

Institutionen för informatik  
Handelshögskolan vid Göteborgs Universitet  
Magisteruppsats 20p VT-99



# Management av heterogenitet & Data Warehouse

En studie om heterogenitet och hur den hanteras av  
Data Warehouse

Mattias Erlandson  
&  
Kianoush Etemadseifi

## Abstrakt

Bakgrunden till denna studie är att företagens informationsförsörjning har blivit mer och mer heterogen, samtidigt som kraven på relevant beslutsinformation har ökat under senare tid. Uppsatsen tydliggör de olika typer av heterogenitet som man kan finna mellan olika system samt utvärderar Data Warehouse lämplighet och applicerbarhet ur ett IT- Management perspektiv.

Uppsatsen presenterar en modell för att överblicka den heterogenitet som man kan finna. Stöd för modellens riktighet finns i den tidigare forskning som utförts inom området. Undersökningens tydligaste begränsning har varit att i fallet med Data Warehouse så har man inte tidigare analyserat detta utifrån ett IT-management perspektiv.

**Handledare: Fil.Dr Thanos Magoulas**

## Innehållsförteckning

1.	Inledning .....	1
1.1.	Bakgrund .....	1
1.2.	Syfte .....	2
1.3.	Avgränsning och problemställning .....	2
1.4.	Disposition .....	2
2.	Utredningsmetodik .....	4
2.1.	Synsätt .....	4
2.2.	Metodik och arbetsätt .....	4
3.	Heterogenitetens teoretiska förutsättningar .....	6
3.1.	Systemtänkande och informationssystem .....	6
3.2.	Kartläggning av heterogenitetens källor .....	8
3.2.1.	Den Kulturella heterogeniteten .....	8
3.2.2.	Den Strukturella heterogeniteten .....	9
3.2.3.	Den funktionella heterogeniteten .....	9
3.2.4.	Den kognitiva heterogeniteten .....	10
3.2.5.	Den informationsteknologiska heterogeniteten .....	11
3.2.6.	Filosofiska konflikter .....	12
3.3.	Ett klargörande av heterogenitet ur informatisk synvinkel .....	14
3.3.1.	Beskrivning av heterogenitet ur informatisk synvinkel .....	15
3.3.2.	Beskrivning av begreppsmässig heterogenitet .....	16
3.3.3.	Beskrivning av innehållsmässig heterogenitet .....	20
3.3.4.	Beskrivning av regelbaserad heterogenitet .....	22
3.3.5.	Beskrivning av andra former av heterogenitet .....	24
4.	Data Warehouse konceptet .....	25
4.1.	Historik och dagsläge .....	25
4.2.	Data Warehouse arkitektur .....	27
4.3.	Data Warehouse konceptets tolkning av heterogenitet .....	31
4.4.	Data Warehouse Arkitektur för hantering av heterogenitet .....	32
4.5.	Data Warehouse konceptets grundläggande principer .....	36
4.5.1.	Abstraktionsprincipen .....	36
4.5.2.	Integrationsprincipen .....	39
4.5.3.	Aggregeringsprincipen (Granularity) .....	40
4.5.4.	Evolutionsprincipen ( Informationens tidshorisont) .....	41
4.6.	Migration till Data Warehouse miljö .....	45
4.6.1.	Heterogenitet i källsystem .....	45
4.6.2.	Design av dimensionstabeller .....	47
4.6.3.	Migrationsprocedur .....	48
4.6.4.	Absorbering av heterogeniteten .....	49
5.	En kritisk analys av Data Warehouse arkitekturella principer .....	54
5.1.	Abstraktionsprincipen .....	54
5.2.	Integrationsprincipen .....	56
5.3.	Aggregeringsprincipen .....	58
5.4.	Evolutionsprincipen .....	59
5.5.	Sammanfattning .....	60
6.	Resultat .....	61
6.1.	Heterogenitet och Data Warehouse .....	62
6.2.	Hur användbart är Data Warehouse konceptet för att hantera heterogenitet? .....	64
6.3.	En studie av två systems information .....	65

7.	Diskussion.....	69
7.1.	Data Warehouse koncept.....	70
7.2.	Validitet.....	77
7.3.	Reliabilitet.....	77
8.	Slutsats.....	78
9.	Litteraturförteckning.....	79

## **Förord**

Vi vill tacka vår handledare Fil.Dr Thanos Magoulas som med faderliga råd, stor kunnighet och stort tålamod väglett oss under arbetets gång.

Vi vill också tacka våra respektive familjer, Nahid och Lina vilkas stöd, tålamod och kärlek gett oss styrkan att slutföra detta arbete.

Göteborg maj 1999

Kianoush Etemadseifi och Mattias Erlandson

# 1. Inledning

## 1.1. Bakgrund

Anledningen till att vi har valt att göra denna studie är att idag så ökar kraven på informationen inom organisationer. Företagen har idag ofta tillgång till mycket stora mängder intern data, detta samtidigt som de lider brist på kvalitativt bra information. Den information som de vill ha kan sägas drunkna i havet av data i de olika systemen. Problemet är att skillnaden mellan de olika systemen kan vara mycket stor vilket gör det svårt att utvinna bra information. Idag så finns det olika sätt för att komma tillrätta med detta problem. Ett sätt är Data Warehouse. Detta kommer vi att behandla i kapitel 4. Vi kommer även att undersöka den heterogenitet som råder mellan olika system inom och utom organisationernas gränser. Med andra ord, bakgrunden till vår studie är att företagens informationsförsörjning har blivit mer och mer heterogen. Vi anser att källan till denna heterogenitet är bland annat:

- Informationsförsörjningens natur.
- Olika teknologier för att tillgodose verksamheternas informationsbehov.
- Användandet av olika konsulter, systemutvecklare m.m. i utvecklingsprojekt.
- Olika vägledningsmodeller etc.

Man kan generellt säga att ju större differentiering och heterogenitet som råder mellan systemen, desto större blir kravet på integration mellan dessa system. Problemet här är hur överblickar vi och därmed hanterar heterogeniteten. Heterogenitetsproblemet är ett av de främsta problemen som man bör ta i beaktande då två eller flera system skall integreras.

En systemtyp som är under utveckling idag och som vinner mark bland organisationer, är som ovan nämnt, Data Warehouse. Ett Data Warehouse samlar in information från multipla databaser inom och utanför organisationen vilket medför att heterogenitetsproblemet bör tas i beaktande. Anledningen till att Data Warehouse har utvecklats är att det finns krav i dagens organisationer på att lösa de komplexa informationsförsörjningsproblemen som har uppkommit till följd av allt högre krav på mer kvalitativt bra beslutsinformation. Med hjälp av ett Data Warehouse så utvinns beslutsinformation ur organisationens befintliga system. Anledningen till att man gör detta är, att det finns en växande kunskap om värdet på den befintliga data organisationen har. Data i dessa system är dock mer eller mindre heterogen vilket påverkar informationsförsörjningen negativt. Om problemen med informationsförsörjningen skall kunna lösas, så är det av yttersta vikt att en förståelse för grundproblemet finns.

För att på ett mer begripligt sätt visa vad vi menar med heterogenitet kommer vi nedan att presentera ett exempel.

Antag följande: En bank använder sig av flera separata banksystem, ett för kontanter, ett för lån och slutligen ett för kredithantering. Antag vidare att en kund begär att få en överföring på 100.000 kr från ett av sina konton till ett externt konto. Om kundens saldo på kontot inte kan täcka transaktionen så kommer den att stoppas. Detta sker även om kunden har en beviljad kredit på

1.000.000 kr. Detta kommer troligtvis att vara irriterande för kunden samtidigt som det kommer att resultera i att man måste gå in manuellt och ta av krediten för att det skall bli möjligt att täcka transaktionen.

Om banken istället kan koppla ihop de tre separata databaserna så att man från ett godtyckligt system kan få tillgång till information som finns i de andra systemen samt göra överföringar mellan systemen så kommer detta att underlätta arbetet för användaren. Utöver användarens syn så kommer man även att få en bättre kundhantering vilket resulterar i starkare konkurrenskraft. Vid en integration av detta slag kommer man troligen att stöta på ett flertal problem vilka kan vara av olika svårighetsgrader.

## 1.2.Syfte

Dagens system presenterar den efterfrågade informationen på olika sätt och inte alltid på ett enkelt, lättbegripligt och överskådligt sätt. Det vi försökt att åstadkomma med detta arbete är att påvisa de olikheter man kan finna mellan olika system som kan vara föremål för en integration. Enligt etablerade teorier skiljer sig systemen oftast åt med hänseende på regelbaser, informationsbaser, begrepps-baser, med mera. Detta innebär en ökad heterogenitet systemen emellan denna heterogenitet har varit föremål för vår undersökning i denna rapport. Vidare har vi även undersökt Data Warehouse konceptet ur ett infologiskt perspektiv för att försöka utröna dess applicerbarhet, lämplighet i olika situationer. Vad vi skall försöka åstadkomma är att påvisa för- respektive nackdelar i samband med införande av ett Data Warehouse koncept, samt användbarheten av ett sådant koncept.

## 1.3.Avgränsning och problemställning

Vår primära frågeställning är att utröna: 1)Vilka olika typer av heterogenitet kan man finna mellan olika system? 2)När är det lämpligt att applicera ett Data Warehouse koncept samt 3)hur korrelerar lämpligheten med formen på heterogeniteten? 4)Hur mycket heterogenitet (olikheter) kan man absorbera med hjälp av Data Warehouse konceptet?

Arbetet syftar även till att ge en magisterexamen i informatik.

## 1.4.Disposition

Nedan följer en redogörelse om vad som har behandlats i respektive kapitel i vårt akademiska arbete.

**Kapitel 1** – Här går vi genom bakgrunden, syftet, avgränsningen och problemställningen samt dispositionen för vårt arbete.

**Kapitel 2** – Här har vi redovisat vår arbetsmetodik, vårt synsätt samt hur vi har gått tillväga i arbetet.

**Kapitel 3** – I detta kapitel har vi redovisat de teoretiska föreställningar om heterogenitet. Vi har även beskrivet informationssystem och systemtänkande, samt analyserat de olika heterogenitets källorna och sist redovisat en analys av heterogenitet ur en informatisk synvinkel.

**Kapitel 4** – Här har vi redovisat Data Warehouse konceptet. Här beskrivs historiken bakom detta koncept och även dagsläget, konceptets tolkning, arkitektur och grundläggande principer. Vidare har vi redovisat problem som uppstår i samband med en migration till en Data Warehousemiljö.

**Kapitel 5** – Detta kapitel innehåller en kritisk analys av Data Warehouse arkitekturella principer såsom abstraktions-, integrations-, aggregations-, och evolutionsprincipen.

**Kapitel 6** – I detta kapitel har vi redovisat resultat av vårt arbete. För att tydliggöra heterogenitetsproblemet visar vi en sammanfattande bild av de olika typer av heterogenitet som vi kommit fram till samt över de personer och etablerade koncept som hanterar dessa olikheter och vilka de hanterar. Vi har även försökt att göra en sammanställande bild över de grundläggande heterogeniteterna, samt kombinerat ihop de olika typer av arkitekturansatser som finns för att lösa heterogenitetsproblemen.

**Kapitel 7** – Här har vi presenterat diskussionsdelen i vårt arbete. Vi har genomgående diskuterat Data Warehouse konceptets mål, lämplighet, problem och vad vi anser att det har saknats i den litteratur vi använt oss av. Vi har även diskuterat de olika heterogenitetstyper såsom den kulturella, den strukturella, den funktionella samt den kognitiva heterogeniteten. I denna del har vi även utvärderat vårt arbete samt gett förslag till vidare forskning inom ämnet.

**Kapitel 8** – Innehåller de slutsatser vi kommit fram till.

**Kapitel 9** – Innehåller litteraturförteckning över böcker, artiklar, etc.

## 2. Utredningsmetodik

### 2.1. Synsätt

I förra kapitel gick vi igenom bakgrunden till denna uppsats, samt syftet med uppsatsen och förklarade avgränsningen och problemställningen. Dessutom gick vi genom uppläggningsen av de olika kapitlen samt vad dessa innehåller. I detta kapitel kommer vi att belysa vår utredningsmetodik samt vilket synsätt vi har valt för vår utredning. Vi har valt en hermeneutisk tillvägagångssätt vilket kommer att förklaras nedan.

Hermeneutiken är ett grekiskt begrepp vilket kommer av Hermes, gudarnas budbärare som tydde gudarnas budskap för människorna, Runa Patel, Ulla Tebelius(1987). Begreppet "Hermeneutik" representerade under medeltiden en metod för bibeltolkning, utläggningar om bibeln som innehållande Guds mening. En hermeneutisk vetenskapsinriktning förutsätter att det finns ett avsikt i människors livsytringar. Med detta menas motiv som förverkligas i handlingar, språk och gester. Vi människor har levt i ett bestämt historiskt och nutida sammanhang som möjliggjort en uppsättning av symboler och begrepp vilka utgör den ram och den verklighet inom vilken vi tolkar och förstår saker och ting. Detta blir verklighet via vårt språk, vilket ger form åt våra tankar, ställningstaganden och värderingar.

Anledningen till att vi valt hermeneutiken som synsätt är att detta synsätt är det som passar oss bäst. Detta ger oss en möjlighet att använda oss av våra egna värderingar i de utvärderingar och analyser som vi gjort. Det innebär att de livserfarenheter och den kunskap som vi besitter, får lov att hjälpa oss att skapa en förståelse och genom detta skapas förutsättningarna för vår tolkning av det insamlade materialet. Vi har därför anpassat vår metodik och vårt arbetssätt utifrån hermeneutiken då detta verkade vara det logiskt mest riktiga.

### 2.2. Metodik och arbetssätt

Grundidéen till arbetet utformades under hösten 1998, i samband med detta skapades även den övergripande föreställningen om hur vi skulle lägga upp arbetet. För att kunna skapa oss en bild av området ifråga så var vi tvungna att genomföra en omfattande litteraturstudie. Då vi studerade Data Warehouse området, så gick vi tillväga på så vis att vi studerade tre författar konstellationers bilder av vad ett Data Warehouse är. Denna litteraturstudies syfte har sålunda varit att skapa oss en bild av DataWarehouse konceptets syfte och applicerbarhet.

Studien utökades sedan till att även innefatta litteratur om heterogenitet och koncept som hanterar denna heterogenitet. I denna del av studien var syftet att försöka fånga upp de tidigare föreställningarna om vad heterogenitet är, var den finns samt hur man skall hantera den.

Den tredje fasen i arbetet innebar att undersöka två system, detta för att påvisa några av de olika formerna av heterogenitet som vi har funnit. Vi valde bibliotekssystem på grund av att dessa var lätta att få tillgång till.

För att kunna påvisa skillnader i informationsbaserna så gick vi tillväga på så vis att vi valde ut ett antal böcker ur hyllorna på ett av de två biblioteken. Då detta var gjort så



sökte vi efter dessa böcker i de båda bibliotekssystemen. Därefter utvärderade vi resultatet av vår undersökning.

Nästa fas i arbetet var analysfasen där vi utvärderade Data Warehouse konceptet ur ett IT-management perspektiv. I denna analysfas skapade vi även en matris där vi förde in de olika heterogenitetstyperna.

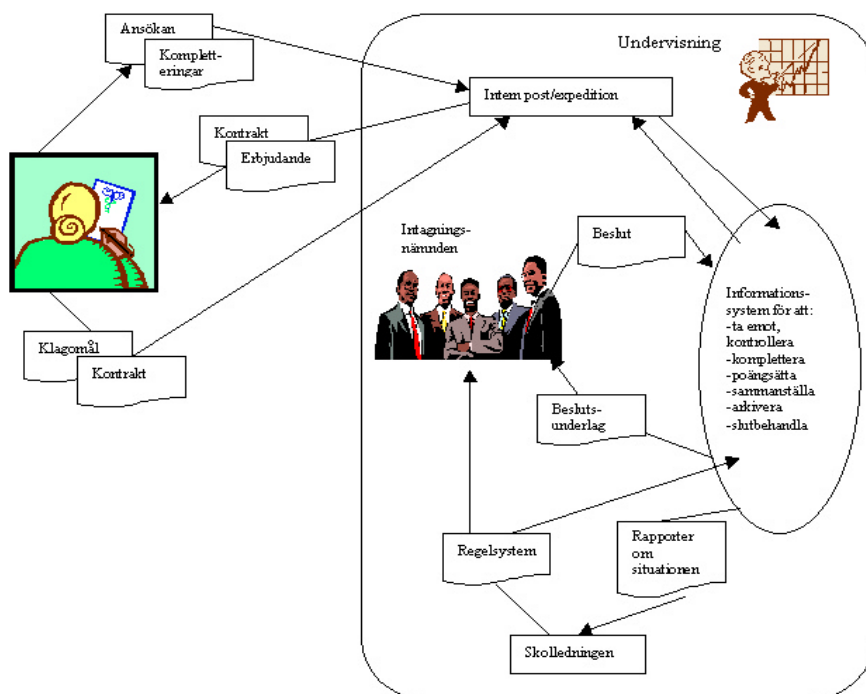
Då dessa faser av arbete var klara så var det dags att strukturera uppsatsen samt utvärdera respektive dels relevans för frågeställningen. Då analys- och strukturingsfasen var genomförda så sammanställdes resultatet, diskussionen och slutsatserna vilka fördes in i arbetet.

### 3. Heterogenitetens teoretiska förutsättningar

I kapitel 2 redogjorde vi för vårt arbetssätt och den metodik som vi använt oss av i detta arbete. Kommande kapitel förklarar hur ett informationssystem (IS) är uppbyggt samt de viktigaste egenskaperna som ett IS innehar. Kapitlet tar även upp de enligt oss fem grundläggande orsakerna bakom den existerande heterogeniteten mellan informationssystem. Vidare genomförs ett klagörande av olika grundläggande typer av heterogenitet.

#### 3.1. Systemtänkande och informationssystem

”Ett informationssystem är ett system för insamling, bearbetning, lagring, överföring och presentation av information.” (Erling S Andersen, 1994). Begreppet information och informationssystem verkar vara två av de mest spridda begreppen i vårt informationssamhälle. Det finns ett stort antal organisationer som satsat stora pengar på att utveckla olika informationssystem och andra kommer med en hög grad av sannolikhet att följa i deras fotspår. Med tanke på den spridning som information har fått i vårt informationssamhälle tycks det vara en nödvändighet med informationssystem. Konstruktion av ett informationssystem är ingen problemfri aktivitet därför dyker det kontinuerligt upp nya koncept, modeller och discipliner vilka skall visa oss vägen vid systemering och kontroll av information och genom detta även kontroll av informationssystem. För att förtydliga senare avsnitt kommer vi först att ge ett exempel på ett bekant informationssystem (Figur 3:1), nämligen antagningen av studenter till en högskolan vilken vi valt kalla högskolan X.



Figur 3:1 En skiss över antagningsarbetet vid högskolan X.

Antagningen av nya studenter är för denna högskolan precis som för andra högskolor en viktig uppgift. Skolan har en antagningsnämnd som består av representanter för

administrationen, lärare och studenter. Det är just denna nämnd som beslutar vem som skall erbjudas utbildningsplats. För att nämnden skall kunna fatta dessa beslut måste de ha tillgång till ett beslutsunderlag. Detta skall innehålla den nödvändiga informationen för att nämnden överhuvudtaget skall ha möjlighet att fatta goda beslut. Vi kommer nu att redogöra de centrala egenskaperna hos ett informationssystem. Dessa egenskaper är viktiga att känna till eftersom kunskapen om dem kan delvis avgöra vilken inställning man har till arbetet med informationssystem. Skapandet av ett informationssystem är en komplicerad uppgift och insikten om detta är viktig för rätt inställning till uppgiften.

Först och främst skall vi vara medvetna om att ett informationssystem är en mänsklig konstruktion. Därför är av naturliga skäl, uppläggningsen av informationsbehandlingen och förmedlingen av informationen från och till de sökande och från och till antagningsnämnden är utarbetat och konstruerat av människor. Den kan ha skapats mer eller mindre planmässigt och mer eller mindre bra, men det är människor som ansvarar för hur systemet fungerar, och dessa människor kan naturligtvis också ta initiativ till förändringen av det.

Sedan är det så att meningen med ett informationssystem är att det måste vara knutet till en viss arbetsuppgift, annars är det meningslöst att ha ett informationssystem. Vårt exempel behandlar det informationssystem för antagning av studenter vid högskola X. Som det märks går det inte att tala om ett generellt informationssystem. Ett informationssystem måste nämligen ses i förhållande till en viss uppgift i en bestämd verksamhet. I det här fallet är det intagningsnämndens arbete som är i behov av stöd.

Informationssystemets främsta uppgift är att förmedla information från vissa personer till andra personer. I exemplet förmedlas information från de sökande till antagningsnämnden och åter tillbaka till de sökande. Mellan dessa två parter finns vissa former av informationsbehandling. Detta mellanled kan "främmandegöra" eller till och med "förvanska" förhållandet mellan de två parterna. Dessa två ser inte varandra och därför uppfattar inte längre att man är två parter som ska kommunicera för att komma fram till ett resultat.

Som tidigare nämnts tar ett informationssystem emot information av olika slag. I exemplet mottas information om faktiska förhållanden. Antagningsnämnden tar också emot regler för antagningsarbetet. Även dessa regler är också ett annat slags information. De uttrycker inte något som ägt rum i verkligheten, utan visar vilka intentioner vissa personer har samt hur ärenden skall behandlas.

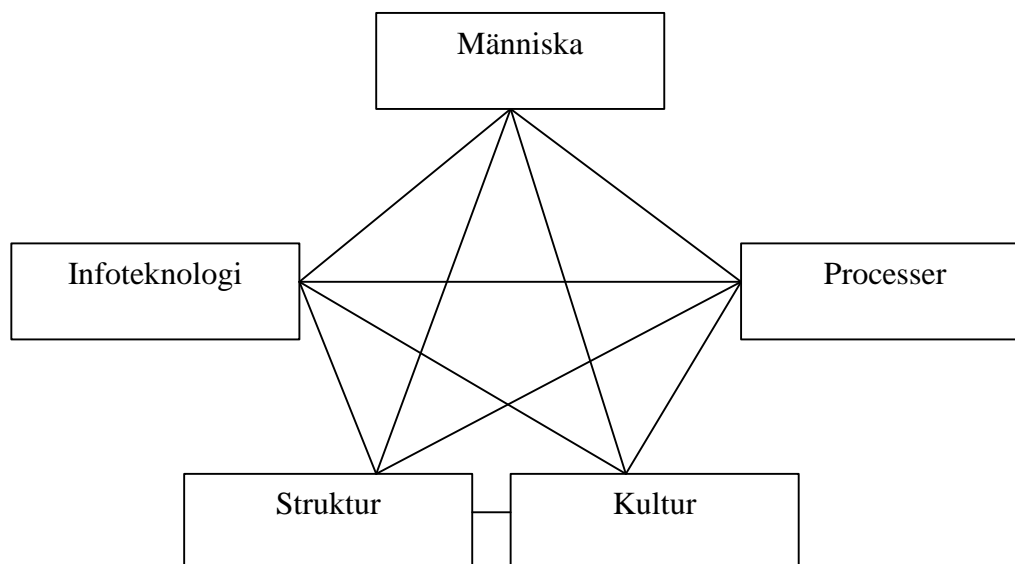
Enligt Erling S Andersen (1994) så brukar ett informationssystem i vanliga fall utföra olika typer av informationsbehandling. Exemplet ovan (Figur 2.1) visar att det sker en insamling av information, vissa bearbetningar i form av uträkning av poängsummor, lagring av information, överföring av information och presentation av information, både för de sökande och för antagningsnämnden

Informationsbehandlingen i ett informationssystem kan vara både manuell och maskinell. I exemplet gavs prov på informationsbehandlingen som kräver en bedömning, vilket förutsätter naturligtvis att uppgiften sköts av människor. Andra uppgifter kan utföras förstås utan människans bedömning, exempelvis rangordningen av de sökande utifrån poängsumman. Sådan behandling kan ske maskinellt.

Dessa var några egenskaper hos ett informationssystem vilka som tidigare nämnt är viktiga att känna till, eftersom kunskapen om dem kan delvis avgöra vilken inställning man har till arbetet med informationssystem då detta är en komplicerad uppgift.

### 3.2. Kartläggning av heterogenitetens källor

En organisations informationsmässiga heterogenitet kan sägas vara produkten av de ömsesidiga förhållanden som råder mellan den berörda organisationens beståndsdelar.



Figur 3:2 Jay R. Galbraiths modell(1977)

Vi använder och utgår ifrån Galbraiths modell(1977) för att sammanfatta heterogenitetens olika källor och typer ur ett organisatoriskt perspektiv.

#### 3.2.1. Den Kulturella heterogeniteten

Den kulturella heterogeniteten kan uttryckas i termer av olikheter i språket, perspektiv, verklighetsuppfattningar, värderingar m.m. Med andra ord allt det som skapar en kultur inom en organisation eller inom en del av en organisation. Inom informatikens sfär uppmärksammar B. Langefors och C. Argyris kulturens påverkan på informationsförsörjningen och den stora heterogenitet som detta kan resultera i.

Världsbanken utgör ett typiskt exempel på hur den kulturella heterogeniteten kan hanteras. Denna är en bank och som namnet antyder så har den sitt verkansområde i hela världen vilket i sin tur påverkar den kulturella heterogeniteten. Om en organisations kulturella olikheter är omfattande och är detta på grund av organisationens mission, så ökar sannolikheten för att de skall ha löst heterogenitetsproblemen utifrån de mjuka skolornas angreppssätt.

Bankens verksamhet omfattar ca: 6500 personer och var och en av dessa har sitt eget system. Ett team inom banken består av ca: 20 personer vilka delar på en och samma informationsbas. Kommunikationen mellan de i banken ingående teamen sköts i princip informellt men den stöds dock av olika informationssystem och informationsnätverk. Inom teamen så är mobiliteten stor men samordningen sker med stöd av den gemensamma

informationsbasen. Team medlemmarna har bara access till sin egen informationsbas. De andra teamens informationsbaser kommer de inte åt på grund av de stora kulturella skillnaderna och deras inverkan på tolkningen av den lagrade informationen.

### ***3.2.2. Den Strukturella heterogeniteten***

Den strukturella heterogeniteten har direkt påverkan på människans och organisationens handlingsfrihet. Den strukturella heterogeniteten kan främst illustreras med stöd av den "formella organisationen" det vill säga de ansvarsmässiga och befogenhetsmässiga beroenden som en organisation har gentemot samhället samt även den "formella organisationens" roll som homogenitets skapande verktyg.

Som exempel kan man ta den interna kontra den externa redovisningens roll. Den externa redovisningen måste följa samhällets regler och föreskrifter medan den interna är tämligen heterogen ur samhällets perspektiv.

Ett annat exempel på den strukturella heterogenitetens påverkan på handlingsfriheten är branschernas påverkan på organisationerna som ingår. Här kan man ta EDI och EDI/FACT som exempel. Detta kräver att man standardiserar vissa begrepp som förekommer i de ekonomiska och affärsmässiga transaktionerna. Just denna standardisering påverkar informationsförsörjningen mellan organisationer och minskar heterogeniteten.

I vissa fall är det så att myndigheterna går in och standardiserar olika uttryck, begrepp och koder för att på så sätt minska heterogeniteten och öka möjligheterna till informationsutbyte. Exempel på detta är personnummer, artikelnummer(EAN koder) och boknummer(ISBN). I vissa fall är det obligatoriskt för alla att följa standarden då denna upphöjts till lag medan i andra fall så är det frivilligt.

Vidare kan den strukturella heterogeniteten beskrivas i termer av standardiserade operativa processer eller olika typer av standardmallar som tillämpas av olika organisationer som ett försök att hantera den organisationella heterogeniteten i det interna kommunikationssystemet.

C. Hewitt (1986)och M. Å. Hugoson(1991, 1986) är två personer inom informatik som utgår ifrån den strukturella heterogeniteten. De förespråkar även någon form av språklig eller begreppsmässig homogenisering. En sak som är värd att notera är att varje försök att minska den strukturella heterogeniteten i informationsförsörjningen leder till begränsning av individens handlingsfrihet, Clas Olof Kall(1984).

