Integration von ETL und OLAP in die relationale DWH-Technologie: mehr Lösung für weniger Aufwand?

Ein Datawarehouse - Praxispapier

Marc Bastien

Oracle Deutschland GmbH
Notkestrasse 15
D-22607 Hamburg
Marc.Bastien@oracle.com

Abstract: Seit vielen Jahren werden Lösungen für Datawarehouse Themen schon nach dem gleichen Schema entwickelt. Dies ist nicht weiter verwunderlich, da doch die Anforderungen –eigentlich- immer ähnlich sind: Daten aus verschiedenen Quellsystemen sollen, unter diversen Randbedingungen, analysiert werden. Auch wenn sich Technologien weiter entwickelt haben, Schnittstellen nun bereits out-of-the-box verfügbar sind und diverse grafische Oberflächen die Entwicklung und den Betrieb vereinfachen, so ist der Weg der Daten in das Warehouse bis zum Nutzer weitgehend gleich geblieben: Extrakt der Daten aus den diversen Quellen, Laden und Überführen der Daten in ein relationales Datenbanksystem, dort die Speicherung in Tabellen, eventuell Überführung aller, oder einiger Daten in eine multidimensionale Datenbank, die Speicherung in „Cubes“, die Berechnung von Kennzahlen und schließlich die Ausgabe der Daten mehr oder weniger übersichtlich an den Benutzer. Dieser Artikel beschreibt am Beispiel eines Controlling Datawarehouse, wie man durch die Integration der verschiedenen Technologien (ETL, Relational und OLAP) in einer Oracle Datenbank, Release 9*i,* bei gleichem Funktionsumfang wesentlich bei der Realisierung und Betrieb vereinfachen (=sparen) kann.

1 Einleitung

Datawarehouse (DWH) kommt nicht aus der Mode. Wurde es vor zehn Jahren noch als Erfolg gefeiert, aus einem, oder vielleicht zwei operativen Systemen Daten innerhalb einer Woche zu extrahieren, zu transformieren, in relationale Tabellen möglichst platzsparend zu laden und einem sehr ausgewählten Benutzerkreis per Text-orientierter Abfragesprache zur Verfügung zu stellen, so sind es heute andere Anforderungen, die ein DWH zu einem fachlich und technisch anspruchsvollen Projekt machen.

1.1 Die Schere wird breiter: Anforderungen an ein modernes Datawarehouse

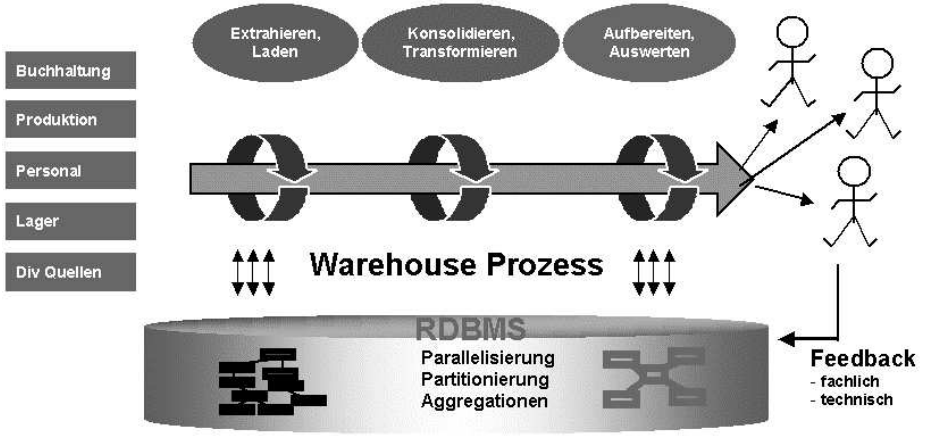
Ein modernes DWH muss vielen Anforderungen genügen. Dazu gehören natürlich alle bekannten Anforderungen, die auch in der Vergangenheit schon wichtig waren. Ich möchte diese unter dem Begriff „*Daten so bereitstellen, dass sie verstanden und analysiert werden können*“ zusammenfassen.

