Seminararbeit in der Seminarreihe:

„Business Intelligence - Teil I:

OLAP & Datawarehousing“

- Im Lehrgebiet Datenverwaltungssysteme -

- Im Sommer Semester 2003 -

„ OLAP & DATAWAREHOUSING

P R O D U K T E “

von

Oliver Parti

Matrkln. 334558 / Studiengang Angew. Informatik

Betreut von

Herrn Prof. Dr.-Ing. Dessloch

GLIEDERUNG

1.Business Intelligence , OLAP & Datawarehouses – Definitionen und Produktmerkmale

- Was ist Business Intelligence ?
- OLAP- Definition und Kriterien zur Produktevaluation.
- OLAP Kategorien - MOLAP/ROLAP/HOLAP
- Datawarehouse – Definition und Konzepte

2.Der Markt – aktueller Markt und historische Eckpunkte

- Der Verlauf bis heute und seine Eckpfeiler.
- Der aktuelle Markt & die wichtigsten Produkte

3.aktuelle Produkte – am Beispiel & im Vergleich

- Hyperion Essbase
- MIcroStrategy 7i
- ORACLE 9i
- weitere Produkte – weitere Produkte mit Marktbedeutung

4. Benchmarks & Leistungsvergleiche

- OLAPCOUNCIL Benchmark

5.Anhang

- Quellenverweise

KAPITEL 1 – BI,OLAP & DW's

Was ist Business Intelligence ?

Die Frage nach einer Definition steigt wahrscheinlich exponentiell mit der umfassenden Bandbreite des zu definierenden Begriffs, auch hier gestaltet sich eine präzise Definition mehr als schwer.

Dies liegt bestimmt nicht zuletzt daran dass jeder Hersteller so ein wenig seine eigene Interpretation dieses Begriffs und seiner vermeintlichen Inhalte bietet. Dies ist sicherlich ein Problem das auf die ebenfalls in diesem Kapitel angesprochenen Begriffe genauso zutrifft. Ganz besonders lassen sich auch hier schon gewisse Unterschiede in den Philosophien der Anbieter solcher Lösung erkennen – dazu später aber im Detail.

Doch nun wollen wir uns doch mal vergegenwärtigen, was denn all die unterschiedlichen Interpretationen gemein haben. Grob gesagt wurde die Business Intelligence aus der Idee geboren, dass die verwalteten Daten mehr sein können als die Abwicklung des Tagesgeschäftes. Die richtige Art der Datenverwaltung (=> Datawarehouses) gekoppelt mit einer effizienten und vor allem zügigen Art der Auswertung (=> OLAP) soll viel mehr dem Management ermöglichen, Tendenzen, Lücken etc. zu erkennen, oder sogar Expansionsmöglichkeiten, zum Beispiel bezüglich der Produktpalette, zu evaluieren.

Also findet sich die Business Intelligence irgendwo im Umfeld der Entscheidungsfindungs-Tools fürs Management wieder. Man muss allerdings klar sagen, dass heutzutage eben nicht nur das Management in der Lage sein muss auf Datenbeständen möglichst schnell eine Analyse ausführen zu können. Zudem muss für alle möglich sein, dies von einem variablen Punkt, in der Zeit und im Ort, tun zu können.Dies bringt uns direkt zum nächsten Abschnitt.

OLAP – Definition und Kriterien zur Produktevaluation.

Online **A**nalytical **P**rocessing ist das Werkzeug der Business Intelligence womit eine höchst mögliche Flexibilität in Bezug auf die Realisierung von Anfragen und Analysen an den aktuellen Datenbestand ermöglicht werden soll.

Um dies in einer angemessen Zeit und mit entsprechender Sicherheit tun zu können, benötigt ein OLAP-Tool nicht nur die Infrastruktur eines gegeben modernen Datawarehouses - zum Beispiel gehört zu dieser auch ein hoher Grad an Vernetzung- sondern auch eigene Konzepte zur Datenhaltung sind nötig.

Bei dem angesprochenen Konzept handelt es sich um das Modell eines multidimensionalen Datenwürfels, das je nach Realisierung der Datenbank, vom OLAP Tool aus den vorhandenen relationalen Daten erstellt wird. oder eben direkt genutzt wird. Nur die Eigenschaften dieser Art der mehrdimensionalen Datenhaltung, die oftmals als **das** Charakteristikums eines OLAP-Tools gehandhabt wird, garantiert im Zusammenhang mit einer strukturabhängigen Indexierung des Datenwürfels die benötigte Geschwindigkeit für die zumeist komplexen ad-hoc Anfragen.

OLAP ist klar von OLTP abzugrenzen (Online Transaction Processing) , da hier komplett unterschiedlich Optimierungsschwerpunkte existieren. Steht bei OLTP die möglichst effiziente Verarbeitung von Transaktionen im Vordergrund , so liegt das Hauptgewicht von OLAP Tools bei der zur minimierenden Bearbeitungszeit komplexer Analyseanfragen.

Der Begriff OLAP wurde hauptsächlich von E.F.Codd geprägt, die von ihm aufgestellten 12 Regeln sind nicht nur hinsichtlich einer Begriffsdefinition oder einer Produkteinordnung maßgeblich, sie sind auch immer noch bei einer Produktbewertung von größter Bedeutung. Aus diesen Gründen wollen wir die besagten Regeln hier nennen & kurz erklären.

Die 12 Regeln von E.F.Codd zur Produktbewertung im OLAP-Bereich sind:

1.Multidimensional conceptual view. Diese Funktion ist wichtig in Bezug auf das „slice & dice“ aus der EIS, und wird vor allem benötigt um finanzielle Zusammenhänge zu verdeutlichen.

2.Transparency. OLAP Systeme sollten Teil eines offenen Systems sein, das heterogene Datenquellen nutzt. Sicherlich sollte der Endnutzer eben nicht mit Details der Datenbeschaffung oder Konvertierung belästigt werden.

3.Accessibility. Um die Verständlichkeit für den Endnutzer zu erleichtern sollte ihm das OLAP System alle Daten in genau einem logischen Zusammenhang präsentieren, und ermöglichen darin zu agieren.

4.Consistent reporting performance. Die Leistung darf nicht negativ von einer steigenden Anzahl in den Dimension des Models beeinflusst werden.

5.Client/server architecture. Eine Notwendigkeit um ein offenes modular gestaltetes System zu ermöglichen.

6.**Generic dimensionality.** Die Dimensionierung ist weder auf 3 Dimensionen limitiert, noch auf eine spezielle Dimension bezogen. Funktionen die man für eine Dimension einsetzt, sollen auch für andere Dimensionen einsetzbar sein.

7.**Dynamic sparse-matrix handling.** OLAP-Systemen muß es möglich sein, sich variabler Möglichkeiten der Datenhaltung & -verwaltung zu bedienen, und unter Einsatz von Kompressionsmöglichkeiten für große Datensätze Elementarwürfel zu verwalten, in der nicht in jeder Zelle Daten vorhanden sind.

8.**Multiuser support.** OLAP-Systeme sollten in der Lage sein, mehreren Nutzern paralleles Arbeiten zu ermöglichen, inklusive individueller Verwaltung der Sichten und der genutzten Teile der gemeinsamen Datenbank, sowie Mechanismen zur Erhaltung der Konsistenz bereitzustellen.

9.**Unrestricted cross-dimensional operations**. Ähnlich Regel 6. Alle Dimension sind gleich gestaltet, interdimensionäre Operationen sollten ohne zellenspezifische Einschränkungen möglich sein.

10**. Intuitive data manipulation.** Möglichst komfortabel & einsichtig gestaltete Endnutzerschnittstelle. Das Auffinden logisch verwandter Daten sollte ohne große Menübedienung oder ähnliches möglich sein.

11**. Flexible reporting.** Um flexible Berichterstattung zu ermöglichen ist es notwendig, verschieden Zugriffswerkzeuge sowie Montage- und Anpassungstools bereitzustellen. Benutzer sollten möglich einfach in die Lage versetzt werden, genau das zu finden was für Sie relevant ist.

12. **Unlimited dimensional and aggregation levels.** Ein ernstzunehmendes OLAP-Tool sollte zumindest 15, besser 20 Dimensionen und Aggregationsebenen unterstützen.

OLAP Kategorien - MOLAP/ROLAP/HOLAP

Nach dem wir nun eine Definition von OLAP versucht haben, wollen wir uns den bereits erwähnten unterschiedlichen Kategorien der OLAP Werkzeuge zuwenden.

Die Klassifikation basiert auf der Architektur der zugrunde liegenden Datenbank. Dementsprechend existiert eine **R**elationales, ein **M**ehrdimensionales, und als

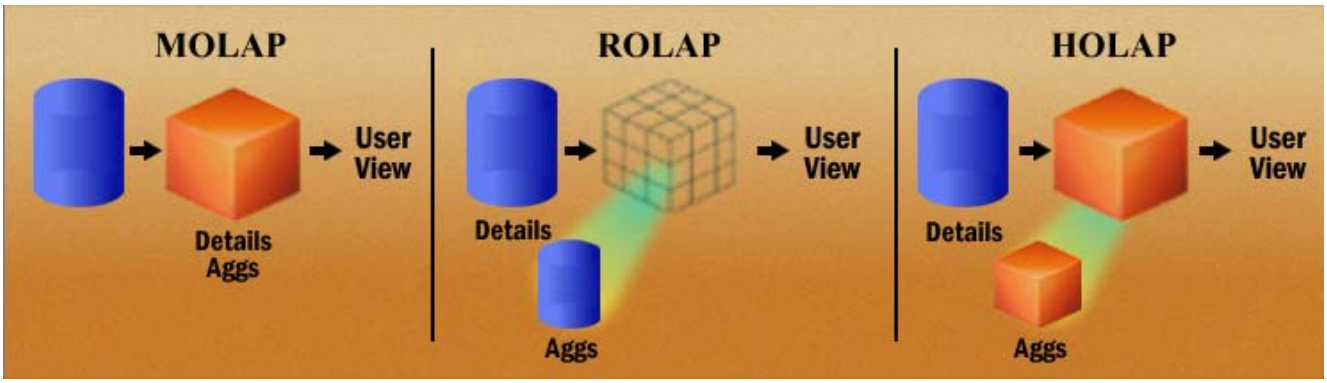
Zwischenlösung ein **H**ybrides OLAP. MOLAP verwenden also entsprechend mehrdimensionale Datenbanksysteme, die Daten werden meist ihrer voraussichtlicher Verwendung nach gespeichert und zusammengefasst, was eine Optimierung des Datenvolumens begünstigt, und eine hervorragende Analyseleistung ermöglicht, falls die Daten nicht für andere Fragestellungen als vorgesehen genutzt werden sollen. Sollte dies aber der Fall sein, so schlägt der Vorteil in einen Nachteil um, für nicht vorgesehene Anfragen müssten der Datenbestand erneut strukturiert werden, bzw. die Analyseleistung würde rapide fallen. MOLAP-Tools sind meist im Vergleich sehr kostenintensiv und komplex was die erforderliche Unterstützung angeht.

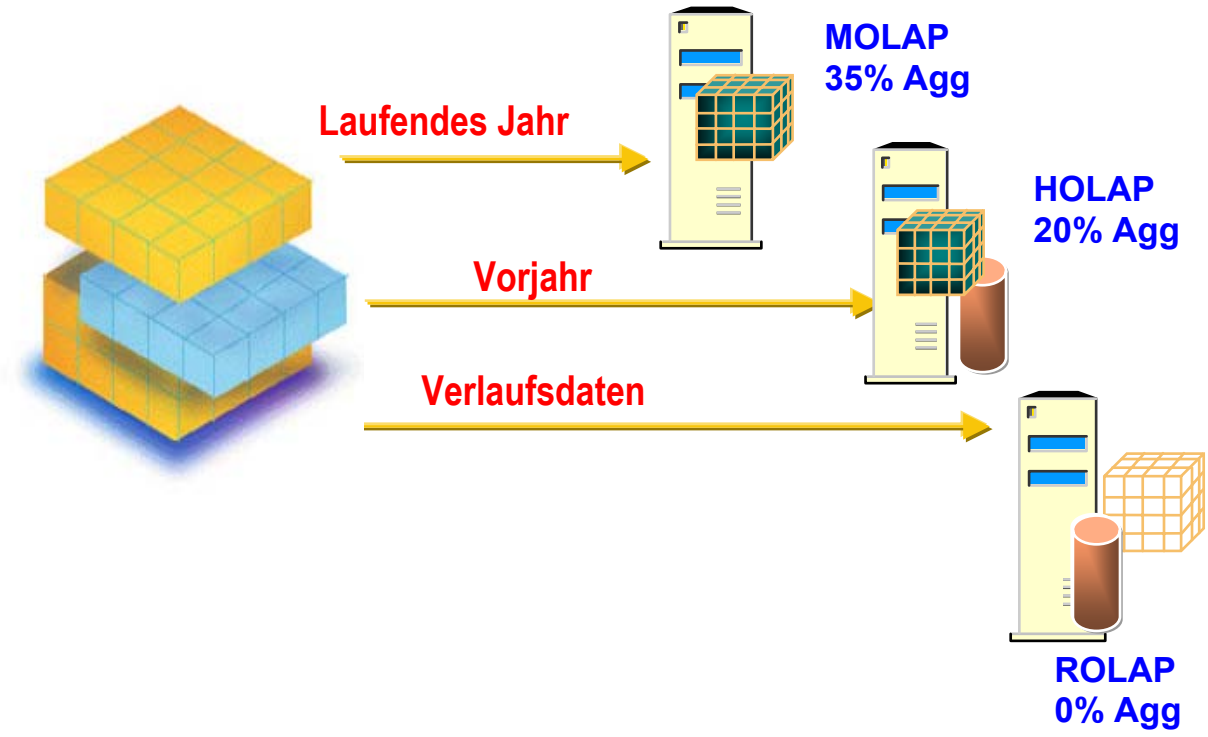
Die am weitverbreitetsten OLAP-Werkzeuge verfolgen einen ROLAP-Ansatz. Basierend auf einer Relationalen Datenbank verwenden diese Werkzeuge eine Metaschicht. Damit wird die Notwendigkeit umgangen, eine mehrdimensionale statische Struktur zu errichten, wie bei MOLAP üblich. Es ist einsichtig, dass sich Performanznachteile bezüglich der Anfrageverarbeitung im Vergleich zu einem dedizierten MOLAP-Ansatz ergeben. Anderseits eine stark erhöhte Flexibilität und Performanzvorteile für wechselnde Anfragen und Analysen.

Hier liegt das Problem in der Entwicklung geeigneter Middleware, die die gespeicherten zweidimensionalen Beziehungen in eine mehrdimensionale Struktur umwandelt. Ein großer Vorteil stellt dagegen die gute Zugänglichkeit und die guten Möglichkeiten zur Verwaltung der meist immensen Datenmengen dar.

Wie erwartet versucht der HOLAP Ansatz die positiven Eigenschaften der beiden anderen Modelle miteinander zu verbinden. HOLAP basiert auf einer multidimensionalen Datenbank die nicht auf spezielle Anfragen „zurechtgestutzt“ wurde, und verwendet wie ROLAP Metadaten zur Auswertung.

Die unten eingefügten Grafiken sollen die erwähnten Sachverhalte ein wenig verdeutlichen:





Wie man anhand der Grafiken gut erkennen kann hängt der Anteil an Aggregationen stark davon ab wie sehr die Datenbank multidimensional ist. Das bedeutet also bei relationaler Datenhaltung keine Aggregationen , ein hoher Prozentsatz von Aggregationen hingegen beim MOLAP-Ansatz, ein mittlerer beim HOLAP Ansatz.

