

**Är objektorienterad modellering ett måste?**

**(HS-IDA-EA-00-409)**

**Anders Johansson (a97andjo@student.his.se)**

*Institutionen för datavetenskap  
Högskolan i Skövde, Box 408  
S-54128 Skövde, SWEDEN*

Examensarbete på det dataekonomiska programmet under  
vårterminen 2000.

Handledare: Kim Laurio

## **Är objektorienterad modellering ett måste?**

Examensrapport inlämnad av Anders Johansson till Högskolan i Skövde, för Kandidatexamen (B.Sc.) vid Institutionen för Datavetenskap.

**2000-06-09**

Härmed intygas att allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och att inget material är inkluderat som tidigare använts för erhållande av annan examen.

Signerat: \_\_\_\_\_

## Är objektorienterad modellering ett måste?

Anders Johansson (a97andjo@student.his.se)

**Nyckelord:** Systemutveckling, UML, datamodellering, processororienterade tekniker.

### Sammanfattning

En central del i ett systemutvecklingsprojekt är de tekniker som används för att strukturera organisationer och informationen i dessa. I flera årtionden har traditionella tekniker utvecklats. Exempel på tekniker som anses traditionella är ER-modellering och dataflödesdiagram. Idag används objektorienterade programspråk mer och mer för att implementera databaser. Det har dock inte funnits stöd för att analysera ett system objektorienterat förrän i början på 1990-talet. De tekniker och metoder som utvecklades då hade flera olika inriktningar och begreppsförvirringen var stor. 1997 kom en standard för objektorienterad modellering i form av Unified Modeling Language (UML). I och med denna standard försökte man klargöra vilka begrepp som skulle gälla och vilken typ av notation som kunde användas. Arbetet försöker att visa vad som kan ersätta de traditionella teknikerna med objektorienterade metoder och tekniker. Koncentrationen har lagts på att analysera vilka tekniker som används i de två första faserna av livscykelmodellen, förändringsanalys och systemering. Resultatet blev att en fullständig ersättning för de traditionella teknikerna inte finns i UML i livscykelmodellens två första faser.

# Innehållsförteckning

<b>1</b>	<b>Bakgrund</b> .....	<b>1</b>
<b>2</b>	<b>Introduktion</b> .....	<b>3</b>
2.1	Systemutveckling .....	3
2.2	Funktionsorienterat synsätt.....	4
2.2.1	Dataflödesdiagram .....	4
2.3	Dataorienterat synsätt .....	6
2.3.1	Datamodellering.....	6
2.4	Objektorienterat synsätt.....	7
2.4.1	Objekt och klass .....	8
2.4.2	Objektorienterade relationer .....	9
2.5	Unified Modeling Language (UML).....	10
2.6	Generella arbetssätt för systemutveckling.....	11
2.7	Systemutvecklingens hjälpmedel .....	12
2.8	Livscykelmodell .....	14
2.8.1	Förändringsanalys .....	15
2.8.2	Systemering.....	16
2.8.3	Realisering .....	17
2.8.4	Implementering .....	17
2.8.5	Förvaltning och drift .....	18
2.8.6	Avveckling.....	18
<b>3</b>	<b>Problembeskrivning</b> .....	<b>19</b>
3.1	Problemställning.....	19
3.2	Avgränsning .....	20
3.3	Förväntat resultat.....	20
<b>4</b>	<b>Metod</b> .....	<b>21</b>
4.1	Litteraturstudie .....	22
4.2	Fallstudie .....	23
4.3	Intervju .....	23
4.4	Val av metod .....	24
<b>5</b>	<b>Genomförande</b> .....	<b>26</b>
5.1	Litteraturstudie .....	26
5.2	Fallstudie .....	27

5.3	Intervjuer .....	27
<b>6</b>	<b>Materialredovisning .....</b>	<b>28</b>
6.1	Förändringsanalys .....	28
6.2	Verksamhetsanalys .....	28
6.2.1	Traditionellt arbetssätt .....	28
6.2.2	Objektorienterat arbetssätt .....	29
6.3	Informationsanalys .....	29
6.3.1	Traditionellt arbetssätt .....	29
6.3.2	Objektorienterat synsätt .....	31
6.4	Fallstudie .....	36
6.4.1	Förstudie .....	37
6.4.2	Konceptuell kartläggning .....	39
6.4.3	Funktionell utformning .....	41
<b>7</b>	<b>Analys .....</b>	<b>43</b>
7.1	Förändringsanalys .....	43
7.2	Verksamhetsanalys .....	43
7.3	Informationsanalys .....	43
<b>8</b>	<b>Resultat och slutsats .....</b>	<b>47</b>
8.1	Traditionell utveckling .....	47
8.1.1	Förändringsanalys .....	47
8.1.2	Analysfas i systemering .....	47
8.2	Objektorienterad utveckling .....	48
8.2.1	Förändringsanalys .....	48
8.2.2	Analysfas av systemering .....	48
8.3	Slutsatser .....	48
8.4	Förslag till framtida arbete .....	49
<b>9</b>	<b>Diskussion.....</b>	<b>50</b>
9.1	Arbetsprocessen.....	50
9.1.1	Val av metod .....	51
9.2	Systemutveckling .....	51
9.2.1	Utvecklingsmodeller .....	51
9.2.2	Traditionella tekniker .....	52
9.2.3	Objektorienterat synsätt .....	52
	<b>Referenser.....</b>	<b>58</b>

## **Bilaga 1**

# 1 Bakgrund

Enligt Mathiassen et al. (1998) är systemutveckling en process som handlar om att utveckla ett datasystem som blir ett effektivt redskap för användarna i deras arbetsuppgifter. Ett lyckat systemutvecklingsprojekt innebär därmed att på bästa möjliga sätt undersöka hur dagens arbetssituation ser ut, och vad det finns för behov hos användarna av det nya datasystemet (Mathiassen et al., 1998). Dagens arbetssituation och önskemålen ligger sedan till grund för ett informationssystem som gör att användarnas arbetssituation i det nya systemet förbättras i jämförelse med den nuvarande arbetssituationen.

Systemutveckling har funnits i flera årtionden och detta har gjort att det har blivit en gradvis utveckling av processen. I början på sextioalet var ett av problemen att användarna och de som skulle utveckla informationssystemen inte kunde kommunicera på ett bra sätt så att de förstod varandra (Mathiassen et al., 1998). Detta kunde medföra att användarna fick ett system som de inte var tillfredsställda med, eller att systemet blev försenat och kostade mer pengar än budgeterat (Mathiassen et al., 1998). I slutet på sextioalet gjordes försök att underlätta systemutvecklingen genom att utveckla olika modeller och metoder. Detta gjordes för att få systemutvecklingen tillförlitlig, så att projekt blev klara i tid och att de utförde vad de var förväntade att göra (Andersen, 1994). Det blev dessutom en utveckling på antalet deltagare som var engagerade i ett projekt ökade, för att ha möjlighet att tillfredsställa både användare och de som utvecklade systemen (Mathiassen et al., 1998).

Utvecklingen har fortsatt och idag har de flesta företag någon form av datasystem och databehandling i sin verksamhet. Det finns olika typer av system att köpa på marknaden som till exempel standardssystem och expertsystem. Ett standardssystem har en mängd fördefinierade funktioner inlagda och de är lika för alla som köper systemet (Andersen, 1994). Denna typ av system har utvecklats för att passa många olika typer av verksamheter, som till exempel ett system för tillverkande industrier (Andersen, 1994). Ett expertsystem är ett informationssystem som är framtaget för att agera som en expert och kunna göra utlåtanden för den specifika domänen som den är anpassad för, som till exempel ett system som läkare kan använda för att ställa diagnos (Avison & Fitzgerald, 1995).

Utvecklingen har också fortsatt i form av att det har utvecklats flera olika typer av synsätt på hur man ska utveckla ett informationssystem (Andersen, 1994). Mathiassen et al. (1998) skriver att ett av de första synsätten var funktionsorienterat. Detta innebar att systemet byggdes upp kring de olika funktioner som ett företag har i sin verksamhet. Ett av de senaste synsätten som har utvecklats är objektorienterat synsätt på systemutvecklingen (Mathiassen et al., 1998). Det innebär att utvecklingen av systemet koncentreras kring de olika objekten i företaget, som till exempel personer, bilar och leksaker. Mer om de olika synsätten tas upp i kapitel 2.

Det förekommer att flera företag idag arbetar med flera olika synsätt som gör att det blir en stor variation på hur ett systemutvecklingsprojekt kan utföras. De vanligaste synsätten som används är de så kallade traditionella angreppssätten, som till exempel

funktionsorienterat synsätt (Mathiassen et al. 1998). Ett av de företag som arbetar med systemutveckling på ett traditionellt vis är IBS Konsult AB. Objektorienterad systemutveckling är ett av de senaste synsätten att arbeta med systemutveckling. Detta synsätt är intressant för företag, men de arbetar i de allra flesta fall fortfarande med de traditionella synsätten för att de är beprövade och fungerar. Uppsatsen har som syfte att undersöka huruvida det går att anpassa traditionell systemutveckling till objektorienterad systemutveckling.

## 2 Introduktion

Detta kapitel har för avsikt att ge en introduktion till ämnet systemutveckling. I kapitlet kommer systemutvecklingens historia att beskrivas och dessutom kommer centrala begrepp för systemutveckling att förklaras.

### 2.1 Systemutveckling

Systemutveckling är enligt Andersen (1998) en utveckling av informationssystem där systemet behandlar information. Andersen anser också att det finns vissa karakteristiska egenskaper hos ett informationssystem. Egenskaperna är enligt Andersen:

- att systemet är en mänsklig konstruktion
- att systemet är knutet till en viss arbetsuppgift
- att systemet förmedlar information
- att systemet tar emot information
- att systemet behandlar information
- att behandlingen kan vara både manuell och maskinell

Enligt Mathiassen et al. (1998) är systemutveckling en process för att utveckla ett informationssystem som underlättar användarnas arbetsuppgifter och behandling av information. Mathiassen et al. skriver att systemet ska bli ett användbart redskap för användarna i deras dagliga arbetsuppgifter, som till exempel ett boklånesystem på biblioteket eller ett biljettbokningssystem för flygresor.

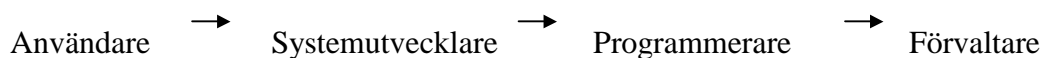
Ur historiskt perspektiv har systemutveckling existerat sedan i mitten på femtiotalet. Under femtiotalet och fram till slutet på sextiotalet fanns det ingen accepterad metod vilket gjorde att många system som producerades blev svåra att hantera för användarna (Avison & Fitzgerald, 1995). Svårigheten låg i att det inte fanns tillräckligt med dokumentation och riktlinjer om hur systemet fungerade, till exempel hur rapporter och dokument skulle produceras (Avison & Fitzgerald, 1995). Enligt Avison och Fitzgerald hade systemutveckling inte blivit tillräckligt förfinad utan att systemen som producerades var framtagna av programmerare och användare. Enligt författarna såg framtagningsprocessen för ett system såg oftast ut enligt följande:



Ett av problemen enligt Avison och Fitzgerald (1995) var att kommunikationen mellan användare och programmerare inte fungerade. Avison och Fitzgerald skriver att ett av de största problemen i kommunikationen var att programmerarna inte förstod användarnas krav och användarna inte hade programmerarnas expertkunskap. I och med problem i kommunikationen blev många system försenade och kostade mer än



beräknat. Utvecklingen fortsatte och fler personer blev involverade i systemutvecklingen vilket gjorde att kommunikationen förbättrades. Avison och Fitzgerald (1995) beskriver processen på följande sätt:



Här syns en tydlig förändring mot föregående utvecklingsprocess. Här visar Avison och Fitzgerald (1995) att kommunikationen mellan användare och programmerare har försvunnit. Istället har en systemutvecklare som har erfarenhet av att kommunicera med användare tagits till hjälp för att översätta användarnas krav på systemet till programmerarna. Detta utvecklades till att i de flesta fall blev användarna tillfredsställda med hur systemen fungerade och även att programmerarna fick en kravspecifikation som gjorde att de kunde konstruera bästa möjliga system (Avison & Fitzgerald, 1995).

Efter hand som kommunikationen mellan användare och systemutvecklare hade förbättrats kom det ytterligare hjälpmedel för utveckling av system. National Computing Centre (NCC) i Storbritannien publicerade i slutet av sextioalet en modell som vanligtvis kallas för livscykelmodellen, vilken tas upp i kapitel 2.8 (Fagerström, 1999). Den antogs av många företag under sjuttioalet och blev även kallad vattenfallsmodellen (Avison & Fitzgerald, 1995). Genom systemutvecklingens historia har det utvecklats olika synsätt på hur den ska bedrivas. En närmare beskrivning av de olika synsätten ges nedan.

## 2.2 Funktionsorienterat synsätt

Ett av de första synsätten som utvecklades var funktionsorienterat (Andersen, 1994). Detta angreppssätt var fokuserat på att identifiera funktioner, det vill säga arbetet i systemutvecklingen var uppbyggt kring att lokalisera de olika arbetsuppgifterna i ett företag (Mathiassen et al., 1998). Sedan undersöktes vilka typer av tekniska lösningar som skulle kunna vara aktuella för att få ett effektivt datasystem (Mathiassen et al., 1998). Mathiassen et al. skriver också att synsättet inte intresserar sig för till exempel anställda i sig i ett företag, utan koncentrerar sig på data de ger upphov till, och den behandling av dessa data de anställda utför.

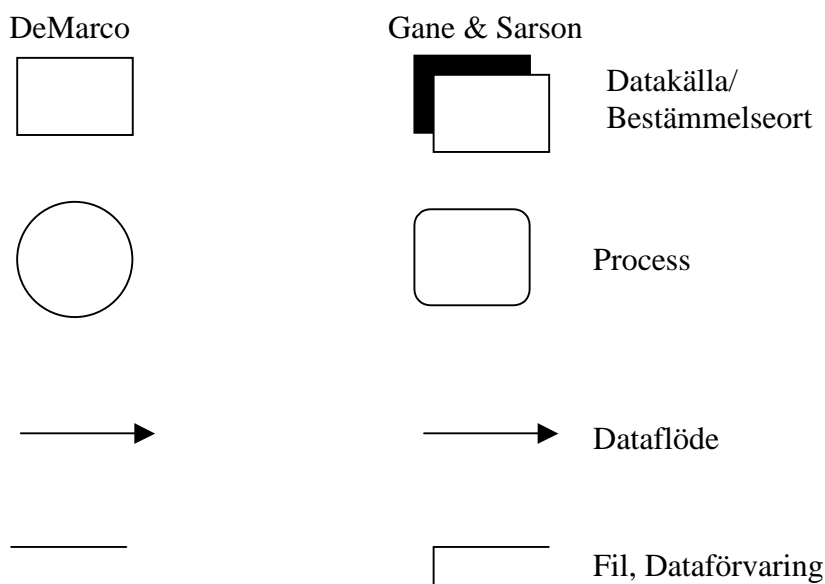
### 2.2.1 Dataflödesdiagram

Tekniker enligt denna princip är en instans av funktionsorienterat tankesätt. En av de tekniker som används med avseende på processer är dataflödesdiagram, andra är till exempel funktionell nedbrytning och strukturerat språk (Avison & Fitzgerald, 1995). Ett gemensamt drag för dessa tekniker är att de försöker hitta centrala processer för att sedan bryta ned dessa i delprocesser.

Avison och Fitzgerald (1995) skriver att dataflödesdiagram inte ska utnyttjas som den mest betydande beskrivningstekniken i en systemutvecklingsprocess, utan ska helst användas tillsammans med en annan beskrivningsteknik som till exempel

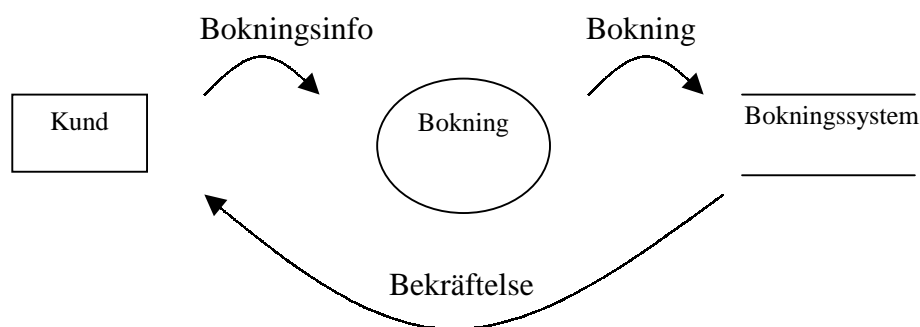
datamodellering (se kapitel 2.3.1). De två nämnda teknikerna kompletterar varandra men var för sig är det inte möjligt att utveckla ett fullständigt system som tillfredsställer användarna (Avison & Fitzgerald, 1995). Dataflödesdiagram beskriver företagets verksamhet och tar fram kraven för det nya datasystemet och datamodellering strukturerar upp informationen i datasystemet vilket gör att dessa två tekniker kompletterar varandra (Avison & Fitzgerald, 1995). Dataflödesdiagram är ett grafiskt hjälpmedel vilket och kan användas som dokumentation och dessutom som hjälpmedel för att främja kommunikationen mellan till exempel användare och systemerare (Avison & Fitzgerald, 1995). Ett dataflöde kan beskrivas som en kanal där det flödar en mängd information med ett innehåll som vi har kunskap om (Andersen, 1994). Författaren skriver också att en process gör sedan om denna inkommande information till utgående information och samtidigt som detta sker sparas informationen ned på en fil. Denna beskrivning redogör i stort sett för vad som är nödvändigt i form av notation för ett dataflödesdiagram.

I dataflödesdiagram finns det två notationer som har blivit erkända, DeMarco samt Gane och Sarson (Andersen, 1994). Grundtanken i de båda teknikerna är densamma, men det är hur notationen ska uttryckas grafiskt som skiljer sig. Notationen för de båda författarna visas i figur 1.



Figur 1: Notation för dataflödesdiagram

I fortsättningen kommer DeMarcos notation vara den som används för att visa hur ett dataflödesdiagram kommer att se ut. Ett exempel på ett dataflödesdiagram kan vara bokningssystem (se figur 2).



Figur 2: Exempel på dataflödesdiagram

Figur 2 beskriver hur en kund skickar ett dataflöde i form av bokningsinfo. Informationen omvandlas i en process till en bokning och bokningen registreras sedan i ett bokningssystem. När bokningen är registrerad skickas sedan en bekräftelse till kund.

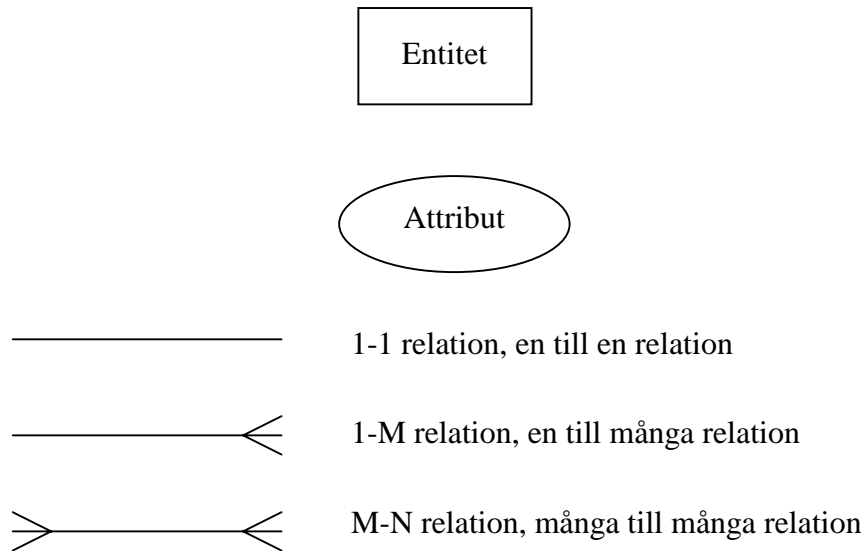
## 2.3 Dataorienterat synsätt

Mathiassen et al. (1998) skriver att utvecklingen ledde till att ett dataorienterat synsätt skapades och blev en komplettering till funktionsorienterat angreppssätt, detta för att de som utvecklade systemen behövde ett sätt att strukturera informationen och inte bara verksamheten. Enligt Andersen (1994) är det i detta synsätt intressant att hålla reda på förhållandena inom och utanför företaget och detta görs med hjälp av modellering, det vill säga man försöker avbilda verksamheten med hjälp av en modell som byggs upp av entiteter och relationer. Idén med att beskriva en verksamhet med dataorienterat synsätt är att försöka beskriva hur informationsbehovet ska täckas idag och i framtiden, samt att informationssystemet ska ge information om de entiteter som finns i och utanför verksamheten (Andersen, 1994). Ett bra exempel på ett dataorienterat synsätt är datamodellering i form av en Entitet-Relationsmodell (Avison & Fitzgerald, 1995).

