Repräsentation im operativen System zu einer denomalisierten Darstellung für das DW zu transformieren. Dabei definieren die Faktentabellen als so genannte „*factless fact tables*" lediglich die Verknüpfungen zwischen den entsprechenden Dimensionen.
- *Anwendungsspezifisch*: Beispiele sind Systeme zur Deckungsbeitragsrechnung und zur Bilanzierung. Diese Systeme unterliegen der Anforderung, dass sie die gleichen Zahlen liefern müssen wie die entsprechenden Finanzmodule im operativen System. Dort sind oft Spezialregeln für die Aggregation (wie z. B. die Soll-Haben-Steuerung) implementiert, die im DW ebenfalls explizit zu implementieren sind. Kennzahldimensionen bilden Beziehungen zwischen diesen Measures ab. Abgesehen von derartigen Anwendungsspezifika gibt es viele Parallelen zum klassischen transaktionsorientierten DW.

Produkte

Best-of-Breed versus Tool Suite

Am Markt existieren zahlreiche BI-Produkte, sowohl branchenspezifisch als auch -übergreifend, die wesentliche Dienste der BI-Referenzarchitektur abdecken. DW-Produkte sind seit über zehn Jahren auf dem Markt und haben mittlerweile eine hohe Reife. Von daher ist der Produkteinsatz im Normalfall einer Individualentwicklung vorzuziehen. Die Auswahl der passenden Produkte anhand priorisierter Auswahlkriterien (Performance, Skalierbarkeit, Funktionalität, Benutzungsfreundlichkeit, Integrierbarkeit, Support, Marktposition des Herstellers etc.) ist für ein Unternehmen eine verantwortungsvolle Aufgabe, da sie eine große Auswirkung auf die Entwicklung, den Betrieb und die Nutzung der Lösung hat [2, 19, 20]. Man unterscheidet die folgenden Ansätze zum Produkteinsatz:

- *Best-of-Breed*: Für jeden Dienst der Referenzarchitektur wird jeweils das für die Anwendung am besten geeignete Werkzeug ausgewählt, auch wenn diese von unterschiedlichen Herstellern angeboten werden.
- *Tool Suite*: Es wird ein Hersteller ausgewählt, der eine integrierte Plattform mit allen wesentlichen Diensten der Referenzarchitektur anbietet.

Beide Ansätze haben Vor- und Nachteile, und es muss jeweils im konkreten Anwendungsfall entschieden werden. Für eine Best-of-Breed-Lösung spricht die bessere funktionale Abdeckung, besonders bei fachlichen Spezialanforderungen. Für eine Tool Suite sprechen reduzierte Integrations- und Managementaufwände. Derzeit ist davon auszugehen, dass vor diesem Hintergrund die Marktkonsolidierung voranschreiten wird. Nur eine Handvoll großer Anbieter (Kandidaten sind etwa SAS, Oracle, Microstrategy, Business Objects, Hyperion, Cognos, SAP, Informatica, Microsoft) werden sich durchsetzen, lediglich einige Spezialisten mit Nischenangeboten bzw. auf bestimmte Branchen oder Analysemethoden zugeschnittenen BI-Lösungen (als „Plug-Ins" für BI-Plattformen) werden überleben [7, 18, 21].

Produktlandkarte

Die Kategorisierung von Produkten nach der BI-Referenzarchitektur hilft, Produkte mit ähnlicher

Funktionalität zu vergleichen und zu einer Produktauswahl, sowohl nach dem Best-of-Breed-Ansatz als auch nach dem Tool-Suite-Ansatz zu kommen. Abbildung 6 zeigt eine solche vereinfachte *Produktlandkarte*, in der exemplarisch einige aktuell wichtige Produkte in der BI-Referenzarchitektur den einzelnen Bereichen zugeordnet sind.

