

# OBJEKTORIENTERADE DATABASER

---

*Faktorer som har påverkat den objektorienterade  
databasens begränsade framgång*

Magisteruppsats, 10 poäng, inom Systemvetenskapliga programmet

*Framlagd:* juni, 2006

*Författare:*

Katrin Hedström

Krister Larsson

*Handledare:*

Bo Andersson

## Objektorienterade databaser

### *Faktorer som har påverkat den objektorienterade databasens begränsade framgång*

© Hedström, Larsson

Magisteruppsats framlagd juni, 2006

Omfång: 48 sidor

Handledare: Bo Andersson

### **Resumé**

Objektorienterade databaser har en fördel gentemot relationsdatabaser när det gäller hantering av komplexa datatyper, dessa datatyper används inom till exempel applikationsområdena CAD, CAM och CASE. Trots detta är det relationsdatabaser som används mest inom dessa områden. Syftet med denna uppsats är att ta reda på vilka faktorer som påverkar att valet av lagringssätt inte faller på objektorienterade databaser.

Vi har valt att enbart titta på företag som utvecklar CAD-system och genomförde fyra semistrukturerade intervjuer på sådana företag. Vi kom fram till att de faktorer som påverkar att objektorienterade databaser inte används är att systemets kod inte är helt objektorienterad, att det inte förmedlas tillräckligt med information om objektorienterade databaser, att det inte finns någon benägenhet hos företagen att prova på ny teknik på grund av begränsad ekonomi samt att prestandafrågor har minskat i betydelse.

**Nyckelord:** *Objektorienterade databaser, CAD, databaser, ODB, ODBMS*

# INNEHÅLLSFÖRTECKNING

<b>1 INLEDNING</b> .....	<b>5</b>
1.1 BAKGRUND.....	5
1.2 PROBLEMMRÅDE OCH FRÅGESTÄLLNING.....	6
1.3 SYFTE.....	6
1.4 AVGRÄNSNING.....	6
1.5 LITTERATURSÖKNING.....	7
1.5.1 Litteraturkritik.....	7
1.6 MÅLGRUPP.....	7
1.7 ORDLISTA.....	8
<b>2 METOD</b> .....	<b>9</b>
2.1 METODVAL.....	9
2.2 UNDERSÖKNINGSMODELL.....	9
2.3 VÅR ANVÄNDNING AV UNDERSÖKNINGSMODELLEN.....	11
2.4 INSAMLING AV PRIMÄRDATA.....	12
2.4.1 Intervjuguide.....	12
2.4.2 Urval.....	13
2.4.3 Genomförande.....	14
2.5 ANALYS.....	15
2.5.1 Kodning av intervjumaterial.....	15
2.5.2 Matris.....	15
2.6 KVALITET.....	17
2.6.1 Reliabilitet.....	17
2.6.2 Validitet.....	17
2.6.3 Objektivitet.....	17
2.7 ETIK.....	17
<b>3 OBJEKTORIENTERADE DATABASER</b> .....	<b>19</b>
3.1 OBJEKTORIENTERADE DATABASER.....	19
3.1.1 Objektorienterad funktionalitet.....	19
3.1.2 Lagring.....	20
3.1.3 Språk.....	20
3.1.4 Underhåll.....	20
3.1.5 Modell.....	21
3.1.6 Annan funktionalitet.....	21
3.1.7 Säkerhet.....	21
3.2 SKILLNADER MELLAN OBJEKTORIENTERADE DATABASER OCH RELATIONS-DATABASER.....	22
3.2.1 Verktyg.....	22
3.2.2 Interaktion mellan applikation och databas.....	22
3.2.3 Objektidentifierare.....	22
3.2.4 Design av databasen.....	23
3.2.5 Prestanda.....	23
3.2.6 Lättanvändbarhet.....	24
3.2.7 Sammanfattning.....	24
3.3 APPLIKATIONSOMRÅDEN FÖR OBJEKTORIENTERADE DATABASER.....	25
3.3.1 Komplexa objekt.....	26
3.4 CAD.....	27
<b>4 UNDERSÖKNING</b> .....	<b>28</b>
4.1 PRESENTATION AV FÖRETAG OCH RESPONDENTER.....	28
4.1.1 Acad – Anders.....	28
4.1.2 Bcad – Bengt.....	28
4.1.3 Ccad – Calle.....	29
4.1.4 Dcad – David.....	29
4.2 PRESENTATION AV INTERVJURESLTAT.....	29

4.2.1 Relativa fördelar .....	29
4.2.2 Lättanvändbarhet.....	29
4.2.3 Kompatibilitet .....	30
4.2.4 Benägenhet att prova på ny teknik.....	30
4.2.5 Medvetenhet.....	30
4.2.6 Kommunikationskanaler .....	31
4.2.7 Avsikter .....	31
<b>5 ANALYS.....</b>	<b>32</b>
5.1 ANALYS UTIFRÅN AGARWAL-PRASAD-MODELLEN.....	32
5.1.1 Intervju med Anders på Acad.....	32
5.1.2 Intervju med Bengt på Bcad.....	33
5.1.3 Intervju med Calle på Ccad .....	34
5.1.4 Intervju med David på Dcad.....	35
<b>6 DISKUSSION .....</b>	<b>36</b>
6.1 INLEDANDE DISKUSSION .....	36
6.2 DISKUSSION UTIFRÅN AGARWAL-PRASAD-MODELLEN.....	36
6.2.1 Relativa fördelar .....	36
6.2.2 Lättanvändbarhet.....	37
6.2.3 Kompatibilitet .....	38
6.2.4 Benägenhet att prova på ny teknik.....	38
6.2.5 Medvetenhet.....	38
6.2.6 Avsikter .....	39
6.2.7 Förhållanden som inte förklaras av Agarwal-Prasad-modellen.....	40
6.2.8 Modellen som helhet .....	40
<b>7 SLUTSATSER.....</b>	<b>42</b>
7.1 FRAMTIDA FORSKNING .....	43
<b>BILAGA 1 – INTERVJUGUIDE</b>	
<b>BILAGA 2 - BESKRIVNING AV KODER</b>	
<b>REFERENSER</b>	

## 1 Inledning

---

# 1 INLEDNING

---

*I detta kapitel beskriver vi bakgrunden till uppsatsens fokus och definierar dess problemområde, frågeställning och syfte. Vi beskriver även våra avgränsningar och uppsatsens tilltänkta målgrupp.*

---

## 1.1 Bakgrund

De tidigaste databassystemen var hierarkiska databassystem och nätverksdatabassystem (Elmasri & Navathe, 2004). Ett av de stora problemen med dessa databassystem var att de skulle hantera både konceptuella förhållanden och den fysiska lagringen. De var enligt författarna inte tillräckligt flexibla för att kunna komma åt data effektivt. För att ställa frågor till databasen krävdes att kod skrevs i det programmeringsspråk som användes, vilket gjorde det mycket tidskrävande och dyrt. Dessa databassystem användes från mitten av 1960-talet fram till slutet av 1980-talet (ibid.).

Enligt Elmasri och Navathe (2004) utvecklades relationsdatabasen på sent 1970-tal för att separera den fysiska lagringen av data från den konceptuella presentationen och för att bidra med en matematisk grund för databaser. Med denna databas kom även ett frågespråk som gjorde det snabbare att skriva frågor. De första kommersiella relationsdatabashanterarna introducerades enligt författarna på tidigt 1980-tal men var då ganska långsamma eftersom det inte användes fysiska lagringspekare för att komma åt relaterad data. Med utvecklingen av nya lagrings- och indexeringstekniker och bättre frågebehandling och optimering förbättrades prestationen avsevärt. Nu finns enligt författarna relationsdatabaser på nästan alla typer av datorer.

I samband med att objektorienterade programmeringsspråk fick ett genomslag på 1980-talet skapades behovet att lagra objekt med komplexa strukturer vilket ledde till utvecklingen av objektorienterade databaser (Elmasri & Navathe, 2004). Till en början sågs de som en konkurrent till relationsdatabaserna eftersom de hade mer generella datastrukturer och kunde hantera bland annat abstrakta datatyper, inkapsling och arv. Författarna menar dock att modellens komplexitet och avsaknaden av en tidig standard bidrog till objektorienterade databasers begränsade användning. De används idag huvudsakligen i specialiserade applikationer såsom engineering design, multimedia och tillverkningsystem. Förutom de renodlade objektorienterade databaserna finns det enligt författarna även vissa relationsdatabassystem som utökats med objekt, dynamiska datatyper och annan objektorienterad funktionalitet.

Objektorienterade databaser har flera fördelar gentemot relationsdatabaser, till exempel används persistent programmering (Bancilhon, 1996), prestandan är bättre vid applikationer som hanterar komplexa objekt (Cattell & Skeen, 1992) och förhållandet mellan applikationens datamodell och databasens datamodell är starkt

## 1 Inledning

---

(Devarakonda, 2001). Objektorienterade databashanterare har dessutom samma funktionalitet som både relationsdatabashanterare (Bancilhon, 1996; Loomis, 1995) och objektorienterade programmeringsspråk (Elmasri & Navathe, 2004).

Vissa data är bättre att lagra i relationsdatabaser, det är data som är enhetliga, upprepande och kan läggas in i tabeller (Beck, 2001). Stora dokument och komplexa datatyper är däremot bättre att lagra i objektorienterade databaser (Beck, 2001; Elmasri & Navathe, 2004; Loomis, 1995). Exempel på tillämpningsområden som hanterar denna typ av data är CAD, CAM och CASE.

## 1.2 Problemområde och frågeställning

Trots dessa fördelar med objektorienterade databaser och det faktum att objektorienterad programmering används så används relationsdatabaser till största delen (Deitel & Deitel, 2005; Elmasri & Navathe, 2004). Vi är intresserade av att undersöka varför det förhåller sig så och har med utgångspunkt från detta utformat följande frågeställning:

*- Varför används inte objektorienterade databaser inom applikationsområden där de hade varit att föredra?*

För att kunna förklara det har vi valt att använda oss av en undersökningsmodell som Agarwal och Prasad (1998) har tagit fram. Modellen förklarar acceptans av en teknisk lösning med bland annat individens uppfattning av den nya tekniska lösningen. Uppfattningarna påverkas enligt författarna av individens medvetenhet samt kommunikationskanaler. Om individen har en positiv uppfattning gentemot den tekniska lösningen kan denne ha en avsikt att börja använda denna förutsatt att individen tycker om att prova på nya verktyg och tekniker.

## 1.3 Syfte

Olika författare hävdar att objektorienterade databaser är att föredra inom vissa applikationsområden. De framhäver dessutom att objektorienterade databaser trots detta inte används i så stor utsträckning. Syftet med denna uppsats är att ta reda på varför det uppstått en ”mismatch” mellan teorins uppfattning om att objektorienterade databaser är att föredra och att objektorienterade databaser inte används.

## 1.4 Avgränsning

Vi har valt att avgränsa oss till att undersöka endast ett applikationsområde där objektorienterade databaser har en fördel framför andra lagringssätt. Det applikationsområde vi valt att undersöka är Computer Aided Design (CAD). Vidare har vi inte undersökt vad som påverkar vid val mellan olika tillverkarens produkter, vi har endast

## 1 Inledning

---

undersökt valet mellan objektorienterade databaser och andra lagringssätt. Vi har enbart undersökt CAD-företag som inte använder objektorienterade databaser. Eftersom vi vill veta varför företag inte använder objektorienterade databaser har vi tittat på företagets acceptans av en teknisk lösning och inte individers acceptans.

### 1.5 Litteratursökning

För att hitta litteratur passande för vårt ämnesområde började vi med att titta på tidigare uppsatser som behandlar databaser. Dessa hittade vi genom listan över uppsatser inom informatik som finns på Ekonomihögskolans bibliotek. Vi sökte sedan upp de referenser som fanns i de uppsatser vi fann mest intressanta genom att använda oss av Lovisa. Först sökte vi på referensernas titlar och sedan på författarna för att se om de skrivit några andra böcker inom ämnet, vilket oftast var fallet. Vi har även sökt på fritext och letat i bibliotek efter böcker som handlar om databaser.

Det behövdes även passande vetenskapliga artiklar inom ämnet och för att hitta dessa använde vi oss artikeldatabaserna ELIN, ACM, Jstor, IEEE och Google Scholar. När vi sökte efter artiklar sökte vi på begreppen "object", "databases", "ODB", "RDB", "RDBMS", "ODBMS", "CAD", "OODB", "OODBMS" och kombinationer av dessa. När det gäller vetenskapliga artiklar ville vi att de skulle vara så nya som möjligt för att kunna se de aktuella tendenserna inom området, vilket även gör att vi kompletterar den lite äldre litteraturen på ett bra sätt.

#### 1.5.1 Litteraturkritik

De böcker och artiklar vi har läst har i stort sett gett oss samma information vilket gör att delar i vår litteraturgenomgång påminner om en lärobok. Vi har sökt efter litteratur med andra uppfattningar men har i många fall inte hittat sådan litteratur.

### 1.6 Målgrupp

Målgruppen som uppsatsen vänder sig till är i första hand personer som ska ta beslut om hur lagring ska ske inom ett område där objektorienterade databaser har en fördel. Det förutsätts att dessa personer har grundläggande kunskaper om databaser och objektorienterad programmering.

I andra hand vänder sig uppsatsen till personer som har grundläggande kunskaper om databaser och objektorienterad programmering och som vill veta varför objektorienterade databaser inte används. Det är till exempel studenter inom datalogi och systemvetenskap, personer som arbetar med relationsdatabaser eller objektorienterade databaser samt personer som arbetar med programmering.

## **1 Inledning**

---

Om läsaren anser att metodiken som använts är viktig förutsätts även grundläggande kunskaper om vetenskapliga metoder.