### 2.3.1 Datamodellering

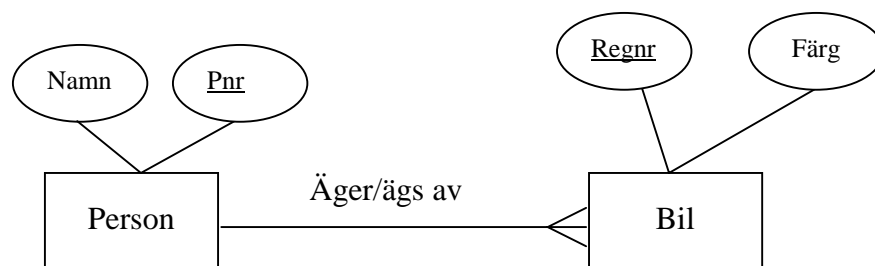
En datamodell kan beskrivas som en modell av data, vilken kan beskriva flera olika saker som till exempel en databas, information eller verkligheten (Sundgren, 1992). Enligt Sundgren finns det flera olika skäl till varför modellering kan användas, som till exempel att datamodellen ska underlätta kommunikationen mellan databaskonstruktörer och användare eller att modellen ska underlätta omorganisation och underhåll av databasen. 1976 kom ett genombrott i modelleringen då "The Entity-Relationship Model" presenterades för första gången (Andersen, 1994). Entitet-Relationsmodellen skapades av P. Chen och han gjorde med denna modell ett försök att få ett gemensamt synsätt på hur modellering skulle genomföras (Chen, 1976). Entitet-Relationsmodellen (ER-modell) beskriver hur en konceptuell modell kan skapas och dessutom skapade den en enighet om hur modellering skulle bedrivas (Andersen, 1994). En ER-modell ger företagsledningen en uppfattning om hur deras verksamhet är uppbyggd och dessutom kan den ge en möjlighet att lösa verkliga problem som är svåra att hitta en lösning till (Avison & Fitzgerald, 1995).

De begrepp som används i ER-modellering är entitet, attribut och relation. Notationen är tagen från Andersen (1994) och Avison och Fitzgerald (1995). Relationer kan representeras på flera olika sätt beroende på vilken typ av relation det är. De olika begreppen representeras grafiskt och notationen kan ses i figur 3.



Figur 3: Entitet, attribut och relationer

En entitet kan vara till exempel en person eller en bil och mellan en person och en bil är relationen att bilen ägs av en person och personen äger en bil (se figur 4). En person kan äga flera bilar och en bil kan bara ha en ägare. Attributen beskriver och identifierar ägaren med namn och personnummer. Bilen beskrivs av färg och registreringsnummer. Genom att registrera information om respektive entitet, det vill säga lagra informationen som data får man ett dataorienterat synsätt (Andersen, 1994).



Figur 4: Exempel på ER-modell

## 2.4 Objektorienterat synsätt

Ett av de senaste synsätten på systemutveckling kallas objektorientering. Detta synsätt är en utveckling av dataorienterat synsätt (Mathiassen et al., 1998). Objektorienterad systemutveckling kännetecknas av att den är iterativ och en kombination av top-down och bottom-up tänkande (Fagerström, 1992). Den iterativa processen kännetecknas av att samma aktiviteter utförs gång på gång på olika detaljnivå och inom olika delar av

en verksamhet (Fagerström, 1992). Top-down och bottom-up tänkandet presenteras mer utförligt i kapitel 2.6.

### 2.4.1 Objekt och klass

Det objektorienterade synsättet har som utgångspunkt objekt som samarbetar med varandra. Ett objekt kan till exempel vara en person eller ett flygplan. Enligt den nya standarden för objektorienterad modellering (Unified Modeling Language, UML) så beskrivs ett objekt på följande sätt:

*”An object is an item we can talk about and manipulate. An object exists in the real world (or more precisely, our understanding of the real world).”* (Eriksson & Penker, 1998, sid 65)

Ett objekt behöver alltså inte bara vara konkreta ting i omgivningen utan det kan också vara abstrakta ting. Ett exempel på ett abstrakt objekt kan vara en speciell klass som aldrig får en instans av något objekt, som till exempel en klass fordon kan delas in i bil och MC. Detta medför att alla objekt instansieras till bil eller MC och inte fordon. I Mathiassen et al. (1998) - som är skriven med UML som grund för deras systemutvecklingsmetod - kan man hitta följande definition av ett objekt:

*”Objekt: en helhet med identitet, tillstånd och beteende.”*  
(Mathiassen et al., 1998, sid19)

I denna definition finns beteende med. Detta utgör en väsentlig skillnad mot entitet (se delkapitel 2.3.1) som är en statisk enhet som bara kan lagra information, medan däremot ett objekt kan dessutom ta emot meddelande och efter en viss tid exekvera kod (Fagerström, 1999). Ett objekt har ett eller flera attribut kopplade till sig. Attribut innehåller information om objekt som till exempel vilken färg det har, och detta medför att i attributen finns det möjlighet att representera vilket tillstånd ett objekt har (Fagerström, 1999).

I ett systemutvecklingsprojekt kan flera olika objekt tänka sig att det förekomma. När detta händer försöker man hitta de objekt som har ett gemensamt mönster och ett gemensamt beteende (Fagerström, 1999). När egenskaperna mellan objekt förefaller att vara gemensamma beskrivs detta som en klass, det vill säga en samling av objekt med liknande egenskaper. Mathiassen et al. (1998) beskriver en klass på följande sätt:

*”Klass: en beskrivning av en mängd objekt med samma struktur, beteendemönster och attribut.”* (Mathiassen et al., 1998, sid 19)

Klasser och objekt är grundstenarna i objektorienterad modellering och för att beskriva hur klasser är kopplade till varandra används relationer (Mathiassen et

al., 1998). En närmare beskrivning av några olika typer av relationer som kan förbinda de olika klasserna presenteras nedan.

### 2.4.2 Objektorienterade relationer

Objektorienterad modellering använder sig av flera olika typer av relationer som till exempel arv, association, aggregat (Fagerström, 1999). Arv är en av de viktigaste relationerna i objektorientering och arv sker mellan olika klasser. I denna typ av relation förutsätts det att de involverade klasserna är över- och underordnade (Mathiassen et al., 1998). Ett arv från en klass till en annan medför att attribut och metoder ärvs med samma utseende som den klass den ärver ifrån (Fagerström, 1999). Klasserna modelleras som över- och underordnade vilket medför att arv kan bara ske från en överordnad till en underordnad klass (Mathiassen et al., 1998). Ett annat ord för arv är generalisering och denna relation kan definieras enligt följande:

*”Generalisering: en struktur i vilken en generell klass (superklassen) beskriver egenskaper och beteendemönster som är gemensamma för ett antal speciella klasser (subklasserna) och går i arv till dessa.”*  
(Mathiassen et al., 1998, sid 97)

I en relation som association kan ett eller flera objekt vara inblandade. De inblandade objekten förutsätts här att vara sidordnade (Mathiassen et al., 1998). Detta innebär att det inte finns någon hierarki som i arvsrelationen, utan klasserna är likvärdiga (Mathiassen et al., 1998). Ett vanligt sätt att beskriva relationen är att objekten ”känner till” varandra (Mathiassen et al., 1998).

*”En association är en strukturell, permanent, dubbelriktad koppling mellan objekt (dvs objekten kommer ihåg kopplingen mellan metदानrop).”* (Fagerström, 1999, sid. 39)

Aggregat kan bestå av två eller flera objekt och på liknande sätt som arv finns här över- och underordnade objekt (Mathiassen et al., 1998). Relationen kan beskrivas med att ett underordnat objekt ”ingår-i”, eller att ett överordnat objekt ”består-av” (Mathiassen et al., 1998).

*”Ett aggregat är en association mellan ett objekt (en helhet) och ett antal andra objekt (delar). Delobjektens livslängd beror på helhetens livslängd, om helheten tas bort måste även delobjekten tas bort.”*  
(Fagerström, 1999, sid 41)

De relationer som har presenterats ovan är de tre mest fundamentala relationerna för objektorienterat synsätt. Med hjälp av klasser och relationerna mellan dessa finns det möjlighet att modellera ett företags informationsstruktur.

### 2.5 Unified Modeling Language (UML)

I början av 1990-talet existerade det många olika inriktningar på hur man skulle bedriva en objektorienterad systemutveckling (Rational University, Rational Software Corporation, 1 Mars 2000). I början av nittiotalet utkristalliserades tre huvudinriktningar (Rational University, Rational Software Corporation, 1 Mars 2000). En av inriktningarna stod Dr. Ivar Jacobsen bakom, och han är en av dem som har arbetat med objektorienterad systemutveckling en längre tid (Rational University, Rational Software Corporation, 1 Mars 2000). Han startade företaget Objectory som utvecklade en produkt (Objectory) som stöd för objektorienterad systemutveckling.

En annan inriktning stod Grady Booch bakom. Han utvecklade Booch-metoden vilken understödjer objektorienterad systemutveckling från analys till design (Rational University, Rational Software Corporation, 1 Mars 2000). Denna metod tog tillvara på de fördelar som finns med objektorientering, som till exempel återanvändning och gjorde det ekonomiskt försvarbart att börja arbeta på detta sätt (Rational University, Rational Software Corporation, 1 Mars 2000).

Den sista av de tre huvudinriktningarna skapades av Dr. James Rumbaugh. Han var huvudansvarig för utvecklingen av en utvecklingsmodell som heter Object Modeling Technique (OMT) (Rational University, Rational Software Corporation, 1 Mars 2000). Dr. Rumbaugh har utvecklat flera verktyg i OMT för att underlätta användandet av denna utvecklingsmodell, och han har dessutom försökt att hitta fler verktyg och metoder för att underlätta objektorienterad systemutveckling (Rational University, Rational Software Corporation, 1 Mars 2000).

I mitten på nittiotalet blev det plötsligt en förändring. De tre huvudinriktningarnas skapare gick samman och började samarbeta i Rational Software Corporation vilket ledde till att Unified Modeling Language (UML) skapades. UML blev antagen som standard (1997) för objektorienterat modelleringsspråk av Object Management Group (OMG) som är standardorganet för objektorientering (Rational University, Rational Software Corporation, 1 Mars 2000). Denna standard accepterades direkt av erkända företag som IBM och Microsoft (Rational University, Rational Software Corporation, 1 Mars 2000).

Unified Modeling Language (UML) är en omfattande standard, i form av olika diagram, vilket gör det svårt att applicera hela standarden på ett systemutvecklingsprojekt (Fagerström, 1999). Fagerström skriver att istället får vissa delar väljas som anses vara relevanta i det systemutvecklingsprojekt som man arbetar med. UML omfattar grundläggande begrepp och notation som tidigare hade varit definierade på olika sätt i olika metoder (Fagerström, 1999). I och med UML har definitioner blivit gemensamma för hela det objektorienterade synsättet (Mathiassen et al., 1998).

Målet med UML är bland annat att underlätta för de företag som framställer mjukvara (Rational Software Corporation, 1999). Genom att modellera i UML fås en ritning över hur ett program ska se ut och att det har en korrekt struktur (Rational Software

Corporation, 1999). Modelleringen ger också en möjlighet att automatisera processen med framtagande av mjukvara genom att UML innehåller möjligheten att arbeta med samma typ av modellering oberoende av plattform och programspråk (Rational Software Corporation, 1999). UML består av olika komponenter som till exempel olika typer av diagram vilka gör det möjligt att visualisera ett komplext system (Rational Software Corporation, 1999).

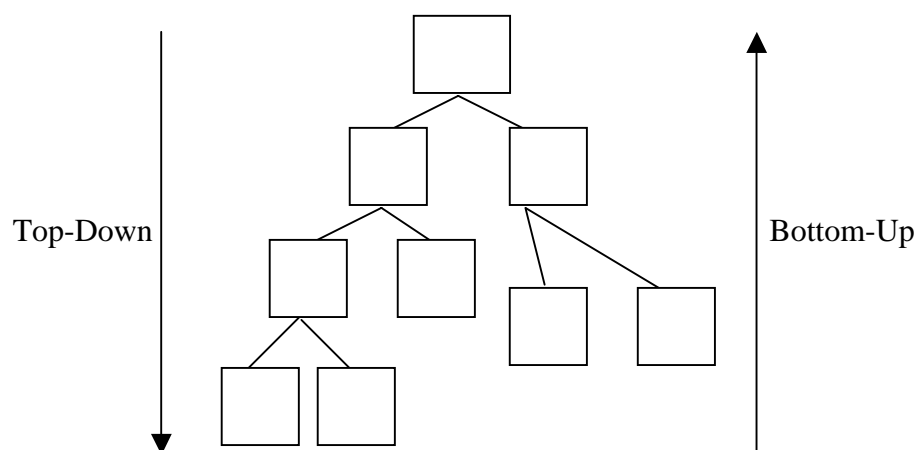
Unified Modeling Language har möjlighet att beskriva ett system statiskt och dynamiskt (Rational Software Corporation, 1999). Den statiska beskrivningen sker genom att använda sig av klassdiagram (Rational Software Corporation, 1999). Diagrammet beskriver vilka objekt som ska existera i ett system och vilka detaljer dessa ska ha (Rational Software Corporation, 1999). Den dynamiska beskrivningen sker genom att använda sig av tillståndsdigram och interaktionsdiagram (Rational Software Corporation, 1999). Ett interaktionsdiagram kan till exempel vara ett sekvensdiagram eller ett samspelsdiagram (Rational Software Corporation, 1999). Interaktionsdiagram har som uppgift att visa hur olika objekt interagerar med varandra (Rational Software Corporation, 1999). Den dynamiska modelleringen beskrivs närmare i kapitel 9.2.3.

## 2.6 Generella arbetssätt för systemutveckling

Det finns flera olika sätt att angripa ett projekt i systemutveckling. Ett av de synsätt som används i de flesta utvecklingsprocesser kallas top-down (Fagerström, 1999). Detta angreppssätt börjar med att man definierar ett överordnat problem för att sedan bryta ner detta till delproblem som i sin tur bryts ner i delproblem osv. Fagerström (1999) skriver också att en av de positiva effekterna med detta synsätt är att man får en överblick över systemet men en av svagheter som är väsentlig för objektorienterad systemutveckling är att återanvändning av komponenter försvåras. Återanvändningen försvåras ju längre arbetet fortsätter för att de komponenter som ska användas igen är svåra att hela tiden hålla tillgängliga i systemutvecklingen (Fagerström, 1999). Figur 5 visar hur en top-down metod kan fungera. Det börjar med att man hittar ett centralt problem för att sedan bryta ned detta i delproblem, och så fortsätter det tills problemen har blivit tillräckligt små för att kunna hanteras i systemutvecklingsprojektet. Ett exempel på en typisk top-down metod är till exempel funktionell nedbrytning där man hittar en central funktion och bryter ner den i delfunktioner (Fagerström, 1999).

Ett annat synsätt för att angripa en utvecklingsprocess är att använda sig av bottom-up metoder. I detta synsätt använder man sig av omvänd teknik från top-down, det vill säga man angriper problemet genom att hitta enskilda komponenter för att sedan bygga ihop dessa till ett färdigt system (Fagerström, 1999). Fagerström skriver också att en av styrkorna i detta synsätt är att återanvändning är mycket förenklad, men när systemet utvecklas är det svårt att få en helhetsbild. Ett exempel på denna typ av angreppssätt är objektorientering.



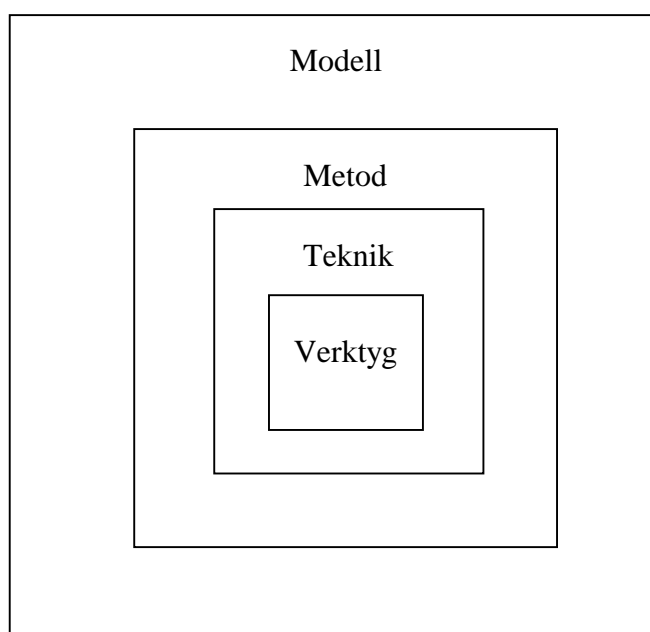


Figur 5: Arbetsätt för systemutveckling

Figur 5 beskriver hur top-down respektive bottom-up metoder fungerar. Som beskrivet ovan finns det rena top-down eller bottom-up metoder. En möjlighet är att kombinera dessa två metoder för att erhålla en bra systemöversikt och dessutom göra det lättare att återanvända komponenter i systemet (Mathiassen et al., 1998). Objektorienterat synsätt är exempel på ett kombinerat arbetssätt (Mathiassen et al., 1998). Objektorienterat angreppssätt kan således presenteras som antingen en kombinerad eller bottom-up metod.

## 2.7 Systemutvecklingens hjälpmedel

I ett systemutvecklingsprojekt finns det i många fall flera olika hjälpmedel som används. Hjälpmedel som innefattar hela systemutvecklingen är modell, metod, teknik och verktyg (Andersen, 1994). Begreppen återfinns idag i många företag, däribland IBS Konsult AB.



Figur 6: Systemutvecklingens hjälpmedel

I figur 6 beskrivs hur systemutvecklingens olika hjälpmedel är sammanbundna. En modell är ett ramverk för hur systemutvecklingen ska gå till (Andersen, 1994). Författaren skriver att metoden har en mer detaljerad beskrivning om hur man ska gå till väga. Teknikerna är hjälpmedel för att kunna realisera metodens teorier (Andersen, 1994). Verktygen är till för att praktiskt kunna förverkliga de olika teknikerna (Andersen, 1994).

Ett medel för att få en framgångsrik systemutveckling kan vara att ha metoder att applicera på en genomarbetad modell (Mathiassen et al., 1998). Modeller beskriver i stora drag vem som ska utföra ett arbete och även vilket arbete som ska utföras (Andersen, 1994). I en modell finns det begrepp som till exempel faser, metodsteg, problemområden och alla dessa uttryck är olika benämningar för de delar som en modell kan innehålla (Andersen, 1994). Han skriver att en modell kan ses som ett ramverk för ett systemutvecklingsprojekt. En av de modeller som är och har varit betydelsefull för systemutvecklingen är livscykelmodellen (Avison & Fitzgerald, 1995). Modeller för systemutveckling har utvecklats men de flesta av dessa har grundkoncepten från livscykelmodellen kvar i sin modell, vilket gör att den finns på ett eller annat sätt närvarande i många olika systemutvecklingsmodeller (Lewis, 1994). Livscykelmodellen beskriver hur ett projekt för systemutveckling kan fortlöpa och hur ett systems "liv" kan se ut (Andersen, 1994). Den beskriver dessutom hur arbetsuppgifterna är fördelade och vem som ska delta i de olika faserna för modellen (Andersen, 1994). Den beskrivs närmare i delkapitel 2.8.