Dazu kommen neuen Anforderungen, die ein DWH-Projekt anspruchsvoll machen:
- Mehr Daten: DWH im zwei bis dreistelligen Terabytebereich stellen bald keine Seltenheit mehr da, häufig wird mit wenig Daten begonnen und dann gesteigert
- Mehr Benutzer: vierstellige Benutzerzahlen muss ein modernes DWH verkraften können, ohne dass die Antwortzeiten sich merklich verlängern.
- Höhere Flexibilität: neue Datenquellen sollen kürzester Zeit, zusammen mit den bisherigen Daten, den Benutzern vorliegen.
- Operationalisierung: (wie es OLTP-Systeme vormachen) in minimaler Zeit die Daten den DWH-Benutzern verfügbar machen (Stichwort: Real-Time Datawarehouse)

Aber es gibt auch gegensätzliche Trends, die einfachere DWH-Projekte erfordern
- Strengere Betrachtungen des ROI: lohnt sich ein DWH? Wenn man den Nutzen schlecht beziffern kann, wie sollen die Kosten beurteilt werden?
- Enge Personalressourcen: durch Personalreduktion in IT-Abteilungen sind weniger IT - Projektmitarbeiter verfügbar; Für ein DWH stehen nun oft die Fachanwender bereit, die einen Teil der Zeit, in der sie z.B. mit ihren Kunden im Kontakt sein sollten, für die Arbeit am DWH aufwenden.
- Kürzere Entwicklungszeiten: lange Projekte kosten Ressourcen und die Gefahr, den Projektfokus aus den Augen zu verlieren, steigt mit der Projektlaufzeit.

1.2 Der Datawarehouse Prozess

Aus den o.g. Anforderungen erwächst der Bedarf, die Vorgehensweise bei der Erstellung eines DWH zu überdenken: ist es immer sinnvoll und notwendig, alle Bereiche (ETL, Datenbank, OLAP, Endbenutzerwerkzeuge, Business Intelligence - Tools) einzeln zu betrachten? Oder kann es vielmehr sinnvoll sein, DWH als Prozess zu verstehen:



Durch diesen Ansatz wäre man nicht ständig bestrebt, für jeden der Teilbereiche die beste, zu diesem Zeitpunkt gerade verfügbare, Technologie zu verwenden, sondern könnte sich auf das Wesentliche konzentrieren, nämlich die Erstellung einer Lösung für die Anwender, mit weniger Aufwand bei der Erstellung und Wartung.
Für die Technologie, die diesen Ansatz unterstützen soll, heißt dies, dass sie mit Hilfe von Metadatenrepositories und verbundenen Schnittstellen (am besten integriert) agieren muss: Metadaten müssen durch ein Werkzeug angelegt werden, und allen anderen Komponenten verfügbar gemacht werden. Wenn erst einmal die verschiedenen Bereiche („Stage“, „Warehouse“, „Mart“) und deren Abhängigkeiten (Verbindungen) innerhalb des DWH durch Dimensionen und Kennzahlen aber auch durch Tabellen, Views, Prozeduren usw. beschrieben sind, können die alle weiteren Prozesse innerhalb des DWH leicht abgeleitet und generiert werden. Erweiterungen oder Änderungen werden dann im Repository vorgenommen, generiert und kurze Zeit später dem Endbenutzer zur Verfügung gestellt. Ebenso wird die Portierung auf andere Umgebungen erleichtert: die Metadaten beschreiben das DWH plattform-unabhängig und ermöglichen quasi ein portables Warehouse.

2 Der Praxisfall

2.1 Beschreibung des Problems

In einem internationalen Großkonzern im Bereich Maschinenbau wurde das unternehmensweite Controlling ausschließlich auf der Basis von Microsoft Excel bzw. integrierte Plug-Ins zu SAP R/3 durchgeführt. Kennzahlen wurden durch die Controller in den Berichten in MS Excel als Makros implementiert. Die erzeugten Berichte wiesen z.T. Fehler auf, waren zu spät und nur einem eingeschränkten Benutzerkreis zugänglich. Sollten neue Berichte erzeugt werden, mussten sich mehrere Controller tagelang nur mit der Erstellung der Berichte und der entsprechenden Formatierung beschäftigen. Die gedruckten Berichte wurden per Post innerhalb und außerhalb Deutschlands verteilt. Es liegt auf der Hand, dass der deutsche Vorstand seine Berichte eher verfügbar hatte, als sein südafrikanischer Kollege.