### ***3.2.3. Den funktionella heterogeniteten***

Den funktionella heterogeniteten kan ses som resultatet av professionell hantering av olika ärenden och arbetsuppgifter. Ju mer differentierad en verksamhet är desto mer uppenbara blir de kunskaps och informationsmässiga olikheterna. En differentiering av en verksamhet kan ses som en designstrategi för att möta organisationens omgivning och dess dynamiska krav.

Enligt en dominerande designteori så kan en organisation endast möta omgivningens dynamik och variation med hjälp av egen variation. Med andra ord en organisations design kan ses som en avspiegling av omgivningens variation medan

informationsförsörjningens utformning kan ses som en avspegling av organisationens design, se M. Å. Hugoson(1986).

Inom informatik så finns t.ex. Anthony(1965), Zachman(1977) vilka har varit sysselsatta med heterogenitetsfrågor. Anthony menar att det är omöjligt och olämpligt att absorbera all heterogenitet samt att uniformera informationsförsörjningen. Omöjligheten relateras till kunskapens natur eftersom endast en del av kunskapen kan kvantifieras och kommuniceras genom data. Olämpligheten kan relateras till informationsförsörjningens sociala roll som motivationsfrämjande faktor. Det är inte all information som har denna funktion eftersom det är endast en del av informationen som flödar i organisationen som kan anges i klara och tydliga termer. Oklarheter och otydligheter skapar mer konflikter och misstolkningar än motivation.

En annan utgångspunkt är Zachmans teori att absorbera den funktionella heterogeniteten. Han menar att heterogenitet bromsar och hindrar såväl samordning som styrning av processer och därför bör absorberas på något sätt. Men alla instrument för att hantera och absorbera heterogeniteten ger inte samma resultat och är inte alltid lämpliga. Zachman menar att endast genom att betrakta informationsförsörjningen i resurs termer så kan man absorbera heterogeniteten och låta informationen flöda fritt inom organisationen.

Ett processbaserat perspektiv som förespråkas av bland andra Anthony kan alltså inte absorbera de informationsmässiga olikheterna på grund av de kunskapsmässiga och professionella olikheter som reflekteras i perspektiv, språk, värderingar och verklighetsuppfattningar. Enligt Anthony(1965) måste den information som homogeniseras för att senare kommuniceras vara i numeriska termer för att homogeniseringen skall fungera.

### ***3.2.4. Den kognitiva heterogeniteten***

Denna kan ses som ett fenomen av individens utveckling. Detta innefattar social utveckling, utbildning, erfarenheter och allt som påverkar oss som individer. Det är dessa olika delarna som gör varje individ till en unik informationsmottagare, informationsbehandlare, informationsskapare och beslutsfattare. March & Simon(1958) och Langefors(1978) förespråkar att beslut och handling är odelbara och här innefattar de även behandling av information. Samtidigt så förespråkar Simon en design som i princip neutraliserar behovet av global samordning och styrning av organisationen och därmed fullständig fränkoppling av informationsutbytet mellan olika informationsenheter. Enbart på detta sätt kan man bemöta heterogeniteten enligt Simon. Med andra ord, Simons strategi är att låta heterogeniteten leva sitt eget liv och att förändra det som går att förändra nämligen organisationens design<sup>1</sup>.

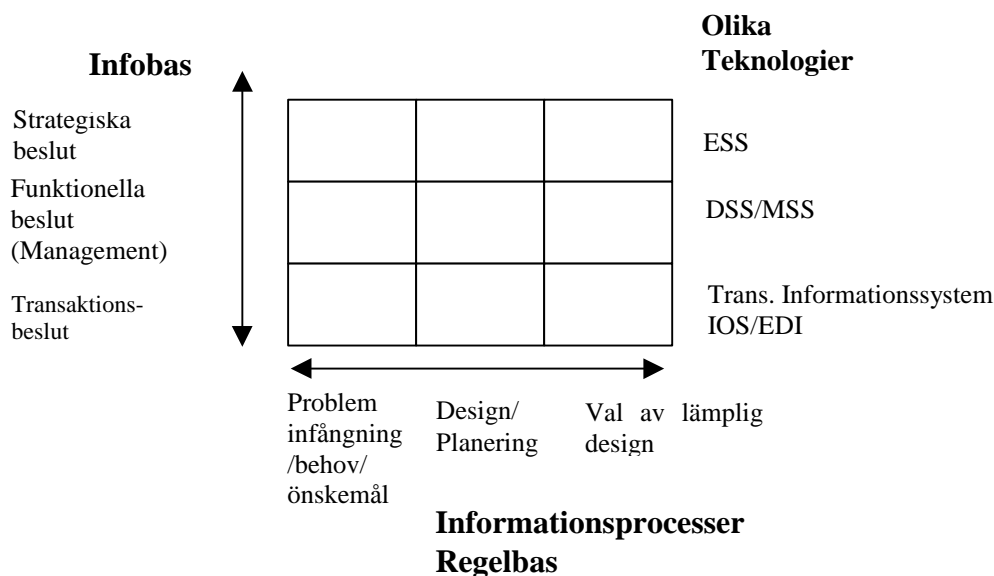
Notera att Simons skola förespråkar formalisering av den lokala informationsförsörjningen men detta har i princip inget att göra med heterogeniteten. Med andra ord det är en sak att standardisera och strukturera information och en helt annan sak att formalisera och reglera information.

Vi går vidare till den strukturella funktionella distansen inom organisationer(vertikal

---

<sup>1</sup> Simons teori, design teori kallas även informationsbehandlings paradigmet vilket innebär att hela organisationen betraktas som ett informationssystem bestående av relativt frikopplade enheter.

och horisontell distans mellan olika positioner inom organisationens struktur). Dessa olikheter kan exemplifieras i konceptuella olikheter i sättet att betrakta händelseutvecklingen och kommunicera information om denna utveckling. De som sitter i toppen av hierarkin bevakar olika trender och variationer för att i god tid upptäcka problem för att sedan åtgärda dessa. Detta innebär alltså att de är beroende av faktainformation från personer som sitter i botten av hierarkin.



Figur 3:3

De som sitter i hierarkins botten är i princip ointresserade av information om den historiska utvecklingen. Deras huvudsakliga intresseområde ligger på information om den framtida utvecklingen av kritiska situationer innan dessa inträffar och utvecklas till problematiska situationer. Detta gör att de blir beroende av målinformation från toppen. De personer som sitter på hierarkins mellannivå kan sägas fungera som filter av informationen som flödar mellan hierarkierna i organisationen. Då dessa personer saknar insikt i kraven på information som informationsmottagarna har så finns risken att de sorterar bort viktig information. Detta i sin tur kan skada informationens kvalitet och meningsfullhet för mottagaren. Just denna aspekt används av Zachman som vill frigöra informationen från mellannivåernas påverkan och göra den tillgänglig för alla berörda parter utan att fördärva informationens kommunikativa innehåll.

### 3.2.5. Den informationsteknologiska heterogeniteten

Information är kunskap som kommuniceras genom data därmed är det viktigt att se hur data skall organiseras och struktureras för att den överhuvudtaget skall bli kommunicerbar. Informatik som vetenskap om design, användning och management av informationsteknologier har alltid sysslat med forskning och utveckling av datamodeller för att just strukturera och organisera information. En datamodell kan ses som en designteori som vägleder oss i arbetet med att organisera information. Den består av ett fåtal begrepp, principer, tekniker och språk som möjliggör strukturering och systematisering av information utifrån ett socialt neutralt perspektiv. Aristoteles "Organon" kan betraktas som den första datamodellen. Den utvecklades bland annat för att harmonisera kommunikation och behandling av information med principer som idag kallas för objektorientering. Objektorienteringsparadigmet är i princip en

tillämpning av de filosofiska skolornas idéer om hur kunskap skall systematiseras och kommuniceras. Det uppenbara med denna Aristoteliska modellen är harmonin mellan verkligheten, tänkandet och språket. Det är därför inte konstigt att modellen vi presenterar i figur 3:4 återspeglar Aristotelismens tanke sätt. Här är objektsystemet lika med verkligheten, aktörssystemet lika med individens tankeverksamhet och informationssystemet lika med kunskap representerat genom språket.

Samma krav på harmoni mellan dessa tre system finns i Langefors(1975) begrepp ”infologisk validitet”(work ability) som i princip betyder att människans tanke sätt, händelseutvecklingen och informationssystemet är ömsesidigt beroende av varandra. Detta innebär i princip att heterogeniteten är uppenbar. En annan känd förespråkare är Harrington.(1991) Han menar på att kunskap i princip är icke kommunicerbart om man inte harmoniserar de semantiska(begreppsmässiga) olikheterna. Datamodeller, konceptuella modeller m.m. uttrycker i princip samma strävan, det vill säga harmonisering av våra begreppsvärldar. Men denna harmonisering kan inte vara slumpmässig om det skall bli effektivt. Det måste stödjas av någon av de etablerade datamodellerna såsom relationsteori, semantiska nätverk, entitets relationsansats, m.m. I mitten av 1980-talet noterade Bubenko(1981, 1984) att det fanns mer än 50 olika ansatser, vilka med stor sannolikhet är fler idag. Han menar att i en värld som översvämmas av information och som samtidigt är beroende av information, är det viktigt att klargöra hur heterogeniteten skall behandlas. Han klassificerar heterogeniteten i termer av:

- **Begreppsmässiga konflikter**- Synonymer och homonymer.
- **Strukturella konflikter**- T.ex. i ett system kan vissa data relaterar till ett objekt medan det i ett annat kan referera till en egenskap.
- **Sociala konflikter**- Dessa konflikter reflekterar hur människor uppfattar, presenterar och kommunicerar sin verklighet. Det är i princip fråga om regelmässiga olikheter och konflikter, såsom avbildningsegenskaper(1:M, M:M, 1:1). Dessa uttryck reflekterar vad som är omöjligt respektive möjligt eller vad som är tillåtet eller inte i den sociala verkligheten. Allt som är omöjligt eller otillåtet i den sociala verkligheten bör även vara det i informationssystemet. Som att ha dubbla födelsedatum eller liknande. Det kan också vara att ett kontos saldogränser, vilka är specificerade i regler.

### ***3.2.6. Filosofiska konflikter***

Tid, rum och sättet att se verkligheten(perspektivet) på är grundläggande källor till heterogenitet. Valet av datamodell innebär därmed också valet av en bestämd strategi för att hantera rummet och perspektivet. Men eftersom det saknas en universell modell för att absorbera begreppsmässiga, strukturella, sociala och filosofiska olikheter, är det viktigt att förstå att olika system som är systematiserade på olika sätt med hjälp av olika datamodeller kommer att vara kompatibla respektive inkompatibla. Inkompatibilitet är ett uttryck som representerar olikheterna i konceptualisering. Det vill säga två eller flera system kan inte kommunicera med varandra på grund av att deras konceptuella modeller är oförenliga. Oförenligheten är inte en fråga om semantik utan snarare en fråga om politik, eftersom det kräver förhandling mellan de olika intressegrupperna för att justera och balansera deras konceptuella bilder.

Utifrån detta perspektivet så kan man förstå ISO-modellen (ISO-82) som syftar till att



just etablera en grund för att diskutera, debattera och samordna heterogeniteten.

ISO-82 modellens uppkomst kan förklaras i termer av klargörande av hur hanteringen av heterogeniteten skall hanteras. Detta efter ett i princip totalt misslyckande av ANSI/SPARC-modellen(ANSI/SPARC-75). ANSI/SPARC-modellens strategi var att harmonisera definitioner och uppfattningar av data i databaser utifrån ett såväl socialt som teknologiskt perspektiv. Med något som man kallade externa scheman ville man markera de sociala olikheterna i definitionen av information, medan man med begreppet interna scheman ville markera de tekniska olikheter som förekommer i hantering och presentation av information. Dessa olikheter, tekniska och sociala, kunde absorberas med införandet av en konceptuell beskrivning av information utifrån hela verksamhetens perspektiv.

ISO-modellens kritik kan sammanfattas i ett fåtal enkla ord. Om vi överhuvudtaget skall prata om gemensamma data och databaser, så skall vi etablera en gemensam verklighetsuppfattning. Denna kan i princip betraktas som ett socialt kontrakt detta eftersom den etableras och modifieras genom förhandling mellan olika parter.

Slutligen så kan vi referera till det sista försöket att hantera heterogeniteten. Detta är UML (Unified Modeling Language). Försöket är enligt vår uppfattning lovvärt men vi är dock tveksamma till språkets inverkan på det olösta heterogenitetsproblemet. Vi låter tiden avgöra hur varaktig denna trend kommer att bli.

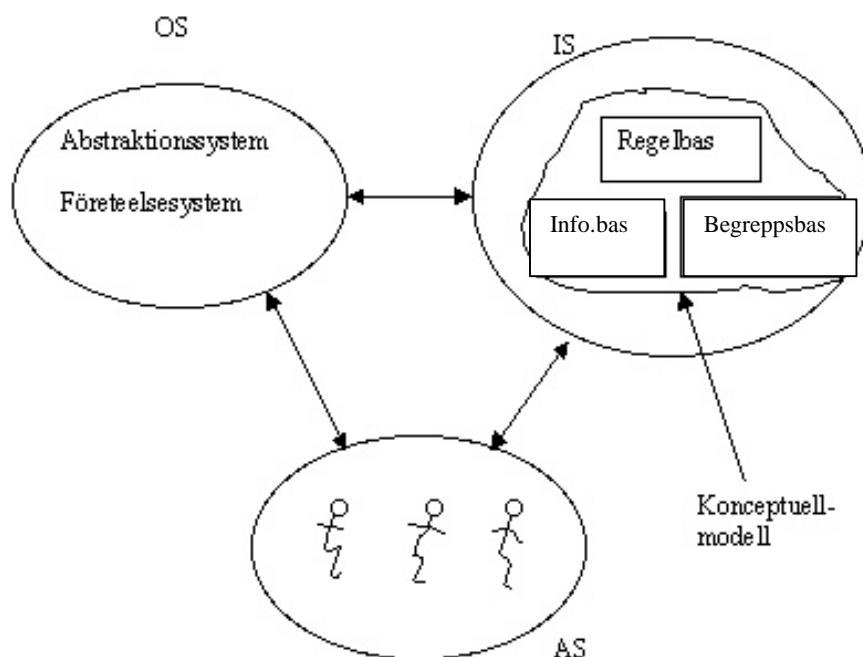
Ett sista påpekande, man kan absorbera en hel del heterogenitet om man tar bort tiden och i princip endast presenterar en momentan bild av verkligheten. Detta kan vara en bra strategi i vissa fall men olämplig i fall där man har administrativa system som kräver historisk information. Utifrån detta så kan man förstå skillnaden mellan relationsmodellen och Bubenkos(1977) CIAM(Conceptual Information Analysis Methodology). CIAM-modellen är fullständigt temporär för att kunna hantera uppgifter om händelseutveckling, nutid och framtid. Relationsmodellen definierar däremot informationen i icke temporära termer.

### 3.3. Ett klargörande av heterogenitet ur informatisk synvinkel

Utöver den funktionella föreställningen om vad ett informationssystem är så finns det även en semantisk bild av vad ett informationssystem är.

Ett verksamhetssystem består av tre grundläggande delar(ISO-82).

- 1) Aktörsystem- dvs de människor som påverkar och påverkas av systemet.
- 2) Objektsystem –dvs den delen av verkligheten som är intressant utifrån aktörernas intresse och informationsbehov.
- 3) Informationssystem- har som främsta uppgift att besvara frågor om objektens tillstånd och genom detta avbilda objektsystemet i termer av konceptuell modell.



Figur 3:4 ISO-modellen

Aktörsystemet – kan delas upp i informationsanvändare, informationsavlämnare och systemägare. Informationsanvändare är den som använder information i beslut och handling (med andra ord inte en sekreterare eller dylikt). Informationsavlämnare är den som är skyldig att avlämna uppgifter om händelser i objektsystemet. Systemägarens uppgift är att specificera informationssystemet i form av en begreppsbas och en regelbas som skall styra informationssystemets interna verksamhet.

Objektsystemet består av två delar:

Företeelsesystem som omfattar alla företeelser, objekt, processer och händelser m.m. som finns representerade i informationssystemet. Den andra delen som kallas abstraktionssystem, utgör en systematisering av objektsystemet i termer av förhållanden mellan objektklasser och attribut samt förhållanden mellan objektklasser och händelser. Varje objektklass som skapas genom abstraktion utgör grunden för att

gruppera den mångfald, heterogenitet och dynamik som kännetecknar företeelsesystemet.

Informationssystemets verksamhet utförs av informationsprocesser och dess uppgift är att ständigt kommunicera med aktörsystemet eller andra informationssystem. Informationsprocessernas huvudverksamhet består av att ta emot meddelande om händelse utveckling, granska korrektheten (validering) och slutligen integrera dessa uppgifter till de redan existerande på ett sätt så att man undviker redundans, motsägelser och andra konflikter.

Den konceptuella modellen som förvaltas, uppdateras ständigt av informationsprocesserna. Den konceptuella modellen utgör en konsistent avbildning av objektsystemet. Den konceptuella modellen består av en regelbas, en informationsbas och en begreppsbas. Regelbasens innehåll kan i princip beskrivas i termer av valideringsregler, integrationsregler (evolutionsregler, synonymer) och härledningsregler. Härledningsregler kan vara enkla såsom olika slags summerade uppgifter eller mycket avancerade såsom olika beräkningar som är typiska för beslutsstödjande system. Informationsbasen representerar i princip företeelsesystemet medan regelbasen utgör en avbildning av abstraktionssystemet. Slutligen så utgör begreppsbasen den gemensamma grunden vilken refererar till såväl abstraktionssystemet som företeelsesystemet, i kommunikation mellan människor, mellan informationssystem, mellan människor och informationssystem.

Med andra ord, så måste både innehållet i informationsbasen och regelbasen beskrivas i termer av begreppsbasen.

### ***3.3.1. Beskrivning av heterogenitet ur informatisk synvinkel***

I dagens organisationer så har utvecklingen av informationssystem inte skötts helt konsekvent. Detta har lett till att man har skapat informationskällor inom organisationen som sinsemellan är tämligen olika, vi kallar dessa olikheter för heterogenitet. Som vi ser det så finns det flera olika typer av heterogenitet som man kan finna mellan olika system. Dessa kan vara följande:

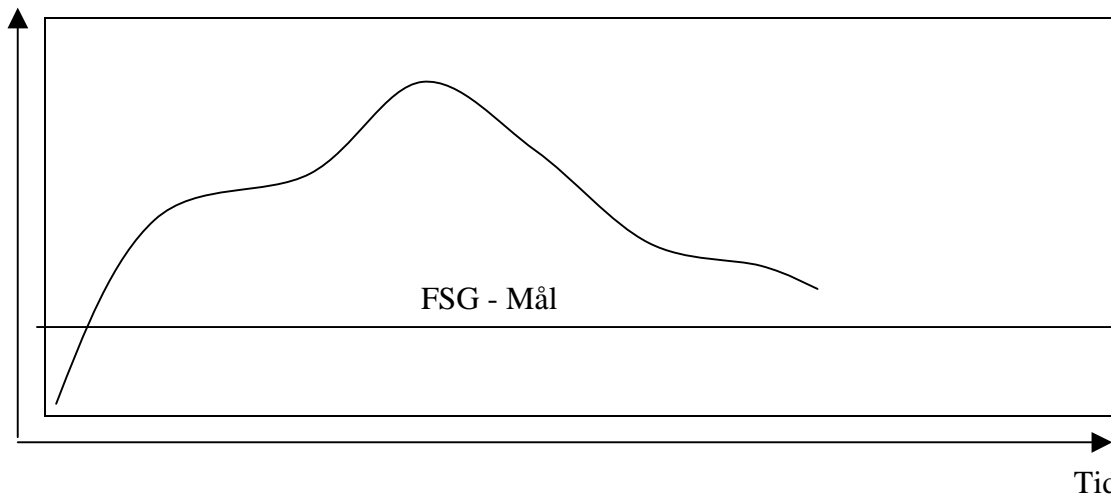
- Regelbaserad heterogenitet
- Innehållsmässig heterogenitet
- Begreppsmässig heterogenitet

Om man tittar på användarsystemen, användarnas referensramar och mål så kan man finna skillnader mellan dessa system. Skillnaderna i referensramar, mål och syften kan medföra att man kommer i konflikt med varandra och den grundläggande orsaken till detta är att man tolkar data på olika sätt.

Om man tittar på exemplet nedan så ser man en kurva vad den betyder varierar beroende på vilket användarsystem man tillhör. Om kurvan visar antal sålda artiklar och den horisontella linjen motsvarar försäljningsmålet så skulle en försäljare tolka detta som att han har nått upp till målet som är uppsatt för honom. Försäljningschefen skulle visserligen även han tolka det som att försäljaren nått upp till målen men han skulle även lägga in en ytterligare tolkning som innebär att kurvan visar försäljningstrenden och hur den har utvecklats. Trenden är vikande och chefen vill

troligtvis få en förklaring till varför. Försäljaren å sin sida anser att han nått upp till målen och förstår kanske inte chefens oro.

Omsättning



Figur 3:5

### 3.3.2. Beskrivning av begreppsmässig heterogenitet

I detta avsnitt kommer vi att gå igenom den begreppsmässiga heterogeniteten. Vi kommer att visa vad vi menar med begreppsmässig heterogenitet samt kommer att visa hur problemen respektive regler yttrar sig.

Den begreppsmässiga heterogeniteten kan också kallas för den formmässiga heterogeniteten. För att klargöra vad vi menar med begreppsmässig heterogenitet skall vi försöka med en exemplifiering av denna.

Om man i en databas har data lagrad om vilka tåg som går mellan två olika orter samt dessa tågens tidtabeller, så kommer ett tåg som avgår vid en given tidpunkt att betecknas på ett specifikt sätt. Detta kan vara en kombination av veckodag, avgångstiden och tågets slutdestination. I en betraktares ögon så kommer t.ex. tisdagståget 14.35 mot Stockholm att vara samma tåg varje tisdag och det kommer med stor sannolikhet att representeras på samma sätt i databasen varje tisdag. Det finns dock inga garantier för att detta tåg är samma tåg rent fysiskt sätt. Med andra ord så finns det inget som garanterar att ett konceptuellt tåg representeras av ett och endast ett fysiskt tåg. Denna osäkerhet och skillnad i vad man menar med tåg representerar en form av heterogenitet som kan leda till problem.

I figur 3:6 så visas de olika stadierna i designen av olika system. Bilden visar även vad som utförs i respektive del när man skapar de olika design produkterna. Bilden visar även vad för typ av heterogenitet som kan uppkomma vid skapandet av respektive produkt. Bilden visar även den komplexitet som finns bakom de olika heterogeniteterna och följaktligen de svårigheter som finns då man skall komma tillrätta med dessa.

Design Design	Design produkt	Begrepp	Information	Regler
Konceptuell design		Etablering av begreppsbas i objekttermer	Beskrivning av händelser i objekttermer	Etablering av regler
Logisk design		Klargörande av egenskaps begrepp & nyckel begrepp	Klargörande av härledd information & Fakta information	Etablering av härledningsregler Integrationsregler Valideringsregler
Fysisk design		Klargörande av interface	Klargörande av lagring	Klargörande av olika slags behandlingar
Heterogenitet i distribuerade miljöer/ mellan flera system		Begreppsmässig heterogenitet	Innehållsmässig heterogenitet	Regelmässig heterogenitet

Figur 3:6

### Exempel på begreppsmässig heterogenitet

Antag att man har två olika system där man för in data om studenter på en skola. Antag vidare att dessa har skillnader i begrepp och att de är föremål för integration, då kan det se ut enligt nedan. De problem som finns här är att det saknas gemensamma begrepp och attribut. Lösningen ser i detta fall ut enligt tabell "efter integrering" i figur 3:7. Man har här skapat en ny tabell där man enats om vilka egenskaper man skall ha på objektet "Student".

Före integrering

Student

SNR	Namn	Adr	Tel

Elev

PNR	Enamn	Fnamn	F.dat

Efter integrering

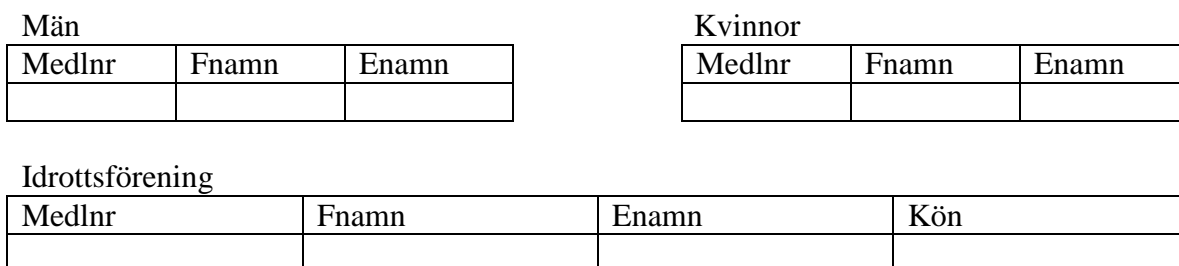
Student

P.NR	Enamn	Fnamn	Adr	Tel	F.Dat

Figur 3:7

För att ytterligare påvisa integrationsproblemen mellan olika system så visar vi ett exempel där en idrottsförening för män och en för kvinnor vill gå ihop och ha ett gemensamt system. Utformningen på de båda systemen är identiska med varandra.

Problemet här är att innehållet inte är jämförbart och att om inte tillägg görs så kan det bli problem med att härleda fram viss information. Den information som inte går att härleda ur informationsbaserna i detta fall är individernas kön. Detta löses här genom att man kommer överens om att införa en ny egenskap som i detta fall är kön. Anledningen till att denna egenskap inte funnits innan är att den inte har varit nödvändig då alla tidigare var av samma kön i systemet. Liknande problem kan uppkomma i större och mer komplexa system vid integrationer.

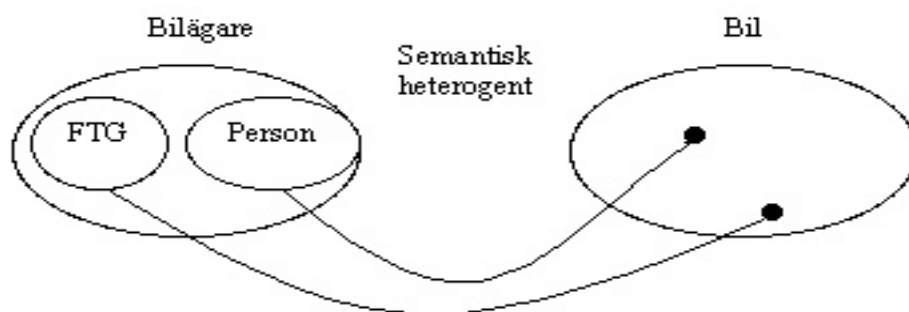


Figur 3:8 Medlemssystem för två idrottsföreningar.

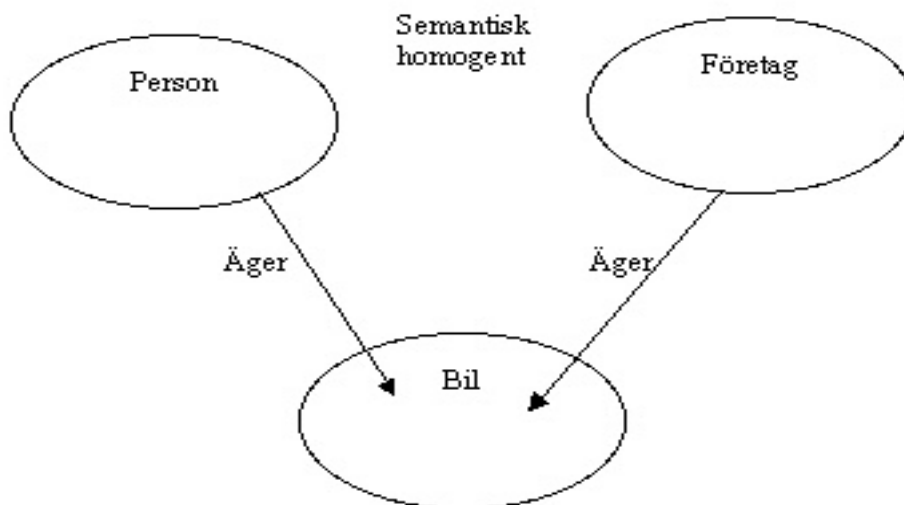
### Semantisk homogenitet

I olika system så kan problem uppstå genom att man förlorar den semantiska homogeniteten. Det som vi menar med detta är att det inom ett system finns två eller fler objekt som slås ihop utan att dessa är av samma typ eller har samma egenskaper. Problemet kan exemplifieras med följande exempel där vi har tagit ett bilregister som slagit ihop personer och företag i en gemensam klass "bilägare".

