



GÖTEBORGS UNIVERSITET

# **Modellbaserad systemutveckling i komplexa verksamheter**

*en fallstudie av Coremetoden*

---

## **Model Based Systems Development in Complex Organizations**

*A case study of the Core Method*

**A. Gustavsson**  
**G. Hilford**

**Kandidatuppsats i Informatik**

**Rapportnummer: 2014:010**

**ISSN: 1651-4769**

# Abstrakt

Vid utveckling av system blir det idag allt mer vanligare att använda sig av modellbaserade systemutvecklingsmetoder. Dessa metoder som är anpassade för agilt arbete med enkla och snabba uppdateringar och implementeringar av nya applikationer i systemen följer de krav som blir viktigare idag, nämligen flexiblare och billigare system som fortfarande håller hög kvalitet och säkerhet. Ett företag som har jobbat med denna typ av systemutvecklingsmetoder under en längre tid är Genicore AB, som dessutom har tagit fram ett eget verktyg och modellspråk för att bättre kunna skapa modeller av informationshanteringssystem inom komplexa och abstrakta verksamheter. Genicore AB försöker idag sprida sin metod vidare till flera systemutvecklare och har därför tagit initiativ till detta arbete där metoden utvärderas och testas av två studenter med relativt lite erfarenhet av modellbaserad systemutveckling i syfte att få ett nytt perspektiv på hur deras metod fungerar att arbeta med. Deras metod har därför applicerats på utbildningsverksamheten för att testa den i en ny domän och detta ledde oss fram till frågeställningen;

*“Vilka kriterier är av särskild relevans vid implementation av en MBSD-metod på komplexa verksamheter?”*

Genom litteraturstudier och intervjuer skapade vi oss en uppfattning om hur modellbaserad systemutveckling fungerade rent teoretisk för att sen gå vidare till fallstudien som gjordes på utbildningsverksamhet på IT-fakulteten vid Göteborgs universitet. Med hjälp av Genicore ABs egna verktyg modellerade vi upp utbildningsverksamheten på IT-fakulteten ur ett fenomenologiskt perspektiv och utifrån det tog vi fram en rad relevanta kriterier för att kunna utvärdera metoden. De slutsatser som vi har kunnat dra är att det finns möjligheter för modellbaserad systemutveckling att utvecklas och tillämpas inom fler områden än vad det gör i dagsläget. Modellbaserad systemutveckling sett ur ett fenomenologiskt perspektiv är något som kräver en del förkunskap för att kunna tillämpas på ett korrekt vis men som genererar en mycket överskådlig och effektiv informationsmodell.

Rapporten är skriven på svenska.

**Nyckelord:** Verksamhetsanalys, modellbaserad systemutveckling, MBSE, MBSD, utbildningsverksamhet, Coremetoden

# Abstract

In system development it is today increasingly common to use a model-based system development method. These methods that are suited for Agile work with simple and quick updates and implementations of new applications in systems conform to the requirements that are becoming more important today, namely more flexible and less expensive system that still maintain high quality and safety. A company that has been working with this type of system development method for a long time is Genicore AB, which also has developed their own tools and model language in order to better model the information management systems in complex and abstract operations. Genicore AB is trying to spread their method other developers and have therefore taken the initiative to start this study. The method is evaluated and tested by two students with relatively little experience in model-based systems engineering in order to gain a new perspective on how the method functions. The method has been applied to an educational activity in order to test it in a new domain, and this led us to our research question;

*“What criteria are of particular relevance in the implementation of a MBSD method on on complex businesses?”*

Through literature reviews and interviews, we formed an idea of how model-based system development works from a theoretical perspective, and then continued to the case study conducted on the educational activities at the faculty of IT at University of Gothenburg. With the help of Genicore AB's own tools we modeled the educational operation from a phenomenological perspective and took out a list of relevant criteria for evaluating the method. The conclusions we have been able to draw are that there are opportunities for model-based systems to be developed and applied in more areas than in the current situation. Model-based systems from a phenomenological perspective are something that requires some prior knowledge to be applied in a proper way, but will generate a very orderly and efficient information model.

The report is written in Swedish.

**Keywords:** Business analysis, model based system development, MBSE, MBSD, educational activity, the Core method

# Tack

Vi vill rikta ett stort tack till vår handledare Alan B. Carlson för all handledning, goda råd och givande diskussioner. Även Faramarz Agahi ska ha ett stort tack för en givande intervju vid Institutionen för Informatik för en bra beskrivning av utbildningsverksamheten.

Vårt största tack går till Tomas Jonsson på Genicore AB som ständigt har funnits där för reda ut våra frågetecken samt föra diskussioner kring ämnet med. Han har även hållit i flertalet föreläsningar för oss där vi har fått ny kunskap inom ett område som vi inte tidigare har studerat.

# Innehållsförteckning

1. Introduktion .....	1
1.1 Bakgrund .....	1
1.2 Problemområde .....	2
1.3 Syfte och frågeställning .....	3
1.4 Avgränsningar .....	3
1.5 Undersökningens upplägg .....	4
2. Teori .....	5
2.1 Modeller och modellspråk .....	5
2.2 Modellbaserad systemutveckling .....	6
2.2.1 Fördelar med MBSD .....	8
2.2.2 Nackdelar med MBSD .....	8
2.3 Domänorienterad systemutveckling .....	9
2.4 Coremetoden .....	10
2.5 Andra modellbaserade systemutvecklingsmetoder .....	13
2.6 Kriterier för MBSD .....	14
3. Fallstudie; Coremetoden.....	17
3.1 Fallstudieobjekt - Coremetoden .....	17
3.2 Mjukvaruverktyget Visual-S och modellspråket "S" .....	17
3.3 Modellering av utbildningsverksamheten på IT-fakulteten.....	18
4. Metod .....	20
4.1 Fallstudie .....	20
4.2 Intervju .....	20
4.3 Dokumentanalys.....	21
4.3.1 Bedömningskriterier .....	21
5. Resultat.....	22
5.1 Modellering av utbildningsverksamheten .....	23
5.2 Bedömningskriterier samt applicering på Coremetoden .....	27

5.2.1 Verktyg .....	29
5.2.2 Metod.....	32
6. Diskussion .....	34
7. Slutsats .....	37
7.1 Studiens relevans och överförbarhet .....	37
7.2 Förslag till vidare forskning .....	38
8. Referenslista .....	39
9. Bilagor .....	41
Bilaga 1 Ordlista.....	41
Bilaga 2 Intervju.....	42
Bilaga 3 Modell av utbildningsverksamheten .....	43

# 1. Introduktion

I detta kapitel presenteras en bakgrund till det undersökta problemområdet som ligger till grund för vår frågeställning. Även avgränsningar som vi valt att göra samt undersökningens upplägg presenteras.

## 1.1 Bakgrund

För att ta reda på hur en verksamhet fungerar för tillfället så görs alltid en analys av verksamheten, en så kallad verksamhetsanalys. Detta görs för att kunna ge en bättre helhetsuppfattning om vilka problem som existerar (Goldkuhl och Röstlinger 1988). I dagsläget finns det mängder av modeller och metoder för att analysera organisationer men det råder skilda åsikter om vilka som är bäst anpassade för olika typer av verksamheter (Cadle et al. 2010). Vid införandet av informationssystem finns det dock två huvudsakliga metoder för systemutveckling som används; dokumentbaserad systemutveckling och modellbaserad systemutveckling. Den första baseras på kontinuerlig dokumentation över vad som ska göras i systemet. Den andra bygger på modeller över verksamheten, det är denna typ av utveckling som kommer granskas i uppsatsen (Amnell et al. u.å.).

Dagens system blir allt mer komplexa, och till följd av detta involveras fler människor i framtagandet vilket kräver en större förmåga att samarbeta. För att få samarbetet att fungera bör det finnas en gemensam vision för att undvika oförutsedda överraskningar. Genom att använda sig av modellbaserad systemutveckling (MBSD) underlättas samarbetet och antalet sent upptäckta fel minskar. Dessa hade annars kunnat leda till förseningar med ökade kostnader som konsekvens. Som metod tillhör MBSD nästa generation efter de objektorienterade programspråk som länge använts. Vid MBSD samlas all information om systemet i en eller flera grafiska modeller. Dessa modeller kan generera, utifrån satta regler och krav, ett helt eller delvis färdigt system. Denna typ av systemutveckling underlättar vid felsökning samt implementering av nya funktioner i systemet. Målet vid användande av MBSD är således att öka abstraktionsnivån i specifikationen samt öka automatiseringen vid den tekniska utvecklingen (Amnell et al. u.å.).

Det finns mycket forskning kring de flesta modeller och metoder som används i dagsläget, men det finns också metoder som det ännu inte finns så mycket forskning kring. Coremetoden är en av dessa metoder, framtagen av Genicore AB som är ett IT-konsultföretag med kontor i Göteborg. Coremetoden är en MBSD-metod med inriktning mot informationsmodelleringsteknik. Syftet med en sådan teknik är att med hjälp av symboler och deras strukturer förenkla och definiera information, samt organisera den så att den kan skapa nytta för användare. Då information idag finns överallt är det viktigt att ha bra verktyg för att ensa upp informationen och göra den tillgänglig för alla intressenter (Mylopoulos, 1998), ett krav som Coremetoden har haft som utgångspunkt. För att undersöka

metodens tillämpbarhet på en komplex verksamhet valde vi utbildningsverksamheten på IT-fakulteten på Göteborgs universitet.

Uppsatsens inriktning kom som ett önskemål från Tomas Jonsson på Genicore AB då vi sökte kandidatuppsatstupdrag från företag runt Göteborgstrakten. Vi hade en öppen frågeställning om verksamhetsanalys till företagen i början men Tomas kom med ett eget förslag inom verksamhetsanalys vilket vi tyckte lät intressant från första början; att analysera en komplex och abstrakt verksamhet. Tillsammans diskuterade vi fram ett syfte som båda parter var tillfreds med och vi beslöt oss även för att ha en kontinuerlig diskussion oss emellan. Det Genicore AB förväntar sig av oss är att tillämpa deras metod på ett universitets utbildningsverksamhet, vilket aldrig tidigare har gjorts. De har även intresse av att låta utomstående personer testa på att använda sig av deras metod. Ytterligare en aspekt som är intressant är att det är två studenter med väldigt lite erfarenhet av verksamhetsanalys och MBSD som ska använda sig av metoden, och detta kan visa på att metoden kan vara enkel att jobba med även för oerfarna användare.

## 1.2 Problemområde

Mycket av dagens forskning kring MBSD går under benämningen MBSE, som syftar till den teoretiska delen av modellbaserad systemutveckling. Det finns inte lika mycket forskning kring MBSD, vilket är den praktiska delen av modellbaserad systemutveckling vilket denna studie är mer inriktad på. Det finns än mindre litteratur om Genicore AB och den metod de utvecklat, Coremetoden. Något som har visat sig vara ett hinder för att sprida metoden till andra utvecklare och intressenter (Jonsson, personlig kommunikation, 14 maj, 2014).

Utbildningsverksamheten är en komplex verksamhet och mer komplex blir den om man ser till hela universitetsverksamheten. Ju mer en verksamhet växer desto komplexare blir den, dels internt men även externt med många samarbeten som måste tas hänsyn till<sup>1</sup>. Det finns ett allt större behov av att ta fram metoder som kan hantera komplexa verksamheter på grund av detta. Det krävs en djup förståelse vid utveckling av informationshanteringssystem för professionellt arbete och de problem som karaktäriseras av komplexiteten i olika professioner. Det är en tydlig trend att kunskapsintensiva verksamheter och deras intensitet i arbetet ökar, och därför blir också problemkomplexiteten viktigare att lägga fokus vid (Alvesson & Sköldberg 1993).

---

<sup>1</sup> Faramarz Agahi vice-prefekt för grundutbildningen på IT-fakulteten, intervju den 5 maj



## 1.3 Syfte och frågeställning

Utifrån den bakgrund och det problemområde som presenterats har vi härlett vårt syfte i denna uppsats, att undersöka om MBSD kan tillämpas för att göra en verksamhetsanalys på IT-fakultetens utbildningsverksamhet på Göteborgs universitet (GU). Studien genomförs som en fallstudie på Coremetoden och dess modelleringsprocess.

För att veta att förutsättningar och verktyg är rätt för att kunna använda sig av MBSD så har vi valt att ta fram kriterier för att bedöma om metoden anses vara gångbar eller inte. Dessa kriterier kommer därför att användas för att utvärdera hur väl Coremetoden fungerar för att skapa en modell av utbildningsverksamheten. Utifrån vårt syfte har vi tagit fram en frågeställning som lyder:

*“Vilka kriterier är av särskild relevans vid implementation av en MBSD-metod på komplexa verksamheter?”*

För att kunna besvara vår frågeställning ska vi applicera Coremetoden på IT-fakultetens utbildningsverksamhet på GU. Vi har lagt vårt fokus på att undersöka hur Coremetoden är uppbyggd samt att undersöka om den går att tillämpa på utbildningsverksamheten, och därmed utöka metodens tillämpningsområde. Fokus har parallellt riktats mot att ta fram kriterier för bedömning av en MBSD-metod.

## 1.4 Avgränsningar

Vi kommer endast att undersöka om Coremetoden är applicerbar på utbildningsverksamheten vid IT-fakulteten på GU och inte någon kringliggande fakultet.

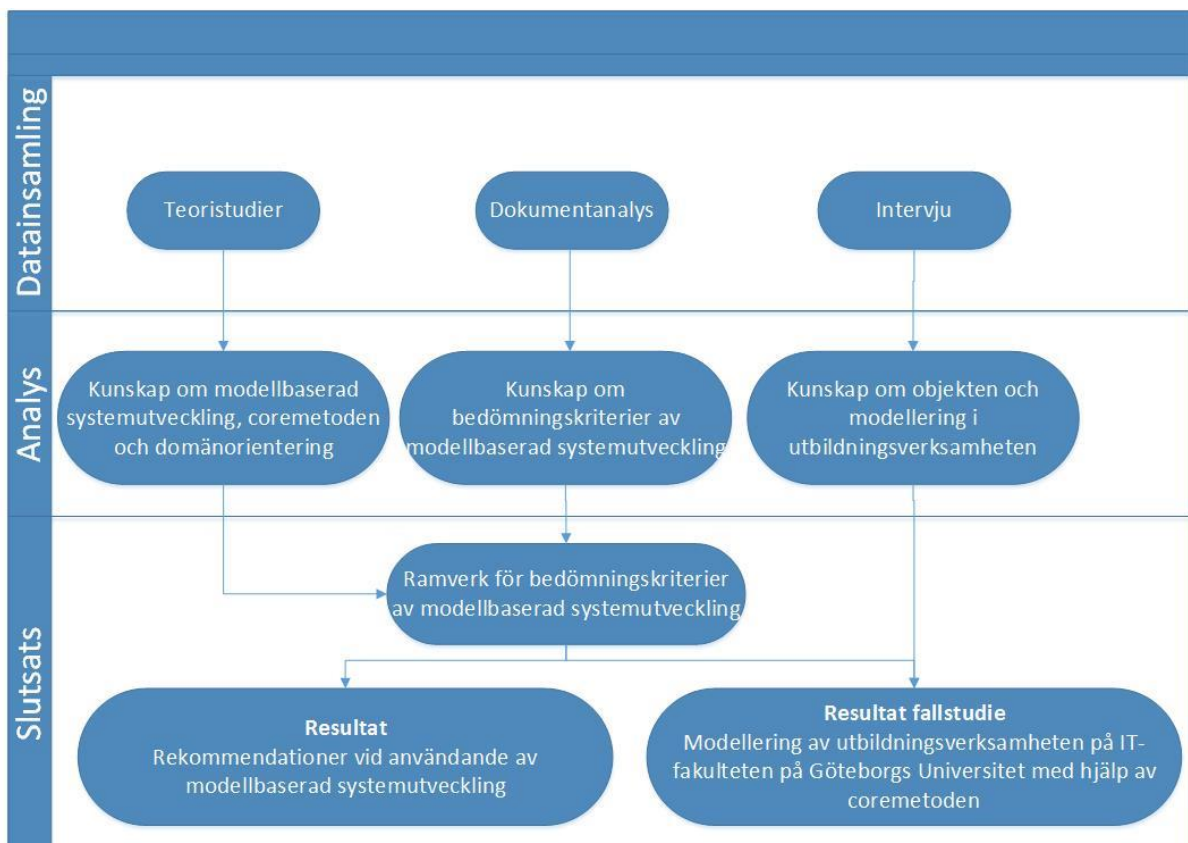
Till följd av begränsningar i tidsramen kommer vi inte att använda oss av alla funktioner som finns att tillgå i modelleringsverktyget. Vi kommer lägga fokus på det analytiska och teoretiska delarna och helt förbise de tekniska aspekterna. Det innebär att vi inte kommer definiera regler och förhållningssätt mellan objekten vilket omöjliggör en automatisk kodgenerering. Modellen kommer även sakna tekniska funktioner. Till följd av detta kommer vi inte ta fram kriterier och för dessa aspekter.