2.2 Das Projekt

Im Rahmen eines konzernweiten Projektes wurde die Einführung eines unternehmensweiten Controlling DWH beschlossen. Wesentliche Eigenschaften sollten sein:

- geprüfte Routinen für das Einladen der Daten aus SAP R/3 in das DWH, kein manueller Eingriff, lückenlose Verfolgung der Routinen
- DWH-gerechtes Datenbank Schema, ausbaubar, pflegbar, nutzbar
- Abdeckung neue Anforderungen an das Reporting:
  - Neue, umfangreiche Kennzahlensysteme (GuV, Erfolgsrechnungen)
  - Möglichkeit zur Prognose
  - Erweiterte Analysemöglichkeiten
  - Speicherbare Auswertungen
  - Zeitreihenbetrachtung mit verschiedenen Szenarien
- Konzernübergreifendes Repository für Kennzahlen, Benutzer, Benutzerrechte und Standardberichte
- Zugriff über das Intranet auf
  - Standardberichte: druckbar, mit einheitlichem Layout, anhand von Parametern veränderbar, auf Knopfdruck verfügbar (Zielgruppe: bis zum Top-Management)
  - Ad-Hoc Berichte: Mehrdimensionale Selektion der Daten und Berichte, auch dort u.a. vordefinierte Berichte, die aber individuell gestaltbar sein sollen (Zielgruppe: Controller und Analysten aus allen berechtigten Bereichen)

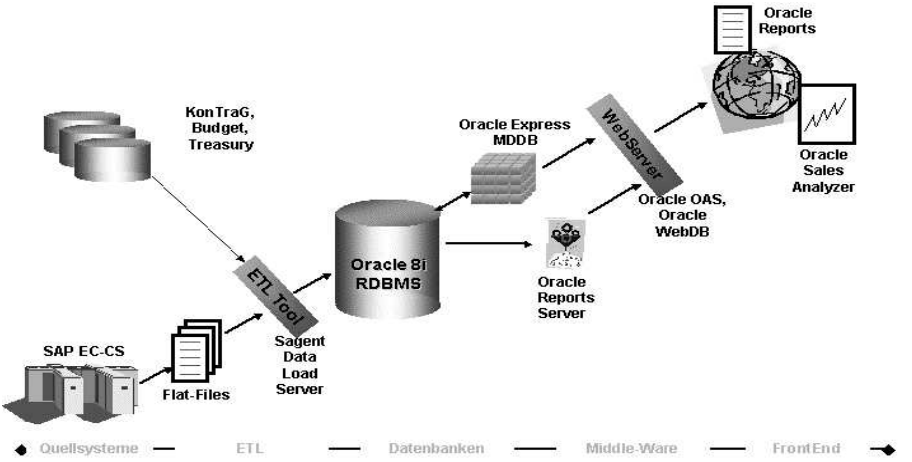
2.3 Die realisierte Lösung

2.3.1 Beschreibung

Es wurde ein zentrales, Intranet basierendes Management Information Portal für konsistentes Reporting, Benutzergruppen-abhängigen Inhalt inklusive neuer Kennzahlen und Berechnungen für Management und Mitarbeiter des Controlling (ca. 500 Benutzer) erstellt. Für die Auswahl der Werkzeuge wurde ein Proof-of-Concept (PoC) mit namhaften Anbietern von Datawarehouse Lösungen durchgeführt. Schon im Vorfeld wurde sich gegen eine Implementierung von SAP BW entschieden, da nicht alle Anforderungen erfüllt wurden.
Als Sieger aus dem PoC gingen folgenden Hersteller mit ihren Werkzeugen hervor:

- ETL Prozess: Sagent Data Load Server 4.2
- Datenspeicherung (relational): Oracle RDBMS 8.1.6
- Datenspeicherung (multidimensional): Oracle Express 6.3
- Middleware, WebServer: Oracle Application Server 4.0.8.1
- Auswertung, Portal: Oracle WebDB 2.2
- Auswertung, Standardberichte: Oracle Reports 6i
- Auswertung, Ad-Hoc: Oracle Sales Analyzer 11i