Från denna klass finns sedan kopplingar till olika instanser av objektet "bil". Problemet är här att person och företag inte har samma egenskaper. En åtgärd som bör göras, är att separera dessa. M.Bunge (1967)se nedan.



Figur 3:9a



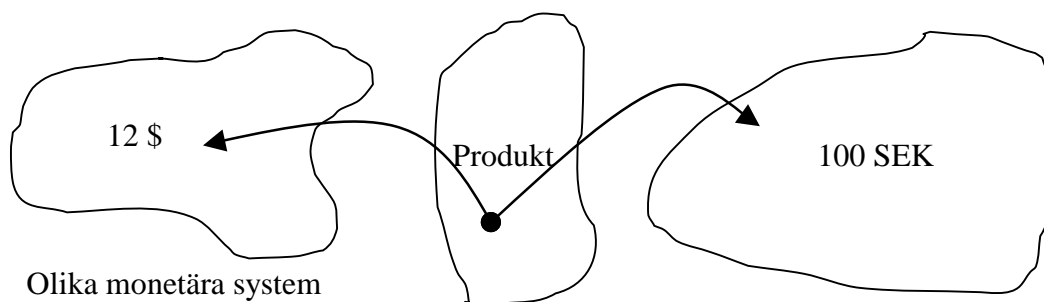
Figur 3:9b

Här är alltså en lösning att dela upp "Bilägare" till två klasser "person" och "företag". Då detta är gjort så tillförs en relation mellan de personer som äger en bil och de bilar som dessa äger. Likadant görs för klassen företag.

### Semantisk slutenhet(Semantic closure)

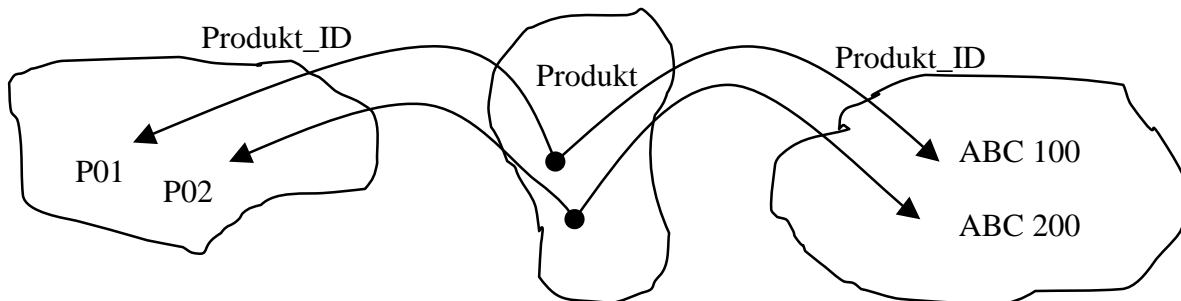
Semantisk slutenhet har den betydelsen att de attribut som tillhör den konceptuella modellen är de enda som förekommer i modellens principer och definitioner. Alla andra som utökar modellen måste vara härledbara ur grundsatserna eller så måste de nya omformas på något sätt så att dessa blir kompatibla med grundsatserna. Med andra ord meningen med semantisk slutenhet är att klargöra olikheterna mellan attributvärden. Krav på slutenhet siktar till att hejda en okontrollerad utvidgning av den konceptuella modellen efter att denna har konstruerats.

Vi ska försöka illustrera detta med hjälp av exemplet (figur 3:10). Man kan se att det förekommer heterogenitet i hur man har definierat valuta i de två olika systemen. Här nedan är det två olika monetära system som visar attributet pris för samma produkt på två olika sätt.



Figur 3:10

I exemplet nedan(figur 3:11) visas två olika referenssystem. Det ena kan exempelvis vara ett leverantörssystem. Här används produkt\_ID P01 för en produkt medan den i det andra systemet, som kan vara ett återförsäljarsystem, benämns som ABC100. Den semantiska heterogeniteten som förekommer mellan olika system är ett av de möjliga problemen som kan finnas när man försöker att integrera flera olikartade databaser.



Figur 3:11 Olika referens system

### Sammanfattning av begreppsmässig design

Det är av stor vikt att den begreppsmässiga designen genomförs riktigt vid en integration eller konstruktion av system. Detta för att möjliggöra användandet av gemensamma referenser eller nycklar om man så vill. Om designen utförs konsekvent och riktigt så kommer det att vara möjligt att undvika den semantiska heterogeniteten. En riktig design är även ett krav för att det skall vara möjligt att uppnå semantisk slutenhet. En korrekt och grundligt utförd design är också en nödvändighet för att man skall kunna undvika onödig redundans.

### 3.3.3. Beskrivning av innehållsmässig heterogenitet

I detta avsnitt kommer vi att gå igenom den innehållsmässiga heterogeniteten. Vi kommer att visa vad vi menar med innehållsmässig heterogenitet samt kommer att visa hur problemen respektive regler yttrar sig.

Innehållsmässig heterogenitet är när man har lagrat samma data fast på två eller flera olika sätt eller med olika värden på samma begrepp och samma objekt. Skillnaderna kan man finna på flera olika ställen såsom:

- ID-begrepp
- Tids presentation
- Attribut utformning
- Pris

Heterogenitet bland ID-begrepp kan yttra sig så att en individ har olika ID-begrepp i olika databaser. I vissa kan denne ha sitt personnummer och i andra kan det vara telefonnumret som används som unika identifierare(figur 3:12). Här har vi två tabeller som endast innehåller kundens ID som används i två tänkta operationella system. Skillnaden är total och det finns ingen gemensam nämnare.

Kund_ID
700731-XXXX

Kund_ID
0300-12345

Figur 3:12



Det andra området där man kan finna innehållsmässig heterogenitet är i fallet med attribututformning och sammansatta attribut. I det ena fallet så skriver man t.ex. förnamn och efternamn i samma fält medan man i det andra separerar dem i två olika fält.

Namn	
Sven Svensson	
F_Namn	E_Namn
Sven	Svensson

Figur 3:13

Problemet i detta fall är alltså att man inte kan finna någon data som är identisk mellan dessa fält. Detta måste lösas för att man skall kunna få systemen att samverka.

Det är alltså enligt oss viktigt att man har samma form på attributen. Det är även viktigt att man är konsekvent när det gäller övrig utformning av attribut här menar vi då valet av datatyper.

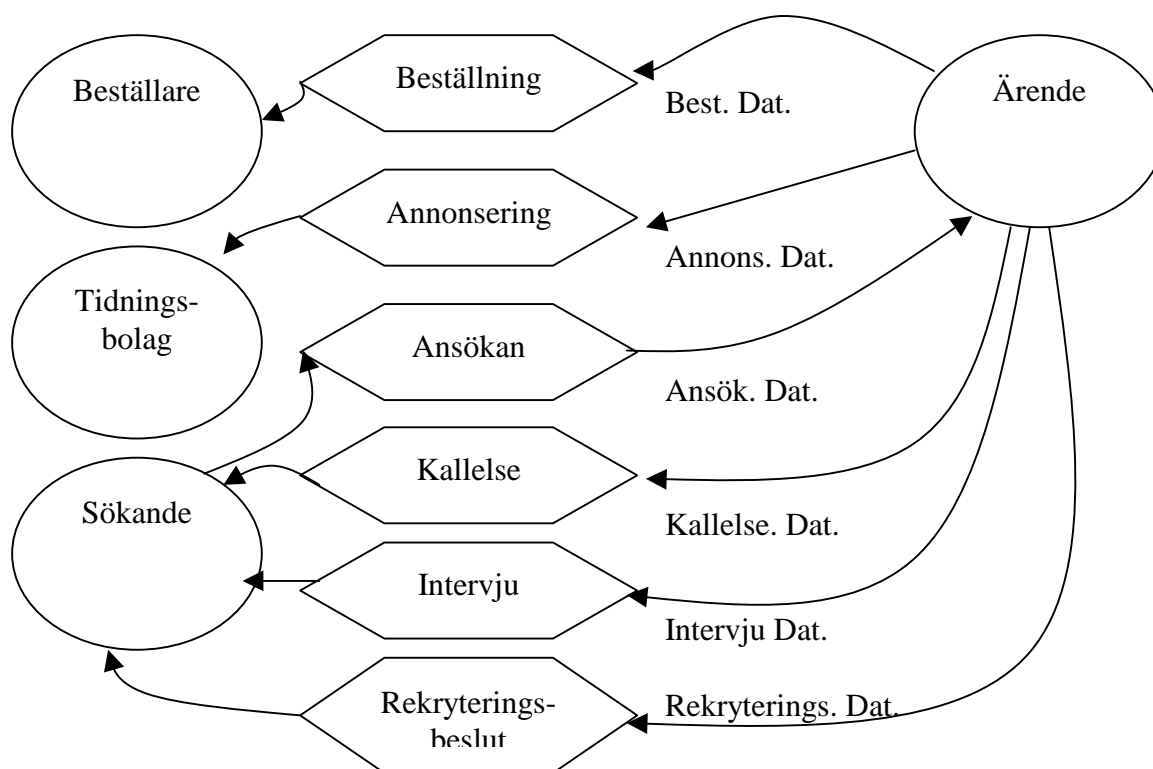
Det tredje området som vi nämnt är fallet med pris. Problemet här är att man kan ha priset i olika valutor vilket gör att det kan bli problem med direkta jämförelser av priser(jmf. semantisk heterogenitet). Om priset anges i dollar i den ena tabellen och i svenska kronor i den andra så är inte dessa direkt jämförbara, utan då måste det ena värdet konverteras till det andra för att en jämförelse skall bli möjlig. Problem kan även uppstå då värdet på två eller fler objekt vid olika tidpunkter skall fastställas detta beroende på den eventuella osäkerheten om hur valutorna har utvecklats under tiden som gått.

Pris	Pris
800	200

Figur 3:14

### Tidsmässig heterogenitet

För att kunna förklara vad vi menar med tidsmässig heterogenitet så har vi valt att göra ett exempel som visar ett ärendes olika tidsformer. Ärendet är ett tänkt rekryteringsärende. De inblandade objekten är: beställare, ärende, tidningsbolag och sökande. Händelserna är: beställning, annonsering, ansökan, kallelse, intervju och rekryteringsbeslut. Under detta ärendets livstid så finns det inte mindre än sex olika tidsangivelser. Detta kan i sin tur leda till problem om man har flera system med liknande funktioner som vart och ett har olika angivelser för tiden.



Figur 3:15

Ett ytterligare exempel på tidsmässig heterogenitet är om man inom ett företag har tre olika försäljningskontor vilka alla har olika system för hur man räknar ut aktuell försäljningssiffra. På avdelning A så har aktuell försäljningssiffra definierats som senaste vardagens totala försäljning. På avdelning B har aktuell försäljningssiffra definierats som senaste veckans totala försäljning. Avdelning C har i stället den senaste månadens totala försäljning som beräkningsgrund. Om ledningen gör en förfrågan om vem som har de mest aktuella försäljningssiffrorna så kommer alla att anse att de har dessa. Alla avdelningar har här olika tidsuppfattningar vilket i sin tur kan ge upphov till att det blir skillnad på tidsbegrepp som i detta fall "Aktuell försäljning".

### 3.3.4. Beskrivning av regelbaserad heterogenitet

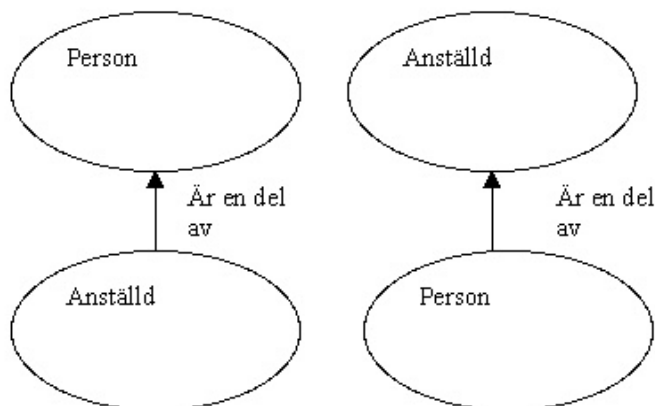
I detta avsnitt kommer vi att gå igenom den regelbaserade heterogeniteten. Vi kommer att visa vad vi menar med regelbaserad heterogenitet samt kommer att visa hur problemen respektive regler yttrar sig.

Den regelbaserade heterogeniteten liksom övriga typer av heterogenitet, kan ta sig lite olika uttryck. Som vi ser det så finns det fem olika typer av regelbaserad heterogenitet, dessa kan illustreras som följande:

- Klassificeringsregler
- Kategoriseringsregler
- Avbildningsregler
- Valideringsregler
- Härledningsregler

### Klassificeringsregler

Klassificeringsregler är regler som talar om hur klassificeringen av olika typer av objekt skall gå till. Man kan i ett fall se en person som en sort av ett objekt anställd eller så ser man anställd som en sort av objektet person.



Figur 3:16

### Kategoriseringsregler

När heterogenitet mellan kategoriseringsregler undersöks så kan ett bibliotek tjäna bra som exempel. Här finns samma klassificering av böcker över hela landet samtidigt som det lokala biblioteket bestämmer själv vilken kategori som en viss bok skall tillhöra. Om de i ett bibliotek bestämmer sig för att en bok som handlar om odling av kryddor tillhör böcker om jordbruk eller hobbyodling spelar mindre roll för deras egen skull. Men om de vid ett senare tillfälle skulle vilja samverka mer intimt med andra system så kan man stöta på problem på grund av detta.

### Avbildningsregler

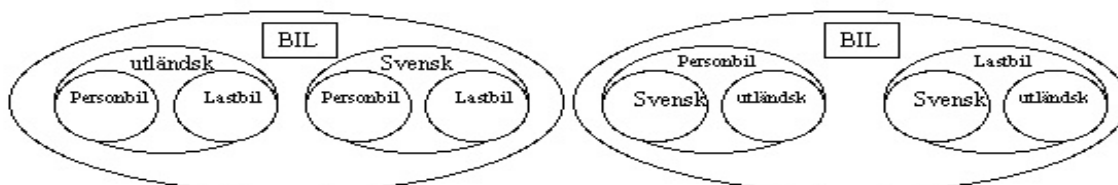
Skillnader i avbildningsregler kan yttra sig på så vis att man får konflikter mellan de olika objektens förhållanden av formen ett till många eller många till ett. I ett tänkt fall så kan det vara så att en bok kan ha en och endast en författare i ett system medan böcker i ett annat kan ha flera författare. Här har man i så fall en konflikt i avbildningsreglerna systemen emellan.

### Valideringsregler

När man tittar på valideringsreglerna så kan man säga att det som är tillåtet i verkligheten även skall vara tillåtet i systemet. Det är t.ex. inte tillåtet att ta sig flera fruar i Sverige men detsamma gäller inte t.ex. i Saudiarabien. Nu behöver inte reglerna som reglerar vad som är tillåtet eller ej vara regler av nationell karaktär utan kan likväl vara lokala regler. Dock är det så att oavsett om reglerna är globala eller lokala så kan man finna regelkonflikter då man skall samverka med system som finns utanför det område där ens egna regler finns och gäller.

### Härledningsregler

Då man tittar på fallet med heterogena härledningsregler så stöter man på polymorfism. Man stöter alltså på objekt som är identiska men som härleds fram på olika sätt enligt olika modeller. T.ex. en utländsk lastbil kan härledas enligt exemplet nedan.



Figur 3:17

De ovannämnda olika typer av heterogenitet, är inte alla komplicerade att åtgärda. Det är dock nödvändigt att dessa synliggörs för att man inte skall stöta på svårigheter längre fram. Lösningen på många av dessa olika typer av heterogenitet är oftast politisk. Man kan alltså lösa det mesta genom förhandlingar mellan alla inblandade parter, härmed är det inte sagt att lösningen alltid är enkel.

### 3.3.5. Beskrivning av andra former av heterogenitet

Utöver de ovannämnda heterogeniteterna i föregående stycken finns det som vi ser det ytterligare former av heterogenitet. Dessa kommer att redovisas utförligt nedan.

#### Stilistisk heterogenitet

Man kan även finna skillnader i hur man tolkar uppenbart lika data. Här kan två tillsynes lika datum representera olika dagar beroende på vilket datumformat som man har valt. I exemplet nedan så har vi två tillsynes helt identiska datum, dock är de skapade efter olika tidsformat nämligen ett europeiskt och ett icke europeiskt. Detta gör att den ena representerar den 6:e mars och den andra den 3:e juni. Det som är orsaken till problemet är i detta fall att man har positionsbestämda värden och att dessa värden inte stämmer överens med varandra.

Dat	Dat
060399	060399

Figur 3:18 tabellexempel över den stilistiska heterogeniteten

#### Förkortningar

Ett område som man kan finna problem med är användandet av förkortningar. Olika organisationer eller olika delar av organisationer kan använda olika typer av förkortningar för samma saker. Dessa skillnader måste övervinnas om det skall vara möjligt att integrera olika system med varandra.

## 4. Data Warehouse konceptet

I detta avsnitt kommer vi att redovisa (1) Data Warehouse konceptets uppkomst och utveckling. (2) en typisk arkitektur av Data Warehouse. (3) Data Warehouse konceptets tolkning av heterogenitetsproblemet. (4) Data Warehouse konceptets lösning till heterogenitetsproblemet. (5) Data Warehouse grundläggande för byggande av Data Warehouse arkitektur, samt (6) De tekniska insatserna som krävs för att migrera till en s.k. Data Warehouse miljö.

### 4.1. Historik och dagsläge

På 1970-talet så kom lagrings media som möjliggjorde direktaccess till filer(DASD). Innan hade det endast funnits sekventiell åtkomst till filer på lagringsmedia såsom tejp av olika slag. Den nya tekniken gjorde det möjligt att komma åt efterfrågad data mycket snabbare då man inte behövde läsa sig igenom all data som låg lagrad innan på tejp. I och med den nya tekniken så kom även en ny mjukvara, databashanteringssystem(DBMS). Syftet med dessa program var att underlätta hantering och lagring av data i direktaccessminnena. Databashanteringssystemen och direktaccessmedierna var den tekniska lösningen på de problem som var följden av den tidigare tejp lagringen. Dessa problem bestod i bland annat, redundant data, felaktig data, och långa accesstider.

Under slutet av 1970-talet så började man med online transaktioner i databaserna. Detta tillsammans med att man hade en snabbare access till befintlig data ledde till att man nu började att använda datorerna på helt nya sätt såsom t.ex. bokningssystem, banktransaktionssystem och produktionskontrollsystem. Det var alltså utvecklingen av de nya lagringsmedierna som ledde in datorutvecklingen på den vägen som vi ser och tar för givet idag.

Under 1980-talet så kom insikten om att datorer kunde användas till mer än att bara lagra data och direktmanipulera data med. Man började att se nya möjligheter att använda data på. Det var nu som MIS(Management information systems) dök upp. Dessa system möjliggjorde användningen av datorer till mer än funktionsgörsmål. Nu kunde man även använda datorer till att underlätta ledningsbeslut samt att man kunde använda samma databas till både transaktioner och till MIS.

Man började också att använda så kallade extrakt program<sup>2</sup>. Det dröjde inte länge förrän dessa program blev populära, detta på grund av att de medförde att man inte sänkte prestandan på databasen i samma utsträckning som skulle varit fallet om man skulle ha analyserat data i den ursprungliga databasen. Nu flyttade man alltså de krävande analyserna till en extern databas och minskade på så vis belastningen på ursprungs databasen. En annan orsak kan ha varit att då man flyttade data ut ifrån operationella delen av organisationen så skedde även ett skifte i ägandet av data. Nu var det användarna av MIS, eller som det kallas idag DSS(beslutstödssystem), som ägde den data som de hade extraherat ur databasen.

Under slutet av 1980-talet och under 1990-talet så hade utvecklingen fortsatt på så vis att nu fanns det program som använde data som var extraherad ur data som var

---

<sup>2</sup> Ett extrakt program fungerade på så vis att med hjälp av utsöknings kriterier så sökte man reda på intressant data som man sedan flyttade till en extern databas.

extraherad ur data också vidare i flera steg. Datorarkitekturen började att utvecklas nästan självständigt och snart liknade den en spindelväv eller en spaghettstruktur, Magoulas, Pessi (1998). Denna utveckling av arkitekturen medförde flera problem då bland andra en känslighet för förändringar. Det kunde bli så att om man ändrade i en databas(DB) så fick det följderna att flera andra DB slutade att fungera eller åtminstone började att fungera sämre. Ett annat problem som kunde och kan uppstå är att pålitligheten och kvalitén på data blir lidande. Detta kommer sig av att beroende på när och varifrån data extraheras så kan den ta sig olika värden. Andra felorsaker kan vara skillnader i primär extern data, på sökningsalgoritmerna eller på saker som hur många nivåer som data är extraherad innan den slutligen når sin destination. Dessa inneboende skillnader kan komma sig av att datasystemen är utvecklade av olika personer vid olika tidpunkter med olika kompetenser och i olika syften.

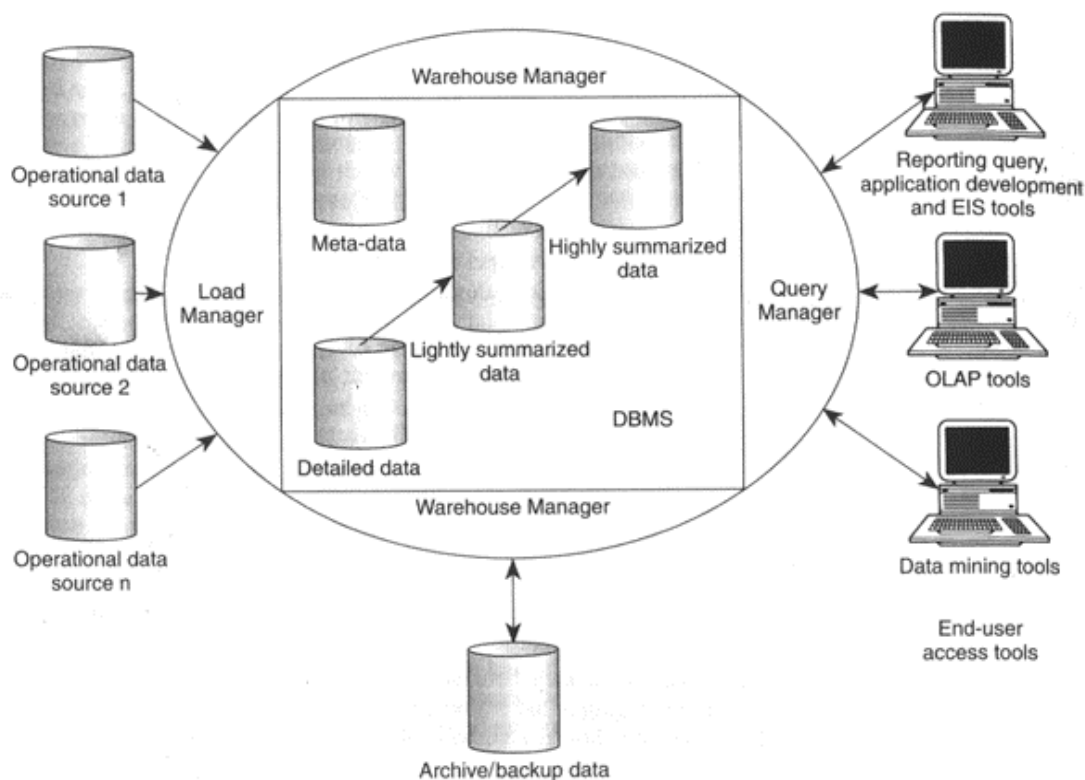
I dagens organisationer är det inte ovanligt att man har tillgång till mycket stora mängder data samtidigt som man lider brist på riktig information. Orsaken till detta är det ovannämnda sättet på vilket företagens datorsystem har utvecklats. Företagen har under sin datorisering saknat förmågan eller insikten om och nödvändigheten i att vara konsekventa då de utvecklade sina databaser. De har istället ägnat sig åt att utveckla ett system åt gången. Om systemet passar ihop med övriga system har inte varit något högprioriterat område att undersöka. Denna frånvaro av att följa en övergripande plan avspeglas tydligt i dagens system där man kan finna okoordinerade och inkonsistenta databaser, Fred R. McFadden et al(1998). Detta leder till att det är svårt om inte omöjligt för beslutsfattare att få tillgång till kvalitativt bra data, vilket vidare leder till en ökad svårighet att fatta korrekta beslut.

Ytterligare en orsak till detta gap mellan önskad information och tillgänglig information kan vara att dagens system är utvecklade för att stödja den dagliga verksamheten och dess transaktioner och manipulationer av data. Detta leder till att de inte är anpassade till att behandla historisk data som kan verka som stöd vid beslutssituationer av olika slag.

För att komma tillrätta med denna skillnad i önskad och tillgänglig information så kan man med fördel använda sig utav ett Data Warehouse då detta samlar in data från olika källor och sammanställer den, samt presenterar den på ett meningsfullt sätt. Man kan således skapa sig en bredare syn av företaget eller organisationen.

## 4.2.Data Warehouse arkitektur

Ovan har vi presenterat beslutstödssystemets utveckling fram till dagens Data Warehouse koncept. I nedanstående delkapitel kommer vi att redovisa hur en typisk Data Warehouse arkitektur är uppbyggd. De i arkitekturen ingående delarna kommer att förklaras övergripande.



Figur 10:1 Typisk arkitektur för ett Data Warehouse. Connolly & Begg, Database Systems (1998)

Inom området så finns det flera olika forskare och dessa skiljer sig lite inbördes då de fokuserar på olika delar av Data Warehouse arkitekturen. Bland andra så fokuserar Inmon (1996) mest på den del av arkitekturen som handlar om verksamhetens operationella system och hur dessa försörjer Data Warehouse med data, medan Kimball (1996) har ett fokus som innefattar hela arkitekturen med de relaterade delarna såsom t.ex. slutanvändarverktyg och data mining<sup>3</sup>. Vårt fokus kommer dock att vara att se hur hanteringen av heterogenitet går till inom Data Warehouse arkitekturen, på grund av detta så kommer vi endast att ytligt behandla övriga arkitekturella delar.

### Operationell data

Den data som Data Warehouse använder sig av kommer från flera olika källor som t.ex. verksamhetens operationella nätverksdatabaser och relationsdatabaser som t.ex. Informix och Oracle vilka ingår i t.ex. RMS system.

<sup>3</sup> Data mining är ett sätt att hitta funktionella mönster i data genom att identifiera, förvärva och behandla data. Data mining fungerar bäst ihop med Data Warehouse. Berry & Linoff(1997)

Utöver detta så kommer data från arbetsstationer och privata servrar. Det är inte ovanligt att man också samlar in data från externa kommersiella databaser eller från kundens- eller leverantörens databaser.

### **Loadhanterare**

Denna del är den del som har hand om extraktionen(insamlandet) av data samt uppladdningen av data i Data Warehouse. Storleken på denna del och komplexiteten av den varierar från fall till fall och är uppbyggd av olika datauppladdningsverktyg och special anpassade program.

### **Warehousehanterare**

Warehousehanteraren utför alla operationer associerade med administration av data i Data Warehouse. Denna komponent är konstruerad på så sätt att den ska använda sig av färdiga datauppladdningsverktyg och speciella kundanpassade program. Operationer utförda av Warehousehanteraren inkluderar:

- Analys av data för att säkerställa konsistens.
- Transformation och sammanfogning av källdata från temporära lagringsenheter till Data Warehouse tabeller.
- Skapande av index och visning i bastabeller.
- Generering av denormalisationer om nödvändigt.
- Generering av aggregationer om nödvändigt.
- Back-up och arkivering av data.