Vi kommer endast att använda oss av den version av Visual-S som är ett PC-program till skillnad från den webbaserade versionen som innehåller en annan lista av funktioner. Detta kommer innebära att vissa funktioner som kan användas i det webbaserade gränssnittet och som ännu inte har införts i den andra versionen inte kommer finnas till förfogande vid vårt arbete med Visual-S.

## 1.5 Undersökningens upplägg

Denna studies teori handlar övergripande om vad MBSE och MBSD är, för- och nackdelar, dess olika benämningar och slutsatser från tidigare forskning. Vidare behandlar den uppsatsens huvudsakliga metod, Coremetoden, och dess arbetsprocess samt tidigare studier. Även kriterier för bedömning av en MBSD-metod framställs i detta kapitel.

En intervju och personlig kommunikation har legat till grunden för en fallstudie där Coremetoden tillämpas på en utbildningsverksamhet. I kapitel tre har vi med hjälp av informationen från intervjun kunnat applicera metoden och modellerat upp den efter Coremetodens arbetsprocess. Till detta har vi tagit fram kriterier som vi jämför metoden med för att kunna utvärdera om den är tillämpningsbar på en universitetsverksamhet. Studiens metod finns i kapitel fyra där fallstudien, intervjun och dokumentanalysen presenteras. Resultatet presenteras i kapitel fem där vi har analyserat metoden och kriterierna, för att sedan diskuteras i kapitel sex. Avslutningsvis så kommer en slutsats i kapitel sju där hela studien sammanfattas och även rekommendationer för vidare forskning redogörs.



Figur 1. Översikt av undersökningen och dess upplägg

## 2. Teori

I teori visar vi på att det finns relevant och viktig kunskap inom området, och hur det kan hjälpa till att lösa vår frågeställning. Tanken är att denna litteraturgenomgång ska ge en djupare förståelse inom ämnet samt att det vidare ska fungera som en teoretisk referensram för kommande resultat och diskussion.

Vid utveckling av informationshanteringssystem så kan slutresultatet vara direkt kopplat till vilken informationsmodell som används menar Worboys och Duckham (2004) och därför ska de som är delaktiga i projektet alltid vara väl insatta i den informationsmodell som de jobbar med. Vidare anser de att det är informationsmodellen som anger hur data skall struktureras i organisationen samt hur de ska förhålla sig till varandra, och därför bör den vara tydligt beskriven och tillgänglig för alla i organisationen.

### 2.1 Modeller och modellspråk

Det finns många typer av modeller och de kan användas på många olika sätt i många olika sammanhang. Enligt Jonsson (personlig kommunikation 2014) kan några typer av modeller kan vara en skiss över en byggnad på papper, det kan även vara en 3D-modell av byggnaden i t.ex. trä eller en modell i en dator. Modellering kan även beskriva datorsystem som hårdvara eller mjukvara eller skapa en modell av verksamhetens olika delar och deras relationer. Inom programvaruvärlden används en del olika modellbegrepp. När det resoneras om modeller utan att specificera vilken typ av modell det är syftar det vanligtvis på en modell av programvara. Den kan sen bli mer specifik med datamodell, som är en modell över datan som används av programvaran. Det diskuteras även om processmodeller och domänmodeller. Processmodell är en modell av exekveringsflödet i en programvara och domänmodell är en modell av domänen som programvaran kommer jobba i, det vill säga verksamhetsobjekt som programvaran ska hantera data om.

Vid arbete i olika typer av modeller används olika typer av modelleringsspråk med olika karaktärer. Dessa kan vara process, procedurell, funktionell, deklarativ, objektorienterad eller grafisk med mera. Vid modellering inom byggnadsritningar används standarden ISO 128. Vid arbete med programvarumodellering används ofta unified modeling language (UML) som standardspråk, UML är en deklarativ objektbaserad grafisk notationssyntax. Många jobbar med modelleringsspråk som bygger på UML men som är mer anpassade för särskilda domäner, Genicore AB arbetar med ett modellspråk som de kallar "S" som kommer behandlas mer under rubrik 2.2. (Jonsson, personlig kommunikation, 14 maj, 2014).

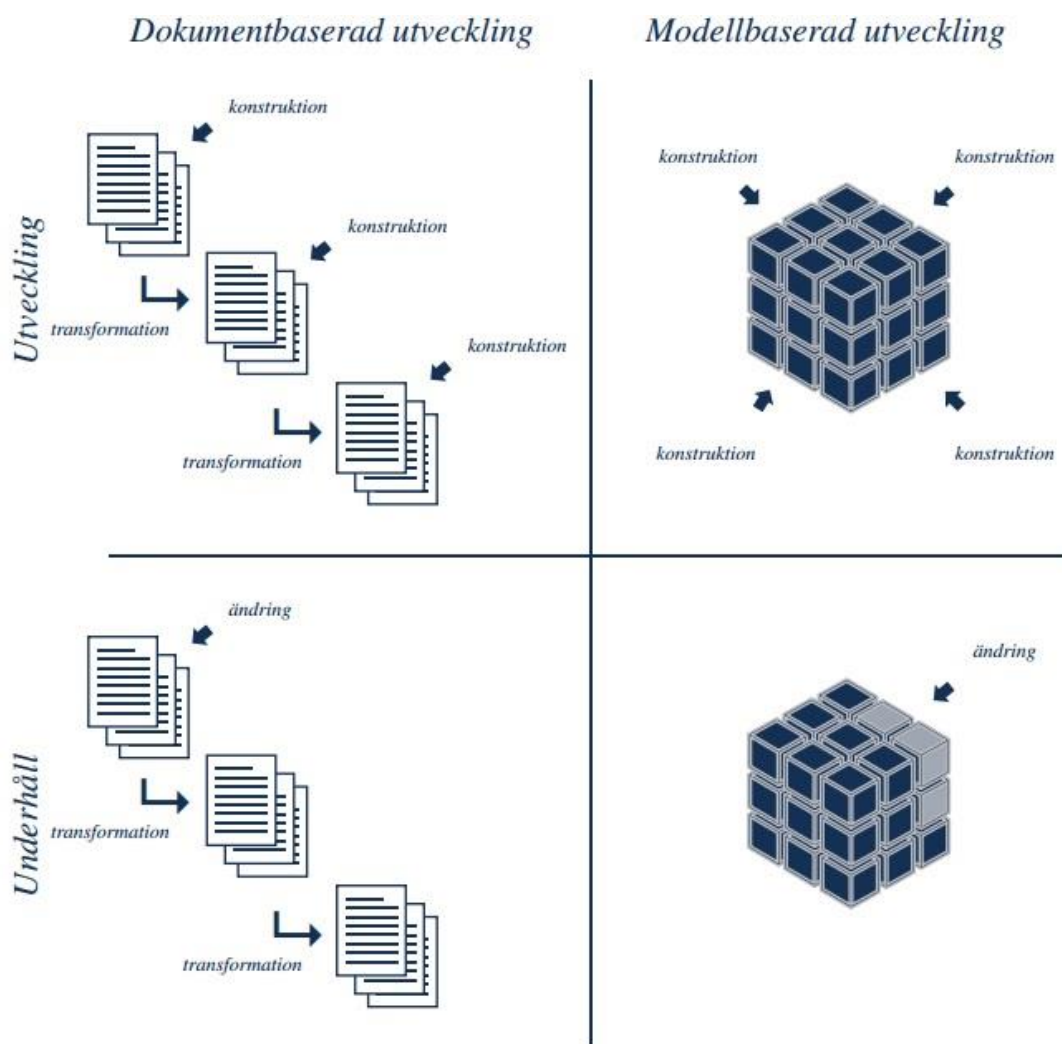
## 2.2 Modellbaserad systemutveckling

Modellbaserad systemutveckling benämns på en mängd olika sätt i många olika artiklar; MBSD, MBSE, MDD, MDE är några exempel. Det vanligast förekommande begreppet och de begrepp som vi valt att använda oss av Model Based System Development (MBSD) som är den praktiska aspekten.

Denna typ av systemutveckling bygger på att en modell av verksamheten skapas som innehåller all information för konstruktion av ett nytt mjukvarusystem. När en kravanalys har gjorts så används exempelvis UML för att beskriva en modell av systemet och de relationer som finns inom det.

Modellen blir en översiktsbild av verksamheten. I MBSD modelleras verksamheten så som den uppfattas och behöver inte följa förformaterade dokument för att visa hur verksamheten fungerar.

Detta innebär en mycket friare modelleringsprocess som även underlättar vid förändringar i systemet då samtlig information kring verksamheten samlas i modellen.



Figur 2.1 visar hur konstruktion av system skiljer sig mellan modellbaserad och dokumentbaserad utveckling samt hur ändringar i systemet sker på olika vis. (Amnell et al. u.å.)

Utifrån den komplexa modellen som tas fram genereras automatisk kod utifrån de satta reglerna mellan objekt och attribut som specificerats i modellen (Amnell et al. u.å.).

I modellen definieras även den dokumentation som behövs så att den automatiskt kan genereras vid behov. Denna dokumentation blir således implementerad i modellen och behöver ej sparas separat utan kan tas fram för ändring eller senare granskning från modellen när så behövs (Amnell et al. u.å.). Modellen fungerar därmed som referens, resurs samt felsökningsverktyg. Vid arbete med modellbaserade metoder har modellen därför ett ihållande värde genom hela projektets livscykel (Stoaks 2008).

### **2.2.1 Fördelar med MBSD**

Det finns fördelar med att arbeta med en MBSD-metod kontra andra metoder. Till exempel så uteblir redundans i informationsmängderna då all information kring ett objekt samlas på ett ställe i modellen. Att minska redundansen i arbetet minskar antalet felkällor samt sparar tid vid utveckling- och underhållsarbetet. Ett MBSD-verktyg hjälper även till att hålla definitioner i systemet konsekventa och utan motsägelser samt möjlighet att beskriva systemet på flera abstraktionsnivåer (Amnell et al. u.å.).

Då dagens systemutvecklingsverktyg kan göra direkt exekverbara modeller så kan systemen testas tidigt men också kontinuerligt under sin utveckling. Det innebär att systemets status kan övervakas kontinuerligt vilket minskar riskerna för sent upptäckta fel som kan orsaka problem. Genom att arbeta med en grafisk representation blir det lättare för individer som inte är så insatta i de tekniska delarna att förstå och bidra med sin syn av modellen, vilket underlättar för komplexiteten. Den grafiska representationen gör det lättare att förutsäga hur förändringar i systemet kommer att påverka helheten, vilket i sin tur resulterar i att omkonstruktioner blir enklare och effektivare att göra. Då all information är samlad i ett verktyg är det lätt att skapa spårbarhet mellan krav, design, integration och test. I slutändan är modellen systemet vilket innebär att ändringar i modellen ändrar realiseringen (Amnell et al. u.å.).

De flesta dokumentorienterade metoderna är anpassade och fungerar bäst när det inte sker några ändringar under utvecklingens gång, Brown (2011) menar dock att förändring är både oundviklig och oftast önskvärd under arbetets gång då det finns möjlighet att lära sig mer om miljön, detta leder till att fundamentala delar av systemet kommer att vilja ändras. Traditionella processer undviker helst förändringar då dessa leder till störningar i arbetet samt omfattande omarbetning av annars färdigt material (Brown 2011).

### **2.2.2 Nackdelar med MBSD**

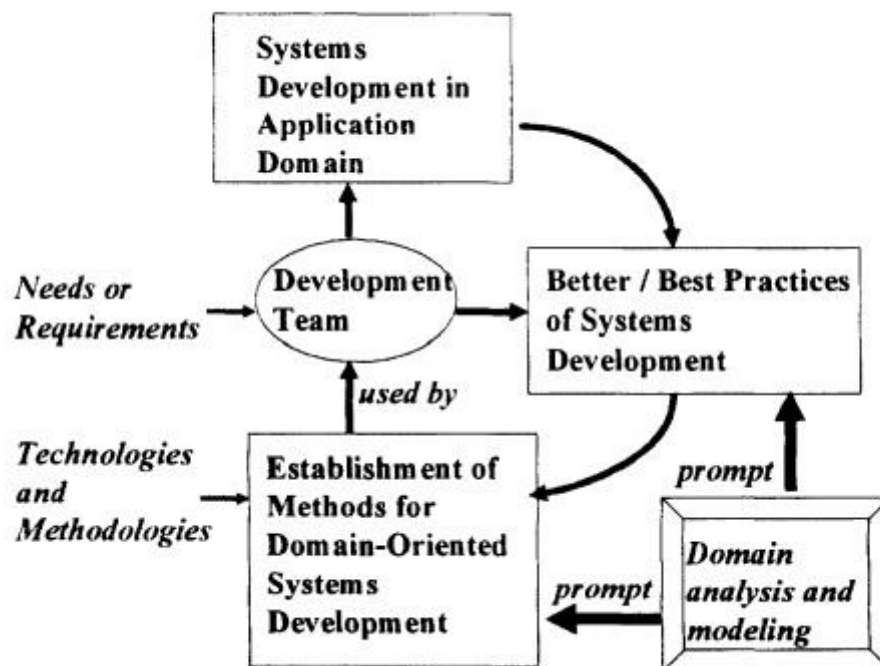
Svårigheter som kan stötas på vid användandet av MBSD är att det modellspråk som används inte kan beskriva alla relevanta egenskaper i verksamheten, och personer tenderar att ha svårt att helt sluta med dokumentbaserad systemutveckling. En del modelleringsverktyg saknar delvis eller helt möjligheten att ändra i koden utan styr all kodning via automatisk generering från modellen, detta kan leda till ineffektiv kod som gör systemet långsammare än nödvändigt. De kommersiella verktygen är i dagsläget också väldigt dyra vilket gör att färre vill gå över till MBSD (Amnell et al. u.å.).