Das Funktionsspektrum der jeweiligen Produkte ergibt sich aus dem abgedeckten Bereich der BI-Referenzarchitektur – Grad und Qualität der Abdeckung sind auf einem feineren Detaillevel zu bewerten. Die Grenzen sind hierbei sicher fließend, da die meisten Anbieter zunehmend umfangreiche Suiten auf den Markt bringen und auch kaum ein Produkt gänzlich ohne Warehouse-Management-Funktionalität auskommt:

- Im Bereich Data Management sind (neben Anbietern klassischer relationaler DBMS) Produkte zur Implementierung multidimensionaler Datenbanken (MOLAP) wie Hyperion Essbase oder Cognos Powerplay zu nennen. SAS ist ein klassischer HOLAP-Vertreter.
- Produkte zur Datenintegration wie Informatica PowerCenter oder Ascential DataStage ermöglichen den Zugriff auf eine Vielzahl verschiedenartiger Datenquellen sowie die Transformation und Zusammenführung der Daten vor dem Laden ins DW. Derartige ETL-Tools können entweder spezifischen Code zur kombinierten Extraktion und Transformation auf Seite der Datenquellen generieren, als separater ETL-Server zwischen Quellen und Ziel-DW den gesamten ETL-Prozess koordinieren oder Transformationen und Ladevorgänge unter Nutzung der Zieldatenbank als ETL-Engine implementieren. Letztere Variante ermöglicht attraktive Lizenzmodelle/-kosten für Tool-Suiten von RDBMS-Anbietern.
- Angebote zum Information Delivery/Analytic Applications (zum Beispiel von Cognos oder Business Objects) sind oftmals ihrerseits mehr oder weniger umfangreiche Suiten von Komponenten für die verschiedenen Einzeldienste und verschiedene Anwendergruppen. Es können Sichten auf das Kern-DW sowie Analysen definiert, die Auswertungen durchgeführt und die Ergebnisse verteilt werden. Weiterhin werden in der Regel ein personalisierbarer Zugriff über Web-Browser sowie Möglichkeiten zur Einbindung in Portale oder andere Applikationen geboten.
- Tool Suites bzw. Plattformen bieten schließlich aufeinander abgestimmte Einzelkomponenten zur Abdeckung aller Bereiche der BI-Referenzarchitektur (wie zum Beispiel SAP-BW oder Oracle 9i) und/oder implementieren eine Plattform mit Schwerpunkt auf dem Data- und Warehouse Management, auf die andere Produkte und Lösungen aufsetzen können (wie z. B. IBM DB2). In unterschiedlichem Umfang werden vordefinierte Standardmodelle und -prozesse angeboten, die dem Anwender einerseits einen Teil von Design und Implementierung der spezifischen Lösung abnehmen, ihn aber andererseits auch in ein mehr oder weniger starres Korsett zwängen.

Ausblick

Analyse- und entscheidungsunterstützende Systeme haben eine lange Historie von über 40 Jahren. Während die ersten Analysesysteme als Teil der operativen Systeme fest in diese integriert waren, wurden mit der DW-Architektur operative und dispositive Systeme strikt getrennt. Die systemtechnische Trennung von Zuständigkeiten war architektonisch ein entscheidender Fortschritt und bildete die Grundlage für umfangreiche Analysefunktionalität, z. B. Slice & Dice auf unterschiedlichsten Kennzahlen und enormen Datenmengen ohne Beeinflussung der Online Performance operativer Systeme.

Auf der anderen Seite reduziert die systemtechnische Trennung der dispositiven Daten von den operativen auch deren Aktualität. ETL-Prozesse werden häufig als nächtliche Batch-Läufe organisiert und erlauben daher nur die Tagesaktualität der dispositiven Daten. Das ist für viele Anwendungsbereiche vollkommen ausreichend, z. B. für Analysen zur Unterstützung strategischer Entscheidungen. Darüber hinaus suchen aber immer mehr Unternehmen auch nach Analysemöglichkeiten zur Unterstützung taktischer Entscheidungen des täglichen Geschäfts in Echtzeit. Dieser aktuelle Trend wird als *Real-Time-Analytics (RTA)* oder auch *Active Data Warehousing* bezeichnet. Einsatzfelder für RTA sind Geschäftsprozesse mit hohem Anteil an Interaktion mit Kunden und Geschäftspartnern, z. B. Internet, mobile Dienste und elektronischer Wertpapierhandel.