Datawarehouse – Defintion und Konzepte

Ein wenig haben wir der Definition schon vorausgegriffen. Wir wissen inzwischen, dass es Datawarehouse Produkte und damit verbundene Philosophien gibt, die auf relationale Datenhaltung setzten , und andere die auf das multidimensionale Modell setzen. Außerdem ist uns auch schon bekannt, dass ein modernes Datawarehouse eine geeignete Infrastruktur benötigt, die u.a. Vernetzung und Client - Server Architektur umfasst. Dadurch , und durch die Unterstützung komplexer Anfragewerkzeuge wird nicht nur der Einsatz von OLAP-Tools ermöglicht, sondern rechtfertigt ein Datawarehouse die Zuordnung zur BI bzw. indirekt zu den Entscheidungsunterstützenden Systemen. Auch hier kann man, ähnlich zu OLAP eine klare Abgrenzung zum tagesgeschäftsseitigen OLTP vornehmen. Auch hier stehen nicht die Anzahl der bearbeiteten Transaktionen im Vordergrund. Was für OLAP E.F.Codd ist für das Datawarehousekonzept Bill Inmon, der sich bei seiner Definition mit 4 nötigen Eigenschaften eines Datawarehouses begnügte:

1. **Subjekt/Themenorientierung.** Da das DW eher entscheidungsunterstützende Daten als anwendungsorientierte speichert, d.h. rein operationale Daten werden nicht berücksichtigt. Vielmehr erfolgt im DW eine inhaltliche Konzentration auf die zentralen Objekte einer Unternehmung , wie z.B. Produkte und Kunden.

2.**Integration/Vereinheitlichung.** Durch das Zusammenfließen der Daten aus allen Anwendungszweigen des Unternehmens , sind die erhaltenen Daten oftmals in sehr unterschiedlichen Formaten dargestellt. Um Inkonsistenzen zu vermeiden, bzw. um Konsistenz herzustellen müssen die Daten in ein einheitliches Format überführt werden.

3.**Zeitorientierung.**Die Daten sollen im DW eine Folge von zeitlichen Schnappschüssen darstellen, d.h. die Daten sind nur in einem bestimmten Zeitraum korrekt und relevant.

4.**Beständigkeit.**Die Beständigkeit wird gewährleistet durch die Tatsache, dass die Daten nicht mit Hilfe direkter Mechanismen aktualisiert werden aus den operationalen Daten. Neue Daten werden immer erst mal der Datenbank hinzugefügt, und nicht direkt als Ersatz für bereits vorhandene Daten behandelt. Die Integration folgt erst später und schrittweise.

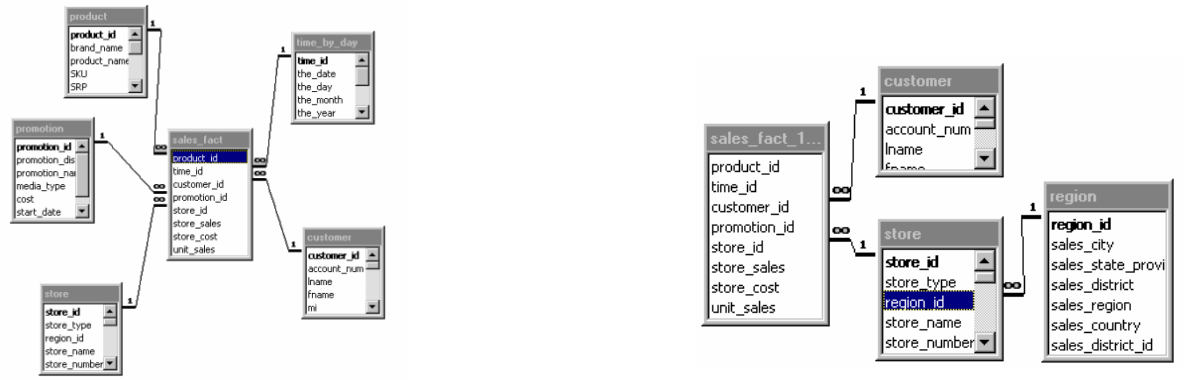
Was in Bezug auf ein Datawarehouse noch Erwähnung finden sollte ist die logische Struktur in denen die Daten abgelegt werden, die natürlich auf eine möglichst effiziente Analyseverarbeitung ausgerichtet sind. Dies wird ermöglicht indem man berücksichtigt, dass viele Anfragen sich auf Fakten beziehen. Es gibt drei Schemata:

1.**Star-Schema.** Im Mittelpunkt steht eine Faktentabelle umgeben von Referenzdaten enthaltenden Dimensionstabellen. Die Faktentabelle besitzt für jede der Dimensionstabellen einen entsprechenden Fremdschlüssel.

2.**Snowflake-Schema.**Das Snowflake Schema ist eine Variante des Star Schemas, hier ist es jeder Dimensionstabelle erlaubt eigene Dimensionstabellen zu haben.

3.**Star-Flake-Schema.**Diese Mischung aus Snowflake und Star Schema hat sich als das für die Entscheidungsunterstützung beste Datenbankschema herausgestellt.

Star –Schema im Vergleich zum Snowflake-Schema



KAPITEL 2 – Der Markt

Der Verlauf bis heute und seine Eckpfeiler

In diesem Kapitel wollen wir uns mit dem Markt für unsere Systeme befassen. Wichtig ist zu wissen, welches die marktbeherrschenden Produkte und Unternehmen sind.

Der Markt heute ist sicherlich eben auch nur ein Produkt der vorangegangen Dekaden was Produktlandschaft ,Firmenlandschaft sowie technologische Entwicklungen und Philosophien entspricht. Aus der bisherigen Geschichte kann man einige Rückschlüsse ziehen.

Die multidimensionalen Anwendungen als solches werden ihren Bestand haben, selbst wenn sie zur Zeit ein Nischendasein fristen. Ihnen wird das größte Wachstum prognostiziert, denn die Hardware Entwicklungen werden es ermöglichen die Kosten zu senken.

Einfache und günstige OLAP Produkte sind auf dem Markt viel leichter an den Mann zu bringen als vergleichsweise bessere aber eben auch somit komplexere und teuere.

Im folgenden möchte ich einen Verlauf in Stichpunkten angeben der die wichtigsten Produkte nennt und kurz darauf eingeht welchen Einfluss die Veröffentlichung dieses Systems auf den weiteren Verlauf und andere Produkte hatte.

- **1962** ***APL.*** A Programming Language. Erste Multidimensionale Sprache. Man kann APL nicht zu den OLAP-Werkzeugen selbst zählen, aber die dort verwirklichten Ideen haben generellen Einfluss gehabt. Außerdem verwenden manche Produkte intern immer noch APL, so z.B. Adaytum e.Planning Analyst , oder Cognos Finance.

- **1970** ***Express.*** Erstes multidimensionales Tool welches sich auf Marketingaufgaben spezialisiert hatte. Die modernisierte Version ist heute als MOLAP Engine im neuen ORACLE 9i Release2 im Einsatz.

- **1982** ***Comshare System W.*** Erstes OLAP-Werkzeug entwickelt für Anwendung des Finanzmarktes. Nicht länger in Gebrauch, hatte aber starken Einfluss auf Essbase.

-**1984** ***Metaphor.*** Erstes relationales OLAP. Wegen hohen Preisen war der Erfolg dieses Produktes sehr limitiert, doch durch die hohe Kundenloyalität ist es noch am Leben.

-**1985** ***Pilot Command Center.*** Erstes OLAP das einen Client-Server Architekturansatz verfolgte.

-**1990** ***Cognos Power Play.*** Erstes OLAP-System für den Desktop bzw. Windows. Marktführendes Produkt im Desktop Bereich.

-**1992** ***Essbasse.*** Erstes ordentlich vermarktetes OLAP-Produkt. Marktführender OLAP-Server seit 1997.

-**1994** ***MicroStrategy DSS Agent.*** Erstes ROLAP ohne Multidimensionale Engine. Eine Anwendung für sehr große Datenbanken.

-**1996** ***Busines Objects 4.0.*** Erstes Werkzeug das die Berichterstattung erlaubt aufgrund eines „desktopcubes“ , der aus relationalen Daten erzeugt wurde.

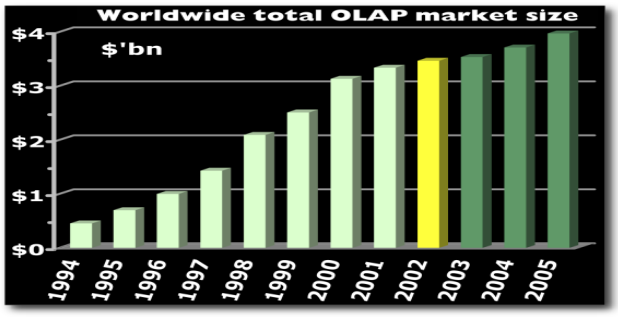
-**1997** ***Microsoft OLE DB for OLAP.*** Standard OLAP-API für die Industrie.

- **1998** ***IBM DB2 OLAP Server***. Diese Version von Essbase speichert alle Daten in einem Stern-Schema, in einer Relationalen Datenbank. Es ist nicht in DB2 integriert.

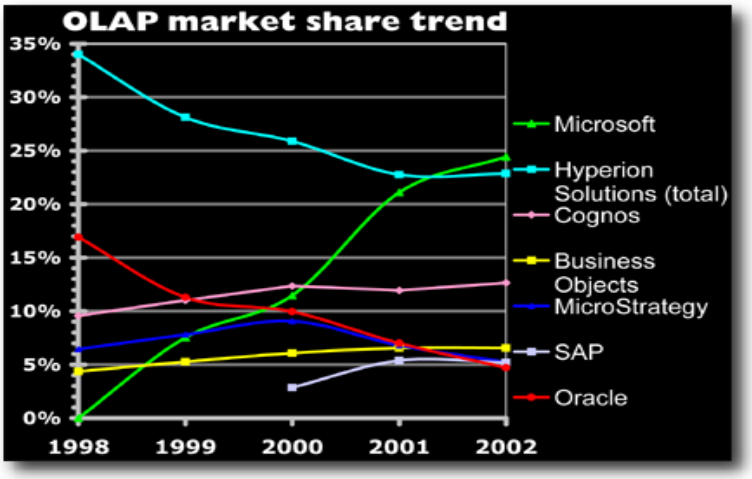
- **2000** ***XML for Analysis.*** Eine XML Implementierung von OLE DB for OLAP.

Der aktuelle Markt

Nachdem wir nun einige Meilensteine auf dem Weg zum heutigen Marktgeschehen betrachtet haben, wollen wir uns dieses selbst in Zahlen und Fakten anschauen. Tatsächlich gibt es in unserem Markt mehr als 30 Anbieter. Der Markt für entsprechende Tools hat momentan ein Volumen von circa 3,5 Milliarden US$ mit steigender Tendenz , wie in nachfolgender Grafik verdeutlicht.



Wegen der großen Anzahl der Anbieter bietet es sich an, die Anzahl der betrachteten Anbieter auf die jeweils wichtigsten zu beschränken. Wir wollen uns hier am Beispiel von Microsoft / Hyperion / Oracle / MicroStrategy / Business Objects / Cognos / IBM die Marktentwicklung von 2000 bis heute inklusive Trends und Marktpositionsentwicklungen vergegenwärtigen. Dabei helfen uns die folgende Grafik und die folgende Tabelle:



	2002 *(preliminary figures)*		2001		2000	
Vendor	**Market position**	**Share (%)**	**Market position**	**Share (%)**	**Market position**	**Share (%)**
Microsoft	**1**	**24.4%**	1	21.1%	3	11.5%
"Hyperion Solutions total market" (Includes all resellers' market shares)	*2*	*22.9%*	*1*	*22.8%*	*1*	*25.9%*
Hyperion Solutions (all products)	**2**	**20.0%**	2	19.5%	1	23.8%
Cognos	**3**	**12.6%**	3	11.9%	2	12.3%
Business Objects	**4**	**6.6%**	6	6.6%	6	6.1%
MicroStrategy	**5**	**5.2%**	5	6.8%	5	9.1%
SAP	**6**	**5.2%**	7	5.4%	8	2.9%
Oracle	**7**	**4.7%**	4	7.0%	4	9.9%
Cartesis/PwC	**8**	**2.7%**	9	2.4%	11	2.2%
Applix	**9**	**2.6%**	8	2.5%	7	3.0%
Comshare	**10**	**2.2%**	10	2.3%	10	2.5%

IBM (DB2 OLAP Server)	**11**	**2.2%**	11	2.1%	9	2.7%
Adaytum (now Cognos)	**12**	**2.1%**	12	1.8%	16	1.1%
MIS AG	**13**	**1.3%**	13	1.7%	12	1.6%
SAS Institute	**14**	**1.1%**	15	1.2%	13	1.6%
Brio Software	**15**	**0.9%**	14	1.3%	14	1.5%
Crystal Decisions	**16**	**0.9%**	16	1.0%	15	1.3%

Grafik und Tabelle stammen von der Internetseite **www.olapreport.com** . Der in diesem Zusammenhang erwähnte „Market share“ bezieht sich auf die Gesamtausgaben der Kunden für OLAP Produkte innerhalb eines Jahres, und will somit auch Doppeldeutigkeiten ausschließen.

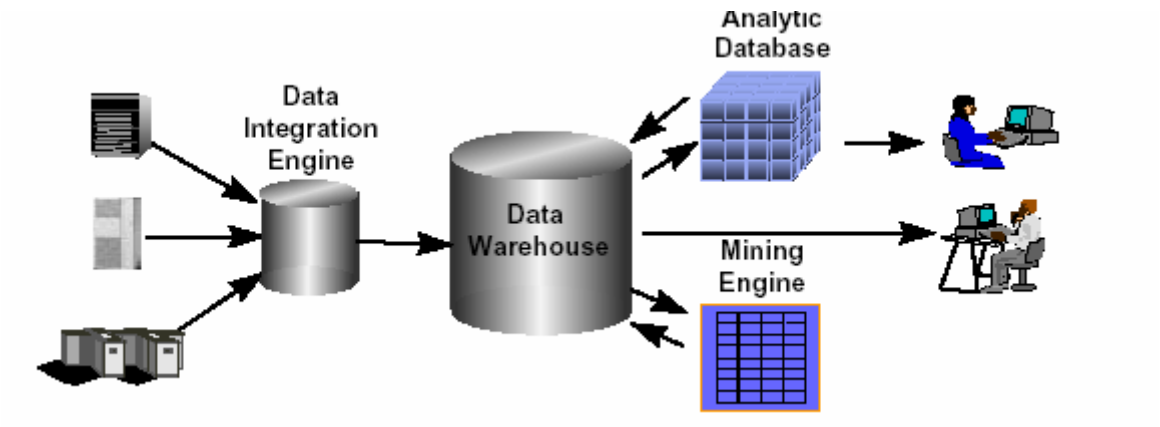
Die beiden eingefügten Grafiken erlauben uns zum einen klar zu erkennen, dass Microsoft und Hyperion klar die Marktführer sind, gefolgt , mit deutlichem Unterschied ,von Cognos. Nochmals auf der Hälfte des Marktanteilsniveaus von Cognos finden wir Unternehmen wie Oracle, MicroStrategy, SAP und Business Objects.

.

KAPITEL 3 – Wichtige Produkte

Während wir uns nun damit befasst haben wie man Produkte klassifizieren, bewerten und vergleichen kann , wollen wir in diesem Kapitel ein paar der wichtigsten Produkte und ihre Komponenten vorstellen. Orientieren wollen wir uns hier an den Systemen der in Kapitel 2 vorgestellten Marktführer wie z.B. Hyperion , Oracle, MicroStrategy, Cognos, Business Objects und Microsoft.

Wir haben ja schon gesehen das man ein OLAP System aufgrund seiner Ausrichtung bezgl. MOLAP, ROLAP oder HOLAP unterscheiden kann, aber wir werden sehen das im Produktvergleich doch sehr viele sehr ähnliche Komponenten auftauchen werden. So ist es heute zum Beispiel selbstverständlich, dass ein OLAP System eine Client Server Architektur zugrunde legt. Jedes der Programme ist Modular aufgebaut, wobei das Zentrum immer durch die Server-Komponente repräsentiert wird, als durch diejenige welche für die Verbindung zur Datenbank verantwortlich ist. Diese Komponente kann mehr oder weniger Bedeutung haben , wie wir sehen werden. Des weiteren existieren verschiedene Module die ebenfalls wiederkehren, so zum Beispiel die Entwicklungsumgebungen für die Anwendungen, die Verwaltung der Metadaten, oder die Möglichkeit spezielle Webanwendungen zu generieren. Viele der Systeme verwenden Data Marts um Anfragen die für Teile der Unternehmung Interessant sind von einander zu separieren.