I vissa fall genererar Warehousehanteraren alltså frågeprofiler för att avgöra vilken indexering och aggregation som är ändamålsenlig. En frågeprofil kan genereras för varje användare, grupp av användare eller alla användare av Data Warehouse och är baserad på information som beskriver frågekaraktäristika såsom frekvens, måltabeller och storleken av resultatet.

### **Frågehanterare**

Frågehanteraren (back-end komponent) utför alla operationer associerade med administration av användarfrågor Kimball(1996). Denna komponent är konstruerad för att använda sig av färdiga Data Warehouse avlyssningsverktyg, databas faciliteter, kundanpassade program samt dataaccessverktyg för slutanvändare. Frågehanterarens Komplexitet avgörs av funktioner tillhandahållna av slutanvändarverktygen och databasen. Operationer utförda av denna komponent inkluderar slussning av frågor till associerade tabeller och schemaläggning av frågeexekvering. I vissa fall kan frågehanteraren också generera frågeprofiler som tillåter Warehousehanteraren att avgöra vilka index och aggregationer är ändamålsenliga.

### **Databas med detaljerad data**

Denna del av Warehouset lagrar all detaljerad data i databasscheman. I de flesta fall är detaljerad data aggregerad till nästa detaljnivå. Detaljerad data är adderat till Warehouset för att komplettera aggregerad data.

### **Databaser med lite eller mycket summerad data**

Denna del av Warehouset lagrar data med olika grad av aggregering som är genererat med hjälp av Warehousehanteraren. Denna del av Warehouset är förgänglig eftersom



den kommer att bli subjekt för ständiga och upprepade förändringar för att kunna möta frågeprofilernas förändringar. Syftet med summerad data är att snabba upp utförandet av frågorna. Det blir dock ökade operationskostnader vilka är en konsekvens av att summera data. Detta borde kunna minimeras genom att ta bort begäran av kontinuerliga summeringsoperationer(som sort eller group by) i svarsalternativet till frågor ställda från användare. Summerad data uppdateras kontinuerlig när ny data införs i Warehouset.

### **Arkivering/ Backup av data**

I denna del av Warehouset lagras detaljerad och summerad data med syftet att möjliggöra arkivering och backup. Även om summerad data är genererad från detaljerad data kan det vara nödvändigt att ta backup på online summerad data om denna data hålls bakom retentionsperiod för detaljerad data. Denna data är överförd till lagringsarkiven iform av t.ex. magnetisk tejp eller optisk disk.

### **Metadata**

Detta område av Warehouset lagrar all metadata (data om data). Metadata används av alla processer i Warehouset och används för olika syften såsom:

- I extraktions- och uppladdningsprocesser används metadata för att katalogisera datakällan och visa informationen på ett ordinärt sätt inom Warehouset.
- Warehousehanteringsprocess. Metadata används för att automatisera produktionen av summeringstabeller.
- Som del av frågehanteringsprocess. Metadata används för att deligera ställd fråga till mest lämpligaste datakällan.

Strukturen på metadata är olika mellan varje process, eftersom deras syften är olika. Med detta menas att multipla kopior av metadata som beskriver samma datainnehåll hålls inom Warehouset. Det bör påpekas att de flesta färdiga verktygen för kopiering och slutanvändardataaccess använder sina egna versioner av metadata. Speciellt kopieringsverktygen använder metadata för att kunna avgöra katalogiseringsregler vid konvertering av källdata till Data Warehouse formen. Slut användaraccessverktyg använder metadata för att avgöra hur det ska utforma sina frågor. Hantering av metadata inom Warehouset är en mycket komplex uppgift, vilken inte bör underskattas.

### **Slutanvändaraccessverktyg**

Huvudsyftet med Data Warehousing är att tillhandahålla information för de som är aktiva inom strategiskt beslutsfattande. Dessa användare kommunicerar med Warehouset genom att använda sig av slutanvändaraccessverktyg. Data Warehouset måste effektivt stödja ad-hoc och återkommande analyser. Hög prestanda nås genom att analysera behoven av access, summering och periodiska rapporter till slutanvändaren. Vi kan katalogisera dessa verktyg i fem huvudgrupper enligt Connoly & Begg (1998).

- Rapportering- och frågeverktyg
- Applikationsutvecklingsverktyg
- Exekutivt informationssystemverktyg(EIS)
- Online analytisk processverktyg(OLAP)
- Data minings verktyg

### **Beslutstödsprogram**

Användaren opererar utifrån sina behov mot Data Warehouse med hjälp av olika typer av applikationer. Dessa applikationerna kan delas upp i tre kategorier:

- Verktyg för ad-hoc. frågor
- Flerdimensionella analysverktyg
- Skräddarsydda Executive Information Systems (EIS)

### **Verktyg för ad-hoc (spontana)frågor**

Med en ad-hoc. fråga, eller spontan fråga, avses en fråga mot en databas som användaren själv formulerar utan att utnyttja fördefinierade frågor eller rapporter. Tidigare var användaren tvungen att skriva frågor direkt i frågespråket SQL, vilket är för komplicerat för användare som inte har kunskaper i programmering. Nu finns det andra, enklare alternativ. Det vanligaste idag är att erbjuda någon form av grafiskt användargränssnitt baserat på visuell frågeteknik. Användaren formulerar frågor genom att peka och klicka i en datamodell. Ytterligare tekniker finns men vi kommer inte att ta upp dessa här eftersom dessa saknar relevans för vår undersökning.

### **Flerdimensionell analys (OLAP)**

Det område inom Data Warehouse marknaden som utvecklas snabbast just nu är slutanvändarverktyg för flerdimensionell analys. Nyckelordet för den här typen av verktyg är drill-down browsing, eller dive-down browsing. Med det avses att man successivt kan bryta ned data från helhet till detaljer. Användaren kan utifrån sina behov vända och vrida på informationen och därmed göra avancerade analyser och prognoser. Denna flerdimensionella ansats brukar benämnas OLAP(online analytic processing). Typiska uppgifter som kan utföras med ett program för flerdimensionell analys är t. ex. trendanalyser och försäljningsuppföljning.

### **Executive Information Systems (EIS)-tillämpningar**

Ett begrepp som var en föregångare till Data Warehousing är Executive Information Systems(EIS). Detta har nu kopplats ihop med Data Warehousing som ett beslutstödsprogram. Dessa är skräddarsydda och specifikt anpassade till en viss beslutsfattarens situation. Programmen har en begränsad användbarhet men är samtidigt ofta lätta att lära sig och använda. Ett Data Warehouse har också en traditionell rapportgenerator vilket möjliggör utformning av standardiserade rapporter efter behov.

### **Enterprise Data Warehouse (EDW) dataskikt (Reconciled data)**

Data i detta dataskikt syftar till att tillhandahålla en enkel övergripande datakälla för att underlätta beslutsfattande. Denna data är detaljerad, normaliserad, historisk, allsidig och kvalitets kontrollerad.

- Data är detaljerad och inte i någon större utsträckning summerad. Detta för att man skall kunna främja den flexibiliteten som behövs för att tillgodose olika användares krav och önskemål.
- Data är helt normaliserad till tredje normalform eller högre. Detta är den för att öka flexibiliteten och öka dess oantastlighet. Data blir alltså mer pålitlig och flexibel än om den hade varit denormaliserad. Man behöver inte denormalisera

data för att öka prestandan då man endast använder sig av viss data periodvis via så kallade batch körningar.

- Data är historisk på så vis att den är lagrad som periodisk data.
- Data är allsidig på så vis att den återspeglar den organisation som ligger till grund för dess data modell.
- Data när kvalitetskontrollerad för att den sedan skall summeras i de så kallade data mart, för att vidare ligga tillgrund för beslutsfattande. Användarna måste veta att befintlig data är av hög kvalitet annars så kommer de inte att använda sig av systemet.

### 4.3. Data Warehouse konceptets tolkning av heterogenitet

I detta avsnitt så kommer vi att redogöra för den miljö som Data Warehouse existerar i och de faktorer som skapar ett behov av Data Warehouse.

En organisations informationsförsörjning är i grunden heterogen. Information flödar åt alla håll. Information kommer från organisationens omgivning i form av transaktioner, uppgifter om marknadsutveckling, uppgifter om konkurrenssituation inom branschen, uppgifter om olika offerter från leverantörer eller förfrågningar från kunder. Informationen går ut till organisationens intressenter i form av organisationens prestationer, fakturor, beställningar, leveranser, följesedlar, etc. Vidare flödar informationen mellan aktörer som tillhör samma avdelning samt aktörer som tillhör olika avdelningar, för att på detta sätt samordna deras insatser. Information flödar upp i form av uppnådda resultat eller problem samt flödar ner till olika aktörer och ansvariga i form av tidsplaner, uppdateringar av tidsplaner, visioner, strategier, etc för att på detta sätt kan alla bli informerade om hur organisationen fungerar. Det var denna föreställning som blev R. Anthonys utgångspunkt för att organisera en organisations informationsförsörjning. Men hur kan man överblicka och beskriva informationsförsörjnings heterogenitet?

Bakom Data Warehouse konceptets uppkomst ligger G. A. Gorrays och M. S. Scott Mortons modell(1971) om informationens egenskaper inom en organisation. Modellen presenterades i början av 70-talet. Klassificeringen grundas bl.a. på R. Anthonys grundidé som bygger på att göra viss information tillgänglig för hela organisationen.

Info-egenskaper	Operativ nivå	Funktionell nivå	Strategisk nivå
Informations Källa	Mest Intern		Mest extern
Omfång (räckvidd)	Väl definierad, smal		Mycket Bred
Aggregeringsnivå	Detaljerad		Aggregerad
Tidshorisont	Historisk		Framtid
Aktualitet	Aktuell		Gammal
Konsistens	Hög		Låg
Användnings frekvens	Hög		Vid behov

Figur 4:1 Informationens egenskaper enligt G. A. Gorry och M. S. Scott Morton

Vanligtvis kan en organisation betraktas som en informationsmiljö. En sådan miljö utgörs av de inbördes förhållanden som råder mellan människor (användare) och organisationens informationssystem. I stora organisationer utgörs informationsmiljön av två distinkta miljöer: (1) den operationella miljön, och (2) beslutsmiljön. Vidare kan beslutsmiljön delas in i ytterligare en s.k. funktionell informationsmiljön och en strategisk informationsmiljön.

Organisationens operationella informationsmiljö omfattar en rad olika verksamhetssystem, dvs system som används i daglig drift i verksamheten. Systemen är integrerade till olika verksamhetsprocesser och genererar data som är detaljerad, aktuell och föremål för förändring. Verksamhetssystemen är skapade för att köras i realtid med aktuella data. Exempel på sådana system är bland andra patientbokningssystem, ordersystem och bokningssystem. Dessa system behandlar ofta stora mängder relativt enkel och förutsägbar data. Dessa verksamhetssystem kallas ofta för transaktionssystem OLTP (OnLine Transaction Processing). Transaktionssystemen är som nämnt skapade för att maximera transaktionsprocesshastigheten detta skiljer dem från beslutstödssystemen som är skapade för att stödja ad-hoc frågeprocesser. Den operativa verksamheten är delvis beroende av den information som produceras i organisationens beslutsmiljö.

Organisationens beslutsmiljö omfattar en rad olika specialiserade och sofistikerade informationssystem, såsom beslutstödssystem (DSS) materialplaneringssystem (MPS), Produktionsplaneringssystem (PPS), Marknadsföringsinformationssystem (MaKiS), Ledningsstödsystem (ESS), Ekonomisystem, (MCIS), etc. Den verksamhet som bedrivs inom en beslutsmiljö är delvis beroende av den information som produceras i organisationens operativa informationsmiljö.

Som vi har redovisat tidigare i kapitel 3, så är informationsöverföringen mellan den operativa miljön och beslutsmiljön varken problemfri eller socialt neutralt. Men det förekommer ofta starka informationsmässiga beroendeförhållanden som kräver att dessa informationsmiljöer, trots deras enorma heterogenitet skall integreras eller samordnas på ett sätt som främjar informationstillgängligheten. Det är alltså i huvudsak tre olika faktorer som är drivkraften bakom Data Warehouse.

- Önskan om en bredare överblick av organisationens informationsförsörjning
- Nödvändigheten av att göra informationen tillgänglig genom att samordna verksamhetens operativa system och beslutstödssystem och därmed underlätta beslutsfattandet
- Villkoret är att varken integrationen eller samordningen skall påverka de operativa systemens funktion, utformning och innehåll.

*‘Det ultimata målet med ett Data Warehouse är att integrera ett företags övergripande data till en enda lagringsplats, från vilken användarna kan hämta data på ett enkelt sätt och bearbeta den, i den takt som passar honom/henne’.*

#### **4.4.Data Warehouse Arkitektur för hantering av heterogenitet**

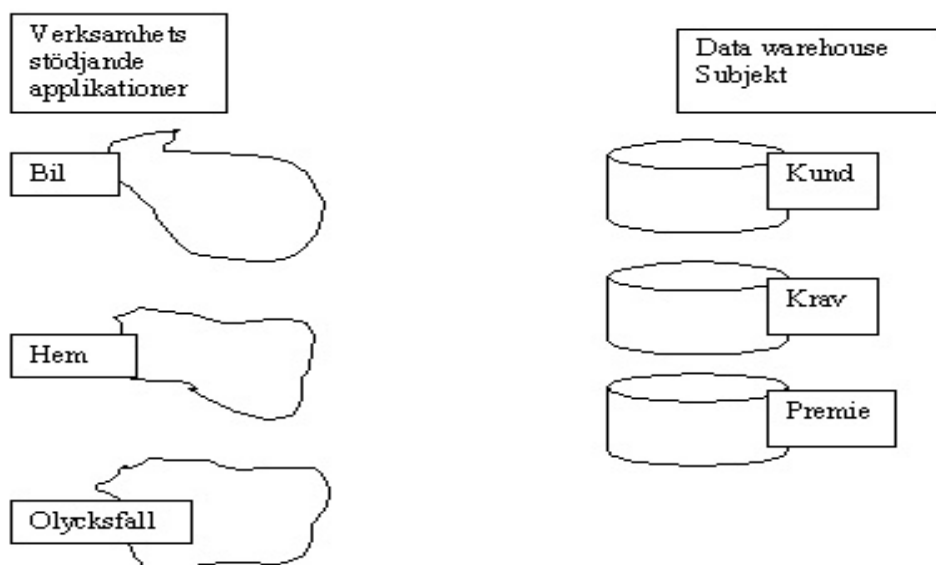
Originalkonceptet av Data Warehouse kom till genom IBM som ”Information Warehouse” och presenterades som en lösning på problemet med tillgängligheten av data som var lagrad i icke relationsbaserade system. Information Warehouse var avsett att göra det möjligt för organisationer att använda sina egna dataarkiv och

hjälpa organisationerna med att utnyttja arkiven på ett affärsmässigt sätt. Det finns talrika definitioner av Data Warehouse. Det senaste definitionerna fokuserar på data karakteristiska av data som hålls i ett Data Warehouse.

*‘Ett Data Warehouse utgör ett "interface" mellan organisationens operationella miljöer och beslutsstödmiljöerna. Detta på så vis att data som är lagrade i olika operationella system (informations källor), organiseras och görs tillgängliga för beslutsfattarna över hela organisationen’.*

Med andra ord, tack vare Data Warehouse konceptet, får organisationen en integrerad källa för all data. Om data är åtkomlig för vem som helst så blir jobbet för användaren av t.ex. BSS-systemen mycket lättare än om han hade jobbat i en traditionell miljö och försökt att få fram samma resultat W.H.Inmon (1996). Kortfattat är ett Data Warehouse, data management- och data analysteknologi.

Operationella system är vanligtvis byggda kring verksamhetsprocesser eller liknande såsom för ett försäkringsbolag t.ex. bil, hem, olycksfall, m.m. W.H.Inmon (1996). Systemen sinsemellan är det stora skillnader i regelbaser samt i vilken information som finns där.



Figur 4:2 Exempel på subjekt orientering av data(W.H.Inmon 1997).

**Databashanteringssystem och Data Warehouse.** Skapandet av Data Warehouse har medfört att det har kommit fram en ny typ av databashanteringssystem som W.H.Inmon kallar för Data Warehouse-specific-database management systems. Dessa system skiljer sig från de vanliga databashanteringssystemen(DBHS) på så vis att de är optimerade för de uppgifterna som är nödvändiga för Data Warehouse, uppladdning och access av data.

**Uppdatering.** Data från de operationella systemen integreras, transformeras och laddas upp i Data Warehouse och när den har kommit dit så uppdateras den inte. Med andra ord så är den största och mest framträdande skillnaden mellan traditionella

DBHS och Data Warehouse specifika DBHS en fråga om uppdatering. Även om man drar ner uppdateringarna till ett minimum i traditionella DBHS så kan man inte ta bort dem helt, detta är inte fallet med ett Data Warehouse specifikt DBHS där uppdatering inte används annat än i undantagsfall.

**Behandling.** Den behandling som utförs på data i Data Warehousemiljö är access och olika slags analyser. Om man behöver göra några rättelser i databasen så görs detta när den inte används för analysarbete.

**Volym.** En ytterligare skillnad mot vanliga DBHS är den att ett Data Warehouse tenderar till att innehålla större mängder data än vad som är vanligt för operationella DBHS.

**Innehåll.** Data Warehouse håller mycket data på grund av att de innehåller:  
Elementära (atomisk, detaljrik) data så väl som högupplöst data  
Historisk data såväl som aktuell och framtids data  
Aggregerad data såväl som detaljerad

**Användning.** Data Warehouse är designat för ett relativt lågt antal transaktioner vilka är ganska oförutsägbara till sin natur. Systemet är organiserat utifrån ett antal potentiella frågor och skall stödja ett relativt litet antal användare, främst inom ledningen.

**Indexering.** Ytterligare en skillnad mellan de olika typerna av DBHS är indexeringen av data. Ett traditionellt system har ett ändligt antal index, detta på grund av att när man uppdaterar eller lägger till poster och objekt så leder detta till att indexeringen kräver en egen datahantering. Då det inte sker någon uppdatering av data i ett Data Warehouse och det som prioriteras är accessen av data så finns det ett behov av många index och även möjlighet att ha detta. Faktum är att det finns möjlighet att skapa en mer robust och sofistikerad indexering än vad som är möjligt i traditionella DBHS.

**Optimering.** De operationella databaserna är optimerade för transaktions access och manipulation av data. Genom att organisera data på detta sätt så kan man länka samman olika typer av data med hjälp av nycklar. I de databaser som är byggda för informations access (dvs Data Warehouse) har data en annorlunda fysisk uppbyggnad. Den är organiserad på så vis att många poster av samma data typer kan accessas samtidigt.

**Beroende.** Trots att OLTP systemen och Data Warehouse systemen är relativt olika till sina karaktärer och syften, så har de en nära relation. Detta på grund av att OLTP systemen är de systemen som försör Data Warehouse med data. Det är denna data som kommer från de olika OLTP systemen som är heterogen och detta måste rättas till innan man kan mata in data i Warehouse.

**Lagringsmedia.** Som bilden nedan (figur 4:3) visar så kan även typen av lagringsmedia skilja sig åt även inom ett subjekt område.

**Externa och ostrukturerad data.** De flesta organisationer bygger sitt första Data Warehouse med hjälp av interndata som det får från existerande system. Denna data

som man får från dessa system är strukturerad data, detta på grund av att den redan tidigare blivit anpassad till organisationens redan förekommande format. Det finns dock mycket data som är viktig för organisationerna i deras dagliga verksamhet som inte genereras av organisationens egna system. Denna data brukar betecknas som externdata. Det är inte ovanligt att denna externa data kommer in till organisationen i ett ostrukturerat oförutsägbart format. Datan i sig säger inget om företaget men den talar om saker om miljön som företaget verkar i och konkurrerar med andra i. Ett Data Warehouse är ett utmärkt ställe att lagra externa data på. Detta löser många utav de problem som kan uppkomma om man inför data manuellt i systemen. Om man för in data manuellt så förlorar man källan. Det lagras alltså ingen data om vad som är källan. Andra problem som man kan råka utför är att om man matar in data manuellt är att när man använt den en gång så är det svårt att använda den vid kommande tillfällen. Detta är olyckligt då mycket av externdata är användbart över tiden. Den typen av data som kommer från externa källor är av varierande slag så som t.ex.

- Affärstidningar
- kreditvärderingsinstitut
- Dagstidningar
- Fackpress
- Produktkataloger
- Marknadsanalyser
- Skatteförvaltning

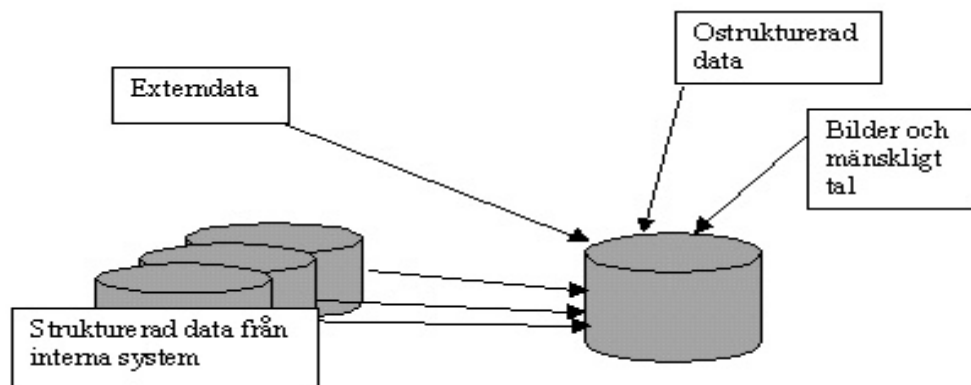
**Extern och ostrukturerad data i Data Warehouse.** Till skillnad mot intern data så finns det inget mönster eller riktig struktur i den externs datastrukturen. Uppdaterings frekvensen är också ett problem som man måste lösa och detta kan man göra genom att ständigt bevaka källan så att man får fångat in rätt data(W.H.Inmon).

Ett ytterligare problem med externdata är att formen är totalt odisciplinerad. För att kunna göra denna data användbar och för att kunna föra in den i Data Warehouse så måste man göra ganska stora formateringar av extern data för att kunna transformera den till ett acceptabelt format.

Den tredje faktorn som är ett problem då man skall använda extern data är att extern data är dess oförutsägbarhet. Den kan komma från i princip var som helst när som helst. Denna oförutsägbarhet när det gäller tillgängligheten hos extern data gör att det är mycket svårt att hålla extern data konsistent.

Som utökning av den externa data som kommer från tidskrifter eller rapporter så finns det en ytterligare klass av extern data. Detta är data i form av bilder eller mänskligt tal. De teknologiska möjligheterna för att kunna sammanställa denna typen av data är mycket begränsad idag på grund av att tekniken är dåligt utvecklad. När tekniken finns så kommer den att kräva mycket stora lagringsutrymmen.

Hursomhelst så finns det stora möjligheter med insamlingen och lagringen av ostrukturerad extern data i ett Data Warehouse.



Figur 4:3 De olika typerna av data som kommer till ett Data Warehouse.

## 4.5. Data Warehouse konceptets grundläggande principer

I kommande del kommer vi att presentera de olika grundläggande avgränsningsprinciper som styr Data Warehouse arkitekturen.

Ett Data Warehouse har data som är: (1) Subjekt orienterade, (2) Integrerade, (3) Historiska, och (4) Beständiga. Det är detta som leder fram till att framställningen av en s.k. Data Warehouse arkitektur styrs av fyra avgränsningsprinciper, nämligen:

- Data Warehouse - innehåller såväl "subjektdata" som metadata (Abstraktionsprincipen)
- Data Warehouse - innehåller integrerade data, (Integrationsprincipen)
- Data Warehouse- innehåller såväl detaljerade som aggregerade data, (Aggregeringsprincipen)
- Data Warehouse - innehåller såväl aktuella som historiska data och framtids- data, (Evolutionsprincipen)

### 4.5.1. Abstraktionsprincipen

Abstraktionsprincipen klargör främst skillnaden mellan objekt och namn. För att data skall bli tillgängliga över hela organisationen och mer specifik för den funktionella och den strategiska miljön är det viktigt att införa objektbegreppet, (Se Orman).

#### Subjektinformation, (objektinformation)

Objektbegreppet möjliggör att se samma data på olika sätt och det är precis vad kännetecknar en Data Warehouse arkitektur. Med andra ord, data i Data Warehouse är organiserade enligt det s.k. avbildningsparadigm. Här är det viktigt att klargöra vår tolkning av begreppen subjekt orientering och objektorientering som identiska. . Vilka dessa objekt/subjekt är har man definierat i datamodellen. Typiska subjekt områden kan vara t.ex., Kund, Produkt, Aktivitet

*'Data Warehouse innehåller såväl "objektdata" som metadata'*



De största subjekt områdena implementeras som relationstabeller i ett Data Warehouse. Se exemplet "kund" nedan.



Figur 4:4 Exempel på upplägg av kund data.

**Nyckelbegrepp.** Det som håller ihop de olika tabellerna är och vidmakthåller deras inbördes relation är (1)Nyckelbegrepp, i detta fallet nyckel fältet "K\_id", (2)tidsreferens. Alla tabeller har någon form av tid i sin nyckel.

**Tidsresolutionen** det vill säga att utformningen av tidsreferensen kan anges som kontinuerlig tid som håller tiden i form av "från datum till datum" eller så kan det vara kumulativa tidsfält som håller tiden i form av t.ex. månad.

**Precisering av tidsreferens.** Utöver den ovan nämnda tidsreferens så finns det mer preciserade tidsreferenser som håller tiden i form av klockslag eller dag. Gemensamt för alla data är att de har någon form av tidsreferens kopplat till nyckeln, även om formen på dessa kan skilja sig åt. Ovan har man två lika bastabeller (kunddata) för kund med lite olika definitioner beroende på år.

**"Systemtid"**(Förvaltningsinformation). Tabellerna kan alltså ha olika definitioner beroende på när de skapades. Detta beror på att det ställdes olika krav vid olika tidpunkter när man utvecklade Data Warehouse. Man har även baser med "K\_aktivitet" vilka uppdateras månadsvis med sammanställningar över vad varje kund har haft för aktiviteter under aktuella perioder. Även här skiljer definitionerna sig åt lite grann ( se J. Bubenko om systemtid, dvs tiden där informationen tas emot av systemet).

### Metadata och externa data

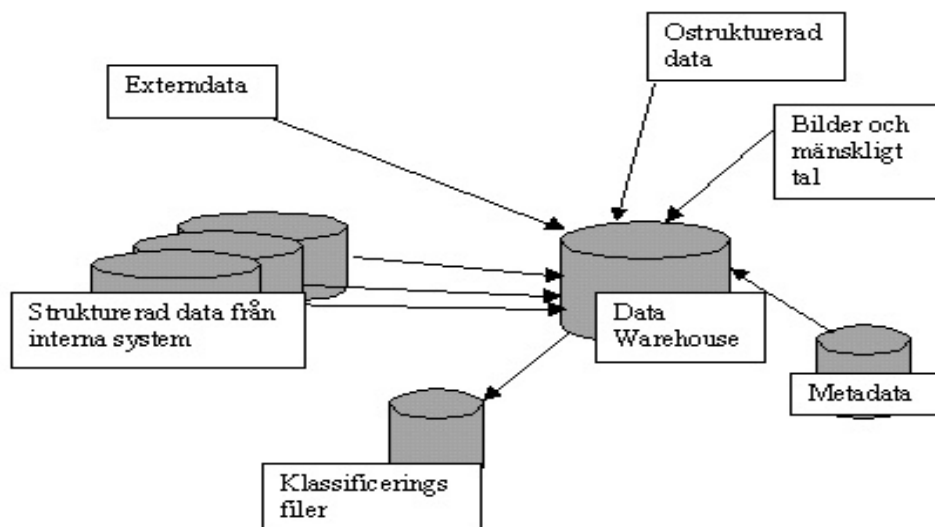
Metadata är en viktig komponent i Data Warehouse. Dess roll när det gäller extern data är dock annorlunda än i övriga fall med intern data. I fallet med extern data så används metadata till att registrering, access och kontroll av extern data i Data Warehouse miljön. För att förklara nödvändigheten av metadata så kan man titta på innehållet i meta data i ett Data Warehouse för extern data. Se nedan.