Brown (2011) hävdar att modeller kan ha samma problem som dokument om inte en god kommunikation förs, då det entusiastiskt modelleras inom verksamheten kan modellerna som skapas vara motsägelsefulla eller redundanta. Det gäller därför att det finns en gemensam uppfattning om verksamheten. En annan nackdel med MBSD är att om inte själva modelleringsprocessen är bland de

första stegen i framtagningen av systemet så blir modellen endast en form av dokumentation och de kreativa fördelarna som tillkommer vid den modellbaserade arbetsprocessen förloras (Brown 2011).

## 2.3 Domänorienterad systemutveckling

Ett sätt att arbeta med MBSD är domänorienterat, Domain-oriented System Development (DOSD), som Itoh et al. (2005) beskriver som en form av best practise metodologi för att ta fram ett effektivt sätt att skapa system inom en viss domän. Genom att återanvända och modifiera gamla eller liknande system kan både både tid och pengar sparas. Det som är viktigt vid systemutvecklingen att urskilja de karaktäristiska drag, egenskaper, begränsningar, strukturer som är specifika för varje applikations domän. DOSD är en metod som siktar på att effektivt utveckla system genom att analysera och modellera de domänspecifika strukturerna. Att genomföra en förstudie på den metod som är tänkt att användas och hur den passar den domän som den är tänkt att appliceras på är viktigt vid arbete med DOSD. Itoh et al. (2005) tar i sin bok upp DOSD-cykeln som är ett begrepp som är allmänt accepterat inom mjukvaruingenjörskretsen. Genom att utvecklingsteam skapar applikationer inom sina domänområden och sedan samlas för att ta fram en “best practise” som sen kan användas inom liknande domäner (Itoh et al. 2005).



Figur 2.2 DOSD-cykeln visar på hur utvecklingsteam skapar nya applikationer och tillämpar “best practise” iterativt för att anpassa utvecklingen av metoder för DOSD (Itoh et al. 2005).

## 2.4 Coremetoden

En metod som bygger på MBSD är Coremetoden, som är en metod specialiserad just för informationshanteringssystem. Coremetodens verksamhetsanalys består av modellering av en verksamhet likt UML, och detta görs för att förstå och dokumentera en komplex verklighet. Objekten och attributen i en verksamhet tas fram genom intervjuer, observationer och liknande metodiker (Jonsson, personlig kommunikation, 5 maj, 2014).

Coremetodens arbetsprocess är objektorienterad och bygger på att verksamheter modelleras med objekt för att göra en lättöverskådlig överblick över verksamheten. Objekten i sin tur innehåller attribut och algoritmer som förklarar modellen mer djupgående (Mathiassen et al. 2001; Jonsson, personlig kommunikation, 14 maj, 2014). Mathiassen et al (2001) nämner att det finns många fördelar med att arbeta på detta sätt, bland annat att det minskar komplexiteten i beskrivningar och modeller blir lättare att förstå, samt att informationssystemen får högre kvalitet på mindre tid.

Coremetoden drar nytta av att objekten har attribut och relationer till andra objekt, och lägger också in regler, villkor och härledning med mera för att med automatik generera kod som slutligen blir till ett informationssystem. Detta är en fördel då uppdateringar och att lägga till funktioner i systemet kan göras efter behov. Genicore AB har utvecklat ett eget mjukvaruverktyg, Visual-S, som de använder för att skapa en modell av verksamheten. De har även utvecklat ett eget kompilierbart modellspråk som heter "S" vilket är specifikt anpassat just för MBSD i deras utvecklingsmiljö. De kan från mjukvaruverktyget och modellspråket generera olika programmeringsspråk som till exempel Java och C++. Med hjälp av det egna verktyget och språket så kan de automatisera kodgenereringen (Jonsson, personlig kommunikation, 24 april, 2014).

Inom Coremetoden är det även viktigt att göra en verksamhetsanalys för att hitta svagheter och styrkor inom verksamheten, och inom detta ämne finns det mycket relaterad forskning. Nilsson et al (2002) har forskat i ämnet och anser att det är faktorer som påverkar ett bolags vinst som är de viktigaste faktorerna, både för verksamheten själv men också för utomstående investerare. Genicore ABs verksamhetsanalys bygger på ett fenomenologisk synsätt och utgår från att objekt som personer, produkter, avtal med mera ska ses som fenomen. En grundsten inom detta synsätt är att utgå ifrån hur något upplevs av olika människor för att komma fram till hur något är i objektiv, alltså kan ett fenomen upplevas på olika sätt beroende på både situationen och individen (Giorgi 1985). Det fenomenologiska synsättet inom informationssystem kan ses som att alla i organisationen ser på objekten utifrån olika behov, och därför vill de ha olika attribut beroende på vilket yrke och roll som personen har. En ekonom använder sig av attribut som handlar om ekonomi, jämfört med att någon i produktionen vill ha attribut som handlar om vilka delar som ska monteras.



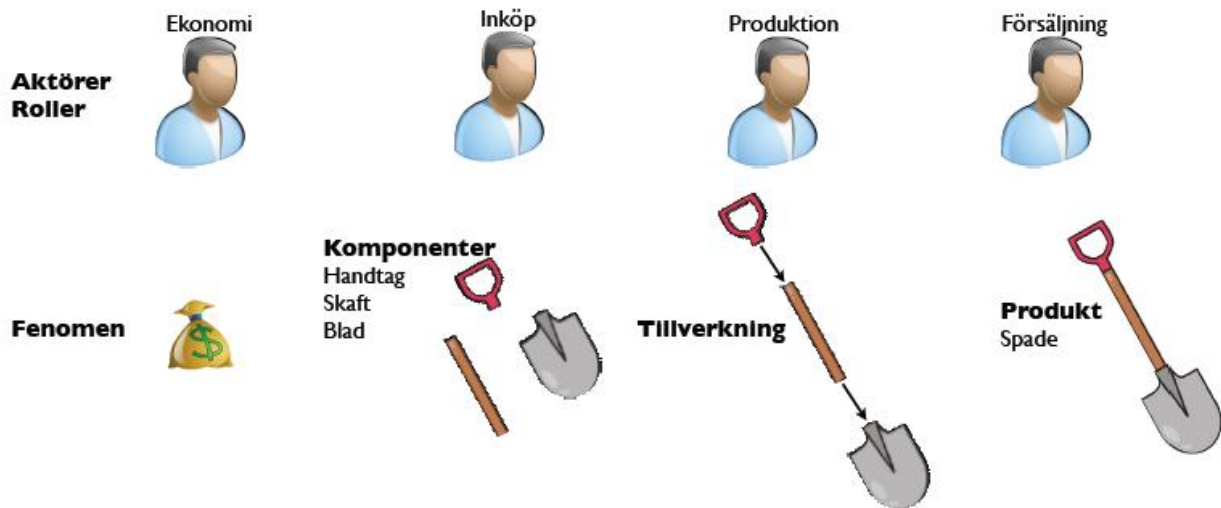


Bild 2.3 Bilden visar hur olika roller i verksamheten vill ha ut olika attribut av ett objekt, i detta fallet ett företag som tillverkar spadar (Jonsson, personlig kommunikation, 14 maj, 2014).

Det man letar efter i det fenomenologiska synsättet är fenomen som kan finnas i olika former:

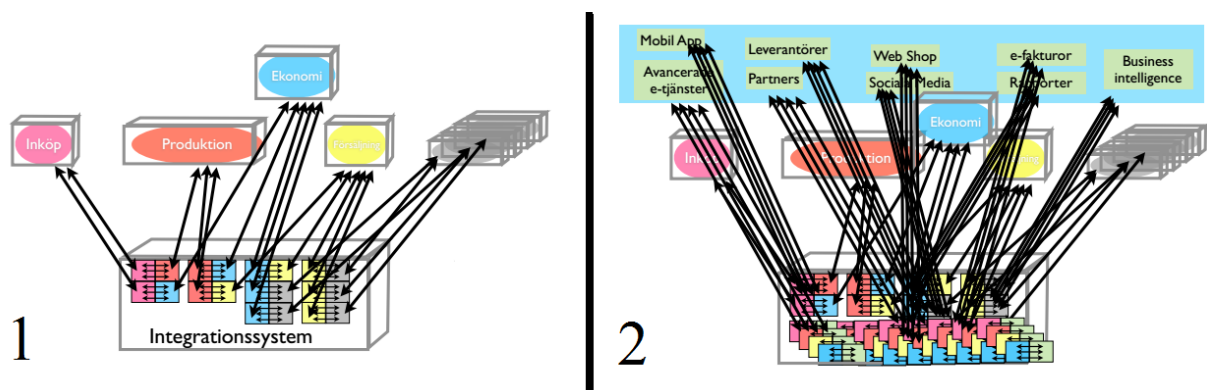
- Materiella saker: en spade, ett klassrum eller en dator.
- Händelseförlopp: ett möte, en föreläsning eller en resa.
- Sociala konstruktioner: ett äktenskap, ett avtal eller en anställning.
- Person: en anställd, en elev eller en chef.

Det är viktigt att generalisera fenomenen i verksamheten så länge de inte har olika egenskaper. Till exempel elever som studerar olika utbildningar som inte har skilda attribut behöver därför inte skiljas åt utan kan generaliseras till elev.

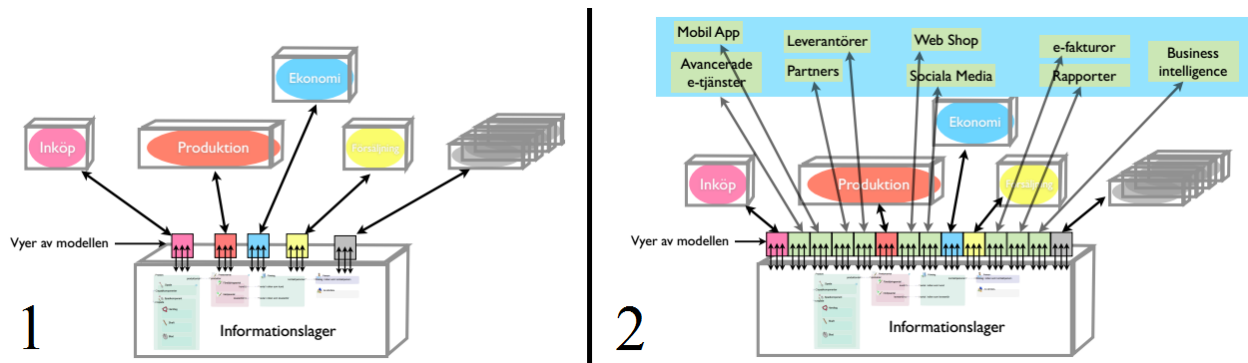
Det fenomenologiska synsättet skiljer från den sedvanliga processororienterade inriktningen som handlar om att alla delmoment har vissa egenskaper och följer en viss process. Jonsson (Jonsson, personlig kommunikation, 5 april, 2014) hävdar att skillnaden mellan fenomenologiskt synsätt och den processororienterade inriktningen kan illustreras med en taxichaufför som kör mellan olika platser varje dag. För att ta sig från A till B fungerar det bra med en processkarta som säger vad som bör göras, men om det skulle komma hinder på vägen som tvingar chauffören att ta en ny väg så tappar processkartan sin funktion. Ett fenomenologiskt angreppssätt hade istället varit att göra en karta över staden där taxichauffören kör. Kartan beskriver hur saker *är*, inte hur det bör *göras*. Chauffören hade lätt kunnat ta sig runt hindret genom att hitta en annan väg som tar honom till destination B, processer kan alltså förändras utan att kartan behöver det. Med hjälp av kartan kan alla i taxibolagets verksamhet använda sig av den, och den är anpassad för allas behov och kan generera alla vägbeskrivningar som

behövs. Resultatet är alltså en karta som kan generera alla möjliga sorters beskrivningar jämfört med att skapa en processkarta för varenda destination de ska till (Jonsson, personlig kommunikation, 5 april, 2014).

Det finns olika arkitekturprinciper vid arbete med de olika synsätten. Exempel på dessa är processaspekt och model-view aspekten med fenomenologisk verksamhetsmodell. Vid arbete med en processaspekt krävs det flera integrationsprocesser för varje ny applikation som ska integreras. Då systemen som informationen hämtas från oftast är skapade av olika utvecklare blir det ofta redundans i det sammanställda informationslagret då samma objekt i olika system kan ha varierande benämningar. Vid varje ny applikation som ska integreras krävs det nya kopplingar mellan informationskällorna, eftersom datan inte alltid är standardiserad behöver den då konverteras. Inom model-view så samlas all data från datasystemen inom domänen i en välstrukturerad informationsmodell som i bilderna nedan till kallas för informationslager. Detta innebär att alla system har en enhetlig källa av data som de hämtar och skickar information från och till. Genom att använda denna arkitekturprincip blir det mycket lättare att integrera nya applikationer. I Coremetoden görs dessa integrationslösningar med hjälp av integrationspråket Extensible Markup Language (XML).



Figur 2.4 Vid användning av ett processorienterat integrationssystem krävs det flertalet integrationsprocesser för varje ny vy. Steg 1 visar de integrationer som krävs för att koppla samman domänens datasystem. Steg 2 visar hur integrationsprocesserna blir fler vid integrationen av nya applikationer (Jonsson, personlig kommunikation, 14 maj, 2014).



Figur 2.5 Steg 1 visar model-view och hur det endast krävs en integration per system i domänen för att skapa ett välorganiserat informationslager där all information sedan uppdateras. I steg 2 visas hur applikationer kan integreras med informationslagret med endast en integrationsvy som sen både uppdaterar och blir uppdaterat av informationslagret (Jonsson, personlig kommunikation, 5 maj, 2014).

## 2.5 Andra modellbaserade systemutvecklingsmetoder

Det finns få modellbaserade systemutvecklingsmetoder som är namngivna likt Coremetoden, utan gemensamt för dem istället är att de arbetar med ett modelleringsspråk likt UML eller Specification and Description Language (SDL). Däremot finns det flertalet verktyg som företag använder sig av för att modellera i. En del av verktygen är strukturfokuserade, de andra är beräkningsfokuserade och det kan vara nödvändigt att använda mer än ett av dessa verktyg vid konstruktion av ett system för att få täcka hela utvecklingen. MBSD-verktyg använder sig av olika programmeringsspråk (C#, C++, java med mera) och inriktar sig på olika typer av modellering. Exempel på verktyg är MagicDraw(No Magic 2013), Simulink(MathWorks 2014). Alla dessa verktyg har olika egenskaper, stödjer utvecklingsprocessen av ett system olika långt med mera. Vissa har stöd för automatisk kodgenerering men inte alla, vilket gör att ytterligare ett verktyg kan behöva läggas till just för det ändamålet (Amnell et al. u.å.; Hause, M 2002).

En metod som till viss del arbetar modellbaserat är Rational Unified Process(RUP) som ägs av IBM Rational Software. De anser att en del av processen att ta fram ett system är att använda sig av visuella modeller för att förtydliga mjukvaran samt att de får struktur och en bra översyn över arkitekturen. Inom RUP används UML för att göra dessa modelleringar (Kruchten 2000).

En annan metod är Naked Objects(NO) som är framtagen av Naked Objects Group som betyder att de jobbar med "nakna objekt" verksamheten som de sedan modellerar upp den. Irlands informationssystem för socialförsäkring och pension är uppbyggt på NO. Skillnaden från det förra

systemet är att de numera jobbar med ett system istället för ett 50-tal olika system och därför kan hantera en medborgares ärenden som en helhet (Pawson 2004).