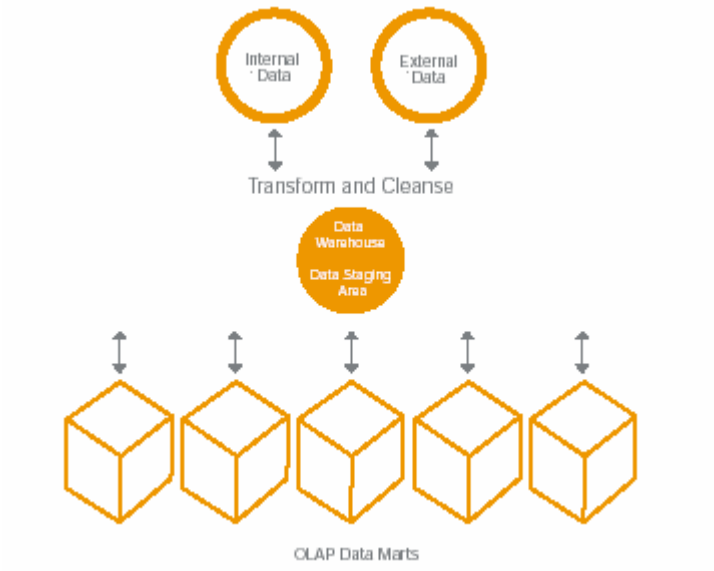
Die folgende Grafik soll diesen auf allgemeingültigen Aufbau nochmals illustrieren.

Hyperion Essbase.

Essbase war ursprünglich ein Produkt der Firma Arbor, die 1998 mit Hyperion fusionierte. Essbasse ist für mittlere bis große Unternehmen ausgelegt, deren Ziel es ist eine firmenweites Datawarehouse aufzubauen. Das Essbase Produkt basiert auf einer Client/Server Architektur, die sich multidimensionaler Data Marts bedient .

Beginnen wollen wir mit dem Ansatz den Hyperion mit dem was Hyperion mit „Best practice DW architecture“ bezeichnet. Im Grunde ist es der Vorschlag die Vorteile, die die relationale Datenhaltung bietet mit denen der multidimensionalen in geeigneter Form zu verknüpfen. Die Verknüpfung soll der Tatsache Rechnung tragen, dass die relationalen Datenbanken hervorragende Eigenschaften in Bezug auf Datenkapazität und Verwaltungsmöglichkeiten bieten, während die notwendige Geschwindigkeit für die meisten analytischen Anfragen besser durch multidimensionale Datenhaltung erreicht werden kann. Man probiert also beide Vorteile auszunutzen , indem man ein zentrales Datawarehouse , als „Staging Area“ bezeichnet, anleget , und mehrere OLAP Data Marts.

Diese Architektur wird auch mit „hub-and-spoke“ bezeichnet. Der „hub“ ist hierbei das relationale Datawarehouse , dies wird Themen übergreifend angelegt, und ist somit auch inhaltsneutral. Anders die „spokes“ , dies sind Multidimensionale Würfel, welche inhaltsorientiert sind. Sie werden aus der relationalen „Stagging Area“ abgeleitet, und regelmäßig aktualisiert. Die nachfolgende Graphik dient der Verdeutlichung des eben erklärten Sachverhalts.



Hyperion verweist darauf das anerkannte Warehousexperten wie Bill Inmon, Gartner Group, das Datawarehouse Institue , und andere , diese Form der Implementierung empfehlen.

Weiterhin erwähnt Hyperion die auf die Tatsache das bereits viele Unternehmen diese Architektur schon verwirklicht haben.

Der Kern eines jeden Essbase-Systems ist der DATA ANALYSIS SERVER. Dies stellt die eigentliche Datenbank dar. Sie ermöglicht den Mehrbenutzerbetrieb indem sie simultane Lese- und Schreibzugriffe unterstützt. Dem Datenbankadministrator wird es ermöglicht, durch ein einfaches Interface verschiedene Anwendungen zu stoppen , zu starten, oder Benutzerzugriffe zu regulieren.

In Bezug auf Business Intelligence ist das Essbasse Werkzeug „WIRED for OLAP“ von besonderer Bedeutung, dieses ermöglicht die Ausführung unterschiedlicher analytischer BI Anfragen, von verschiedenen Nutzergruppen.

Das Hauptverwaltungstool ist der APPLICATION MANAGER. Der APPLICATION MANAGER ermöglicht es von einem Server oder von einem Client aus die verschiedenen Anwendungen zu steuern, bzw. zu beeinflussen. Hierbei kann man eine Anwendung als eine Sammlung von Datenbanken, Datenbankteilen, Berechnungsskripten, Reportskripten oder Regeln für den Datenimport sehen, die alle in einem direkten kausalen Zusammenhang stehen. Die Verwaltung der Zugriffsrechte ist ebenfalls Sache des APPLICATION MANAGER. So ist er in der Lage, verschiedenen Nutzern ,bzw. Nutzergruppen unterschiedliche Rechte einzuräumen, und erlaubt durch den Einsatz von Filtern die Definition der Benutzerrechte bis hin auf Zellenebene.

Ein weiteres Werkzeug der Essbasse Produktreihe ist der DATA PREPARATION MANAGER, dieser ermöglicht die Anbindung einzelner Datenquellen. Mit ihm ist es möglich, relationale oder ASCII-Datenquellen anzubinden, und zudem erlaubt er die Definition von Transformationsregeln. Die besagten Transformationsregeln würden dann während des Datenimports ausgeführt werden.

Zum Entwickeln der Multidimensionalen Datenbankstrukturen wird dem Entwickler ein Tool namens OUTLINE EDITOR zur Verfügung gestellt. Hier wird es dem Datenbankentwickler über ein graphische Oberfläche ermöglicht, die einzelnen Dimensionen zu editieren bzw. anzulegen. Der OUTLINE EDITOR bietet die Möglichkeit, komplexere Zusammenhänge aus einer fertigen Datei zu importieren.

Wie der Name schon vermuten lässt, ist es Sache des PARTITION MANAGER dem Entwickler die Verwaltung und Erzeugung verteilter Anwendungen zu ermöglichen. Durch ihn wird es möglich einzelne Anwendungen in logische wie auch physische Partitionen

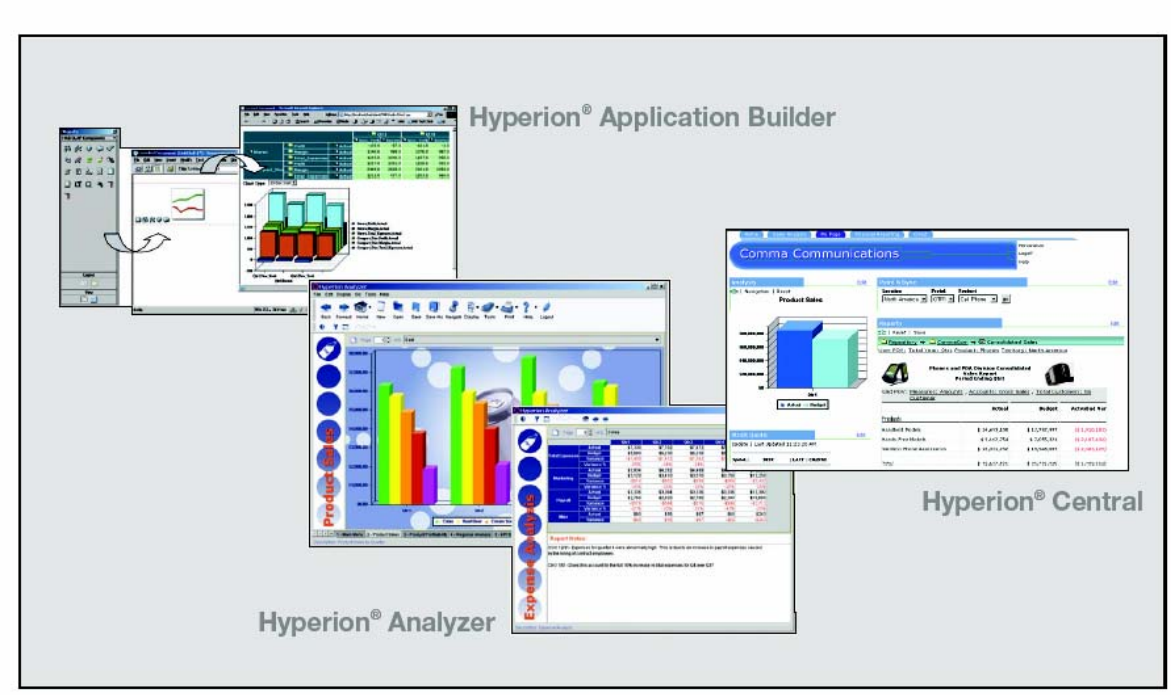
aufzuteilen, und erleichtert somit die Verteilung in einem Netzwerk. Das Abgleichen der Daten einer verteilten Anwendung erfolgt hierbei vollständig automatisch.

Ebenfalls leicht am Namen zu identifizieren ist das CURRENCY CONVERSION MODUL welches die Verwaltung verschiedener Währungen ermöglicht. Es wird dem einzelnen Anwender dadurch die Möglichkeit eingeräumt, Daten aus unterschiedlichen Unternehmensstandorten und Bereichen auf eine einheitliche Währung zu konvertieren.

Der SPREADSHEET CLIENT ermöglicht es mit Hilfe eines Add-Ins, die Tabellenkalkulationsprogramme Excel und Lotus 1-2-3 zur Datenausgabe und Datenanalyse zu nutzen.

Das bereits erwähnte Modul WIRED FOR OLAP stellt selbst eine Ansammlung unterschiedlicher Tools zur Generierung Editierung und Verwaltung von BI-orientierten Anwendungen dar. Innerhalb WIRED FOR OLAP stellt das Modul ADMINISTRATOR den Kern dar, hier werden alle Anwendungen die mit WIRED FOR OLAP geöffnet werden sollen, verwaltet. Die Verwaltung schließt hier auch die Vergabe von Zugriffsrechten, bzw. die Sperrung oder Freigabe einzelner Applikationen, mit ein. Innerhalb WIRED FOR OLAP existiert ein Modul DESIGNER, das es dem Benutzer erlaubt, auf individuelle Ansprüche einzugehen, bzw. fallspezifische Oberflächen für BI Anwendungen zu entwerfen.

Das eigentliche Businnes Intelligence Tool ist der ANALYSER . Mit ihm betrachtet man die gewünschten Daten, dabei ist es einem möglich dies über eine im DESIGNER entwickelte Oberfläche zu tun, oder über die standardmäßig zur Verfügung gestellte , Spreadsheet-ähnliche Oberfläche. Der ANALYSER ist in zwei Versionen erhältlich, einer normalen Client-Version und als Java-Applet. Somit wird der Zugriff auf die Essbasse Datenbanken mit einem Java-fähigen Browser ermöglicht.



Die eingefügte Grafik soll mit Hilfe von ein paar Screenshots die genannten Module APPLICATION BUILDER und ANALYZER zeigen.

Weiter Module von Essbasse Produkten sind PILLAR und REPORTING, sowie OBJECTS und der HYPERION INTEGRATION SERVER. PILLAR und REPORTING sind dem speziellen Bereich der Budgetierung und Planung zuzuordnen. OBJECTS stellt eine weitere Entwicklungsumgebung für individuelle BI Applikationen dar.

Der HYPERION INTEGRATION SERVER dient ebenfalls der Anbindung von Datenquellen und Erzeugung von Metadaten. Durch ihn wird die Anbindung des relationalen Datawarehouses realisiert.

Die Tatsache dass es Module gibt die teilweise das selbe , oder ähnliches tun ist auf die Fusion von Arbor Software und Hyperion zurückzuführen.

Nachdem wir nun eine allgemeinen Überblick über die Architektur von Hyperion Essbase und seiner Module gewonnen haben, wollen wir uns einige Detaillösung und Module genauer anschauen.

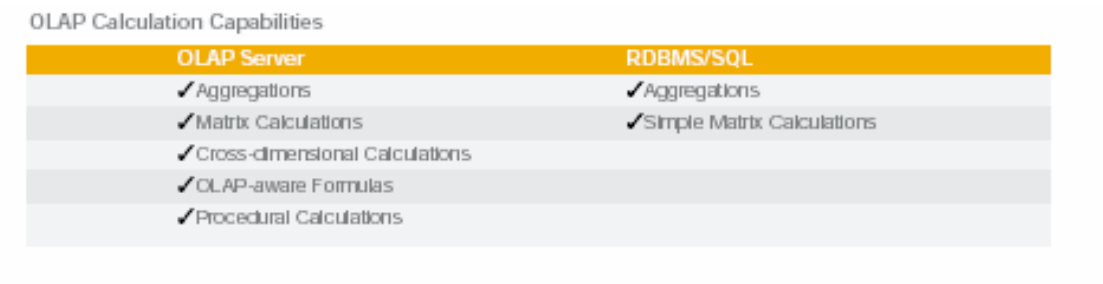
Die nächste Frage die wir uns mit Berechtigung stellen wollen ist die nach der eigentlichen Ursache für die so angepriesenen analytischen Fähigkeiten von Essbasse. Betrachten wir uns die Angaben die von einer durchschnittlichen Datenmenge für die „spokes“ mit jeweils 150GB an Daten im Schnitt ausgeht, und einer zu Verarbeiteten Menge von circa 500GB pro Anfrage, und somit auch der Nutzung mehrerer „spokes“, so zeigt sich die Notwendigkeit

geeignete Mechanismen zu finden die Anfragegeschwindigkeit zu optimieren. Hyperion behauptet das es Aufgrund verschiedener anerkannter Benchmarks erwiesen ist, das Essbase nicht nur in der Lage mehr als 7000 Benutzer gleichzeitig in angemessener Zeit bedienen zu können, auch mehr als 6800 komplexe Analytische Anfragen die Minute seien gar kein Problem.

Die Hauptursache für dieses hohe Potential liegt in der Möglichkeit drei Arten der Berechnung kombinieren zu können. Mit der Möglichkeit „precalculation“ nutzen zu können trägt man dem Aspekt der immer wiederkehrenden Anfragen Rechnung, hat aber das Problem das man auf die Menge der in der multidimensionalen Datenbank gehaltenen , vorprozessierten Daten achten muss. Mit dem „calc on the fly“ verzichtet man ganz darauf den Zwischenspeicher als Ablagefläche zu nutzen. Dies ist sicherlich aus gründen der Speicherhaltung sehr positiv zu bewerten, wird aber mit einer verschlechterten Antwortzeit erkauft. Ein Kompromiss zwischen beiden Berechnungsarten stellt die Option „calc on the fly and store“ da. Dies bedeutet das eine Berechnung die angefragt wurde, ist sie noch nicht vorher von Jemand anders ausgeführt worden, wird direkt ausgeführt. Das Ergebnis wird dann aber nicht verworfen, sondern wird für denn Fall das Jemand das selbe Fragen sollte, gespeichert

Der bereits angesprochene INTEGRATION SERVER ist das maßgebliche Instrument von ESSBASE um die Verbindung zur relationalen Datenbank herzustellen. Dabei bedient sich das Modul einem von ihm erzeugten, zentral angelegtem Metadatenkatalog. Hier sind Dimension , Hierarchien ,Templates oder Berechnungsvorschriften hinterlegt. Hierbei ist es jeder Anwendung innerhalb ESSBASE möglich auf diesen Katalog zuzugreifen. Die im Metadatenrepositorium gehaltenen Daten werden regelmäßig mit der Datenbank synchronisiert. Der INTEGRATION SERVER ermöglicht es damit dem Benutzer bei Anfragen nicht nur auf die Daten zuzugreifen die von ESSBASE direkt im multidimensionalen Bereich gehalten werden, sondern auch die „drilldown“ Möglichkeit bis auf Detailebene wird unterstützt. Dies bedeutet dann auch, dass man Daten direkt aus der Datenbank in die Anfragen mit einbeziehen muss. Unter zu Hilfenahme der Metadatenbank wird es dem INTEGRATION SERVER ermöglicht auf die relationale Datenbank abgestimmtes SQL zu generieren und mithilfe einer Anfrage an die Datenbank die gewünschten Detailinformation zu liefern. Auf diese Weise entsteht ein gewisser Verbund von MOLAP und ROLAP, dieser äußerst sich in der Weise, dass häufig frequentierte , allgemeine Daten, im multidimensionalen Cube gehalten werden, und Detailinformationen eben direkt aus der Datenbank abgeleitet werden können.