### **1.7 Ordlista**

**Persistent data:** Persistent data lagras på disk och finns kvar så länge ingen tar bort den (Bancilhon, 1996). Se även transient data.

**Transient data:** Transient data har lika lång livslängd som applikationen och tas därmed bort när applikationen avslutas (Bancilhon, 1996). Se även persistent data.



## 2 Metod

---

# 2 METOD

---

*Metodkapitlet innehåller en redogörelse för de olika angreppssätt vi har valt att använda oss av i vår uppsats och varför vi har valt dessa sätt. Vi redogör för de övergripande metodval vi har gjort, hur insamling av primärdata har gått till och hur vi har analyserat resultatet av empirin.*

---

## 2.1 Metodval

I vår uppsats har vi använt oss av en kvalitativ ansats med det som Jacobsen (2002) kallar för intensivt upplägg. Detta innebär att vi har koncentrerat oss på några få undersökningsenheter för att få en så fullständig bild som möjligt av det vi har undersökt. Vi har vidare använt oss av ett deduktivt sätt att samla information enligt Jacobsen (2002), vi samlade först kunskap genom teori och tidigare undersökningar för att sedan gå ut och empiriskt undersöka området. Risken med detta är enligt författaren att vi som forskare endast funnit det som vi ansåg vara relevant och att vi därmed stödjer våra förväntningar. Eftersom vi varit medvetna om denna risk anser vi att vi inte varit fullt lika begränsade i vårt letande efter information.

## 2.2 Undersökningsmodell

Vi har valt att som utgångspunkt för insamling av empiri samt analys använda den undersökningsmodell Agarwal och Prasad (1998) utvecklat (se figur 2.1). Vi kommer fortsättningsvis kalla den för Agarwal-Prasad-modellen. Denna modell är en av de modeller som undersöker vad det är som påverkar acceptans av en teknisk lösning. Modellens olika delar är baserade på modellerna Media Richness Theory (MRT), Technology Acceptance Model (TAM), Theory of Reasoned Action (TRA), the Fishbein/Ajzen attitude-behaviour model samt olika författares utvärdering av dessa modeller. Anledningen till att vi valt att använda denna undersökningsmodell är just att den är baserade på dessa välkända modeller samt att den innehåller faktorer som vi anser som intressanta för vår undersökning. Förklaring av Agarwal-Prasad-modellen följer på nästa sida.

## 2 Metod

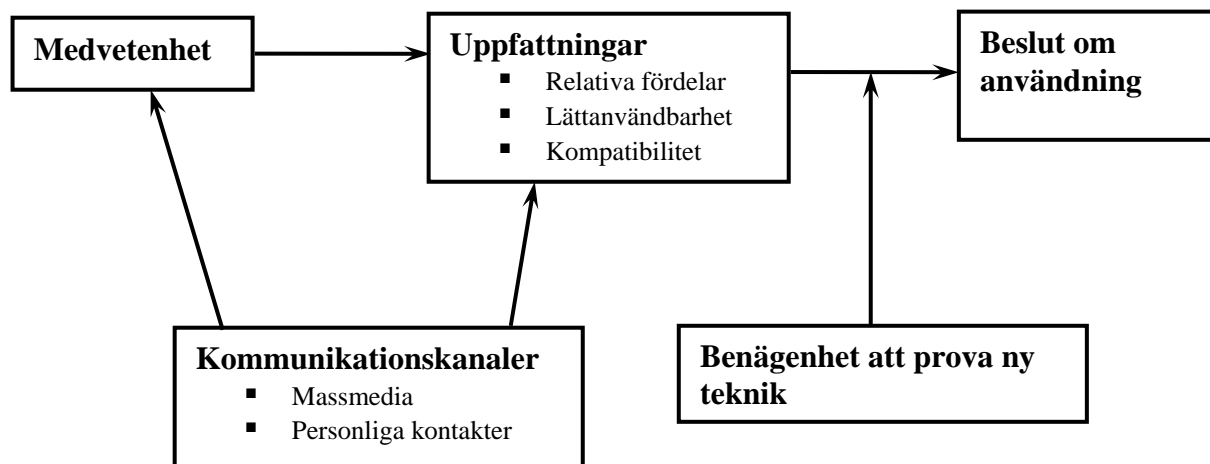


Fig 2.1 Agarwal-Prasad-modellen (Agarwal & Prasad, 1998) egen översättning

En individs beslut om användning av en teknisk lösning kan enligt modellen till stor del förklaras av individens uppfattningar, som består av relativa fördelar, lättanvändbarhet och kompatibilitet. Relativa fördelar innebär enligt Agarwal och Prasad (1998) i vilken grad personen anser att den nya tekniska lösningen har fördelar gentemot tidigare sätt att utföra samma uppgift. Fördelarna har främst att göra med om arbetet kan utföras på kortare tid eller mer effektivt med hjälp av den nya tekniska lösningen, om den nya tekniska lösningen är lättare att använda, ger större kontroll över arbetet eller förbättrar kvaliteten och produktiviteten.

Lättanvändbarhet är i vilken grad individen ser att användandet av den nya tekniska lösningen kräver relativt lite ansträngning (Agarwal & Prasad, 1998). Tekniska lösningar som uppfattas som lättförståeliga och lätta att lära sig har större chans att bli accepterade och använda. Kompatibilitet är i vilken grad den nya tekniska lösningen uppfattas som överensstämmande med individens värderingar, behov och tidigare erfarenhet (Agarwal & Prasad, 1998). Det handlar även om i vilken grad arbets sättet ändras i och med den nya tekniska lösningen.

Förhållandet mellan uppfattningar och avsikten att använda den tekniska lösningen kan enligt Agarwal och Prasad (1998) påverkas av personlighetsfaktorer. Benägenhet att prova på ny teknik är enligt författarna om individen tycker om att experimentera med nya verktyg och metoder och om denne helst ser att andra reder ut buggar och problem med verktyg innan egen användning. Benägenhet att prova på ny teknik påverkar förhållandet mellan uppfattningar och användning positivt (Agarwal & Prasad, 1998). I deras undersökning var det dock bara förhållandet mellan kompatibilitet och införande som påverkades positivt.

Medvetenhet är att individen har utvecklat en positiv attityd gentemot den tekniska lösningen (Agarwal & Prasad, 1998). En positiv attityd skulle enligt författarna kunna kännetecknas av att individen anser att den nya tekniska lösningen är en viktig innovation, att den är kritisk för att få en konkurrensfördel och att individen anser att

## 2 Metod

---

det vore lämpligt att börja använda den tekniska lösningen. Medvetenhet påverkar enligt Agarwal och Prasad (1998) individens uppfattning om den tekniska lösningen positivt.

Det finns två typer av kommunikationskanaler. Massmediakanaler som är en av dessa förmedlar generell information om teknikens nytta (Agarwal & Prasad, 1998). Massmedia påverkar individens medvetenhet om den tekniska lösningen mer än personliga kontakter. Den andra kommunikationskanalen är enligt Agarwal och Prasad (1998) personliga kontakter som förmedlar teknikens värde för individen. Personliga kontakter påverkar individens uppfattningar positivt mer än massmedia, i deras undersökning var det dock bara relativa fördelar och kompatibilitet som påverkades positivt.

Beslut om användning syftar till individens avsikt eller beslut att börja använda den tekniska lösningen. Förhållandet mellan en individs uppfattningar och beslutet om användning påverkas positivt om det finns en benägenhet att prova på nya verktyg.

### 2.3 Vår användning av undersökningsmodellen

Agarwal-Prasad-modellen är främst framställd för att undersöka individens acceptans men eftersom vi vill veta varför företag inte har objektorienterade databaser har vi använt den snarare för att undersöka företagets acceptans. Respondenten på företaget har alltså representerat företaget och svarat utifrån företagets synvinkel. Eftersom det i vårt fall handlar om företags införande av en teknik och inte en individs användande av en befintlig teknik tror vi att beslutet och avsikten skiljer sig åt mer än om det bara handlar om en individ. Vi vill därför se hur både avsikter och beslut påverkas av modellens olika faktorer. Under undersökningens gång har vi även blivit tvungna att göra en mindre förändring i modellens kommunikationskanaler. Vi har inte kunnat kategorisera alla resultat vi fått från intervjuerna som antingen massmedia eller personliga kontakter, vilket har lett till att vi har sett kommunikationskanaler som en enhet istället för två skilda delar.

Modellens olika delar har även nyttjats för att utforma vår intervjuguide och presentera empirin. Vi använder även modellen i analysen för att analysera varje intervju för sig för att se vilka faktorer som påverkat varandra inom varje intervju.

En fördel med att använda Agarwal-Prasad-modellen har varit att vi på ett vetenskapligt sätt har kunnat rikta arbetet och därigenom endast undersökt de faktorer som ingår i modellen. Eftersom Agarwal-Prasad-modellen är framtagen från annan forskning och det är statistiskt säkerställt att relationerna, fransett de tre relationer som vi uppgett ovan, har visat sig vara korrekta i Agarwal och Prasads (1998) undersökning, anser vi att vi har hittat relevanta faktorer för att besvara vår frågeställning.

Övriga fördelar med att använda Agarwal-Prasad-modellen har varit att uppsatsens empiri-, analys- och diskussionsavsnitt har fått en bra struktur. Med hjälp av modellen

## 2 Metod

---

blev analysen dessutom väldigt organiserad och vi kunde använda modellen som ett verktyg för att besvara vår frågeställning.

En nackdel med att använda modellen är att vi har blivit lite låsta av modellens faktorer och hade svårt att placera in material som kom upp under intervjuerna inom modellens olika delar. Genom att vi avvägt och placerat in de svar som är relevanta under de rubriker där de passar bäst anser vi att vi har hanterat detta problem på ett bra sätt. En annan nackdel är att modellen är framtagen för att undersöka individens uppfattningar och inte företags, men på det sätt som frågorna ställts i Agarwal och Prasads (1998) undersökning anser vi att det går lika bra att använda frågorna för att undersöka företag.

### 2.4 Insamling av primärdata

För insamling av primärdata använde vi oss av semistrukturerade intervjuer. Vi använde denna form av intervju på grund av att vi ville ha frågor att utgå från under intervjun men ändå ha en möjlighet att ställa följdfrågor för att få ut så mycket väsentlig information ur intervjun som möjligt, något som är genomförbart med en semistrukturerad intervju (Bryman, 2001; Kvale, 1996). Eftersom vi hade en intervjuguide att utgå ifrån visste vi vilka områden intervjuerna skulle behandla, vilket gjorde att vi kunde styra tillbaka intervjun på rätt spår om intervjupersonen tog upp oväsentliga saker. I och med att semistrukturerade intervjuer får formen av ett samtal tror vi att respondenterna kände sig mer bekväma och avslappnade, vilket kan ha bidragit till att intervjuerna flöt på smidigt.

På grund av geografiskt avstånd var två av våra fyra intervjuer telefonintervjuer och de andra två besöksintervjuer. Detta kan ha medfört skillnader mellan intervjuerna då det i en telefonintervju inte finns faktorer hos intervjuaren som påverkar respondentens svar (Bryman, 2001). Det som är negativt med telefonintervjuer enligt författaren är att intervjuaren inte kan se på ansiktsuttryck om respondenten förstått frågan. Eftersom vi skickade ut sammanfattningar av intervjuerna till respondenterna och fick dem godkända kan vi dra slutsatsen att vi har tolkat respondenterna rätt.

#### 2.4.1 Intervjuguide

Vår intervjuguide (se bilaga 1) har flera delar. Den första delen, verksamhet och individ, har vi tagit fram för att få en bild av personen vi intervjuar och det företag som personen arbetar på. Första frågan är till stor del för att få respondenten att börja prata om något som är bekant och på så sätt få bort eventuell nervositet och skapa en avspänd stämning. De övriga frågorna i denna del är framtagna för att kunna sätta in hans eller hennes svar i ett sammanhang (Bryman, 2001).

Den andra delen består av information om företagets CAD-system och hur lagring sker från CAD-systemet. Detta använder vi för att kunna avgöra till exempel om CAD-systemet är objektorienterat eller inte samt vilket lagringssätt som används och

## 2 Metod

---

om både objektorienterade delar och relationsorienterade delar används i databaser som stödjer båda.

Resterande delar är framtagna från frågorna som Agarwal och Prasad (1998) har använt i sin artikel. Frågorna har anpassats för att stödja forskning om objektorienterade databaser. Eftersom vår undersökning är kvalitativ har vi kompletterat frågorna med följdfrågor för att ta reda på varför respondenten har svarat på det sätt som denne har gjort.

### 2.4.2 Urval

Vi ville undersöka CAD-företag som inte använder sig av objektorienterade databaser och därmed är vad vi anser som mest typiskt. Enligt Jacobsen (2002) är det negativa med denna typ av urval att det inte finns några garantier för att det vi tror är det typiska verkligen är det. Vi har dock i vår undersökning stöd från teorin att användning av objektorienterade databaser generellt sett inte är det mest förekommande.

Vi valde inte ut företagen slumpmässigt utan sökte på Gula Sidorna och Cadtorget (2006) efter företag som utvecklar CAD-system och har verksamhet i Sverige för att sedan ringa upp dessa. Anledningen till att det blev just fyra företag är för att detta var det antal företag vi fick tag på där det fanns anställda som var insatta i hur de lagrar data i sina system och vad som påverkat valet av lagringssätt. Ett icke-sannolikhetsurval som detta kan medföra att vi får ett systematiskt snett urval vilket gör att vi inte kan generalisera vårt resultat (Jacobsen, 2002). Generalisering är dock inget mål i denna uppsats utan vi vill snarare frambringa klarhet och kunskap om det fenomen som undersökts.