- Dokument Id
- Datum för införande i Data Warehouse
- Beskrivning av dokumentet
- Källan/ursprunget för dokumentet
- Datumet på källan/ ursprunget för dokumentet( när ar det skrivet)
- Klassificering av dokumentet
- Index ord, sök ord
- purge data
- Adressen till filen
- Filens storlek
- Relaterade dokument

Det är genom denna meta data som man som användare får information om extern data. I många fall så är det så att man tittar enbart på metadata om ett dokument under processen att finna rätt dokument. Man sparar alltså mycket tid på att inte behöva läsa alla dokument eller filer för att finna ett relevant dokument.

Riktigt uppbyggd och rätt underhållen metadata är av yttersta vikt när det gäller användandet av externdata i ett Data Warehouse.

Som komplement till metadata så finns det en fil som är skapad för användarna av Data Warehouse vilken klassificerar den data som är relevant för en specifik användare.



Figur 4:5 Exempel på de olika typerna av data som är förknippade med ett Data Warehouse.

### **Kraftigt ostrukturerade data**

Extern och ostrukturerad data kan lagras i Data Warehouse om det är kostnadseffektivt. Det finns dock fall då det inte är möjligt eller ekonomiskt försvarbart att lagra denna typen av data i Data Warehouse. Det man kan göra i dessa fall är, att i metadata som tillhör Data Warehouse, lägga in en hänvisning om var man kan finna eftersökt data någonstans. Alltså var man har lagrat eftersökt data någonstans utanför Data Warehouse.

### **4.5.2. Integrationsprincipen**

Enligt vår tolkning reflekterar begreppet integration de logiska förhållandena mellan flera olika partitionerade data. Partitionering förekommer i tre former: (1) vertikal, (2) horisontell, och (3) innehållsmässig, dvs partitionering mellan fakta information och referensinformation.

#### *'Data Warehouse innehåller integrerade data'*

**Partitionering av data.** Ett annat mycket stort och viktigt design område är det som handlar om partitionering av data. Om man har löst problemen med partitionering och problemen med aggregering så är övriga design frågor lätta att lösa och om man inte gör detta så spelar övriga designområden egentligen ingen roll (W.H.Inmon, 1996).

**Vertikal partitionering.** Denna form av partitionering förekommer för att eliminera alla beroende förhållande mellan attributen. Data i ett Data Warehouse representerar elementär information. Med andra ord så omfattar ett Data Warehouse icke normaliserade data. Den form av partitionering utgör grunden till Data Warehouse koncept. Enligt Inmon kan ett Data Warehouse kallas för elementär databas, (E-bas) på grund av att den innehåller i princip elementär information.

**Horisontell partitionering.** Syftet med denna form av partitionering av detaljerad data som har en hög grad av aktualitet är att fördela den över fler och mindre fysiska enheter. Detta gör man för att det ger en större grad av flexibilitet för personalen i hanterandet av data och utrustning än om det hade legat på en gemensam enhet. Fördelarna med en uppdelning av data över flera enheter är bland annat följande:

**Partitionering av fakta information och referensinformation.** Data Warehouse Schema utgör en logisk struktur vilken har en faktatabell (innehåller riktiga data) i centrum, omgiven av dimensionstabeller (innehåller referensdata, referensinformation). Faktatabellen innehåller en främmande nyckel för varje dimensionstabell. Denna struktur har hand om karakteristiska av faktadata såsom fakta genererade av händelser som skett och inte är troligt att ändras oberoende av hur dessa är analyserade. Eftersom största delen av data i ett Data Warehouse är representerad med fakta kan faktatabeller bli enormt stora i förhållande till dimensionstabeller därför är det viktigt att behandla dessa data som read-only referensdata vilken kommer att ändras med tiden. För att kunna identifiera fakta data från dimensionsdata, man skall identifiera kärntransaktioner i varje verksamhetsapplikation.

Sammanfattningsvis, i Data Warehouse sammanhang så är inte frågan om man skall dela upp data utan hur man skall göra det. Samtidigt små enheter med data kan lätt bli:

- Omstrukturerade
- Indexerade
- Sekventiellt avlästa om nödvändigt
- Återskapade efter en krasch
- Övervakade
- Oberoende enheter kan ha olika definitioner
- Olika enheter kan delta i olika operationer och processer samtidigt utan sänkt prestanda

De ovan nämnda punkterna gäller inte för stora fysiska enheter. Själva essensen hos ett Data Warehouse kan sägas vara flexibiliteten i accessen av data (W.H.Inmon 1996). Om man har stora fysiska enheter så motverkar man detta syftet. Därför bör man dela upp all aktuell detaljerad data.

**Redundansfrihet.** Sammanfattningsvis kan man säga att med partitionering av data menas, att data med samma struktur delas upp på två eller fler fysiska enheter. När data är uppdelad så kommer en given del data att finnas på enbart ett ställe, redundans tillåts alltså inte

**Partitioneringskriterier.** Det finns många olika kriterier vilka man kan dela upp data efter. Vilka av kriterierna man väljer att kombineras är upp till utvecklaren. Exempel på dessa kriterier är:

- Datum/tid
- Organisationsdel
- Geografi
- Kombination av flera

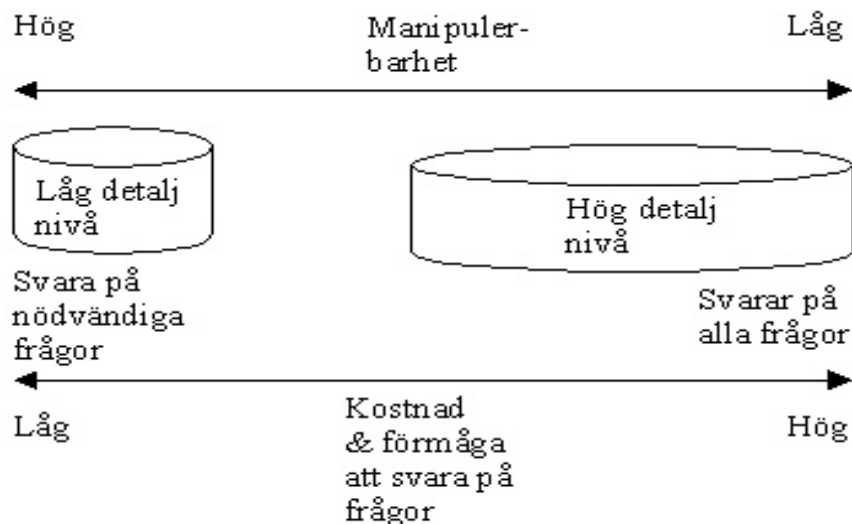
Det är dock vanligt att man brukar kombinera tid med andra kriterier då man gör uppdelningen.

#### ***4.5.3. Aggregeringsprincipen (Granularity)***

W.H.Inmon(1996) anser att aggregeringsprincipen utgör Data Warehouse arkitekturens viktigaste princip. Principen kan sammanfattas i följande termer;

#### ***Data Warehouse innehåller såväl detaljerade som aggregerade data***

För att t.ex. svara på en fråga av typen – Hur många samtal ringde stockholmarna i snitt under december månad 1998? Så kan man använda sig av filer med varierande grad av detaljrikedom. Om man använder sig av filer med hög detalj nivå så kommer detta att kräva stora resurser. Det kan vara fullt möjligt att besvara frågan med låg detalj nivå om man har rätt detaljer. Man måste alltså göra en avvägning mellan vilken data som man behöver och vad som man är beredd att satsa. De olika faktorerna samverkar enligt bilden nedan.



Figur 4:6

Inom varje subjekt området så finns det både aggregerade (summerade) och detaljerade data över samma tidsperiod. Detta benämner W.H.Inmon som "shift of granularity" vilket innebär att data finns representerade med olika härledningsgrad (dvs kornighetsgrad).

**Graden av härledbarhet** (kornighet) beror på den nivå av summering som man gjort på data i Data Warehouse enheter och således på dess detalj rikedom. Härledbarheten (kornigheten) hänger samman med detaljeringen på så vis att när detaljeringen ökar så minskar härledbarheten (kornigheten) och tvärt om. Som liknelse kan man ta skärpan på en bild. En bild med hög upplösning (detaljering) har låg kornighet. Detaljeringsnivån påverkar storleken på bilderna, detta är även fallet med Data Warehouse där filernas storlek ökar med detaljeringsnivån. Detta med att hög detaljering ökar storleken på filerna är det som gör att detta område är så viktigt. Man måste vid designen av ett Data Warehouse göra en avvägning av mängden data som man behöver och den detaljeringsgrad som man anser vara nödvändig.

Det är dock ofta så att man behöver ha möjligheten att kunna svara på alla typer av frågor och det är därför lämpligt att man noga överväger vad man behöver för svar innan man börjar att bygga något som är för stort.

**Dubbla detaljnivåer.** Oftast så är det nödvändigt att ha en hög effektivitet i lagringen, accessen av data och i möjligheten att analysera data i detalj. Man kan alltså behöva ha en hög detalj nivå samtidigt som man inte vill ha detta av effektivitets och underhålls skäl. Man vill äta kakan och ha den kvar med andra ord. Dessa kraven medför att man kan behöva ha två eller fler detaljnivåer av data.

#### 4.5.4. Evolutionsprincipen ( Informationens tidshorisont)

Händelser är förändringar i objektens tillstånd. Det finns två slags händelser som är relevanta med Data Warehouse arkitektur, (1) existentiella händelser som skapar nya

objekt och tillståndshändelser som förändrar de berörda objektens attributvärde. I den bemärkelse:

*'Data Warehouse innehåller såväl aktuella som historiska och framtids data'*

**Statusdata och händelsedata.** Beroende av hur data integreras i systemen kan man identifiera två slags integrationsfilosofier. Den första integrationsfilosofi kan bara ge information om det aktuella tillståndet (statusdata) medan den andra integrationsfilosofi kan ge information om så väl historiska data som data om den framtida händelseutvecklingen. Med andra ord, statusdata beskriver det aktuella läget på ett objekt som t.ex. ett konto. Händelse data är data som representerar en händelse som t.ex. ett uttag från kontot ifråga. Man kan beskriva en företags transaktion som en aktivitet som resulterar i en eller flera händelser på databas nivå. ( T.ex. händelse information reflekteras i "insert into" medan status information reflekteras i "update"). I ett Data Warehouse är möjligt att lagra både status och händelse data. Det är dock vanligast att man lagrar status data. Händelse data lagras oftast endast en kortare tid för att sedan arkiveras eller raderas, detta för att spara utrymme. ( Båda typerna lagras dock i databasernas loggfiler, detta för att man skall kunna göra back up eller för att man skall kunna återskapa förlorad data. Dessa loggfiler spelar en viktig roll då man skall fylla Data Warehouse med data )

**Transistent data kontra periodisk data.** Det är ofta nödvändigt att lagra data i Data Warehouse om när en viss händelse inträffade. Detta behövs om man skall kunna jämföra speciella händelser vid samma tidpunkter eller vid jämförelser av händelser vid olika tidpunkter.

De flesta operationella verksamhetssystemen brukar använda sig av transistent data vilket innebär att när en händelse inträffar så skrivs det befintliga värdet över med det nya. Detta görs utan att det ursprungliga värdet sparas

*'Transienta data     $A = A + B$  Eller  $A = A * B$ '*

*'Periodiska data     $C = A + B$  Eller  $C = A * B$ '*

Man förlorar alltså det tidigare värdet i databasen. I och med att man använder databaser så har man tillgång till loggfiler och genom detta så kan man återskapa bilden av vad som har varit. Nedan visas ett exempel på hur ursprungliga data skrivs över. Enda sättet att få fram ursprunglig data är sedan att söka i logg filerna. Pilarna visar var ändringarna är gjorda.

Tabell Anstalld(1999-02-08)

Id	F_namn	E_namn	Tillgodo
9090	Olle	Carlsson	2190
9091	Pontus	Svensson	1531
9192	Bror	Nilsson	5568
9793	Anders	Berg	110
9898	Peder	Gran	465

Tabell 1

Tabell Anstalld(1999-02-11)

Id	F_namn	E_namn	Tillgodo
9090	Olle	Carlsson	2190
9091	Pontus	Svensson	0
9793	Anders	Berg	110
9898	Peder	Gran	400
9903	Mats	Von Löv	0

Tabell 2

Figur 4:7

I filen syns nu inga spår av de tidigare värdena på data. Det är detta som är utmärkande för transistent data. För att man skall kunna återskapa data så kan man som nämnt använda sig av logg filerna. I loggfilerna finns både status och händelse data lagrad. Filerna kan se ut enligt nedan.

Före

9091	Pontus	Svensson	1531	1999-02-08
------	--------	----------	------	------------

Update

9091
1999-02-11
-1531

Händelse:  
Uttag

Efter

9091	Pontus	Svensson	0	1999-02-11
------	--------	----------	---	------------

Tabell 3

Figur 4:8 Loggfil

Periodiska data. Om man jämför transistenta data med den periodiska så är skillnaden att den periodiska inte ändras eller tas bort. Är den väl införd en gång så stannar den där. För att veta det aktuella värdet på ett objekt så får man använda sig av de tids angivelser eller tidsstämplar som det också kallas, som talar om när senaste ändringen är gjord. Den data som har den senaste ändrings stämpeln är den som är den aktuella. I och med att man sparar tidigare värden så kan man lätt gå in och se de historiska

värdena. Nedan visas ett exempel på detta. Jämför skillnaden med ovan visade exempel.

Tabell Anstalld(1999-02-08)

Id	F_namn	E_namn	Tillgodo	Datum	Händelse
9090	Olle	Carlsson	2190	1999-02-08	C
9090	Pontus	Svensson	1531	1999-02-08	C
9192	Bror	Nilsson	5568	1999-02-08	C
9793	Anders	Berg	110	1999-02-08	C
9898	Peder	Gran	465	1999-02-08	C

Tabell 4

Tabell Anstalld(1999-02-11)

Id	F_namn	E_namn	Tillgodo	Datum	Händelse
9090	Olle	Carlsson	2190	1999-02-08	C
9091	Pontus	Svensson	1531	1999-02-08	C
9091	Pontus	Svensson	0	1999-02-11	U
9192	Bror	Nilsson	5568	1999-02-08	C
9192	Bror	Nilsson	5568	1999-02-11	D
9793	Anders	Berg	110	1999-02-08	C
9898	Peder	Gran	465	1999-02-08	C
9898	Peder	Gran	400	1999-02-11	U
9903	Mats	Von Löv	0	1999-02-10	C

Tabell 5

Figur 4:9

Det som skiljer de periodiska datatabellerna är de två nyafälten Datum och Händelse. Fältet datum fungerar som tidsstämpel liknande den i databas loggen vilken talar om när en rad blev modifierad, även klockslag kan förekomma. Fältet händelse visar vilken händelse som utfördes vid ett visst datum, som alltså visades i datumfältet. C, U, D, står för create, update, delete. Ut över dessa fält så är tabellerna uppbyggda så att alla data lagras, har de en gång skrivits in så kommer de alltid att finnas där. Detta är en av anledningarna till att kravet på lagringsutrymme ökar snabbt. Detta ställer krav på beslutsprocessen då man skall bestämma vilken nyckeldata som skall lagras så här.

**Användningsfrekvens.** Identifiering av tidsperioden som krävs för varje beslutssupport applikation. Detta uppnås genom identifikation av tidsperioden som signifikant för beslutsfattande processen och behovet av nivån på detaljerade data. Till exempel för att svara på en fråga om de möjliga intäkter för uthyrda objekt för det kommande halvåret kanske inte krävs att man undersöker intäkterna för det föregående halvåret men kanske krävs en undersökning av samma månad under de senaste två åren. För övrigt nivån på detaljkravet för att svara denna fråga kan begränsas till jämförelse av aggregerat data och kanske inte kräver tillgång till detaljerat data. Behållningsperioden och nivån på nödvändigt detaljerat data borde vara identifierad för varje beslutssupport applikation.



Avgöra kravet för statistiska exempel på delmängd av data gentemot kravet för detaljerat data. Det är möjligt att reducera volymen av detaljerad information lagrad i Data Warehouse genom att spara en perspektiv exempel av detaljerat data tillsammans med växlande aggregationer för ett komplett set av data. Till exempel för att svara på föregående fråga (Se föregående stycke) kan endast krävas en jämförelse av intäktsmedelvärdet av egendom för uthyrning i varje stad hellre än detaljerad intäkter av varje filial av företaget.

## **4.6. Migration till Data Warehouse miljö**

I detta delkapitel så redovisar vi hur det går till när man migrerar till en Data Warehouse miljö. Vi går igenom de olika relaterade procedurerna som är nödvändiga att genomföra då man migrerar till en Data Warehouse miljö.

### ***4.6.1. Heterogenitet i källsystem***

När man skall fylla ett Data Warehouse med data så plockar man data från flera olika typer av system. Dessa kan vara flera olika interna system som är skapade i olika syften vid olika tider samt från olika externa källor så som databaser och tidskrifter.

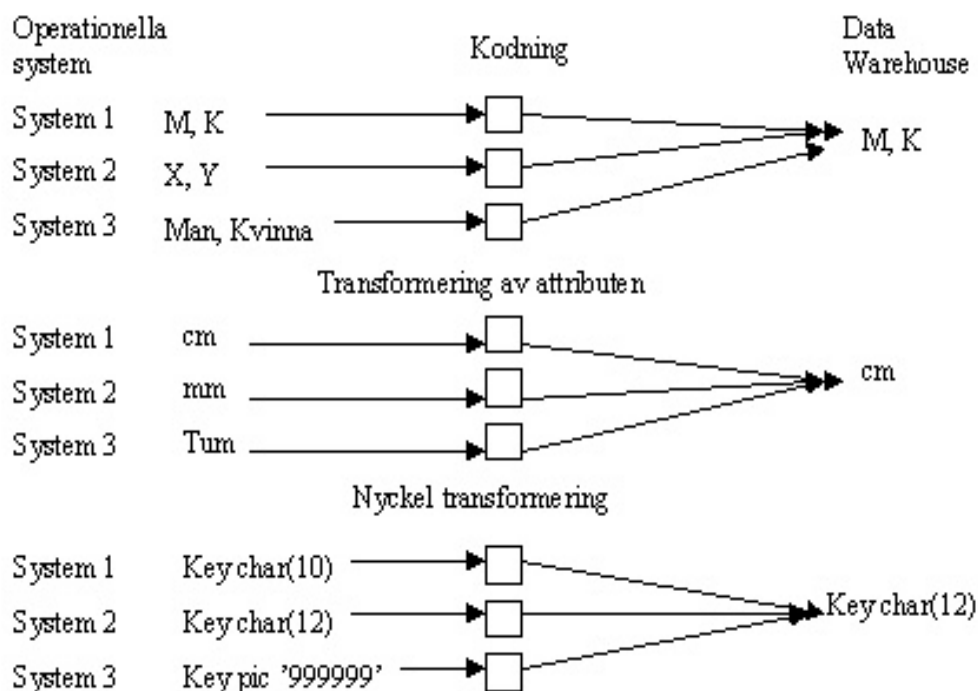
Interna system och interna data. När man skall fylla ett Data Warehouse och använda sig av tillgänglig data i olika typer av interna operationella system som finns inom så är det troligt att man kommer att finna stora skillnader på data som finns tillgänglig i dessa system.

De vanliga operationella systemen är idag uppbyggda kring organisationens olika operationella verksamheter samt kring de olika delverksamheter som finns inom dessa.

Dessa systemen är skapade under många år utav många olika människor i olika syften. Följden av detta är att under årens lopp så har designerna tagit tusentals beslut som är i mångt och mycket olika. Detta har fått till följd att saker som att det finns ingen konsistens i t.ex.

- Kodning( vilka förkortningar eller liknande man skall använda)
- Reglerna för hur man skött namngivning av objekt
- Definitionen av vilka nyckelattribut man skall ha.
- Hanteringen av attributen(Vad man skall ha för data typer på attributen)

Detta är bara ett axplock av de olika besluten som kan tas och skillnaderna har uppkommit på grund av de fria tyglar som man gett designerna under utvecklingen av systemen.



Figur 4:10

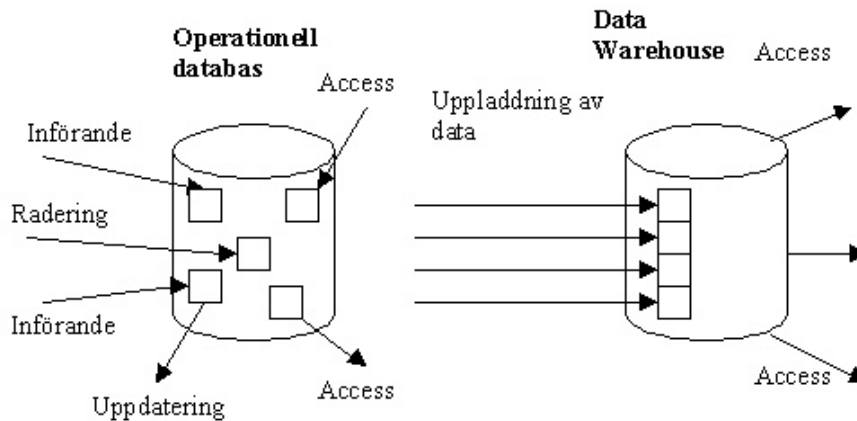
När man för in data till Data Warehouse så rättar man till de inkonsistenser och den heterogenitet som finns mellan de olika operationella systemen. Se bilden ovan. Om man tittar på resultatet i Data Warehouse så spelar det ingen roll om man benämner könet som Man, Kvinna eller som M, K. Det som spelar någon roll är att det görs och att det görs konsekvent och att informationen blir konsistent.

En som Inmon ser det stor skillnad mellan intern operativa data och Data Warehouse data är skillnaden i hur man använder denna data. Den interna operativa data som finns i de operativa systemen är föremål för ständig förändring skapas, accessas och manipuleras objekt för objekt. Data Warehouse data å andra sidan laddas in i stora mängder och uppdateras sedan inte i vanlig mening. Då man använder data från Data Warehouse så accessar man endast data, man utför alltså ingen förändrande manipulation i datan. Detta medför att data i Data Warehouse är tidsbeständigt.

Skillnader mellan interna data och Data Warehouse data är följande:

- Tids horisonten hos de olika typerna av data. Operationell data brukar ha en tids horisont på 60 – 90 dagar medan Data Warehouse data har en tidshorisont på någonstans mellan 5 – 10 år.
- Operationell data har aktuella värden och talar således om hur nuläget är och denna data kan bli uppdaterad. Data Warehouse data är däremot inget annat en sofistikerad serie av historiska ögonblicks bilder.

- Nyckel strukturen i Data Warehouse data innehåller alltid tid som variabel. Detta är inte fallet med operationell data.

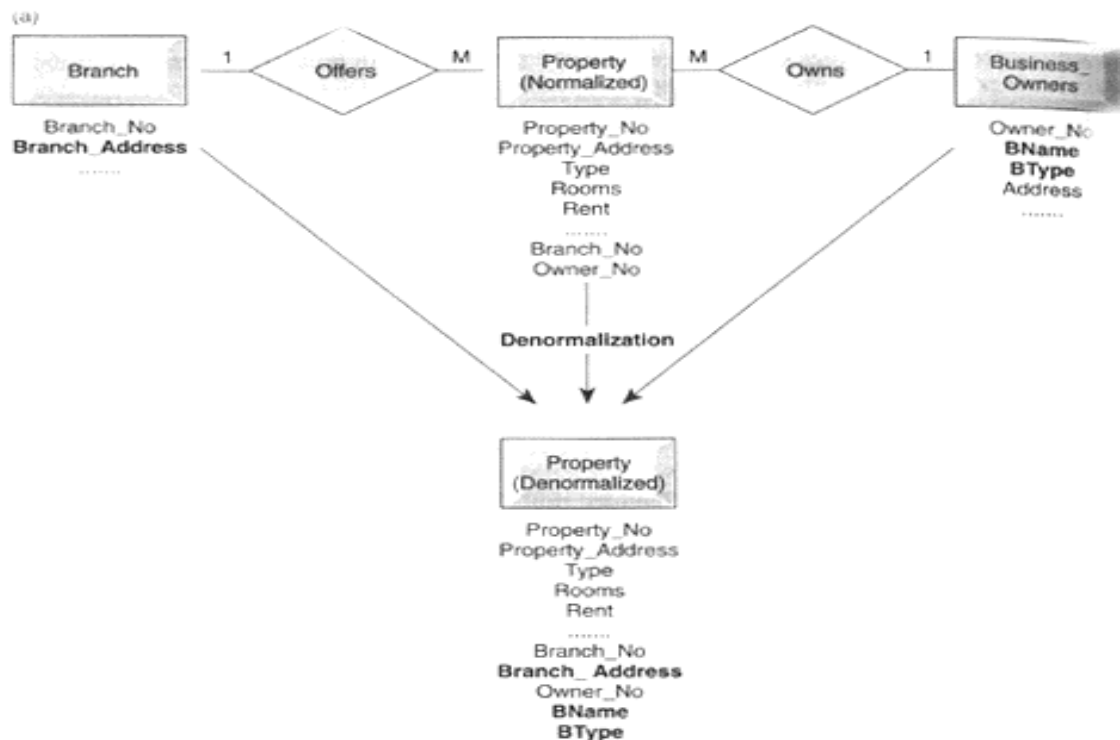


Figur 4:11 Skillnader i användning av databaser.

#### 4.6.2. Design av dimensionstabeller

När faktatabellen har blivit identifierad är det dags för design av dimensionstabeller. Lagringsutrymme som krävs för dessa tabeller är relativt små (< 5 GB) jämfört med faktatabeller. Omstrukturering av dimensionstabeller är inte kostsamma med förutsättning att primär nycklarna i faktatabellen inte har ändrats. Starschema kan användas för att snabba upp frågeprestandan genom denormalisering av referensinformation i en enkel dimensionstabell. Star dimensioner är beroende av mest använda data genom typiska frågor där omfången av frågor är analyseringsfakta genom att ett antal restriktioner appliceras på en enkel dimension. Till exempel property viewing information kanske inkluderar frågor om:

- Typen av egenskaper som visas.
- Avdelningen där egenskaper är registrerade.
- Egenskaper ägda av en speciell affärs ägare.



Figur 4:12 Typisk design av dimensionstabeller, Connolly & Begg(1998).

**Designa faktatabeller.** Först skall man examinera sammansättningar associerade med design av faktatabell i centrum av starschemat. Målet är att producera design som upprätthåller en optimal balans mellan värdet av data som lagras och kostnaden för lagring av denna data. Storleken på faktatabeller kan bli enorma ibland så stor som 1 terabyte( $10^{12}$  bytes). För att nå den optimala databasdesignen måste vi beakta nedanstående fakta.

- Identifiera tidsperioden som krävs för varje beslutssupport applikation.
- Avgöra kravet för statistiska exempel på delmängd av data gentemot kravet för detaljerat data.
- Identifiera onödiga kolumner för att ta bort.
- Reducera kolumnstorleken för faktatabeller.
- Avgöra den bästa användningen av intelligenta- respektive icke- intelligenta främmande nycklar.
- Avgöra den optimala ansatsen för att introducera tid i faktatabeller.
- Dela upp faktatabeller för att hjälpa till med bättre tillsyn.

#### 4.6.3. Migrationsprocedur

När man skall föra data från de operationella systemen till Enterprise Data Warehouse och således omvandla data från operationell data till reconciled data så utförs detta vid i huvudsak två olika typer av tillfällen.

- Första uppladdningen då man fyller Data Warehouse för första gången
- Uppdateringarna av Data Warehouse. Den process där man fyller på med aktuell data och utökar Data Warehouse.

Processen med att omvandla operationell data till data som kan användas av och lagras i Data Warehouse kan delas in i fyra separata delar.