En metod är en detaljerad beskrivning för hur det är möjligt att lösa ett visst problem (Andersen, 1994). Det finns idag en mängd olika metoder och dessa är inriktade på hur man ska lösa en specifik problemtyp (Andersen, 1994). Han skriver också att den största skillnaden mellan en modell och en metod är att metoder är mer detaljerade och inte så generella som en modell. De inriktningar som har utvecklats finns representerade av olika metoder som har arbetats fram, som till exempel Ethics, ISAC, STRADIS. Ethics (Effective Technical and Human Implementation of Computer-Based Systems) skapades av E. Mumford 1995 och är en människoorienterad utvecklingsmetod där användarna ska vara i centrum (Avison & Fitzgerald, 1995). För att denna metod ska fungera är det viktigt att informationssystemet utvecklas i nära samband med ett företags organisation och systemets användare (Avison & Fitzgerald, 1995). ISAC skapades av en forskningsgrupp vid Kungliga Tekniska Högskolan och Stockholms Universitet 1971 (Avison & Fitzgerald, 1995). ISAC är en förkortning som betyder Information Systems Work and Analysis of Changes (Avison & Fitzgerald, 1995). I metoden ligger tyngdpunkten på att försöka hitta användarnas problem och föreslå olika angreppssätt för att lösa problemen (Avison & Fitzgerald, 1995). STRADIS (Structured Systems Analysis) skapades av Gane och Sarson 1979 och har som viktigaste del att arbeta med funktionell nedbrytning i form av dataflödesdiagram (Avison & Fitzgerald, 1995). Enligt Andersen (1994) kan man karakterisera en metod enligt följande:

- Vilken typ av problem som metoden ska appliceras på
- Vad det är för typ av arbete som ska utföras, och hur ska det organiseras

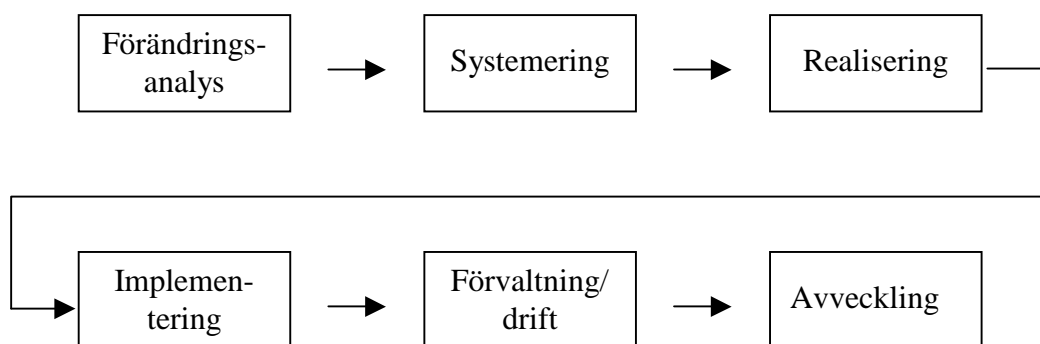
- Vad det är för beskrivningstekniker som ska användas, samt hur dessa ska användas

I en metod kan det finnas en eller flera beskrivningstekniker och teknikerna har för det mesta tydliga regler för hur symboler och text får användas (Andersen, 1994). Enligt Andersen finns det olika krav man kan ställa på en teknik som till exempel hur lätt det är att lära sig tekniken eller hur lätt det är att ändra och reproducera tekniken. Han säger också att en teknik ska vara möjlig att applicera på flera olika typer av verksamheter och dessutom vara lättläst. Tekniken bör ge medarbetarna en bra bild av verksamheten de försöker utforma ett system för (Andersen, 1994). Det har utvecklats många olika tekniker som till exempel rika bilder, entitetsmodellering, (Avison & Fitzgerald, 1995). Rika bilder ger en överblick av organisationen på ett sådant sätt att den fungerar som en karikatyr av verksamheten (Avison & Fitzgerald, 1995). Entitetsmodellering däremot är till för att analysera data i verksamheten och är oftast kopplade till dataorienterade metoder (Avison & Fitzgerald, 1995). Ett annat ord för denna typ av modellering är datamodellering. De tekniker som är beskrivna ovan är grafiska, vilka ger fördelen att det är lättare att få en överblick av vad man håller på med, samt att medarbetare får det lättare att sätta sig in i problemet (Andersen, 1994). En nackdel med dessa tekniker är att när ett system blir stort och komplext försvinner överskådligheten (Andersen, 1994).

Modeller, metoder och tekniker har många regler för hur man ska gå till väga men för att kunna presentera något konkret behövs ett verktyg (Andersen, 1994). I dagens systemutveckling finns det metoder som förutsätter att utvecklarna har tillgång till vissa datorstödda verktyg vilka går under benämningen CASE verktyg (Computer Aided Systems Engineering) (Avison & Fitzgerald, 1995). Det finns flera fördelar med CASE verktyg som till exempel att det underlättar beskrivningsarbetet, beskrivningarna blir mer strukturerade och att det underlättar möjligheten att angripa mer omfattande problem (Andersen, 1994). Nackdelen med CASE verktyg är att de kan vara dyra att köpa in (Andersen, 1994).

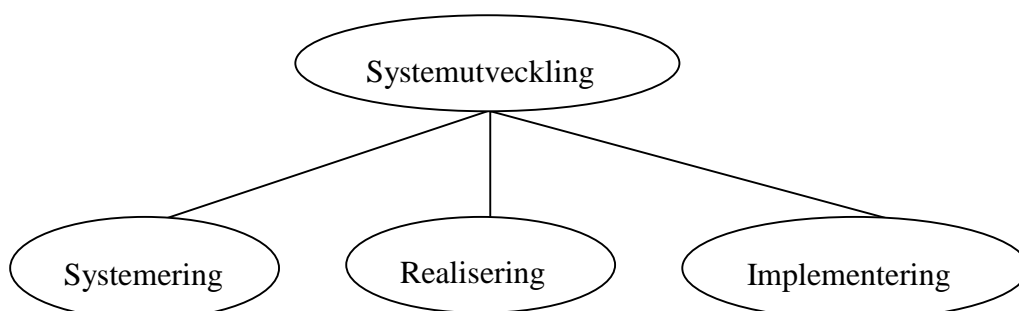
## 2.8 Livscykelmodell

Enligt Andersen (1994) är livscykelmodellen ett exempel på hur informationssystemets "liv" kan följas, det vill säga den följer hur systemet går igenom de olika faserna som finns för att utveckla ett system tills det att systemet inte längre är tillräckligt för att kunna hantera informationen i verksamheten och avvecklas. Fördelar med denna modell är att det finns möjlighet för kunden att följa hur arbetet utvecklas och att det håller tidsplanen (Avison & Fitzgerald, 1995). En nackdel med livscykelmodellen är till exempel att modellen inte är flexibel, det vill säga att när en fas officiellt är avslutad är det svårt eller omöjligt att gå tillbaka och förändra i föregående fas (Avison & Fitzgerald, 1995). Enligt Andersen (1994) och Avison och Fitzgerald (1995) består livscykelmodellen av följande delar som beskrivs i figur 7.



Figur 7: Livscykelmodellen

I livscykelmodellen finns det klara skillnader mellan systemutveckling och systemering (Andersen, 1994). Många användare och systemutvecklare använder dessa två ord som synonymer men enligt livscykelmodellen så är det skillnad mellan dessa uttryck (Andersen, 1994). Systemutveckling innefattar systemering, realisering och implementering och som kan ses i figur 8 (Andersen,1994). Systemering beskrivs närmare i delkapitel 2.8.2.



Figur 8: Systemutvecklingens komponenter

För att kunna få en större förståelse för vad som utförs i de olika delarna i livscykelmodellen följer en mer detaljerad beskrivning av de enskilda faserna.

### 2.8.1 Förändringsanalys

I denna första fas undersöks det nuvarande systemet, vilka krav det systemet hade på sig när det utvecklades, vilka problem det var att tillmötesgå kraven och vilka nya krav som har tillkommit (Avison & Fitzgerald, 1995). När de nya kraven kommer fram måste det undersökas att de inte bryter mot någon policy inom företaget, att de skulle fungera i organisationen och socialt hos användarna samt att kraven ska vara ekonomiskt försvarbara och tekniskt genomförbara (Avison & Fitzgerald, 1995).

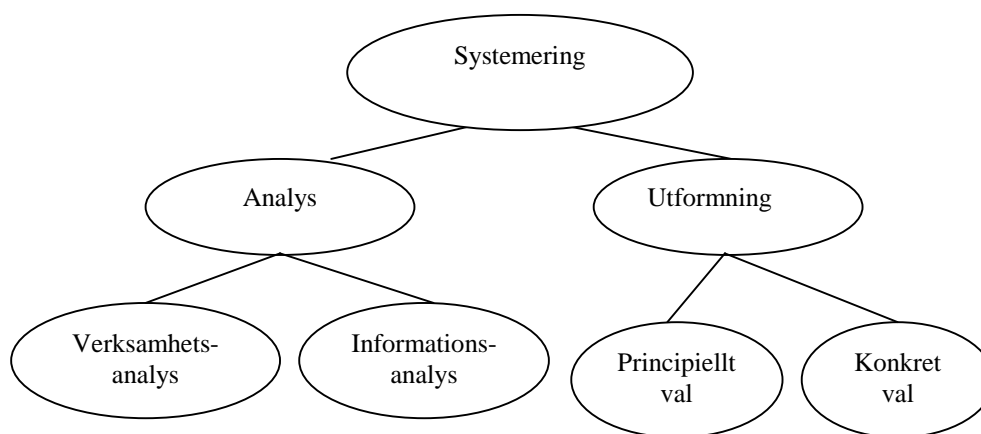
Det är två saker som står i centrum i denna fas, verksamhetens problem och möjligheter (Andersen, 1994). Andersen säger att de olika arbetsuppgifterna är att beskriva nuläget, önskad situation och förändringsbehov. Här är det fördelaktigt med ett samarbete mellan användare och experter (Andersen, 1994). Användarna har

kunskap om nuläget och önskemålen som finns för förbättringar (Andersen, 1994). Experterna har kunskap om hur man skall strukturera informationen som kommer in på ett sätt så att de inblandande får en översikt (Andersen, 1994).

Förändringsbehov kan vara en utveckling av systemet eller framtagning av en ny organisation. Andersen (1994) understryker hur viktigt det är för experterna att vara uppmärksamma och skilja mellan systemkrav och organisationskrav. Vidare skriver Andersen (1994) att i förändringsanalysen kan det uppkomma krav och problem som ett nytt system inte kan lösa, utan att det behövs en organisationsförändring. PSO-utveckling, vilket står för person-, system- och organisationsutveckling, blir följden av att systemutvecklarna uppmärksammar att det måste ske en utveckling av organisationen och personerna i verksamheten samtidigt som en systemutveckling sker (Andersen, 1994). En PSO-utveckling ger en grund för att användarna får en större nytta av systemet, det vill säga systemet gör att användarnas arbetsuppgifter underlättas och inte blir så komplexa (Andersen, 1994). När förändringsanalysen avslutas har man kommit till en beslutspunkt (Avison & Fitzgerald, 1995). Beslutet som fattas ska bestämma om ett nytt system ska utvecklas eller ej (Avison & Fitzgerald, 1995). Om ett nytt system ska införas, tar nästa fas vid, nämligen systemeringen.

### 2.8.2 Systemering

Denna fas i modellen består av två delar, analys och utformning. Denna uppdelning beskrivs i figur 9.



Figur 9: Systemeringens två faser

I figur 9 visas det hur analysen kan delas upp i verksamhetsanalys och informationsanalys (Andersen, 1994). I verksamhetsanalysen utreds vilket stöd som det nya systemet ska ge verksamheten (Andersen, 1994). Författaren skriver att här analyseras hur systemet ska göra det lättare för medarbetarna att utföra sina uppgifter. I denna fas är det viktigt att både systemerare och representanter för användarna finns närvarande (Andersen, 1994). Verksamhetsanalysen kan beskrivas med grafiska tekniker som dataflödesdiagram för att få en överblick över organisationen (Avison & Fitzgerald, 1995) I informationsanalysen identifieras vilken data som

informationssystemet ska innehålla och det görs genom ett samarbete mellan användare och systemerare (Andersen, 1994). Denna analys kan göras med olika typer av diagram som till exempel ER-modellen som visar hur informationen ska vara strukturerad i verksamheten (Avison & Fitzgerald, 1995).

Analysen är den viktigaste delen av hela systemutvecklingsprojektet och resultatet från denna fas i systemeringen är kravspecifikationen (Andersen, 1994). Kravspecifikationen är ett dokument som ska klargöra vilka behov och krav som ställs av företaget på det nya systemet (Andersen, 1994). Om analysen blir undermålig är det svårt att lyckas med ett systemutvecklingsprojekt, för att det som inte kommer fram under analysen kommer heller inte med i kravspecifikationen som ligger till grund för det nya systemet (Andersen, 1994).

Nästa fas i systemeringen är utformning av systemet. Grunden till utformningen är kravspecifikationen. Utformningen består av två delar, principiellt val och konkret val av teknisk lösning (Andersen, 1994). När den principiella tekniska lösningen ska väljas är det viktigt att välja den mest ändamålsenliga lösningen för att bästa möjliga resultat uppnås (Andersen, 1994). När det gäller att göra det konkreta valet är det viktigt att se över sin arsenal av verktyg och se om något passar eller så får man införskaffa ett verktyg som passar (Andersen, 1994). I utformningen av systemet är det systemerare och programmerare som samarbetar (Andersen, 1994). Enligt Avison och Fitzgerald (1995) är det också viktigt att se på de problem som det gamla systemet hade och varför de uppstod.

### **2.8.3 Realisering**

I realiseringen ska det nya informationssystemet utarbetas och systemet skapas med hjälp av programmerare som skapar ADB program (Automatisk Data Behandling) (Andersen, 1994). Det är viktigt att programmerarna dessutom beskriver hur deras tekniska lösning ser ut i någon form av dokumentation, för att det ska vara möjligt att se hur systemet är strukturerat (Andersen, 1994). Här är det också viktigt att försöka vara systematisk i programmeringen så att rätt delprogram görs vid rätt tidpunkt för att realiseringen ska förlöpa på ett så problemfritt sätt som möjligt (Avison & Fitzgerald, 1995).

### **2.8.4 Implementering**

I denna fas startas hela det nya systemet. Här är det viktigt att användare, systemerare och programmerare samarbetar, för om det blir några problem så finns det möjlighet att analysera dem från olika synvinklar (Andersen, 1994). I och med att det är ett nytt system blir det också nya rutiner som användarna ska lära sig och de får lära sig rutinerna genom utbildning och anvisningar från systemerare/systemutvecklare (Andersen, 1994). I och med denna fas så tar systemutvecklingen slut. En viktig del i denna fas är kvalitetskontroll (Avison & Fitzgerald, 1995). Kontrollen ska vara så noggrann att både programmerare, systemutvecklare och användare är nöjda (Avison & Fitzgerald, 1995).

### **2.8.5 Förvaltning och drift**

Under denna fas är syftet att underhålla systemet så det fungerar problemfritt för alla användare (Andersen, 1994). Om det måste göras större förändringar får man gå tillbaka till fas ett och gå igenom hela livscykeln igen, det vill säga när en förändring ska ske så är det upp till förvaltaren att avgöra om det är nödvändigt att gå igenom hela livscykeln på nytt (Andersen, 1994). Denna bedömning är individuell och får göras från fall till fall. När en förändring har skett är det hela tiden viktigt att försöka ha en uppdaterad dokumentation som stämmer med hur systemet verkligen ser ut (Andersen, 1994).

### **2.8.6 Avveckling**

Ett informationssystem kan inte förvaltas och förändras i all evighet (Avison & Fitzgerald, 1995). När förändringarna blir för stora och kommer för ofta avvecklas det gamla systemet och ett nytt system tar vid (Andersen, 1994). Om ett äldre system är i en ständig förändring får verksamhetsledningen avväga om det är lönsamt att ha kvar systemet och undersöka ifall användarna är tillfredsställda (Andersen, 1994). Om inte, kan ett nytt system vara en lösning (Andersen, 1994). Ifall ledningen bestämmer sig för att utveckla ett nytt system sker en avveckling av det gamla systemet när det nya är redo att ta vid.

### 3 Problembeskrivning

I många år har de traditionella synsätten på systemutveckling dominerat marknaden och idag har marknaden störst antal systemutvecklingsmetoder enligt de äldre och traditionella synsätten (Mathiassen et al., 1998). En bidragande orsak till detta kan vara att i över tjugo år har verktyg utvecklats för att hjälpa till med systemutveckling enligt ett funktions- och dataorienterat synsätt. Det existerar idag verktyg som stödjer den traditionella systemutvecklingsprocessen från analys till design. Objektorienterade verktyg för systemutveckling har först och främst utvecklats under nittioalet vilket gör att de inte har nått fram till samma förmåga som de äldre synsättens verktyg har.

I och med att de äldre synsätten är beprövade och till synes fungerar på ett tillfredsställande sätt, så har objektorienterad systemutveckling hamnat i skymundan. Enligt Karlander (1998) var det i början på nittioalet många som hade förutspått att det objektorienterade synsättet skulle bli ett arbetssätt som skulle bli allmänt vedertaget innan år 2000. Detta har inte hänt och de flesta håller kvar vid de synsätt som är gamla och beprövade (Karlander, 1998).

Ett annat problem som har gjort det svårt för objektorientering att hävda sig är att det inte har funnits någon accepterad standard som har kunnat användas. De traditionella synsätten fick en modell som de kunde följa vilket gjorde det möjligt för angreppssätten att utvecklas (se delkapitel 2.1). Objektorienterade modelleringspråk fick inte en standard förrän i slutet på nittioalet och detta har gjort att objektorientering inte har blivit ett synsätt som företagen har accepterat till fullo (se delkapitel 2.1).

Ur det traditionella synsättet på systemutveckling kan till exempel datamodellering och dataflödesdiagram användas som beskrivningstekniker för att modellera ett företag. Den traditionella modelleringen i form av ER-modellen går ej att använda i ett objektorienterat synsätt för att den inte kan beskriva detaljrikedomen i det objektorienterade synsättet, i stället får UML användas som beskrivningsteknik (Mathiassen et al., 1998). UML är ett modelleringspråk för objektorienterad systemutveckling (Mathiassen et al., 1998). Det objektorienterade angreppssättet är en utveckling av dataorienterade synsättet för hur ett informationssystem kan utvecklas (Mathiassen et al., 1998).

#### 3.1 Problemställning

För de faser som ingår i livscykelmodellen (förändringsanalys, systemering (analys och utformning), realisering, implementering, förvaltning/drift och avveckling) undersöks det om det är möjligt att med hjälp av objektorienterade tekniker och begrepp ersätta de traditionella tekniker som är funktions- och dataorienterade. De tekniker som kan anses som traditionella är till exempel dataflödesdiagram och datamodellering enligt ER-modellen. Arbetet kommer att försöka svara på följande fråga:



*Kan objektorienterad modellering i form av UML ersätta traditionella tekniker i livscykelmodellens olika faser?*

Det centrala i uppsatsen kommer alltså att vara att undersöka om det är möjligt att ersätta de traditionella teknikerna med UML för att på så sätt få ett modelleringsspråk enligt det objektorienterade synsättet. Den objektorienterade modelleringen kommer att implementeras med UML, för att detta notationssätt gäller idag som standard för modellering inom objektorienterat synsätt.

### **3.2 Avgränsning**

Detta arbete begränsas till att göra en jämförelse av förändringsanalys och analysdelen av systemeringen. I livscykelmodellen förekommer även realisering, implementering, förvaltning/drift och avveckling. Realisering och implementering ingår i systemutveckling, men för att arbetet ska vara möjligt att realisera inom tidsramarna görs en begränsning till de första två problemområdena vilka är förändringsanalys och systemeringens första delfas i livscykelmodellen. Förvaltning/drift är oftast en repetition av livscykeln vilket gör att denna del inte är väsentlig för detta arbete och avveckling är ett specifikt problemområde i sig som inte är likt något av de tidigare problemområdena i livscykeln och är därför ointressant för detta arbete. Inom systemutvecklingen förekommer det flera olika modeller men detta arbete har avgränsat sig till att använda den traditionella livscykelmodellen som grund. Med den traditionella livscykelmodellen avses den modell som utvecklades av NCC i Storbritannien 1969 (Daniels & Yeates, 1971).

Det finns en mängd olika metoder och tekniker som har utvecklats i det traditionella perspektivet. Dataflödesdiagram och ER-modell är de två teknikerna som kommer att representera det traditionella synsättet i detta arbete. I objektorienterad modellering med UML som grund är det bara den statiska modelleringen som är intressant för denna jämförelse. En begränsning kommer också att göras i fallstudien där de tre första faserna är aktuella i detta arbete.