När det däremot gäller val av respondenter har vi använt oss av urvalet information (Jacobsen, 2002) och försökte därmed få tag på den person på företaget som hade information och kunskap om lagring av data i företagets CAD-system och vad som påverkat det valet. Det negativa med detta urval är enligt författaren att vi inte vet på förhand om de respondenter vi fått tag på verkligen har den information och kunskap som krävs. En av respondenterna visste ingenting om objektorienterade databaser. Detta har resulterat i att denna respondent inte kunnat svara på alla frågor i intervjuguiden. Vi har hanterat detta med hjälp av undersökningsmodellens kommunikationskanaler.

Det visade sig i vår undersökning att ingen av respondenterna hade någon erfarenhet om objektorienterade databaser, vilket kan förklaras med att de inte har infört objektorienterade databaser. Eftersom vår frågeställning handlar om varför objektorienterade databaser inte införts så anser vi att det inte har påverkat vår forskning negativt. Troligen hade vårt resultat förändrats om vi hade intervjuat någon som har erfarenhet av objektorienterade databaser, men vi kan inte dra någon slutsats om hur resultatet hade förändrats.

## 2 Metod

---

### 2.4.3 Genomförande

Den information som respondenten fick vid bokningen av intervjun var att vi studerar vid institutionen för informatik vid Lunds Universitet och att vi skriver en magisteruppsats om lagring av data i CAD-system. Dessutom fick de information om att intervjun beräknades ta högst en timme och att vi skulle behandla deras svar konfidentiellt.

Innan vi genomförde intervjun berättade vi för respondenten vad vår undersökning handlar om och frågade om det gick bra om vi spelade in intervjun. Vi berättade även att vi skulle behandla intervjun konfidentiellt så till vida att vi inte skriver med företagets namn, respondentens namn eller information som gör att det är lätt att spåra företaget eller respondenten i uppsatsen. Detta i enlighet med Trosts (1997) råd att en intervjuare aldrig ska skriva namnet på respondenterna i sin uppsats även om denne har fått ett godkännande från respondenten. Vi tror att detta medförde att intervjupersonen kände sig tryggare i intervjusituationen.

Under intervjun följde vi intervjuguiden, men kastade om och hoppade över frågor när det behövdes. Vi ställde även följdfrågor där vi ansåg att vi behövde ytterligare information eller enbart för att få bekräftelse på att vi hade förstått det respondenten sa. Genom att spela in intervjun behövde vi inte anteckna allt, vilket innebar att vi kunde följa Kvales (1996) rekommendation att fokus ska ligga på respondenten. Problemet med att spela in intervjun är att respondenten kan känna sig hämmad (Bryman, 2001), men vi tror i likhet med författaren att respondenterna vände sig efter ett tag och glömde bort att de spelades in.

I slutet av intervjun frågade vi om respondenten hade något övrigt att tillägga. Detta är bra dels för att det är en lagom mjuk avslutning på intervjun dels för att, som Jacobsen (2002) och Kvale (1996) tar upp, denna typ av frågor fungerar som säkerhetsventil som fångar upp information som annars inte skulle ha framkommit. Respondenterna fick information om att vi skulle skicka över en sammanfattning av intervjun, med andra ord vår tolkning av det respondenten sagt, för att respondenten skulle kunna kontrollera att vår tolkning var korrekt.

Efter varje intervju transkriberade vi det inspelade materialet. Anledningen till att vi ville transkribera materialet är för att det är betydligt lättare och smidigare att analysera utifrån ett textdokument istället för att hoppa fram och tillbaka i det inspelade materialet.

## 2 Metod

---

### 2.5 Analys

#### 2.5.1 Kodning av intervjumaterial

Efter transkriberingen kodade vi transkriptet (se bilaga 2 för koder). Det gjordes i största möjliga mån innan kommande intervju, detta för att kunna få uppslag till viktiga områden och följdfrågor. Kodningen innehåller, som Miles och Huberman (1994) skriver, kontexten från transkriptet snarare än orden därifrån. Utöver kodning av kontext har vi använt Miles och Hubermans (1994) rekommendation att utnyttja marginalanteckningar för att visa sammanhanget. Kodningen har gjorts av den person som inte transkriberat intervjun, varefter den person som transkriberat kontrollerar att båda har tolkat transkripten på samma sätt och att allt väsentligt finns med. I de fall där vi tolkat något olika har vi diskuterat och kommit fram till en gemensam tolkning.

När vi hade kodat alla transkript och vi var nöjda med kodningen gick vi igenom transkripten en gång till med koderna som utgångspunkt. Vi tror att detta kan ha medfört att vi undviker att missa information som borde finnas med i kodningen.

#### 2.5.2 Matris

Vi har använt oss av matriser som hjälpmedel för att analysera intervjuerna. De två matriser som Miles och Huberman (1994) nämner som var lämpliga i vår undersökning är Checklist matrix och Case dynamics matrix. Checklist matrix används när forskaren ska undersöka en stor variabel eller ett huvudsakligt intresseområde. I och med att vi var intresserade av en variabel i vår frågeställning, vilka faktorer som har en avgörande roll, ansåg vi att det är möjligt att använda denna sortens matris.

Det andra alternativet, att bygga en Case dynamic matrix, används för att länka data med förklaringar (Miles och Huberman 1994). Matrisen används alltså främst för att visa förklaringar som är relevanta för en specifik fråga.

Vi har inte använt någon av dessa matriser rakt av utan har utgått från tanken bakom dem för att skapa en egen med hjälp av de riktlinjer Miles och Huberman (1994) föreslår. I matrisen vi framställde kunde vi både analysera variabler för sig och ställa dem mot varandra och se om något i matrisen förklarade något annat, alltså en blandning av tankarna bakom Checklist matrix och Case dynamics matrix

Ett utdrag ur matrisen visas i tabell 2.1. Observera att tabellen är bearbetad för att endast innehålla vad som behövs för att förklara hur vi har använt matrisen. Matrisen vi använde innehåller både mer data och är, med hjälp av färger, mer överskådlig.

## 2 Metod

Tabell 2.1: Utdrag ur matrisen

	<b>Avsikter</b>	<b>Medvetenhet</b>
<b>Bengt</b>	DB ej fokus nu, nöjda som det är (I2R52) {3}	ODB ej konkurrensfördel eftersom prestandafrågor minskat i betydelse (I2R60) [Prestanda] {3}
<b>Calle</b>	Ja, om snabbhet tillräcklig för kunder. Men finns problem med installation och drivande av DB (I3R28) [Prestanda] {7, 8}	ODB viktig innovation eftersom utveckling av OO. Snabbhet är dock ett problem. (I3R32) [Lagring] {7}

Utifrån koderna i transkripten la vi in alla data som vi ansåg vara relevanta i matriserna, vi sammanfattade det för att kunna få en översikt över vad som låg i matriserna. I samband med detta skrev vi också referenser till den unika kod som varje bit rådata har i transkriptet, det som står inom parantes i tabell 2.1. Det har inneburit att vi, vid behov, har haft en möjlighet att gå tillbaka till transkripten för att se mer exakt vad som framkommit. Från denna matris skapade vi sedan de modeller som kan ses under 5.1 Analys utifrån Agarwal-Prasad-modellen. Vi utgick från modellen vi använt och fyllde på tolkningen av det som står i matrisen. Arbetet med att överföra från matrisen till modellerna gjordes för att vi skulle kunna göra en bra analys utifrån modellen som helhet, vi fick en bättre överblick på det sättet.

Efter det kopplade vi ihop de olika delarna i teorin med matrisen, rubrikerna i teorin motsvarar det som står inom hakparenteser i tabell 2.1. Diskussionen av de olika delarna i modellen har sedan skett utifrån denna länkning till teorin. Som ett sista steg har vi utifrån matrisen försökt att se om vi har fått fram något i en intervju som kan förklara något annat i samma intervju. I matrisen har vi markerat detta med siffror på båda ställena (inom måsvingar i tabell 2.1), hur vi tolkar sammanhanget har vi antecknat för sig med referens till matrisen. Observera att vi inte tagit någon hänsyn till modellen i detta steg av analysen, därmed har vi kunnat se sammanhang som både går inom modellens ram och utanför den.

Detta tillvägagångssätt har hjälpt oss att analysera och diskutera från tre olika utgångspunkter: Modellen har tillåtit oss se hur dess relationer kan besvara vår forskningsfråga, den vertikala analysen har tillåtit oss att jämföra de olika respondenternas svar inom samma område och den horisontella analysen har givit oss möjlighet att förklara respondenternas svar oberoende av modellen. Därmed anser vi att vi har lyckats använda flera av Miles och Hubermans (1994) taktiker för att skapa mening, nämligen att se mönster och teman, göra jämförelser, se förhållanden samt se relationer mellan variabler. I och med denna grundliga analys anser vi att vi med hög sannolikhet har upptäckt alla viktiga faktorer och relationer från vår empiri.



## 2 Metod

---

### 2.6 Kvalitet

#### 2.6.1 Reliabilitet

För att öka reliabiliteten i vår undersökning har vi dokumenterat processen. Något som minskar reliabiliteten enligt Svenning (2000) är om vi tolkar intervjuresultaten fel. När det gäller personliga intervjuer finns det dessutom alltid utrymme för subjektiva tolkningar vilket kan leda till ett annorlunda och ogiltigt resultat. Vi har därför varit väldigt noggranna med att ställa följdfrågor under intervjuerna då vi varit osäkra på om vi tolkat respondentens svar rätt. När vi hade transkriberat och sammanfattat intervjuerna visade vi respondenterna en sammanfattning av transkriptet för att få verifierat att vi har tolkat svaren rätt.

#### 2.6.2 Validitet

För att upprätthålla vår undersöknings validitet har vi försökt ha en röd tråd genom uppsatsen där vi hela tiden har försökt utgå från vår frågeställning och vårt syfte. Dessutom bad vi intervjupersonen utveckla sitt svar när det var nödvändigt för att vi ska kunna tolka dennes svar så riktigt som möjligt, detta i enlighet med Kvales (1996) rekommendationer. Under intervjun utgick vi från vår intervjuguide vilket vi anser gjorde att intervjun inte svävade iväg för mycket.

#### 2.6.3 Objektivitet

Enligt Kvale (1996) är alla tolkningar av det empiriska materialet subjektiva bedömningar. Vi hade för avsikt att i vår undersökning vara så objektiva som möjligt, vi anser dock inte att det är möjligt att vara helt objektiv då människor enligt oss har en tendens att göra egna tolkningar och subjektiva bedömningar. Vi tror dock att det har varit en fördel att vi är två personer med olika tolkningssätt vilket gör att vi har kunnat diskutera och komma fram till en saklig bedömning.

### 2.7 Etik

Det finns enligt Kvale (1996) tre stora delar för att säkra de etiska aspekterna mot sina respondenter. Det är informerat samtycke, säkra konfidentialiteten och överväga konsekvenser. Vi har hanterat informerat samtycke genom att ge information om det generella syftet med undersökningen och meddelat att de deltar frivilligt. Vi har även, som framgår i avsnitt 2.5.3, givit respondenterna möjlighet att komplettera sina svar samt att ställa frågor till oss. Samtycke till intervjun har givits av respondenterna själva, de har haft möjlighet att i sin tur söka samtycke hos någon annan på företaget om de funnit att det behövts.

## **2 Metod**

---

Säkra konfidentialiteten är den andra etiska aspekten enligt Kvale (1996). Vi har säkrat konfidentialiteten genom att i vårt arbete använda fingerade namn. För att i största möjliga mån se till att de verkligen är konfidentiella har vi valt att inte publicera transkripten eller något annat som skulle kunna bidra till att det går att spåra respondenterna.

Risken för konsekvenser är den sista etikaspekten som Kvale (1996) tar upp. Risken för yttre faktorer är minimerad i och med att intervjuerna har behandlats konfidentiellt. Den risk vi kan se är att respondenten, när rapporten är färdig, kommer fram till att de inte arbetar på ett optimalt sätt.

För att ytterligare öka kvaliteten och se till att de etiska aspekterna uppfyllts har vi skickat en sammanfattning av vår tolkning av intervjun till respondenten för godkännande. Genom att vi lagt ner relativt mycket arbete på tolkningen av vad de har sagt har ingen respondent nekat oss att använda intervjun.

### 3 Litteraturgenomgång

---

## 3 OBJEKTORIENTERADE DATABASER

---

*I detta kapitel beskriver vi först objektorienterade databaser, deras funktionalitet, fördelar och nackdelar. Vi tar sedan upp skillnaderna mellan objektorienterade databaser och relationsdatabaser samt beskriver vilka applikationsområden som är bäst lämpade för objektorienterade databaser. Därefter beskriver vi det applikationsområde vi valt att undersöka.*

---

### 3.1 Objektorienterade databaser

Vanligtvis vid utveckling av mjukvara finns det tre olika delar: analys och design, programmering samt databasdefinition och databasaccess (Loomis, 1995). Traditionellt har olika modeller använts för dessa delar, vilket innebär att det blir problem att gå mellan de olika delarna. Ett av problemen som författaren tar upp är att det uppkommer buggar mellan programmeringen och databasen eftersom de dels använder olika språk och dels för att programmeringsspråket ofta är objektorienterat och databasspråket är deklarativt. För att förhindra problem bör väggarna mellan de tre delarna tas bort, något som objektorienterade databaser har försökt lösa. Det finns enligt Loomis (1995) tre standardanledningar för utvecklare att använda objektorienterade databaser:

- Beslut har tagits att använda objektteknologi och objektorienterade databaser är en del av det beslutet.
- Databasapplikationen är för komplex för att hanteras på ett bra sätt av relationsmodeller.
- Organisationen har distribuerade applikationer som kan dra fördel av de distribuerade särdragen och den samverkande funktionaliteten för objektorienterade databaser.

Objektorienterade databashanterare kombinerar koncepten från objektorienterad programmering med egenskaperna hos ett databssystem (Wretling & Karlquist, 1990). Det är med objektorienterade databaser möjligt med presentation i form av komplexa datastrukturer och typhierarkier. Det går även att definiera datatyper där datastruktur och procedurdefinition kombineras (Wretling & Karlquist, 1990).