Damit die Implementierung von RTA aber architektonisch in dem Sinn kein Rückschritt wird, dass operative und dispositive Systeme wieder eine untrennbare Einheit bilden, ist eine Weiterentwicklung der Datenintegrationstechniken und -werkzeuge erforderlich. Neue Generationen von Produkten werden ihre Wurzeln sowohl in klassischem ETL als auch in Techniken der *Enterprise Application Integration (EAI)* haben [3]. Dabei bleibt die BI-Referenzarchitektur unverändert, die Aktualisierung der dispositiven Daten erfolgt jedoch bedarfsweise near-time oder real-time.

Die richtigen Entscheidungen zu treffen, sowohl strategisch als auch taktisch, ist für jedes Unternehmen essentiell. Viele Unternehmen mit starken Bestrebungen zur Verbesserung ihrer Wettbewerbsfähigkeit setzen heute erfolgreich und vielfältig Business Intelligence ein und haben BI eng mit dem Geschäft des Unternehmens integriert. Dass entscheidungsunterstützenden Systemen nach ihrer langen Historie auch noch eine lange Zukunft beschert sein wird, ist keine gewagte Prognose.

Literatur

1. Bauer, A.; Günzel H. (Hrsg.):„Data Warehouse Systeme – Architektur, Entwicklung, Anwendung“, 2. Auflage. Heidelberg: dpunkt.verlag 2004.
2. Bange, C.; Narr, J.; Keller, P.; Dahnken, O.:„BARC Studie Data Warehousing und Datenintegration. 15 Software-Lösungen im Vergleich“. München: Oxygon Verlag 2003.
3. Brobst, S.A.:„Enterprise Application Integration and Active Data Warehousing“, Proceedings of Data Warehousing 2002: From Data Warehousing to the Corporate Knowledge Center. November 12–13. Heidelberg: Physica-Verlag 2002, S 15–23
4. Bulos, D.:„A new dimension“, Database Programming & Design, 6/1996, S. 33–37
5. Chen, P.P.:„The entity relationship model – towards a unified view of data“. ACM Transactions on Database Systems 1(1): 9–36 (1976)
6. Coad, P.; Yourdon, E.:„Object-Oriented Analysis“. Yourdon, 1991.
7. Dresner, H.; Hostman, B.; Tiedrich, A.; Buytendijk, F.:„Magic Quadrants for Business Intelligence, 1H04“, Gartner Research, April 2004.
8. Ester, M.; Sander, J.:„Knowledge Discovery in Databases. Techniken und Anwendungen“. Heidelberg: Springer 2000.
9. Grothe, M.; Gentsch, P.:„Business Intelligence – aus Informationen Wettbewerbsvorteile gewinnen“. Addison-Wesley 2000.
10. Humm, B.; Zech, B.:„Architekturzentriertes Vorgehen für Integrationsprojekte“, Lecture Notes in Informatics, Tagungsband GI-Jahrestagung 2004.
11. Inmon, W.H.:„Building the Operational Data Store“. New York: John Wiley & Sons 1999.
12. Inmon, W.H.:„Building the Data Warehouse“ 2nd edn. New York: John Wiley & Sons 2002.
13. Imhoff, C.:„Mastering Data Warehouse Design“. New York: John Wiley & Sons 2003.
14. Jarke, M.; Lenzerini, M.; Vassiliou, Y.:„Fundamentals of Data Warehouses“. Berlin Heidelberg New York Tokio: Springer, Juli 2001.
15. Kimball, R.:„The Data Warehouse Toolkit: The complete guide to dimensional modelling“, 2nd edn. New York: John Wiley & Sons 2002.
16. Lehner, W.:„Datenbanktechnologie für Data-Warehouse-Systeme“. Heidelberg: dpunkt.verlag 2002.
17. Mucksch, H.; Behme, W.:„Das Data Warehouse-Konzept“. Wiesbaden: Gabler 2000.
18. MetaGroup Deutschland GmbH:„Business Intelligence –Marktanalysie und Markttrends – Deutschland 2004“, 2004.
19. „OVUM evaluates OLAP“, OVUM, 2003.
20. Pendse, N.:„The OLAP Survey 3“. München: Oxygon 2003.
21. Root, N.L.:„BI Platform Shootout“, IT View and Business View Brief, TechRanking, Forrester Research, May 2003.
22. http://www.omg.org/cwm
23. http://www.rational.com/uml
24. http://www.uml.org
25. http://www.xmla.org