## 2.6 Kriterier för MBSD

Vid modellbaserad modellering är kvalitetstestning av mjukvara vanligt på ett tidigt stadie, då kodgenerering sker automatiskt är det enkelt och snabbt att undersöka hur funktionerna fungerar i praktiken. Genom att kontinuerligt genomföra så kallad quality assurance (QA) så undviks problem som annars kan uppkomma senare under projektets gång (Fey 2007).

Målet med MBSD formuleras av Oldevik et al. (2006) att snabbt och billigt kunna skapa högpresterande IT-system. För att lyckas med detta krävs det verktyg som stödjer utvecklingarna i de olika systemutvecklingsprocesserna. Att hantera och automatisera dessa processer är komplexa processer i sig och det bedrivs idag mycket forskning kring hur det på effektivast sätt kan hittas lösningar för dessa problem. Till följd av detta dyker det upp många system som försöker lösa komplexiteten inom detta område på ett så effektivt sätt som möjligt.

Ett sätt att effektivisera och automatisera kodning är att genom att använda sig av produktlinjer, vilket innebär att en kodbas utvecklas som sedan kan användas för att skapa mjukvara. Detta arbetssätt är anpassat efter kunder som efterfrågar olika produkter som finns inom samma domän och istället för att utveckla en helt ny kodbas för varje kund så bygger programmerarna på den kodbas som finns för att utöka registret av funktioner som kan generera system (Karlsson 2004). Oldevik et al. (2006) skriver i sin artikel att det är svårt att leta efter vilka funktioner som är speciella men de har tagit fram ett generellt ramverk för att utvärdera verktyg inom området. Att ta fram en fullständig lista av kriterier som helt täcker ett systems behov är dock inte möjligt enligt Oldevik et al. (2006). De kriterier som presenteras här är framtagna från relevant litteratur i ämnet. De olika punkterna är inriktade på att utvärdera olika delar av metoden. Kriterierna 1-6 ämnar att utvärdera allmän support för MBSD och hur väl ett verktyg stödjer modelleringens specifikation och transformation. Kriterium 7 utvärderar om metoden hanterar tjänstens kvalitet, eller som det vanligtvis benämns inom ämnet, quality of service (QoS), under modelleringens specifikation och transformation. Hur flexibel metoden och det verktyg som används är samt om det stödjer standarder, integration och intern samverkan evalueras i kriterierna 8-13. Om modellen är iterativ och inkrementell undersöks i 14-16. 17-21 handlar om hur specialanpassat verktyget är för att stödja de speciella behoven för produktlinjeutveckling. Generella egenskaper som ej är kopplade mot funktioner i verktyget ligger i fokus bland kriterierna 22-25. Kriterietabellen är uppdelad i fyra kolumner; characteristic, description, weight och critical. Det i syfte för att presentera vilken karaktär kriteriet är av, en beskrivning av kriteriekravet, hur stor vikt kriteriekravet har för verktyget samt om det är ett viktigt kriterium att uppfylla (Oldevik et al. 2006).

CID x.y	characteristic	description/question	weight 1–5	critical Y/N
1	model specification	does the tool support specification of systems as graphical models? <i>{Yes/No}</i>	4	N
2	graphical notation for model transformation	does the tool support graphical specification of transformation? <i>{Yes/No}</i>	1	N
3	lexical notation for model transformation	does the tool support lexical specification of transformation? <i>{Yes/No}</i>	5	N
4	model-to-model transformation support	does the tool support model-to-model transformation? (e.g., from one UML model to another?) <i>{Yes/No}</i>	4	N
5	model-to-text transformation support	does the tool support model-to-text transformation, such as generation of source code? <i>{Yes/No}</i>	5	Y
6	support for model analysis	is there any support for model analysis? <i>{Yes/No}</i>	1	N
7	support for QoS management	is there any support for managing QoS during model specification and transformation? <i>{Yes/No}</i>	1	N
8	metamodel-based	is the tool based on explicit descriptions of the metamodels of source and target transformation? <i>{Yes/No}</i>	3	N
9	MOF integration	is the tool integrated with a MOF (or other metamodel-based repository)? <i>{Yes/No}</i>	4	N
10	XMI integration	is the tool integrated with XMI? <i>{Yes/No}</i> which version(s) of XMI is supported? <i>{list of versions}</i>	4	Y
11	based on UML	is the tool based on UML models as source and/or target models for transformation? <i>{Yes/No}</i>	2	N
12	UML specification	does the tool provide support for UML modeling <i>{Yes/No}</i>	4	N
13	UML tool integration	can the tool be integrated with existing UML tools? either directly, as active plug-ins in UML tools, or indirectly through model exchange via, e.g., XMI? <i>{Yes/No}</i> or <i>{names of the set of techniques}</i>	4	N



14	iterative and incremental transformation support	does the tool handle reapplication of transformation after model updates? <i>{Yes/No}</i>	3	N
15	bidirectional transformations	does the tool support bidirectional transformations? <i>{Yes/No}</i>	1	N
16	traceability	does the tool handle traceability of transformations, i.e., can it maintain traces of the source and targets of a transformation? <i>{Yes/No}</i>	4	N
17	product line variability modeling	is there support for modeling product line variability? <i>{Yes/No}</i>	4	N
18	product line variability Resolution	is there support for variability resolution? <i>{Yes/No}</i>	5	Y
19	DSM language support	is there support for defining domain-specific modeling languages (e.g., UML profiling) and DSM transformations? <i>{Yes (1)/DSM Transformations (0,5)/No.(0)}</i>	4	N
20	QoS variability	is there support for modeling and resolving QoS variability? <i>{Yes/No}</i>	3	N
21	decision process support	is there support for a decision process? <i>{Yes/No}</i>	5	N
22	maturity	what is the maturity of the tool? <i>{Mature (0.7–1), medium(0.4–0.6), under development (0–0.3)}</i>	2	N
23	usability	what is the usability level of the tool? is it <i>{Easy and intuitive (0.7–1), medium learning curve (0.4–0.6), steep learning curve (0–0.3)}</i>	1	N
24	availability and license	what is the license for the tool? <i>{Open source (1), freeware (0.4–0.9), commercial(0–0.3)}</i>	2	N
25	pricing	what is the pricing of the tool? <i>{the approximate pricing (0–0.9), N/A (1)}</i>	4	N

Figur 2.6 Tabell över kriterielista framtagen av Oldevik et al. (2007).

## 3. Fallstudie; Coremetoden

En fallstudie har gjorts för att kunna applicera teorin på som har diskuterats ovan. Fallstudien har gjorts på en metod som heter Coremetoden, och studien går ut på att skapa en modell av IT-fakultetens utbildningsverksamhet på GU för att kontrollera ifall det är en gångbar metod för detta ändamål. I detta kapitel appliceras metoden på utbildningsverksamheten samt de kriterier som ligger till grund för att bedöma metoden.

### 3.1 Fallstudieobjekt - Coremetoden

Coremetoden är som tidigare nämnts en MBSD-metod som är utvecklad av Genicore AB med fokus på utveckling av informationshanteringssystem. Genicore ABs fokus med denna studie är att applicera metoden på en utbildningsverksamhet för att kontrollera om den ger ett användbart resultat, och vad slutsatsen blir av två studenter med minimala kunskaper och erfarenheter av MBSD. Metoden är speciellt framtagen för att appliceras på komplexa organisationer, vilket skol- och utbildningsverksamheten är enligt Agahi<sup>2</sup>. Då vi som undersöker metoden har spenderat snart tre år som studenter på GU har vi fått en grundlig inblick i hur utbildningsverksamheten ser ut som är en bra grund inför modellerandet. Vi har dock utökat vår förståelse av verksamheten och hur den kan kopplas till metoden genom intervjuer och litteraturstudier.

En intressant aspekt är att vi som har undersökt och modellerat med hjälp av Coremetoden i denna studie inte har någon tidigare erfarenhet av MBSD då vi mestadels har lärt oss dokumentbaserad systemutveckling på den systemvetenskapliga utbildningen. Dock har vi fått bekanta med många av de tekniker som återfinns i Coremetoden, bland annat UML där vi har jobbat med kluster, objekt och attribut. Vi har även fått lära oss att använda diverse mjukvaruverktyg som låg till en bra grund för att kunna komma igång snabbt och modellera. Poängen är att metoden undersöks och används av personer utan någon erfarenhet av den sedan tidigare samt på en verksamhet som Coremetoden inte har applicerats på, och därför kan hantering av komplexiteten i metoden sättas på prov.

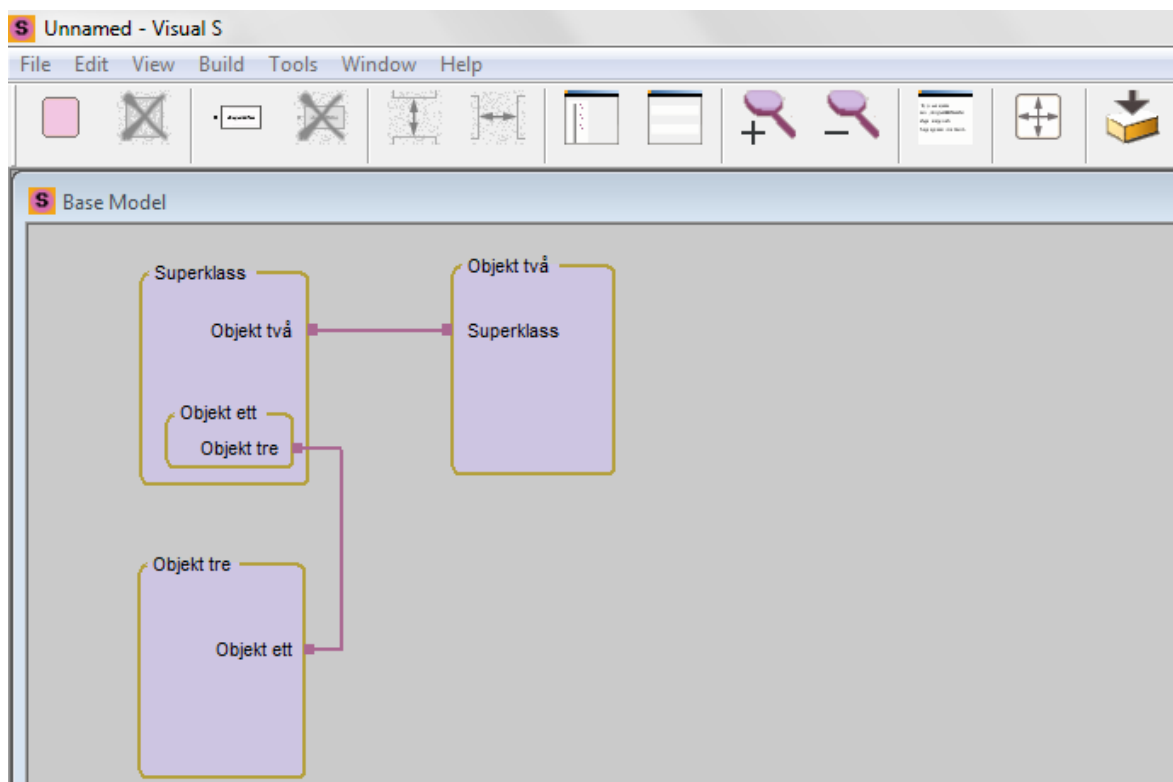
### 3.2 Mjukvaruverktyget Visual-S och modellspråket “S”

Vad som skiljer Coremetoden från andra MBSD-metoder är framförallt att Genicore AB har utvecklat ett eget verktyg som stödjer hela utvecklingsprocessen från modellering till kodgenerering. Metoden är alltså strukturfokuserad och beräkningsfokuserad. De har ihop med verktyget även tagit fram ett

---

<sup>2</sup> Faramarz Agahi vice-prefekt för grundutbildningen på IT-fakulteten, intervju den 5 maj 2014

eget modellspråk som är anpassat efter just deras verktyg och behov. Anledningen till att de valt att ta fram ett eget verktyg och språk är att det i deras mening inte fanns något språk eller modelleringsverktyg anpassat för modellering av informationshanteringssystem som behandlade problemområdet ur ett fenomenologiskt perspektiv. Genom att själva utveckla ett språk kunde de säkerställa att termologin och modelleringsprocessen var anpassad efter de krav som ett fenomenologiskt perspektiv ställer. I Visual-S är det viktigt att generalisera objekten så mycket man kan samt att lägga dem i möjliga superklasser, så att det blir lätt att på ett översiktligt sätt kunna se hur modellen, alltså verksamheten, är uppbyggd. Det är sedan objekten som blir fullmatade med attribut och relationer för att skapa viktig data om vad de gör och ska hantera (Jonsson, personlig kommunikation, 5 maj, 2014).



Figur 3.1 övergripande översikt av hur Visual-S ser ut vid modellering

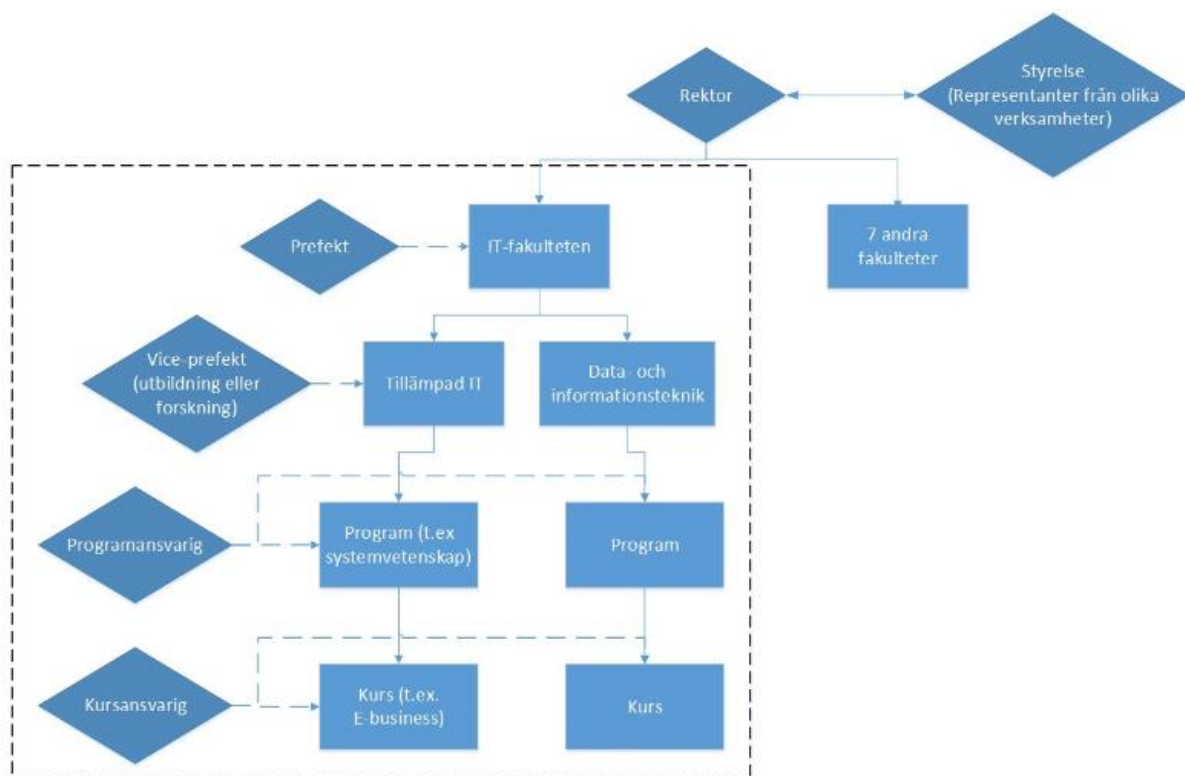
### 3.3 Modellering av utbildningsverksamheten på IT-fakulteten

Det kom fram under intervjun att de som arbetar inom utbildningsverksamheten använder sig av fler än fem system i deras vardagliga arbete, beroende på vilket yrke och vilka uppdrag de har så kan det bli fler. Några av de system de använder är deras läroplattform, antagningssystem, rapporteringssystem för utbytesstudenter med mera. Som nämnts tidigare skapar varje nytt system fler



integrationer och fler anpassningar behöver göras för standardisering, vilket ökar komplexiteten för verksamheten. Intervjun med Agahi<sup>3</sup> låg till grund för att kunna skapa en modell av utbildningsverksamheten i Visual-S i kombination med författarnas egen erfarenhet samt personlig kommunikation med lärare och studieadministration. I intervjun nämndes det flera gånger att verksamheten har en komplex verklighet, vilket uppdagas i denna studien trots att den endast hanterar en del av verksamheten.