### **3.3 Förväntat resultat**

Detta arbete kommer att påvisa hur det är möjligt att översätta förändringsanalys och systemering enligt ett data- och funktionsorienterat synsätt till ett objektorienterat synsätt med livscykelmodellen som utgångspunkt. Denna ersättning av traditionella tekniker ska realiseras med UML som modelleringsteknik för det objektorienterade tankesättet. Det är också viktigt att försöka utveckla ett objektorienterat arbetssätt för hur analysfasen ska fungera, det vill säga vad är viktigt att koncentrera sig på när man utvecklar ett informationssystem enligt ett objektorienterat synsätt.

## 4 Metod

Med arbetets problemställning som grund fanns det möjlighet att använda flera olika metoder. Metodvalet är en förklaring av hur det preciserade problemet löses med hjälp av ett empiriskt tillvägagångssätt (Backman, 1998). Vidare skriver Backman att metodsektionen på ett detaljerat sätt beskriver metoden för att replikation och evaluering ska vara möjligt. Replikation innebär att undersökning ska vara möjlig att göra om under exakt samma förhållanden och evalueringen gör det möjligt att kunna värdera det empiriska förfarandet (Backman, 1998).

Under flera århundraden har processen för forskning utvecklats och detta har gjort att flera olika inriktningar på forskningsprocessen kan användas (Andersen, 1994). En av de första inriktningarna som utvecklades heter empirism. Detta synsätt hävdar att all kunskap bygger på observationer av verkligheten (Andersen, 1994). Positivismen har sin föregångare i empirismen och de båda synsätten har flera punkter som är lika, till exempel att de båda metoderna bygger på iakttagelser från verkligheten (Andersen, 1994). Positivismen skapades med fysiken som förebild av en fransk sociolog vid namn Auguste Comte (Patel & Davidsson, 1994). Synsättet var ett försök att skapa en vetenskaplig metodologi som var lika för alla vetenskaper och detta skulle resultera i att alla vetenskaper skulle vara uppbyggda på ett och samma sätt, detta brukar kallas vetenskaplig monism (Patel & Davidsson, 1994). Från positivismen utvecklades den logiska positivismen (Andersen, 1994). Detta synsätt på forskningen ville få fram en enhetlig vetenskap som skulle utgå från det innehåll man fick från erfarenheten vid direkta observationer (Andersen, 1994). Angreppssättet skulle utvecklas med hjälp av den formella logiken (Andersen, 1994).

Ett synsätt som anses vara motsatsen till positivismen är hermeneutik (Patel & Davidsson, 1994). Hermeneutik har utvecklats under flera århundraden, och till en början var den en metod för att tolka bibeltexter för att under nittonhundratalet utvecklas till att bli en existentiell filosofi med grund i mänsklighetens existens (Patel & Davidsson, 1994). Detta kan jämföras med positivismens synsätt där allt utgår från fysiken eller logiken. Skillnaderna mellan positivismen och hermeneutiken finns på flera olika plan. Några av de viktigaste skillnaderna är att positivismen försöker göra en supermetodologi och försöker vara objektiv men hermeneutiken är mer uppbyggd på förståelse och den blir subjektiv utifrån forskarens egen uppfattning (Patel & Davidsson, 1994). Denna subjektivitet kommer ifrån forskarens egen världsuppfattning och med den som utgångspunkt försöker forskaren tolka de olika iakttagelser som görs i observationen (Patel & Davidsson, 1994).

En forskningsprocess kan även klassificeras med hjälp av begreppen kvantitativ eller kvalitativ. Kvalitativt och kvantitativt synsätt är inte varandras motsatser utan det är snarare två synsätt som har gemensamma kännetecken som till exempel att utgångspunkten är att ett fenomen studeras (Andersen, 1994). Det kvalitativa angreppssättet på forskningen är en undersökning av fenomen som består av en unik sammansättning av egenskaper eller kvaliteter (Andersen, 1994). På grund av att egenskaperna eller kvaliteterna inte kan mätas på ett objektivt sätt utgör detta en skillnad mot ett kvantitativt synsätt (Andersen, 1994). Man kan här se en koppling mellan kvalitativt synsätt och hermeneutiken där man skall tolka situationer som

observationen ger (Andersen, 1994). Kvantitativt synsätt utmärks med att det som ska studeras ska vara mätbart och resultatet ska sedan presenteras numeriskt (Andersen, 1994). Här kan man se en klar koppling till positivismen och den logiska positivismen som har sina utgångspunkter i logiken och fysiken där allt är mätbart (Andersen, 1994).

En undersökning kan göras med den ena eller den andra metoden, kvantitativt eller kvalitativt. Det finns flera avgörande faktorer som bestämmer vilken metodik som ska användas. Valet kan göras med följande faktorer (Andersen, 1994):

- Vilket ämne som ska undersökas.
- Hur undersökningsämnet uppfattas.
- Vilket syfte undersökningen har.

Med dessa tre punkter i åtanke ska denna undersökning genomföras med hjälp av de tekniker som är mest lämpliga för den problemställning som finns.

Innan en metod väljs bör man ha i åtanke vilken form av undersökning som ska göras. Det existerar flera olika inriktningar på hur ett arbete kan planeras. En undersökning kan vara explorativ, deskriptiv eller hypotesprövande (Patel & Davidsson, 1994). En explorativ undersökning försöker att undersöka ett problemområde så noggrant som möjligt och med denna information försöka svara på så många frågor som möjligt inom problemområdet (Patel & Davidsson, 1994). En deskriptiv undersökning försöker att beskriva förhållande som finns i nutid eller har funnits i dåtid (Patel & Davidsson, 1994). Om undersökningen har som mål att bli omfattande och man utvecklar egna teorier kallas den hypotesprövande (Patel & Davidsson, 1994). Detta innebär att det finns flera typer av undersökningsmetoder att välja emellan.

- Litteraturstudie
- Fallstudie
- Intervju

### **4.1 Litteraturstudie**

En litteraturstudie är en sökning av information i olika typer av dokument (Patel & Davidsson, 1994). Informationen som söks kan hittas i flera olika former, till exempel litteratur, statistik och register, audiovisuellt material eller material från institutioner (Andersen, 1994). Litteraturen kan vara till exempel skönlitteratur eller facklitteratur, statistik, register eller audiovisuellt material (Patel & Davidsson, 1994). Register finns i form av till exempel kundregister eller medlemsförteckningar, audiovisuellt material kan vara till exempel band, film eller material från institutioner finns i form av till exempel diarium eller protokoll (Patel & Davidsson, 1994).

Patel och Davidsson (1994) betonar att alla dokument inte är offentliga och detta gör att problemställningen kanske måste revideras för att kunna besvara de frågeställningar som har ställts i sin problemställning. Fördelarna med en litteraturstudie är att det är en snabb och billig teknik att besvara frågeställningar (Andersen, 1994). En svaghet som finns hos litteraturstudier är att materialet som används är styrt, det vill säga materialet är redan fastlagt (Andersen, 1994).

### **4.2 Fallstudie**

En fallstudie är en undersökning av ett speciellt fenomen i sin realistiska miljö (Backman, 1998). Studien kan utföras på en eller flera individer där en individ kan vara en organisation eller en situation (Patel & Davidsson, 1994). Fallstudien är ett bra medel för att undersöka processer eller förändringar (Patel & Davidsson, 1994). I fallstudien ska det dessutom undersökas om det finns några generaliseringar som kan göras av resultatet och det beror på vilken typ av fall som ligger till grund för studien (Patel & Davidsson, 1994). En fallstudie kan vara av olika typer, deskriptiv eller explorativ (Backman, 1998). En deskriptiv fallstudie är beskrivande, den beskriver ett fenomen som har uppkommit, och en explorativ studie är undersökande, den undersöker om fenomenet som uppkommit är sant (Backman, 1998).

### **4.3 Intervju**

Denna typ av undersökning kan beskrivas som ett samtal där intervjuaren skaffar sig information från respondenten (Andersen, 1994). En intervju kan genomföras på flera olika sätt men de två vanligaste är att en person utför intervjun vid ett personligt möte eller genom telefonsamtal (Patel & Davidsson, 1994). En intervjuare måste klargöra syftet med intervjun och få respondenten att känna att hans/hennes roll är viktig (Patel & Davidsson, 1994). En sak som är viktig att förklara är huruvida respondentens bidrag är anonymt, konfidentiellt eller om undersökningen blir en offentlig handling där dennas uppgifter och svar finns med (Patel & Davidsson, 1994). Skillnaden mellan anonymt och konfidentiellt är att på en anonym intervju finns det ingen möjlighet att spåra vem som har svarat vad, men i en konfidentiell intervju finns det nummer och liknande för att hålla reda på att alla är intervjuade (Patel & Davidsson, 1994).

Det finns flera olika faktorer som en intervju kan byggas upp kring. Standardisering är en av dessa och den bedöms på om frågorna i intervjun är planerade eller ej (Patel & Davidsson, 1994). Intervjun kan ha en hög grad av standardisering vilket gör att alla frågor är planerade i förväg. Ett annat mått som används är hur strukturerad en intervju är (Patel & Davidsson, 1994). Struktureringen bedöms av hur frågorna är formulerade, det vill säga är frågorna öppna eller finns det bara fasta svarsalternativ (Patel & Davidsson, 1994).

En intervju utformas med en inledning, mittendel och en avslutning (Patel & Davidsson, 1994). I inledningen ställs frågor om individens bakgrund som det behövs information om och i avslutningen ställs det även frågor om det finns ytterligare kommentarer eller någon form av tillägg (Patel & Davidsson, 1994). Mellan de

ovanstående delarna av intervjun kommer de frågor som är relaterade till problemet (Patel & Davidsson, 1994). Frågorna i en intervju kan ordnas på olika sätt och de sätt som existerar är tratt-teknik eller omvänd tratt-teknik (Patel & Davidsson, 1994). Tratt-tekniken går ut på att först ställa de stora generella frågorna för att sedan gå över till de mer specifika och i den omvända tekniken gör man tvärtom (Patel & Davidsson, 1994). När det gäller frågorna i en intervju är det viktigt att ställa frågor som respondenten förstår och har möjligheter att besvara på ett bra sätt (Andersen, 1994). Det är också viktigt att tänka på att inte ha för långa frågor, ledande frågor, negationer eller dubbelfrågor (Patel & Davidsson, 1994).

### 4.4 Val av metod

I problemställningen står det att är avsikten att göra en jämförelse mellan traditionella och objektorienterade synsätt. Detta innebär att arbetet kommer att bli en verbal analys av det problem som är specificerat och vilket gör att arbetet får en kvalitativ inriktning.

Utifrån arbetets problemställning har uppsatsen utvecklats enligt ett positivistiskt tankesätt, det vill säga att utifrån de traditionella tekniker som presenteras i arbetet försöka hitta ett arbetssätt som använder sig av objektorienterade tekniker. En annan aspekt på varför detta arbete är positivistiskt inriktat är att inom detta område som ska undersökas finns det inte utrymme för fria tolkningar, utan de tekniker som presenteras har sin grund i datalogi som är uppbyggt från den formella logiken. Problemställningen indikerar att det är en hypotes som ska falsifieras, det vill säga: Finns det objektorienterade tekniker som kan ersätta de traditionella fullt ut?

I problemställningen står det att en jämförelse av traditionella och objektorienterade tekniker ska göras vilket innebär att arbetet får även en deskriptiv inriktning. Med hjälp av den deskriptiva inriktningen och hypotesen som existerar har arbetet valt att baseras på litteraturstudie, fallstudie och intervjuer.

- Första delen i uppsatsen består av en litteraturstudie för att klargöra begrepp och förhållanden mellan de olika förfaringssätt som existerar för att utveckla ett informationssystem. Studien syftar också till att ge en kunskapsbas så att en korrekt jämförelse kan göras.
- Fallstudie och intervju kommer att användas för att få en realisering av litteraturstudien. Fallstudien kommer att ske i samarbete med ett företag som arbetar professionellt med att utveckla informationssystem. Deras utvecklingsmodell kommer att ligga till grund för fallstudien och i samband med studien kommer även intervjuer göras med systemutvecklingsansvariga på företaget.

Denna kombination av metoder gör det möjligt för undersökningen bli genomförd på flera olika sätt, men arbetet blir deskriptivt där man undersöker dagsläget i problemområdet som har specificerats och hypotesprövande där det undersöks om problemställningen kan falsifieras.

Intervjun kommer att vara ostrukturerad och något standardiserad. Det kommer att vara öppna frågor som utgår ifrån företagets utvecklingsmodell. Frågorna kan på så sätt förberedas med modellen och litteraturstudien som utgångspunkt vilket gör att intervjuerna kommer att vara uppbyggda med basfrågor och följdfrågor. Basfrågorna kommer att bestå av frågor som grundar sig i litteraturstudien och företagets utvecklingsmodell vilket gör att de blir standardiserade. Följdfrågorna blir de som uppkommer vid diskussion runt basfrågorna och de blir ostandardiserade.

För att kunna strukturera uppsatsens genomförande kommer det att finnas flera olika arbetssteg som kommer att följas. Arbetsstegen ska hjälpa till att besvara problemställningen och stegen är följande:

- För att kunna göra en jämförelse av traditionell och objektorienterad systemutveckling är en litteraturstudie gjord. Studien bestod i att undersöka om det finns ersättning för de tekniker som används i en traditionell utveckling med objektorienterade metoder och tekniker.
- En fallstudie har använts för att stärka de slutsatser som dras i litteraturstudien. Studien är genomförd på ett företag som idag använder sig av traditionella tekniker men är intresserade att se vad som krävs för att kunna börja utveckla system objektorienterat från förändringsanalys till implementation.
- Intervjuer är kopplade till fallstudien för att ge förklaring till de notationer som används i företagets traditionella tekniker. Intervjun har även försökt att utreda vilka begrepp som används i företaget för att kunna koppla dessa till litteraturen. Intervjuernas respondenter är idag ansvariga för systemutvecklingen på företaget.

## 5 Genomförande

I detta avsnitt av uppsatsen är syftet att redogöra hur litteraturstudien är genomförd och vilket fenomen som fallstudien har undersökt. Det kommer också att redogöras för hur intervjuerna har strukturerats och vilka respondenter som har valts.

### 5.1 Litteraturstudie

För att kunna hitta litteratur har en omfattande litteratursökning gjorts. I sökningen har Internet och bibliotek använts för att kunna hitta den litteratur som har varit av intresse för arbetet. Sökningen av litteratur för traditionella tekniker har inneburit att mycket litteratur har letats upp för att genomgå och utvärderas. I utvärderingen har det varit värdefullt att ur den traditionella synvinkeln hitta litteratur som ger dels en överblick av traditionell systemutveckling, samt en djupare inblick i flera olika traditionella metoder och tekniker att utveckla system. Inom det objektorienterade synsättet var det svårare att hitta litteratur om ett gemensamt synsätt för systemutveckling. Skälet till detta kan vara att den standard för objektorienterad modellering som arbetet grundar sig på (UML) blev erkänd 1997. Detta medförde att det fanns endast ett fåtal böcker att välja emellan. En kort introduktion till den litteratur som har varit central för arbetet ges nedan.

**Andersen** (1994) *Systemutveckling – principer, metoder och tekniker* är en bok som uppfyller ovanstående krav på en systemutvecklingsbok med traditionellt synsätt. Den tar upp livscykelmodellen på ett grundläggande sätt samt att den förklarar de olika traditionella teknikerna. Andersen har tjugo års erfarenhet inom systemutveckling både från undervisning och från praktisk erfarenhet (Andersen, 1994).

**Avison och Fitzgerald** (1995) *Information Systems Development: Methodologies, Techniques and Tools* är en bok som kompletterar Andersen (1994) väldigt bra för att de traditionella arbetssätten förklaras med en annan infallsvinkel gentemot Andersen (1994). Boken förklarar på ett bra sätt de metoder och tekniker som är aktuella för arbetet. Båda författarna är professorer i informationssystem och undervisar vid universitet i England. Avison undervisar vid Southampton University och Fitzgerald undervisar vid Birkbeck College, University of London (Avison & Fitzgerald, 1995).

**Eriksson och Penker** (1998) *UML Toolkit* fungerar som en uppslagsbok för notationen i UML och den är en av de få böcker som finns i objektorienterad systemutveckling med UML som grund. Eriksson har lång praktisk erfarenhet av objektorienterad utveckling och har dessutom skrivit flera böcker i ämnet som till exempel ”Objektorienterad programutveckling i C++” och ”Programutveckling för Windows och Windows NT” (Eriksson & Penker, 1996). Penker har arbetat som projektledare och utvecklingschef i objektorientering samt har dessutom skrivit boken ”Praktikfall av objektorienterad systemutveckling” (Eriksson & Penker, 1996).

**Mathiassen et al.** (1998) *Objektorienterad analys och design* beskriver metodstegen för en objektorienterad systemutveckling med UML som grund för notationen.

Mathiassen är professor i datalogi vid Aalborgs universitet och är verksam som konsult i IT frågor (Mathiassen et al., 1998). Munk-Madsen är verksam som konsult och undervisar dessutom i systemutveckling (Mathiassen et al., 1998). Han har också skrivit boken "Strategisk projektledning" (Mathiassen et al., 1998). Nielsen är lektor i datalogi vid Aalborgs universitet och är bland annat medförfattare till en mängd vetenskapliga artiklar och böcker (Mathiassen et al., 1998). Stage är lektor i datalogi vid Aalborgs universitet och forskar om metoder för analys och design inom systemutveckling.

### **5.2 Fallstudie**

För att få en verklighetsanknytning till det som undersöktes i litteraturstudien var fallstudie ett naturligt val. Ett företag kontaktades för att se om det var möjligt att se hur deras systemutveckling bedrivs. Företaget utvecklar sina system genom att använda sig av traditionella tekniker. Fallstudien har företagets systemutvecklingsmodell som grund och det visade sig att flera av de traditionella tekniker som existerar används i deras modell.

### **5.3 Intervjuer**

För att intervjuerna skulle vara genomförbara krävdes först en noggrann genomgång av företagets utvecklingsmodell för att se en möjlig koppling till litteraturen. Intervjuerna var relativt enkla att genomföra i och med att företagets utvecklingsmodell och litteraturstudien var utgångspunkten. Frågorna formulerades på ett sådant sätt att de gav svar på de oklarheter som fanns i modellen och för att kunna göra det möjligt att se en koppling till litteraturstudien. Med detta som utgångspunkt blev intervjuerna en blandning av de olika sätten att strukturera en intervju. Intervjuerna påbörjades i ostrukturerat och standardiserat sätt för att övergå till en öppen diskussion. Valet av respondent för intervjuerna var ganska enkelt, för det fanns bara en person som var utvecklingsansvarig på det aktuella företaget. Med denna respondent gjordes sedan en intervju med de förutsättningar som har förklarats ovan.



## 6 Materialredovisning

I detta kapitel ges en förklaring till de steg i livscykelmodellen som har varit aktuella i detta arbete, förändringsanalys och systemering, och på vilket sätt de skulle kunna utföras traditionellt och objektorienterat.

### 6.1 Förändringsanalys

Förändringsanalysen genomförs med hjälp av att en diskussion förs inom det företag som hade tänkt sig att utveckla ett nytt system (Andersen, 1994). Beskrivningen för denna fas gäller för både traditionellt och objektorienterat tänkande, för att en diskussion inte kan föras på ett specifikt traditionellt eller objektorienterat sätt. Diskussionen ligger sedan till grund för om det verkligen behövs ett nytt system eller ej (Andersen, 1994). När slutet av denna fas har nåtts har man kommit till en beslutspunkt (Avison & Fitzgerald, 1995). Beslutet gäller om det är lönt att utveckla ett nytt system eller ej (Avison & Fitzgerald, 1995).

*Systemeringen* består av två olika faser, analys och utformning (Andersen, 1994). Analysen är den del av systemutvecklingen där det bestäms vilken typ av information som ska existera i systemet (Andersen, 1994). I utformningen försöker man att strukturera informationen från analysen så att det är tekniskt genomförbart att realisera den typen av modell som analysen har utvecklat (Andersen, 1994). I delkapitel 2.8.2 beskrivs dessa två faser närmare. Som beskrivits tidigare tar detta arbetet bara upp analysfasen, vilken i sin tur är uppdelad i två stycken delar, nämligen verksamhetsanalys och informationsanalys.

### 6.2 Verksamhetsanalys

Verksamhetsanalysen är till för att kartlägga en verksamhets aktuella situation genom olika modeller eller diagram (Eriksson & Penker, 1996). De modeller eller diagram som används ska analysera verksamheten och visa hur systemet ska göra det lättare för verksamheten att utföra sitt arbete (Andersen, 1994).