Der nächste wichtige Mechanismus innerhalb ESSBASE auf den wir eingehen wollen, ist die OLAP CALCULATION ENGINE. Die Notwendigkeit eine eigene OLAP CALCULATION ENGINE überhaupt zu verwenden liegt in der Inflexibilität der relationalen Datenbanken und des SQL an sich. Innerhalb eines OLAP Produktes müssen zumindest 5 Typen von Berechnungen unterstützt werden. Zum einen die Aggregationen, sie dienen der einfachen Summation einzelner Einträge hinzu einer höheren Ebene, zum Beispiel tägliche Einträge hinzu wöchentlichen , etc. Die Aggregat Funktionen können auf der relationalen Datenbank ausgeführt werden, und rechtfertigen den Einsatz einer eigenen Berechnungskomponente nicht. Ganz anders mit den 4 verbleibenden Berechnungsarten. Die Matrizen-, die Kreuzdimensionalen-, und die Prozedurahlenberechnungen , sowie die „OLAP-aware formulas“ benötigen eine eigene Berechnungskomponente innerhalb des OLAP Programms. Während man sich bei der Matrizenrechnung noch streiten kann , denn es ist möglich sehr einfache Matrizenrechnungen mit Hilfe von SQL durchzuführen, so wäre bei den Kreuzdimensionalen ein Verständnis von Hierarchien notwendig, und somit die Sachlage klar. Matrizenberechnungen begegnen uns zum Beispiel bei Vergleichen von Periode zu Periode. Sie sind denn Berechnung in Spreadsheet Programmen wie Excel sehr verwandt, durch sie wird es möglich in der vorhanden Menge eine Berechnung aufgrund von Beziehungen verschiedener Zellen untereinander zu definieren. Kreuzdimensionalenberechnungen gehen noch einen Schritt weiter, sie definieren über verschiedene Dimensionen und Aggregationsstufen hinweg, nehmen wir den Excelvergleich nochmals auf, so kann man sich dies als Berechnung zwischen N verschieden Spreadsheets vorstellen. Die „OLAP-aware formulas“ stellen eine Bibliothek von fertige Funktionen dar, die die Bereiche welche für die Erstellung umfassender Analysen notwendig sind, abzudecken versucht. Die folgende Abbildung soll diesen Sachverhalt erneut Verdeutlichen.



Über diese Funktionalität hinaus ermöglicht ESSBASE die Integration individuell gestalteter Funktionen , die in Form von Java Programmen integriert werden können. Dieses werden dann als CDF (Custom Defined Function) oder CDM (Custom Defined Macro) bezeichnet.

sparse and dense dimensions

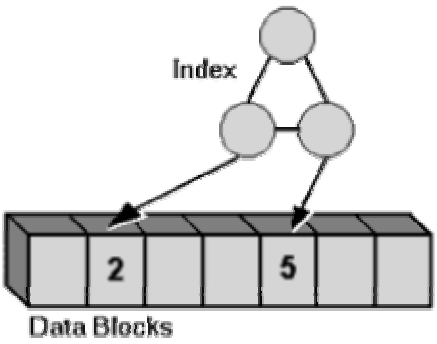
Ein Sachverhalt der uns überall begegnet wo wir mit multidimensionaler Datenhaltung zu tun haben, ist die Frage nach „sparse and dense dimensions“.

„sparse dimensions“ sind hierbei als Dimensionen anzusehen ,wo zu erwarten ist, dass nur wenige Zellen die in Bezug auf die anderen Dimensionen angelegt wurden, relevant, und damit überhaupt nötig sind. Das Gegenteil sind die „dense dimensions“ hier werden erwartungsgemäß nahezu alle Zellen mit Werten besetzt.

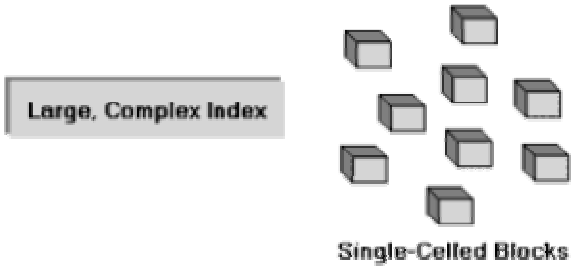
Dementsprechend ist das Speicherkonzept darauf ausgelegt bei den „sparse dimensions“ wirklich erst beim befüllen einer Zelle den entsprechenden Speicherplatz zu allokieren, und bei den „dense“ vorsichtshalber schon mal alle.

Die multidimensionale Speicherung hat hauptsächlich zwei Stützen , zum einen das multidimensionale Datengebilde selber, zum anderen den Index.

Um auf dem multidimensionalen Gebilde zu navigieren wird ein Index verwendet, der den schnellen Zugang zu allen mit Daten gefüllten Zellen ermöglichen soll. Wie nachfolgend illustriert.

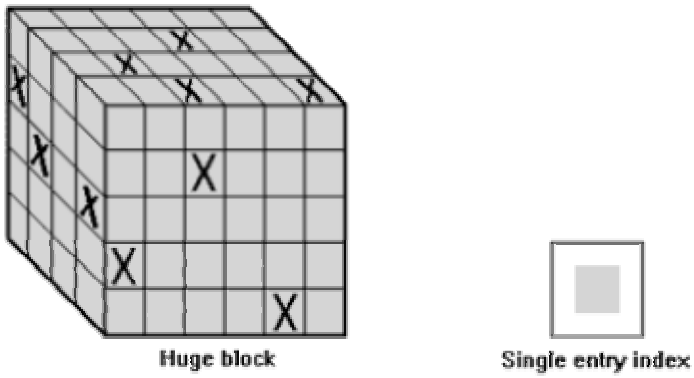


Im Mittelpunkt steht hier das multidimensionale Gebilde. Im generellen sollte für jede mögliche Kombination der Dimensionen eine Zelle existieren. Aber um Speicher zu sparen, und möglichst schnellen Zugang durch den Index zu erlauben ist die korrekte und ausgewogene Definition der entsprechenden Dimensionen essentiell. In den nächsten drei Abbildungen soll uns dies vorgeführt werden. Zum einen sehen wir den extremen Fall das alle Dimensionen als „sparse“ definiert worden sind.

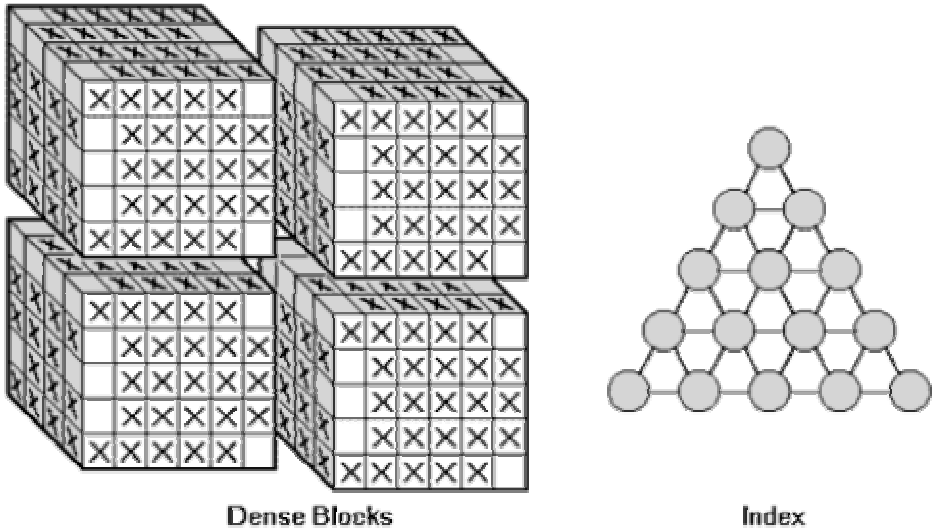


Wie man leicht erkennen kann gibt es hier keinen multidimensionales Gebilde mehr , vielmehr sind die Daten auf einzelne Zellen verteilt. Dies ist vielleicht aus Sicht des Speichers günstig, man büsst aber durch den sehr großen Index sehr viel Performance bezüglich der Anfragegeschwindigkeit ein.

Ein anderes Extrem stellt die Definition aller Dimensionen als „dense“ dar.



Es ist offensichtlich, dass dieses Gebilde, in Relation zu seinen mit Information gefüllten Zellen, unnötig viel Speicherplatz benötigt. Zum anderen ist der Index von einer Navigationshilfe zu einem singulären Eintrag entartet. Im folgenden soll der Idealfall dargestellt sein , die geeignete Kombination von „sparse“ und „dense“ Dimensionen.



Die Blöcke sind nicht vollbesetzt, und wo sie nicht besetzt sind, ist auch kein unnötiger Speicherplatz allokiert. Der Index stellt eine ausgeglichene Baumstruktur dar, die optimale Navigationsgeschwindigkeit erlaubt.

MicroStrategy 7i

MicroStrategy hat sich das Ziel gesetzt, sehr komplexe analytische Anfragen auf sehr großen relationalen Datenbanken zu unterstützen. Die Gesamtlösung selber ist also mehr auf die Großindustrie zugeschnitten.

In diesem Produktvergleich wollen wir uns mit dem neusten Produkt der Firma MicroStrategybefassen, und zwar MicroStrategy 7i. Mit dieser Betrachtung haben wir auch einen Vertreter des ROLAP-Ansatzes vor uns. MicroStrategy 7i ist keine Komplettlösung , wie zum Beispiel ORACLE 9i das sein kann, da es auf relationalen Datenbanken anderer Hersteller aufsetzt.

Die vorliegende Betrachtung ist in drei Teile eingeteilt. Zum einen möchte ich mir die zugrunde liegende Architektur des Systems anschauen und die wichtigsten Komponenten mit Aufgaben nennen, zum anderen dann bezüglich zweier Aspekte ein wenig in die Tiefe gehen. Im zweiten Teil der Betrachtung wollen wir uns die Metadatenhaltung anschauen, und im dritten dann ein Modul im Detail betrachten. Am besten eignet sich hier, wegen seiner in jedem Zusammenhang zentralen Rolle, der MicroStrategay Intelligence Server.

System Architektur.

Wir haben bereits einführend zu diesem Kapitel die groben Merkmale einer jeden Produktarchitektur für den OLAP Markt benannt. Auch in diesem Produkt werden wir alle die bekannten Punkte wie Entwicklungswerkzeuge, etc. antreffen.

Ausgehen möchte ich von einer Skizze die uns die Zusammenhänge der einzelnen Module Innerhalb des Systems verdeutlich soll:

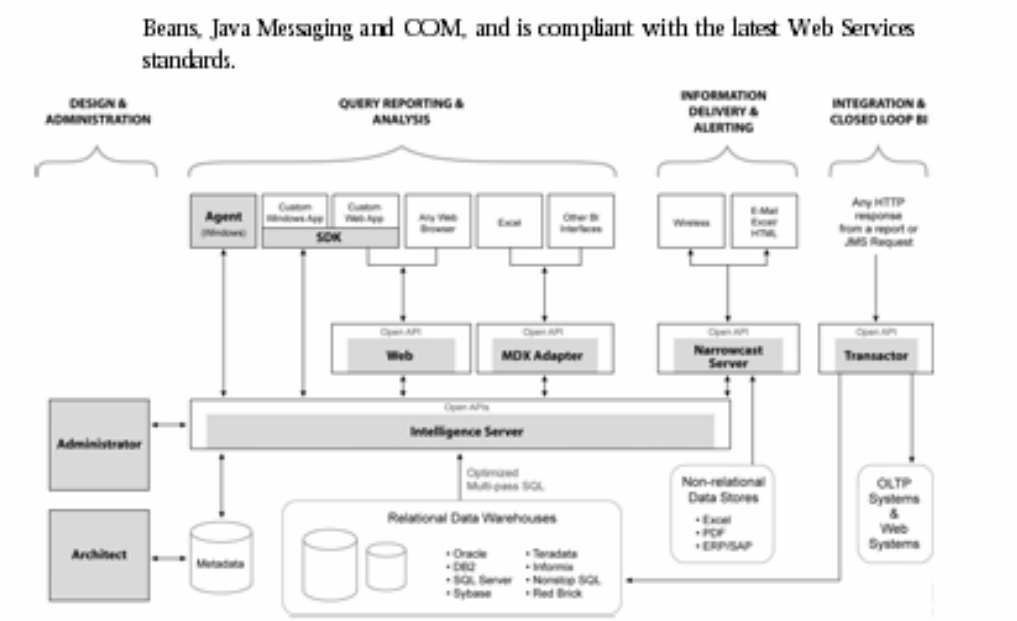


Figure 2-1 MicroStrategy 7i's product architecture provides the full range of end-user functionality, power for developers and control for administrators that you expect from an enterprise-class BI platform

Wie unschwer zu erkennen ist spielt der Intelligence Server die Hauptrolle an sich. Er ist mit seinen Aufgaben und Fähigkeiten das logische und architektonische Kernstück. So umfassen seine Aufgaben zum Beispiel die zentrale Haltung und Verwaltung der Metadaten, die Erzeugung und das Update der Intelligence Cubes, wie MicroStrategy ihre Variante des Multidimensionalen Datenwürfels getauft hat, oder falls die Daten nicht bereits in den Cubes vorhanden sind, eine auf das Datenbankprodukt optimierte SQL Anfrage zu erstellen und die Ergebnisse analytisch aufzubereiten. Auch sind administrative Kontrollfunktionalitäten und Sicherheitsfragen teil seiner Aufgabenliste. Wie gesagt wollen wir uns dies Modul noch im Detail anschauen.

Eine Erweiterung des gerade erwähnten Intelligence Servers sind die neuen OLAP Services, hier wird es dem Benutzer via Interaktiver Schnittstelle ermöglicht die multidimensionalen Cubes zu benutzen oder gleichermaßen zu verändern. MicroStrategy verweist darauf, dass zum einen die Möglichkeit einer sehr effizienten Analyse innerhalb des multidimensionalen Datenanteils gibt, ohne dem Nutzer die Möglichkeit zu nehmen weiter Daten aus der Datenbank direkt in seine Betrachtung zu integrieren.

Der MicroStrategy Desktop ist das Tool zum erstellen verschiedenartigster Reports, oder Scorecards auf dem Desktop. So läuft es zum Beispiel auch unter Windows. Das Tool beschränkt sich darauf die Anfragen an den Intelligence Server weiterzureichen, der daraufhin, je nach Caching-Lage, die Intelligence Cubes nutzt , oder eine Datenbankanfrage generiert.

Mit dem Modul MicroStrategy Web wird es ermöglicht mit einem Web browser die volle Funktionalität des Systems zu nutzen. Dabei ist das Web Portal bewusst sehr Benutzerfreundlich ausgelegt um einer sehr breiten Schicht von Anwendern einen einfachen Einstieg zu gewährleisten.

Der Narrowcast Server wird neben dem Intelligence Server als zweiter „Report-Server“ innerhalb der 7i Architektur bezeichnet. Die Aufgaben des Narrowcast Servers beziehen sich darauf innerhalb gewisser Intervalle, oder Aufgrund gewisser Ereignisse, Mitarbeiter, Geschäftspartner etc. , selbsttätig zu informieren. Wobei die Information dabei auch via email, Handy oder Pager erfolgen kann. Durch die Möglichkeit für jeden einzelnen Nutzer nicht nur die Art und Weise zu wählen auf die er informiert werden will, sondern diese Information auch zu selektieren , wird die Informationsflut zum einen gesteuert und zum anderen
eingeschränkt.