2.3.2 Lösungsarchitektur (Hardware/Betriebssystem): Sun / Solaris 4.6



2.3.3 Datenfluss

- die Daten werden durch SAP EC-CS als ASCII-Dateien bereit gestellt
- die Dateien werden durch das ETL Werkzeug Sagent erkannt und in die „Staging-Area“ der Oracle Datenbank geladen
- die Daten werden bereinigt, aufbereitet und in das DWH Schema überführt
- Daten werden z.T. in die mehrdimensionale Datenbank Oracle Express kopiert
- In Express wird die Verdichtung über alle Hierarchien (Aggregation), sowie die Berechnung aller Kennzahlen durchgeführt
- Die verdichteten Daten und die berechneten Kennzahlen werden aus Express zurück in die relationale Datenbank exportiert
- Zugriff auf die mehrdimensionalen Daten durch Oracle Sales Analyzer (Ad-Hoc Reporting) und Zugriff auf die relationalen Daten durch Oracle Reports (Standardberichte)

2.3.4 Vorteile dieser Lösung

Durch den Einsatz der Oracle RDBMS war zu keinem Zeitpunkt in Frage gestellt, ob das aufkommende Datenvolumen aufgenommen werden konnte, oder nicht. Durch die zusätzliche mehrdimensionale Datenhaltung ergänzte man die Vorteile der relationalen Datenhaltung um die Möglichkeit, Daten auch multidimensional auszuwerten. Die multidimensionale Datenbank Express lieferte durch die Applikation Oracle Sales Analyzer (OSA) eine so flexible Analyse, wie sie mit rein relationalen Werkzeugen nicht möglich war. Durch die Nutzung von OSA war die Speicherung der Daten in einer MDDB vorgegeben, deshalb konnten die bereits in Oracle Express enthaltenen Prozeduren zur Aggregation über mehrdimensionale Hierarchien und Kennzahlenberechnung genutzt werden und man brauchte keinen Aufwand in die komplizierte Entwicklung der Kennzahlenberechnung und Aggregation in der RDBMS stecken.
Durch die bereits reichlich vorhandenen analytischen Funktionen und Programmen konnte auf umfangreiche Entwicklung in Oracle Express verzichtet werden, nur für die Steuerung der Berechnungen und für den Import und Export mussten Programme entwickelt werden. Dabei erwies es sich als Vorteil, dass direkt in der MDDB Programme erstellt werden konnten, die in der Endausbaustufe über UNIX-Scripts direkt aus Sagent aufgerufen werden konnten. So war durchgängig abgesichert, dass alle Prozesse in der richtigen Reihenfolge ablaufen und, wenn nicht, eindeutig den Stand der Verarbeitung bestimmt werden konnte. Die Geschwindigkeit der Kennzahlenberechnung und der Aggregation war trotz der hohen Datenmengen äußerst zufriedenstellend.

2.3.5 Nachteile dieser Lösung

Architektur: es waren zwar, wie damals häufig üblich, für alle Teilbereiche optimale Lösungsbausteine gewählt, an deren individuellen Leistungsfähigkeit auch nichts auszusetzen war, aber durch die zahlreichen, nicht genormten Schnittstellen gingen gewonnene Vorteile verloren, und die Komplexität des Projektes stieg deutlich. Das Projekt dauerte länger und erforderte mehr Skills.
Metadaten: Es gab in dem ETL Werkzeug nicht die Möglichkeit, standardisierte Warehouse-Metadaten abzulegen. Dies führte dazu, dass in der relationalen Struktur nicht erkennbar war, welche Dimensionen oder Fakten vorhanden waren. Alle Informationen dazu waren nur in den Köpfen der Entwickler bzw. in der Dokumentation nachzulesen. Die Projektlaufzeit verlängerte sich, da einige Schnittstellen häufig umgeschrieben werden mussten, weil sich Entwickler missverstanden.
Datensicherung: Durch die völlig unterschiedliche Technologien der RDBMS und MDDB mussten Backup und Recovery auch hier komplett getrennt betrachtet werden. Abweichungen führten z.T. dazu, dass die MDDB nicht zurückgesichert werden konnte, sondern komplett neu aus der RDBMS befüllt werden musste.
Sicherheit: durch die unterschiedlichen Sicherheitskonzepte von RDBMS und MDDB mussten Programme entwickelt werden, die die Benutzerrechte aus einem zentralen Repository lesen und einsetzen konnten.