#### 3.1.1 Objektorienterad funktionalitet

Objektorienterade databaser har till stor del samma funktionalitet som objektorienterad programmering (Elmasri och Navathe, 2004). Bland annat uppger författarna att objektorienterade databaser stödjer inkapsling, typ- och klasshierarkier med arv samt operatoröverlagring.

### 3 Litteraturgenomgång

---

#### 3.1.2 Lagring

Objektorienterade databaser lagrar persistenta objekt permanent i sekundärminne och tillåter att många program och applikationer har tillgång till objekten (Elmasri & Navathe, 2004). Detta kräver bland annat indexeringsmekanismer, samtidighetskontroll och återhämtning. Enligt författarna är ett mål med objektorienterade databaser att bibehålla en direkt korrespondens mellan verkliga objekt och databasobjekt så att objekten inte förlorar sin integritet och identitet och enkelt kan identifieras och användas.

#### 3.1.3 Språk

Object Database Management Group (ODMG) var en grupp som bildades av leverantörer av objektorienterade databaser för att skapa standarder (Elmasri & Navathe, 2004). Object Query Language (OQL), som ODMG har utvecklat, är för tillfället standardspråket för objektorienterade databaser, men det är enligt Connolly och Begg (2005) inte en officiell standard.

OQL är, enligt Connolly och Begg (2005), ett språk som tillåter deklarativ access till data i objektorienterade databaser genom att använda en syntax som liknar SQL. Författarna uppger att det i OQL ingår funktionalitet för att hämta data från databasen, det innebär att select-, projekt- och join-frågor hanteras. Det tillåter dock inte uppdateringar av data, det görs enligt Connolly och Begg (2005) istället i det objektorienterade programmeringsspråket.

För att hantera problematiken som uppkommer när både databasprogrammerare och applikationsprogrammerare behöver programmera mot databasen anser Loomis (1995) att det ska finnas multipla interface. Det innebär enligt Loomis (1995) att databasprogrammerare programmerar i ett SQL-liknande språk medan applikationsprogrammerare kan programmera i det objektorienterade språk de använder, till exempel Java. Detta uppfylls inte av OQL.

#### 3.1.4 Underhåll

I objektorienterade databaser kan verksamhetsregler kodas som delar av metoder enligt Niyomthym och Chittayasothorn (2003). Så när reglerna ändras är det bara metoderna som behöver modifieras och inte applikationsprogrammet. Författarna anser att detta är en anledning till att objektorienterade databaser är lätta att underhålla. Eftersom alla attribut är inkapslade menar de att en nackdel med objektorienterade databaser är att det inte går att använda ad-hoc-frågor.

Med en bra design är databasen i ett objektorienterat databassystem en direkt mappning till objekt i verkligheten (Zhu, Crouch och Tabrizi, 2005). Det gör det lätt

### 3 Litteraturgenomgång

---

för användare att förstå databasen och färre fel uppstår vid hantering av systemet. Därmed blir systemet även lättare att underhålla.

#### 3.1.5 Modell

I system som använder en objektorienterad databas finns det ett starkt samband mellan applikationens datamodell och databasens datamodell (Devarakonda, 2001). Det starka bandet resulterar enligt författaren i mindre kod, mer naturliga datastrukturer samt bättre underhållbarhet och kodåtervinning. Eftersom utvecklaren bara behöver koncentrera sig på en enda datamodell kan det dessutom enligt Zhu et al. (2005) resultera i förbättrad design/arkitektur. Loomis (1995) anser till och med att huvudanledningen att använda en objektorienterad databas är att det går att använda en objektmodell istället för en relationsmodell.

#### 3.1.6 Annan funktionalitet

Utöver den funktionalitet som beskrivs ovan så stödjer objektorienterade databashanterare även samma funktionalitet som relationsdatabashanterare (Bancilhon, 1996; Loomis, 1995), det vill säga:

- transaktionshantering och samtidighet
- recovery
- distribuering
- versionshantering
- utveckling av schema
- frågehantering
- åtkomstkontroll

#### 3.1.7 Säkerhet

Enligt Cattell (1994) finns det två sorters säkerhet när det gäller databaser, att säkerställa att en användare inte kan förvränga den fysiska datastrukturen samt att en användare inte kan komma åt data utan att ha tillstånd. Vi kommer nedan behandla båda dessa aspekter av säkerhet.

Säkerhet hanteras inte på samma sätt i objektorienterade databaser som i relationsdatabaser (Cattell, 1994). Objektorienterade databaser cachar ofta objekt från databasen i samma datautrymme som applikationsprogrammet använder och därmed är det möjligt att ändra den fysiska datastrukturen från programspråket. Författaren uppger att det, genom att använda ett säkert programmeringsspråk, går att generera kod för att kontrollera detta.

### 3 Litteraturgenomgång

---

Connolly och Begg (2005) menar att objektorienterade databaser inte har tillräckliga säkerhetsmekanismer då de ofta inte tillåter accessrättigheter till individuella objekt eller klasser. Cattell (1994) uppger däremot att det går att reglera säkerheten genom att basera den på fysiska filer där användarens eller gruppens rättigheter reglerar filåtkomsten. Det går på detta sätt att lagra objekt, attribut och metoder i olika filer och därmed ge olika rättigheter även inom ett objekt. Författaren menar att detta tillvägagångssätt kan passa de flesta applikationer.

## 3.2 Skillnader mellan objektorienterade databaser och relationsdatabaser

### 3.2.1 Verktyg

Det saknas mognad hos de tillgängliga kommersiella verktygen för att utveckla och underhålla objektorienterade databaser (Beck, 2001). Många av verktygen som används av relationsdatabasutvecklare såsom schemadesignverktyg, kodgeneratorer, och användargränssnittsverktyg finns inte för objektorienterade databaser i samma utsträckning.

### 3.2.2 Interaktion mellan applikation och databas

I en relationsdatabas finns det en mur mellan databassystemet och applikationsprogrammet (Bancilhon, 1996). Enligt författaren innehåller databasen enbart persistent data medan applikationsprogrammet enbart manipulerar transient data. Interaktionen mellan applikationen och databasen upprättas alltid genom read- och write-operationer. Den objektorienterade databasen löser enligt författaren detta problem genom att använda persistent programmering. Med detta angreppssätt finns enligt författaren ingen separation mellan databasen och applikationen, applikationsprogrammet manipulerar både transient och persistent data. Persistent data uppdateras i transaktioner och modifieringarna är persistenta när de genomförts.

### 3.2.3 Objektidentifierare

Objektorienterade databaser har en unik systemgenererad objektidentifierare för varje objekt, något som skulle kunna jämföras med relationsdatabasens primärnyckel (Elmasri & Navathe, 2004). Detta förbättrar enligt Devarakonda (2001) dataaccessprestandan. Värdet av objektidentifieraren är inte synligt för den externa användaren och att ändra värdet är inte tillåtet (Elmasri & Navathe, 2004). Detta för att bevara identiteten hos det verkliga objektet som representeras. I objektorienterade databaser har objekt enligt författarna även en komplex objektstruktur för att kunna innehålla all nödvändig information som beskriver objektet till skillnad från relationsdatabasen där

### 3 Litteraturgenomgång

---

information om objektet är spritt över olika tabeller som leder till saknad av korrespondens mellan verkliga objekt och databasobjekt.

#### 3.2.4 Design av databasen

Största skillnaden mellan objektorienterad databasdesign och relationsdatabasdesign är enligt Elmasri och Navathe (2004) hur relationer hanteras. I objektorienterade databaser hanteras relationer genom att ha relationsegenskaper eller referensattribut som inkluderar identifierare av det relaterade objektet. Både enkla referenser och samlingar av referenser är tillåtna. I relationsdatabaser är relationer mellan tupler specificerade av attribut med matchande värden. Begränsningen här är att det bara är tillåtet med enkla referenser. Multivärdesreferenser är inte tillåtna, det vill säga att en många-till-många relation måste presenteras som en separat tabell.

En objektorienterad databashanterare bygger verkliga relationer mellan objekt medan en relationsdatabashanterare måste använda join-operationer (Loomis, 1995). Det innebär att en objektorienterad databashanterare normalt presterar bättre med applikationer som kräver korsning av relationer. Loomis (1995) uppger att både en-till-en, en-till-många och många-till-många relationer kan skapas i en objektorienterad databashanterare utan att det behövs någon join-operation. Dock behöver en många-till-många-relation med attribut hanteras, och det görs ofta genom att skapa en klass för relationen, med andra ord på samma sätt som relationen hanteras i en relationsdatabas (Elmasri & Navathe, 2004). Det kan enligt författarna även hanteras genom att placera attributet i en av klasserna, vad som väljs är upp till designern.

En annan stor skillnad mellan designen av relationsdatabaser och objektorienterade databaser som tas upp av Elmasri och Navathe (2004) är hur arv hanteras. I objektorienterade databaser är dessa strukturer inbyggda i modellen så att mappningen genomförs genom att arvsconstructorn används, med andra ord får en klass alla attribut och metoder från superklassen. I relationsdatabaser finns dock inte arv med i modellen utan måste hanteras på annat sätt. En del av de relationsdatabassystem som utökas med objektfunktionalitet försöker dock ha med denna funktionalitet.

#### 3.2.5 Prestanda

Tester har visat att vid den arbetsbörda som uppkommer när det handlar om applikationer som hanterar komplexa objekt så har objektorienterade databaser bättre prestanda än relationsdatabaser (Cattell & Skeen, 1992). Att prestandan är bättre hos objektorienterade databaser när det gäller denna typ av applikationer kan enligt Bancilhon (1996) enkelt förklaras:

- Databasmotorn är särskilt designad för att stödja objektmodellen. Den erbjuder stöd för komplexa objekt och undviker join-operationen när den bygger dessa objekt.

### 3 Litteraturgenomgång

---

- Objektorienterade databaser är designade med en klientcentrerad arkitektur. De överför så mycket som möjligt av arbetsbördan till klienten och stödjer klientcachning av objekt.

Enligt Beck (2001) är det mindre effektivt att lagra och ta fram objekt från och till en relationsdatabas eftersom objektet måste brytas ner och sättas ihop genom att mappa till relationsform. Nedpackning och upppackning av objekt kan bli kostsamt då det tar lång tid (Zhu et al., 2005). Eftersom objektorienterade databashanterare sparar komplexa objekt och deras förhållanden direkt kan de enligt författarna därför vanligtvis prestera bättre än relationsdatabashanterare. Att relationsdatabaser är långsammare beror enligt Loomis (1995) på att relationsdatabashanterare måste joina tabeller och sortera resultatet. Dessa två operationer är bland de långsammaste operationerna som en relationsdatabashanterare utför.

Prestandan vid frågehantering när stora mängder objekt ska hämtas och undersökas sekventiellt är för objektorienterade databaser däremot ganska låg (Beck, 2001). Till skillnad från relationsdatabashanterare är frågeoptimering väldigt komplext i objektorienterade databashanterare (Devarakonda, 2001).

#### 3.2.6 Lättanvändbarhet

Objektorienterade databashanterare är enligt Zhu et al. (2005) lätta att använda vilket kan spara mycket tid när en applikation ska distribueras inom organisationen. De menar att IT-användares inlärningskurva för den objektorienterade databashanteraren de i sin undersökning tittade på var mycket lägre än för relationsdatabashanteraren. I denna typ av system behöver användarna inte tänka på hur de ska designa databasen, de behöver bara definiera klasserna och spara objekten vid behov. Därmed besparas användare, enligt författarna, att lära sig hur uppdelning av komplexa klasser ner i olika tabeller i relationsdatabasen går till. Beck (2001) menar däremot att objektorienterade databaser är svårare att lära sig eftersom de är mer komplexa än relationsdatabaser. Chung, Lin och Chang (1995) uppger att relationsmodellen är lätt att förstå och att den har en teoretisk grund, detta anser de kan förklara succén med relationsdatabaser.

#### 3.2.7 Sammanfattning

Tabell 3.1 är en sammanfattning av de skillnader mellan objektorienterade databaser och relationsdatabaser som framkommit ovan.



### 3 Litteraturgenomgång

Tabell 3.1: Skillnader mellan objektorienterade databaser och relationsdatabaser

	<b>Objektorienterade databaser</b>	<b>Relationsdatabaser</b>
<b>Verktyg</b>	- Saknar mognad	- Schemadesignverktyg, kodgeneratorer, användargränssnittsverktyg
<b>Interaktion mellan applikation och databas</b>	- Persistent programmering innebär att det inte finns någon separation mellan databasen och applikationen	- Mur mellan applikationsprogrammet och databassystemet
<b>Objektidentifierare</b>	- Objektidentifierare - Komplex datastruktur	- Primärnyckel - Information om objektet spritt över olika tabeller
<b>Design av databasen</b>	- Relationer genom relationsegenskaper eller referensattribut - Verkliga relationer - Arv stöds i modellen	- Relationer genom attribut med matchande värden  - Join-operationer - Arv finns ej i modellen
<b>Prestanda</b>	- Bättre prestanda vid applikationer som hanterar komplexa objekt - Frågeoptimering väldigt komplext	- Mindre effektivt  - Bättre prestanda när stora mängder objekt ska hämtas och undersökas sekventiellt
<b>Lättanvändbarhet</b>	- Inlärningskurvan lägre (Zhu et al. 2005)	- Lättare att lära sig (Beck 2001) - Lätt att förstå (Chung et al. 1995)

### 3.3 Applikationsområden för objektorienterade databaser

Vilken databas som är bäst att använda beror på applikation (Beck, 2001). Datastrukturer som behövs för stora dokument hanteras mest effektivt med hjälp av objektorienterade databashanterare. Detta är enligt författaren väldigt svårt att hantera med hjälp av relationsdatabashanterare och det är dessutom omöjligt att uppnå samma prestanda.