- Capture
- Scrub
- Transform
- Load and index

### Capture

Capture eller insamling som man också kan säga, är den process vid vilken man samlar in relevant data ur de operationella systemen. Man behöver alltså inte samla in all data. Insamlingen som sådan baseras på en grundlig analys av både käll- och målsystemen. Denna analys genomförs med fördel av en gruppering av Data Warehouse specialister och slutanvändare.

### Scrub

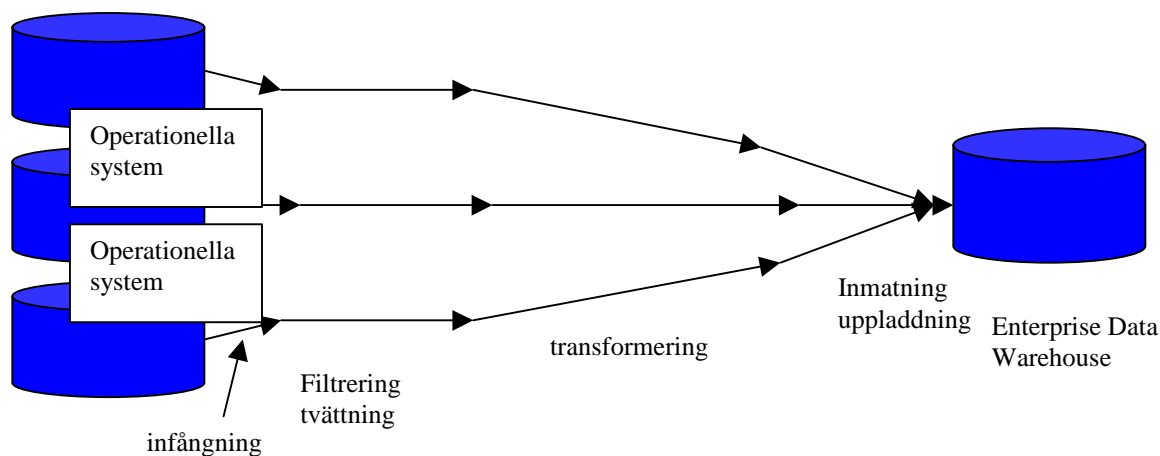
Scrub- processen är den process vid vilken man åtgärdar de fel som ofta finns i de olika operationella systemen. Felen kan vara utav olika karaktär så som t.ex. felstavade namn, saknad data, fält som används till uppgifter de aldrig var avsedda till och icke matchande adresser och postnummer. Målet med scrubben är alltså att rätta till fel.

### Transform

Under transformeringsprocessen så genomför man en konvertering av data formatet från källsystemens olika former till målsystemets form. Man försöker alltså att skapa en bas med data där man eliminerat de olika källsystemens inbördes skillnader. Arbetet utförs på både post och på fält nivå för att göra det möjligt att klara uppgiften.

#### 4.6.4. Absorbering av heterogeniteten

På grund av de stora skillnaderna mellan operationell data från olika källor samt den skillnad som finns mellan operationell data och Enterprise Data Warehouse data så är denna processen den svåraste i byggandet av ett Data Warehouse. Processen är indelad i två delar. Del ett är då Enterprise Data Warehouse fylls med data för första gången. Det andra är då man skall uppdatera desamma.



Figur 4:13

### **Infångning**

Den process som innebär att man samlar in relevant data från de olika operationella systemen kallar vi för infångning. Vanligtvis så behövs inte all den data som finns i de operationella systemen. Det räcker därför att en del av den data som är nödvändig samlas in. Vilken denna del är måste kartläggas genom olika typer av behovs analyser som görs av IT-avdelningen och användarna av det kommande systemet.

Den första gången data samlas in så extraheras den data som är nödvändig. De andra gångerna så tar man bara in den data som är ändrad sedan senaste infångningen. Här använder man sig av databas loggarna som vi beskrev i tidigare avsnitt

### **Filtrering /tvättning**

Under filtreringsprocessen så rättar man till fel som t.ex.

- Felstavade namn
- Omöjliga födelsedata
- Fel använda fält
- Adresser och postnummer som inte matchar
- Förlorad data

Som exempel på namn fel så kan vi ta ett namn som Coca cola som kan stavas på 100 olika sätt enligt McFadden Fred R, Hoffer Jeffery A, Prescott Mary (1998). Det kan stavas enligt ovan eller The Coca cola company eller coca-cola osv. Det är ganska vanligt att data som inte är kritisk har en låg kvalitet. Det spelar ingen roll för ett operativsystem hur man stavar coca-cola men för applikationer som ett Data Warehouse kräver att datan är rätt och riktig och likadan.

Under tvättningsprocessen så använder man sig av program som letar efter mönster och som har även andra artificiella intelligens tekniker inbyggda. Att hålla på att öka kvaliteten på data genom tvättning/filtrering är en tillfällig lösning på problemet med dålig datakvalitet och löser inte problemet på lång sikt. Ett sätt att komma tillrätta med problemet är att öka kvalitetsmedvetandet. Detta kan göras med TQM total quality management(McFadden Fred R, Hoffer Jeffery A, Prescott Mary, 1998).

### **Transformerering**

Själva transformeringen är det centrala i processen att Transformera operationell data till Enterprise Data Warehouse data. Det är i denna process som datan ändras från det format som den hade i det operationella systemet till den form som den har i Enterprise Data Warehouse. Graden av transformation kan variera från att ändra ett dataformat från en form till en annan, till att bli en mycket invecklad övning i att formatera och integrera data. Mer om transformationen kommer nedan i nästa kapitel.

### **Inmatning / uppladdning**

Det sista steget i processen är att ladda upp den data som man valt ut samt att skapa de index som behövs. Uppladdningen sker i huvudsak genom två olika operationer eller angreppssätt dessa är "refresh" samt "update".

- Refresh är ett angreppssätt som används då man initialt skall fylla Warehouset samt om man skall uppdatera hela Warehouset vilket man i så fall gör med jämna mellanrum. I denna operationen så ersätter man alltså befintlig data med helt ny och det finns således inget kvar av tidigare data. Se tabell 1 - 2 ovan
- Update är ett angreppssätt som man använder då man endast skall skriva in de data från de operativa systemen som har ändrats sedan senaste uppdateringen. Vid denna operation så skrivs inte befintlig data över utan kompletteras endast med den nya. Se tabell 4 - 5 ovan.

### **Transformeringsprocessen av data**

Som nämnt ovan så är alltså transformeringen det centrala i processen att föra data från de operativa systemen till Warehouset. Det är vidare viktigt att förstå skillnaden mellan transformeringen och filtreringen av data. Kärnan i filtreringen är att rätta till fel i den data som man skall ladda upp medan transformeringen är den process där man rättar till och ändrar formatet på data från källsystemens- till målsystemens format.

För att transformeringsprocessen skall kunna fortlöpa så är det av yttersta vikt att man filtrerar data innan annars så har man kvar felen i transformerad data efter transformeringsprocessen.

### **Transformeringsprocessfunktioner**

Transformeringsprocessen består av två huvudtyper av funktioner den ena är "record-level" funktioner vilket ungefär kan översättas med en funktion som jobbar med lagrad data i form av filer och tabeller, datahanteringsfunktioner. Den andra är "field lever" funktioner eller fältfunktioner vilket innebär att det är en funktion som konverterar data från en form i källsystemet till en ny form som skall passa målsystemet man kan t.ex. ändra data typer på fält.

### **Datahanteringsfunktioner**

Datahanteringsfunktionerna jobbar som ovan nämnt med tabeller och filer. De viktigaste operationerna är:

- Selektion
- Sammanslagning
- Aggregering
- Normalisering

**Selektion (Select)**

Selektion är en funktion som används för att välja ut den data som är relevant för Data Warehouse ur källsystemen. selektion är även en typisk del i infångningen av data som vi nämnde ovan. Om eftersökt källdata är av relationstyp så kan man använda SQL-fraser för att finna eftersökt data.

En select-sats för utsökning av alla anställda som hade 0:- tillgodo 1999-02-11 kan se ut enligt nedan (appliceras på tabell 2 ovan)

```
SELECT *
FROM ANSTALLD
WHERE Tillgodo = 0 AND Datum>=990211;
```

Denna utsökning resulterar i följande :

9091	Pontus	Svensson	0
------	--------	----------	---

Figur 4:14

**Sammanlagning ( Join)**

Genom att utnyttja sammanlagning så kan man kombinera data från flera olika källor för att presentera de i en gemensam tabell eller skärmbild. Detta är en mycket viktig funktion då det ofta krävs att man kan samla in data från olika källor. Tittar man på t.ex. patient data så finns denna spridd över flertalet databaser och för att kunna skaffa sig en överblick så är sammanlagning nödvändig.

Man kan här liksom i fallet ovan använda SQL om man jobbar med relationsdata. Ofta så förekommer det dock att man jobbar med data som inte är relationsdata och då kan man inte använda SQL vilket försvårar arbetet på så vis att man i detta fallet blir tvungen att koda egna procedurer och funktioner för att lösa uppgiften. Detta gäller även fallet med selektion. Även om man har relationsdata så kan det vara fallet att data kommer från olika källsystem vilket medför att man kan bli tvungen att förändra nycklarna för att kunna göra en join - operation. Det kan också vara fallet att källdata är behäftad med fel som gör att join - operationer blir riskabla. Dessa skall man försöka att åtgärda i tidigare skede.

**Aggregering**

När man aggregerar data så tar man data från en detaljerad nivå och för den till en summerad nivå. Som exempel kan nämnas att man slår ihop vårdavdelningars antal vårdplatser för att skapa en bild av hela sjukhusets antal vårdplatser. Om ett Data Warehouse skall komma att innehålla detaljerad data så utförs inte denna operation i detta skedet men om man vill ha summerad information så kan man använda den.

**Normalisering**

När man normaliserar data då dekomponerar man (bryter ner) relationerna till mindre och mer väl strukturerade relationer. Vanligtvis så är data i operativa system lite normaliserade eller inte normaliserade överhuvudtaget vilket får till följd att man måste göra detta i denna process.



### **Fältfunktioner**

En fältfunktion konverterar data från ett givet källformat till ett nytt målformat som tidigare nämnt. Det finns två typer av fältfunktioner, enkelfält- och multipelfältfunktioner.

Enkelfältfunktioner konverterar data från ett källfält till ett målfält. Detta kan t.ex. vara en konvertering från kronor till dollar eller från meter till tum, dessa kallas för algoritmfunktioner. Det kan även vara konverteringar från förkortningar till fulltext som t.ex. från landskaps bokstav till landskapsnamn. Denna operation kallas för "Tabel look-up" vilket kan översättas tabell kontroll. Alltså att man kontrollerar i en tabell vad som stämmer överens med det som man vill konvertera till.

### **Teknik oberoende**

Vi betraktar konceptet som oberoende av operativsystem, kommunikationssystem, datorutrustning och övrig kommunikationsutrustning.

De tekniska frågorna är intressanta men de omfattas inte i denna studie. Detta bland annat på grund av ANSI/SPARC arkitektur som kräver att tekniken skall vara oberoende av data.

## 5. En kritisk analys av Data Warehouse arkitekturella principer

I detta kapitel så kommer vi att analysera Data Warehouse arkitekturella principer. Utgångspunkten för analysen av begreppet Data Warehouse är de fyra grundläggande principerna som styr byggande av en Data Warehouse arkitektur, nämligen: ett Data Warehouse utgörs av subjekt orienterade, integrerade, historiska och beständiga mängder av data som används som underlag i olika beslutssituationer., (Se Inmon, Connolly et al, McFadden Fred R, Hoffer Jeffery A, Prescott Mary (1998).)

### 5.1. Abstraktionsprincipen

Informationsmanagement i allmänhet och management av information med stöd av en så kallad Data Warehouse arkitektur i synnerhet, förutsätter klarhet och tydlighet om vad data representerar. Därmed låter det rimligt att Data Warehouse konceptet bör omfatta även "metadata" förutom objektdata, men vad menas egentligen med dessa begrepp?.

*'Data Warehouse innehåller såväl "objektdata" som metadata'*

Langefors(1978) säger att information är kunskap som kommuniceras genom språket, det vill säga data. Språket utgör kommunikationens främsta intellektuella verktyg för att kommunicera erfarenheter, idéer, världsbilder, fakta, drömmar, instruktioner, önskemål, m.m. Varje ting, så som människa, artefakt, företeelse har ett namn för att man kunna skall kommunicera uppgifter om händelseutvecklingen. Adjektiva (beskrivande) "etiketer" utgör lämpliga namn för att uttrycka olika typer av egenskaper medan verb "etiketer" används för att uttrycka olika slags förhållande som råder i den sociala verkligheten. Men hur lämpligt är språket för att organisera informationen i t.ex. databaser eller i Data Warehouse? Vi kommer att analysera språkets lämplighet för att organisera informationen utifrån följande premisser:

*'Namn är ett sak och objekt en helt annan sak'*

*'Klargörande av "metadata" kräver meta-metadata'*

Det råder inget tvivel om att relationsmodellen har haft ett stort bidrag till informationsteknologins utveckling. Relationsmodellen är i grunden en så kallad namnbaserad ansats som med sin struktur i princip har slagit bort övriga modeller, såväl den hierarkiska som nätverksmodellen, vilka i sin tur är så kallade datamodeller. Förklaringen till denna utklassning ligger i att relationsmodellen stör språkets "naturlighet" så litet som möjligt. Om verkligheten skulle uppfattas och kommuniceras på ett homogent sätt och om informationen i alla informationssystem var organiserad med stöd av relationsmodellen samt om normaliseringen skulle stanna

på den s.k. 1NF skulle behovet av ett Data Warehouse koncept vara i princip obefintligt. Denna ideal situation finns knappast i verkligheten. Informationssystemen utvecklas av olika personer, vid olika tillfällen, i olika syften och med olika "datamodeller, allt för att tillgodose de bestämda informationsbehov som råder för tillfället. Ju mer distribuerade en organisations informationsresurser är desto större blir behovet av att skapa överblickbarhet. Utan överblickbarhet finns små förutsättningar för IT-management att säkra informationens kvalitet och tillgänglighet. Man kan alltså säga att utan överblickbarhet så försvinner de två nödvändigaste kraven för att en organisations informationsförsörjning skall fungera.

Det var på grund av detta som man etablerade ISO-modellen och därmed objektorienteringsparadigmet. Om man skall kunna kommunicera genom IT-systemen på ett effektivt sätt bör informationen organiseras i termer av objekt och objektstrukturer. Med andra ord så kan man säga att människan måste lämna den konkreta världen, vilken kommuniceras dagligen genom språket och höja abstraktionen. Detta för att gruppera olika slags ting och företeelser. Verkligheten bör uppfattas och kommuniceras i termer av objekt, objektklasser, objektstrukturer, inbördes förhållande mellan objekt, objektens tillstånd och händelser i objektens tillstånd.

Enligt vår tolkning så innebär subjekt orienteringen av Data Warehouse konceptet att den bygger på detta paradigmet. Det innebär att informationen i Data Warehouse är organiserat runt organisationens nyckelobjekt. Detta kan vara bland annat kunder, produkter, leverantörer m.m. En sak är hur denna omgivning organiseras och en helt annan sak är hur Data Warehouse lyckas med att samordna den enorma heterogenitet som råder inom organisationens informationsförsörjning. Det vill säga, hur de inbördes förhållandena mellan Data Warehouse miljön och den operativa omgivningen hänger ihop, samt förhållande mellan Data Warehouse miljön och beslutsmiljön.

Enligt vår tolkning så saknar Data Warehouse konceptet en tydlig förklaring om vilket slags objektorienterings datamodell som tillämpas för att absorbera heterogeniteten. Därmed blir det oklart hur man skall säkra kommunikationen mellan olika informationsmiljöer såväl mellan människor som mellan informationssystemen. Utifrån IT-management perspektiv så gör de intuitiva premisserna och de diffusa deklARATIONERNA att konceptet ter sig nästan oanvändbart och mycket riskfull.

Med andra ord så krävs det att man inför abstraktioner för att öka begripligheten, men det råder stort oklarhet vilka slags abstraktioner som blir gällande inom Data Warehouse miljön. Abstraktionerna syftar till att öka begreppens begriplighet, det vill säga ett klagörande av objektreferenserna och ett tydliggörande av objektens tillstånd, samt förändringar i dessa objekts tillstånd. Utan klara objektreferenser och utan tydliga attributs och associations förhållanden så finns det små förutsättningar för att man skall klara heterogenitetens komplexa och dynamiska natur.

Objektbegreppet är ett besvärligt tillvägagångssätt att organisera informationsförsörjningen på. Utan färdigheter och djupa kunskaper inom ämnet, så kommer uppgiften att bli i princip omöjlig. I en heterogen Data Warehouse miljö så finns olika slags objekt såsom entiteter, associativa objekt (dvs, objektifierade

förhållande), generiska objekt, aggregerade objekt, sammansatta objekt, abstrakta objekt, objektifierade attribut, m.m. Följande premisser bör vara vägledande.

*‘Ingen entitet utan identitet. Ingen identifiering utan entiteter’*

Det innebär att utan att klargöra objektens identitet så saknar vi förutsättningar för en entydig referens till verklighetens händelseutveckling som representeras av informationen. Därmed saknar vi möjligheter att förstå hur nyckelbegreppet är utformat och vad det egentligen representerar.

Sammanfattningsvis så är vi inte säkra på hur Data Warehouse konceptet klarar av att hantera detta problem. Vi är inte heller säkra på hur så kallad "metadata" hanteras i en sådan miljö. Metadata är "data" om data. Men den tolkning som ges i litteraturen av Data Warehouse konceptet är enligt oss bristfällig. En så kallad metadatabas bör enligt vår uppfattning klargöra de inbördes förhållandena mellan de datamodeller som har varit inblandade i systematiseringen av informationen, informationsflödena, informationsbehandling m.m. Någon sådan information har vi inte hittat i litteraturen. På detta sätt kan man resonera att Data Warehouse konceptet kan utgöra en farlig utvecklingsmetod på grund av de oklara, otydliga, intuitiva förklaringar som dokumenterar Data Warehouse konceptets tillämpbarhet.

Sammanfattningsvis kan sägas att abstraktionsprincipen förväntas säkra informationsförsörjningens entydighet, tydlighet, kommunicerbarhet och begriplighet. Vi är inte övertygande om att någon av dessa önskade egenskaper kan tillgodoses av Data Warehouse konceptet. Nästa del i vår analys är den del om integrationsprincipen.

## 5.2. Integrationsprincipen

Data Warehouse konceptet grundas på en enligt oss otydlig integrationsprincip, detta anser vi på grund av att det har olika och motsägelsefulla innebörder.

Enligt vår uppfattning saknar begreppet integration en entydigt tolkning. Integrationsbegreppet förekommer såsom (1) arkitekturell integration, (2) logisk integration av fysiskt partitionerade data, (3) integration av normaliserade data, samt (4) integrationsfilosofi för förvaltning av historiska, aktuella och framtidsorienterade data. I detta avsnitt kommer vi att analysera, kritisera och precisera principens egentliga innebörd.

**Arkitekturell integration.** Denna form av integration utgör konceptets förklaringsargument och härleds från Data Warehouse konceptets ultimata målbild, det vill säga:

- Överblickbarhetsbehovet, dvs. önskan om en bredare överblick av organisationens informationsförsörjning. Med andra ord, att ha bättre kontroll över informationens kvalitet
- Informationstillgänglighets krav, det vill säga att göra informationen tillgänglig genom att länka samman verksamhets operativa system och beslutstödssystem för att främja beslutsfattandet
- Krav om systemoberoende, dvs. länkningen mellan de berörda systemen skall inte

orsaka något beroende. De operativa systemens strukturer skall inte påverkas av integrationen.

### *'Data Warehouse innehåller integrerade data'*

Överblickbarhetsbehovet kan förklaras i termer datakvalitet. Data i operationella och verksamhetsnära system är ofta av dålig kvalitet. Detta till följd av att systemen är skapade vid olika tidpunkter och i olika syften. Det är därför inte ovanligt att man stöter på olika grader av heterogenitet databaserna emellan. Det som man kan stöta på är t.ex.:

- Begreppsmässig heterogenitet
- Innehållsmässig heterogenitet
- Regelbaserad heterogenitet
- Teknologibaserad heterogenitet

Denna heterogenitet gör att det är mycket svårt att skapa sig en överblick över organisationen. Arbetet med att komma tillrätta med dessa problemen kan vara mycket stort och tids krävande. Detta på grund av det vanligt vis stora antalet filer och ännu större antalet poster i dessa filer. även om arbetet är stort så är det av yttersta vikt att man gör detta för annars så kan man inte använda denna data i ett Data Warehouse.

**Logisk integration av vertikalt och horisontellt partitionerade data.** Behovet av den logiska integrationen förekommer på grund av att data i Data Warehouse är såväl horisontellt som vertikalt partitionerade.

*Vertikal integration.* Denna form av integration utgör grunden i Data Warehouse - konceptet. Informationen skall vara i sin elementära (odelbara form). Den vertikala integrationen utgör en logisk form som förekommer i distribuerade databaser. Det innebär att integrerade datamängder måste kännetecknas av olika former (intention) men de måste ha samma referenser (extension). Därmed utgör integrationen en logisk operation som kan användas för att skapa olika former av mer sammansatta datamängder.

*Horisontell integration.* Denna form av integration förekommer på grund av datavolymens omfång. Enligt vår tolkning kan historiska data partitioneras fysiskt på ett horisontellt sätt, med andra ord på samma sätt som förekommer i distribuerade databaser. Den enda skillnaden är att i en Data Warehouse miljö finns all data på ett ställe. På detta sätt kan man hålla responstiderna nere.

**Integration av normaliserade data.** Enligt teorin så innehåller Data Warehouse konceptet inte normaliserad data. Samtidigt förekommer det mycket klart att partitionering i faktainformation och referensinformation utgör samma form av integration som förekommer i relationsmodellen och som förverkligas med hjälp av främmande nycklar (dvs referensinformation). Som det framgår av vår tolkning här finns en motsägelse mellan att hålla data i en elementär form och att hålla data i en normaliserad form.

**Integrationsfilosofi vid uppdatering av data.** Denna form av integration förekommer för att klargöra de förhållanden som råder mellan data och tiden. Beroende av hur data integreras i systemen kan man identifiera två slags integrationsfilosofier. Den första integrationsfilosofin kan bara ge information om det aktuella tillståndet (statusdata) medan den andra integrationsfilosofin kan ge information om så väl historiska data som data om den framtida händelseutvecklingen. Med andra ord så beskriver statusdata det aktuella läget på ett objekt som t.ex. ett konto.

En annan tolkning av aktuella data förekommer i termer av transistenta data. Frågan om vad man menar med aktuella data blir därmed oklart. T.ex. uppgifter om konto saldo kan förekomma som härledbar information eller som transistent information.

Sammanfattningsvis kan man säga att integrationsprincipen förväntas säkra informationsförsörjningens överblickbarhet, tillgänglighet, redudansfrihet, härledbarhet, kommunicerbarhet, lätt "navigering" i tid och rum, etc. Vi är inte övertygade om att alla dessa önskade egenskaper kan tillgodoses av Data Warehouse konceptet.

### **5.3. Aggregeringsprincipen**

Den tredje delen i vår analys av de olika principerna är den som berör aggregering. Aggregeringsprincipen utgör grunden för Data Warehouse konceptet. Med andra ord kan man säga att, utan aggregering inget Data Warehouse. Men den tillgängliga litteraturen ger inga uppgifter om hur aggregeringsproblemen skall lösas. Det finns endast knapphändiga uppgifter som skall tydliggöra innebörden av aggregerade objekt, byggandet av aggregerade strukturer, klargörandet av identitet och identifiering av dessa objekt, förhållande av kollektiva respektive enkla attribut som associeras med aggregerade objekt, etc. Vad som finns är premissen:

*'Data Warehouse innehåller såväl detaljerade som aggregerade data'*

Aggregering handlar mycket om härledbarhetsbehov som förekommer på grund av de vertikala förhållande som råder mellan olika informationssystem. På grund av härledbarhetens komplexitet blir det enklare om informationssystemet förvaltar även härledbar information. Detta önskemål ställer krav för införande av såväl aggregerade objekt som aggregerade hierarkiska strukturer. Tiden ( som vi kommer att diskutera senare) ställer ytterligare krav för mer komplicerade aggregeringsstrukturer sådana som förekommer t.ex. i JSD ansatsen och som används för att representera de aggregerade objektens händelseutveckling.

Situationen försvåras ytterligare i de fall, och dessa är många, där de aggregerade objektens händelseutveckling, som klargörs genom härledbarheten, skall jämföras med objektens önskade utveckling. Redan 1977 exemplifierade B. Langefors svårigheterna. T.ex. det är en vanlig företeelse att ett företag säljer sina produkter i en segmenterad och differentierad marknad. Marknaden kan bestå av flera regioner, som i sin tur kan bestå av flera distrikt, som i sin tur kan omfatta flera försäljningsenheter. etc. Denna situation illustrerar hur komplicerat aggregeringsproblemet blir, om det

blir frågan om att organisera informationen om respektive objekts egenskaper, i termer av försäljningsmål, efterfrågan, den verkliga försäljningen, den potentiella försäljningen, prognos information m.m.

En helt annan företeelse är den så kallade produktstrukturen, vilken i grunden består av aggregerade objekt. Vi tror knappast att man utan att ha en väl utformad aggregerad struktur kan realisera ett JIT-konceptet eller IOS-konceptet. Men vilken vägledning får vi av Data Warehouse konceptet?

### *‘Utan aggregering inget Data Warehouse’*

Den grundläggande skillnaden mellan Relationsmodellen och Data Warehouse modellen ligger just i aggregering. Relationsmodellens elegans försvinner i situationer som kräver systematisering av information som refererar till aggregerade objekt. Varken de platta tabellerna eller SQL-konstruktioner kan troligtvis absorbera den komplexitet som relateras till aggregerade strukturer. Komplicerade aggregeringsstrukturer kräver avancerade mekanismer som möjliggör sökningar i hierarkiernas djup eller bred. AI (artificiell Intelligence) blev känd som en disciplin, inte på grund av dess avancerade mekanismer att representera komplexa matematiska och logiska problem utan istället på grund av den klarar av att representera och hantera komplexa produktstrukturer och marknadsstrukturer.

Sammanfattningsvis kan sägas att aggregeringsprincipen förväntas säkra informationsförsörjningens härledbarhet, jämförbarhet, kommunicerbarhet och begriplighet. Vi är inte helt övertygade om att någon av dessa önskade egenskaper kan tillgodoses av Data Warehouse konceptet.

### **5.4.Evolutionsprincipen**

Den fjärde och sista av principerna är evolutionsprincipen. Enligt teorin så förväntas Data Warehouse konceptet organisera informationen om de olika objektens historiska, aktuella och framtida tillstånd. Det vill säga i ett Data Warehouse skall det finnas uppgifter tillgängliga om objektens uppkomst, tillståndsförändringar, och "upplösning".

### *‘Data Warehouse innehåller såväl aktuella som historiska data’*

Det som saknas i detta resonemang är begrepp, principer och tekniker för att representera principens bakomliggande syften. Det finns knappast någon indikation om tiden och hur tiden representeras i en så kallad Data Warehouse miljö. Janis Bubenko(1985) säger t.ex. att utan ett tydligt och begripligt tidssystem har vi knappast någon chans att klara representationsproblemet. Men hur viktigt är detta påstående?

- Händelser och deras inbördesförhållande kräver klara tidsreferenser annars finns det knappast någon förutsättning att härleda information om t.ex. olika marknads trender.