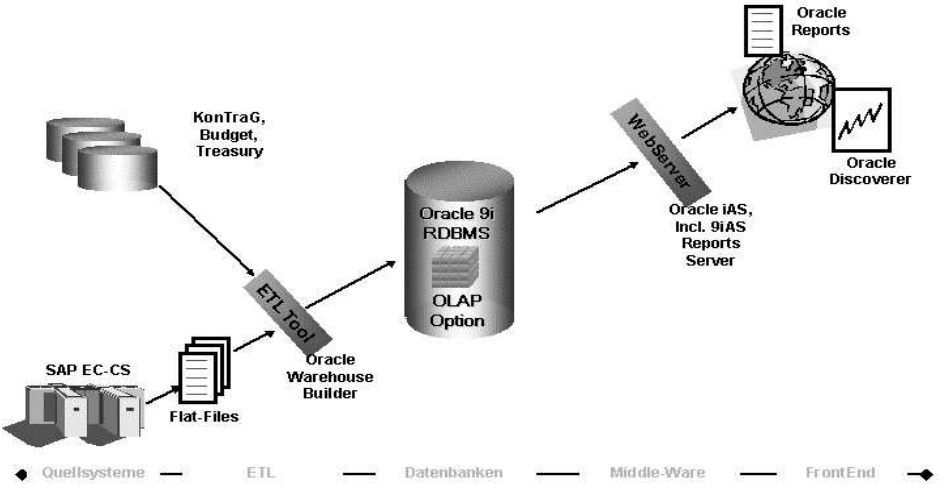
2.4 Eine alternative Lösung

2.4.1 Beschreibung

Für die alternative Lösung ist zu beachten, dass, während die Nachteile reduziert, oder eliminiert werden, die Vorteile der o.g. Lösung erhalten bleiben. Durch die Weiterentwicklung der DWH-Funktionalitäten in der Oracle 9*i* Datenbank und die Entwicklung von passenden Werkzeugen ist es möglich, einen komplexen Warehouse Prozess zu implementieren, und nicht Einzelkomponenten miteinander zu verbinden.
Die Lösungskomponenten der neuen Lösung:

- ETL Prozess: Oracle Warehouse Builder 9i
- Datenspeicherung (relational + OLAP): Oracle RDBMS 9i + OLAP Option
- Middleware, WebServer: Oracle 9i Application Server 9i
- Auswertung, Standardberichte: Oracle Reports 9i
- Auswertung, Ad-Hoc: Oracle Discoverer 9i

2.4.2 Lösungsarchitektur der Alternative



2.4.3 Datenfluss

- die Daten werden durch SAP EC-CS als ASCII-Dateien bereit gestellt
- die Dateien werden durch das ETL Werkzeug Oracle Warehousebuilder erkannt und in direkt in das DWH-Schema geladen. Während dieses Prozesses werden die Daten bereinigt und aufbereitet
- Teile der Daten werden in den „Analytic Workspace“ (den mehrdimensionalen Bereich) der relationalen Datenbank überführt
- Im Analytic Workspace wird die Verdichtung über alle Hierarchien, sowie die Berechnung alle Kennzahlen durchgeführt
- Auf die Daten in den relationalen Tabellen und im Analytic Workspace kann direkt mit allen SQL-Werkzeugen zugegriffen werden