Andra applikationsområden där objektorienterade databashanterare är att föredra är distribuerade datormiljöer som involverar samarbete, långa transaktioner och versionhantering samt områden som har högprestandabehov med förutsägbara åtkomstmönster och begränsade arbetsomgångar (Loomis, 1995). En annan anledning som Loomis (1995) nämner att välja objektorienterade databaser är när applikationen är utvecklad med objektorienterad programmering och har många användare som behöver dela data.

Även när det gäller styrning har objektorienterade databaser en fördel (Loomis, 1995). Styrning innebär att något innehåller, har, består av eller är uppbyggt av något annat, till exempel består en cykel av delar (hjul, ram, styre och så vidare). För att hantera detta i en objektorienterad databas används enligt Loomis (1995) containers där containern för cykeln innehåller de olika delarna. Containern har även vissa funktioner, till exempel hur monteringen ska ske. I en relationsdatabashanterare kan containern sparas, men den interna strukturen i containern försvinner, vilket innebär

### 3 Litteraturgenomgång

---

att kodning och avkodning läggs på programmet och inte databasen. I en objektorienterad databashanterare kan däremot databasen förstå strukturen i containern, vilket innebär att programmet inte behöver hantera detta (ibid.).

Relationsdatabasen är enligt Beck (2001) att föredra när data är enhetliga, upprepande och kan stoppas i tabeller. De klarar dock inte av kraven från de lite nyare applikationsområdena såsom Computer Aided Design (CAD), Computer Aided Manufacturing (CAM), Computer Aided Software Engineering (CASE), Computer Aided Engineering (CAE), geografiska informationssystem (GIS), kunskapsbaserade system (KBS), kontorsautomationssystem (KAS), vetenskap, medicin, tidsserier och multimedia (Date, 2004; Delobel, Lécluse & Richard, 1995; Loomis, 1995). Dessa applikationer kräver databaser som kan lagra stor mängd data som komplexa strukturer samt kunna ge stöd för de datatyper som inte finns i konventionella databashanterare (Wretling & Karlquist, 1990).

Objektmodellen ger enligt Chung et al. (1995) användare ett mer naturligt och flexibelt sätt att modellera komplexa datastrukturer som används i multimedia, stora dokument och CAD-applikationer.

#### 3.3.1 Komplexa objekt

En relationsdatabas kan endast hantera de inbyggda datatyperna medan en objektorienterad databas har fördefinierade datatyper samt egna datatyper som utvecklaren lägger till (Loomis, 1995). Dessa kallas komplexa datatyper eller komplexa objekt. Elmasri och Navathe (2004) definierar två typer av komplexa objekt, strukturerade och ostrukturerade. Enligt författarna är ett strukturerat komplext objekt uppbyggt av komponenter och den objektorienterade databashanteraren känner till objektstrukturen. Ett ostrukturerat komplext objekt är vanligtvis en datatyp som kräver stort lager, till exempel en datatyp som representerar en bild eller ett stort textobjekt. Objekten är ostrukturerade i den mening att databashanteraren inte vet hur deras struktur ser ut. Det är bara applikationen som använder objekten som kan tolka deras mening (Elmasri & Navathe, 2004).

I de fall där fler än de inbyggda datatyperna används (till exempel HTML-filer och PDF-filer) lagrar relationsdatabaser delar av informationen medan resten lagras i filer (Beck, 2001). I objektorienterade databaser däremot kan vilken typ av data som helst lagras vilket gör att lagring utanför databasen inte behövs. Det innebär att en objektorienterad databashanterare kan använda alla typer som programmeringsspråket kan hantera, till exempel bilder (Loomis, 1995). Enligt författaren är komplexa data, som står i ett högt inbördes förhållande och/eller inkluderar multimediatyp, lämpligt för objektorienterade databaser.

### 3 Litteraturgenomgång

---

#### 3.4 CAD

CAD (Computer aided design) innebär att datorer används i konstruktions- och ritarbete. Det datorbaserade system som används till detta kallas CAD-system (Pärletun, 2006). Första utvecklingen inom CAD-system skedde i början av 1960-talet och under 1980-talet nådde tekniken ut till både små och medelstora företag.

CAD-system används inom flera branscher såsom verkstadsindustrin, elektronikområdet och byggbranschen. I verkstadsindustrin används CAD-system för framtagning av ritningar över både detaljer och hela konstruktioner samt layout-, el- och offertritningar (Pärletun, 2006). Elektronikområdet använder sig av CAD-system för att konstruera kretskort och integrerade kretsar. Inom byggbranschen används det för arkitekt- och konstruktionsritning samt för el- och VVS-projektering.

Konstruktionsarbetet kan utföras i två eller tre dimensioner (Pärletun, 2006). Vid tvådimensionell ritning byggs ritningen upp med hjälp av räta linjer, cirkelbågar, kurvor och text medan ritningar i tre dimensioner byggs med hjälp av kurvor, ytor eller solida kroppar.

CAD är ett tillämpningsområde där en elektronisk krets kan bestå av flera delar, där vissa av delarna kan vara av samma typ och ha samma egenskaper (Silberschatz, Korth & Sudarshan, 2002). Det går då att referera till de delar som ingår som i sin tur kan referera till delar som ingår och så vidare. Om en relationsdatabashanterare skulle användas för att få ut hur mycket ström som förbrukas är det enligt författarna väldigt svårt att skriva SQL-kod för detta. Det krävs då dessutom kunskap om hur den interna strukturen i databasen ser ut. I en objektorienterad databashanterare används istället en metod som är implementerad i alla klasser, denna räknar ut förbrukningen utan att det behövs kunskap om objektens interna struktur (Silberschatz et al., 2002).

## 4 Empiri

---

# 4 UNDERSÖKNING

---

*Empirikapitlet innehåller en presentation av de företag och respondenter vi har valt att intervjua samt en presentation av det resultat intervjuerna gav. Resultatet är uppdelat utifrån intervjuguidens teman.*

---

## 4.1 Presentation av företag och respondenter

De fyra företag som respondenterna arbetar på finns antingen både i Sverige och någon annanstans i världen eller enbart i Sverige. Den typ av data de har behov av att lagra i sina CAD-system är till största del någon form av ritning eller bild. Alla fyra respondenter som intervjuades har en teknisk utbildning på högskolenivå. Nedan följer en kort presentation av varje företag.

### 4.1.1 Acad - Anders

På Acad lagras data från CAD-systemet med hjälp av en relationsdatabas och ett filsystem. Deras CAD-system är utvecklat i Java och HTML, kan hantera flera samtidiga användare och kan distribueras på olika datorer. Deras system måste hålla en hög säkerhet eftersom detta är ett krav från några av deras kunder. Anders vet inte vilka olika lagringssätt de valde mellan när de valde att använda en relationsdatabas. Han har ingen erfarenhet av och kännedom om objektorienterade databaser och har därmed inte kunnat svara på frågor om relativa fördelar, lättanvändbarhet, kompatibilitet och avsikter.

### 4.1.2 Bcad - Bengt

Bcads CAD-system lagrar data med hjälp av en relationsdatabas. Systemet är utvecklat i både C och C++, har flera samtidiga användare och är skalbart så att det kan ligga på flera maskiner, CPUer och diskar. Vid val av lagringssätt i sitt CAD-system valde Bcad mellan två relationsdatabaser av olika märken. Bengt har ingen personlig erfarenhet av objektorienterade databaser men de tittade på det för några år sen och kom fram till att de inte skulle ha någon nytta av en objektorienterad databas om inte hela CAD-systemets kod byttes ut så att det blev helt objektorienterat. Han har därmed kännedom om vad en objektorienterad databas är.

## 4 Empiri

---

### 4.1.3 Ccad – Calle

Lagring av data från Ccads CAD-system sker på fil. Deras lösning är inte uppbyggd för att stödja simultana användare. De programmerar nu för tiden enbart i C++, tidigare programmerade de dock i basic och assembler. Calle har ingen erfarenhet av objektorienterade databaser men har generell kännedom om vad det innebär.

### 4.1.4 Dcad – David

Dcads CAD-system lagrar sina data i ett filsystem. Vid utvecklingen av systemet var lagring på fil det enda sätt de kunde tänka sig att lagra sina data på och därför stod inte valet mellan olika alternativ. Dcad utvecklar sina system med programmeringsspråken C++, Delphi och Visual Basic. Många användare kan använda samma data i deras CAD-system samtidigt. Även David har generell kännedom om vad objektorienterade databaser innebär men har inte någon erfarenhet av dem.

## 4.2 Presentation av intervjuresultat

### 4.2.1 Relativa fördelar

Bengt anser att utvecklingen av deras CAD-system hade kunnat utföras på mindre tid om de valt en objektorienterad databas eftersom de då sluppit mellanskiktet i applikationen där de måste omvandla objekten för att de ska kunna lagras i tabeller. Både Bengt och Calle tycker dessutom att det hade varit lättare att utveckla systemet och Calle lägger till att det även skulle vara lättare att ändra eftersom ändringen då skulle bli lokal och inte påverkar andra objekt. Bengt tror dessutom att utvecklingen hade kunnat bli mer effektiv men tror dock inte att det hade påverkat produktiviteten på hela projektet eftersom databasen bara är en liten del. Enligt Bengt hade deras CAD-system varit snabbare med en objektorienterad databas eftersom mycket kod då skulle kunna rationaliseras bort och svåra och tidskrävande frågor skulle kunna undvikas.

David anser att de inte kunnat välja en objektorienterad databas istället för det sätt de lagrar på eftersom de då inte haft någon produkt. Deras kunder hade inte kunnat använda den eftersom de inte har några databaser. Han har därmed inte kunnat svara på frågorna om relativa fördelar.

### 4.2.2 Lättanvändbarhet

Bengt anser att relationsdatabaser är lättare att förstå och lära sig än objektorienterade databaser. Han tror dock att objektorienterade databaser är tydligare än relations-

## 4 Empiri

---

databaser. Calle tycker däremot att det är lättare att lära sig objektorienterade databaser men lägger till att det förutsätter att det finns en förståelse för objektorientering i stort. David anser inte att det spelar någon roll vad som är lättast att få att göra som man vill eller lära sig. De lyssnar på vad kunden vill ha och tittar på hur det ser ut på marknaden.

### 4.2.3 Kompatibilitet

Bengt tycker att objektorienterade databaser skulle passa Bcads arbetssätt bättre än relationsdatabaser eftersom det rent logiskt är lättare att tänka sig de saker de lagrar som objekt. Om de hade valt en objektorienterad databas istället hade de ändrat sättet CAD-systemet utvecklades på genom att de hade programmerat allt i C++. Calle anser inte att han kan svara på frågan om kompatibilitet då det blir för hypotetiskt. David tycker även här att det inte spelar någon roll med samma argument som ovan.

### 4.2.4 Benägenhet att prova på ny teknik

Enligt Anders ligger det i deras natur på Acad att testa nya saker, deras kunder förväntar sig det. Bengt, Calle och David menar alla tre att deras ekonomi reglerar hur mycket de kan experimentera. Bcad försöker dock experimentera i samarbete med högskolor medan Ccad enbart tar de vägar de vet är säkra. Alla tre anser även att det aldrig är bra att vara först med att använda ett verktyg och vill helst att buggar och problem ska vara utredda innan de använder sig av dem.

### 4.2.5 Medvetenhet

Bengt, Calle och David anser alla tre att objektorienterade databaser är en viktig innovation, David menar dock att det är viktigt på sikt. Bengt anser att det är en viktig innovation på grund av att det är en naturlig fortsättning på objektorienterad programmering. Calle tycker att det är en viktig innovation på grund av att objektorienterad utveckling är så vanligt förekommande. David tror att det kan ge helt andra möjligheter i och med att det går att få in all information i en databas.

Anders anser inte att objektorienterade databaser i sig skulle ge någon konkurrensfördel i deras bransch men menar att om objektorienterade databaser är något man skulle kunna differentiera sig med så skulle det kunna ge en konkurrensfördel. Ett annat sätt att differentiera sig är enligt Anders hur robust och felsäkert ett system är. Han anser att tillämpningen är det viktigaste och inte hur data lagras i systemet. Bengt tycker som Anders inte att det är en konkurrensfördel men uppger anledningen att trots att objektorienterade databaser är ”grymt snabba” så har prestandafrågan minskat i betydelse. Han anser att det är viktigare med ett effektivt gränssnitt. Bengt anser dock att det är lämpligt för Bcad att använda en

## 4 Empiri

---

objektorienterad databas eftersom deras utveckling numera sker i C++. David vet inte om det skulle vara lämpligt med en objektorienterad databas i Dcads CAD-system.

### 4.2.6 Kommunikationskanaler

Anders har inte fått information om objektorienterade databaser genom någon typ av kommunikationskanal och har därmed ingen kännedom om objektorienterade databaser. Bengt däremot fick information om objektorienterade databaser när de skulle byta till att programmera objektorienterat. De hade då en kurs i objektorientering där objektorienterade databaser ingick. Calle kommer inte riktigt ihåg var ifrån han har fått information om objektorienterade databaser men tror att det var en artikel han läst på Internet. David har fått information genom personliga kontakter inom företaget och samarbetspartners.

### 4.2.7 Avsikter

Bengt skulle inte hellre se att de hade en objektorienterad databas i deras CAD-system eftersom de just nu inte anser att det är något som de har i fokus. Bcad är för tillfället nöjda med det de har. Calle skulle hellre se att de hade en objektorienterad databas om hastigheten inte skulle försämrats och filtillgången skulle vara tillräckliga för deras kunder. Han tror dock att det skulle kunna bli problem med installationen eftersom det då skulle krävas någon som driver databasen. David tycker det skulle vara bra med både en objektorienterad databas och det sättet de lagrar på nu. Han menar att objektorienterade databaser börjar bli ett krav från marknaden men lägger till att det nog ligger några år fram i tiden.

