1 Ursprüngliche Frage / Problemstellung / Vorgeschichte

1.1 Entwicklung bis zum "Spiderweb"

Das, was heute unter einem Datawarehouse verstanden wird, gehört zur Gruppe der DSS ("decision support systems"), deren Ursprünge bis in die 60er Jahre des 20. Jahrhunderts zurückreichen. Ziel solcher DSS ist es, die Entscheidungsfindung eines Firmenmanagements durch Fakten bzw. Schlüsse, die aus den Firmendaten gewonnen werden, zu unterstützen.

Abbildung 1 zeigt die Entwicklung dieser DSS von Anfang der 60er bis zum System, das heute als "Datawarehouse" bezeichnet wird. In den folgenden Abschnitten werden die einzelnen Stationen näher beleuchtet.

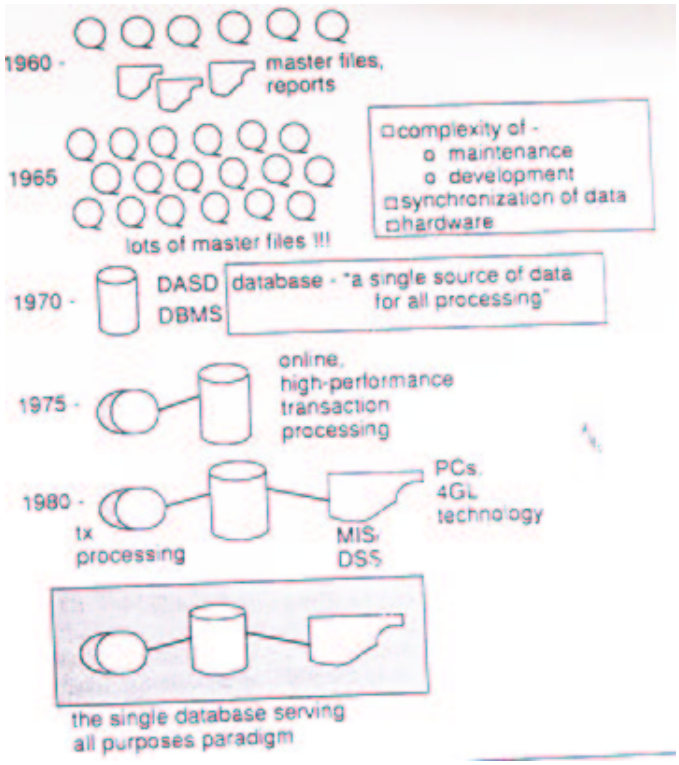

Abbildung 1: Entwicklung der "decision support systems"

Bis ca. zur Mitte der 60er-Jahre bestand die Welt der DSS aus sogenannten "master files", die die Daten auf Magnetbändern enthielten. Für die jeweilige Anwendung wurden eigene Programme geschrieben, die Reports für die Daten lieferten. Die Anzahl solcher "master files" nahm jedoch stark zu, was zu einigen erheblichen Problemen führte:

- Synchronisation der Daten bei einem Update

- Hohe Komplexität bei der Wartung und Entwicklung neuer Programme
- Die Hardware war den Anforderungen nicht gewachsen

Speziell der letzte Punkt stellte ein sehr grosses Problem dar: zu dieser Zeit gab es – wie oben bereits erwähnt – lediglich Magnetbänder als Speichermedium. Wollte man nun einen Block am Ende des Bandes auslesen, war es nötig, das gesamte Band durchzuspielen, was bis zu 30 Minuten dauern konnte (wie in [Inm92] dargestellt).

Anfang der 70er-Jahre kam es dann jedoch zur Entwicklung des sogenannten "direct access storage device" (DASD) – darunter fallen alle heute üblichen Speichermedien: Harddisks, CD-ROM, usw. Im Gegensatz zum Magnetband konnte nun jeder Sektor des Mediums einzeln ausgelesen werden, ohne alle vorhergehenden Sektoren betrachten zu müssen. Zusätzlich zu diesen DASD wurden die ersten DBMS ("database management systems") entwickelt. Aufgabe dieser DBMS war es, dem Programmierer die Speicherung, Indizierung, ... der Daten zu erleichtern. Durch diese Technologien war es nun erstmals möglich, Daten schnell und effizient zu adressieren, was die Tür zu neuartigen Systemen (z.B. Reservierungssysteme, Kontrollsysteme zur Überwachung von Produktionsvorgängen, usw.) öffnete.

Mitte der 80er-Jahre wurde ein neues Programm immer populärer: das "extract"-Programm (dargestellt in Abbildung 2).