#### 6.2.1 Traditionellt arbetssätt

Inom det traditionella synsättet finns det flera möjliga alternativa tekniker att kartlägga en verksamhet (se delkapitel 2.8.2). En teknik som kan användas är ett diagram för att kartlägga processer (Eriksson & Penker, 1996). Detta sätt att tänka finns strukturerad i form av dataflödesdiagram (Avison & Fitzgerald, 1995). Här är huvudtanken att hitta en central process för att sedan bryta ned den i delprocesser (Avison & Fitzgerald, 1995). Dataflödesdiagram är en typisk top-down metod och den finns mer detaljerat beskriven under delkapitel 2.2.1.

## 6.2.2 Objektorienterat arbetssätt

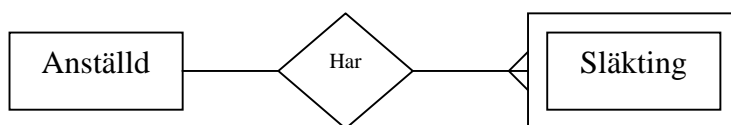
Objektorientering har ingen verksamhetsanalys implementerad i sin utvecklingsmodell (Eriksson & Penker, 1996). Den första fas som finns i ett objektorienterat synsätt kallas i och för sig analys men den avser bara den infologiska delen av analysfasen, det vill säga den del som analyserar vilken information som existerar i företaget (Eriksson & Penker, 1996). I denna analys modelleras det kommande informationssystemets funktionalitet upp och dess koppling till verksamheten i form av en objektorienterad modellering (Eriksson & Penker 1996).

## 6.3 Informationsanalys

Informationsanalysen används i första hand till att bestämma informationssystemets innehåll (Andersen, 1994). Detta görs genom beskriva systemet med hjälp av en infologisk modell (Sundgren, 1992). Modellen ska förklara begrepp som ska användas och dessutom på ett formellt sätt beskriva informationsbehovet i verksamheten (Sundgren, 1992).

### 6.3.1 Traditionellt arbetssätt

Inom det traditionella synsättet så är datamodellering ett sätt att beskriva det nya informationssystemet. Datamodelleringen har flera benämningar som infologisk datamodellering eller konceptuell modellering (Sundgren, 1992). En form av datamodellering är Entitets-Relationsmodellen (Andersen, 1994). Entitets-Relationsmodellen har som syfte att strukturera vilken information som systemet ska innehålla (Andersen, 1994). I Entitets-Relationsmodellen (ER-modellen) är de grundläggande begreppen entitet, attribut och relation (Andersen, 1994). En entitetstyp kan vara statisk eller dynamisk (Andersen, 1994). En statisk entitetstyp har hela tiden samma antal objekt men en dynamisk byter entiteter, det vill säga en instans av en entitet försvinner och en annan tillkommer (Andersen, 1994). En entitet kan också klassas som stark eller svag (Connolly et al., 1998). En svag entitet kan inte existera i den mening att den inte kan ha några instanser ifall det inte finns instanser i den starka entiteten (Connolly et al., 1998).

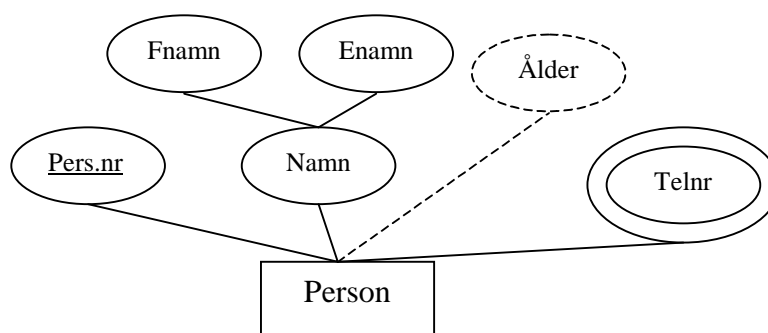


Figur 10: Exempel på svag och stark entitet

Figur 10 beskriver hur en anställd kan ha noll eller flera släktingar men en släkting kan inte existera om det inte finns någon anställd.

Attributen går att dela in i två grupper, identifierande och beskrivande (Andersen, 1994). Ett identifierande attribut används för att identifiera en entitet och ett beskrivande används för att ange de särdrag en entitet har (Andersen, 1994). Ett attribut kan också vara statiskt och dynamiskt, det vill säga ett statiskt attribut byter

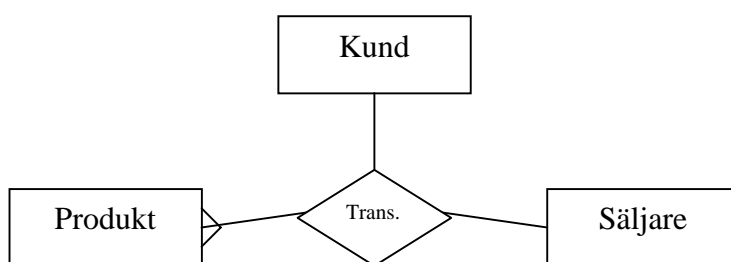
inte värde till exempel personnummer men i ett dynamiskt attribut sker förändringar av värdet till exempel datum (Andersen, 1994). Det finns flera typer av attribut, enkla, sammansatta, flervärda och härledda (Connolly et al., 1998). Ett enkelt attribut är ett annat ord för atomära attribut, det vill säga de kan inte delas ned i mindre delar (Connolly et al., 1998). Ett sammansatt attribut består av flera attribut (Connolly et al., 1998). Ett exempel kan vara adress som kan bestå av flera komponenter som till exempel gatuadress, stad, postadress men som är modellerad som ett sammansatt attribut, adress. Flervärda attribut kan innehålla ett antal olika värden inom ett visst intervall som till exempel 1-10 (Connolly et al., 1998). Ett härlett attribut kan räknas ut med hjälp av två andra attribut eller entiteter (Connolly et al., 1998). Ett exempel på detta är genomsnittsålder på anställda i ett företag där värdet räknas fram genom att addera de anställdas åldrar för att sedan dividera denna summa med antalet anställda.



Figur 11: Exempel på attribut

Figur 11 beskriver notationen för de olika attributen som är förklarade ovan. Ett enkelt attribut har en notation enligt personnummer. Personnummer är också det identifierande attributet vilket syns i form av understrykningen. Namn är ett sammansatt attribut som består av förnamn och efternamn. Ålder är det härledda attributet för i detta fallet kan ålder räknas ut genom att ta dagens datum minus de första sex siffrorna i personnumret. Telefonnummer är flervärdt för det kan finnas mer än ett telefonnummer som till exempel hemnummer och mobiltelefonnummer.

I ER-modellen är entiteterna kopplade till varandra genom relationer av typen associationer (Avison & Fitzgerald, 1995). Denna typen kan vara tvåställig eller flerställig (Avison & Fitzgerald, 1995). I en tvåställig relation associeras två entiteter till varandra och i en flerställig är det tre eller flera entiteter som kopplas ihop (Sundgren, 1992). I relationen finns det dessutom möjlighet att bestämma vilken kardinalitet den har (Avison & Fitzgerald, 1995). En närmare beskrivning av kardinalitet finns beskrivet i delkapitel 2.3.1.



Figur 12: Exempel på en treställig relation

I figur 12 finns en relation mellan kund, säljare och produkt. Relationen heter transaktion och involverar alla tre entiteterna. En kund och en försäljare har en transaktion över en eller flera produkter. Kardinaliteten är ett på kund och försäljare men den är en till många på produkt.

ER – modellen har ett tillvägagångssätt som liknar en typisk top-down metod (Avison & Fitzgerald, 1995). Detta innebär att man försöker hitta ett huvudflöde för att sedan bryta ned detta till delflöden. När man modellerar finns det tre steg att följa, det första är att man försöker att göra är att identifiera objekten, sedan försöka hitta relationerna mellan objekten och till sist hitta detaljerna i form av attribut och nyckelattribut, det vill säga man börjar beskriva övergripande för att fortsätta ned på detaljnivå (Avison & Fitzgerald, 1995). ER – modellering är dessutom en iterativ metod, det vill säga samma arbetsuppgifter utförs gång på gång tills alla detaljer är hittade och då är modellen klar (Andersen, 1994).

### 6.3.2 Objektorienterat synsätt

För att kunna kartlägga informationssystemets innehåll i ett objektorienterat perspektiv kan modellering användas. Denna modellering kommer från standarden UML vilket gör att notationen är accepterad av företag som till exempel IBM (Eriksson & Penker, 1998). I objektorienterad modellering försöker man följa två grundläggande principer (Mathiassen et al., 1998). Den första principen innebär att verkligheten ska beskrivas som användarna vill se den, och den andra principen är att prioritera överblick framför detaljer (Mathiassen et al., 1998). Med den första principen vill systemutvecklarna tillsammans med användarna skapa ett framtida system som kan tillfredsställa användarnas behov och se till att de kan utföra sina sysslor på ett effektivt sätt (Mathiassen et al., 1998). Den andra principen ska skapa en modell så att användarna ser hur systemet ska fungera, och dessutom ska de vara med och diskutera förslag som testas och värderas dessa kritiskt (Mathiassen et al., 1998).

Modelleringen är en iterativ process som hela tiden växlar mellan att modellera klasser, händelser, struktur och beteende för att hitta alla detaljer och få en komplett modell (Mathiassen et al., 1998). Det är två typer av modellering som är aktuella med det objektorienterade tankesättet, statisk och dynamisk modellering (Eriksson & Penker, 1998). Den statiska modellen består av klasser och relationerna mellan dessa (Eriksson & Penker, 1998). Den dynamiska modelleringen beskriver beteendet hos olika objekt och vilka tillstånd dessa kan ha (Eriksson & Penker, 1998). Detta arbete har avgränsats till den statiska modelleringen, vilken beskrivs närmare i nedanstående stycke.

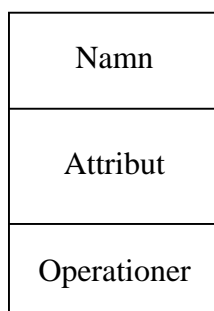
Den *statiska* modelleringen börjar med att strukturera upp klasser. En klass är en beskrivning av en objekttyp, vilket innebär att alla objekt är en instans av en klass (Eriksson & Penker, 1998). Den första fasen av modelleringen består av att strukturera och selektera objekt (Mathiassen et al., 1998). För att kunna välja klasser måste problemområdet undersökas och därigenom får man information om konkreta fenomen där fenomenen uppfattas som objekt (Mathiassen et al., 1998). Objekten

klassificeras sedan till klasser som datasystemet ska innehålla information om (Mathiassen et al., 1998). När klassificeringen är utförd är också avgränsningen av systemet klar (Mathiassen et al., 1998).

Under denna första fas i objektorienterad modellering står objektet i centrum (Mathiassen et al., 1998). Detta för att ett objekt utsätts för händelser och objekten ligger också till grund för de klasser som är grundstenarna i informationssystemet (Mathiassen et al., 1998). Ett objekt består av identitet, tillstånd och beteende (Mathiassen et al., 1998). Identiteten är vad som särskiljer ett objekt från ett annat, tillståndet är de egenskaper som beskriver varje objekt och beteendet är de händelser som objektet blir utsatt för (Mathiassen et al., 1998). För att underlätta hur man ska kunna hitta objekt för att sedan kunna klassificera dessa och göra klasser finns det vissa riktlinjer man kan använda sig av (Mathiassen et al., 1998):

- Leta efter substantiv i samtal med användare och i till exempel verksamhetsgraf.
- Leta efter generella typer av objekt som kan bilda en klass.
- Utnyttja erfarenheten eller titta på liknande system.
- Använda litteratur för att få uppslag på vilka klasser som kan användas.

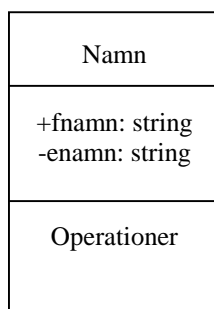
I denna fas är det sedan dags för att skapa ett klassdiagram (Mathiassen et al., 1998). Ett klassdiagram är en statisk modell över hur klasser och relationerna mellan dessa är strukturerade (Eriksson & Penker, 1998). Fördelen med att använda sig av ett klassdiagram är att klassen som finns representerad kan implementeras direkt i ett objektorienterat programspråk (Eriksson & Penker, 1998). I objektorientering finns det olika nivåer av synlighet för attributen (Eriksson & Penker, 1998). Privat betyder att en annan klass varken kan se eller använda sig av det speciella attributet (Eriksson & Penker, 1998). Offentlig har en motsatt funktion till privat och det betyder att attributet kan användas av objekt utanför sin egen klass (Eriksson & Penker, 1998). Notationen för en klass kan ses i figur 13.



Figur 13: Notation för klass

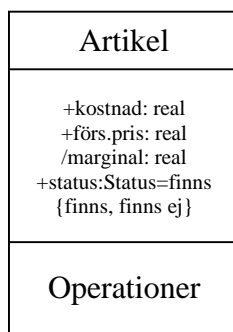
Figur 13 visar de tre fält som ska finnas med i notationen för en klass i ett klassdiagram (Eriksson & Penker, 1998). Den översta rektangeln ska innehålla namnet på klassen, den andra ska innehålla de attribut som är kopplade till objektet och den nedersta innehåller operationer som kan användas för att manipulera attribut

eller utföra andra aktiviteter som till exempel anropa en operation i en annan klass (Eriksson & Penker, 1998). Vid beskrivning av attributen kan olika nivåer av detalj användas (Eriksson & Penker, 1998). En beskrivning består av vad attributet heter, ifall det är privat (-), offentlig (+) och vilka värden de kan anta (Eriksson & Penker, 1998). Ett exempel på detta finns i figur 14.



Figur 14: Exempel på attributnotation

I figur 14 kan man se att Namn är objektets benämning och att attributet förnamn är offentligt samt efternamn är privat. Det visas också vilken typ av värden som är acceptabla i detta fält. En påbyggnad av detaljeringen kan göras med hjälp av härledning och begränsning (Eriksson & Penker, 1998). Ett härlett attribut är framtaget från två eller flera andra attribut (Eriksson & Penker, 1998). Med begränsningar bestämmer man vilken typ av värden det speciella attributet kan anta (Eriksson och Penker, 1998).



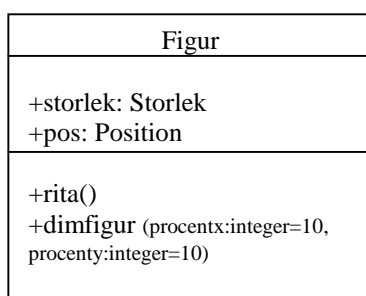
{marginal= försäljningspris - kostnad }

Figur 15: Exempel på attributdetaljer

Figur 15 visar om attributet är privat eller offentligt och dessutom vilken typ det ska implementeras som. Tillägget i detaljer syns på marginal som är ett härlett attribut från försäljningspris och kostnad. En mer detaljerad förklaring av attributet marginal ses under objektet. Attributet status har en begränsning och det noteras genom att de två godkända värdena står sist inom parentes.

Nästa del av notationen för en klass är operationsrektangeln. Operationer kallas funktioner som manipulerar attribut eller genomför andra typer av handlingar som till

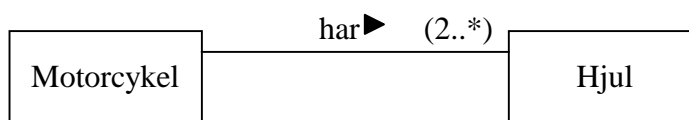
exempel anropar en annan operation (Eriksson & Penker, 1998). Ett vanligt sätt att implementera en operation är i form av en metod (Eriksson & Penker, 1998). En operation finns inuti en klass och kan bara användas av objekt som finns i klassen (Eriksson & Penker, 1998). En operation måste vara unik och den kan kännetecknas av returtypen, namnet eller sin parameterlista (Eriksson och Penker, 1998). Returtypen är vad funktionen returnerar och en parameterlista beskriver vilka parametrar som måste skickas med funktionen för att den ska exekvera och den kan bestå av noll eller flera parametrar (Eriksson & Penker, 1998). Operationen har samma möjlighet som attribut till synlighet i form av privat och offentlig (Eriksson & Penker, 1998).



Figur 16: Exempel på operationer

I figur 16 kan man se hur operationer har implementerats i klassen. Den första funktionen ritar figuren, har inga parametrar och är offentlig. Den andra är offentlig, har parametrar som används till att automatiskt dimensionera figuren. Ifall inga värden anges vid anropet kommer funktionen att automatiskt dimensionera figuren med 10 procent.

Det finns olika typer av relationer mellan klasser (Eriksson & Penker, 1998). Association är en relation som binder ihop två eller flera klasser och kan dessutom visa vilken multiplicitet som finns mellan de sidoordnade objekten (Eriksson & Penker, 1998). Multipliciteten visas med hjälp av (1), (2) eller (\*) (Mathiassen et al., 1998).

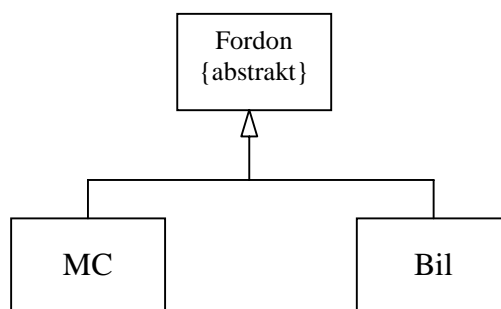


Figur 17: Exempel på en association

I figur 17 kan man se hur ett exempel på en association med multipliciteten två till flera. Detta realiserats genom att säga att en motorcykel har 2 eller flera hjul.

Generalisering gör det möjligt att grafiskt visa arv mellan objekt (Mathiassen et al., 1998). Rent språkligt skulle strukturen kunna beskrivas som "är-en" (Mathiassen et al., 1998). I notationen för UML finns också möjligheten att visa abstrakta klasser. En

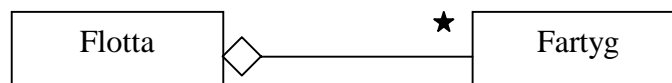
abstrakt klass kan inte ha instanser av några objekt (Eriksson & Penker, 1998). Denna klassen finns till för att bli ärvd från till andra konkreta klasser (Eriksson & Penker, 1998). Den grafiska notationen för denna struktur visas i figur 18.



Figur 18: Exempel på generalisering

Figur 18 kan beskrivas med att bil och MC båda är fordon. I och med denna struktur ärver bil och MC de attribut och operationer som finns i fordonsklassen. Fordon blir superklass samt MC och bil blir subklasser. I denna notation kan man även se att fordon är en abstrakt klass som inte kan ha några instanser av objekt.

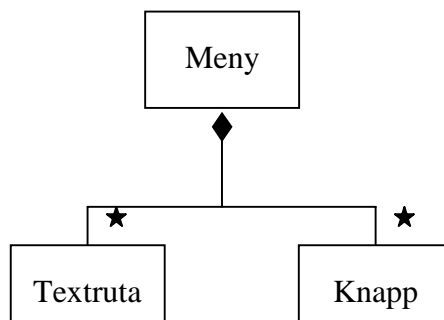
Nästa typ av relation heter aggregat och det är ett specialfall av association (Eriksson & Penker, 1998). Språkligt kan aggregat beskrivas med ”ingår-i” eller ”har-en” (Mathiassen et al., 1998). I figur 19 kan man se ett exempel på hur aggregatrelationen fungerar genom att en flotta består av ett eller flera fartyg och ett fartyg ingår bara i en flotta. Notationen visar med stjärnan att det kan existera flera fartyg i en flotta.



Figur 19: Exempel på aggregat

Ett specialfall på ovanstående relation är sammansatt aggregat (Eriksson & Penker, 1998). Relationen har fortfarande samma grundförutsättningar som ett vanligt aggregat men i ett sammansatt aggregat existerar delarna inuti det stora hela (Eriksson & Penker, 1998). Denna relation medför att starkt ägarskap som kan förklaras med att när det hela tas bort så försvinner även delarna (Eriksson & Penker, 1998). Denna typ av relation kan ses i figur 20 och det som skiljer från ett vanligt aggregat är att diamanten närmast det hela är fylld (Eriksson & Penker, 1998). I detta exempel kan man se hur en meny består av en eller flera textrutor och knappar. När menyn försvinner så tas även textrutor och knappar bort.