## 5 Analys

## 5 ANALYS

I det här kapitlet försöker vi med hjälp av vår undersökningsmodell se mönster i våra data.

## 5.1 Analys utifrån Agarwal-Prasad-modellen

För att se om modellens olika delar påverkar varandra i de olika fallen har vi analyserat intervjuerna var för sig genom att lägga in resultatet från intervjuerna i modellens olika delar.

## 5.1.1 Intervju med Anders på Acad

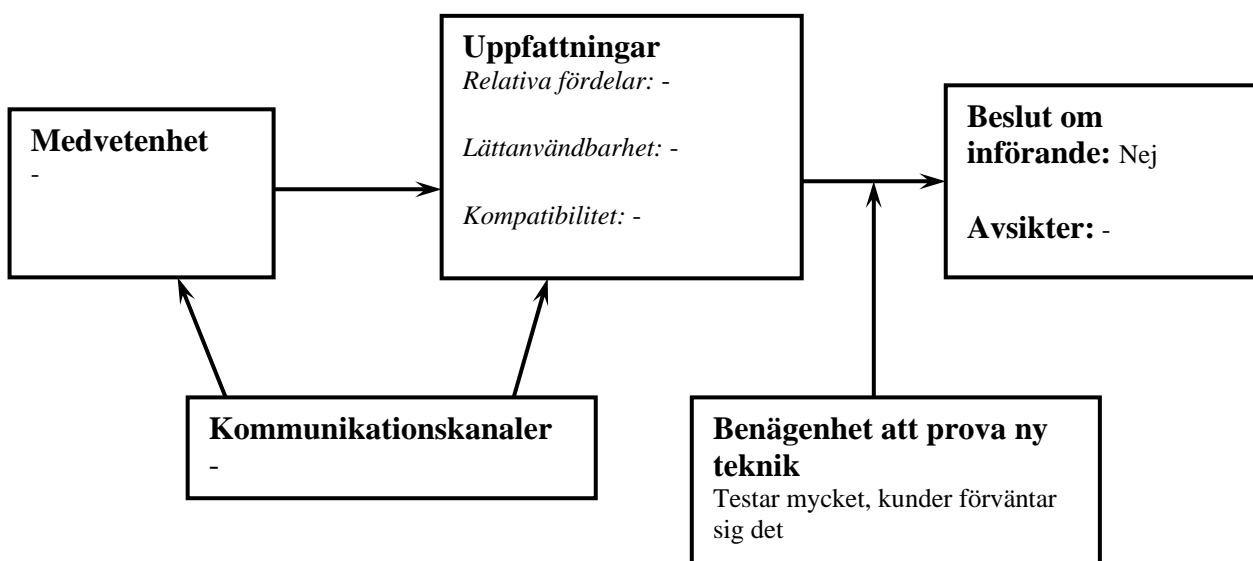


Fig 5.1 Undersökningsmodell för intervju med Acad

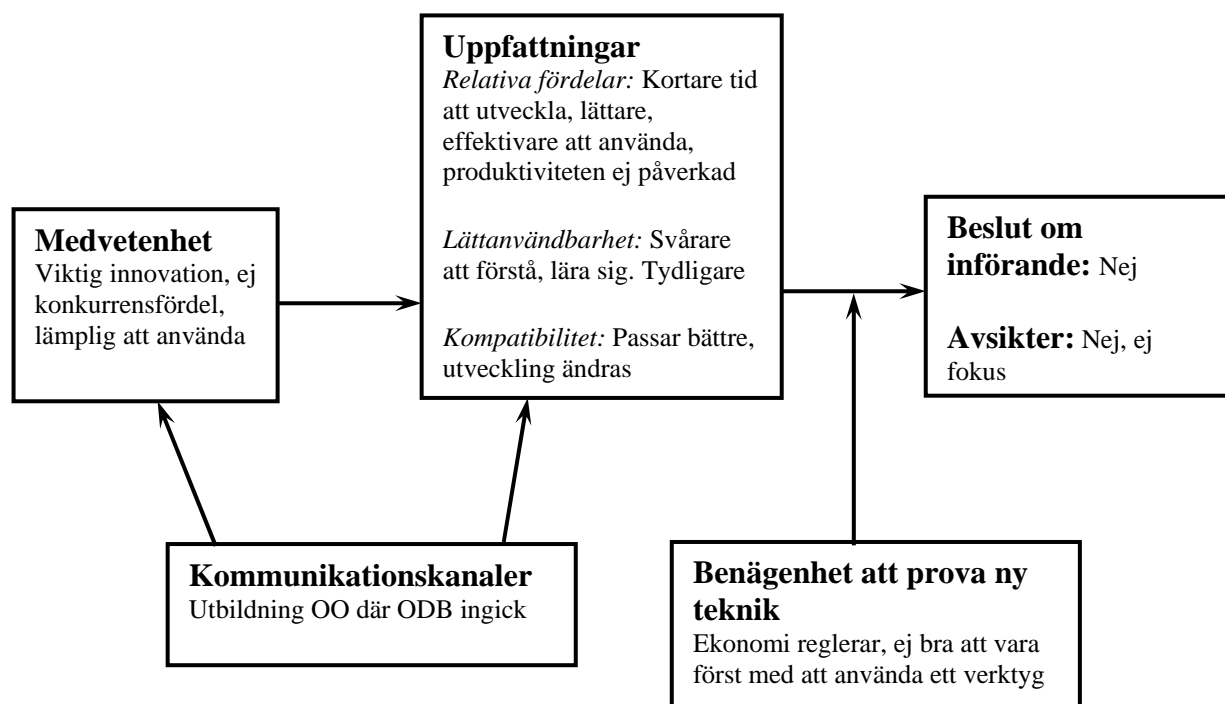
Eftersom det i Anders fall inte har existerat några kommunikationskanaler har han inte någon kännedom om objektorienterade databaser. Det kan vara på grund av detta som Anders inte har utvecklat någon medvetenhet angående objektorienterade databaser. Att Anders inte har någon uppfattning om objektorienterade databaser kan alltså ha påverkats både av bristande kommunikationskanaler och av en brist på medvetenhet, som även det beror på att kommunikationskanalerna har varit bristfälliga. Att Anders och Acad tycker om att experimentera med nya verktyg kan i detta fall alltså inte ha påverkat förhållandet mellan uppfattningar och beslut samt avsikter eftersom det brustit redan vid kommunikationskanalerna och Anders därmed



## 5 Analys

inte utvecklad några uppfattningar. Här stöds alltså det samband som Agarwal och Prasads (1998) undersökning fick fram mellan kommunikationskanaler och medvetenhet samt uppfattningar. Anledningen till att Acad inte använder objekt-orienterade databaser förklaras alltså med att kommunikationskanalerna har brutit.

### 5.1.2 Intervju med Bengt på Bcad



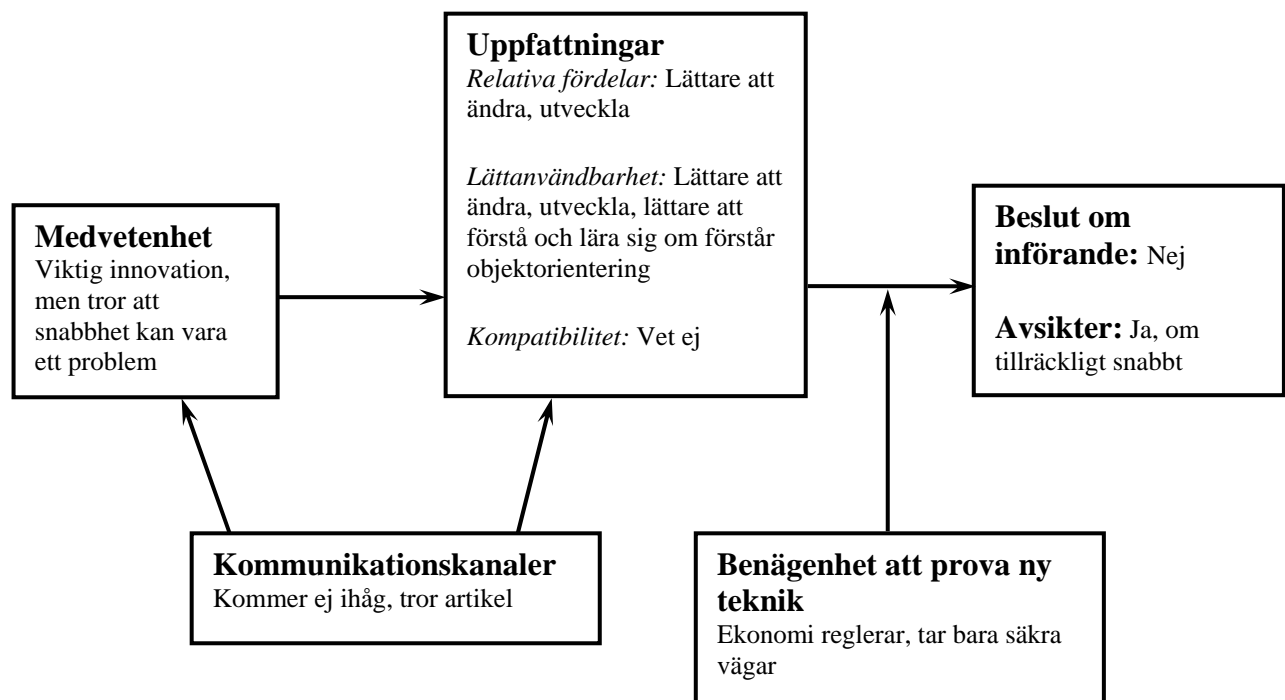
**Figur 5.2** Undersökningsmodell för intervju med Bcad

Bengt har fått information om objektorienterade databaser genom en utbildning vilket kan ha bidragit till hans medvetenhet angående objektorienterade databaser. Det kan även ha påverkat hans uppfattningar positivt, ett förhållande som i så fall stöds av modellen. Han anser att det finns relativa fördelar med objektorienterade databaser och att det skulle passa bättre ihop med Bcads arbetssätt. Lättanvändbarheten ser vi i det här fallet som en ganska neutral uppfattning eftersom han både tycker att det är svårare att förstå och lära sig objektorienterade databaser och att objektorienterade databaser är tydligare. Hans positiva uppfattning har dock inte påverkat beslut eller avsikter positivt. Detta skulle kunna bero på att Bcads ekonomi reglerar hur mycket de kan testa nya verktyg, alltså kan det vara benägenhet som går in och bryter det positiva förhållandet mellan uppfattningar och beslut samt avsikter. Denna påverkan tas upp i Agarwal och Prasads (1998) undersökning då de menar att benägenhet påverkar förhållandet mellan uppfattningar och användning positivt. Deras undersökning visade dock att det bara är kompatibilitet som påverkas. I detta fall vet vi inte om benägenhet hade kunnat påverka förhållandet positivt eftersom det här

## 5 Analys

snarare har påverkat förhållandet negativt. Att Bcad inte använder en objektorienterad databas kan alltså förklaras med att ekonomin reglerar benägenheten att prova på ny teknik.

### 5.1.3 Intervju med Calle på Ccad

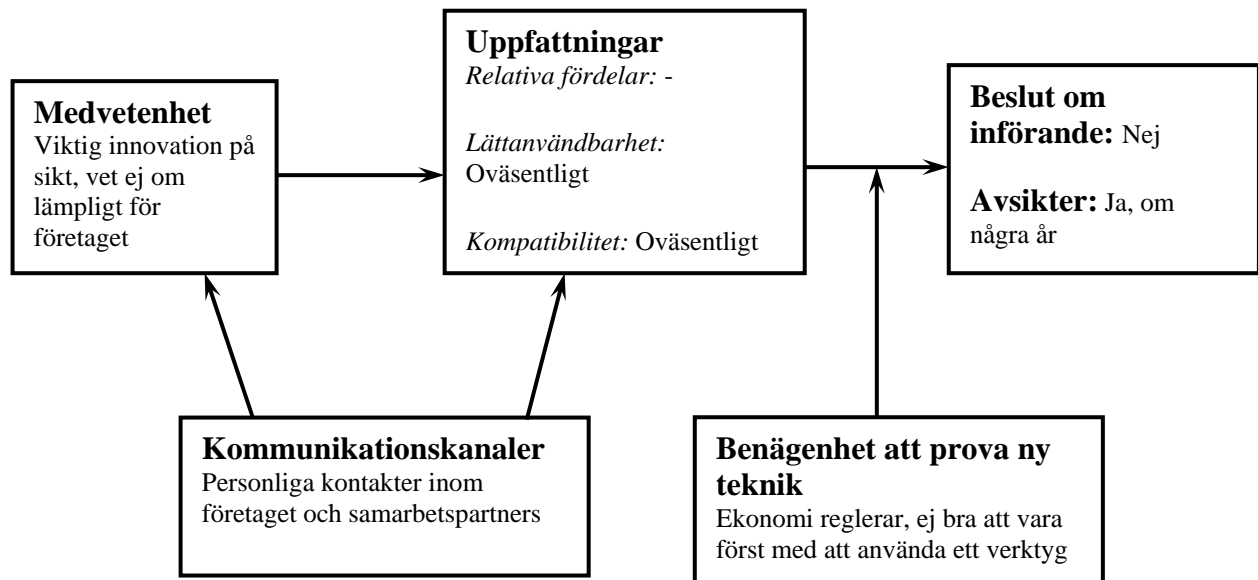


**Figur 5.3** Undersökningsmodell för intervju med Ccad

Calle tror att hans kännedom om objektorienterade databaser är grundad i en artikel som han har läst på Internet. Även i detta fall kan kommunikationskanalen därmed ha påverkat hans medvetenhet och uppfattning om ämnet positivt. I Calles fall stöds troligen det positiva förhållandet mellan medvetenhet angående objektorienterade databaser och hans uppfattningar. Han anser både att det finns relativa fördelar och att objektorienterade databaser är mer lättanvändligt. Hans positiva uppfattning om objektorienterade databaser har inte påverkat beslutet om införande positivt vilket skulle kunna bero på att benägenhet även i detta fall går in och bryter. Ekonomin påverkar benägenheten hos Ccad och Calle menar på att de endast tar säkra vägar. Benägenhet har dock inte gått in och brutit det positiva förhållandet mellan uppfattningar och avsikter. Anledningen till att Ccad inte använder objektorienterade databaser kan förklaras med att deras ekonomi reglerar benägenheten att prova på ny teknik.