Damit es Nutzern von verschieden Business Intelligence Tools möglich ist problemlos die Funktionalität von MicroStrategey7i zu nutzen, wurde der MicroStrategy MDX Adapter entworfen. Er erlaubt die Nutzung der 7i Plattform mit Hilfe der multidimensionalen Anfragesprache MDX, oder via der API OLE DB for OLAP. Dabei werden die Anfragen, wie zum Beispiel von Excel, dann von MDX in SQL übersetzt.

Wie wir später noch genauer betrachten werden , spielt die Metadatenschicht eine gewichtige Rolle im gesamten System. Eine der wichtigsten Aufgaben ist die schichtenweise Abstraktion der in der Datenbank vorhandenen Daten zu geschäftsrelevanten Daten. Das Modul welches die Zuordnung des in der Datenbank physisch vorhandenen Datenmodels, hinzu dem Objekt orientierten Model der Geschäftsdaten erlaubt, ist der MicroStrategy Architect. Es wurde Wert darauf gelegt diese Oberfläche möglichst benutzerfreundlich zu gestalten.

Das hauptsächliche Tool zur Administration der Nutzer , zum Erkennen von Nutzungstrends und zum Management der Infrastruktur, ist der MicroStartegy Transactor. Er ist selber wiederum in 3 Komponenten eingeteilt , den Command Manager, Enterprise Manager und Object Manager. Die Verwaltung der Nutzer inklusive der Sicherheitsaspekte fällt dabei dem Command Manager zu, der Enterprise Manager ermöglicht die Analyse von Systemlast , Benutzerstatistiken, etc. Der Object Manager hat hingegen die Aufgabe den Nutzern verschiedene Umgebungen individuell aus „Report“ Objekten zusammenzustellen.

Mit dem MicroStrategy Transactor wird es dem Anwendungsentwickler erlaubt innerhalb der Report Objekte Buttons oder Dialoge einzufügen, als dem Nutzer des Reports eine Interaktion zu erlauben.

Das Modul MicroStrategy SDK ist eine Entwicklungsumgebung die es Programmierern mit Hife von Java oder COM, ermöglichen soll, MicroStrategy individuell zu gestalten bzw. an bestehende Systeme anzupassen. Hierbei kann man die zusätzlichen Funktionen als eine Plattformerweiterung sehen, die als Windows oder Web Anwendungen realisiert worden ist.

Um diese Möglichkeit zur individuellen Entwicklung von Applikationen zu beschleunigen existiert das MicroStrategy BI Developer Kit. Dieses Kit besteht aus verschiedenen Modulen wie Kundenanalyse, Finanzanalyse, etc. Innerhalb dieser Module werden dann fertige Hierarchien ,Maßzahlen und Attribute passend zum Themenkomplex angeboten.

Dieser grobe Überblick sollte nur einen Eindruck des Zusammenhangs der Systemkomponenten vermitteln. Tangiert haben wir nun auch schon die im folgenden genauer betrachteten zwei Elemente des Systems.

Metadaten

Metadaten sind als „Daten über Daten“ aufzufassen und ermöglichen den Übergang vom physischen Model der Daten wie sie innerhalb der Datenbank vorhanden sind , hin zum Objekt orientierten Model der Daten wie es für Business Intelligence Anwendungen nötig ist.

Metadaten werden auf allen logischen Abstraktionsebnen genutzt. Umso höher die Abstraktionsebene , umso weiter ist das logische Model von der in der Datenbank repräsentierten physischen Datenbasis abstrahiert. So wird zum Beispiel auf der untersten Ebene erstmal eine Definition für das multidimensionale Model getroffen, während auf der höchsten Abstraktionstuffe es um die Arten der Informationspräsentation geht.

Im MicroStrategy System werden Metadaten ebenfalls in einer relationalen Datenbank gespeichert. Dies hat den Vorteil das man auch sehr große Metadatenbestände in sehr effizienter weise verwalten kann.

Neben dieser Eigenschaft lenkt MicroStrategy den Augenmerk auf drei weitere Aspekte der Metadatenhaltung die in 7i verwirklicht sind. So werden die Metadaten verschlüsselt gespeichert um dem Sicherheitsaspekt Rechnung zu tragen.

Erst durch die absolute Objektorientierung der Metadaten werden auch Eigenschaften wie Vererbung und Einbettung der Datenobjekte möglich. Dies wiederum führt dazu, dass sich die analytischen Fähigkeiten und die Zuverlässigkeit des Systems steigern lassen.

Als letzter Fokus ín Bezug auf die Metadaten und ihre Haltung, wird die Eigenschaft betont entsprechenden Abstraktionsebenen zur Verfügung zu stellen. Hiermit wird der objektorientierte Ansatz Stufenweise implementiert , und Änderungen , zum Beispiel in der Datenbasis, haben keinen direkten, oder keinen zu großen, Einfluss auf das logische Model. Die Abstraktionsschichten verhelfen dem Objektorientierten Ansatz also zur notwendigen Stabilität.

Innerhalb 7i existieren davon drei. Die Drei Schichten verwenden unterschiedliche Module der Architektur von 7i, wie sie bereits im Vorfeld Erwähnung fanden, zur Modellierung der Metadaten. Die Schichten bauen direkt aufeinander auf, und haben definierte Elemente.
Die Elemente einer unter Ebene dienen dabei immer als Teile zur Generierung eines Elementes einer höheren Ebene. Die drei Schichten werden in der eingefügten Grafik mit ihren Hauptelementen , und in Abstraktionsreinfolge genannt. Leicht ersichtlich ist auch die Zentrale Rolle des MicroStrategy Architect , der der Erstellung der untersten Schicht , und ihrer Objekte dient.

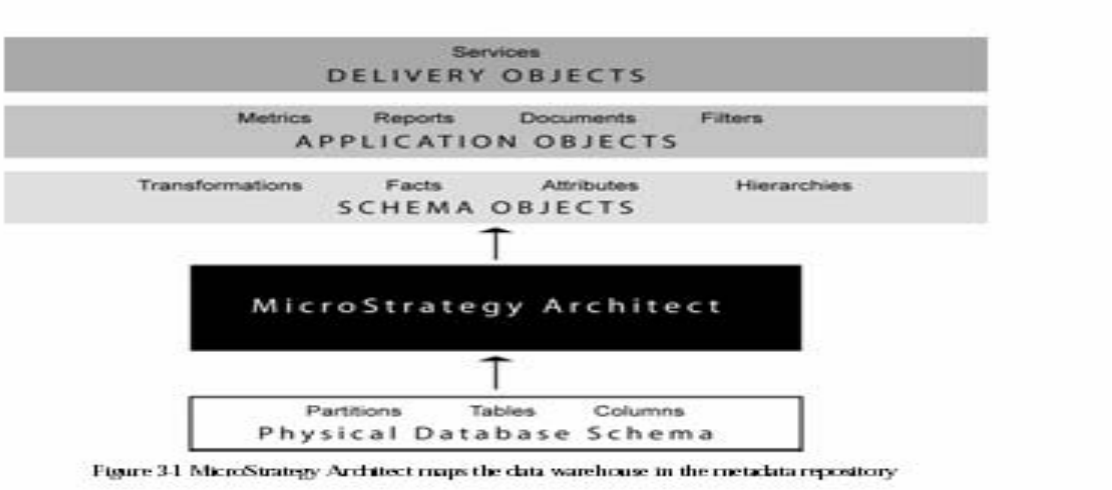


Figure 3.1 MicroStrategy Architect maps the data warehouse in the metadata repository

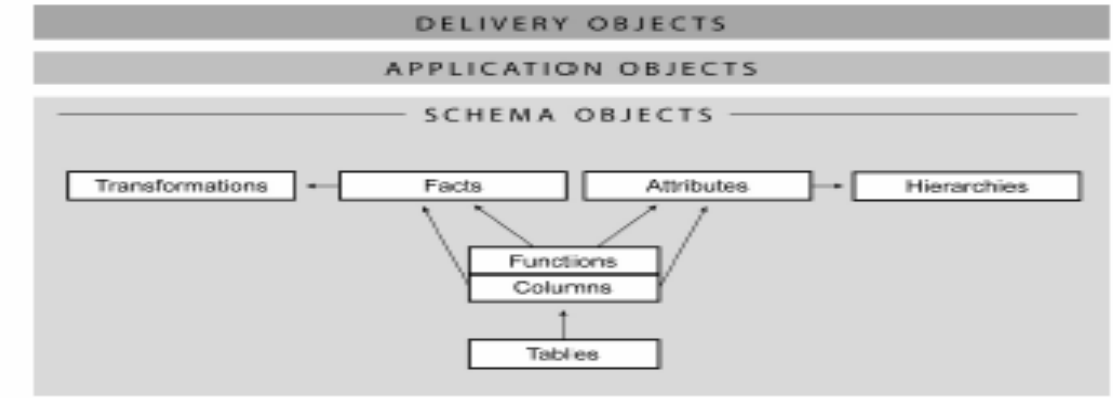
Die Schicht der Metadaten welche dem physischen Modell am nächsten kommt wird mit „Schema Objects“ bezeichnet. Die Aufgabe dieser Schicht ist Objekte mit direkter Bedeutung für die BI Anfragen zur Verfügung zu stellen die aus Daten der physischen Datenbasis abgeleitet wurden. Somit stellen diese Objekte eine direkte Abstraktion der Ur-Daten dar.

Das Architect-Tool erstellt die Schema Objekte automatisch in dem er die im Datawarehouse hinterlegten Daten und das logische Model interpretiert. Beide Modelle werden in zwei unterschiedlichen Graphen angezeigt. Der „physical view“ dient hierbei der Wiedergabe der

in der Datenbank definierten Tabellen und Spalten, der „logical view“ hingegen zeigt jede Tabelle mit ihrem multidimensionalen Zusammenhang, so wie sie im Metadatenmodell gespeichert worden ist.

Um eine möglichst hohe Flexibilität bei der Schemaumgebung zu Garantieren, und somit die Möglichkeit zu schaffen sehr komplexe Geschäftsabläufe abzubilden, ist auch eine Definition eines Schemaobjektes aufgrund eines anderen , mehrerer anderer, oder einer Kombination aus Schemaobjekten und Ableitungen aus der physischen Datenbasis möglich.
Diese Funktionen aus dem Standard SQL gegriffen worden sein , oder plattformspezifisch bezüglich des Datawarehouses.

Durch die Möglichkeit das sich Schemaobjekte auch auf mehrere Elemente der Datenbasis beziehen können, in dem diese Tabellen oder Spalten logisch miteinander verknüpft werden können, wird die Notwendigkeit für Änderungen in der Datenbasis so gering wie möglich gehalten. Die folgende Grafik zeigt die besprochene Metadatenschicht unter Erwähnung ihrer wichtigsten Elementtypen.



Wir wollen nun Stichpunktartig die wichtigsten Elemente nennen und erklären.

- Ein „Fact“ ist eine im Metadaten Repositorium gespeicherte Repräsentation einer Tabelle aus dem Datawarehouse, oder ihre Interpretation aufgrund einer logischen Funktion, die entweder Basisdaten enthält oder Daten die aufgrund einer Aggregation zustande gekommen sind.

.

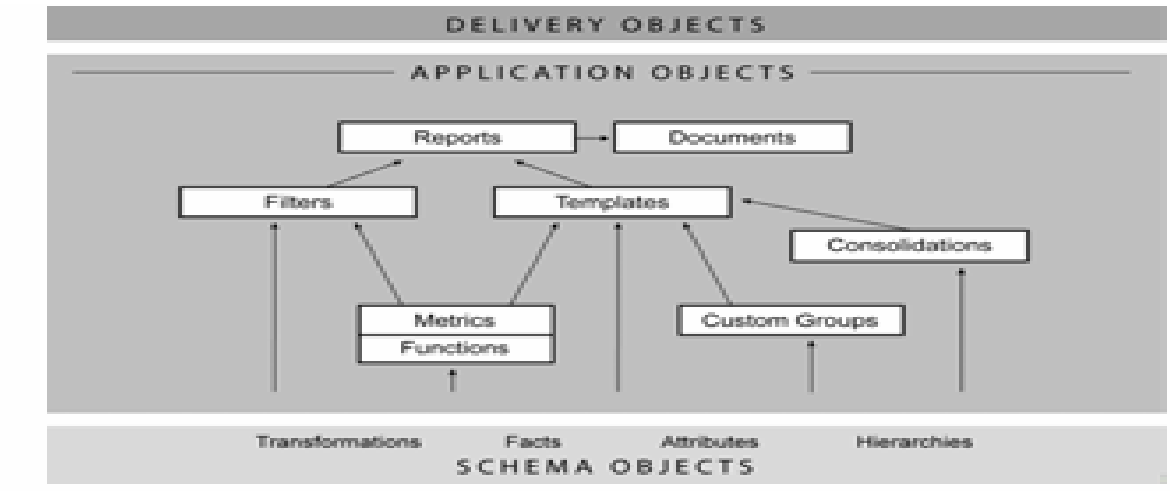
- Ein „Attribute“ ist gekennzeichnet durch seine Möglichen Werte und die Art der Werte. Über ein Attribut wird es möglich zusätzlich Information zu den erwähnten „Facts“ zu speichern. Mit Hilfe dieser Information ist zum Beispiel ein Filtern der Daten aus dem Datawarehouse möglich.
- „Hierarchies“ sind als logische Verbände von Attributen anzusehen , bei denen eine fest

definierte Beziehungsstruktur besteht. Innerhalb 7i unterscheidet der Hersteller zwischen zwei Arten von möglichen „Hierarchies“, der „system hierarchy“ und der „user hierarchy“. Die „system hierarchy“ ist als Abbild der in den Metadaten abgelegten logischen Struktur zu sehen. 7i erlaubt es auch dem Nutzer „hierarchies“ selbst zu erschaffen, und damit die Möglichkeit eigene Beziehungen zwischen „Atttributes“ zu definieren. In beiden Fällen kann man die „hierarchy“ stellvertretend für ein normales „Attribute“ in ein Report Objekt einbauen. Die Auswertung , und eigentliche Wertzuweisung, ist dann zur Laufzeit.

- „Transformations“ sind Objekte die vor allem die zeitlich orientierten Analysen begünstigen. Es wird zwischen einer „one-to-one“ und einer „many-to-many Transformation“ unterschieden. innerhalb der „one-to-one“ betrachtet man identische große Zeiträume, beim „many-to-amy“ Vergleich spielt hingegen keine Rolle.

Nun kommen wir eine Schicht höher im Abstraktionsmodel der Metadaten. Die „Application Objects“ sind per Definition die nachfolgenden Objekte , welche auf den „Schema Objects“ aufbauen, bzw. daraus aufgebaut sind. Dies wiederum wird mit Hilfe des Moduls MicroStrategy Desktop , oder einem Web Interface verwirklicht. Die zentrale Aufgabe aller in dieser Schicht vorhanden Objekte ist der Erstellung von Report-Objekten zu dienen. Durch ein sehr hohes maß an Flexibilität wird es deutlich erleichtert dieser Aufgabe gerecht zu werden. Die erwähnte Flexibilität drückt sich zum Beispiel so aus, das man ohne weiters ein Objekt in eine anderes einbetten kann.

Die folgende Grafik zeigt die „Application Objects“ Schicht des Metadatenmodells mit seinen wichtigsten Elementen und deren intendiertem Zusammenspiel :



Wie in der vorangegangen Schicht wollen wir auch hier die hier dargestellten Elemente

Stichpunktart besprechen:

- „Metrics“ stellen eine mögliche Berechnung auf dem Datenbestand dar. Diese kann durch eine Formel, eine Aggregatfunktion oder einen Arithmetischen Ausdruck repräsentiert werden. „Metrics“ sind die Darstellung wichtiger Geschäftszahlen , oder zentraler Performanz Indikatoren.

- „Filters“ sind , wie vom Namen bereits impliziert, die Darstellung der Anforderungen, welche die Daten erfüllen müssen, um in ein Report oder Metric Objekt integriert werden zu dürfen. Diese Selektionseigenschaft grenzt die Auswahl der Daten im Datawarehouse ein, und hilft die für einen bestimmten Nutzer , oder Report, wichtigen Daten zu spezifizieren.

- „Custom Groups“ ermöglichen die Erstellung von Datengruppierungen wie so vorher in der Datenbank nicht vorhanden waren. Dies Gruppierung kann mit Hilfe von „Filters“ und „Metrics“ realisiert werden.