- Uppgifter om objektens framtida tillstånd kräver ett mer komplicerat tidssystem eftersom vi måste referera till två eller fler olika tidsreferenser samtidigt. Antag till exempel att idag lämnar vi prisuppgifter till systemet som skall bli gällande nästa vecka. Systemet måste klara av att hantera såväl mottagningstiden som giltighetstiden.
- En verksamhet som är distribuerad i det fysiska rummet förväntar sig att olika informationssystem kan representera samma företeelse och samma händelseutveckling med helt olika tidsresolutioner. Hur klarar man då att integrera informationen? Hur klarar man att härleda eller jämföra informationen från flera olika system?
- Klargörande av händelser innebär även klargörande om vilken information som är basinformation (lägsta detaljeringsnivå) och vilken som är härledbar information. Det innebär vidare ett klargörande av vilka valideringsregler som har kontrollerat informationens korrekthet. Men hur klarar vi en sådan uppgift utan vägledning?
- Jämförbarhet innebär bland annat att kraven för kompatibilitet i rum och tid är uppfyllt. Objektens verkliga och önskade tillstånd kan användas för att illustrera problemets natur. Utan klara tidsreferenser och en klar uppfattning om händelser och deras inbördesförhållande finns det knappast någon chans att använda sig av ett Data Warehouse koncept.

Sammanfattningsvis kan man säga att evolutionsprincipen förväntas säkra informationsförsörjnings korrekthet, jämförbarhet, kompatibilitet och validitet m.m. Vi är inte övertygade om att någon av dessa önskade egenskaper kan tillgodoses av Data Warehouse konceptet.

### **5.5.Sammanfattning**

I detta avsnitt har vi granskat Data Warehouse konceptet utifrån ett informationsteknologiskt perspektiv. Granskningen har gjorts med stöd av J. Bubenkos(1977) modell om validering och verifiering av informationsförsörjning. Bubenkos modell har samma dimensioner som Inmons modell. Det vill säga båda modeller refererar till (1) abstraktionsprincipen, (2) aggregeringsprincipen, (3) integrationsprincipen samt (4) evolutionsprincipen. I denna mening är modellerna jämförbara. Den stora skillnaden mellan dessa två modellerna, vilka båda i princip angriper heterogenitetsproblemet är att Bubenkos modell(1977) presenterar en professionell kritik medan Inmon et al(1996) har presenterat en intuitiv vision som saknar såväl teoretiskt och empiriskt stöd. Sammanfattningsvis kan man enligt vår uppfattning säga att:

- Data Warehouse modellen utgör snarare en vision än en verklighet. Konceptet saknar de förutsättningar som behövs för att hantera heterogeniteten på ett sunt och riskfritt sätt.
- Data Warehouse modellen saknar en så kallad datamodell vilken skall kunna säkra frågan om hur man "länkar" och förvaltar heterogena informationsdomäner som utgör beståndsdelar av en och samma informationsmiljö.
- Data Warehouse modellen är full av motsägelser, oklarheter och "reklam". Den utgör därmed en dåligt vägledningsgrund för att ge ett entydigt svar på heterogenitetsproblemet.



## 6. Resultat

I denna resultatdel så kommer vi att presentera de olika resultaten som vi kommit fram till under vårt arbete. Detta är bland annat två sammanställande bilder över heterogenitet. Vi presenterar även en del där vi redogör för konceptets förmåga att hantera heterogenitet samt en del där vi redogör för dess användbarhet. Resultatet av vår lilla fältstudie presenteras även här.

Vi har under vårt arbete med att kartlägga heterogenitetsproblemet skapat en sammanfattande modell av de heterogeniteterna vilka vi kommit fram till samt över de olika personer och koncept som hanterar dessa och vilka de hanterar.

Info. system Verksamhets- system	Begreppsmässig Heterogenitet	Info.mässig Heterogenitet	Regelmässig Heterogenitet	Filosofisk Heterogenitet
Funktionell Heterogenitet	Wetherbe Simon Putnam Hugoson	Wetherbe Simon Putnam -----	----- Simon Putnam -----	Zachmans - Resurs/Process perspektiv - Tidsresolution
Strukturell Heterogenitet	Wetherbe Huwitt Anthony C.D Kall	Wetherbe Huwitt Anthony	----- Hewitt -----	- Globalisering av objekt är orimlig i dynamiska miljöer - Tvetydighet
Kulturell Heterogenitet	Wetherbe Langefors Argyris CSF	Wetherbe Langefors Argyris	----- Langefors Argyris -----	- Harmonisering av målinfo. - Olika världsbilder - Jämförbarhet
Kognitiv Heterogenitet	Wetherbe Simon Langefors	Wetherbe Langefors Ackoff	Simon Hedberg Ackoff	- Verksamhets- föreställningar - Lokala föreställningar
Informations- teknologisk Heterogenitet	ISO-82 UML EDI-FACT	ANSI/SPARC ----- -----	ISO-82 Bubenko -----	- Tidsdimension - Härledbarhet - Integration - Objekt/Namn
Informations- försörjnings- arkitektur	Informations utbytesarkitektur	Informations arkitektur	Informations- behandlings- arkitektur	

Figur 6:1

Som synes i figur 6:1 så är området mycket komplext och genom detta så blir det även svår överblickbart. Området och problemet som sådant är dock varken nytt eller

utforskat. Problemet har sysselsatt forskare under flera år och genom detta så har det kommit fram flertalet koncept som syftar till att lösa problemet. Framgången på de olika har dock varit varierande. Modellen som vi presenterat kan tjäna som en vägledningsmodell i arbetet med att synliggöra och hantera de olika heterogenitetsproblemen som aktualiseras vid systemintegrationer av olika slag.

Vi har även gjort en sammanställande bild över de grundläggande heterogeniteterna. Vi har här även försökt att kombinera ihop de olika typer av arkitekturansatser som finns för att lösa heterogenitetsproblemen, med de olika heterogeniteterna. Vi vill med denna bild även sätta in människan, detta då i princip alla system har mänskliga användare och utvecklare. Detta leder till att om man inte tar hänsyn till den mänskliga aspekten så riskerar man enligt oss få problem med acceptansen av olika integrationslösningar, vilket i sin tur verkar menligt på organisationen.

Info.sys arkitektur Social arkitektur	Begreppsmässig heterogenitet	Innehållsmässig heterogenitet	Regelmässig heterogenitet	
Människa och funktionell heterogenitet				Funktionell arkitektur
Människa och strukturell heterogenitet				Strukturell arkitektur
Människa och kulturell/ kognitiv heterogenitet				Infologisk arkitektur
	Informations- utbytesarkitektur	Informations arkitektur	Informations- behandlings- arkitektur	Makro arkitektur

Figur 6:2

### 6.1.Heterogenitet och Data Warehouse

Data Warehouse kan enligt vår tolkning vara en hyfsad strategi för att möjliggöra en ökning av tillgängligheten av information inom en organisation, vilken präglas av aktivitets-, besluts-, resurs-, miljö- och målberoende. Ju större dessa beroenden är desto större blir behovet av samordning och där med kommunikation. Då organisationers informationsförsörjning oftast är differentierad så beror detta på att det är flera olika system som tillsammans skall tillgodose såväl informationsmässiga som behandlingsmässiga krav. De etablerade strategierna angriper problemet på design nivå och utifrån IT-Management perspektiv. Designmässiga förändringar

resulterar alltid i att man blir tvungen att utföra funktionella, strukturella och infologiska beslut. Data Warehouse strategin erbjuder däremot en filosofi som accepterar organisationens informationsarkitektur. Samtidigt så skapar den förutsättningar för att öka informationens tillgänglighet över horisontella och vertikala gränser. Enligt vår tolkning av konceptet så kan Data Warehouse angripa följande slags situationer:

- Enkla begreppsmässiga konflikter ( i princip synonymitet)
- Enkla stilistiska skillnader i språket (förkortningar)
- Enkla strukturella konflikter (entitet och egenskaper)

Vi har dock svårt att föreställa oss att konceptets förmåga att förutsättningslöst angripa heterogeniteten om man glömmer de sociala aspekterna som finns representerade i systemen, som till exempel ägarförhållanden.

Det är viktigt i det här sammanhanget att även nämna bland annat Goldkuhls(1986) definition om informationssystem. Han ser informationssystem som sociala konstruktioner som delvis är realiserade med stöd av informationsteknologi. Detta innebär alltså att en viss del av informationsförsörjningen utförs av människor och att varje försök att förändra och homogenisera informationen ökar distansen mellan de tekniska lösningarna och individens föreställnings värld. Vidare kan det vara passande i sammanhanget att nämna Inmons strategi för att angripa heterogeniteten vilken kan anges i fyra principer enligt nedan:

- Integrationsprincip (normaliserad kontra icke normaliserad data)
- Härledbarhetsprincip (härledbara och icke härledbara data)
- Tidsuppfattnings princip (aktuell och historisk data)
- Den filosofiska principen (Skillnaden mellan namn och objekt m.a.o. mellan data om objekt och metadata)

Enligt oss så förefaller det som om Bubenko tog upp ämnet med heterogenitet redan 1977. Bubenko utgår helt och hållet utifrån samma dimensioner för att diskutera hur svårt det är att hantera konceptualiseringsproblemen. Han menar att i det princip är endast genom förhandlingar mellan intressenter som kan skapa en strategi som absorberar heterogeniteten och möjliggör harmonisering av informationsförsörjningen. Inmon hanterar alltså heterogeniteten utifrån ett ”information-engineering” perspektiv medan Bubenko ser det som en IT- Management fråga. Detta gör Bubenko eftersom arbetet kräver förhandlingar. Vad är det då som stödjer Bubenkos argumentation?

- Det finns ingen designteori eller metod som kan sägas vara rätt eller fel. Varje modell är främst beroende av den sociala målbilden som ska behandlas. Detta får till följd att det finns varken objektiva eller vetenskapliga grunder för att klargöra modellens fruktbarhet eller lämplighet.
- I situationer där alla använder samma designteori, modell (datamodell) så är det mycket troligt att människornas olikheter i betraktelse och uppfattning av verkligheten kommer att bli stora, på grund av deras begrepps- och erfarenhetsmässiga skillnader. Frågan är hur dessa individuella konceptuella

modeller kan samordnas för att skapa en konceptuell bas för behandling av information, utifrån verksamhetens perspektiv.

- Det finns, enligt oss, ingen möjlighet att på ett formellt sätt klargöra de konceptuella olikheterna som var och en av de modeller som beskriver verkligheten har, t.ex. normalisering förutsätter att de semantiska problemen är lösta innan den genomförs.
- Man kan säga att de semantiska konflikterna har sina rötter i sociala, organisatoriska och funktionella olikheter. Dessa kräver alltså politik och förhandling samt en intuition och en medvetenhet i designen.

En utgångspunkt som liknar den som Inmon presenterar finns i BSP-metoden och utifrån detta perspektiv så kan vi se att här finns det finns även målinformation. Detta tar enligt oss inte Inmon någon hänsyn till. Men hur skall vi få klarhet i denna information som kan vara explicit i tabellform eller inprogrammerade regler.

## **6.2. Hur användbart är Data Warehouse konceptet för att hantera heterogenitet?**

Data Warehouse är en bra plattform för att göra data tillgänglig för hela organisationen samtidigt som man undviker förändringar i organisationens funktionella system. Man kan säga att Data Warehouse blir som ett paraply som täcker hela eller delar av organisationen. Man bör dock vara försiktig med en generalisering av konceptet då det inte finns några garantier för att det alltid är lämpligt, paraply bär man t.ex. inte inomhus men det är bra när det regnar ute.

Data Warehouse och dess nytta måste klargöras utifrån den målbild som skall betjänas och hur denna är beskaffad. Med andra ord så utgör Data Warehouse en bra ingenjörslösning, men det är inte givet att denna lösningen är en bra organisatorisk lösning. T.ex. i Data Warehouse litteraturen förekommer begreppet Data Warehouse arkitektur men vi har inte hittat något om kritiska problem eller vilka balanseringsmekanismer som finns för att hantera heterogeniteten. Om man tittar på biblioteksväsendet så kan man se att endast nödvändiga data har homogeniserats och att man låter en viss del heterogenitet leva kvar mellan de olika bibliotek systemen. Detta sätt att komma tillrätta med heterogeniteten kan i många fall vara en bra lösning. Detta då man minimerar kravet på förhandlingar mellan olika parter. Om det är färre begrepp och regler m.m. som skall jämkas så kommer arbetet att bli smidigare.

**Sammanfattningsvis** kan man säga att ju mer homogenisering som eftersträvas desto mindre förväntas handlingsfriheten och tankefriheten bli. Ju större homogenisering som eftersträvas desto större kommer den kognitiva distansen att bli och genom detta så ökar sannolikheten för felaktiga handlingar och felaktiga eller osammanhängande beslut. Den infologiska distansen riskerar också att öka på grund av att Data Warehouse konceptet inte ger några referenser till målinformation. Med andra ord till de grundläggande förutsättningarna för korrekt tolkning av information.

Organisationer kan ses som informationssystem på grund av att det som håller samman individerna är målinformationen, som Langefors kallar för systeminformation. Målinformationen etableras genom förhandlingar och uttrycks genom direkt kommunikation mellan intressenterna. Denna processen är sällan

datoriserad utan sköts man till man så att säga. Man kan med andra ord säga att om Data Warehouse accepteras som en teknisklösning på grund av dess förutsättningar att hantera tillgänglighets och kvalitets problem på ett ekonomiskt godtagbart sätt, då måste det få godkännande även av den sociala verkligheten som kommer att påverkas av och påverka Data Warehouse's framgång. Vi vill ansluta R. Bolands(1979, 1995), P. Checklans(1985) filosofi om arkitekturell balans som innebär att en bedömning av en lösning bör avgöras utifrån såväl teknologiska som sociala krav.

Slutligen kan vi tillägga att vi inte anser att ett ensidigt ingenjörsmässigt underlag räcker till för att fatta ett riktigt IT-Managementbeslut, med den påverkan som detta får på den sociala verkligheten hos aktörerna.

### 6.3.En studie av två systems information

Vårt tanke med denna studie har varit att undersöka två synbarligen likvärdiga system för att vidare kunna skapa oss en lämplig bild av förekommande likheter och olikheter mellan dessa system. Vårt huvudsyfte med denna studie har varit att påvisa likheter och olikheterna mellan olika system.

Vårt arbetssätt kan illustreras enligt följande:

För att på ett bra sätt kunna göra en kartläggning på två olika system som har samma målgrupp, användningsområde och yttligt sett samma information så har vi valt att titta på två bibliotekssystem. Något av det första som slår en är att systemen är förvånansvärt annorlunda i sin gränssnitts utformning. Förvisso är båda textbaserade men likheterna stannar i princip vid detta. För att prova funktionerna i systemen så gjorde vi utsökningar i båda systemen. För att vidare kunna påvisa skillnader så sökte vi efter samma böcker i de båda systemen. Vi gjorde även utsökningarna efter samma mönster i de båda systemen. Vi började med titelsökning för att sedan göra en sökning på ämnes ord.

**System A** som var det systemet vi först tittade, på ger ett något förvirrande intryck samt en känsla av att sakna något. Systemet är uppbyggt på så vis att man kan söka på titel, författare och ämnesord. För att kunna göra en sökning på ämnesord så måste man ha en katalog att söka efter ämnesorden i. Katalogen är inte datorbaserad utan förekommer i ett vanligt bokformat. Ämnesordskatalogen är uppdelad på två volymer, en systematisk- och en alfabetisk del.

**System B** vilket var det andra systemet som vi tittade på gav ett renare och mer logiskt intryck. Här saknades även de externa ämnesordskatalogerna. Vidare så hade man ett användarvänligare gränssnitt med alla valalternativen snyggt och prydligt uppgradade. I detta systemet så kan man söka på följande:

- 1 Författare + titel
- 2 Titel
- 3 Författare
- 4 Titel + ämnesord
- 5 Ämnesord a-ö
- 6 Författare a-ö
- 7 Boktitlar
- 8 Tidskrifter

- 9 Konferenser
- 10 Boolesk sökning
- 15 ISBN/ISSN nummer

Numren framför alternativen är den siffra som man använder för att kunna göra den specifika sökningen, system A sköts på liknande sätt, det är dock inte lika lättöverskådligt.

En skillnad som vi mötte på mellan systemen är den att man använder samma ord fast för olika saker alltså en skillnad i begrepp. Det som man kallar för sökning på ämnesord i system A är inte detsamma i system B. I system A så har man ett mycket krångligt system för ämnesord. Hela ämnesordssystemet baseras på två kataloger som nämnt i tidigare stycke. I system B så har man löst problemet på ett annat sätt och här kan man söka på ett sätt som liknar fritext sökning i ämnesordssökningen. Begreppet ämnesord skiljer sig alltså åt mellan de båda systemen.

När vi skulle genomföra vår undersökning så gick vi tillväga på så vis att vi plockade ut ett antal böcker ur hyllorna, titlarna finns i figur 6:2. Dessa böcker försökte vi sedan att finna i bibliotekssystemen. De böckerna som vi plockade ut valdes ut i bibliotek A, där efter så sökte vi efter dessa i de båda systemen.

Vi började vår undersökning genom att göra en ämnesordssökning i system A. Här valde vi att söka efter information om projektstyrning, folkmord samt böcker om tobak. Den systematiska delen av ämnesords katalogerna i system A klarade vi inte av att använda då denna har en klart förvirrande uppbyggnad och verkar vara skapad för proffs. Det visade sig i det närmaste vara omöjligt att finna vilket ämnesord som man skulle använda i sökningen. Detta i sin tur medförde att det blev i praktiken omöjligt att genomföra en sökning på ämnesord med hjälp av denna katalog. Vi gjorde i alla fall ett försök och sökte efter ”projektstyrning”. Detta hittade vi inte alls. Vi gjorde därför ett nytt försök och sökte på ”folkmord”. Här fann vi ämnesordet Oepa under kategorin ”samhälls- och rättsvetenskap”. De träffar som vi fick var följande:

- 1 Bok om LSD ”Acid the sectet history of lsd”.
- 2 Bok på franska (förstår inte titeln så vi vet inte om detta var en träff)
- 3 ”skyldig eller oskyldig ” en bok om ett rätts fall i England
- Sedan följer diverse böcker om tumör patologi.

Totalt så fick vi vid denna sökningen 34 träffar varav endast en bok var inom det område som vi sökt (Oepa) nämligen den om LSD. Alla andra träffar var inom andra ämnesordskategorier.

Vi gjorde ett försök att söka efter ”tobak ” i den systematiska delen. Detta fann vi förvånande nog under ”teknik, industri och kommunikationer”. Ämnesordet vi fick fram var Pmbab. När vi gjorde sökningen på detta ämnesord så fick vi inga relevanta träffar, resultatet var följande:

- Plötslig spädbarnsdöd
- Pmbb. 05-C:k.4
- Pneumococcial infektion : drug therapy.

Som synes så är träffarna helt meningslösa och i vissa fall helt förvirrande. Boken om plötslig spädbarns död kan tyckas vara relevant men hade ett annat ämnesord, sålunda var även denna en miss.

Då vi använde den alfabetiska delen så hittade vi ibland de ämnes ord som vi sökte och ibland inte. Vi valde att söka på böcker om projektstyrning i den alfabetiska delen här fick vi ingen träff som ledde till ett ämnes ord. Vi fann dock meningen ”projektarbete i driftsorganisationer ” vilket gav oss ämnesordet Qbiaa. Då vi genomförde sökningen så hittade vi inga relaterade böcker överhuvudtaget. De träffar som vi fick spände över området från ”BASIC” (programmeringsspråk) till ”människa rättigheter i europeisk praxis”. Ingen relevant träff alltså. Alla träffar var på andra ämnesord än det som vi hade skrivit in alltså. Det är mycket förvånande att man skapar ett system som redovisar träffar på felaktiga saker.

I system B så har de ett system med ämnesord som skiljer sig markant från system A som nämnt. I system B så kan man välja själv vilket ord man vill söka på, alltså en typ av fritextsökning. Då vi sökte på folkmord så fick vi noll träffar i system B. Här presenterade alltså systemet ingen orelevant information om andra böcker vilket system A gjorde.

	System A	System B	Träff A = B
Sökt ord 1	Projektstyrning	Projektstyrning	
Ämnesord 1	Qbiaa	Projektstyrning	
Träffar	Ett 20 tal icke relaterade	12 relaterade böcker	Nej
Sökt ord 2	Tobak	Tobak	
Ämnesord 2	Pmbab	Tobak	
Träffar	Ett 20 tal icke relaterade	2 relaterade böcker	Nej

Figur 6:3 Sökning med ämnesord

Titel sökningen som vi gjorde på de båda systemen utfördes först på bibliotek A, vilket var det bibliotek där vi plockade ut böckerna ur hyllorna. Detta tillvägagångssättet valdes för att vi skulle kunna påvisa skillnader i vilken information som de båda systemen hade lagrat i sina informationssystem. Genom detta tillvägagångssätt så var vi garanterade att åtminstone ett av biblioteken skulle ha böckerna. I figur 6:2 så har vi gjort en översikt över de titlar som vi sökt efter i de båda systemen. I system A fick vi naturligtvis träff på alla titelsökningarna. Då vi gjorde utsökningarna i system B så märkte vi vid en jämförelse de båda systemen emellan att de bådas informationsbaser inte var likadana.

Vi fann alltså en skillnad i begrepp och en skillnad i information mellan de båda bibliotekssystemen. Ut över detta så fann vi en stor skillnad i gränssnitt de båda systemen emellan. Dessa skillnader är exempel på faktorer som kan försvåra tillgängligheten och minska överblickbarheten av information.

Titel	System A	System B	Träff A = B
Familjepension	3 Träffar	0 Träffar	Nej
Objektorienterad systemutveckling	2 Träffar	2 Träffar	Nej
Effektiv räddningstjänst	1 Träff	0 Träffar	Nej
Inrikesflyget under 1980-talet	1 Träff	1 Träff	Ja
Målinriktad projektstyrning	1 Träff	0 Exakta träffar 1 relaterad	Nej

Figur 6:4 Titelsökning



## 7. Diskussion

I detta kapitel så kommer vi att diskutera kring våra resultat som vi kommit fram till under vår studie. Vi kommer först att diskutera kring de olika typerna av heterogenitet. Detta följs upp av en del med diskussion kring Data Warehouse och dess lämplighet. Vi utvärderar även vårt arbete i denna del och ger förslag på vidare studier.

**Den kulturella heterogeniteten** som vi tog upp med dess ingående delar såsom språk, perspektiv och verklighetsuppfattningar skapar heterogenitet mellan de olika systemen i en organisation. Skillnaderna kan yttra sig på många olika sätt såsom regler, innehåll och begrepp som finns i respektive system. När man skapar separata system på olika avdelningar inom organisationer så kommer lösningarna att präglas av den miljö som de skapas i.

**Även den strukturella heterogeniteten** påverkar utvecklingen av system. Den strukturella heterogeniteten är beroende av den grad av frihet som råder vid utvecklandet av system. Saknas det övergripande regler och förordningar så kommer det med stor säkerhet att bli stor skillnad mellan systemen. Resultatet av fria tyglar och få regler i en utvecklings process leder enligt oss fram till att systemen kommer att få en hög grad av språklig och begreppsmässig heterogenitet. Som vi upplever det så finns det en klar koppling mellan individens handlingsfrihet och den strukturella heterogeniteten.

**Den funktionell heterogeniteten** kommer man enligt oss även att få i hög grad inom organisationen om man har en hög grad av kulturell och strukturell heterogenitet. Detta i sin tur påverkar utvecklingen av system på så vis att det kommer att finnas en skillnad mellan de olika delarna ifråga om vad ett system skall göra och klara av. En organisation som präglas av en stor funktionell heterogenitet kan sägas klara av att möta omgivningens varierande krav bättre än om den hade varit homogen. Värt att notera är dock att om man har en för stor funktionell heterogenitet så riskerar man att få problem med styrningen av och samordningen av processer och funktioner inom organisationer. Det gäller alltså enligt oss att man försöker att finna någon form av balans så att det blir möjligt att möta varierande krav samtidigt som t.ex. styrningen inte blir lidande. Hur balansakten skall gå till vill vi låta vara osagt. En stor funktionell heterogenitet leder enligt oss till att det utvecklas olika språk, kunskaper, värderingar med mera inom olika delar av organisationen, vilket för över oss till den kognitiva heterogeniteten.

**Den kognitiva heterogeniteten** kan som nämnt ses som ett fenomen av individens utveckling. Individens utveckling styrs enligt oss genom erfarenheter och kontakten med människor etc. Detta i sin tur kommer med största sannolikhet att påverka individens krav på system under utveckling. Dessa heterogena krav i kombination med övriga heterogeniteter kommer att leda till en ännu mer omfattande heterogenitet. Simon anser som vi nämnt tidigare i arbetet att man skall låta heterogeniteten leva och ge sig på det problem som man klarar av att lösa nämligen problemet med organisationens design. Detta har som vi ser det ingen direkt påverkan på heterogeniteten. Designen kan dock skönjas bland de bakomliggande problemen som ligger till grund för heterogenitetsproblemen. Dessa kan vara starkt hierarkiska

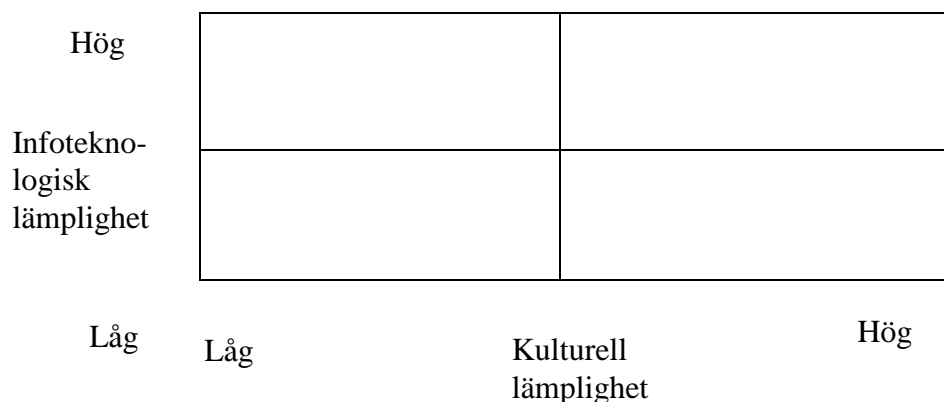
organisationsstrukturer vilket leder till olika behov, informationskrav och begreppsmässiga olikheter.

Ända sedan de stora filosofernas tid så har man försökt att överbrygga de olikheter som förhindrat kommunikation av olika slag. Idag anser många att det är nödvändigt att harmonisera begrepp för att möjliggöra kommunikation. Försöken har varit många och vilket som är bästa sättet anser vi vara omöjligt att avgöra, detta då det finns en uppsjö av olika angreppssätt vilka vi inte är insatta i. Vi låter historien utvisa vilket sätt som är bäst. Vi anser dock att man bör göra denna harmonisering utifrån någon utav de existerande modellerna som finns.

### 7.1.Data Warehouse koncept

Framgången, enligt oss, med Data Warehouse konceptet beror delvis på att Data Warehouse är i samklang med organisationen. Med andra ord så kan man få ett beteende som inte är önskvärt om det görs organisationsförändringar utan att motsvarande görs i Data Warehouse.

Konceptets lämplighet är sålunda inte självklart. Lämpligheten varierar beroende på situationen och kan sägas bestämmas av två variabler, infoteknologisk och kulturell lämplighet. I figuren nedan visar vi de båda dimensionerna, om båda är höga så blir det en bra lösning om inte så är vi mycket tveksamma till applicerbarheten. Langefors betonar vikten av att ta med de kulturella aspekterna och att inte enbart ta hänsyn till infoteknologin. Detta är enligt oss ett mycket sunt synsätt.



Figur 7:1

Som vi nämnt tidigare i diskussionen så är verklighetsuppfattningen eller perspektivet om man så vill en av orsakerna till att heterogenitet finns. Utöver detta så finns det filosofiska olikheter som t.ex. sättet att betrakta rum och tid på. Det finns som nämnt också flera olika angreppssätt för att komma tillrätta med heterogeniteten mellan olika system. Utvecklingen av olika system kommer att leda till att man kommer att få skillnader i systemen som bottnar i att man använt olika modeller för utvecklingen, att aktörerna har olika världsbilder, olika begrepp och olika erfarenheter. Detta är svårt att undvika enligt oss då organisationernas funktionella, kulturella, strukturella och kognitiva verkligheter påverkas av de aktörer som ingår i organisationen. Det har gjorts försök att skapa modeller för att hantera heterogenitet, framgången för dessa har dock varierat.