## 5 Analys

## 5.1.4 Intervju med David på Dcad



Figur 5.4 Undersökningsmodell för intervju med Dcad

David har fått kännedom om objektorienterade databaser genom personliga kontakter inom Dcad och genom samarbetspartners. Kommunikationskanalen skulle kunna ha påverkat hans medvetenhet angående objektorienterade databaser positivt men det har dock inte, som påvisas i Agarwal-Prasad-modellen, påverkat hans uppfattningar positivt. Trots sin medvetenhet angående objektorienterade databaser vet David inte om det är något som är lämpligt för Dcad att börja använda. Hans uppfattning om objektorienterade databaser är inte så positiv då han inte kan se några direkt fördelar med objektorienterade databaser eller anser att lättanvändbarhet eller kompatibilitet är något som spelar någon roll. Därmed har hans medvetenhet inte påverkat hans uppfattning om objektorienterade databaser positivt. Detta är det enda fall som inte i enlighet med modellen stödjer att medvetenhet påverkar uppfattningar positivt. Att Dcad inte har infört objektorienterade databaser kan förklaras med att uppfattningen inte är positiv samt att ekonomin reglerar benägenheten att prova på ny teknik. Avsikterna har dock inte påverkats då David anser att det skulle vara bra med både en objektorienterad databas och en filstruktur några år fram i tiden.

**6 Diskussion**

---

**6 DISKUSSION**

---

*I diskussionsnittet använder vi teorin för att analysera våra data. Vi diskuterar både de resultat som vi kan förklara med hjälp av teori och de resultat som inte kan förklaras från teori. Upplägget utgår från Agarwal-Prasad-modellen, där vi avhandlar delarna i modellen en efter en varefter vi belyser faktorer som inte passar in i den.*

---

**6.1 Inledande diskussion**

Respondenterna använder objektorienterad programmering och deras CAD-system har flera simultana användare. Cads system är dock inte uppbyggt för simultana användare och vi vet inte om de är i behov av det överhuvudtaget. Två av respondenterna har distribuerade CAD-system och alla respondenter lagrar komplexa objekt i form av ritningar eller bilder i systemen. Alla dessa faktorer är något som exempelvis Loomis (1995), Beck (2001) och Elmasri och Navathe (2004) tar upp som orsaker att använda objektorienterade databaser.

**6.2 Diskussion utifrån Agarwal-Prasad-modellen****6.2.1 Relativa fördelar**

Det som har framkommit i undersökningen är att några av respondenterna har en uppfattning om att det hade varit lättare och effektivare att utveckla och ändra i CAD-systemet om de hade haft en objektorienterad databas istället för det lagringssätt som används. Vi tror att en förklaring till det är att objektorienterade databaser har samma funktionalitet som objektorienterad programmering (Elmasri & Navathe, 2004). Dessutom sker utvecklingen av de CAD-system respondenterna har i objektorienterade programmeringsspråk. Det sammantaget gör att vi kan se flera möjliga förklaringar utifrån teorin till dessa relativa fördelar:

- Det är modeller med starkt inbördes förhållande som används till både programmeringen och databasen (Devarakonda, 2001).
- I en objektorienterad databas separeras inte databasen från programmeringen, vilket innebär att ingen separat hantering av databasen behövs (Bancilhon, 1996).
- Olika hierarkier, som arv, kan lagras direkt i databasen (Elmasri & Navathe, 2004).
- Operatoröverlagring stöds (ibid.).

## 6 Diskussion

---

Calle tar även upp att han tror att det är lättare att utveckla och ändra i ett system som använder en objektorienterad databas eftersom ändringen blir lokal och inte påverkar andra objekt. Med andra ord är det lättare att ändra på grund av att objektorienterade databaser har inkapsling (Elmasri & Navathe, 2004).

Bengt tror att det hade tagit mindre tid att utveckla deras CAD-system om de hade haft en objektorienterad databas istället för en relationsdatabas eftersom de då sluppit mellanskiktet där konvertering mellan objekt och tabeller sker. Detta påstående stämmer överens med det Devarakonda (2001) skriver om att applikationsmodellen och databasmodellen har starka band vid användandet av en objektorienterad databas. Författaren menar att detta resulterar i mindre kod, mer naturliga datastrukturer samt bättre underhållbarhet och kodåtervinning.

En annan relativ fördel som Bengt uppgett är att systemet blir snabbare med en objektorienterad databas eftersom den kod som konverterar mellan objekt och tabeller rationaliseras bort. Prestandan vid arbete med komplexa objekt är bättre hos objektorienterade databaser än hos relationsdatabaser (Bancilhon, 1996; Beck, 2001; Cattell & Skeen, 1992; Zhu et al., 2005). Bengts uppfattning överensstämmer med teorier. Bengt menar dock att prestandafrågor har minskat i betydelse och att datalagring bara är en liten del av systemet vilket vi tror leder till att Bcad inte har fokus på datalagring för tillfället

### 6.2.2 Lättanvändbarhet

Zhu et al. (2005) uppger att inlärningskurvan i deras undersökning var lägre för objektorienterade databaser än för relationsdatabaser. Beck (2001) har motsatt uppfattning, objektorienterade databaser är mer komplexa än relationsdatabaser och därmed svårare att lära sig. Bengt har samma uppfattning som Beck (2001), det vill säga att relationsdatabaser är lättast att lära sig. Calle har kommit med en möjlig förklaring till att det finns skillnad i vad olika författare och respondenter anser, han uppger att det är lättare att lära sig objektorienterade databaser för dem som redan förstår objektorientering. Vår tolkning blir då att det blir lättare att lära sig relationsdatabaser än objektorienterade databaser för de personer som inte har någon förkunskap inom objektorienterad programmering. De personer som har kunskap om objektorienterad programmering har däremot lättare att lära sig objektorienterade databaser än relationsdatabaser.

Chung et al. (1995) anser att relationsmodellen är lätt att förstå och att det kan vara en anledning till relationsdatabasens framgång. På motsvarande sätt anser Bengt att en relationsdatabas är lättare att förstå än en objektorienterad databas. Zhu et al. (2005) menar dock att objektorienterade databaser är lätta att förstå då objekt i databasen är en direkt avbild av objekt i verkligheten. Dessa författare har dock inte jämfört objektorienterade databaser med relationsdatabaser i sina uttalanden.

Trots att Bengt tror att en relationsdatabas är lättare att förstå tycker han att objektorienterade databaser verkar tydligare. Detta antar vi beror på att han tycker det

## 6 Diskussion

---

är mer logiskt att tänka sig ritningarna som objekt, vilket även Zhu et al. (2005) framhåller då de menar att objekten är en direkt avbild.

### 6.2.3 Kompatibilitet

Bengt anser att kompatibiliteten mellan arbetssättet och objektorienterade databaser är hög. Detta på grund av att det är mer logiskt att tänka sig ritningen och dess delar som objekt. En förklaring till detta är, som nämns ovan, att objekten i en objektorienterad databas är en direkt avbild av de verkliga objekten (Zhu et al., 2001). Indirekt behandlas detta också i avsnittet om modeller där författarna anser att det är en fördel att applikationens datamodell och databasens datamodell har ett starkt inbördes förhållande. Att Bengt tycker det är mer logiskt att tänka sig ritningens delar som objekt skulle därmed också kunna bero på att de använder ett objektorienterat programmeringsspråk vilket gör det lättare att tänka objektorienterat även i databasen.

Bengts åsikt att det är mer kompatibelt skulle kunna bero på att databasens funktionalitet är integrerad i det objektorienterade programmeringsspråket (Bancilhon, 1996). Programmerarna behöver alltså inte använda ett separat språk för databasåtkomst.

### 6.2.4 Benägenhet att prova på ny teknik

Det har ur intervjuerna framkommit att ekonomin reglerar hur mycket företagen kan experimentera och prova på nya verktyg. Detta leder antagligen till att dessa begränsade pengar inte läggs på en databashanterare som enligt Beck (2001) saknar en officiell standard, mognad och verktyg. Detta ser vi alltså som en möjlig förklaring till att objektorienterade databaser inte används.

### 6.2.5 Medvetenhet

Objektorienterade databaser är enligt respondenterna en viktig innovation av två olika anledningar, eftersom det är en naturlig fortsättning på objektorienterad programmering och eftersom utvecklingen av applikationen sker objektorienterat. Elmasri och Navathe (2004) menar att det går att spara persistenta objekt direkt från det objektorienterade utvecklingspråket. Det innebär att författarna ser objektorienterade databaser som en integrerad del av objektorienterad programmering och inte som ett separat sätt att lagra objekten, detta är överensstämmande med respondenternas svar.

Prestandan inom CAD-området är bättre med objektorienterade databaser än med relationsdatabaser (Bancilhon, 1996; Beck, 2001; Cattell & Skeen, 1992; Zhu et al., 2005). Bengt tar i likhet med detta upp att objektorienterade databaser är väldigt snabba. Trots det anser Bengt att det inte är en konkurrensfördel att använda

## 6 Diskussion

---

objektorienterade databaser eftersom datorer idag är så snabba att prestandafrågor har minskat i betydelse. Anders har uppfattningen att tillämpningen är den viktigaste konkurrens fördelen och inte hur data lagras i systemet. Den enda anledningen han kan se att objektorienterade databaser kan vara en konkurrens fördel är om det går att differentiera sig med hjälp av dem. Vidare anser Anders att något som kan användas för att differentiera sig är hur robust och felsäkert systemet är. Cattell (1994) menar att säkerhet går att hantera i objektorienterade databaser på ett sätt som passar de flesta applikationer medan Connolly och Begg (2005) uppger att det inte finns tillräckliga säkerhetsmekanismer. Att det inte finns en enighet som uppger att säkerhet hanteras på ett bra sätt i objektorienterade databaser skulle kunna påverka Acads beslut om införande negativt även om de utvecklar en medvetenhet och en positiv uppfattning. Detta på grund av att deras kunder kräver stor säkerhet i systemet.

Bengt tar även upp att det hade varit lämpligt för Bcad att använda objektorienterade databaser eftersom utvecklingen av applikationens kod sker i C++, det vill säga ett objektorienterat programmeringsspråk. En anledning att använda objektorienterade databaser är att utvecklingen sker objektorienterat och att objektorienterade databaser är en del av det beslutet (Loomis, 1995). Bcads CAD-system är dock inte utvecklat enbart i C++ utan det ligger fortfarande kvar gammal C-kod. Bengt anser därför att det inte är lämpligt att använda objektorienterade databaser förrän de kan spara objekten rakt ner, det vill säga när de bytt ut den gamla koden.

### 6.2.6 Avsikter

Calle skulle hellre se en objektorienterad databas i Ccads system än filsystemet som de använder nu. Men han ser ett problem, han tror inte att snabbheten är tillräcklig. Bcad har undersökt möjligheten att använda objektorienterade databaser, och de kom fram till att objektorienterade databaser var ”grymt snabba”. Många författare tar upp att objektorienterade databaser, vid hantering av komplexa objekt, har bättre prestanda än relationsdatabaser (Bancilhon, 1996; Beck, 2001; Cattell & Skeen, 1992; Zhu et al., 2005). Sammantaget kan vi då konstatera att Calle har en uppfattning som inte stämmer med övriga källor. Anledningen till denna skillnad kan vara att Calle troligtvis inte bara tar hänsyn till databasens snabbhet utan även den tid det tar från det att användaren har givit ett kommando tills han ser resultatet på bildskärmen. Eftersom Bancilhon (1996) tar upp att hela objektet kan överföras till klienten vid bearbetning av data så att klienten arbetar internt med objektet efter det anser vi inte att Calles argument håller. En anledning till att Calle har avvikande uppfattning angående snabbheten kan vara att hans kommunikationskanaler inte har tagit upp hanteringen av data tillräckligt detaljerat för att han ska ha information om denna aspekt. Enligt denna diskussion hade med andra ord en ökad kommunikation kunnat innebära att avsikterna blivit positiva utan reservationer. En lämplig kommunikationskanal för detta är personliga kontakter, vilka förmedlar teknikens värde anpassat för individen.

## 6 Diskussion

---

### 6.2.7 Förhållanden som inte förklaras av Agarwal-Prasad-modellen

David anser att det är oväsentligt vilket lagringssätt som är lättast att få att göra som man vill och vad som passar arbetssättet bäst. Denna inställning har han trots att han anser att objektorienterade databaser är en viktig innovation på sikt och trots att han fått information om objektorienterade databaser från flera olika håll. Dessutom är avsikterna att införa objektorienterade databaser om några år. Agarwal-Prasad-modellen stödjer inte denna uppfattning och vi har inte hittat någon övrig litteratur som behandlar denna möjlighet. Vi kan därför inte dra någon slutsats från detta baserat på teori utan bara belysa en oväntad faktor, nämligen att medvetenhet och kommunikationskanaler påverkar avsikter. En tanke som uppkommer då är ifall medvetenhet och kommunikationskanaler påverkar avsikter direkt, med andra ord om uppfattningar inte nödvändigtvis är ett mellanled.