-„Consolidations“ sind den „Custom Groups“ sehr ähnlich. Es wird durch die „Consolidations“ ermöglicht anhand von arithmetischer Verknüpfung zweier Gruppen eine neue zu definieren.

-„Prompts“ ermöglichen die Nutzerinteraktion in Reports zur Laufzeit. Es werden hier 4 verschiedene Kategorien unterschieden, Die „Filter prompts“ dienen der Auswahl der sich für die Ergebnismenge qualifizierten Daten. „Value prompts“ dienen der Eingabe von alphanumerischen Werten. „Object promts“ dienen als Platzhalter für Report Elemente innerhalb eines Report-Objektes. Die „Level prompts“ kontrollieren die Aggregationsstuffe der „Metrics“ innerhalb eines Reports.

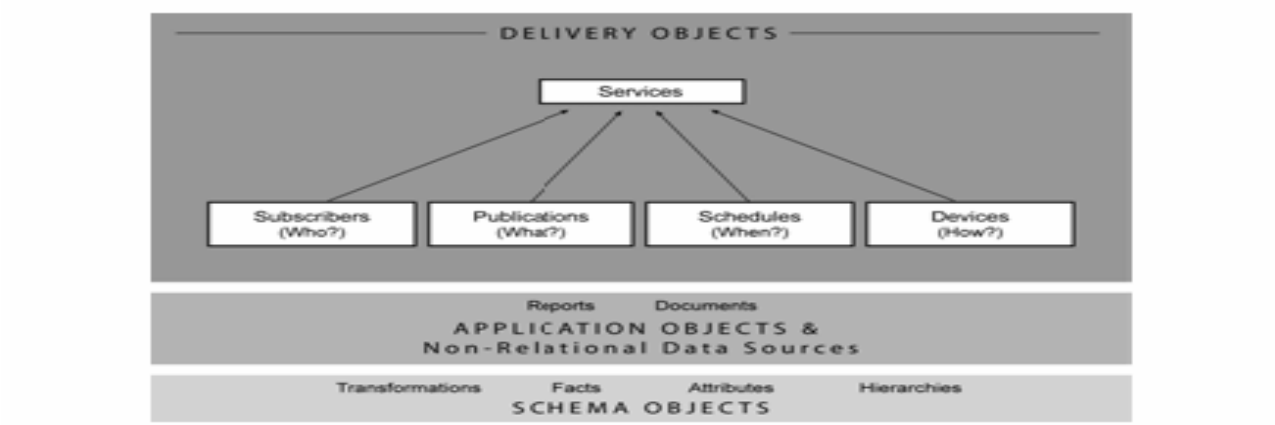
-„Templates“ geben die Formate und das Layout der Daten für ein Report-Objekt vor.

-Die „Reports“ sind die eigentlichen Ergebnisse der Business Intelligence Anwendung.
Durch sie wird letztendlich der Einblick in die angeforderten Analysen gewährt.
Dabei kann man einen Report als eine Anfrage an die Datenbank verstehen.
Das zugrunde gelegte Template sorgt dabei für die Auswahl und Formatierung der Daten.

- „Documents“ vereinigen mehrere Reports und weitere Objekte, in einem HTML Dokument.

Die letzte , und oberste, Schicht der Abstraktion stellen die „Delivery Objects“ dar.

Ihre Aufgabe ist es die individuell zugeschnittenen Informationen in geeigneter Form an den Mann zu bringen. Dabei können dies Daten vom Intelligence Server sein, aber auch von anderen Quellen. Das Modul, das zur Generierung der „Delivery Objects“ dient ist der Narrowcast Server. Er benutzt wiederum Elemente aus der eins niedrigeren Abstraktionsebene als Grundelemente. Auch hier wieder ein graphischer Überblick über Elemente dieser Schicht und deren Zusammenhang;



Wie man leicht an der Graphik erkennen kann ist die zentrale Aufgabe des Narrowcast Server also die Erstellung und Lieferung der „Services“. Ein „Service“ ist dabei Produkt aus vier anderen „delivery objects“. Die „Publications“ beantworten die Frage nach dem Was. Sie enthalten die eigentliche Nachricht. Mit den „Subscribers“ ist die Liste von interessierten Empfängern gemeint. Das „device“ definiert die Art und Weise der Informationsdarstellung und Übermittlung, „Schedules“ ermöglichen einen frequenzbasierten oder reaktionsbasierten Generierungsprozess.

MicroStrategy Intelligence Server

Vom Narrowcast Server wollen wir nun wieder zum zweiten Report Server innerhalb der 7i Architektur zurück kehren , dem Intelligence Server. Seine zentrale Bedeutung haben wir uns am allgemeinen Model der 7i Architektur schon verdeutlicht. Hier soll nun im Detail auf die Aufgaben, Fähigkeiten und die Architektur dieses Herzstücks von 7i eingegangen werden.

Wollte man die Aufgabe in wenigen Worten zusammenfassen, so kann man sagen das die Umwandlungen von analytischen Anfragen in datenbankspezifisch optimiertes SQL, und die Ausführung der eventuelle zusätzlich nötigen analytischen Berechnungen dem Kern der Summe der Aufgaben entspricht.

Wir wollen uns nun die wichtigsten Komponenten des Intelligence Servers unter zu Hilfenahme eines Anfrageablaufs anschauen. Somit haben wir quasi zwei Fliegen mit einer Klappe geschlagen , werden ja nicht nur die Komponenten am Beispiel erklärt ,sondern auch der im Intelligence Server implementierte Ablauf verdeutlicht.

Hier eine Abbildung eines solchen Anfrageablaufs;

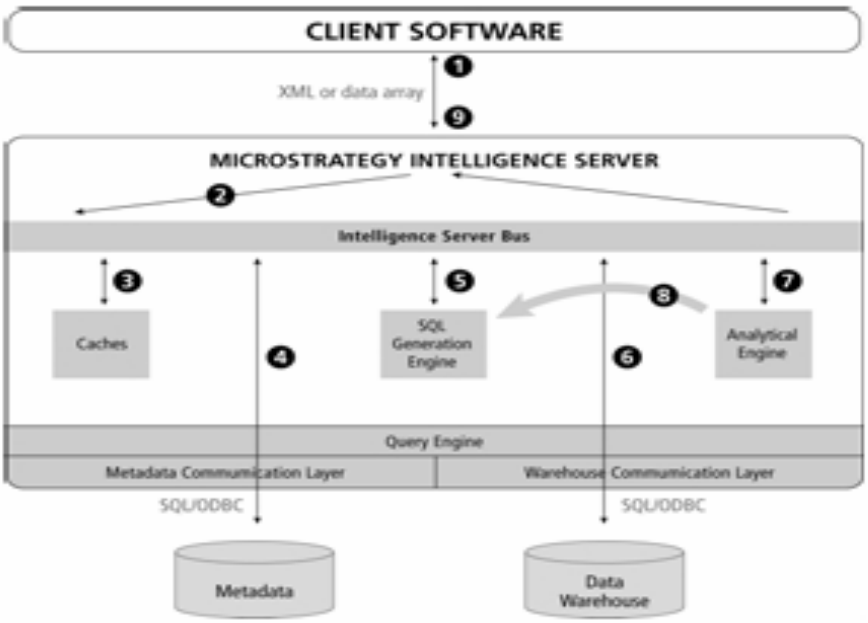


Figure 5-1 MicroStrategy Intelligence Server query flow

Die wichtigsten zu nennenden Teile des Intelligence Servers sind hierbei:

1. Der Intelligence Server bus
2. Der Intelligence Cubes server
3. Kommunikationsschichten für die metadaten und das Datawarehouse
4. Die SQL Generation Engine
5. Die Query Engine
6. Die Analytical Engine

Und hier der an der Graphik nachzuvollziehende Beispielhafte Ablauf:

1. Der Intelligence Sever erhält von irgendeinem Interface eine report Anfrage.
2. Die Anfrage wird dabei durch den Intelligence Server bus gereicht, welcher alle Notwendigen Schritte zur Bearbeitung veranlasst.
3. Der Intelligence Server überprüft ob die benötigten Ergebnisse eventuell schon im Cache vorhanden sind oder nicht. Sollte ein anderer Nutzer diese Anfrage schon mal gestellt haben, und das Caching ermöglicht die sofortige Rückgabe der Ergebnisse, bedeutet dies eine enorme Zeitersparniss.
4. Sollte kein passender Cacheeintrag vorhanden sein, so stellt der Intelligence Server

bus die benötigten Report Definitionen und „application objects“ der Metadaten zur Verfügung.

5. Der bus reicht die benötigten Informationen an die „SQL Generation Engine“ weiter, welche dann auf das Datenbankprodukt optimiertes SQL erzeugt, und dies dann wieder an den bus zurückgibt.
6. Der bus sendet die fertigen SQL Anfragen an die „Query Engine“ die diese dann auf das Datawarehouse anwendet, und die entstandenen Ergebnisse dann auch wieder zurück an den bus sendet.
7. Der Intelligence server bus spricht die „Analytical Engine“ an , falls zusätzliche analytische Daten benötigt werden. Die Ergebnisse fließen wieder an den bus zurück.
8. Je nach Komplexitätsgrad der gestellten Anfrage wird die Anfrage nochmals zur „SQL Generation Engine“ zurück gegeben , um die Datenabnk nochmals anzusprechen. Dies ist quasi eine mögliche Schleife im Ablauf der die Punkt 4 bis 8 umfasst, und die Anzahl der Durchläufe richtet sich nach dem Komplexitätsgrad der Anfrage.
9. Der Intelligence Server gibt die Informationen in geeignetem Format zurück und das Tool welche die Anfrage ursprünglich abgesetzt hatte.

Da wir nun eine Idee von den im Intelligence Server vorhandenen Elemente und deren Bedeutung gewonnen haben, wollen wir nun noch paar wichtige allgemeine Eigenschaften erwähnen.

Mit der „Automatic Resource Allocation“ ist die Eigenschaft des Intelligence Servers gemeint automatisch entsprechend der vorhandenen Arbeitslast den verschiedenen Modulen einen gewissen Teil der Systemressourcen selbsttätig zu zuteilen. Da jede der Aufgaben intern durch eine oder eher mehrere C++ threads repräsentiert wird , ist die Anzahl pro Komponente natürlich gleichbedeutend mit der Verteilung der Systemlast. So werden im Fall einer komplizierten analytischen Aufgabe der „Analytical Engine“ einfach mehr threads zugeteilt. Dies geschieht voll automatisch, dem Administrator ist aber immer noch die Möglichkeit des Eingriffs gegeben.

Anhand der Graphik eindeutig erkennbar ist der Intelligence Server als Herzstück auch das einzige Modul innerhalb der 7i Architektur das für die Kommunikation mit Datawarehouse und dem Metadatenrepositorium ist. Diese zentralisierte Form der Kommunikation ist zwei

wichtig Aspekten sehr förderlich. Zum einen soll dadurch der administrative Aufwand gesenkt werden, zum anderen dient es der Sicherheit dem Web Server , oder anderen Diensten , keinen direkten Zugriff zu gewähren.

Ein weiterer großer Vorteil soll die plattformabhängige Generierung von SQL Anfragen sein, da je nach System unterschiedliche Extensionen anbietet zum Standard SQL. So werden die Anfragen auf das jeweils zugrunde liegende Datawarehouse optimal angepasst.

Um den Aufgaben eines Nutzers gerecht werden zu können der sich einen umfassenden Überblick über die wichtigsten Geschäftszahlen verschaffen will müssen eine ganze reihe von analytischen Funktionen zur Verfügung stehen. Die in vier Kategorien eingeteilte Funktionsbibliothek umfasst 150 analytische Funktionen. Zusätzlich ist es möglich noch weitere individuelle Funktionen zu erstellen. In der nachfolgenden Graphik sind die 4 Kategorien der Funktionsbibliothek dargestellt, und der Zusammenhang zwischen einer analytischen Anfrage, der „Analytical Engine“ und der „SQL Engine“ soll visualisiert werden.

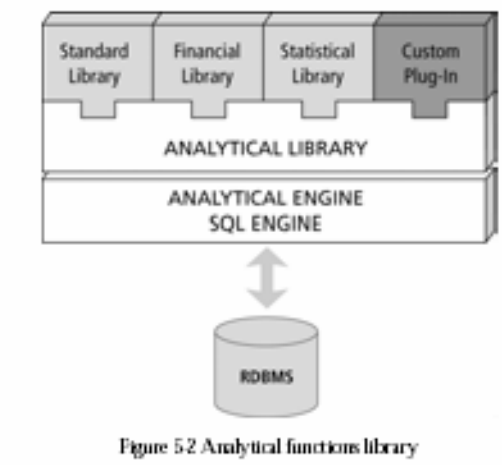


Figure 5.2 Analytical functions library

Die analytische Anfrage bedient sich der „Analytical Engine“ und diese wiederum nutzt für die Erstellung optimierter SQL Anfragen die „SQL Engine“. Die Funktionen stehen entweder im Standardteil (wie zum Beispiel Rankings oder Vergleiche) , dem Finanz orientierten Teil, dem Statistisch orientierten Teil, oder als selbst gestaltete Funktion im Plug-In.

Die Fähigkeit komplexe Analysen zu bearbeiten wir letztendlich durch den „Iterative Analysis“ Mechanismus bezeichnet. MicroStartegy will damit die Möglichkeit des Intelligence Servers in Worte packen Analysen zu bearbeiten die nicht durch eine einfache SQL Anfrage beantwortet werden können. In diesem Fall entscheidet der Intelligence Server, unter dem Minimierungsaspekt der Transferkosten , welche Berechnung in welchem Teil des Systems , zu welchem Zeitpunkt erfolgen kann.

Ein wichtiges Instrument um Anfragezeiten zu optimieren ist das Caching. Das in 7i implementierte Verfahren versucht die bereits gestellten Anfragen in den Cache zu legen, auf diese Art und Weise sind häufig wiederholte Anfragen, schnell, und ohne die Notwendigkeit die Datenbank einzuschalten, zu beantworten. Nach MicroStrategy soll dies den Effekt der Verlangsamung des Systems aufgrund hoher Benutzerzahlen deutlich senken. Aber immer in der gefälligen Annahme, dass so gut wie jeder ähnliche Anfragen stellt.

Das Hauptinstrument der multidimensionalen Datenhaltung sind die erwähnten „Intelligent Cubes“. Gleichzeitig spielen sie die Hauptrolle im Caching System. Mit ihrer Hilfe wird es erst möglich die eigentliche analytische Performanz zu erreichen. Wie im Anfrage-Ablauf bereits beschrieben, wird immer erst der Cache untersucht bevor eine Anfrage an die Datenbank weitergereicht wird. Im folgenden ist dieser Zusammenhang zwischen multidimensionalem Caching dem Datawarehouse und den Anwendungen nochmals verdeutlicht.

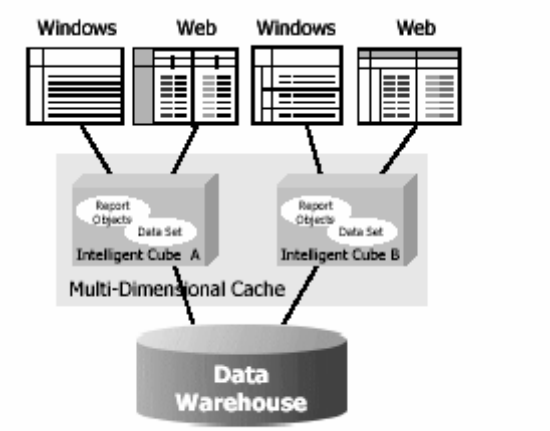


Figure 5.3 Intelligent Cubes™ in MicroStrategy 7i

Der Intelligence Server interagiert komplett selbstständig mit dem multidimensionalen Cache um die Notwendigkeit von Datenbankanfragen und Neuberechnungen so gering wie nur möglich zu halten.