Nedan är de roller och de organisationsdelar som finns i universitetsverksamheten presenterad. Den del som är i fokus i denna studie är avgränsad och befinner sig inom den streckade rektangeln. Olika roller representeras i diamantformade rutor och organisationsdelarna som rektangulära rutor. De streckade strecken som går från rollerna till organisationsdelarna är de områden som rollerna beslutar över, och det streck som går åt båda hållen innebär att de påverkar varandra eller hjälper varandra att ta beslut. De streck som är heldragna visar hur organisationsdelarna består av andra organisationsdelar. Denna modellering av utbildningsverksamheten låg till grund vid vidare arbete i Visual-S för den riktiga modelleringen.



Figur 3.2 De roller och organisationsdelar som är i fokus i utbildningsverksamheten. Figuren visar hur IT-fakulteten består av två institutioner som i sin tur består av program och kurser. På varje nivå finner man ansvariga för området. Varje fakultet har liknande strukturer under sig men det är just den struktur som IT-fakulteten har som kommer ligga i fokus i vår fallstudie.

<sup>3</sup> Faramarz Agahi vice-prefekt för grundutbildningen på IT-fakulteten, intervju den 5 maj 2014

## 4. Metod

För att kunna svara på vårt syfte har vi satt oss in i Coremetodens arbetsprocesser samt skaffat en övergripande förståelse för vilka steg de arbetar i. Därefter har vi gjort en noggrann litteraturstudie av allt material vi fått av Genicore AB, men också allt annat material som vi använt oss av. Vi har fört en öppen och ständig dialog med Genicore AB i syfte att skaffa oss den förståelse som behövs för att fullt ut kunna tillämpa metoden. Än så länge finns det bara litteratur som Genicore AB själva producerat kring Coremetoden att referera till. Detta i sin tur har gjort att vi varit extra kritiska. Vi har valt att använda oss av kvalitativa metoder som fallstudie och intervju då vi inte ville bredda kunskapen i uppsatsen, utan istället få ett djup i studien. (Patel & Davidson 2011). Det har visats sig att kvalitativa metoder inte enbart medför fördelar, utan det har även påvisats att de är svårare att kvalitetssäkra litteratur från forskare då de kan ha en subjektiv åsikt till ämnet (Klein & Myers 1999). Jämfört med kvantitativa metoder som kan kvalitetssäkras med hjälp av statistiskbearbetning och dataanalys (Patel och Davidsson 2011).

### 4.1 Fallstudie

Fallstudien har baserat sig på en metod som är framtagen av ett IT-företag stationerat i Göteborg, nämligen Genicore AB. Företaget inriktar sig på affärs- och systemutveckling med fördjupad kunskap i domänorienterad modellbaserad systemutveckling för framtagning av informationshanteringssystem. Företaget har varit aktivt inom området sedan 2001 och själva tagit fram den metod som vi valt att tillämpa på utbildningsverksamheten på IT-fakulteten vid GU. De verktyg och mjukvara som har används vid appliceringen är tillhandahavna av IT-företaget själva. En verksamhet som använder sig av ett informationshanteringssystem som är framtaget med Coremetoden är Försvarets materielverk (FMV) och det har cirka 1200 användare. Fallstudien går ut på att undersöka ifall utbildningsverksamheten går att modellera med hjälp av Coremetoden och ifall det är en gångbar metod för det ändamålet (Jonsson, personlig kommunikation, 24 april, 2014).

### 4.2 Intervju

Vi utförde även en semistrukturerad intervju för att få information om utbildningsverksamhetens komplexa verksamhet (se bilaga 2). Vi valde att ha en semistrukturerad intervju därför att vi ville kunna ställa frågor som är öppna men ändå har fokus mot ett visst ämne inom verksamheten. Denna intervjumetod öppnar också upp för följdfrågor och vi anser att detta gav oss stor nytta att specificera kunskapen ännu mer. Schneiderman och Plaisant (2010) anser att om intervjuerna läggs upp på detta

sätt så blir inblicken tydligare av de specifika problemen kontra om frågorna istället hade varit helt öppna eller stängda. Personen vi intervjuade är Faramarz Agahi, viceprefekt på IT-fakulteten och ansvarig för grundutbildningen på fakulteten. Han är väl insatt och har flera års erfarenhet i verksamheten samt de olika samarbeten och nivåer som finns inom den. Vi valde att spela in intervjun för att på bästa sätt kunna analysera den (Patel & Davidson 2011).

## 4.3 Dokumentanalys

De kriterier som är framtagna i denna studie har tagits fram genom dokumentanalys av relevant forskning och publikationer. Vi har sedan gjort ett urval av kriterierna med hänsyn till de avgränsningar vi satt. De kriterier vi valt att använda oss av är de som vi anser är mest relevanta för studien och vi har valt ett fåtal kvalitativa kriterier hellre än att ha en stor bredd av mindre relevanta. Vi la in de kriterier vi valt ut i ett Excel-ark där vi kunde strukturera upp dem på ett överskådligt sätt och lägga till kolumner där vi la till kommentarer samt en bedömningskala för att bedöma kriteriernas relevans och betydelse.

Dokumentanalys har valts som metod i syfte att skapa ett bredare dataunderlag för kriterierna samt att studien ska belysas ur fler synvinklar. Vid dokumentanalys så är det av stor vikt att förhålla sig kritisk till dokumentet för att skapa en trovärdighet till analysen (Patel & Davidson 2011).

### 4.3.1 Bedömningskriterier

Genom dokumentanalys av relevant litteratur kring MBSD-metoder och de verktyg som används vid MBSD då dessa är en stor del av själva arbetet, har vi tagit fram en egen lista med kriterier som vi anser är applicerbara på Coremetoden. Framst har Oldevik et al. (2006) och de kriterier som presenteras i deras artikeln (se Figur 2.6) legat till grund för studiens kriterier, men även Mohagheghi & Dehlen (2008) har fungerat som en försäkring om att de kriterier som studien redogör för är av relevans vid MBSD. Eftersom den verksamheten vi modellerat är komplex har vi valt att endast ta med de karaktäristiska dragen av kriterierna men bortsett från de värden som presenteras under kolumnerna weigh och critical. Vi har gjort ett urval av kriterierna som har anpassats efter de erfarenheter och lärdomar vi fått från andra litteraturstudier och fallstudien för att ta fram de kriterier vi anser mest relevanta för Coremetoden. En hel del av kriterierna faller automatiskt bort då vi avgränsat oss från de tekniska aspekterna som innefattas i många av de kriterier som listats av Oldevik et al. (2006). Även kolumnerna weight och critical har utelämnats då vår studie ska bygga på egna resonemang och slutsatser, något de kolumnerna inte var.

## 5. Resultat

Nedan redovisas den modell av verksamheten som metodanvändningen resulterade i för att kunna göra en utvärdering. Uppsatsen resulterade i en analys i form av en modell av en utbildningsverksamhet som kan ligga till grund för att vidare kunna kommunicera verksamhetens uppbyggnad, både internt och externt, men även kravställning och tillslut framtagning av ett IT-system. Vi har förhoppningar om att denna studie ska resultera i att öppna upp fler analysområden för Coremetoden att appliceras på. Samtidigt vill vi tillsammans med Tomas på Genicore AB kunna låta andra studenter och akademiker få läsa om och bedöma Coremetoden och ta tillfället i akt att sprida vidare ännu en metod att arbeta inom verksamhetsanalys med.

Med hjälp av Genicore ABs egna modelleringsmjukvara Visual-S har vi arbetat fram en informationsmodell, som är en typ av domänmodell, av utbildningsverksamheten på GU. Modellen (bilaga 3) visar på de informationsflöden som finns inom verksamheten och ska fungera som grunden till ett informationslager (se figur 2.5) vid eventuellt fortsatt arbete med mål att skapa ett informationshanteringssystem. Målet med den informationsmodell vi har skapat är att strukturera upp den information som skolan hanterar på ett överskådligt vis. Utöver det ska modellen fungera som en karta över hur verksamheten är uppbyggd som enkelt kan förstås av alla individer kopplade till verksamheten. Det som ligger till grund för hur modellen formats är de intervjuer vi genomfört med personal på universitetet samt de erfarenheter av hur verksamheten fungerar som vi som studenter har erhållit efter nästan tre års studier. Till grund ligger också de teorier kring MBSD som gäller vid modellering, som presenteras i teoriavsnittet, som ska hjälpa till att hantera komplexitet och i slutändan skapa kvalitativa modeller. Enligt tidigare presenterade avgränsningar har vi inte skapat några regler eller förhållningssätt mellan objekten i modellen och därmed inte heller skapat någon funktionell kod för utvärdering. Det vi har modellerat är de objekt som skolverksamheten bör ses innehålla ur ett fenomenologiskt perspektiv samt hur dessa objekten är kopplade mot varandra.

Vi har utifrån de erfarenheter vi erhållit efter arbetet med modellen kopplat dessa mot litteraturstudier kring andra MBSD-metoder och tagit fram en lista på kriterier som är relevanta vid arbete med modellbaserad systemutveckling. Dessa kriterier har vi sedan använt för att utvärdera hur väl coremetoden gick att applicera på utbildningsverksamheten på GU.

Vår studie har resulterat i en modell av utbildningsverksamheten samt kriterier för MBSD-verktyg och MBSD-metoder som vi sedan applicerat på Coremetoden. Genom den intervjun och dokumentanalys som har gjorts har vi fått en bild av hur verksamheten ser ut samt vilka de aspekter som är viktiga inom MBSD vad gäller verktyg och metod. Nedan kommer vi presentera resultatet av undersökningen i form av bilder från modelleringen av utbildningsverksamheten, samt tabeller och förklaring till tabellerna för att redovisa hur coremetoden bedöms med hjälp av kriterierna. Vår studie har resulterat i

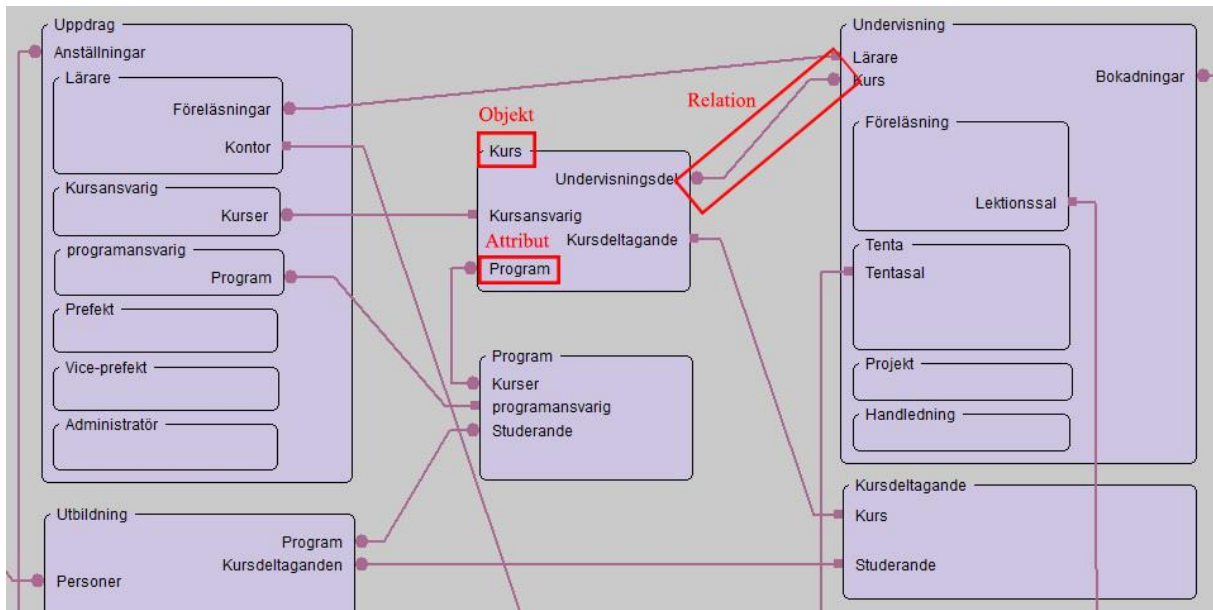
en modell av utbildningsverksamheten samt kriterier för MBSD-verktyg och MBSD-metoder. Genom den intervjun och dokumentanalys som har gjorts har vi fått en bild av hur verksamheten ser ut samt vilka aspekter som är viktiga inom MBSD vad gäller verktyg och metod.

## 5.1 Modellerung av utbildningsverksamheten

Vi har med hjälp av Visual-S gjort en modellering av utbildningsverksamheten på IT-fakulteten på GU. Vårt mål med denna modell har varit att få bättre förståelse över det fenomenologiska tankesätt som används vid modellering i coremetoden samt att se om metoden är lämpad för den modellerade verksamheten.