2.4.4 Vorteile der neuen Lösung

Die grundsätzliche Architektur bleibt unverändert, weshalb es auch nicht verwundert, dass die Vorteile sich nicht verändern: Datenspeicherung relational und mehrdimensionale Kennzahlenberechnungen.
Zusätzlich ergeben sich aber neue Vorteile:
Metadaten: der Warehousebuilder erzeugt beim Anlegen des Schemas und der Verbindungen zwischen Quelle und Ziel bereits alle Metadaten für das DWH. Die Metadaten sind durchgängig und vom ETL-Prozeß bis zur Auswertung nutzbar. Für die wenige Ausnahmen sind fertige „Bridges" definiert, die die zusätzlichen Daten automatisch erzeugen. Die durchgängigen Metadaten ermöglichen zusätzlich eine Herkunftsanalyse, d.h. es kann jederzeit ermittelt werden, aus welcher Quelle welche Daten stammen.
eine integrierte Technologie: alle Daten befinden sich in einer Datenbank, d.h. die Metadaten, die Tabellen mit den Warehouse Daten und selbst die mehrdimensionalen Daten sind in einer DB gespeichert. Dies bedeutet nicht nur Vorteile bei der Wartbarkeit oder der Performanz, sondern auch, dass jetzt nur ein Backup/Recovery Konzept entwickelt und umgesetzt werden muss: wird die Datenbank gesichert, werden alle Daten, egal, ob relationale Tabellen, mehrdimensionale Objekte oder Metadaten gesichert. Des weiteren bietet die integrierte Technologie natürlich auch ein integriertes Benutzerkonzept, d.h. es gibt keine Notwendigkeit für eine doppelte Benutzerverwaltung in RDBMS und MDDB.
Integriertes OLAP bedeutet auch, dass die Daten des OLAP Würfels nun allen Benutzern zur Verfügung stehen. Es werden keine speziellen Applikationen benötigt, da alle Daten per SQL im Zugriff sind. Vorher benötigte Schnittstellen von der MDDB zur RDBMS können entfallen.

2.4.5 Mögliche Nachteile der neuen Lösung

Migration: der Aufwand für die Migration ist im Abschnitt „Aufwand für die Migration".
Frontend: Oracle Sales Analyzer (OSA) basiert auf der Express Technologie und kann nicht zusammen mit 9i OLAP eingesetzt werden. Es muss auf eine andere Applikation, den Oracle Discoverer, zurückgegriffen werden, was ich hier unterstellt habe. Dies würde eine Umstellung der Endanwender erfordern.
Alternativ könnte eine eigene Applikation erstellt werden, die OSA gleicht. Dazu böte sich der Oracle JDeveloper an, der, zusammen mit den Oracle BI-Beans, mit fast gleichen Analysemöglichkeiten (verglichen zu OSA) aufwarten kann.
Beide Werkzeuge setzen natürlich auf den Metadaten vom Warehousebuilder auf.

2.4.6 Aufwand für eine Migration

Der Aufwand einer Migration kann in mehrere Teile zerlegt werden:

- ETL: der komplette ETL Prozess muss neu im Oracle Warehousebuilder erstellt werden. Falls die vorher verwendete Lösung standardisierte Metadaten (CWM) unterstützen würde, wäre ein Import möglich.
- Datenbank: kaum Aufwand, da nur ein Upgrade von 8i nach 9i erfolgen muss
- OLAP: alle Programme und Daten können nach 9i OLAP übernommen werden, allerdings müssen Anpassungen auf die neuen Metadaten vorgenommen werden
- Middleware: ein Upgrade von Oracle OAS nach iAS ist möglich
- Frontend
    - Oracle Reports: Upgrade nach 9i ist mit wenig Aufwand möglich
    - Oracle Sales Analyzer nach Discoverer: alle Berichte müssen neu erstellt werden
    - (alternativ) OSA nach eigene Applikation: eine komplette Applikation kann mit dem JDeveloper und BI-Beans neu entwickelt werden

3 Fazit

Für den konkreten Fall ist sicherlich entscheidend, wie hoch der tatsächliche Aufwand für die Migration wäre, schließlich sind bereits alle Applikationen und Schnittstellen implementiert. Wahrscheinlich bietet es sich an, die Migration (oder Teile hiervon) im Rahmen von sowieso geplanten Änderungen oder Upgrades an der Lösung durchzuführen.

Für alle neuen Projekte zeigt dieses Beispiel wie viel leichter, und damit billiger, sich DWH – Projekte sich mit den Komponenten:

1) integrierte Metadaten
2) integrierte Datenbanktechnologie
3) angepasste Tools für ETL und BI

durchführen lassen können. Schließlich hat sich gezeigt, dass nicht Hard- und Software den Preis eines Projektes maßgeblich beeinflussen, sondern Dauer und Komplexität der Implementierung und des Betriebes.