ORACLE 9i

Während uns bei den beiden voran gegangenen Lösungen jeweils nur ein Produkt für die OLAP Seite präsentiert wurde, mit der Möglichkeit zur Anbindung an eine Datenbank, so kann ORACLE mit Produkten für beides aufwarten, und bietet als einziges der drei betrachteten Systeme die Möglichkeit einer Komplettlösung. Dieses und die Tatsache das ORACLE 9i sich eines hybriden OLAP Ansatzes bedient, wird von ORACLE in den

verschiedensten Zusammenhängen als die Hauptgründe für die, aus ihrer Sicht gegebene, Überlegenheit von ORACLE 9i genannt.

Wie immer wollen wir uns zuerst dem allgemeinen Überblick über die Komponenten und deren Zusammenhang verschaffen. Allerdings wollen wird dies entsprechend ORACLE in Verbindung mit einem Ablauf tun, dem Ablauf zum Erstellen eines Unternehmensweiten Business Intelligence Systems. Laut ORCALE ist dieser Prozess in drei aufeinander folgende Phasen zu unterscheiden, der „consolidation“, der „discovery“ und letztendlich dem „sharing“.
In jeder Phase spielen verschiedne Komponenten von 9i verschieden gewichtige Rollen.
Wir wollen die Komponenten in diesem Ablauf kurz nennen, und werden später noch mal tiefer auf verschiedene Aspekte und Module eingehen.

In der ersten Phase, der Konsolidierungsphase ist der ORACLE9i Data Warehouse Builder das Modul welche alle notwendigen Funktionalitäten zur Verfügung stellt. Innerhalb der Konsolidierungsphase kann man nochmals verschiedene Stufen ableiten. So ist es zunächst einmal wichtig für eine Anbindung der Datenquellen zu sorgen und dann das Datawarehouse aufzubauen.

Der Datawarehouse Builder bedient sich beim Aufbau des Datawarehouses einem der Datenquelle entsprechenden „Integrator“ Element. Mit der Datenquelle verbunden werden mit Hilfe des Integrators Daten und Metadaten extrahiert. Ein anderes Element das vom Warehouse Builder im Zusammenhang mit der Erstellung eines Datawarehouses genutzt wird , ist das „warehouse module“. Dieses enthält Information über die Dimensionen die „facts“ , und andere Definitionen, welche in ihrer Summe den Plan für das zu erstellende Datawarehouse widerspiegeln. Nach dem diese Planungsphase abgeschlossen ist, soll die Datenbank nun bevölkert werden, dies macht der Datawarehouse Builder mit Hilfe entsprechender SQL Anweisungen, die wiederum von ihm selbst generiert wurden. Dabei ist der Code Generator in der Lage optimal auf die Datenquelle zugeschnittenes SQL zu erstellen. Ist das Datawarehouse nun fertig aufgebaut werden die Metadaten auch den restlichen Anwendungen zugänglich gemacht. Dies ermöglicht jeder Applikation ein „Verstehen“ der Strukturen innerhalb der Datenbank. Sobald die Datenbank als solches in Betrieb genommen worden ist, verlagert sich das Problem. Die Verwaltungen von Veränderungen ist nun der zentrale Punkt. Auch hier ist es wieder Aufgabe des Datawarehouse Builders die Datenbestände konsistent zu halten. Dies geschieht Hauptsächlich über einen Synchronisationsmechanismus zwischen den Datenobjekten in der Quelle und den dazugehörigen Metadaten im Repositorium.

In der „Discovery Phase“ hat das ORACLE9iAS Discoverer Modul die Schlüsselrolle inne. Da das Datawarehouse aufgebaut worden ist geht es darum die Möglichkeit für ad-hoc Anfragen und Analysen auszubauen, beziehungsweise, zu ermöglichen. Ein nützliches Instrument um die Anfragegeschwindigkeit zu optimieren ist die Nutzung von sogenanten „Business ares“, diese sind inhaltsorientierte Ausschnitte aus der Summe von analytischen Funktionen und relevanten Daten. Diese Ausschnitte werden im Discoverer Modul Administrator entwickelt, welches selbst wieder denn Datawarehouse Builder benutzt. Das direkte Zusammenwirken von Datawarehouse Builder und Discoverer erleichtert natürlich die einfach Bevölkerung mit Daten. Die Hauptaufgabe des Discoverer ist es aber alle komplexen Sachverhalte von Systemseite unsichtbar zu machen, und dabei eine Oberfläche zu bieten die sehr sehr hohen analytischen Ansprüchen gerecht werden kann. So werden natürlich die unterschiedlichsten Funktionalitäten unterstützt, wie drilling , sortieren oder Vergleiche. Zudem ermöglicht das Modul die einfache Erstellung individueller Funktionalitäten, indem es viele geschäftlich relevante Funktionen als Bausteine für das Erstellen eigener Templates zur Verfügung stellt.

In der dritten Phase geht es darum die Ergebnis die man Aufgrund der Arbeit in der zweiten Phase erlangt hat, in geeigneter Form und Geschwindigkeit zu verbreiten, beziehungsweise dem Rest der Firma zugänglich zu machen. Dies geschieht in dem die Discoverer Komponente ihre Ergebnis an die ORACLE9i Reports Komponente via XML Datei übermittelt. Eine weitere Form in der die entsprechenden Ergebnisse dem gesamten Unternehmen zugänglich gemacht werden können ist die Möglichkeit sie als Webpages zur Verfügung zu stellen. Dies wird durch die Interaktion der Discoverer und der ORACLE9iAS Portal Module möglich. Auch das Format indem die Reports sind essentieller Gesichtspunkt für die allgemeine Zugänglichkeit. Das Format, das jetzt hauptsächlich genutzt wird sind JSP (Java Server Pages) Seiten , diese Technologie erlaubt nicht nur die Darstellung im Webseitenformat, sondern auch die Editierung der Reports in einem HTML Editor. Innerhalb der entstandenen Page wird dann die Integration der unterschiedlichsten Datenquellen möglich. Ein weiterer Vorteil bei der Nutzung von JSP’s ist das im Fall eines Updates der entsprechenden Seite, alle damit Verknüpften Seiten einen automatisch mit aktualisiert werden. Da es hauptsächlich darum geht eine große Anzahl von Empfängern zu erreichen ist es sehr sinnvoll das nicht nur über AS Portal die Reports als Webseiten zu präsentieren oder via email zu verschicken sind, sondern ORACLE Reports es erlaubt selbsterstellte Erweiterungen in Form von Java Programmen zu zulassen. Dadurch wird es möglich die Reports auch per Fax oder FTP zu versenden.

Die Generierung dieser , und für alle anderen Applikationen wichtigen, Java-Applikationen werden durch den ORACLE JDeveloper sehr vereinfacht. Ganz besonders wird hier das Paket ORACLE9i Business Intelligence Beans hervorgehoben, welches fertige high-level Anwendungen für die unterschiedlichsten Anwendungen zur Verfügung stellt.

Die Hauptentwicklungsumgebung für die Applikationen in Bezug auf Analysen ist ORACLE 9i OLAP. Mit Hilfe der erwähnten BI Beans und einem QueryBuilder ,der SQL Kenntnisse hinfällig macht, ist ein leichtes komplexe wirtschaftliche Zusammenhänge zu modellieren.

Mit Hilfe von ORACLE 9i Data Mining versucht man neue Einblicke zu gewinnen und diese in entsprechenden Anwendungen zu verarbeiten. Dies wird ermöglicht durch das systematischen „mining“ des Gesamtdatenbestandes. Der Ausbau dieses Moduls kann unter Umständen der Neugewinnung von Kunden , oder der Identifizierung der besten Kunden dienen.

Das Modul ORACLE9iAS Personalization versucht eine echte individuelle Beratung für den Nutzer durch das System zu verwirklichen. Diese „Beratung“ bezieht auf das gerade bearbeitete Scenario uns soll Vorschläge für relevante Berechnungen oder ähnliches machen. Dazu gibt es die ORACLE9i Personalization Engine, welche Aufgrund von historischen Daten und den vorhandenen „Click“ Daten vom Browsen, konkrete Vorschläge machen soll.

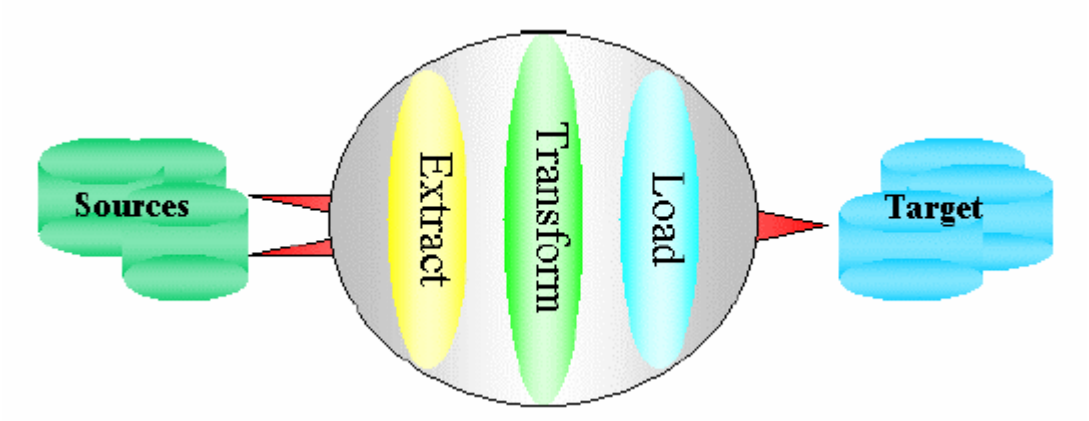
Wie schon erwähnt wird durch ORACLE9iAS Portal die Darstellung von Ergebnissen als Webseite möglich, aber dies ist nur ein Teil seiner Aufgaben. Mit dieser Komponente des Systems lassen sich komplette Web Portal einrichten. Es werden aber auch Teilbereiche wie Administration, Sicherheitsaspekte oder Anpassung der Inhalte abgedeckt.

Datawarehouse Builder

Wir wollen uns nun noch mal dem Datawarehouse Builder widmen, da ihm eine sehr zentrale Rolle zukommt. Tangiert haben wir seine Dienste in Bezug auf die erwähnte Phase 1 schon. Möchte man aber zusammenfassen für was dieses Tool innerhalb von 9i alles zuständig ist, so umfasst dies nicht nur den Aspekt der Anbindung von Datenquellen , es ist auch das Haupthilfsmittel zur Erstellung des Datawarehouses selbst , zur Generierung von Data Marts und der Entwicklung von BI Anwendungen. Der ORACLE Warehouse Builder kann als architektonischer Mittelpunkt der ORACLE Business Intelligence Tools innerhalb von 9i verstanden werden. Um die Entwicklungen und Anwendungen aus dem BI Bereich zu

unterstützen ist es notwendig ad-hoc Anfragen in geeigneter zu unterstützen, genauso wie mit der relationalen Datenbank umgehen zu können.

Mit Hilfe des Datawarehouse Builders wird der Benutzer in die Lage versetzt Datenquellen festzulegen, das Schema der entstehenden Datenbank zu generieren, denn Datenaustausch zwischen Quelle und Ziel zu regulieren (Siehe Abbildung) , die Erstellung der notwendigen OLAP Umgebung und ad-hoc Anfrage-Tools.

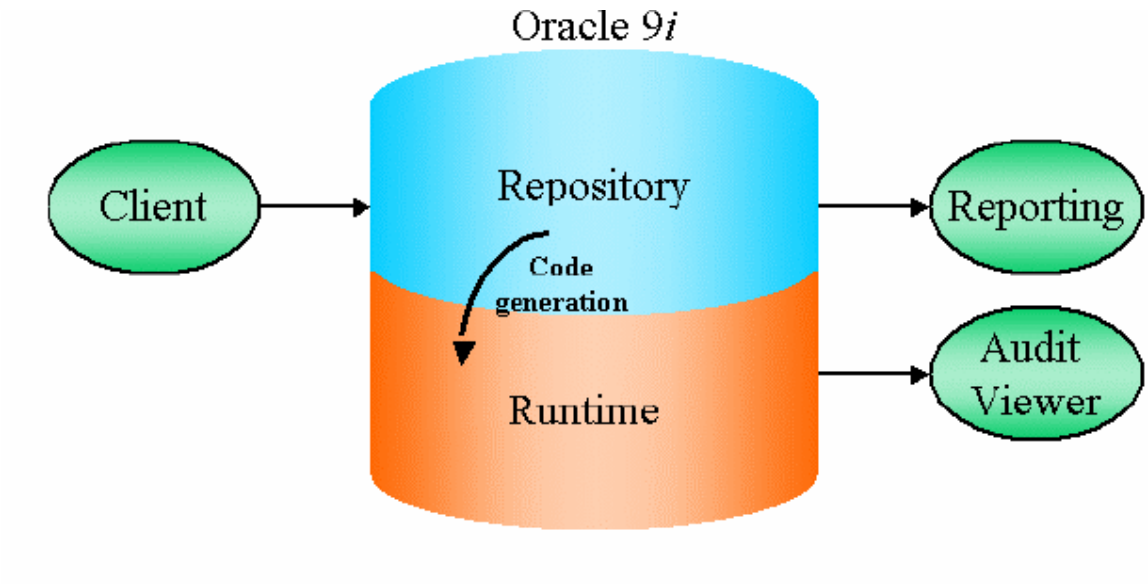


Die eingefügte Abbildung soll die Grundaufgaben vom Warehouse Builder darstellen.

Um der Fülle seiner Aufgaben gerecht werden zu können , besteht natürlich auch der Warehouse Builder wieder aus einzelnen Komponenten. Einige wollen wir hier nennen.
Das Repositorium ist auch hier zum halten der Metadaten und ihrer Definition gedacht.
Eine Client Anwendung die als Java Applikation verwirklicht ist, und eine einfach zu nutzende grafische Benutzeroberfläche darstellt, ist ein weiteres Element.

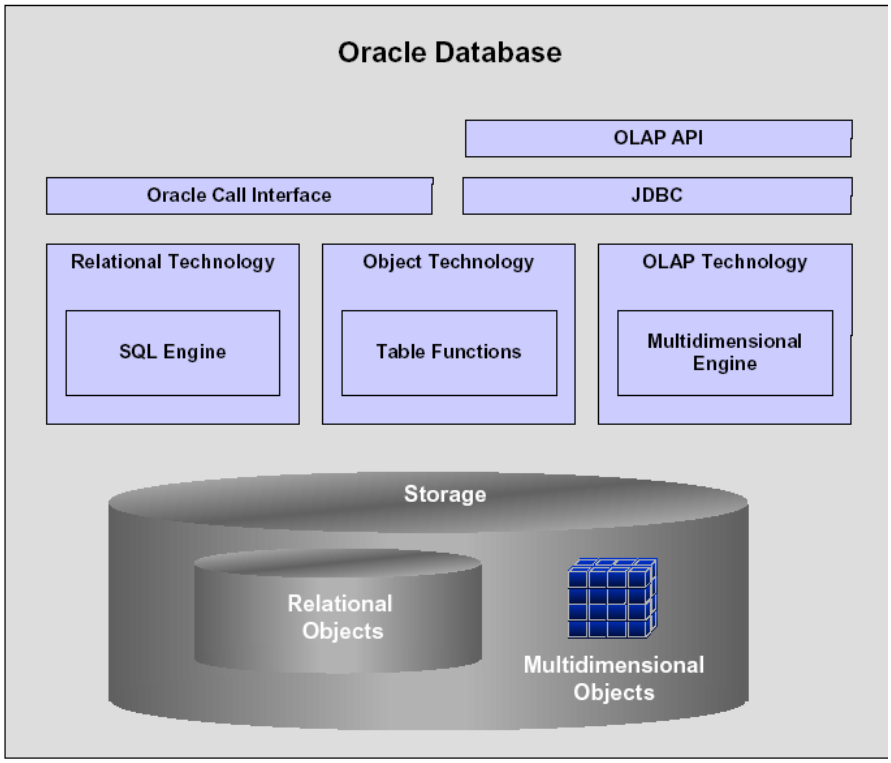
Ebenfalls von großer Bedeutung ist das Teil des Warehouse Builder welches mit Hilfe der Metadaten auf die Datenbank zu geschnittenes optimales SQL erzeugt, von ORACLE „Code Generator“ genannt. Um die Daten an die gewünschte Stelle , in dem gewünschten Schema, zu bekommen, und dabei auch noch über den Verlauf informiert zu werden , benötigt man eine dementsprechend ausgelegte Laufzeitumgebung.