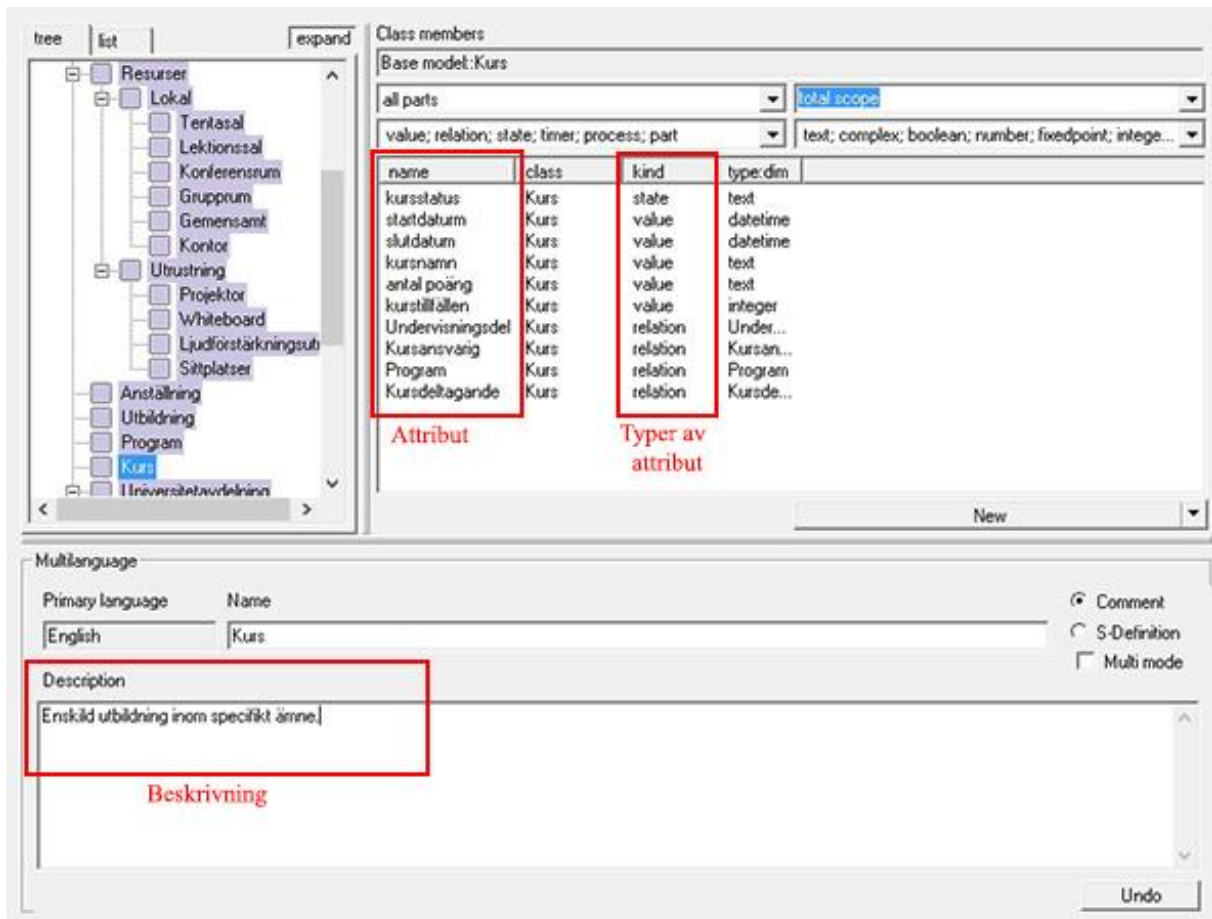
Som tidigare nämnts används ett fenomenologiskt synsätt vid modellering i Coremetoden. Det innebär att fenomen ligger till grund för alla delar i modellen. Ett exempel på hur fenomen kan upptäckas i verksamheten är att i en intervju fråga efter vilken information som informanten behandlar i olika situationer. Det kan då framkomma att informanten jobbar mycket med kurser och vem som bär ansvaret för vilken kurs. Utöver det hanterar kanske personen också vem som är ansvarig för olika program. Utifrån informantens uppgifter kan man härleda objekten kurs och program som händelseförlopp. Även objekten kursansvarig och programansvarig kan upptäckas som sociala konstruktioner i form av ett avtal mellan universitet och en person.

Då utbildningsverksamheten är väldigt komplex har modellen fått en något studentaktig synvinkel av de delar som inte lyckats beskrivas av informanterna. Hela grafiska vyn av modellen finns i bilaga 3 men i detta avsnitt visas utdrag på en mer detaljerad nivå. Dessa kan tas fram i modellen för att bättre se hur modellen är uppbyggd. På nästa sida visas hur några av de objekt som modellen innefattar hänger ihop med olika relationer av attribut. En rund prick i ena sidan av en relation menar att det kan finnas flera relationer av detta attribut till objektet, en liten rektangel innebär att endast en relation kan finnas till attributet. Exempelvis en kursansvarig kan ha ansvar för mer än en kurs, men en kurs kan bara ha en kursansvarig.



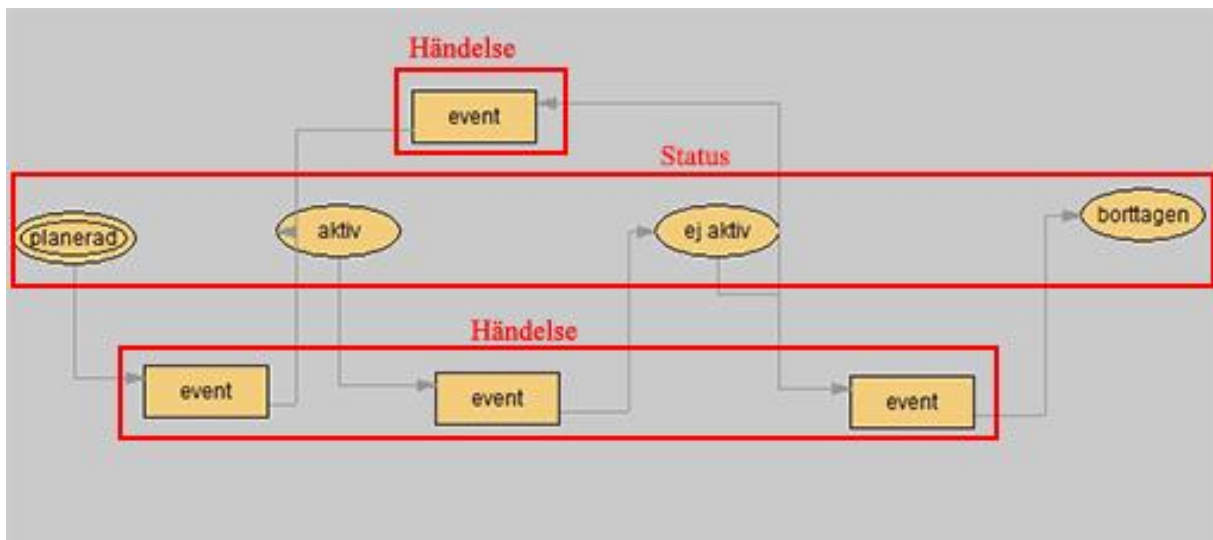
Figur 5.1 Översiktsvy av objektet kurs i modellen.

För att lägga in attribut i objekten så dubbelklickar användaren på objektet. Då kommer en mer detaljerad vy för objektet kurs, som visas i bilden nedanför. Objektet har flertalet gömda attribut, med andra ord attribut som inte syns från översiktsvyn (figur 5.1). Dessa kan vara av olika typer så som relation, tillstånd eller värde. Ett relationsattribut är en koppling mellan två objekt, ett värdeattribut är ett värde för något i objektet, så som kursnamn eller startdatum för en kurs, och ett tillståndsattribut (state) visas i figur 5.3. En beskrivning av objektet finns även längst ner för en tydligare definition av objektets roll.



Figur 5.2 Objektet kurs och de attribut som lagts in i detta objekt.

Attributen kan ligga som förändringsattribut i objekt, nedanför visas förändringsschemat för förändringsattributet kursstatus i objektet Kurs. Detta visar de olika tillstånd ett attribut i objektet kan ha och hur dessa tillstånd kan ändras i ett flödesschema. I detta fall ser vi hur en kurs börjar som statusen planerad och därefter går till aktiv. Som aktiv kurs kan den sen gå till ej aktiv och därifrån antingen till aktiv igen eller till borttagen. Rektanglerna med event symboliserar att en händelse sker, till exempel att ett beslut tas för att ta bort en kurs.



Figur 5.3 Förändringskedjan för attributet kursstatus.



Nedanför visas all kod som genererats för objektet kurs, I koden finns all information om objektet och dess attribut, relationer och förändringsscheman. Allt som visas i figurerna 5.1, 5.2 och 5.3 finns representerat koden.

```
class 'Kurs' "Enskild utbildning inom specifikt ämne."
  stateAttribute 'kursstatus' "olika statusarna en kurs kan ha."
  as text
  possible
    'planerad'
    'aktiv'
    'ej aktiv'
    'borttagen'
  value
    from 'planerad' to 'aktiv'
    from 'aktiv' to 'ej aktiv'
    from 'ej aktiv' to 'aktiv'
    from 'ej aktiv' to 'borttagen'
  default 'planerad'
  valueAttribute 'startdatum' "Datum för kursens start."
  as datetime
  valueAttribute 'slutdatum' "Datum för kursens slut."
  as datetime
  valueAttribute 'kursnamn' "namnet på kursen."
  as text
  valueAttribute 'antal poäng' "Antalet högskolepoäng som kursen motsvarar."
  as text
  valueAttribute 'kurstillfällen' "Antalet instanser då kursen genomför undervisning."
  as integer
  relationAttribute 'Undervisningsdel' "Koppling till de undervisningsdelar som kommer ingå i kursen."
  many 'Undervisning' role 'Kurs'
  relationAttribute 'Kursansvarig' "kopplingen mot vem som kommer vara kursansvarig."
  a 'Kursansvarig' role 'Kurser'
  relationAttribute 'Program' "Koppling mot program som kursen kan vara del av."
  many 'Program' role 'Kurser'
  relationAttribute 'Kursdeltagande' "Koppling mot de individer som blivit antagna till kursen."
  a 'Kursdeltagande' role 'Kurs'
end class
```

Figur 5.4 Utdrag av kod dokumenterad i modellspråket S av objektet Kurs.

På detta sättet är alla objekt uppbyggda med relevanta relationer mellan objekten, förändringskedjor och definitioner. Det är utifrån skapandet av denna modellen som vi har valt att betygsätta verktyget och metoden med de olika kriterierna vi har tagit fram.

## 5.2 Bedömningskriterier samt applicering på Coremetoden

Utifrån de bedömningskriterier som är presenterade i figur 2.6 så har vi sammanfattat en lista med kriterier som är anpassade för just utvärdering och bedömning av hur väl Coremetoden fungerar att applicera på skolverksamheten. Vi har valt att endast använda de ställda kriterierna som presenteras

och inte till de kolumner som betygsätter kriteriet. Då den verksamhet och metod som vi har modellerat efter är så komplex blev det svårt att sätta relevans för vad siffrorna representerade. Relevansen i dessa tabeller är den relevans vi själva uppfattat att de har i MBSD-verktyg och MBSD-metoder. De kriterier som har genererats från dokumentanalysen kriterier för att bedöma modelleringsverktyget och metoden ur ett så brett perspektiv som möjligt. För att få varierande kriterier som bedömer olika delar av metoden och verktyget har vi valt att plocka ut kriterier som både fångar upp styrkor och svagheter samt mer allmänt konstruktiva kriterier som kan kopplas mot alla systemutvecklingsmetoder. Vi anser att denna bredd kan resultera i en översiktlig utvärdering av fallstudien som kan jämföras med andra systemutvecklingsmetoder. I dokumentanalysen upptäcktes det inte några kriterier för MBSD-metoder, vilket hämmar denna studie men har istället gett ett större fokus på kriterier för att bedöma verktyget. Kriterierna är därför uppdelade i verktygsanpassade och metदानpassade. De sistnämnda är framtagna genom att omvandla verktygskriterierna till att bli metदानpassade.

Nedan följer de olika kriterierna vi har tagit fram samt det resultat vi har kommit fram till genom att arbeta med verktyget och metoden på egen hand. Det är alltså våra egna erfarenheter och tolkningar som ligger till grund för det resultat som presenteras. Bedömningsskalan som finns i tabellen har en skala från ett till fem där en etta i till exempel metodens användbarhet betyder att den är svår att modellera med medan en femma betyder att metoden är lätt att modellera med. I fallen med programmering och dokumentation så syftar skalan på hur mycket programmering och dokumentation som behövs göras, där en etta är minimalt och en femma betyder att det krävs mycket av dem för att använda sig av metoden eller verktyget. Vi anser att en låg siffra är ett bra betyg då vi anser att speciellt inom metod så ska det krävas så lite programmering som möjligt. Att jobba med metoden bör blanda in så lite teknik som möjligt då det underlättar så att alla kan förstå. Varje kriterium tillsammans med betyg från skalan har en avslutande kommentar i tabell där ett resonemang till betyget finns presenterad. Där är de mest väsentliga synpunkterna samlade för att återkoppla till varför de har fått det betyg de har fått, och en vidare beskrivning av de egenskaper som ligger till grund för deras betyg finns under varje tabell samt med samma rubrik som kriterierna har i tabellen. I vissa fall har det funnits aspekter som vi anser har större påverkan än andra, och därför har dragit upp eller ner betyget. Till exempel var metoden användbar i sin helhet, men att förstå sig på det fenomenologiska synsättet krävdes alltför mycket tid, kraft och bearbetning för att användaren skulle förstå hur det fungerade och vad det gick ut på. I detta fallet drog detta ner betyget till en tvåa då det fenomenologiska synsättet krävde så mycket utrymme av metodens modellerande sett till användbarhet.

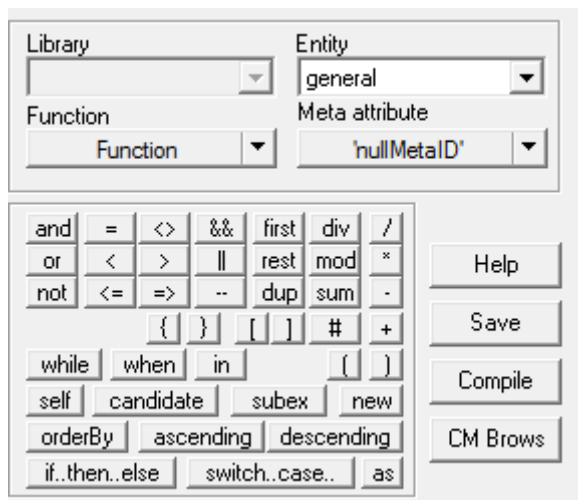
## 5.2.1 Verktyg

Kriterier för verktyg	Skala (1-5)	Kommentar
Användbarhet - Hur lätt är verktyget att använda?	4	Sett till modellering är verktyget i nästan alla aspekter logiskt och konsekvent att arbeta i. Det finns dock mindre buggar och saknas vanliga kortkommandon.
Mognad - Hur pass utvecklat är verktyget?	3	Verktyget har fortfarande några buggar och saknar en del kommandon men har i sin helhet en bra grund.
Stödjer verktyget dubbelriktad omvandling (mellan modell och kod)?	3	Verktyget kan hantera både manuellt skriven kod i språket S eller generera kod vid modellerande i det grafiska gränssnittet.
Kan dokumentation skapas från modellen?	3	Dokumentation med beskrivningar av modellens delar skapas i verktyget. Ett fönster med koden går även att framställa.
Finns det stöd för automatisk felsökning?	4	Verktyget sköter sin egen kompilering vilket innebär att den automatiskt upptäcker fel som kan uppstå när koden bryter mot de syntaktiska regler språket är uppbyggt efter.
Främjar verktyget översiktlighet?	4	Det går lätt att modellera i verktyget och det är även väldigt lätt att översiktligt kunna se över modellen.
Stödjer verktyget integration och Kooperation med andra vanligt förekommande systemutvecklingsverktyg?	2	Verktyget kan producera en fil ämnad för att skapa kod i andra mer vanligt förekommande programmeringsspråk. Detta kräver dock ett externt verktyg.
Stödjer modelleringen iterativa uppdateringar?	5	Verktyget stödjer iterativa uppdateringar både i modellerandet och i kodgenereringen

Figur 5.1 Visar de kriterier och den skalenliga bedömning Visual-S fått. Även en kommentar till varje bedömning.