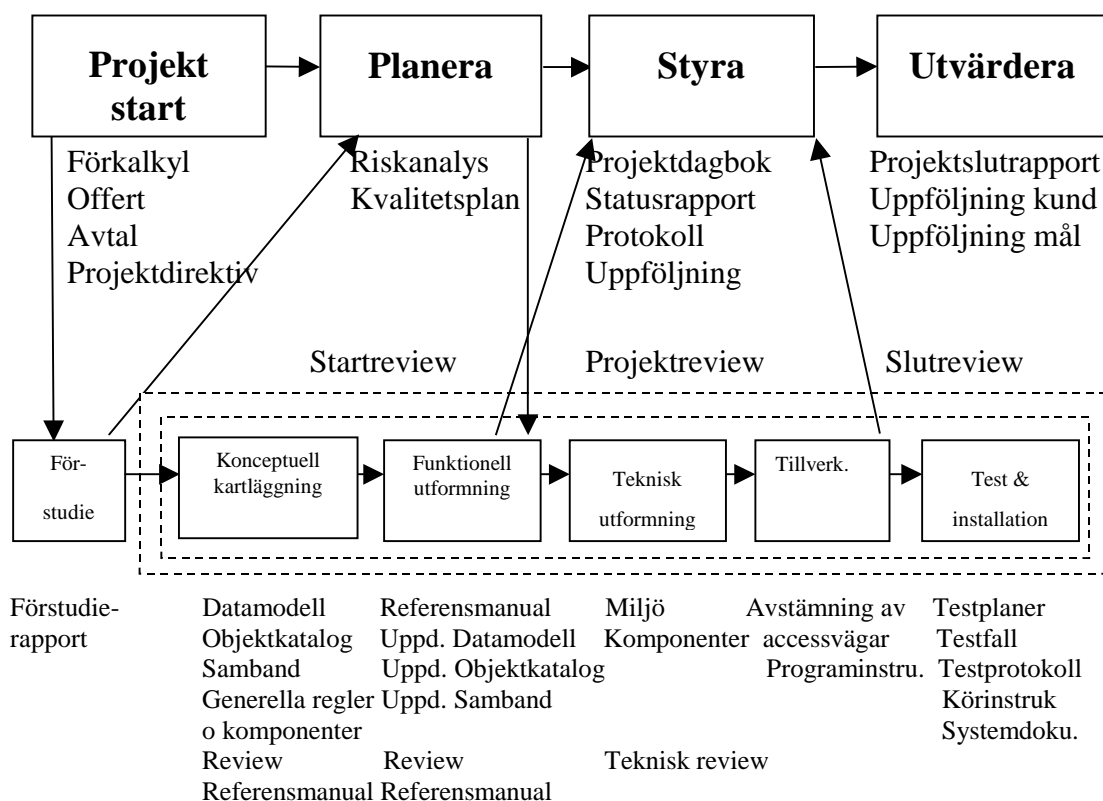
Figur 20: Exempel på sammansatt aggregat

En instans av ett klassdiagram heter objektdiagram (Eriksson & Penker, 1998). Denna instans kan modelleras på samma sätt som ett klassdiagram med attribut, operationer och relationer (Eriksson & Penker, 1998). Skillnaden mellan dessa två diagram finns i notationen för namnrektangeln. I rektangeln för namn blir skillnaden att i ett klassdiagram skrivs det **Bil** men i ett objektdiagram skrivs instansen **Ford : Bil** för att visa vilken klass instansen tillhör (Eriksson & Penker, 1998).

## 6.4 Fallstudie

Fallstudien är genomförd med ett företags modell som grund och fallstudien är en beskrivning på hur ett systemutvecklingsföretag kan arbeta. Företaget som fallstudien är baserad på heter IBS Konsult AB. Företagsidén är att utveckla och installera informationssystem (WWW-dokument: [www.ibs.se](http://www.ibs.se), 2000-05 04). IBS har etablerat sig i 30 länder och har totalt 2400 anställda och ungefär 5000 kunder (WWW-dokument: [www.ibs.se](http://www.ibs.se), 2000-05 04). Systemutvecklingen i företaget bedrivs enligt den modell som finns i figur 21. Grafen är hämtad ur IBS dokumentation för hur deras systemutvecklingsprojekt bör bedrivas. Överst beskriver grafen övergripande hur ett projekt bedrivs och vilka dokument som produceras för att ha kontroll i de olika momenten. De övergripande momenten beskrivs sedan mer detaljerat i den undre delen av grafen. I den undre delen beskrivs också vilka dokument som produceras för att stödja utvecklingsprocessen.

Arbetet har som tidigare beskrivits begränsat sig till att gå igenom de tre första faserna av denna modell för att en koppling ska kunna göras till problemställningen. Förstudie, konceptuell kartläggning och funktionell utformning är de tre faser som har varit aktuella. De aktuella faserna producerar flera olika dokument och de presenteras mer detaljerat i figurerna 23, 26 och 27 vilka alla är hämtade från IBS dokumentation för systemutveckling.



Figur 21: Utvecklingsmodell för IBS Konsult AB

### 6.4.1 Förstudie

Förstudien börjar med att kontrollera det material som kunden har angivit som underlag till det nya systemet, det vill säga en kravspecifikation som visar vad det nya informationssystemet bör innehålla (IBS Konsult AB, 2000). Denna kontroll sker i form av samtal med kund för att få en korrekt genomgång av underlaget (IBS Konsult AB, 2000). Sedan kartläggs vilka mål och behov kunden har med det nya systemet (IBS Konsult AB, 2000). Fortsättningsvis arbetas en verksamhetsbeskrivning och en rutiningraf fram för att beskriva dagsläget och önskade förändringar i kundens verksamhet (IBS Konsult AB, 2000). Verksamhetsbeskrivningen är en grov beskrivning av hur huvudprocesserna i företaget ser ut, för att alla involverade ska få samma verklighetsuppfattning. Rutiningrafen som IBS använder sig av är ett dataflödesdiagram (IBS Konsult AB, 2000).

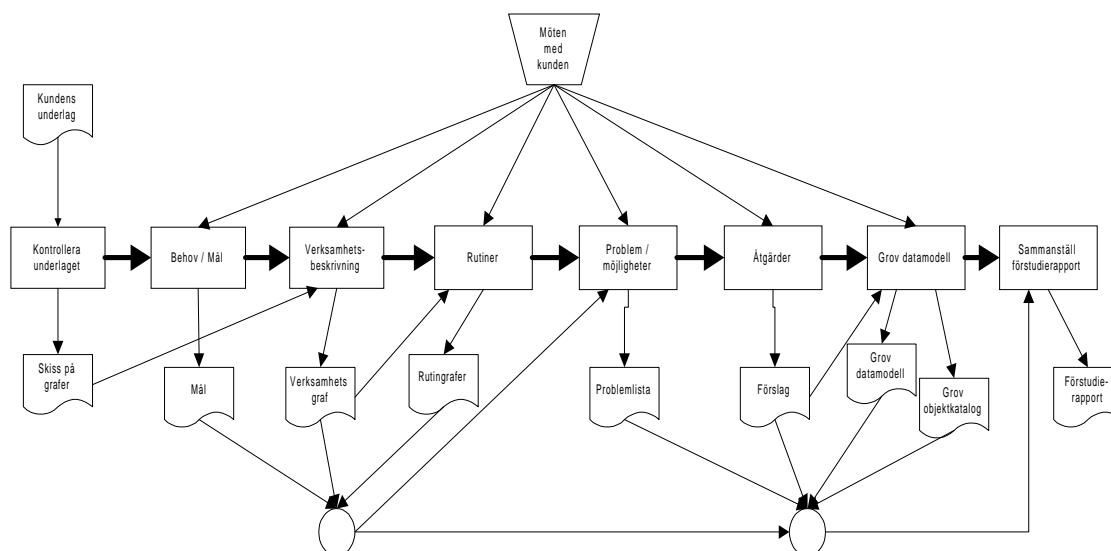
Detta resulterar sedan i en detaljerad beskrivning av vilka problem och möjligheter kunden har (IBS Konsult AB, 2000). Nästa steg är att gå igenom vilka åtgärder som kan existera för problemen (IBS Konsult AB, 2000). För att kunna hålla kontroll över de problem som har uppkommit struktureras de enligt figur 22.

Problem	Orsak	Åtgärd	Möjlighet
Långa handläggningstider på reklamationer	Reklamationsrutinen är för krånglig	Sätt in mer personal eller förenkla rutinen	Reklamationsrutinen snabbas upp genom att slopa vissa steg

Figur 22: Problem/möjligheter tabell

Figur 22 beskriver att företaget beskriver först problemet som existerar för att sedan undersöka vilka orsaker ligger som skapar problemet. Nästa steg är att fundera över vilka åtgärder som ska lösa problemet. I kolumnen med möjligheter tittar man på vilka konsekvenser som de olika åtgärderna får.

Ifall det konstateras att ett nytt system är lösningen konstrueras en grov datamodell (IBS Konsult AB, 2000). Den grova datamodellen IBS använder sig av är en traditionell teknik i form av ER-modellering (IBS Konsult AB, 2000). För att få fram de objekt som anses aktuella till datamodellering genomgås kundunderlaget, verksamhetsbeskrivningen och rutingrafen (IBS Konsult AB, 2000). Det IBS benämmer objekt har samma funktion som entitet i ER - modellering. Hela förstudien avslutas med en sammanställning av förstudien i form av en rapport som presenteras för den presumtiva kunden (IBS Konsult AB, 2000).



Figur 23: Graf över förstudien

Figur 23 beskriver grafiskt hur förstudien ska bedrivas. Från kunden fås ett underlag som kontrolleras. IBS går här igenom materialet som kunden har lämnat för att kontrollera att det verkligen stämmer. I detta steg produceras skisser över företaget. Nästa steg är att se vilka behov och mål kunden har, vilka tas fram i samtal med kund. Här produceras ett dokument som beskriver de mål som projektet ska uppnå. I nästa steg görs en verksamhetsbeskrivning i form av ett dataflödesdiagram. Diagrammet bör inte ha för mycket detaljer utan har som uppgift att beskriva företaget som

helhet. Diagrammet blir sedan underlag för rutingrafen som är ett mer detaljerat dataflödesdiagram. Mål, verksamhetsgraf och rutingraf ligger sedan till grund för en diskussion om problem och möjligheter hos kunden. Sedan kommer IBS överens med kunden om vilka åtgärder som ska göras. Efter att ha kommit överens om vilka åtgärder som ska göras, skapas en grov datamodell. De dokument som har producerats fram till detta steg är: Måldokument, verksamhetsgraf, rutingraf, problemlista, förslag på åtgärder, grov datamodell och grov objektkatalog. Dessa ligger sedan till grund för en förstudierapport vilken tillsammans med den grova datamodellen utgör grunden för de nästkommande faserna. Under hela arbetsgången i förstudien hålls ett nära samarbete med kunden.

#### 6.4.2 Konceptuell kartläggning

Enligt figur 21 är nästa steg i utvecklingen konceptuell kartläggning. Detta sker med att utveckla en datamodell (se figur 26) som kan ligga till grund för det nya systemet (IBS Konsult AB, 2000). Som grund till den mer detaljerade datamodellen ligger den grova datamodellen som skapades i förstudien (IBS Konsult AB, 2000). Den grova datamodellen och samtal med kunden gör att en ny datamodell utvecklas med kundens behov och önskemål i centrum (IBS Konsult AB, 2000). Tekniken IBS använder sig av för att modellera är en traditionell teknik att strukturera ett företag i form av ER - modellering (IBS Konsult AB, 2000). I IBS utveckling av datamodellen existerar normalisering av datamodellen (IBS Konsult AB, 2000). Detta innebär att man strukturerar den datamodell som har modellerats till en datamodell med ett korrekt och logiskt tänkande (IBS Konsult AB, 2000). Denna datamodell gör det sedan möjligt att implementera modellen som ett informationssystem utan att det existerar dubbellagring (IBS Konsult AB, 2000).

Enligt figur 21 ska en datamodell, objektkatalog, generella regler och komponenter samt samband dokumenteras. Datamodellen dokumenteras med hjälp av ER – modellering (IBS Konsult AB, 2000). Objektkatalogen är en beskrivning av varje objekt som innehåller definition av objekt, namn på objekt och vilken grupp objektet tillhör (IBS Konsult AB, 2000). En beskrivning av objektkatalogens notation kan ses i figur 24.

Rubrik	Förklaring
Alias	Flera benämningar på samma objekt
DDS namn	
Objektprefix	Objektprefix för objekttyper inom samma grupp
Objektklass	Den klass objektet tillhör, används bara vid OO.

Figur 24: Exempel på en objektkatalog

I detta moment skapas även en objekt-fält katalog för att beskriva de attribut som tillhör objektet (IBS Konsult AB, 2000). Objekt-fält katalogen innehåller fältnamn,

fälttyp, nyckel och en kort beskrivning (IBS Konsult AB, 2000). Ett exempel på denna kan ses i figur 25.

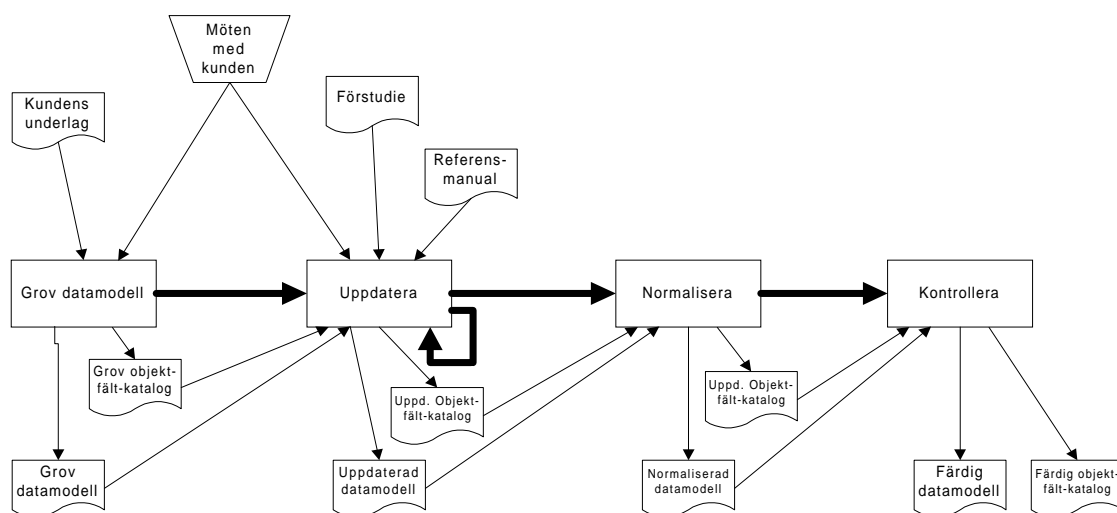
<b>Fältnamn</b>	<b>Fälttyp</b>	<b>Nyckel</b>	<b>Beskrivning</b>
Kundnr	6 N	K 1	Identifierar kund
Namn	40 A		För- och efternamn på kund

Figur 25: Exempel på en objekt-fält katalog

Denna fas av utvecklingen, det vill säga datamodellering, skapande av objektkatalog och objekt-fält katalog är en iterativ process som inte avslutas förrän kunden har tillfredsställt sina behov (IBS Konsult AB, 2000). Denna iterativa process kan ses i figur 26 i fältet uppdatera.

I figur 21 ska dessutom generaliseringar av regler och komponenter dokumenteras. Generaliseringar av regler är riktlinjer för hur systemutvecklingsprojektet ska genomdrivas och de tas fram med hjälp av samtal med kund och tidigare erfarenheter från andra systemutvecklingsprojekt. De komponenter som ska generaliseras är delar av systemet som man vet kommer att användas fler än en gång och genom generalisering försöker komponenternas utseende att utvecklas så att alla deltagare i systemutvecklingsprojektet är överens.

Enligt figur 21 skapas en referensmanual och en review görs. En referensmanual ska beskriva hur systemet används, det vill säga den är till för användarna som en vägledning för hur olika moment i systemet genomförs (IBS Konsult AB, 2000). En review är en återblick på hur arbetet har förflutit som ska visa om det har uppkommit problem eller om arbetet har fungerat som planerat (IBS Konsult AB, 2000).

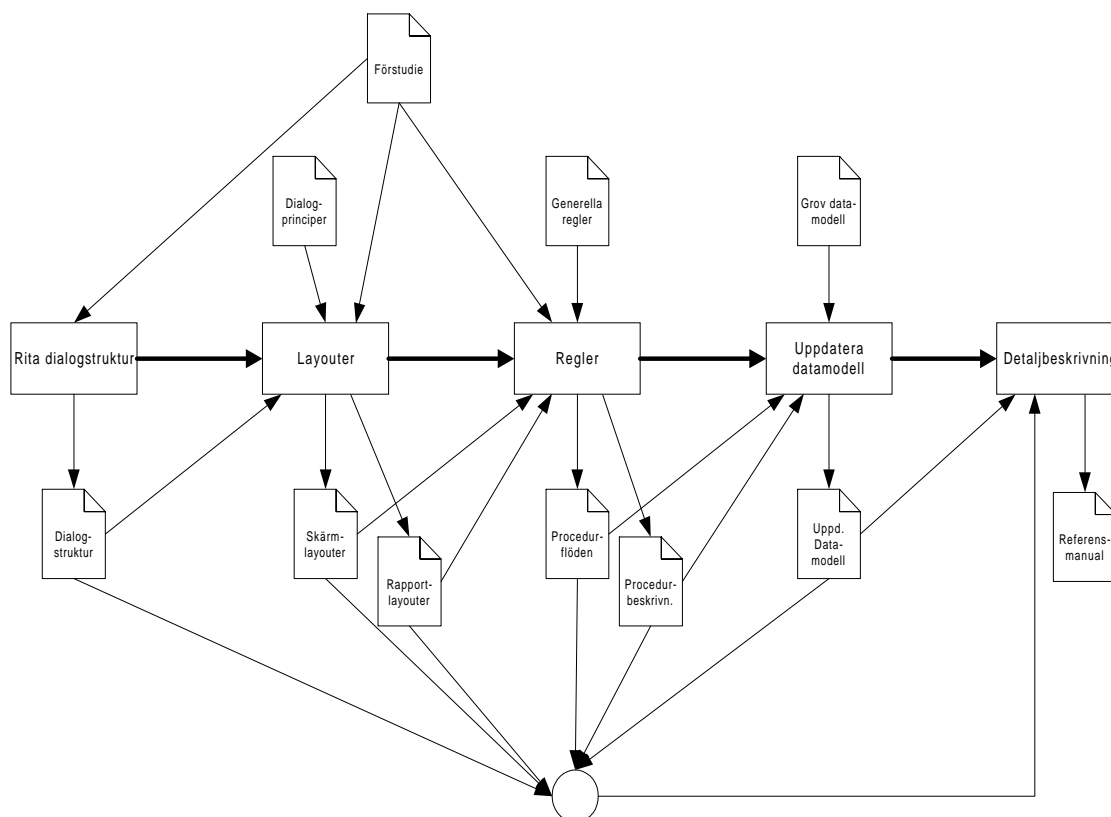


Figur 26: Graf över utvecklingen av datamodell

Figur 26 beskriver hur den konceptuella kartläggningen går till och vilka dokument som produceras. IBS börjar med den grova datamodellen som skapades i förstudien. Datamodellen går sedan igenom med kund för att bryta ned huvudflödet i delflöden och på så vis få en ökad detaljrikedom. Förstudierapporten används här som underlag för att säkerställa att det som togs upp i förstudien kommer med i modellen. När kunden tycker att datamodellen täcker behoven, normaliseras modellen för att all redundans ska försvinna. I det sista steget görs en kontroll att datamodellen fortfarande uppnår de krav som kunden ställer. För att beskriva attributen skapas en objekt-fältkatalog och den uppdateras när det sker en förändring av attributen i ER-modellen. Slutdokumenten är en korrekt datamodell och objekt-fältkatalog.

### 6.4.3 Funktionell utformning

Nästa steg enligt figur 21 är att arbeta med den funktionella utformningen. Denna fas har som mål att beskriva de dialogstrukturer och layouter som existerar i systemet (IBS Konsult AB, 2000). Materialet som ligger till grund för detta är rutiningrafen som utvecklades i förstudien (IBS Konsult AB, 2000). I denna fas utvecklas regler för hur en layout ska se ut som till exempel vart knapparna ska placeras i fönstret (IBS Konsult AB, 2000). Dialogprinciperna ska redogöra för hur kommunikationen mellan användare och databas ska struktureras som till exempel vad ska F1 som funktionsknapp kunna göra i alla fönster. Den funktionella utformningen är en mer detaljerad ER-modell, det vill säga den beskriver inte bara en konceptuell bild över företaget utan går in på detaljer som rapport- och skärmlayouter (IBS Konsult AB, 2000). I figur 27 beskrivs arbetsgången för en funktionell utformning. Enligt figur 21 skapas här inga nya dokument utan det sker bara en uppdatering av de existerande dokumenten från föregående fas. Efter denna fas görs också en review för att kontrollera att arbetet flyter som planerat.