För att försöka reda ut den tanken har vi valt att titta på de tre andra intervjuerna utan att ta någon hänsyn till uppfattningar. Acad faller bort eftersom Anders inte har någon kännedom om objektorienterade databaser och därmed inte heller någon avsikt. Bcad har en medvetenhet och kommunikationskanalerna har gett Bengt en bra översikt över vad objektorienterade databaser är. Däremot finns ingen avsikt att införa objektorienterade databaser, vilket inte stödjer vår tanke. Även i Calles fall har kommunikationskanalerna fungerat och han har en medvetenhet om objektorienterade databaser. Avsikten är att införa objektorienterade databaser, under förutsättning att det är tillräckligt snabbt. Problem med snabbheten har han även nämnt när han svarat på frågorna om medvetenhet, vilket innebär att vi ser en koppling mellan medvetenhet och avsikter. Däremot har snabbheten inte nämnts i uppfattningar och vi kan alltså inte använda snabbhet som en variabel när vi bedömer om uppfattningar är överflödigt. Vi kan däremot se att medvetenhet och kommunikationskanaler leder till avsikter.

Om vi nu tar in även uppfattningar som en variabel ser vi att Bengts uppfattningar speglar medvetenheten, på samma sätt speglar Calles uppfattningar hans medvetenhet. Davids uppfattningar speglar dock inte medvetenheten. Det leder till att vi kan anta att uppfattningar går att bortse från vid analys av avsikter och att medvetenhet och kommunikationskanaler leder direkt till avsikter.

### 6.2.8 Modellen som helhet

De flesta av sambanden i modellen stöds av vår undersökning. Det har påvisats att kommunikationskanaler påverkar både medvetenhet och uppfattningar. Vi har dock enbart tittat på kommunikationskanaler i stort och inte de enskilda kanalerna. Medvetenhet påverkar i sin tur uppfattningar. En negativ uppfattning om objektorienterade databaser leder till att det inte tas ett beslut om att införa. Om en positiv uppfattning påverkar beslut om införande positivt vet vi däremot inte eftersom benägenhet att införa en ny teknik antagligen går in och bryter detta förhållande. Uppfattningar påverkar dessutom avsikten att införa objektorienterade databaser, detta förhållande går alltså benägenhet inte in och bryter. Att benägenhet inte bryter



## **6 Diskussion**

---

förhållandet mellan uppfattningar och avsikter tror vi skulle kunna bero på att ett beslut om införande i längden kostar mer än en avsikt.

## 7 Slutsatser

---

# 7 SLUTSATSER

---

*I detta avslutande kapitel presenteras uppsatsens slutsatser. Kapitlet knyter an till uppsatsens frågeställning och syfte. Avslutningsvis ger vi även förslag till framtida forskning.*

---

Syftet med denna uppsats är att identifiera de faktorer som gör att objektorienterade databaser inte används inom vissa applikationsområden trots att litteraturen anser att objektorienterade databaser är att föredra inom dessa applikationsområden. Vi valde applikationsområdet CAD för att undersöka detta.

De företag vi undersökte har alla visat sig lagra komplexa data i sina CAD-system. Flertalet hade dessutom en distribuerad lösning med många simultana användare och använder sig av objektorienterad programmering. I och med att ingen av dessa företag använder sig av en objektorienterad databas har vi konstaterat att dessa kriterier inte är tillräckliga för att välja en objektorienterad databas.

Våra slutsatser angående vad som påverkar beslut om införande av objektorienterade databaser negativt är att:

- Det inte finns någon benägenhet att prova på nya verktyg
- Koden inte är helt objektorienterad
- Kommunikationskanalerna är bristfälliga
- Prestandafrågor har minskat i betydelse

Att tre av företagen inte har någon benägenhet att prova på nya verktyg och metoder har troligen påverkat beslutet om införande negativt. Det har visat sig att benägenheten påverkas av hur mycket ekonomiska resurser företaget använder för att experimentera. Ekonomiska resurser påverkar alltså indirekt ett beslut om införande.

Ett av företagens CAD-system är inte helt utvecklat i ett objektorienterat programmeringsspråk. Det ligger fortfarande kvar gammal C-kod och i och med detta kan objekten inte sparas rakt ner i en objektorienterad databas. Företaget skulle alltså inte dra nytta av en sådan.

I ett av fallen har kommunikationskanalerna varit bristfälliga. Tillräcklig information om objektorienterade databaser har alltså inte nått ut till företaget, något som krävs för att det ska kunna skapas en inställning.

Prestandan är en av de fördelar med objektorienterade databaser som tas upp i teorin. En av respondenterna nämnde dock att eftersom datorer idag är så snabba så har prestandafrågor minskat i betydelse. Datalagringen är bara en liten del och de koncentrerar sig hellre på att utveckla ett effektivt gränssnitt.

## 7 Slutsatser

---

Utifrån vår undersökningsmodell har vi kommit fram till att det som har påverkat avsikten att välja lagringssätt är att i det fall där de inte fått tillräcklig information om objektorienterade databaser utvecklade de inte heller en positiv attityd och generell uppfattning och därmed inte en positiv avsikt. Det var även ett fall där de fått information, utvecklat en positiv attityd samt en generell uppfattning och därmed en positiv avsikt. Det fall där det fanns en positiv attityd och en generell uppfattning utan en positiv avsikt kan förklaras med att deras system inte var helt objektorienterat. Fallet där de inte hade någon uppfattning om objektorienterade databaser men ändå en positiv avsikt beror på att kommunikationskanalerna och den positiva attityden i detta fall direkt har speglat avsikterna.

### 7.1 Framtida forskning

Eftersom vi bara undersökt de företag som inte använder objektorienterade databaser ser vi det som intressant att även undersöka företag som valt att använda objektorienterade databaser. Här finns både möjlighet att undersöka vad som påverkat detta val och hur användningen i praktiken fungerar.

# BILAGA 1 – INTERVJUGUIDE

## Verksamhet & individ

1. Gör en kort beskrivning av företaget.
2. Vilken bakgrund har du (t ex utbildning, erfarenhet)?
3. Vad har du för befattning?
4. Vilka arbetsuppgifter har du?
5. Vad har du haft för roll vid utvecklande av företagets CAD-system?
6. Vad har du för erfarenhet av objektorienterad
  - a. Systemutveckling?
  - b. Databaser?
7. Hur går systemutveckling till inom företaget traditionellt?

## CAD-system & lagring

8. Beskriv ert CAD-system.
9. Hur lagrar ni data från CAD-systemet?
  - a. Leverantör?
  - b. Används objektorienterade delar? Vilka? Varför? Upplever du att det hade varit en fördel att lagra all data på samma ställe? (Om lagring på flera olika sätt)
10. Vilka olika lagringstyper valde ni mellan?

## Relativa fördelar

Om ni valt en objektorienterad databas istället för ... tror du då att...

11. utvecklingen av CAD-systemet hade kunnat utföras på kortare tid?
12. det hade varit lättare att utveckla CAD-systemet?
13. utvecklingen av CAD-systemet hade kunnat utföras på ett mer effektivt sätt?
14. det skulle gett dig större kontroll över ditt arbete med utvecklingen av CAD-systemet?
15. det skulle ha förbättrat produktiviteten i utvecklingsprojektet?
16. det skulle ha förbättrat kvaliteten på systemet?
17. CAD-systemet hade varit snabbare?

### **Lättanvändbarhet**

Om du jämför objektorienterad databaser med ..., vilken av dem tror/anser du är..

18. lättast att få att göra som du vill?
19. lättast att lära sig?
20. kräver mest ansträngning i form av tänkande?
21. mest tydlig och förståelig?

### **Kompatibilitet**

22. Passar objektorienterad databaser eller ... ditt arbetssätt med utveckling av system bäst?
23. Om ni hade valt en objektorienterad databas, hade då sättet CAD-systemet utvecklades på ändrats?

### **Avsikter**

24. Skulle du hellre se att ni hade en objektorienterad databas i CAD-systemet än en ...?

### **Benägenhet att prova på ny teknik**

25. Tycker du om att experimentera med nya verktyg och utvecklingsmetoder?
26. Brukar du tveka inför att prova nya och oprövade utvecklingsmetoder eller verktyg?
27. Vill du helst att andra personer ska reda ut buggar och problem med ett verktyg innan du använder det?

### **Medvetenhet**

28. Anser du att objektorienterade databaser är en viktig innovation?
29. Tror du att objektorienterade databaser är kritiskt för att få en konkurrensfördel i er bransch?
30. Tror du att det är lämpligt för ert företag att börja använda objektorienterade databaser?

### **Kommunikationskanaler**

31. Var ifrån har du fått uppgifter/information om objektorienterade databaser?

## BILAGA 2 - BESKRIVNING AV KODER

<b>Kod</b>	<b>Förklaring</b>
UtvGen	Utveckling Generellt
Distr	Distribuerat
Föret	Företagsbeskrivning
Funk	Funktioner i systemet
Bakg	Bakgrund (respondenten) inkl arbetsuppgifter osv
Lagr	Lagring
Ek	Ekonomi
Rel	Relativa fördelar
Lättanv	Lättanvändbarhet
Komp	Kompatibilitet
Avs	Avsikter
Öpp	Öppenhet för nya ideer
Med	Medvetenhet
Komm	Kommunikationskanaler

# REFERENSER

- Agarwal, R., Prasad, J. (1998). The antecedents and consequents of user perceptions in information technology adoption. *Decision Support Systems*, Vol 22, s. 15-29.
- Bancilhon, F. (1996). Object Databases. *ACM Computing Surveys*, Vol. 28, No. 1.
- Beck, H. (2001). Agricultural enterprise information management using object databases, Java and CORBA. *Computers and Electronics in Agriculture*, Vol. 32, No. 2, s. 119-147.
- Bryman, A. (2001). *Samhällsvetenskapliga metoder*. (Björn Nilsson övers.) Malmö: Liber.
- Cadtorget (2006). [www.cadtorget.se](http://www.cadtorget.se) [Besökt: 2006-04-19].
- Cattell, R. (1994). *Object data management – Object-oriented and extended relational database systems* (Rev. uppl.). Reading, Massachusetts: Addison-Wesley.
- Cattell, R., & Skeen, J. (1992). Object operation benchmark. *ACM Trans. Database Syst* Vol. 17, No. 1.
- Chung, Y. Lin, Y. och Chang, D. (1995) *Objects and Relational Databases*. Tillgänglig: <http://delivery.acm.org/10.1145/270000/260273/p164-chung.pdf?key1=260273&key2=6020777411&coll=Portal&dl=ACM&CFID=71452663&CFTOKEN=37729726> [Hämtad: 2006-05-01].
- Connolly, T. & Begg, C. (2005). *Database Systems – A Practical Approach to Design, Implementation, and Management* (4:e rev. uppl.). Harlow, England: Addison-Wesley.
- Date, C. J. (2004). *An introduction to database systems*. Boston: Pearson Addison-Wesley.
- Deitel, H. M. & Deitel, P. J. (2005). *Java – How to program* (6:e rev. uppl.). Upper Saddle River, New Jersey: Pearson Education.
- Delobel, C., Lécluse, C. & Richard, P. (1995). *Databases: From Relational to Object-Oriented Systems*. London: International Thompson.
- Devarakonda, S. (2001). *Object-Relational Database Systems – The Road Ahead*. Tillgänglig: <http://www.acm.org/crossroads/xrds7-3/ordbms.html> [Hämtad: 2006-05-01].
- Elmasri, R. & Navathe, S. B. (2004). *Fundamentals of database systems*. (4:e rev. uppl.). Boston: Pearson Education.
- Jacobsen, D. I. (2002). *Vad, hur och varför? Om metodval i företagsekonomi och andra samhällsvetenskapliga ämnen* (Gunnar Sandin övers.). Lund: Studentlitteratur (Originalarbete publicerat 2000).

- Kvale, S. (1996). *Interviews: An introduction to Qualitative Research Interviewing*. London: SAGE.
- Loomis, M. E. S. (1995). *Object Databases, The Essentials*. Reading, Massachusetts: Addison-Wesley.
- Miles, M. B. & Huberman, A. M. (1994). *Qualitative data analysis: an expanded sourcebook* (2:a rev. uppl.). London: SAGE.
- Niyomthum, K. & Chittayasothorn, S. (2003). *A transformation from an Object Database to an Object Relational Database*. Tillgänglig: <http://ieeexplore.ieee.org/iel5/8956/28384/01268424.pdf?tp=&arnumber=1268424&isnumber=28384> [Hämtad: 2006-05-01]
- Pärletun, L. G. (2006). *Datorstödd konstruktion*. Nationalencyklopedin. Tillgänglig: [http://www.ne.se/jsp/search/article.jsp?i\\_sect\\_id=150967&i\\_history=1](http://www.ne.se/jsp/search/article.jsp?i_sect_id=150967&i_history=1) [Besökt: 2006-05-09].
- Silberschatz, A., Korth, H. F. & Sudarshan, S. (2002). *Database System Concepts* (4:e rev. uppl.). New York: McGraw Hill.
- Svenning, C. (2000). *Metodboken*. (4:e uppl.) Eslöv: Lorentz.
- Trost, J. (1997). *Kvalitativa intervjuer*. (2:a rev uppl.) Lund: Studentlitteratur.
- Wretling, U. & Karlquist, S. (1990). *ODBMS: objektorienterade databaser: utvecklingen i USA*. Stockholm: Sveriges tekniska attachéer.
- Zhu, Y, Crouch, J och Tabrizi, M. H. N. (2005) *In-process object-oriented database design for .NET*. Tillgänglig: <http://delivery.acm.org/10.1145/1100000/1095747/p139-zhu.pdf?key1=1095747&key2=7839677411&coll=Portal&dl=ACM&CFID=71451771&CFTOKEN=63417464> [Hämtad: 2006-05-01].