Ett Data Warehouse är en perfekt lösning för att göra data tillgänglig för hela organisationen utan att göra förändringar i befintliga system. I vissa fall fungerar det bra och i andra dåligt. Detta får till följd att tillämpbarheten inte går att generalisera. Om man tittar på den litteratur som vi läst om Data Warehouse framstår dock lösningen, som vi uppfattar det, som något av en messias lösning. Vi anser även att man missat den kulturella biten helt och hållet. Vi anser vidare att författarna varit mycket ytliga i sina genomgångar av vilka olika heterogenitetsproblem som finns samt hur dessa skall lösas.

Inmon(1996) går in på området heterogenitet även om han inte använder ordet som sådant. Han är dock mycket ytlig i sin genomgång av vilken typ av heterogenitet som han anser vara möjlig att absorbera med ett Data Warehouse koncept. Vidare går han endast in kort på vilka olika typer av beslut och ställningstaganden som kan vara gjorda under de olika systemens tillkomst. De områden som han berör, om än ytligt är följande. Kodning, med andra ord, vilka förkortningar som man beslutat sig för att använda. Det närmaste vi kan komma detta område är det som vi kallar för den innehållsmässiga heterogeniteten och då inom delområdet attribututformning. Inmon(1996) separerar områdena som rör förkortningar och områdena som rör övrig namngivning. Detta ter sig enligt oss ganska underligt då det enligt oss i grund och botten är samma sak.

McFadden Fred R, Hoffer Jeffery A, Prescott Mary (1998). har liksom Inmon(1996) fokus på den innehållsmässiga heterogeniteten. F. MacFadden tar bland annat upp problemet med synonymer och här är fokus på nyckelfälts inkonsistens mellan olika databaser. Här har fälten samma form fast man har döpt dessa olika. F. MacFadden påvisar också att det kan förekomma inkonsistens mellan primärnyckelfälten på så vis att man i en databas har kundnummer som ID och i en annan har kundnamn som ID. Han belyser även problemet med sammansatta fält kontra separerade fält som T.ex. förnamn och efternamn.

T. Connolly och C. Begg(1998) behandlar även de den innehållsmässiga heterogeniteten. Här berörs olika områden om hur och på vilket sätt man skall upprätthålla ett Data Warehouse samt vilka delar som ingår i en sådan procedur. De talar om olika delområden som ingår i ett Data Warehouse såsom den viktigaste delen, nämligen den data karakteristiska av data som hölls inom Warehouse. Fortsättningsvis beskriver de skillnader och likheter mellan ett Data Warehouse system och ett OLTP-system, där nämns målet för de två systemen. Ett OLTP-systems mål är att maximera kapaciteten på transaktionsprocesser medan ett Data Warehouse system är designat för att stödja spontana (ad-hoc)frågeprocesser. Det dessa författare inte behandlar är om hur man skall undvika fallgroparna och överhuvudtaget vilka är dessa fallgropar. Vi ser dessa som brister i att kunna presentera ett koncept fullständigt.

Ett område som Inmon poängterar är att man kan finna stora skillnader om man jämför de operativa systemen och Data Warehouse systemen och hur dessa har sitt fokus gentemot organisationen. Data Warehouse fokuserar på subjekten medan de operationella fokuserar på objekten. Enligt oss har detta dock inget med heterogenitetsproblemet att göra.

Inmon nämner också att man kan finna stora skillnader mellan de olika operationella systemen med hänseende på regelbaser och lagrad information. Att han inte går in mer på vad han menar med regelbaserade skillnader ser vi som en brist i hans bild av Data Warehouse.

Genomgående kan man säga att Inmon och övriga förespråkare för Data Warehouse konceptet endast talar om vad man kan ha ett Data Warehouse till nämligen att samla in och distribuera beslutsinformation. Enligt oss så missar de helt eller åtminstone nästan helt biten med att redogöra för vilka problem som kan finnas inom en organisation när det gäller informationsförsörjning, samt hur dessa skall lösas. De är även mycket noga med att poängtera de skillnader som finns mellan traditionella databassystem och Data Warehouse. Denna skillnad kan förvisso vara intressant ur ett rent informationsteknologiskt arkitekturellt perspektiv, men säger inget om konceptets applicerbarhet.

Man kan jämföra det som Inmon et al skriver med en handböcker av något slag. Den talar om vad som skall lösas, integreringen av operationella databaser och hur det kommer att fungera och se ut när det är gjort, i ett idealt fall. De säger dock inget om var man kan finna alla fallor och problem med integrationen. De kan sägas belysa önskemålet och målet men de har missat stora delar av vägen dit och de bakomliggande orsakerna till att man vill nå önskemålet.

Något som vi anser saknas i den mesta av litteraturen vi gått igenom är fokus på affären, både som process och kravställare. Här kommer kraven i sin tur från miljön. Inmon, MacFadden och Connolly & Begg saknar enligt oss målfokus i sina beskrivningar av konceptet. Enligt dem är målet ökad tillgänglighet av information genom användandet av Data Warehouse. Vi anser dock att enbart en ökad tillgänglighet av information inte är lösningen på informations bristen. Vi vill snarare se det som om lösningen ligger i att förse rätt person med rätt information. Man skall alltså ägna sig åt precisionsskytte i stället för att skjuta med avsågat hagelgevär. Enligt oss så saknar information utan syfte mening. Vi anser att man bör lägga fokus på just att försöka att finna vilken information som är kritisk. Vi anser dock att i vissa fall så kan det vara lämpligt att förändra i organisationen så att behovet av en viss information försvinner. För att exemplifiera vad vi menar så kan vi ta ett exempel från bilindustrin.

I väst Europa så lägger man ner mycket tid på efterkontroll av nybyggda bilar och behöver ett informationssystem för att hantera all data som man får från denna process. I Japan däremot så har man eliminerat felen i produktionen i mycket större grad och behöver inte samma omfattande efterkontroll för att hålla kvaliteten på en hög nivå. Japanerna behöver av denna anledningen inte samma omfattande stöd i utvärderingen av felen då de i princip är obefintliga.

För att man skall kunna nå målen med att minska heterogeniteten på informationen så finns det flera olika koncept förutom ett Data Warehouse koncept som kan användas. Några av dessa är IRM, VBS, mapping med flera. Som vi ser det så har dessa olika koncept olika användningsområden. Vilket som passar bäst i en given situation varierar beroende på vad man har för syfte och mål.

Målen och medlen för att nå målet, kommer att påverka informationsmiljön i varierande grad. Vad man väljer är i högsta grad är en IT-management fråga. Några av de ställningstaganden som man måste göra är att validera vilken strategi som passar bäst i det specifika fallet, samt under vilka villkor skall homogeniseringen ske och vad får det lov att kosta.

För att man skall kunna nå målen med att minska heterogeniteten så är det enligt oss av yttersta vikt att det finns en medvetenhet om vad det finns för olika typer av heterogenitet och vad dessa bottnar i. Om denna insikt saknas så anser vi att utsikterna för att lyckas är starkt begränsade. Vi anser H. Levitts modell som sammanfattar de olika typerna av heterogenitet är något som bör finnas med i bedömningen om vad som är lämpligt.

**Ett Data Warehouse lämplighet** för att hantera informationsbaserad heterogenitet varierar som vi ser det beroende på informationskällornas beskaffenhet samt syftet med homogeniseringen. Om man har samma typ av data i flera databaser och denna information är heterogen bland annat med avseende på definitionerna så anser vi att ett Data Warehouse blir i princip olämpligt. Detta anser vi för att i detta fall så blir det mest en fråga om förhandlingar mellan flera olika parter och inte en fråga om transformation. Om man där emot har data som är olika eller endast delvis överlappande så anser vi att ett Data Warehouse koncept går att använda.

Om man skall lyckas med ett Data Warehouse projekt så krävs det att man klargör vad man har för mål med projektet. Denna process med att fastställa målen och att strukturera detsamma är av yttersta vikt, men kan vara tids ödande. Det som kan göras för att korta tiden är att koncentrera sig på de för organisationen kritiska framgångsfaktorerna. Vad man menar med kritiska framgångsfaktorer kan variera lite. Vi använder oss dock av Peter Söderströms definition "... en omständighet vilken på ett avgörande sätt påverkar måluppfyllelsen". Det sätt som vi anser att man skall använda dessa på är att ha dem som ledning när man skapar ett Data Warehouse för att detsamma skall kunna möta kraven på informativ data som beslutsfattarna ställer.

**Något som vi anser saknas** i mycket av den litteratur vi har läst är vikten av att Data Warehouset skall stödja de för verksamheten viktiga processer. Vi anser vidare att om man inte har någon beskrivning över processerna så skall denna göras. Hur den görs är inte så noga så länge man följer någon utav de många metoder som finns för detta ändamål. Det viktiga är att det görs så att man får klarhet i vilka processer som skall stödjas. Denna brist påverkar lämpligheten negativt som vi ser det.

**Ett problem som vi ser med Data Warehouse** är att det kan krävas mycket förhandlingar då man skall komma tillrätta med heterogeniteten i de befintliga databaserna. Ett område som här kan komma att ta tid att lösa är problem med strukturering av begreppen, vilka kan vara vitt skilda inom olika delar av organisationen. Om man har en organisation med tillverkning, försäljning och reparationer av fordon så kommer man att finna skillnader i hur en bil framställs. Detta på grund av att de olika avdelningarna har modeller som är avsedda att stödja just deras områden. Peter Söderström anser att just problemen med begreppsapparaten är själva källan till heterogeniteten. Detta överensstämmer till viss del med vår åsikt. Vi vill dock gå ett ytterligare steg i problemhierarkin och lägga källan till problemen på organisationens differentiering av begrepp. Vi anser att det är denna skillnad i

människornas uppfattningar av olika begrepp som leder till att separat utvecklade system blir så åtskilda med hänseende på begreppsapparaten.

När man skapar ett Data Warehouse så kan man råka ut för en del problem. Det är viktigt att man har dessa i åtanke vid planering och genomförande av ett Data Warehouse projekt. några av dessa problem kan vara följande:

- Underskattning av resurskraven för att fylla Data Warehouse med data.
- Dolda problem i källsystemen.
- Efterfrågad data ej insamlad.
- Ökande användarkrav.
- Problem vid datahomogeniseringen.
- Höga resurskrav.
- Ägarskapsproblem, vem äger aktuell data.
- Projekten tar mycket tid i anspråk.
- Data Warehouse projekt är långtidsprojekt som kräver komplex integration.

Då användarna får tillgång till fråge- och rapportverktygen så är det inte ovanligt att de efterfrågar mer hjälp från IT-avdelningen. Detta kan bero på den ökande medvetenheten hos användarna om värdet och möjligheterna hos Data Warehouse. Vilket i sin tur kan leda till att användarna ställer krav på fler och mer komplexa utsökningsfrågor och rapporter. Den ökande belastningen på IT-avdelningen kan också bero på bristande utbildning i kombination med komplexa program.

Stora Data Warehouse projekt kan komma att utvecklas till att bli en övning i att homogenisera data detta kommer troligtvis att ske genom olika typer av förhandlingar. Detta på grund av den i många fall mycket stora heterogeniteten systemen emellan och aktörerna emellan. Följden av detta blir stor resurs åtgång ifråga om tid, manskap och i slutändan pengar.

Ett Data Warehouse kan ändra på attityderna hos slutanvändarna om vem som äger aktuell data. Ett som en del upplever det aktuellt problem är att tidigare lokal känslig data i och med ett Data Warehouse blir global. När data blir global så suddas gränserna ut om vem som äger denna data i fråga.

Ett Data Warehouse kan komma att kräva mycket resurser i fråga om tidsåtgången för underhåll. Detta beror på att om ett Data Warehouse skall vara en värdefull resurs för företaget så måste det vara samstämmigt med det samma. Med detta menas att om en organisation genomför organisations förändringar så måste man även göra ändringar i Data Warehouse och anpassa detta till den nya organisationen.

Data Warehouse projekt är projekt som kan ta mycket lång tid i anspråk för genomförande, upp till tre år(Thomas Connoly, Carolyn Begg, Anne Strachan). För att lösa detta problem så inför en del organisationer så kallade data marts vilket endast stödjer kraven från en delverksamhet eller avdelning. Detta att de endast stödjer en delverksamhet medför att de inte blir så komplexa och därför kan byggas och implementeras snabbare.

Data Warehouse konceptets primära syfte, roll är att tillgodose människans informationsbehov. Detta görs genom att absorbera den enorma heterogenitet som kännetecknar de informationskällor som nyttjas för att tillgodose detta behov. Med andra ord en Data Warehouse arkitektur kopplar samman heterogena informationskällor och heterogena informationsmottagare. Denna dubbla heterogenitet resulterar i att det kommer att krävas mycket förhandlingar innan man kan klara av att lösa uppgiften.

Om man gör en för ytlig genomgång av problemen som finns inom organisationen så finns risken att man behandlar ett underordnat problem som ett överordnat. Detta kan verka trivialt men kan få följden att man hamnar i en situation som gör fortsatt arbete omöjligt, om man inte börjar om. Man bör alltså göra en djuplodande problemundersökning och beskrivning. Peter Söderström(1997) ger stöd för en djupare problembeskrivning, han poängterar dock att det kan väcka starka känslor om man vill rota för djupt i problem som ytligt sett inte är relaterade. Men som sagt så anser vi att detta är av yttersta vikt för att man skall lyckas. Problemet är förvisso inget problem som är specifikt för Data Warehouse projekt men vi anser att man bör nämna de problem som man kan komma att stöta på under arbetets gång. Denna kartläggning ger sedan en bra bild av vad som skall hanteras och vilken data som kommer att krävas.

En fråga som faller sig naturlig är, vem skall organisera begreppen? Vi anse att denna uppgift skall drivas inom kärnverksamheten. Detta då det är här som man har den klaraste bilden om vad som skall göras. IT avdelningen skall sedan bidra med sin kunskap om hur arbetet skall göras och vilka metoder som man skall använda.

Om man går tillbaka till exemplet med banken i inledningen så kan man säga att man kommer att stöta på stora problem rörande integreringen. De tekniska problemen undantagna så kommer man att finna problem med motsägelser, inkonsistens samt en uppsjö av andra problem som är relaterade till struktureringen av data. Wang och Madnick (1989) talar om lite olika tekniker som finns för att komma tillrätta med dessa problemen och göra en sammankoppling möjlig.

Om integrationen av de olika banksystemen har genomförts på ett riktigt sätt, där man har löst de grundläggande problemen med att jämka i hop de olika begreppen samt de övriga informations baserade problemen, så kommer man att ha en struktur på informationen som har goda chanser att bli överblickbar. Detta är som vi ser det en klar fördel om man vid senare tillfälle vill införa ett beslutstödssystem av Data Warehouse typ. Fördelen ligger i som vi ser det att man redan genomgått mycket av de förhandlingar som krävs för att man skall kunna enas om vilka begrepp som skall gälla.

I bankexemplet i inledningen så har man system som inte är helt överlappande utan endast delvis överlappande. Detta medför enligt oss att man kan applicera ett Data Warehouse koncept om man vill få till gång till adekvat beslutsinformation. Om systemen varit helt överlappande och varit föremål för integration av något slag så hade det troligtvis inte varit en bra lösning med ett Data Warehouse koncept då detta hade krävt mycket förhandling.

Om vi går vidare och tittar på exemplet med resesystemen så visar detta exempel att det finns problem på två olika nivåer som måste lösas i detta fallet, nämligen på schema- och instansvärdenivå. Dessa skillnader inkluderar synonymer, strukturella skillnader och ofullständighet.

Hur man skall gå tillväga då man skall samla in information från de tre systemen är inte självklart. Enligt oss räcker det om man har någon typ utav meddelandesamverkan mellan systemen. Denna bör vara så beskaffad att man eliminerar de olikheter som finns mellan systemen. Om detta görs på ett riktigt sätt så kommer man att få en högre grad av användbarhet än vad man har idag. Dagens system är ursprungligen skapade för sin specifika verksamhets aktörer och efter deras begreppsramar. Nu när man skall göra en integration så kommer problemen just på grund av att man inte förstår de andra systemen och vad deras olika begrepp betyder som till exempel fallet med vad som menas med exklusiv. Problemen här uppkommer bland annat på grund av de kulturella, funktionella och kognitiva heterogeniteter som finns mellan de berörda organisationerna, aktörerna och systemen.

I detta fallet så anser vi dock inte att man behöver göra en djupare integration detta då det inte skulle vara försvarbart med tanke på de tidskrävande förhandlingar som skulle bli följden vid en fullständig integration där man homogeniserar begreppen. Kostnaden i förhållande till den eventuella vinsten man skulle göra på en fullständig integration är inte enligt oss försvarbar. Vi anser således att i detta fallet så får man fullgod funktion enbart med meddelande samverkan mellan systemen.

Dessa problemen som vi gått igenom påvekar naturligtvis Data Warehouse konceptets lämplighet. Konceptet är som vi ser det en snygg och smart ingenjörslösning men detta räcker inte alltid enligt oss. Vi anser att det är av yttersta vikt att human och organisationsperspektivet lyfts fram. Mycket av arbetet är att jämkä olika verklighetsuppfattningar och då räcker det inte att man har en snygg ingenjörslösning detta då den missar många enligt oss viktiga aspekter. Data Warehouse lägger sig på en mycket hög nivå enligt oss då det inte berör de bakomliggande orsakerna till heterogeniteten eller hur man skall gå till väga för att lösa problemen med den samma.

Det bidrag som vår undersökning gjort till forskningen är att synliggöra den otroligt omfattande heterogenitet som finns mitt ibland oss. Uppsatsen presenterar även en modell för att överblicka den heterogenitet som man kan finna. Stöd för modellens riktighet finns i den tidigare forskning som utförts inom området.

Ut över detta så har vi klargjort Data Warehouse begränsningar. Om alla de begränsningar och brister som vi funnit är helt riktiga är inte helt säkert. Detta för att vi inte läst all litteratur som finns skrivet inom ämnet. Vi anser dock att vi har skapat en rättvis bild av Data Warehouse applicerbarhet samt dess förmåga att hantera heterogenitet. Detta anser vi bland annat för att W. H. Inmon, som räknas som Data Warehouse fader, har skrivit en del av den litteratur som vi bygger vår studie på. Om någon kan redogöra för konceptet på ett riktigt sätt så borde det vara han.

Vidare forskning inom ämnet kan vara att gå på djupet inom någon av de olika typerna av heterogenitet. Detta för att skapa sig en djupare förståelse om dessa. Den djupare förståelsen skulle sedan vara till hjälp vid hanteringen av heterogenitetsproblemen inom denna del.



## 7.2. Validitet

Vi har sammanställt och presenterat de olika typer av heterogenitet som vi kommit fram till vilket varit ett av våra syften med uppsatsen. Utöver detta så har Data Warehouse konceptet utvärderats utifrån vår frågeställning. Detta ger enligt oss arbetet en hög grad av validitet. Om man ser på vår fältundersökning som syftade till att synliggöra de olika heterogeniteter man kan finna mellan olika system så anser vi att undersökningen uppnått det önskade resultatet. Detta då den påvisade heterogenitet mellan informationsbaserna. Detta bidrar enligt oss till att ytterligare höja arbetets validitet.

## 7.3. Reliabilitet

Slutligen kan man undra om vårt resultat av arbetet är riktigt. Vi anser att vårt arbete har en hög grad av reliabilitet. Detta då arbetet stödjer sig på mycket av tidigare forskning inom ämnet. Att förkasta vårt arbete innebär enligt oss att man måste förkasta all tidigare forskning inom ämnet. Vi har mycket svårt att se hur ett sådant förkastande skall kunna motiveras.

Resultaten utav vår fältundersökning påvisade heterogenitet mellan systemen. Den var dock ganska grund och visade därför inte all typ av heterogenitet som kan förekomma mellan olika system. Detta gör att reliabiliteten på denna del av arbetet blir något lägre än arbetet i övrigt. Anledningen att fältstudien fick en något ringa omfattning var att tiden var mycket begränsad, detta till följd att det system som skulle undersökas från början inte blev tillgängliga, därav valet av en grund studie på biblioteken.

## 8. Slutsats

Slutsatserna som vi kommit fram till i resultatet och diskuterat i diskussionen redovisar vi i detta avsnitt. Vi har kommit fram till att området som berör heterogenitet är mycket omfattande och komplext. Om det skall vara möjligt att förstå problemet så måste man se till hela problemet och inte endast till en begränsad del.

Vi har under arbetets gång lyckats att lokalisera de fem övergripande typer av heterogenitet som man kan finna. Alla dessa påverkar och påverkas av varandras utveckling och dynamik. De fem sorterna som vi funnit är funktionell, kulturell, strukturell, kognitiv och informationsteknologisk heterogenitet. De olika heterogeniteterna går inte att rangordna utan är alla lika viktiga enligt oss.

Vi har vidare kombinerat dessa olika heterogeniteter i en modell tillsammans med begreppsmässig, informationsmässig, regelmässig och slutligen filosofisk heterogenitet. Modellen kan tjäna som hjälp då man försöker att överblicka de olika heterogeniteterna. I modellen är även de berörda arkitekturerna införda.

Vi har även gjort en utveckling av modellen som gör att den även visar det mänskliga perspektivet för att kunna synliggöra den sociala arkitekturen. Modellen kan fungera som ett verktyg för att tydliggöra de aspekter som är viktiga att tänka på vid utvecklandet av system och vid integration av de samma.

I analys delen så utvärderade vi Data Warehouse lämplighet för att hantera heterogenitet. Här kom vi fram till att lämpligheten inte går att generalisera detta då lämpligheten varierar från fall till fall. Det som man kan hantera med ett Data Warehouse är snarast heterogenitet som uppkommer som en följd av de fem huvudsorterna. Dessa följd heterogeniteterna är:

- Enkla begreppsmässiga konflikter
- Enkla stilistiska skillnader i språket
- Enkla strukturella konflikter( mellan entiteter och egenskaper och inte den organisatoriska strukturella heterogeniteten)

På dessa tre områden så fungerar ett Data Warehouse bra det saknar dock målfokus vilket är en svaghet.

Om man har en informationsmiljö där databaserna är helt eller till stor del överlappande så minskar lämpligheten av ett Data Warehouse. Man kan säga att ju mer heterogen en organisation är desto mer förhandlingar kommer att krävas mellan berörda parter för att man skall kunna införa Data Warehouse. Lämpligheten är således beroende på hur mycket tid och pengar man måste lägga ner på bland annat förhandlingar och hur mycket man får till baka på satsat kapital. Nyttan måste även bestämmas utifrån målbilden som skall betjänas och hur denna är.

## 9. Litteraturförteckning

Ackoff, R. L. (1974). *Redesigning the Future: A systems Approach to Societal Problems*, John Wiley & Sons

Ackoff, R. L. (1967). Management Misinformation Systems. *Management Science*, Vol. 4, No.4

Andersen, E. S.(1994). *Systemutveckling – principer, metoder och tekniker*.

Anthony, R (1965). *Planing and Cotrol Systems: A Framewrk for Analysis*. Boston, Harvard Univesity.

Argyris, C.(1977). Oranizational Learning and Management Information Systems.*Accounting, Organizations and Society*, Vol.2, No.2

Argyris, C.(1980). Some Inner Contradictions in Management Information Systems. IH. Lucas, F. Land, T. Lincoln och K. Supper, red. *The Information Systems Environment*, North-Holland.

Berry, M., Linoff.(1997). *Data Mining Techninques for Marketing, Sales and Customer Support*, Wiley Computer Publishing.

Bubenko jr., J. (1977). *Validity and Verification Aspects of Information Modeling*. VLDB 77.

Bubenko jr., J.(1981). *On Concepts and Strategies for Requirements and Information Analysis*. Stockholm och Göteborg, SYSLAB Report No.4.

Bubenko jr., J.(1984). *Konceptuell modellering – Informationsanalys*, Studentlitteratur, Lund.

Bunge, M.(1967). *Scintific Research*, Springer-Verlag.

Connolly, Thomas AND Begg, Carolyn. (1998). *DataBase Systems A Practical Approach to Design, Implementation and Management* Second Edition.

Galbraith, Jay. R.(1977). *Oragnization Design*, Addison-Wesley Publishin Company.

Goldkuhl, G.(1986). *Utveckling av användarens verksamhets språk – Ett nytt synsätt på informationsbehovs analys*, Institutionen för datavetenskap, Universitetet i Linköping.

Gorry, A. och M. S. Scott-Morton (1971). A Framework for Management Information Systems. *Sloan Management reviw*, Vol. 13, No.1:55-70.

Harrington, J.(1991). *Organzational Structure and Information Technology*, Prentice-Hall.

- Hewitt, C.(1986). Offices are Open Systems. *ACM Transactions on Office Information Systems*, Vol. 4, No.3.
- Hugoson, M.-Å.(1986). Informationssystem i ett verksamhets perspektiv. Funktionell strukturering. *Nordisk DATAnytt, Nr10*.
- Hugoson, M.-Å.(1987). *Systemstrukturering*. Göteborg, Programator AB.
- Hugoson, M.-Å.(1989). *On the Interaction Between Information Systems*, Institutionen för informationsbehandling, Göteborgs universitet.
- Hugoson, M.-Å.(1990). *Verksamhetsbaserad Systemstrukturering. Principer och tillämpningar*, Göteborg, Programator AB.
- Kimball, Ralph. (1996). *The Data Warehouse Toolkit*. Wiley
- Langefors, B.(1975). Control Structure and Formalized Information Analysis in Organizations. IE. Grochla och N. Szyperski, red. *Information Systems and Organizational Structure*. Berlin och New York, Walter de Gruyter.
- Langefors, B.(1978). Analysis of User Needs. IG. Bracchi och P. C. Lockerman, red. *Informations Systems Methodology*. Berlin, Springer-Verlag.
- Langefors, B.(1981). Kartläggning av informationsbehov – en (decentraliserad) läroprocess. I Malmberg, red. *Decentralisering och den ekonomiska uppföljningen i företaget*. Stockholm, Management Press, SAF:s förlag.
- Langefors, B.(1984). *The Inforlogical Equation and Decentralization*. SPOT-3, Göteborg, SYSLAB, Chalmers tekniska högskola, Göteborg.
- March, J. G. och H. A. Simon(1958). *Organizations*, John Wiley & Sons.
- Magoulas, T. & Pessi, K.(1998). *Strategisk IT-management* Institutionen för informatik.
- McFadden, Fredrik R. , Hoffer, Jeffrey A., Prescott Mary B., (1998) Fifth edition *Modern database management*.
- Patel, R., Davidson, B. (1994). *Forskningsmetodikens grunder, att planera, genomföra och rapportera en undersökning*. Lund: Studentlitteratur
- Patel, R, Tebelius, U., (1987)*Grundbok i forskningsmetodik*. Lund: Studentlitteratur.
- Putnam, A., E. R. Barlow, et al.(1966). *Samordnad företagskontroll – Totala informationssystem i praktisk belysning*, J Beckmans förlag.
- Söderström, Peter. (1997). ”Data Warehouse” *Datalager, Verksamhet, Metod, Teknik*.
- Wallén, G. (1996) *Vetenskapsteori och forskningsmetodik*. Lund: Studentlitteratur.

Wang, R., Madnick, S. E.(1989). *Facilitating Connectivity in Composite Information Systems*. DATA BASE, Fall.

Wetherbe, J. C.(1988). *Systems Analysis and Design*, West Publishing Company.

W.H.Inmon,(1993). *Information Systems Architecture, Development in the 90's* , Wiley & Sons, Inc, New York.

W.H.Inmon,(1996). *Building the Data Warehouse, second edition*, Wiley Computer Publishing.

Zachman, J. A.(1977). Control and Planning of Informations Systems. *Journal of Systems Management*, Vol. 28, No.7.

Zachman, J. A.(1978). The Informations Systems Management System: A Framework for Plannig. *DATA BASE*, Winter.

Zachman, J. A.(1987). A Framework for Informations Systems Architecture. *IBM Systems Journal*, Vol. 26, No.3:276-292.

Handledare, Fil Dr. Thanos Magoulas,(1999). Institutionen för informatik

Föreläsningar inom informatik som vetenskap(1998). Handelshögskolan vid Göteborgs Universitet.

Professor Bo Dahlbom, Carsten Sørensen m.fl.