Figur 27: Graf över den funktionella utformningen

Figur 27 beskriver den funktionella utformningen. Förstudien ligger till grund för denna fas som börjar med att bestämma vilka dialogstrukturer som ska gälla. När detta är bestämt ligger detta sedan till grund för hur layouterna ska se ut. När rapport- och skärmlayouter har bestämts sätts det upp regler för hur de ska kunna hanteras. I nästa steg uppdateras datamodellen. Dokumenten som skapas under denna process ligger sedan till grund för detaljbeskrivningen.

## 7 Analys

Detta kapitel analyserar det material som presenterades i det föregående kapitlet. Dessutom redovisar detta kapitel vilka tekniker som är användbara för att ersätta de traditionella teknikerna med ett objektorienterat synsätt och om ersättning kan göras i livscykelmodellens olika faser.

### 7.1 Förändringsanalys

Enligt livscykelmodellen sker här en diskussion hos kunden om vad som behöver förändras. Denna diskussion ska visa vilka problem och möjligheter som finns i dagens situation. En diskussion bedrivs inte på ett objektorienterat eller traditionellt sätt utan det är snarare en generell företeelse. I fallstudien kan man se att förstudien har en del av denna filosofi och den existerar som genomgång av det kundunderlag som har ställts till företagets förfogande. Den diskussion som förs har därför ingen inverkan på hur man sedan modellerar eller utvecklar ett nytt system och ger därför inga skillnader traditionellt eller objektorienterat. I denna fas av livscykelmodellen har det inte applicerats någon speciell traditionell teknik vilket gör att det inte finns någon att ersätta. Inom det objektorienterade synsättet finns det heller ingen speciell teknik för att hålla en diskussion och därför blir resultatet vad som är presenterat ovan.

### 7.2 Verksamhetsanalys

Den traditionella synvinkeln av verksamhetsanalys finns representerad av dataflödesdiagram i detta arbete. Denna typ av diagram visar de processer som är aktuella i företaget och på så sätt upptäcks problem eller möjligheter. I objektorienterad modellering med UML som grund existerar inga diagram eller tabeller för att kartlägga ett företag på ett liknande sätt. I fallstudien kan man se att företaget redan i förstudien gör både en verksamhetsbeskrivning och ett dataflödesdiagram (det som heter rutiningraf). I verksamhetsgrafan modelleras det stora hela upp med avseende på processer och hur olika delar av företaget är kopplat till varandra. I rutiningrafen går man sedan ned på detalj för att se hur de centrala rutinerna fungerar. De båda modelleringssätten utgår från ett funktionsorienterat tänkande. I denna fas av utvecklingen finns det ingen standardiserad ersättning för kartläggning av verksamheter i ett objektorienterat tänkande. Resultatet av detta är att man inte får ersättning för de traditionella synsätten och vill man ha en verksamhetsbeskrivning får man fortsätta arbeta på det traditionella sättet.

### 7.3 Informationsanalys

Ur traditionell synvinkel har detta arbete valt att genomföra informationsanalysen med hjälp av en ER-modell. ER-modellering har tre centrala begrepp, entitet, attribut och relation. Entiteterna är de objekt som finns i företaget och relationer kopplar ihop entiteterna. Relationer kan finnas i form av tvåställiga eller flerställiga. Attribut kan modelleras på olika sätt. Enkla, sammansatta, flervärda och härledda är de attribut som är aktuella i ER-modellering.

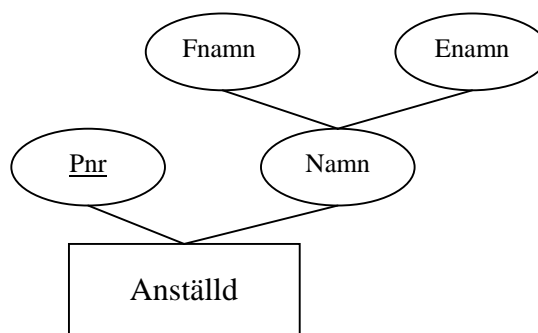


Statisk modellering görs i form av ett klassdiagram som beskriver en klass och hur dessa är kopplade till varandra ur ett statistiskt perspektiv. Modelleringen innefattar att det görs en generalisering av de objekt som existerar. Den statistiska delen har klasser, relationer, attribut och operationer som centrala begrepp. Klasser är en beskrivning av en samling objekt. Relationer visar vad de olika klasserna har för samband och de kan vara två- eller flerställiga. Ett attribut kan vara härlett eller det kan vara begränsat. Operationer visar att objektet har ett beteende och gör att objektet kan hamna i olika tillstånd.

Traditionella attribut	Objektorienterade attr
Enkelt	Enkelt
Härlett	Härlett
Flervärd	Begränsat
Sammansatt	-----

Figur 28: Jämförelse av attribut

I figur 28 jämförs de olika typer av attribut som existerar i de båda notationssätten. Traditionellt enkla och härledda attribut har en direkt motsvarighet i UML. Flervärda attribut har sin motsvarighet i begränsade attribut. Ett sammansatt attribut har ingen direkt motsvarighet i notationen för UML men i objektorienterad programmering finns möjligheten att lösa detta problem med hjälp av en struktur.



Figur 29: Exempel på sammansatt attribut

I figur 29 beskrivs ett sammansatt attribut enligt ER-modellen. Enligt figur 28 existerar det ingen notation för sammansatt attribut i UML, men denna typ av attribut kan erhållas med hjälp av objektorienterad programmering (se figur 30). Strukturen **Namn** deklarerar **fnamn** och **enamn** för att sedan slå ihop dessa till ett "sammansatt attribut". I den objektorienterade, grafiska, notationen kan **fnamn** och **enamn** deklareraras, men för att det sammansatta attributet också ska kunna dokumenteras grafiskt så krävs det ett tillägg till UML notationen.

```
typedef struct {
    char fnamn [];
    char enamn [];
}Namn;
```

Figur 30: Exempel på objektorienterad programkod

Relationer mellan objekten kan vara av flera olika typer, association, aggregat, generalisering. En objektorienterad association har samma funktionalitet som en association enligt traditionella synsättet. Sammansatt aggregat som beskriver vad ett objekt består av har sin motsvarighet i en svag entitet. En instans av den svaga entiteten kan inte existera ifall en instans av den starka entiteten inte finns. En utökning av relationsbegreppet finns med ett objektorienterat synsätt. Utökningen av relationer gör att modelleringen kan beskrivas något mer detaljerat jämfört med en ER-modell. Generalisering som realiserar arvsfunktionen och aggregat är något som inte existerar i en traditionell modellering. En jämförelse kan ses i figur 31.

Traditionella relationer	Objektorienterade relationer
Association	Association
Svag entitet	Sammansatt aggregat
-----	Aggregat
-----	Generalisering

Figur 31: Jämförelse av relationer

I fallstudien används ER-modellering och den kompletteras med objektkatalog och en objekt-fält katalog. Objektkatalogen beskriver objektet samt vilken grupp den tillhör. En objekt-fält katalog beskriver de attribut som är kopplade till objektet. Dessa kataloger skulle inte behövas i en objektorienterad modell där denna typ av information beskrivs i ett klassdiagram. I fallstudien kan man se att normalisering är en del av konceptuell kartläggning. Denna aktivitet innebär att man försöker få bort all redundans i datamodellen. När normaliseringen är fullbordad har man en datamodell utan redundans och dessutom är den klar för implementation. I det objektorienterade tänkandet finns inte begreppet normalisering. Anledningen till detta kan vara att möjligheten till beskrivningen av datamodellen är mer detaljerad och med denna möjlighet försvinner också nödvändigheten av normalisering.

För att få en överblick av analysen presenteras här de likheter och olikheter som har framkommit i litteraturstudie och fallstudie. De båda modelleringssätten arbetar iterativt, det vill säga man fortsätter att arbeta med modellerna tills alla detaljer är hittade. Den begreppsskillnad som finns är att det som heter entitet i ER-modellering kallas objekt i UML. ER-modellering kan ersättas av en statisk modellering förutom de sammansatta attributen. Den utökning som sker i ett objektorienterat tänkande

gentemot en traditionell modellering är operationer, två typer av relationer (aggregat och generalisering). Fallstudien har visat att företaget idag arbetar med dataflödesdiagram och ER-modellering. Det som beskrivs i fallstudiens förstudie har sin motsvarighet i den teoretiska analysfasen. Företagets arbetar med dataflödesdiagram och ER-modellering redan i förstudien för att sedan övergå till att arbeta med enbart ER-modellering i de nästa två faserna, konceptuell kartläggning och funktionell utformning. Detta för att få en mer detaljerad ER-modell som kan utformas till ett nytt informationssystem.

## 8 Resultat och slutsats

I detta arbete har förändringsanalys och första delen av systemeringen utvärderats. Denna utvärdering bestod i att ersätta traditionella tekniker med objektorienterat synsätt i de två första faserna av livscykelmodellen. I detta kapitel kommer det redovisas hur ett traditionellt projekt kan genomföras med de tekniker som har använts i arbetet. Kapitlet kommer också att gå igenom de möjligheter som finns att kunna ersätta de traditionella arbetsgängen med en objektorienterad.

### 8.1 Traditionell utveckling

I ett traditionellt tänkande finns det flera olika tekniker att välja emellan. I detta arbete presenteras endast ett par tekniker (dataflödesdiagram och ER-modellering) som kan vara användbara för en traditionell systemutveckling med funktioner som grund. Med traditionellt synsätt som grund har följande resultat uppnåtts, baserade på litteratur och fallstudien.

#### 8.1.1 Förändringsanalys

Detta är fas ett i en traditionell utveckling och i denna fas sker en diskussion hos kunden med avseende på att ta fram problem och möjligheter i den nuvarande organisationen.

#### 8.1.2 Analysfas i systemering

Ur ett traditionellt perspektiv kan analysfasen i ett systemutvecklingsprojekt bedrivas på flera olika sätt. Detta arbete har valt att koncentrera sig på dataflödesdiagram och ER-modellering vilket gör att analysfasen i ett projekt kan se ut enligt följande:

1. Utveckla en verksamhetsanalys. Verksamhetsbeskrivning består av att kartlägga det aktuella företags processer med hjälp av ett dataflödesdiagram.
2. Strukturera upp informationen som finns i företaget med hjälp av ER-modellering. Detta görs med dataflödesdiagram och samtal med kund som grund.
3. ER-modellen kompletteras med en objektkatalog för i närmare beskriva varje enskilt objekt.
4. För att få en mer detaljerad beskrivning av attributen görs en objekt-fält katalog.

Analysen sker enligt top-down principen. Börja med att strukturera hela företaget med en grov ER-modell utan för mycket detaljer för att fastlägga en huvudstruktur. Detaljnivån får bestämmas från projekt till projekt. Bryt sedan ned denna struktur i mindre delar med mer detaljer. Detsamma gäller för de olika katalogerna som först beskriver den grova ER-modellen för att sedan blir mer och mer detaljerade.

## 8.2 Objektorienterad utveckling

I detta avsnitt kommer det att presenteras ett arbetssätt som har sin härkomst ur Unified Modeling Language (UML). UML täcker dock inte livscykelmodellen helt vilket gör att en komplettering med hjälp av traditionella tekniker kommer att behövas. Detta kommer att presenteras i de arbetssteg som finns nedan för de olika faserna.

### 8.2.1 Förändringsanalys

I förändringsanalysen finns det inget speciellt sätt att föra en diskussion objektorienterat vilket gör att den blir likvärdig en traditionell förändringsanalys.

### 8.2.2 Analysfas av systemering

Objektorienterat synsätt har i denna fas endast en ersättning för ER-modellering. Dataflödesdiagram har ingen motsvarighet i UML och baserat på detta kan följande arbetssätt presenteras:

1. Utveckla en verksamhetsbeskrivning med hjälp av dataflödesdiagram. Detta är en traditionell teknik men det finns ingen objektorienterad ersättning i standarden UML.
2. Dataflödesdiagram och samtal med kund ligger sedan till grund för den statiska modelleringen. Den första fasen är att skapa ett klassdiagram.
  - A. Detta görs genom att undersöka dataflödesdiagrammet och samtala med kund om vilka objekt som är aktuella för ett nytt system. De objekt som anses aktuella blir klassificerade och klasser skapas.
  - B. Ur dessa klasser utvecklas sedan ett klassdiagram där det finns beskrivning av vilka relationer som finns mellan de olika klasserna.
  - C. Modelleringen fortsätter sedan ner på en större detaljnivå där attribut och operationer blir specificerade.

Dessa tre steg utförs sedan iterativt, det vill säga de utförs om och om igen tills alla detaljer är hittade.

## 8.3 Slutsatser

Frågeställningen i detta arbete var om UML kan ersätta de traditionella tekniker som existerar. Svaret på denna fråga är: NEJ.

- I förändringsanalysen finns det ingen skillnad.
- I verksamhetsanalysen existerar en traditionell teknik i form av dataflödesdiagram. Det objektorienterade synsättet har ingen ersättning för denna teknik.

- Informationsanalysen struktureras traditionellt sett av en ER-modell. I UML har det sin motsvarighet i den statiska modelleringen.
- ER-modellens relationer kan ersättas i UML till fullo men attributen kan inte ersättas fullt ut. De attribut som inte kan ersättas i notationen kan ersättas i objektorienterad programmering.

### 8.4 Förslag till framtida arbete

Att objektorienterad systemutveckling är intressant kan ha sin grund i att programmeringsspråken idag är för det mesta objektorienterade men företagen använder fortfarande traditionella systemutvecklingstekniker. En effekt som har en positiv inverkan på objektorienterad systemutveckling är att det har utvecklats en standard. En sak som skulle vara intressant att undersöka är om hur andra företag arbetar med sin systemutveckling. Har de kvar sina traditionella utvecklingstekniker eller använder de sig av en objektorienterad metod som till exempel UML?

En annan fortsättning på detta arbete skulle vara att ta reda på om vilken effekt en objektorienterad modellering skulle få på fortsättningen av livscykelmodellen. Med tanke på att företag många gånger använder sig av ett objektorienterat programmeringsspråk skulle det vara intressant att undersöka om det underlättar att till exempel implementera diagrammen eller skulle det inte bli någon skillnad alls när objektorienterad modellering används som underlag?

Ett begrepp som är använt i den traditionella systemutvecklingen är normalisering. Det är en metod för att få ER-modellen i en logisk struktur och redundansfri. I objektorienterad modellering är detta en metod som inte existerar. Ett fortsatt arbete kan vara att undersöka varför denna metod inte existerar i objektorienterad systemutveckling. När normalisering inte existerar så kan det vara intressant att se vad det får för konsekvenser, det vill säga blir det automatiskt ingen redundans i den objektorienterade modellen eller blir det dubbellagring av informationen som kanske förbises?

## 9 Diskussion

I detta kapitel kommer en diskussion att genomföras kring detta arbete. Resultaten som har framkommit och erfarenheter från arbetsprocessen kommer att stå i centrum. I denna diskussion kommer det även att tas upp vad som kunde ha gjorts bättre eller annorlunda.

### 9.1 Arbetsprocessen

I genomförandedelen beskrivs hur de olika metoderna har applicerats på arbetet. I litteraturstudien användes böcker som gav en generell bild av en traditionell systemutveckling. En aspekt på arbetet kunde ha varit att använda sig av en specifik modell och haft denna som utgångspunkt. Avsikten var dock att få en så generell bild som möjligt över en traditionell utveckling och därför valdes det att välja tekniker från flera olika modeller för att påvisa att det finns en mängd traditionella tekniker som har utvecklats. Om en enstaka modell hade använts hade den generella bilden av traditionell systemutveckling blivit väldigt smal och så är ju inte fallet.

Litteraturen som användes i den objektorienterade studien hade UML som grund. I Mathiassen et al. (1998) är boken inte bara grundad på UML utan de har kompletterat UML för att kunna täcka in både förändringsanalys och informationsanalys på ett objektorienterat sätt. Den andra boken som användes är helt grundad på UML och eftersom UML är så omfattande kunde inte allt i denna bok användas utan bara det som var intressant för detta arbete. Tillgång till litteratur inom detta område är begränsat men efter att utvärderat de böcker som existerade valdes dessa två som grund för arbetet.

Traditionell systemutveckling har funnits i årtionden vilket gör att det är ganska enkelt att hitta relevant litteratur för en litteraturstudie. Objektorienterad programmering har funnits sedan början på 1970-talet men idén om att utveckla en standard för systemutveckling med objektorienterat tankesätt utvecklades först i början på det föregående årtiondet. Detta innebär att det finns litteratur om många olika sätt att utveckla ett system objektorienterat men det har inte existerat ett genomgående tankesätt för alla inriktningar. 1997 kom det en standard för att ena tankesätten i objektorienterad systemutveckling i form av UML men tyvärr är denna standard inte tillräckligt omfattande för att kunna inkludera alla faser i livscykelmodellen. Ett av skälen till detta kan vara att standarden inte har existerat tillräckligt länge så att den inte har utvecklats till att täcka alla livscykelmodellens faser.

Företaget som blev undersökt i fallstudien arbetar idag med traditionella metoder. Idén var att få erfarenhet från hur ett systemutvecklingsföretag kan arbeta och att få en återkoppling till litteraturstudien. Fallstudien kunde ha strukturerats på ett sådant sätt att flera företag kunde ha undersökts och studien hade då blivit aktuell för fler företag. Om fler företag hade undersökts kunde det varit möjligt att dra slutsatser om vilka tekniker som används i olika systemutvecklingsföretag.

### 9.1.1 Val av metod

I litteraturstudien var syftet att få klart vilka begrepp som används samt att bilda sig en uppfattning hur de traditionella teknikerna skulle kunna ersättas. Detta var möjligt för att utgångspunkten för hela studien var livscykelmodellen som är en generell beskrivning på hur ett systems liv ser ut från förändringsanalys till avveckling. Med traditionella begrepp och modelleringsförfarande som grund kunde sedan en koppling lätt göras till objektorienterat synsätt.

I fallstudien är det fenomen som ska undersökas med en eller flera individer. Med detta i åtanke kontaktades IBS Konsult AB. Det är ett företag som utvecklar och implementerar system. Deras sätt att arbeta i de två första faserna av livscykelmodellen kan klassas som traditionellt. Detta gjorde att företaget blev intressant för en fallstudie då man hade möjlighet att undersöka deras sätt att arbeta och försöka ersätta det med objektorienterade tankegångar. Intervjun var en del av fallstudien i form av att respondenten är systemutvecklingsansvarig och frågorna hade sin grund i företagets utvecklingsmodell. Detta medförde att det blev mer en diskussion än en intervju.

Ett annat sätt att utföra detta arbete hade varit att fokusera på en fallstudie, det vill säga att utveckla en mer beskrivande introduktion och sedan involverat flera systemutvecklingsföretag. Detta kunde på så sätt presenterat fler traditionella utvecklingstekniker och givit en större möjlighet att generalisera sina resultat. Respondenten kunde inte ha varit någon bättre person, i och med att personen har mångårig erfarenhet från att utveckla system och var dessutom väl insatt i företagets egen modell.

## 9.2 Systemutveckling

Detta diskussionsavsnitt kommer att behandla de olika antaganden och avgränsningar som har gjorts i detta arbete med avseende på systemutveckling.

### 9.2.1 Utvecklingsmodeller

En modell som har haft en stor betydelse i det traditionella systemutvecklingsarbetet är livscykelmodellen. Den återkommer och nämns i flera böcker som har haft ett stort inflytande på hur traditionell systemutveckling har utvecklats. Genom att den har varit utgångspunkt för arbetet så finns det en naturlig återkoppling till det traditionella sättet att utveckla för att sedan försöka hitta de motsvarande sätten att utveckla objektorienterat. Livscykelmodellen har gjort det möjligt att utgå från en generell utgångspunkt som har varit likvärdig för både objektorienterad och traditionell systemutveckling.