### **Användbarhet - Hur lätt är verktyget att använda?**

Verktyget Visual-S är ett bra verktyg och det är lätt att komma igång på direkten när man startar det, och det finns en bra övergripande panel överst i programmet för de mest använda funktionerna. Objekt, attribut och relationer läggs enkelt till. Det som kräver lite kunskap av användaren är när attribut, förändringskedjor och regler ska läggas till då det kräver att man tänker programmeringsmässigt och bör ha erfarenhet sen tidigare av det, men detta gick att kompensera med hjälp av manualen som följde med. Funktioner för dessa kommer man enkelt åt genom att dubbelklicka på det objekt som skall få de specifika egenskaperna.



Figur 5.21 Hjälpverktyg för att lägga till regler i Visual-S

En annan funktion är att det går att lägga in beskrivningar om alla objekt, attribut och relationer så att alla delar i modellen är väl förklarade och det blir lätt för utomstående personer att förstå dess funktioner. Visual-S användbarhet är mycket god och får 4 av 5 i betyg.

### **Mognad - Hur pass utvecklat är verktyget?**

Genicore AB har jobbat med och utvecklat Coremetoden i företaget sedan 2001, men startade under mitten av 80-talet på det som nu är Coremetoden. Metoden har utvecklats under tiden företaget har utvecklats och har mognat under denna tid, vilket har gjort att verktyget har utvecklats parallellt. Verktyget har en välutvecklad grund och understödjer utvecklingen från modellering till att generera kod i språket S. Det är sedan möjligt att via ett annat Genicore-utvecklat system generera kod på andra programmeringsspråk, vilket dock inte har testats i denna studie.

### ***Stödjer verktyget dubbelriktad omvandling (mellan modell och kod)?***

Visual-S är ett verktyg som hanterar modellspråket S, det är möjligt att utifrån detta språk eller grafiskt i verktyget skapa modeller. Omvandling från modell till kod och kod till modell (skriven i språket S) kan hanteras av verktyget Visual-S. Annan kod än den skriven i programspråket S kan ej hanteras direkt i verktyget.

### ***Kan dokumentation skapas från modellen?***

Det skapas ständig dokumentation i verktyget som kan kommas åt via panelen överst i verktyget. Via den får användaren fram ett dokument med koden där alla objekt, attribut, relationer och regler finns att finna. Har man även definierat alla delar så presenteras detta i dokumentet. Verktyget kan skapa en typ av XML-fil, en SXML-fil, som används för att överföra modellen till andra verktyg eller skapa kod skriven i andra programmeringsspråk

### ***Finns det stöd för automatisk felsökning?***

En viktig funktion som finns i verktyget är att man kan kontrollera om modelleringen är notationsmässigt rätt, alltså om objekt, attribut och relationer med mera är gjorda på rätt sätt och stämmer överens. Detta görs varje gång man kompilerar modellen (programmet kontrollerar så att det användaren har skrivit stämmer överens med modellspråkets notation). Skulle ett fel i notationen ha gjorts under någon del av modellerandet så kommer det upp ett felmeddelande vid detta tillfälle. Samtidigt som detta sker så sparas även programmet.

### ***Främjar verktyget översiktlighet?***

Visual-S bygger på att modelleringen av verksamheten sker på ett så överskådligt sätt som möjligt. En grundaspekt är att användaren ska generalisera objekten så mycket som möjligt, vilket också främjar översiktligheten. Därför kan Visual-S anses som ett verktyg som främjar översiktlighet på ett gångbart sätt. Då målet med modellen som byggs upp är att alla i verksamheten ska förstå den fyller den också värde för alla som ser på den till skillnad från ett långt dokument med endast kod som i princip enbart förstås av programmerare.

### ***Stödjer verktyget integration och Kooperation med andra vanligt förekommande systemutvecklingsverktyg?***

Då den kod som genereras från modeller i Visual-S är skriven i språket S som är ett språk som endast används av Genicore AB krävs det en omvandling från S till annat språk för att integrering mellan system skapade i Visual-S och system skapade av andra verktyg ska kunna genomföras. Visual-S kan dock generera en typ av XML-fil, denna fil kan skickas över till Genicore ABs CoreBuilder, ett verktyg som används för att generera kod i andra språk än S, för att generera SQL, Java eller C++. Från denna kod kan integration ske mellan olika typer av system.

### *Stödjer modelleringen iterativa uppdateringar?*

Visual-S stödjer iterativa uppdateringar av modeller vilket vidare appliceras i verktyget för kodgenerering och därmed även stödjer iterativ utveckling av IT-system.

## 5.2.2 Metod

<b>Kriterier för metod</b>	<b>Skala (1-5)</b>	<b>Kommentar</b>
Användbarhet - Hur lätt är det att modellera i metoden?	2	Att modellera var lätt, men att förstå sig på det fenomenologiska synsättet och veta vad som är fenomen med mera är det som vi känner hämmar metod och är svårast för andra att förstå
Mognad - Hur pass utvecklad är metoden?	4	Metoden har en gedigen historia och har utvecklats under en längre tid. Vi fick ett lyckat resultat och vi vet att metoden fungerar praktiskt för att framställa IT-system, och därför anser vi den vara en välutvecklad metod
Hur mycket programmering kräver metoden?	2	På grund av att vi har avgränsat oss från att tekniskt gå in och sätta regler och villkor så har vi inte fått behandla det som riktar sig mot programmering i metoden, men man bör ha kännedom om programmering för att förstå metodens upplägg.
Hur mycket dokumentation behöver göras i metoden?	2	Den dokumentation som behöver göras är att beskriva alla objekt, relationer, attribut med mer. Utöver det görs all dokumentation automatiskt.
Främjar metoden översiktlighet?	5	När en färdig modell av ett system är skapad ges en mycket översiktlig bild av hur verksamhetens informationsflöde ser ut och hur verksamheten är uppbyggd. Oavsett vilken roll inom företaget betraktaren har kommer modellen vara begriplig och därmed generera värde.

*Figur 5.1 Visar de kriterier och den skalenliga bedömning Coremetoden fått. Även en kommentar till varje bedömning.*

### ***Användbarhet - Hur lätt är det att modellera i metoden?***

Har användaren erfarenhet av UML sedan tidigare, så underlättare det modelleringen avsevärt. Modelleringspråket "S" är lätt att förstå och bygger på UMLs objekt och attribut, vilket är relativt lätt att lära sig på kort tid. Det svåraste med modelleringspaketet var generaliseringen av alla objekt och att inte fördjupa sig för mycket i objekten. Metoden ser verksamheten ur ett fenomenologiskt perspektiv vilket inte är så vanligt förekommande i andra metoder. Detta innebär att användaren måste vara införstådd i metodens tankesätt för att kunna modellera på ett korrekt sätt.

### ***Mognad - Hur pass utvecklad är metoden?***

Likt metodens verktyg så har metoden själv mognat under tiden som Genicore AB har jobbat med den, och metoden har en lång erfarenhet av utveckling. Metoden bygger på UML som är en vedertagen metod som många användare har erfarenhet av sedan tidigare, och även regler och tillstånd som finns bygger på vedertagna språk.

### ***Hur mycket programmering kräver metoden?***

För att kunna lägga in regler och tillstånd måste användaren kunna grunderna i något programmeringsspråk som Java eller C#, och därför bör användaren kunna en del programmering för att kunna tillämpa metoden. Dock bygger metoden framförallt på modellering vilket gör att programmering kommer i andra hand.

### ***Hur mycket dokumentation görs i metoden?***

Den dokumentation som behöver göras är de beskrivningar som görs kopplat till varje objekt, attribut och relation. Dessa beskrivningar bygger på endast en eller två rader som beskriver det valda modellelement. Detta görs framförallt för att andra användare snabbt ska förstå vad varje del betyder och hänvisar till.

### ***Främjar metoden översiktlighet?***

Som tidigare nämnts så bygger Coremetodens modelleringspråk på att man modellerar upp verksamheten och lägger attribut på varje objekt som finns modellerat. Detta resulterar i en mycket god översikt där man enkelt kan se verksamhetens uppbyggnad ur ett perspektiv som alla delar av verksamheten begriper.

## 6. Diskussion

Visual-S är ett enkelt verktyg att arbeta i som håller god logisk och konsekvent uppbyggnad, det är därför lätt att börja sin modellering. Svårare är det att få grepp om det fenomenologiska tänket som metoden kräver för att man ska lyckas skapa en bra och tydlig modell. Detta kan krävas av alla metoder, såväl modellbaserade som dokumentbaserade. Vi har som presenterades i avgränsningarna inte haft tillgång till det webbaserade verktyget som har fler funktioner än den version vi använt oss av. De betyg som metoden och verktyget har fått i våra kriterier är baserade på hur väl det verktyg vi har använt oss av är anpassat för att fungera i vår fallstudie. Exempel på funktioner som inte har kunnat tas med i vår utvärdering är en ordlista som annars kan plockas ut ur modellen, samt en grafiskt tilltalande funktion som låter användaren färglägga olika objekt vilket kan ge en ökad översikt i den grafiska representationen. Det har inneburit att vissa delar av det vi har utvärderat kanske hade fått ett annorlunda resultat om vi använt den webbaserade versionen istället.

I Coremetoden skiljer sig inte benämningar åt mellan olika avdelningar av företaget. Det kan vara en bra sak som minskar missförstånd vid kommunikation men som samtidigt kan ligga till grund för att en del fackliga termer zonas ut. Då terminologin inom ett område kan vara viktig för att tydligt definiera specifika delar inom det området, kan det samtidigt vara viktigt att ha i åtanke att denna metod är anpassad för att få information att bli välorganiserad ur ett perspektiv som ska fungera för hela verksamheten.

Då det var svårt att ha några förväntningar på en metod som sedan tidigare är för oss okänd har vi inte direkt kunnat få fram något oväntat resultat. Det som kanske kom som en överraskning var den enorma komplexitet som den lilla del av universitetets verksamhet vi valt att modellera presenterade. Vi anser ändå att metoden var bra på att hantera den komplexitet som vi ställdes inför. Det faktum att vi angränsat oss från att skriva kod var något som i efterhand var ett bra val då det hade tagit stora delar av vår tid och detta hade resulterat i en lägre kvalitet på övriga uppsatsen, något vi inte hade önskat. Vi ser hellre att fokus i denna uppsatsen fick ligga på just modelleringen och analysen av den komplexa verksamheten som fallstudien gjordes på. Även om studien och det resultat som presenteras från studien känns relativt litet så har det innefattat stor komplexitet för oss som arbetat med den, det har varit svårt att förmedla den komplexitet arbetet innefattat ut i text vilket har betytt att vi har fått jobba extra mycket med det.

Vid en analys av resultatet så är det lätt att förstå att verktyget har en stor inverkan på den metoden som användarna arbetar med. Aspekter som hur långt verktyget stödjer utveckling, från modellering till automatisk kodgenerering, till vilka funktioner som finns i verktyget har en stor inverkan på hur bra möjligheter det finns till att jobba med metoden. Därför kan vi se en klar fördel med att Genicore AB har utvecklat ett eget verktyg med tillhörande modellspråk som stödjer hela deras



utvecklingsprocess, jämfört med företag som köper in färdiga MBSD-verktyg som inte är anpassade efter deras egen utvecklingsprocess.

Något vi upplever kan bli smidigare med just coremetoden är dess flexibilitet kring integration med andra system. I dagsläget är coremetoden ensam med att använda sitt programspråk och verktyg, detta kommer försvåra för applikationer utvecklade av andra parter att integreras med coremetodens applikationer. Det finns lösningar på detta men vi anser att dem kan vara något väl krångliga och kanske ska försöka underlättas om metoden ska få större gehör på marknaden.

Översiktlighet både i metoden och i verktyget är av stor vikt för att modellen ska kunna förstås och användas av så många olika anställda som möjligt inom verksamheten. Desto fler som använder sig av modellen och begriper den kan även hjälpa till att utveckla och förbättra den, och även jobba mot de gemensamma mål som finns i verksamheten. En översiktlig modell kan även vara ett bra sätt att se hur pass väl designat systemet kommer bli, är modellen rörig finns det en överhängande risk att även systemet kommer att få krångliga interna processer.

Att ha en mogen metod är också viktigt, speciellt när det kommer till verktyget som ska användas och att detta är välutvecklat. Att verktyget har alla de funktioner som behövs och att det inte är fullt av buggar som får det att krascha är viktigt för den som jobbar med verktyget, även att det finns stöd för hela utvecklingen från modellering till kodgenerering. Andra aspekter är att det ser bra ut, är konsekvent och smidigt att jobba. Dessa egenskaper kommer att förstärka användarens vilja att fortsätta arbeta i verktyget. Att ha ett moget verktyg går hand i hand med att ha ett verktyg med hög användbarhet som också var ett kriterium vi valt att ta med. Med en mogen metod menar vi en metod som har ett bra arbetssätt och som resulterar i fungerande system i slutändan. Givetvis är detta viktigt och något som alla metoder eftersträvar då ingen vill använda en metod som genererar ett dåligt system.

Trots att det enligt oss förekommer onödig mycket dokumentation inom utvecklingsfasen av system så anser vi fortfarande att dokumentation behövs till viss del, både i verktyg och i metod. Detta kriterium har paralleller med översiktlighet då dokumentationen ligger som underlag för att skapa en gemensam bild samt för att veta hur processerna i verksamheten ser ut för de som jobbar i den. Dokumentation behövs även inom verktyget då förutsättningar för att andra utvecklare ska kunna jobba med det ökar då skaparna har gjort en användbar dokumentation från början.

Det sista kriteriet som vi anser är av större väsentlig grad är att metoden och verktyget har stöd för en iterativ arbetsprocess, speciellt för verktyget då vi anser att det ligger till störst grund för att kunna uppdatera och effektivisera verksamheten. Då dagens teknik utvecklas kommer även de system som hanterar tekniken behöva göra det, att snabbt kunna uppdatera sina system utan stora investeringar blir viktigare och viktigare.

Efter att ha jobbat med både dokumentbaserad systemutveckling under vår utbildning och nu testat på den modellbaserade systemutvecklingen ser man tydligt hur mycket mer överskådlig och lättare det är att ändra i den modellbaserade. Har man förstått en modell och en verksamhets arbetssätt är det enkelt att förstå hur man ska inkorporera nya delar i modellen. Motsatt är det med dokumentbaserad systemutveckling där man noggrant beskriver hur en viss del ska fungera i verksamheten men inte får någon överblick av verksamhetens helhet vilket försvårar implementation av nya applikationer.

Vid en jämförelse med tidigare forskning och teori som presenterats så kan vi se att det är ett relativt utforskat område med få metoder. Resultatet visar att många företag ser nytta att använda sig av modellbaserad systemutveckling när de använder sig av produktlinje i kodgenereringen, men inte vid fortsatt arbete. Vi kunde även se i vårt resultat att det finns möjligheter att använda sig av Coremetoden för att utveckla system för stora myndigheter och liknande verksamheter som hanterar dokument och information i stora informationshanteringssystem. Däremot anser vi att företag som till exempel tillverkningsindustrier inte är lämpade för denna metod. Detta på grund av att tillverkningsprocessen på en industri är väldigt processinriktad, och därför bör metoden de jobbar med också vara processinriktad. Likt Agahi<sup>4</sup> nämnde i intervjun så är utbildningsverksamheten en kunskapsorganisation som hanterar information, och därför är den anpassade för att hanteras av Coremetoden som är utvecklad för informationshantering.