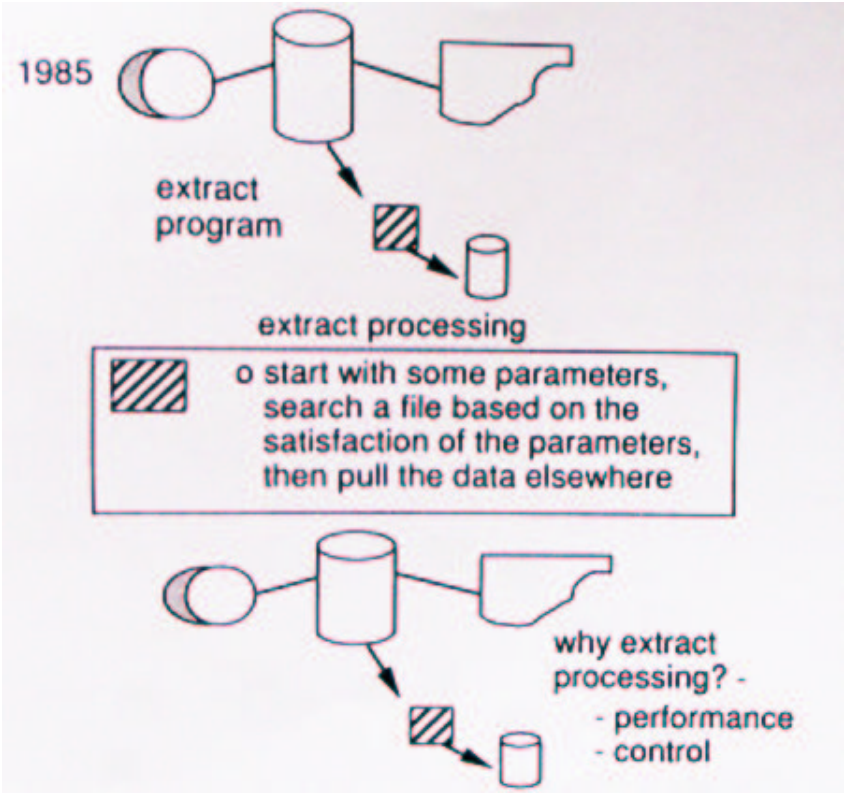

Abbildung 2: Funktionsweise der extract-Programme

Von der Funktionalität bietet das "extract"-Programm sehr wenig: es durchforstet Dateien, extrahiert Daten nach bestimmten Kriterien und speichert diese Daten an einen anderen Ort. Die wesentlichen Gründe für die Akzeptanz dieser Programme lagen in zwei Faktoren:

- Performance
- Datenkontrolle

Durch das Speichern an einen anderen Ort kam es bei – auch grossen – Datenanalysen zu keinen Performanceproblemen. Ausserdem wechselte die Kontrolle über die extrahierten Daten zu der Person, die das extract-Programm gestartet hatte, dem User "gehörten" also die Daten. Gleichzeitg führte die Verwendung solcher extract-Programme zu sehr grossen Problemen, die im folgenden Abschnitt näher erläutert werden.

1.2 Das Spiderweb

Der Einsatz der extract-Programme blieb jedoch nicht auf eine einzelne Datenbasis beschränkt, sondern extract-Programme extrahierten Daten aus Datenbasen, die ebenfalls Resultat eines extract-Prozesses waren. Inmons schreibt in [Inm92], dass es für eine grosse Firma durchaus üblich war, bis zu 45000 extracts pro Tag durchzuführen. Dies führte in Unternehmen schliesslich zu einer sogenannten "naturally evolving architecture", welche auch unter dem Namen "spiderweb" bekannt ist (siehe Abbildung 3).

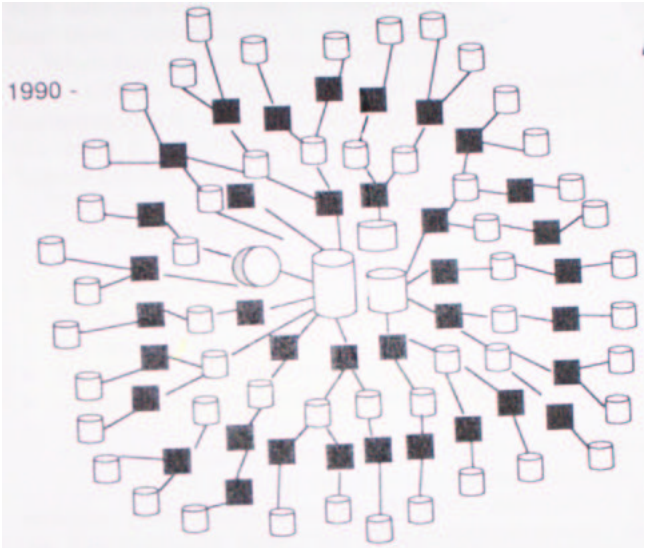

Abbildung 3: Aufbau einer "naturally evolving architecture" – aka Spiderweb

Spiderwebs führten jedoch zu einer Reihe von Problemen:

- Glaubwürdigkeit der Daten

- Produktivität
- Daten, aber keine Informationen

1.2.1 Glaubwürdigkeit der Daten

Abbildung 4 illustriert das Problem der Glaubwürdigkeit der Daten:

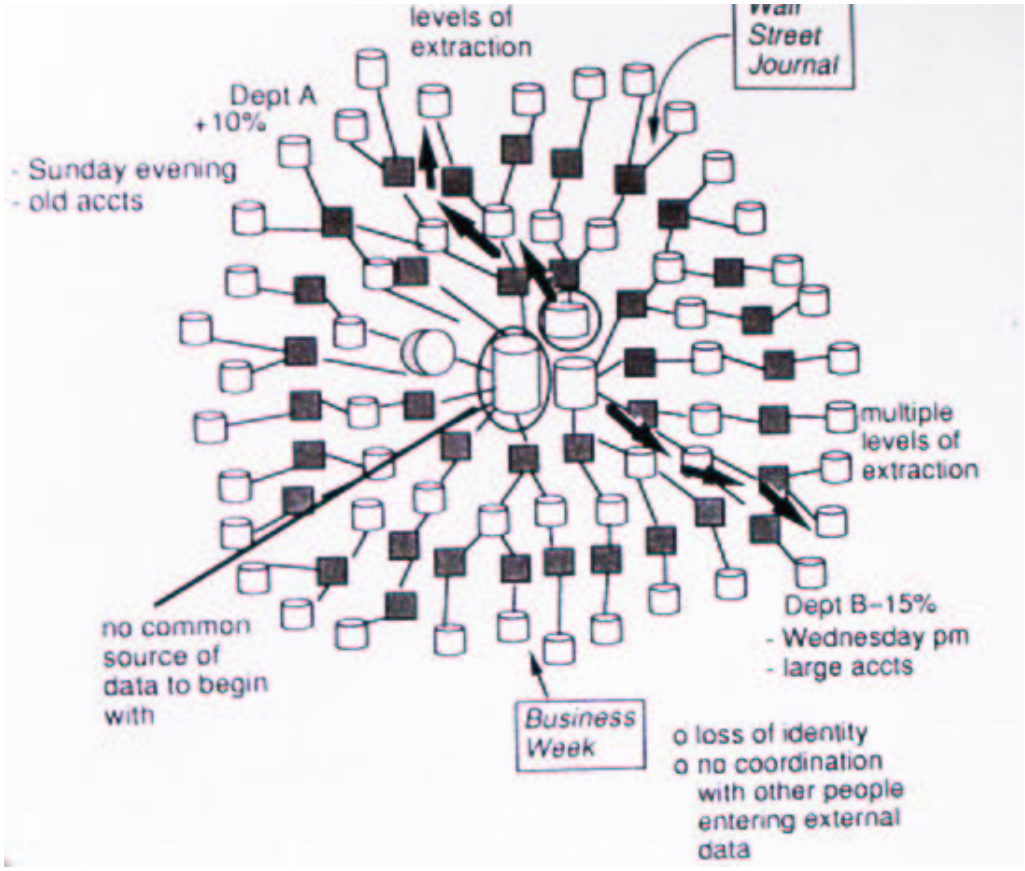

Abbildung 4: Verschiedene Ergebnisse der gleichen Auswertung

obwohl beide Abteilungen die gleiche Auswertung durchführen und daher auch das selbe Ergebnis erhalten sollten, ist dies nicht der Fall. Dies hat mehrere Ursachen:

- Kein gemeinsamer Zeitpunkt der Auswertung
- Tiefe der Extraktion
- Probleme des Einbindens von externen Daten
- Beginn der Extraktion bei unterschiedlichen Datenbasen

Startet man Reportprozesse zu **verschiedenen Zeitpunkten**, so ist das Ergebnis dieser Reports nur in den seltensten Fällen gleich, da sich Daten innerhalb eines Unternehmens klarerweise sehr schnell ändern können. Darüberhinaus stellt das Ergebnis eines extract-Programms nur einen "snapshot" einer anderen Datenbasis zu einem bestimmten Zeitpunkt dar. Es ist daher bei einer "naturally evolving architecture" rein zufällig, dass zwei Auswertungen dasselbe ergeben.