I och med att arbetets mål var att ersätta traditionella tekniker till ett objektorienterat synsätt blev livscykelmodellen en utgångspunkt. Ett annat tillvägagångssätt hade varit att använda sig av en objektorienterad modell. Företaget som har utvecklat UML har också utvecklat en objektorienterad modell, Rational Unified Process (RUP) (Rational



Software Corporation, 2000). Med RUP som utgångspunkt kunde UML ha blivit heltäckande, det vill säga UML hade haft olika diagram för att täcka alla faser i RUP. Den traditionella livscykelmodellen ligger till grund för detta arbete för att huvudsyftet var att utgå från det traditionella perspektivet. Om arbetet hade haft syftet att utgå från ett objektorienterat synsätt hade det varit naturligt att använda en modell som till exempel RUP. Genom att de traditionella tekniker som presenterades i detta arbete har sitt ursprung i traditionell systemutveckling och inte i objektorienterad systemutveckling var det ett naturligt val att arbeta med den traditionella livscykelmodellen som utgångspunkt.

Företaget som låg till grund för fallstudien har erhållit en lösning för att kunna ersätta traditionella tekniker med objektorienterade. Detta kanske innebär att företagets utvecklingsmodell, vilken har sin grund i ett traditionellt tänkande, inte stödjer den nya processen fullt ut. Nästa steg kan då vara att se över sin utvecklingsmodell och utvärdera om det är lönt att fortsätta utveckla sin traditionella eller kanske anamma en objektorienterad modell som till exempel RUP. Det bästa för företaget kan vara att göra en stegvis övergång till objektorienterad systemutveckling, det vill säga att först försöka modellera i UML, för att sedan helt och hållet gå över till en objektorienterad systemutvecklingsmodell. Detta för att systemutvecklarna först skulle hinna arbeta med och lära sig de nya objektorienterade teknikerna för att sedan implementera en ny modell.

### 9.2.2 Traditionella tekniker

Ur den traditionella synvinkeln så har två av en mängd olika tekniker studerats. Uppfattningen är att dessa två tekniker används och har använts under en längre tid i systemutvecklingsprojekt med hänvisning till litteraturen. Dataflödesdiagram används för att kartlägga processer i ett företag och ER-modellering för att strukturera upp ett företags framtida databas. Teknikerna utgör en bra kombination för att genomföra ett systemutvecklingsprojekt vilket också står i litteraturen att funktions- och dataorienterade tekniker kompletterar varandra. I och med att dessa två tekniker har existerat under en längre tidsperiod finns det mycket litteratur och de finns beskrivna på ett mycket bra sätt.

Det har skett en utveckling av ER-modellering som heter "The Enhanced Entity-Relationship Model" (EER-modellen) (Connolly et al., 1998). Den har kvar de grundläggande koncepten från ER-modellering men har utvecklats mot ett objektorienterat tänkande (Connolly et al., 1998). I EER-modellen finns det bland annat generalisering som innebär att det är möjligt att modellera arv (Connolly et al., 1998). Detta kan vara tillräckligt för att kunna modellera en verksamhet på ett effektivare sätt än ER-modellering. En svaghet är dock att den dynamiska modellering inte existerar vilket gör att den inte kan ersätta objektorienterad modellering helt.

### 9.2.3 Objektorienterat synsätt

Det objektorienterade synsättet har flera metoder för att utveckla ett system. De olika metoderna skapades under början på nittioalet och har ingen förankring i någon speciell standard eller modell vilket har gjort att utvecklingen på detta område har gått

långsamt och en begreppsförvirring har existerat. 1997 utvecklades det dock en standard som gjorde att det finns vissa riktmärken att börja utveckla efter. UML är inte på något sätt en färdigutvecklad standard som går att applicera från första till sista fasen i den traditionella livscykelmodellen. Ett intressant perspektiv är om det blir samma utveckling av objektorienterade tekniker som det har varit med traditionella tekniker livscykelmodellen utvecklades. En skillnad är att UML inte är en modell men standarden har gjort att begreppsförvirring som har inte existerat längre. Det är en positiv effekt på detta arbete för ett vidare arbete kan använda sig av samma begrepp och de är fortfarande aktuella.

I UML existerar det en statisk och en dynamisk modellering. Den statiska har redan behandlats i tidigare avsnitt men den dynamiska som är en ganska stor utvidgning av modelleringen i sig och den presenteras i detta avsnitt. Den dynamiska modelleringen beskriver vilka tillstånd ett objekt kan försättas i eller hur olika objekt interagerar med varandra. I och med att den traditionella modelleringen bara är statisk har den ingen möjlighet att på ett bra sätt beskriva dynamiken i ett system.

Den *dynamiska* modelleringen görs genom att använda sig av ett tillståndsdigram, aktivitetsdiagram eller interaktionsdiagram (Mathiassen et al., 1998). Dessa typer av diagram beskriver beteendet hos objekten (Mathiassen et al., 1998). Tillståndsdigrammet beskriver de olika tillstånd ett objekt kan ha i en klass och aktivitetsdiagrammen beskriver hur aktiviteter utförs i ett objekts olika operationer (Mathiassen et al., 1998). Interaktionsdiagrammen beskriver hur olika objekt interagerar (Eriksson & Penker, 1998). Interaktionsdiagram är sekvensdiagram och samspelsdiagram vilka beskriver hur olika objekt interagerar (Eriksson & Penker, 1998). Ett sekvensdiagram visar hur objekt interagerar under en viss tidsrymd och samspelsdiagram hur olika objekt interagerar genom att visa hur meddelande blir skickade från ett objekt till ett annat (Eriksson & Penker, 1998). Oftast görs ett val mellan sekvensdiagram och samspelsdiagram beroende på det systemutvecklingsprojekt som är aktuellt (Eriksson & Penker, 1998). Nedan följer en närmare beskrivning av tillståndsdigram och sekvensdiagram.

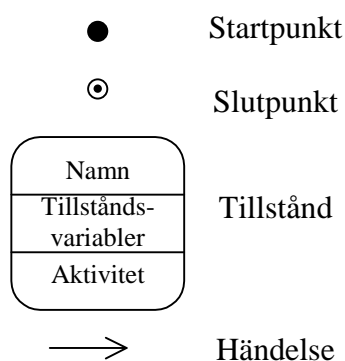
Tillståndsdigram har som uppgift att beskriva de olika tillstånd som ett objekt kan ha (Mathiassen et al., 1998). Tillstånden som objektet har kan förändras med hjälp av olika händelser (Eriksson & Penker, 1998). Händelser är tilldragelser som utnyttjar ett eller flera objekt (Mathiassen et al., 1998). En händelse kan beskrivas som en process eller en aktivitet i det problemområde som är specificerat (Mathiassen et al., 1998). En händelse representerar därför de dynamiska aspekterna i problemområdet (Mathiassen et al., 1998). För att på ett effektivt sätt kunna hitta händelser finns det vissa riktlinjer som man kan använda sig av (Mathiassen et al., 1998):

- Leta efter verb i samtal med användarna och till exempel i verksamhetsbeskrivningen.
- Leta efter generella typer av händelser.
- Använda sig av erfarenheten och titta på gamla system.
- Använda sig av litteratur för att få uppslag om vad för händelser som aktuella.

När tillräckligt många kandidater har hittats är det viktigt att värdera dessa och göra ett noggrant urval (Mathiassen et al., 1998). Det finns fyra typer av händelser i UML (Eriksson & Penker, 1998).

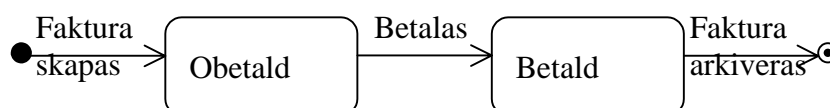
- Ett villkor blir sant.
- En explicit signal mottas från ett annat objekt.
- En signal från en operation i ett annat objekt eller en operation i det egna objektet.
- Att en bestämd tidsperiod har gått ut.

Händelser och objekt modelleras sedan i ett tillståndsdigram (Mathiassen et al., 1998). I figur 32 beskrivs de olika komponenternas notation för ett tillståndsdigram.



Figur 32: Notation för tillståndsdigram

De tre olika styrstrukturerna sekvens, selektion och iteration kan användas i ett tillståndsdigram (Mathiassen et al., 1998). I en sekvens inträffar händelser i den ordning som de är angivna och i en selektion händer precis en av de händelser som är angivna (Mathiassen et al., 1998). Iterationen gör det möjligt att en händelse kan inträffa noll till flera gånger (Mathiassen et al., 1998).



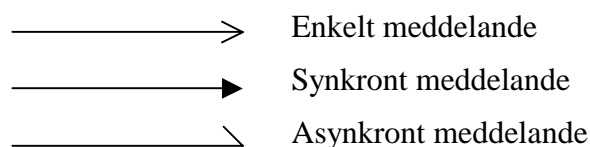
Figur 33: Exempel på tillståndsdigram

I figur 33 kan man se händelseförloppet för en faktura. Den skapas och då hamnar den i ett tillstånd som heter obetald för att sedan betalas och hamna i ett tillstånd som heter betald. Händelsen efter gör att fakturan arkiveras och då är slutpunkten i tillståndsdigrammet nådd. Detta tillståndsdigram är ett exempel på hur en sekvens kan beskrivas.

Ett sekvensdiagram ska beskriva hur ett antal objekt har en interaktion under en viss tidsperiod (Mathiassen, 1998). Fokuseringen i detta diagram ligger på hur meddelanden skickas mellan olika objekt och i vilken ordning (Eriksson & Penker, 1998). En viktig del i sekvensdiagram är hur meddelanden tolkas och hur ett meddelande blir skickat (Eriksson & Penker, 1998). Ett meddelande kan vara implementerad som en enkel operation när olika objekt anropar varandra (Eriksson & Penker, 1998). Ett meddelande skickas oftast i form av ett protokoll, det vill säga en struktur över hur ett meddelande kan skickas (Eriksson & Penker, 1998). Meddelandet kan vara i tre olika former:

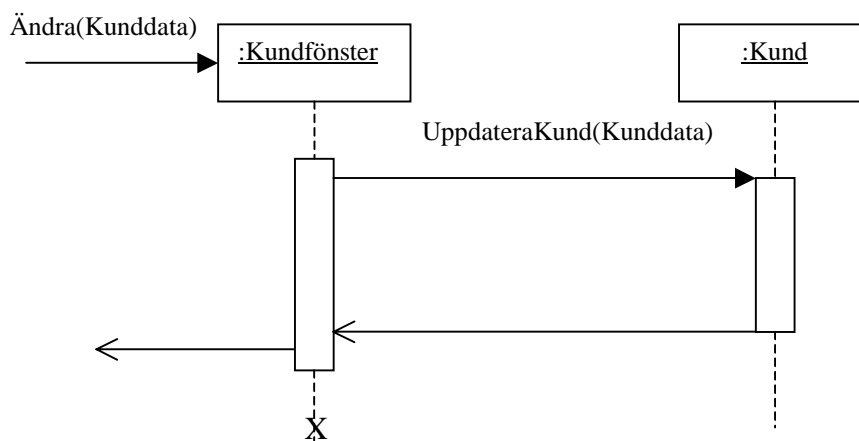
- Enkelt
- Synkront
- Asynkront

Ett enkelt meddelande visar hur kontrollen skiftar från ett objekt till ett annat vilket oftast är i form av ett returmeddelande (Eriksson & Penker, 1998). Ett synkront meddelande är ett anrop från en operation eller att en händelse har inträffat och på denna typ av meddelande kan man också ställa villkor (Mathiassen et al., 1998). Ett asynkront meddelande har som specialitet att kontrollen inte försvinner från det objekt som skickar sitt meddelande, utan det fortsätter att exekvera samtidigt som det objektet som mottagit meddelandet och ett exempel på detta är realtidssystem (Eriksson & Penker, 1998). Notationen för dessa tre typer av meddelanden finns beskriven i figur 34.



Figur 34: Exempel på meddelande

Ett sekvensdiagram börjar med att lista upp de objekt som är aktuella för sekvensen längst upp på sidan (Eriksson & Penker, 1998). Ett objekts livslinje visas genom en prickad linje (Eriksson & Penker, 1998). När objektet kundfönster sedan stängs avbryts livslinjen med ett X (Eriksson & Penker, 1998). I figur 35 kan man se hur detta kan realiseras.



Figur 35: Exempel på sekvensdiagram

Figur 35 beskriver ett förlopp för att ändra kunddata. Ett anrop kommer till kundfönster med Ändra(Kunddata). Ett anrop skickas till kund för att uppdatera kundregistret. När uppdateringen är klar skickas ett meddelande tillbaka för att klargöra att uppdateringen är klar. Sedan skickas ett meddelande till användaren att redigeringen är klar.

I och med att det finns en dynamisk modellering i UML kan det arbetssätt som presenterades i resultatet kompletteras och det fullständiga arbetssättet av analysen skulle då se enligt följande:

1. Utveckla en verksamhetsbeskrivning med hjälp av dataflödesdiagram. Detta är en traditionell teknik men det finns ingen objektorienterad ersättning i standarden UML.
2. Dataflödesdiagram och samtal med kund ligger sedan till grund för den statiska modelleringen. Den första fasen är att skapa ett klassdiagram.
  - A. Detta görs genom att undersöka dataflödesdiagrammet och samtala med kund om vilka objekt som är aktuella för ett nytt system. De objekt som anses aktuella blir klassificerade och klasser skapas.
  - B. Ur dessa klasser utvecklas sedan ett klassdiagram där det finns beskrivning av vilka relationer som finns mellan de olika klasserna.
  - C. Modelleringen fortsätter sedan ner på en större detaljnivå där attribut och operationer blir specificerade.

Dessa tre steg utförs sedan iterativt, det vill säga de utförs om och om igen tills alla detaljer är hittade.

3. Nästa fas av utveckling blir en dynamisk modellering. Den dynamiska modelleringen består av tillståndsdigram och interaktionsdiagram. För att kunna bestämma de olika klassernas tillstånd skapas ett tillståndsdigram.
  - A. Den första fasen i att skapa denna typ av diagram är att bestämma vad som ändrar en klass beteende. Detta görs med hjälp av dataflödesdiagrammet och kundsamtal.

- B. När händelserna är fastlagda struktureras vilka olika tillstånd som är aktuella för de enskilda klasserna.
- 4. Att modellera interaktionsdiagram är det sista steget är att strukturera upp det dynamiska. Ett sekvensdiagram visar hur objekten interagerar med varandra under en viss tidsrymd.
  - A. För att skapa ett sekvensdiagram selekteras först de objekt som är aktuella för den utvalda sekvensen.
  - B. Sedan görs en inventering för hur de olika objekten interagerar med hjälp av meddelande och detta förs sedan in i diagrammet.

## Referenser

- Andersen, E.S. (1994) *Systemutveckling – principer, metoder och tekniker* (2:a rev. uppl.). Lund: Studentlitteratur.
- Andersen, H. (1994) *Vetenskapsteori och metodlära – En introduktion*. Lund: Studentlitteratur.
- Avison, D.E. och Fitzgerald, G. (1995) *Information systems development: Methodologies, Techniques and Tools* (2<sup>nd</sup> ed.). UK: McGraw-Hill.
- Backman, J. (1998) *Rapporter och uppsatser*. Lund: Studentlitteratur.
- Booch, G. (1994) *Object-Oriented Analysis and Design with Applications* (2<sup>nd</sup> ed.). US: Addison-Wesley.
- Chen, P. (1976) *The Entity-Relationship Model: Toward a Unified View of Data*. ACM Transactions on Database Systems, Vol. 1, Nr. 1.
- Connolly, T., Begg, C., Strachan, A. (1998) *Database Design – A Practical Approach to Design, Implementation and Management*. UK: Addison Wesley Longman Limited.
- Daniels, J. och Yeates, D.A. (1971) *Basic Training in Systems Analysis*. (2<sup>nd</sup> ed.). UK: Pitman, London.
- Eriksson, H-E. och Penker, M. (1996) *Objektorientering – handbok och lexikon*. Lund: Studentlitteratur.
- Eriksson, H-E. och Penker, M. (1998) *UML Toolkit*. US: John Wiley & Sons, Inc.
- Fagerström, J. (1999) *Objektorienterad analys och design – en andra generationens metod* (2:a rev. uppl.). Lund: Studentlitteratur.
- Fagerström, J. (1993) *Objektorienterad systemutveckling – en introduktion*. Lund: Studentlitteratur.
- IBS AB (2000, Maj 5) *Företagsinformation – affärsidé*. [www dokument]. URL <http://www.ibs.se/sv/>
- IBS Konsult AB (2000) *Dokumentation för utveckling av system*. Göteborg: IBS Konsult AB.
- Karlander, L. (1998) Objekt databaser lyfter inte. *Datateknik*, 16, 34-36.
- Lewis, P.J. (1994) *Information-Systems Development*. UK: Pitman Publishing.
- Mathiassen, L. Munk-Madsen, A., Nielsen, P.A., Stage, J. (1998) *Objektorienterad analys och design*. Lund: Studentlitteratur.
- Patel, R. och Davidsson, B. (1994) *Forskningsmetodikens grunder: Att planera, genomföra och rapportera en undersökning* (2:a rev. uppl.). Lund: Studentlitteratur.
- Rational Software Corporation (1999) *Inside the UML*[CD-ROM]. US: Rational Software Corporation.
- Rational Software Corporation (2000, Mars 1) *Rational university*. [www dokument]. URL <http://www.rational.com/university/rubios.jtimpl>
- Rational Software Corporation (2000, Maj 1). [www dokument]. URL [http://www.rational.com/products/rup/prodinfo/whitepapers/dynamic.jtimpl?doc\\_key=100420](http://www.rational.com/products/rup/prodinfo/whitepapers/dynamic.jtimpl?doc_key=100420)

Sundgren, B. (1992) *Databasorienterad systemutveckling*. Lund: Studentlitteratur.



## Bilaga 1

Intervjun genomfördes hos IBS Konsult AB i Göteborg den 28 april, 2000. Inger Björner, som är systemutvecklingsansvarig på företaget, var respondent och författaren till denna rapport var intervjuare. Syftet med denna intervju var att klargöra de begrepp som IBS använder i sin systemutveckling och försöka få en koppling till litteraturen i litteraturstudien.

### Intervjufrågor med svar

Vad utgår ni ifrån i er förstudie?

- Vi försöker att titta på de problem och möjligheter som finns i verksamheten. I förstudien försöker vi också att få ut så mycket information som möjligt för att kunna realisera de krav som kunden har.

Har ni något specifikt arbetssätt som ni arbetar efter i er förstudie?

- Vi tittar på kundens verksamhet och försöker att hitta ett operativt flöde. Det operativa flödet är kundens affärsprocess, det vill säga vad tjänar kunden pengar på. Vi lokaliserar ett central funktion och utifrån denna bygger vi det nya systemet.

Hur dokumenterar ni detta?

- Vi dokumenterar detta grafiskt med text som stöd. Dokumentationen görs i form av en verksamhetsbeskrivning och en rutiningraf.

Vad har verksamhetsgrafens syfte?

- Verksamhetsgrafens syfte är att strukturera upp kundens huvudprocess och ge alla medverkande i systemutvecklingsprojektet en gemensam syn på företagets struktur.

Det ni benämner rutiner, har det samma betydelse som processer?

- Ja, vår rutiningraf har samma grundprinciper som ett dataflödesdiagram.

Hur arbetar ni när en ER-modell ska skapas?

- Först identifieras entiteterna för att sedan beskriva relationerna mellan dessa. Vi fortsätter sedan med att bestämma attribut och vilka attribut som är nycklar. Arbetet med ER-modellen är en iterativ process, det vill säga vi modellerar tills alla detaljer är hittade.

Vad menar ni med nycklar, är det detsamma som identifierande attribut?

- Ja.

Försöker ni att strukturera de problem och möjligheter som uppkommer?

- Ja. Vi strukturerar upp detta genom att försöka beskriva problemet, hitta orsaken till problemet för att kunna se vilka åtgärder som skulle kunna vara möjliga. Vi avslutar att se vilka möjligheter åtgärden ger företag och vilka konsekvenser den har.

Hur strukturerar ni er analys?

- Vi delar upp den i två delar, process- och datamodellering.

Vad betyder generalisering av regler och komponenter?

- Generalisering av regler menas att systemutvecklingsprojektet ska följa vissa riktlinjer för att det ska bli lättare att följa upp samt att systemutvecklare och företaget arbetar på samma villkor.

Vad menas med dialogprinciper?

- Dialogprinciper är hur vissa generella regler ska gälla i alla typer av layouter. Det kan vara till exempel att funktionsknappar har samma funktion eller att ett visst felmeddelande ser likadant ut i alla miljöer.