För att testa Coremetoden på bästa möjliga sätt så borde vi ha studerat fler fakulteter för att kunna se hur den hanterar komplexitet, som metoder inom MBSD ska skapa en modell av och hantera på ett effektivt sätt. Vi anser att mjukvaruverktyget Visual-S är ett användbart verktyg för att modellera i, och att det fyller sin funktion för att göra Coremetoden applicerbar på en utbildningsverksamhet. Vi borde även ha skickat ut en mall med de frågor vi ställde till Faramarz Agahi innan vi intervjuade honom för att han på bästa sätt skulle kunna vara förberedd och ge så bra svar som möjligt under intervjun.

---

<sup>4</sup> Faramarz Agahi vice-prefekt för grundutbildningen på IT-fakulteten, intervju den 5 maj 2014

## 7. Slutsats

Avsikten med denna studie har varit att utreda frågan: *”Vilka kriterier är av särskild relevans vid implementation av en MBSD-metod på komplexa verksamheter?”*.

Att arbeta med översiktlighet och integrationsmöjligheterna kopplade mot metoden anser vi vara mycket viktigt. I dagens IT-miljöer jobbar inget system helt självständigt, alla system har kopplingar som möjliggör integration av fler funktioner i verksamheten. Genom att ha en bra översiktlighet underlättas integrationen av nya system då det blir lättare att se hur det nya systemet ska kopplas in till de existerande.

Användbarhet och mognad var två av de kriterier som vi tagit fram i analysen och som vi anser har stor påverkan på den metod och det verktyg som används. Genom att ha hög mognadsgrad i metoden ökar det användbarheten i verktyget vilket låter användaren arbeta effektivt och ostört.

Två kriterier som vi också vill lägga fokus på är att dokumentation kan skapas i både verktyg och metod för att kunna skapa en så gemensam och ensad bild av hur verksamheten ser ut. Även att det finns stöd för en iterativ arbetsprocess i verktyg och metod då dagens verksamhetsmiljö står under ständig utveckling i ett snabbt tempo.

### 7.1 Studiens relevans och överförbarhet

Denna studie är skriven på uppdrag av Genicore AB för att få MBSD i fokus och är realiserad genom en fallstudie av deras egen metod, Coremetoden. Syftet var att skapa en modell av utbildningsverksamheten på IT-fakulteten på GU och forska i om det gick eller inte med hjälp av Coremetoden. Vårt fokus har varit på att skapa en modell av verksamheten samt att ta fram kriterier för att utvärdera modellbaserade systemutvecklingsmetoder. Kriterierna, eller det ramverk som vi har tagit fram, kan ligga till grund för att appliceras på andra MBSD-metoder. Detta är av relevans då MBSD är ett relativt utforskat område men att det finns framtidsutsikter för det, kanske inte för en hel process av systemutveckling men delar av den. Framförallt kan det få störst relevans inom produktlinjeprogrammering. Studien har även stor relevans för Genicore AB då det för dem är en chans att se ifall individer som inte har tidigare erfarenheter av modellbaserad systemutveckling kan tillämpa deras metod för att skapa en modell av en verksamhet på ett väldefinierat och fungerande vis. Det är även ett steg mot ett nytt verksamhetsområde som metoden kan tillämpas inom vilket innebär en större användbarhet av metoden om den skulle visas vara applicerbar på verksamheten. Detta utökar i sin tur Genicore ABs potentiella intressentmarknad.

## 7.2 Förslag till vidare forskning

I denna studie har vi studerat hur MBSD går att applicera på en utbildningsverksamhet samt tagit fram kriterier för att utvärdera en MBSD-metod. Vidare forskning skulle kunna vara att applicera Coremetoden på fler fakulteter än IT-fakulteten samt att också ta med forskningsverksamheten som pågår på universitetet. Även andra verksamheter som kan ha nytta av ett informationshanteringssystem kan vara intressanta att göra en studie på, som till exempel en myndighet. Förslagsvis finns det också utrymme för att studera modellbaserad och dokumentbaserad systemutveckling och jämföra dem mot varandra. Vad finns det för för- och nackdelar i jämförelse med varandra, och skulle en sammanslagning av dem vara att föredra?

Annan typ av vidare forskning som kan vara relevant kan vara att studera de kriterierna vi kommit fram till och utveckla dem. Det finns även möjligheter till att fördjupa sig i endast en av de inriktningar vi har tagit fram: verktyg eller metod.

## 8. Referenslista

- Alvesson, M & Sköldberg, K (1994) *"Tolkning och reflektion, Vetenskapsfilosofi och kvalitativ metod"*. studentlitteratur, Lund, Sweden
- Amnell, T. Berggren, D. Johansson, G. Mattsson, A. Nilsson, B. Rasmusson, A. Stefan, R. Skoog, M. Tissink, R. Vesterlund, M & Ärlig, U. *"Våra erfarenheter av modellbaserad systemutveckling"*  
[http://www.combitech.se/Documents/Bilder%20och%20filer%20sidor/Utbildning/Filer%20kursprogram/Modellbaserad%20systemutveckling/Om\\_MBSE.pdf](http://www.combitech.se/Documents/Bilder%20och%20filer%20sidor/Utbildning/Filer%20kursprogram/Modellbaserad%20systemutveckling/Om_MBSE.pdf) [2014-03-15]
- Brown, B. (2011) IBM *"Model-based system engineering: Revolution or evolution?"*  
<http://public.dhe.ibm.com/common/ssi/ecm/en/raw14289usen/RAW14289USEN.PDF> [2014-04-28]
- Cadle, J. Paul, D & Turner, P (2010) *"Business Analysis Techniques; 72 Essential Tools for Success"*  
E-artikel från BCS. <http://bcs.org/upload/pdf/business-analysis-techniques.pdf>. [2014-03-12]
- Fey, I. & Stürmer, I. (2007) *"Quality Assurance Methods for Model-based Development: A Survey and Assessment"*. <http://www.ichmaschine.de/Papers/SAE2007/2007-01-0506.pdf> [2014-03-12]
- Giorgi, A. (1985) *"Sketch of a psychological phenomenological method, Phenomenology and psychological research"* (s. 8-22). Pittsburgh: Duquesne University Press.
- Goldkuhl, G. och Röstlinger, A.(1988) *"Förändringsanalys, Arbetsmetodik och förhållningssätt för goda förändringsbeslut."* Lund: Studentlitteratur.
- Hause, M (2002) *"Artisan Real.Time Studio Support For Model Driven Architectura(MDA)(White paper)"*. Portland: Artisan Software Tools.
- Itoh, K. Kumagai, S & Hirota, T. (2005) *"Domain Oriented Systems Development: Perspectives and Practises"* e-library: Taylor & Francis.
- Karlsson, D. (2012) *"Produktlinjer och konfigurabilitet"*.  
[http://fileadmin.cs.lth.se/cs/Personal/Lars\\_Bendix/Teaching/Lund/Theses/Karlsson04/Karlsson04-summary.pdf](http://fileadmin.cs.lth.se/cs/Personal/Lars_Bendix/Teaching/Lund/Theses/Karlsson04/Karlsson04-summary.pdf) [2014-05-10]
- Klein, H. K. & Myers, M. D. (1999) *"A set of principles for conducting and evaluating interpretive field studies in information systems."* MIS quarterly, 67-93.
- Kruchten, P. (2000) *"The Rational Unified Process - an introduction, second edition"*. Reading, USA: Pearson education.

MathWorks (2014) "*Simulink*"(Version 8.3 - R2014a)[Programvara] Tillgänglig:

<http://www.mathworks.se/products/simulink/>

Mathiassen, L. Munk-Madsen, A. Nielsen, P.A. & Stage, J. (2001) "*Objektorienterad analys och design.*" Andra upplagan. Lund: Studentlitteratur.

Mohagheghi, P & Dehlen, V. (2008) "*Developing a Quality Framework for Model-Driven Engineering*". Models in Software Engineering, Lecture Notes in Computer Science Volume 5002, ss. 275-286.

Mylopoulos, J (1998) "*Information modeling in the time of the revolution*" E-artikel.

[http://ac.els-cdn.com/S0306437998000052/1-s2.0-S0306437998000052-main.pdf?\\_tid=695cce6a-a9d1-11e3-96e7-00000aab0f27&acdnat=1394620443\\_c942673db0d60647c67b49465ce9f9b3](http://ac.els-cdn.com/S0306437998000052/1-s2.0-S0306437998000052-main.pdf?_tid=695cce6a-a9d1-11e3-96e7-00000aab0f27&acdnat=1394620443_c942673db0d60647c67b49465ce9f9b3). [2014-03-12]

Nilsson, H. Isaksson, A & Martikainen, T. (2002) "*Företagsvärdering: med fundamental analys.*"

Lund: Studentlitteratur.

No Magic (2013) "*MagicDraw*"(Version 17.0.3)[Programvara]. Tillgänglig:

<http://www.nomagic.com/products/magicdraw.html>

Oldevik, J. Solberg, A. Haugen, Ø & Møller-Pedersen, B. (2006) "*16 Evaluation Framework for Model-Driven Product Line Engineering Tools*".

<https://www.google.se/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CDUQFjAA&url=http%3A%2F%2Fwww.springer.com%2Fcontent%2Fdocument%2Fdocument%2F9783540332527-c16.pdf%3FSGWID%3D0-0-45-309189-p147126310&ei=BYVrU6KjvSCyQODz4GYBA&usg=AFQjCNH7AvPE5A-tmlQI9nD0f1bRmsDKhg&sig2=DPGAR-mPxJXAt8QQDT77wA&bvm=bv.66330100,d.bGQ> [2014-03-27]

Patel, R. & Davidson, B. (2011) "*Forskningsmetodikens grunder.*" Fjärde Upplagan. Lund: Studentlitteratur.

Pawson, R. (2004) "*Naked Objects*". Dublin, University of Dublin, Trinity college.

Umble, E.J & Umble, M.M. (2002) "*Avoiding ERP Implementation Failure*". Industrial Management 44(I): 25-34

Worboys, M. & Duckham, M. (2004) "*GIS in land and property management.*" New York;London: Taylor & Francis.

# 9. Bilagor

## Bilaga 1 Ordlista

Domän – Ett visst område som något syftar till

DOSD - Domain-Oriented System Development

FMV - Försvarsmaktens Materielverk

GPL - General-Purpose Language

GU - Göteborgs Universitet

Informationsmodell – Den modell som information struktureras efter

MBSD - Model Based System Development (Modellbaserad systemutveckling)

MBSE - Model Based System Engineering

MDD - Model Driven Development

MDE - Model Driven Engineering

Metod – Tillvägagångssätt för att gå från en punkt till en annan

NO - Naked Objects

Process – Ett förlopp där något förändras och som hänger samman gentemot andra förlopp

QA - Quality Assurance

SDL - Specification and Description Language

UML- Unified Modeling Language

XML - Extensible Markup Language

## Bilaga 2 Intervju

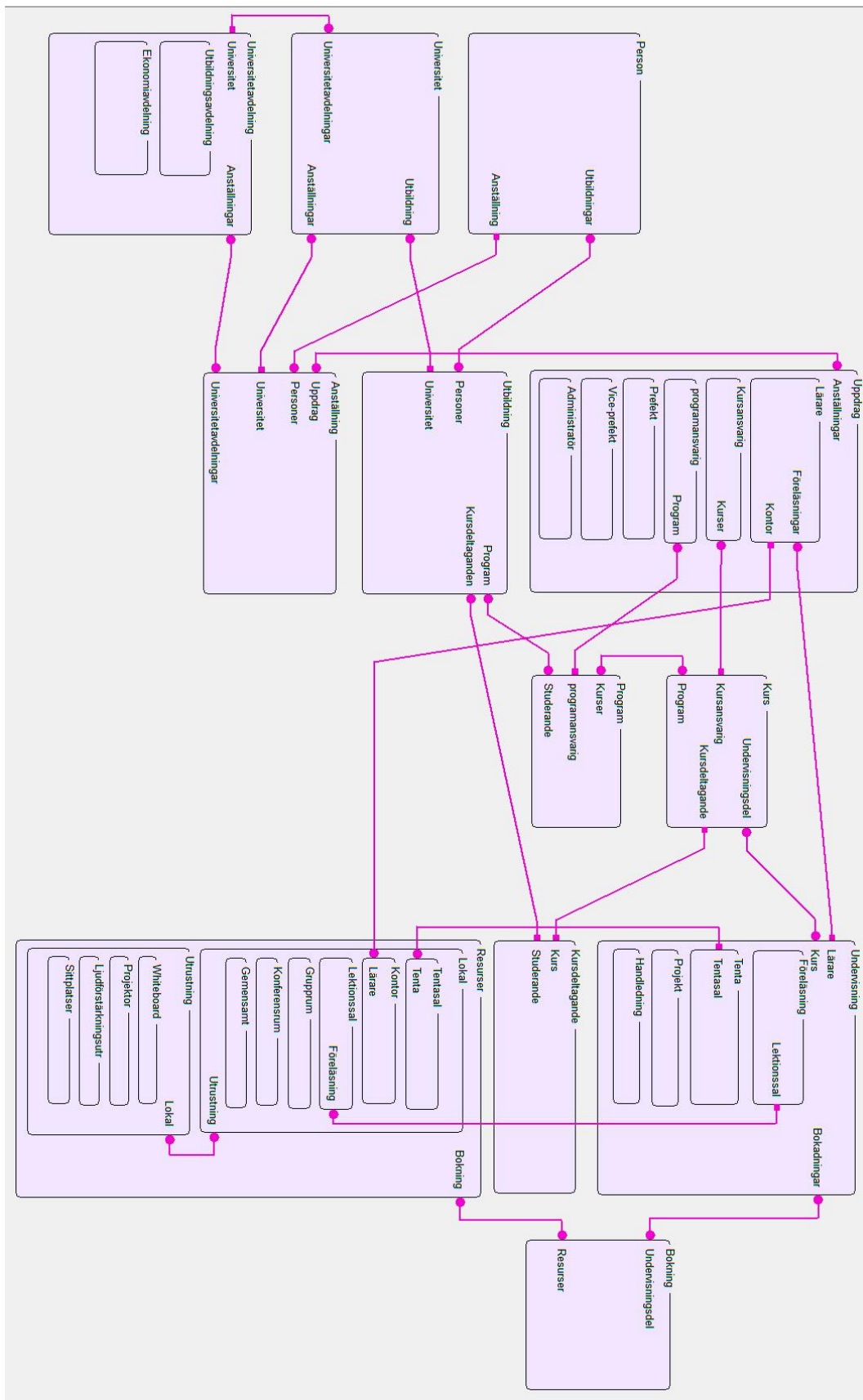
### Intro till intervju:

**Vi har fått i uppgift av Genicore AB att tillämpa deras modelleringsteknik på utbildningsverksamheten på GU. För att lyckas med denna uppgift behöver vi få en djupare förståelse för hur verksamheten är uppbyggd. Då deras modell är anpassad för att bygga informationshanteringssystem kommer informationen som cirkulerar inom verksamheten vara i centrum.**

1. Vad har du för roll på universitetet och vilka delar av verksamheten är du ansvarig för?
2. Vilken information innehåller de dokument som du hanterar?
3. Vilken info är viktig för dig? Vilken info ger du?
4. Vilken information känner du att du saknar?
5. Vilka system använder sig utbildningsverksamheten av?
6. Vad innehåller systemen/et?
7. Vilka andra delar av utbildningsverksamheten är du beroende av?
8. Vilka andra roller/grupper finns? (lärare, elever, administratörer t.ex.)



# Bilaga 3 Modell av utbildningsverksamheten



Figur 1.1 Modell av utbildningsverksamheten vriden 90 grader till höger