Die zur Laufzeit erstellten Berichte werden mittels einer Java Anwendung kontrolliert. Zu guter letzt ist natürlich auch der Aspekt der öffentlichen Zugänglichkeit ein sehr wichtiger Aspekt. Mit dem „reporting Environment“werden die Metadaten jeder Anwendung im System bereitgestellt. Zur Visualisierung der gerade genannten Zusammenhänge habe ich die folgende Grafik eingefügt:



SQL/QUERY API's

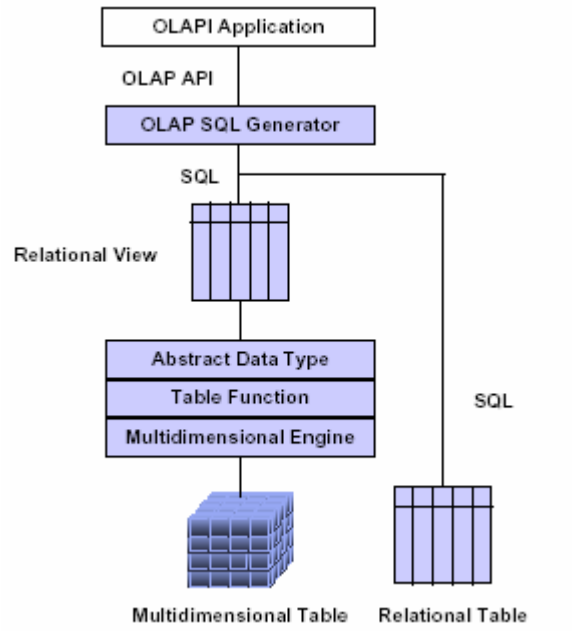
Ein weiterer Punkt mit dem wir uns im Zusammenhang mit ORACLE9i im speziellen befassen wollen, ist die Frage nach den verschiedenen API's. Dies ist natürlich auch zum einen wieder ein Aspekt den wir hier exemplarisch für viele andere Produkte aufgreifen können, ist aber aufgrund der Hybridarchitektur von ORACLE9i hier vielleicht von noch größerer Bedeutung.



Wie wir an dieser Abbildung gut erkennen können befinden sich innerhalb der Datenbank (hier „Storage“ genannt) sowohl relationale als auch multidimensionale Objekte. Um hieraus wirkliche Vorteile ziehen zu können muss es möglich sein, die Datenbank aus einer relationalen Sicht her anzusprechen, genauso wie ein Interface für objektorientierte Anwendungen bereitzustellen. Die „OLAP API“ ist die eben geforderte Schnittstelle für objektorientierte Anwendungen, zum Beispiel Java Applikationen. Der Zugriff auf die Datenbank wird über „JDBC“ ermöglicht. Die Alternative zu „JDBC“ stellt das „ORACLE Call Interface“ dar.

Wie wir an der vorangegangen Illustration leicht erkennen können stehen auf der Ebene des eigentlichen Zugriffs die „Multidimensional Engine“ , und die „SQL Engine“ zur Verfügung. Je nachdem, ob man ein relationales oder multidimensionales Objekt ansprechen möchte.

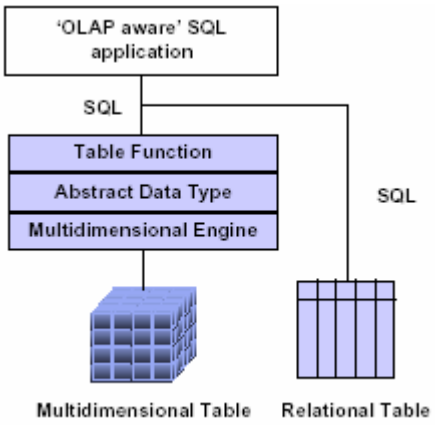
Damit allerdings ein hybrider Ansatz in vernünftiger Weise arbeiten kann, ist es notwendig Möglichkeiten gegenseitiger Integration zu bieten. Diese Funktion hat die „Objekt Technology“ mit den „Table Funktions“ inne. Mit Hilfe von Ihnen und dem Konstrukt von „ Abstract Data Types“ wird es möglich, zum Beispiel multidimensionale Strukturen innerhalb von SQL direkt anzusprechen. Der Verdeutlichung diese Inhalts dient die eingefügte Abbildung.



Wie wir hier sehen können bedient sich die „OLAP API“ einem entsprechenden „OLAP SQL Generator“ Mit dessen Hilfe SQL erzeugt wird , das sowohl relationale wie multidimensionale Konstrukte anspricht. Die „Table Functions“ sind dabei dafür zuständig

entsprechende „SELECT“ Anweisungen, oder andere SQL Anweisungen, in die multidimensionale Sprache DML zu übersetzen , so dass sie von der „Multidimensional Engine“ bearbeitet werden kann. Wie hier klar zu sehen ist, wird aus der multidimensionalen Tabelle eine relationale Sicht abstrahiert, auf denn sich dann die erwähnten SQL Anweisungen beziehen.

Eine art Abwandlung dieses Mechanismus sehen wir in dem nachfolgenden Bild dargestellt. „OLAP aware“ Funktionen werden als solche bezeichnet die sich direkt der „Table Functions“ bedienen. Diesen short cut, kann man im graphischen Vergleich leicht nachvollziehen.



Der direkte Aufruf von „Table Functions“ erlaubt den direkten Aufruf von „OLAP Stored Procedures“ als Teil der „SELECT“ Anweisung. Im folgenden SQL Beispiel spiegelt sich genau dieser Sachverhalt wieder.

```
SELECT sales
from
table(olaptf('AW_SALES' , 'AW_SALES_TBL' , 'call
ProductRanking('TOP' '10' 'ALL PRODUCTS' 'ALL CUSTOMERS'
YEAR2001')' , 'SALES.LIMIT.MAP')
where
CUSTOMER_ID = 'ALL CUSTOMERS' and
TIME_ID = 'YEAR2001'
```

Wie man sieht wird die Table Funktion „olaptf“ angesprochen und die Stored Procedure „ProductRanking“ verwendet.

Weitere Produkte

Nun haben wir für alle drei OLAP Ausprägungen ein Beispiel mehr oder minder im Detail betrachtet. Natürlich besitzen die Hersteller ORACLE, MicroStrategy und Hyperion ebenfalls eine enorme Marktsignifikanz, wie bereits in Kapitel 2 erläutert.

Im folgenden seinen noch weitere Produkte mit großer Marktbedeutung, kurz umrissen.

BusinessObjects

BusinessObjects wird bereits seit 1990 von dem gleichnamigen Hersteller angeboten. Wegen seiner besonderen Fokussierung auf Reportingfunktionalitäten richtet sich das Augenmerk als angestrebter Zielgruppe mehr auf das untere bis mittlere Management. Dessen Aufgabe es ist Informationen zu sammeln , zusammenzufassen und zu verbreiten. Bei der Verbreitung hilft eine ausgeprägte Web Komponente , in Kombination mit einer ausgefeilten Verteilungsfunktionalität. Analytische Ansprüche wie sie komplexeren Fragestellungen aus dem höheren Management Bereich zugrundeligen, sind trotz OLAP Komponente nicht zu erfüllen.

Cognos BUSINESS INTELLIGNECE SUITE

Aufgrund einer sehr hohen Flexibilität was die Applikationen belangt, ist die Cognos BI SUITE für alle Managementebenen interessant. Darüberhinaus ist sie nicht auf irgendwelche Branchen eingeschränkt, bietet aber die Möglichkeit auch sehr umfangreiche und komplexe Analysen auf den unterschiedlichsten Datenquellen zu realisieren. Der Kern eines jeden Cognos BI SUITE Produktes bildet dabei PowerPlay. PowerPlay ist ein Tool das es erlaubt multidimensionale Daten zu visualisieren , sei es in Form einer Graphik oder einer Tabelle.

Das Modul TRANSFORMER stellt denn von PowerPlay zur Ad-hoc Auswertung benötigten

multidimensionalen Datenwürfel zusammen. Dies kann entweder aus einer lokal gespeicherten multidimensionalen Struktur heraus geschehen, oder die Struktur kann mit Hilfe des PowerPlay ENTERPRISE SERVERs auch auf relationalen Datenbanken hinterlegt werden.

In diesem Zusammenhang ist auch noch das Modul IMPROMPTU zu erwähnen, das verschiedenste Schnittstellen zu relationalen Datenbanken bereitstellt , und es so dem TRANSFOMER ermöglicht daran anzuknüpfen.

KAPITEL 4 – Benchmark & Leistungsvergleiche

Da wir uns schon einen Marktüberblick, sowohl historisch gesehen , sowie von dem Aspekt der Marktanteile verschafft haben, wird uns dies bei der Einordnung der betrachteten Produkte sicher helfen. Auch nicht vernachlässigen wollen wir die 12 Punkte von Codd aus dem 1.Kapitel. Trotzdem wollen wir einige wichtige Punkte bezüglich Leistungsevaluierung bzw. anerkannte Benchmarks , wie den des OLAPCOUNCIL, gerne hier erwähnen.

Um sich für Produkte entscheiden zu können, möchte man gerne zum einen die Neuerungen der Produkte sehen. Vor allem interessiert einen Unternehmer ,bzw. eine Unternehmung nicht nur die doch sehr oberflächlichen verkaufsorientierten Informationen die man von den Herstellern selber erhält. Nein, es wäre wichtig zu wissen wie sich das angepriesene System in einer bestimmten , am besten einer am Arbeitstag orientierten Situation verhält. Geschieht dies auf einer Basis die es erlaubt mehre Systeme in der gleichen Situation miteinander zu vergleichen haben wir bestimmt das Instrument zur Kaufentscheidung.

Einer der wichtigsten Benchmarks ist der APB-1 des OLAPCOUNCIL. Bemerkenswert ist, dass OLAPCOUNCIL ein Konsortium von 11 Firmen ist , die zumeist selber Anbieter von DW und OLAP-Lösungen sind.

Der vom OLAPCOUNCIL veröffentlichte APB-1 Benchmark wurde mit dem Anspruch entwickelt die Anforderungen die eine Verkaufsanalyse an ein solches System stellen würde, nachzuvollziehen. Der Benchmark versucht zu berücksichtigen, welche Systeme eine Präprozessierung nutzen in Bezug auf manche Daten , und welche nicht, bzw. in anderen Bereichen nutzen. Die Datendichte wird als sehr gering vorausgesetzt, allerdings ist es dem Hersteller erlaubt sie innerhalb eines vorgegebenen Intervalls auszuwählen. Die Daten werden vom Benchmarkprogramm selbst generiert.

Der Benchmark selber ist in 3 Phasen aufgeteilt. In der ersten und der zweiten Phase werden je nach art des Systems eine gewisse Datenmenge verarbeitet bzw. vorverarbeitet, in der letzten werden dann Anfragen ausgeführt.

Die entscheidende Unterscheidungsgröße ist AQM (analytical queries per minute), diese Größe wird einfach als Quotient zwischen Bearbeitungszeit und der Anzahl der bearbeiteten Anfragen gebildet. Nachfolgend sind tabellarisch relativ (!) aktuelle Ergebnisse des Benchmarks für einige immer noch sehr marktrelevante Produkte dargestellt. Was an dem Beispiel auch gut sichtbar wird ist nicht nur die Performanz unterschiede der Produkte, sondern was für ein Einfluss die Datendichte auf die Leistung der Systeme hat.

	When published	Server hardware	Base data load & pre-calc (minutes)	Incremental data load & pre-calc (minutes)	Time for 250,000 queries (minutes)	Total query time (minutes)	Analytical Queries per Minute (AQM)	Queries processed per minute	Final database size (Mb)
Interpretation		***Less is better***	***Lower is better***	***Lower is better***	***Lower is better***	***Lower is better***	***Higher is better***	***Higher is better***	***Lower is better***
	APB-1 R1 Benchmarks with 0.011 percent density data:								
Essbase 4.1	June 1997	4xPentium Pro 200MHz, 1Gb RAM	5	304	350	654	**382**	714	7200
Express 6.1	October 1997	4xPentium Pro 200MHz, 1Gb RAM	166	83	44	127	**1972**	5682	2580
TM1 Server 6.0	December 1997	2xPentium Pro 200MHz, 0.5Gb RAM	41	13	147	160	**1562**	1701	23
Essbase 5.0	March 1998	4xPentium Pro 200MHz, 1Gb RAM	2	11	36	47	**5293**	6849	347
Express 6.1 (Solaris)	May 1998	4xUltra-Sparc 300MHz, 1Gb RAM	*Not disclosed*	11	20	31	**8072**	12,346	616
Essbase 5.0.1 (Solaris)	November 1998	4xUltra-Sparc 400MHz, 1Gb RAM	0.5	5.1	10	15.1	**16,221**	25,818	351

	APB-1 R1 Benchmarks with 0.55 percent density data:								
Express 6.1 (Solaris)	May 1998	4xUltra-Sparc 300MHz, 4Gb RAM	*Not disclosed*	259	34	293	**852**	7346	11,468
Express 6.2 (Xeon)	November 1998	4xIntel Xeon 400MHz, 2Gb RAM	*Not disclosed*	181	34	215	**1161**	7386	11,472
Essbase 5.0.1 (Solaris)	November 1998	4xUltra-Sparc 400MHz, 4Gb RAM	41	132	16	148	**1690**	15,259	6215
	APB-1 R2 Benchmark with 5 percent data density (4.56Gb input data):								
Essbase 5.0.2 (IBM AIX)	April 1999	4xIBM RS64-II 340MHz, 8Gb RAM	121	283	158 (500,000 queries)	441 (500,000 queries)	**1135** (500,000 queries)	3173	

Leider ist es mir nicht gelungen einen kostenfreien, oder auch nur kostengünstigen, Benchmarkvergleich mit ganz aktuellen Produkten , und entsprechender Hardware zu finden.

Neben dem OLAPCOUCIL Benchmark wird auch oft der „MicroStrategy StressTest“ erwähnt, dieser wurde im Gegensatz zum OLAPCOUNCIL nur von MicroStrategy selber entwickelt, ohne Rücksicht auf weitere Hersteller, etc.

Leider ist dieser Benchmark, wie fast jeder herstellerinterne Benchmark, auf sehr unrealistischen, zugeschnitten Voraussetzung aufgebaut worden. Deshalb kann er eigentlich nur ein Beispiel dafür sein, daß Benchmarks, die von einer Interessengruppe entwickelt wurden, ihren Sinn gänzlich einbüssen, nicht nur in der Einzelbewertung , sondern auch komplett in Bezug auf einen vernünftigen Produktvergleich.

ANHANG

Quellenverweise:

Es wurden keine direkten Zitate verwendet. Die benutzten Grafiken kamen zu großen Teilen aus den angegeben Internetseiten, allen voran olapreport.com. Nichtsdestotrotz habe ich mich auf verschiedene Bücher, Aufsätze und Internetseiten gestützt, diesen seinen im folgenden aufgezählt.

1. Bücher:

- Der Datawarehouse Spezialist / Reinhard Höhn / Addison-Wesley
- Datenbanksysteme / Conolly, Begg, Strachan / Addison-Wesley
- Analytische Informationssysteme / Chamoni, Gluckowski / Springer
- Data Warehouse und Data Minig / Schinzer, Bange , Mertenes / Vahlen

2. Internetseiten:

- www.olapreport.com
- www.olapcouncil.com
- www.hyperion.com
- www.oracle.com
- www.cognos.com
- www.businessobjects.com
- www.microstrategy.com

3.Papers:

1. “An Overview of Data Warehousing and OLAP Technology” *Chaudhuri,Dayal*

2. “Oracle9*i* Warehouse Builder” *Oracle White Paper*

3. “Oracle 9i Applikation Server: Business Intelligence Overview” *Oracle White Paper*

4. “The Role of OLAP Server in a Data Warehousing Solution” *Hyperion White Paper*

5. “The origins of today’s OLAP products” *The OLAP Report*