Darüberhinaus stellt die **extract-Tiefe einer Datenbasis** ein weiteres Problem für die Glaubwürdigkeit der Daten dar. Stellt eine Datenbasis das Ergebnis mehrerer extract-Prozesse dar, so ist es wahrscheinlicher, dass Daten bereits im Vorfeld weggefiltert wurden, die für die aktuelle Auswertung wesentlich sind.

Ein drittes Problemfeld stellt das Einbringen externer Daten dar: sobald jemand Daten einer anderen Quelle in das System eingebracht hat, ist die Unterscheidung zwischen internen und externen Daten in einem Spiderweb nicht mehr möglich (sofern nicht in diesen externen Daten entsprechende Vorkehrungen getroffen wurden).

Ausserdem ist in einer "naturally evolving architecture" möglich, dass Auswertungen bei verschiedenen Datenbasen starten und daher die Resultate verschieden sind.

1.2.2 Produktivität

Durch die Architektur eines solchen Spiderwebs ergeben sich zwangsläufig Probleme mit der Produktitvität. Will man eine Analyse eines Bereiches, den ein gesamtes Unternehmen betrifft, sind folgende Schritte notwendig:

- Die benötigten Daten werden gesucht
- Die Daten werden für die Analyse aufbereitet

Der Suchprozess kann sich durch den "Wildwuchs" an Daten, die extrahiert, extern hinzugefügt, usw. wurden, sehr verzögert werden. Um die gewünschten Daten zu erhalten, müssen unter Umständen **sehr viele** Dateien durchsucht werden (dabei können Dateien z.B. auch gleiche Namen haben, aber verschiedenen Inhalts sein, ...).

Sind die Daten verfügbar, müssen diese für die Analyse aufbereitet werden. Erschwerend kommt in diesem Arbeitsschritt hinzu, dass die Programme dafür für ein jede Analyse angepasst werden müssen, da die Daten jedes Mal (nicht nur von der Information her) andere sind.

1.2.3 Daten, aber keine Informationen

Ein weiteres grosses Problem des Spiderweb ist die Umsetzung der vorhandenen Daten zu Information. Das Spiderweb hat – aus der Entstehungsgeschichte heraus – zwei grosse Probleme:

- Integration war nie ein Designkonzpt
- Nur aktuelle Daten wurden benötigt, alte Daten wurden nicht berücksichtigt

Durch ungehindertes Hin- und Herkopieren innerhalb des Spiderwebs ist es nicht möglich, diese Daten wieder auf eine einzelne Datenbasis zurückzuführen. Man hat also die Daten in verschiedenen Files liegen, es gibt jedoch keine Zusammenhänge zwischen den einzelnen Files. Darüberhinaus wurden alte Daten meist nicht beachtet, da das Spiderweb für das sogenannte "current balance processing" geacht war. Die "Vorgeschichte" der Daten wurde daher nicht gespeichert.

1.3 Paradigmenwechsel

Durch die Probleme, die das Spiderweb mit sich brachte, war ein Paradigmenwechsel notwendig: weg vom Spiderweb, hin zur "architected datawarehouse environment". Das neue Prinzip war, dass es zu einer Trennung zwischen Daten aus operativen Systemen und daraus abgeleiteten Daten gab.

Operative Daten werden jeweils auf den neuesten Stand gebracht, sind detailliert und bilden die Grundlage für die **abgeleiteten Daten**. Diese richten sich primär an die Manager eines Unternehmens (Inmons spricht in [Inm92] auch von "DSS data") und bilden die Zusammenfassung (möglicherweise) redundanter Daten.

Aufgrund der oben angeführten Eigenschaften war eine "naturally evolving architecture" nicht mehr möglich. Stattdessen kam eine vierschichtige Architektur, wie sie in Abbildung 5 zu sehen ist:

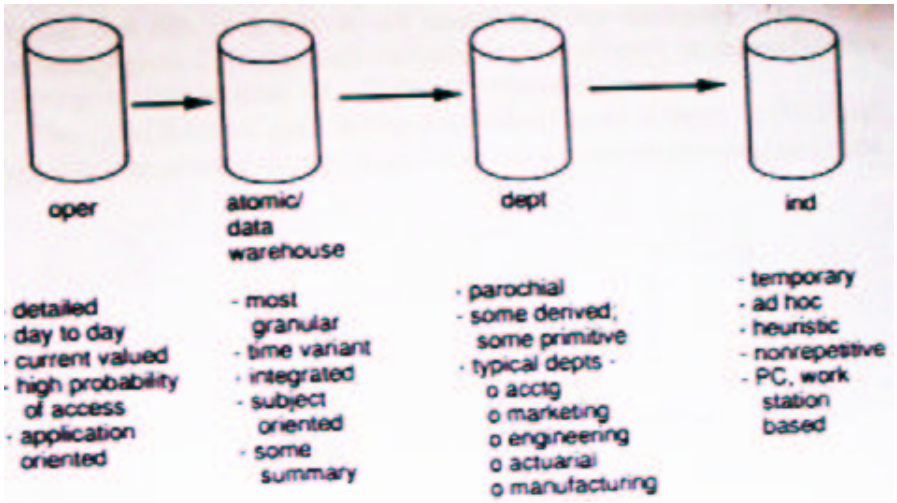

Abbildung 5: Aufbau einer der vierschichten Architektur

In [JB97] geben die Autoren eine Definitionen eines Datawarehouses: "(...) Als zentrales Datenlager sammelt es in regelmässigen Intervallen Informationen aus den operativen Systemen, konsolidiert sie, filtert Unwichtiges heraus, ordnet die Daten nach Themen und zeigt mit Hilfe von Analysewerkzeugen Relevantes an."

Nach unserer Definition ist das Datawarehouse jedoch nur das oben beschriebene Zentrallager für Daten, die Auswertungen werden jedoch auf den Stufen 3 und 4 durch-

geführt. Wie diese Auswertungen durchgeführt werden, findet man in den Kapiteln der Gruppe "data mining" (**VERWEIS!**).

2 Bild des Users

Walter Inmon, der "Vater" des Datawarehouse hat in seinem Buch [Inm92] eine klare Vorstellung davon, wer die zukünftigen Nutzer eines Datawarehouse sind:

> The user of the data warehouse is a person who can be called the DSS analyst. The DSS analyst is first and foremost a business person and secondly, a technician. The primary job of the DSS analyst is in the definition and discovery of information used in the decision making of the corporation."

Weiters spricht er davon, dass ein Datawarehouse-User erst nach und nach die Möglichkeiten des Datawarehouse entdeckt. Er spricht von

> "give me what I say I want, then I can tell you what I really want"

3 Umfeld der Entwicklung

Wie in den bisherigen Abschnitten bereits erläutert, war für die Entwicklung des Datawarehouse einige Voraussetzungen notwendig. In [Gup02] spricht der Autor von zwei wesentlichen Voraussetzungen:

- Rasante Entwicklung im Bereich der Hard- und Software
- Globalisierung bzw. die Änderung bestehender Geschäftsmodelle

3.1 Hard- und Software

Erst duch die Entwicklungen im Bereich des Hardware ist die Realisation eines Datawarehouse möglich. Zu Beginn der Entwicklung der DSS war es undenkbar, mehrere Hundert Gigabyte an Daten zu speichern bzw. schnell nach Informationen durchsuchen zu können. Doch durch die günstigen Fertigungstechniken für Mikroprozessoren und der Verdopplung der Leistung der Prozessoren alle 18 Monate ("Moore's Law") sind diese Architekturen möglich geworden. Durch die Leistungssteigerung, die auch im PC-Bereich stattgefunden hat und noch immer stattfindet, ist aber erst auch der Ansatz von Inmons vielschichtiger Architektur möglich geworden. Analysen werden nicht auf grossen Servern, sondern auf den jeweiligen PCs der Analysten durchgeführt.

3.2 Wirtschaftliche Entwicklung

Durch die wirtschaftlichen Entwicklungen, die seit Mitte der 80er-Jahre stattfanden, war eine Unterstützung des Managements bei der Entscheidungsfindung durch ein Datawarehouse unabdingbar. Durch die Globalisierung der Märkte müssen Unternehmen in Wettstreit mit Mitbewerbern aus anderen Ländern und Kulturen treten. Für das Top-Management hingegen ist das "big picture" notwendig, was beispielsweise durch ein Spidweb niemals erreicht werden könnte.

Literatur

[Gup02] Vivek Gupta. An Introduction to Data Warehousing, 2002. `http://system-services.com/dwintro.asp`.

[Inm92] Walter Inmons. *Building the Data Warehouse*. John Wiley & Sons, Inc., 1992.

[JB97] Dr. Rudolf Munz Jo Bager, Jörg Becker, 1997. c't 3/97.