

# Verksamhetsmodellering med EBML<sup>1</sup>

Patrik Pettersson

Jaana Wäyrynen

Institutionen för data- och systemvetenskap

Stockholms universitet / Kungliga Tekniska Högskolan

Augusti 2001

## Sammanfattning

BML (Business Modeling Language) är ett meddelandeorienterat processmodelleringsspråk som främst används för att beskriva datorsystem med inriktning på applikationsintegration. Ett problem med BML är att manuella aktiviteter inte kan beskrivas, vilket beror på att processerna som beskrivs måste kunna exekveras. Kartläggningen av manuella aktiviteter är dock viktig i analysfasen av ett systems livscykel då verksamheten ska modelleras och möjliga förändringar tas fram. Således krävs det tillägg till BML för att det ska gå att använda vid verksamhetsmodellering och förändringsanalys.

Syftet med uppsatsen är att presentera en av oss utökad version, EBML (Extended BML). EBML innehåller bland annat nya modelleringssymboler för beskrivandet av manuella aktiviteter. Ett arbetsätt för hur EBML ska kunna användas vid verksamhetsmodellering och förändringsanalys presenteras också. Vidare tillämpas EBML i en verklig fallstudie där verksamheten hos en serviceverkstad modelleras. Slutligen presenteras en utvärdering av EBML och arbetssättet som visar att dessa uppfyller sitt syfte att lösa BML:s begränsningar.

---

<sup>1</sup> Uppsatsen motsvarar 20 poäng för vardera författare.

# Business modelling with EBML<sup>1</sup>

Patrik Pettersson

Jaana Wäyrynen

Department of Computer and Systems Sciences  
University of Stockholm / Royal Institute of Technology  
Augusti 2001

## Abstract

BML (Business Modeling Language) is a message oriented process modelling language mainly used to describe computer systems, with focus on Enterprise Application Integration (EAI). A problem with BML is that manual activities cannot be described, due to the fact the described processes must be able to execute. However, the modelling of manual activities is important in the analysis phase of a systems life cycle when the business and possible changes are to be modelled. Additions to BML are thus needed to enable BML to be used in business modelling and change analysis.

The purpose of this paper is to present an extended version, EBML (Extended BML), developed by the authors. EBML contains, among other things, new symbols for describing manual activities. A method for how EBML is to be used with business modelling and change analysis is also presented. Furthermore, EBML is used in a real case where the business of a workshop is modelled. Finally, an evaluation of EBML and the method is presented, showing that these fulfil their purpose, to solve the limitations of BML.

---

<sup>1</sup> This thesis corresponds to 20 weeks of full-time work for each of the authors.

# Innehållsförteckning

<b>1</b>	<b>INTRODUKTION .....</b>	<b>1</b>
1.1	BAKGRUND .....	1
1.2	PROBLEM .....	2
1.3	SYFTE OCH MÅL .....	3
1.4	UPPDRAG .....	3
1.5	AVGRÄNSNINGAR .....	3
1.6	ARBETSMETODIK .....	4
1.6.1	<i>Datainsamling</i> .....	4
1.6.2	<i>Verksamhetsmodellering</i> .....	4
1.6.3	<i>Analys</i> .....	4
1.6.4	<i>Utveckling av ett arbetssätt för EBML</i> .....	4
1.6.5	<i>Utvärdering</i> .....	5
1.7	UPPSATSENS STRUKTUR .....	5
<b>2</b>	<b>FORSKNING INOM BESLÄKTADE VETENSKAPSOMRÅDEN .....</b>	<b>5</b>
2.1	SYSTEMSPECIFIKATION OCH DESIGN .....	5
2.2	PETRINÄT .....	6
2.3	SPECIFICATION AND DESCRIPTION LANGUAGE (SDL) .....	6
2.4	UNIFIED MODELING LANGUAGE (UML) .....	6
2.5	ENTERPRISE KNOWLEDGE DEVELOPMENT (EKD) .....	7
2.6	ARV AV PROCESSER .....	7
<b>3</b>	<b>BUSINESS MODELING LANGUAGE (BML) .....</b>	<b>7</b>
3.1	GENERELL BESKRIVNING .....	7
3.2	SYMBOLER I BML .....	8
3.2.1	<i>Person</i> .....	8
3.2.2	<i>Applikation</i> .....	8
3.2.3	<i>Process</i> .....	8
3.2.4	<i>Start- och sluttillstånd</i> .....	8
3.2.5	<i>Vänta på händelse</i> .....	9
3.2.6	<i>In- och utmeddelande</i> .....	9
3.2.7	<i>Automatiserad aktivitet</i> .....	9
3.2.8	<i>Automatiserat val</i> .....	10
3.2.9	<i>Timer (start av och utgången)</i> .....	10
3.3	DYNAMISK PROCESSMODELL I BML .....	10
3.4	STATISK PROCESSMODELL I BML .....	11
<b>4</b>	<b>EXTENDED BML (EBML) .....</b>	<b>11</b>
4.1	GENERELL BESKRIVNING .....	12
4.2	SYMBOLER OCH ANDRA NOTATIONSASPEKTER I EBML .....	12
4.2.1	<i>Artificiellt processmeddelande (APM)</i> .....	12
4.2.2	<i>Slumpmässigt starttillstånd</i> .....	13
4.2.3	<i>Manuell aktivitet</i> .....	13
4.2.4	<i>Manuellt val</i> .....	14
4.2.5	<i>Anteckning</i> .....	14
4.2.6	<i>Numrering</i> .....	14
4.2.7	<i>Versionshantering</i> .....	15
4.2.8	<i>Aktörssymboler</i> .....	15
4.2.9	<i>För-varjemönster</i> .....	15
4.3	DYNAMISK PROCESSMODELL I EBML .....	16
4.4	STATISK PROCESSMODELL I EBML .....	16
4.5	ARVSHIERARKISKA MODELLER I EBML .....	17
4.5.1	<i>Arv</i> .....	17
4.5.2	<i>Abstrakthet</i> .....	17
4.5.3	<i>Ersättning</i> .....	18
4.5.4	<i>Processhierarki</i> .....	18
4.5.5	<i>Processmodell med arv</i> .....	18

<b>5</b>	<b>ATT ARBETA MED EBML.....</b>	<b>19</b>
5.1	BAKGRUNDEN TILL FÖRESLAGET ARBETSSÄTT .....	19
5.2	VERKSAMHETSMODELLERING MED EBML.....	20
5.2.1	<i>Steg 1: Skapa en konceptuell modell.....</i>	<i>20</i>
5.2.2	<i>Steg 2: Modellera verksamhet med dynamiska processmodeller i EBML.....</i>	<i>21</i>
5.2.3	<i>Steg 3: Skapa en rollmatris.....</i>	<i>23</i>
5.2.4	<i>Steg 4: Modellera verksamhet med statisk processmodell i EBML.....</i>	<i>24</i>
5.2.5	<i>Steg 5: Skapa arvshierarkiska modeller i EBML.....</i>	<i>25</i>
5.3	FÖRÄNDRINGSANALYS MED EBML.....	26
5.3.1	<i>Steg 1: Skapa en målmodell.....</i>	<i>26</i>
5.3.2	<i>Steg 2: Analysera befintliga verksamhetsmodeller.....</i>	<i>27</i>
5.3.3	<i>Steg 3: Modellera de förändrade processerna i EBML.....</i>	<i>28</i>
5.3.4	<i>Steg 4: Uppdatera övriga verksamhetsmodeller.....</i>	<i>29</i>
<b>6</b>	<b>TILLÄMPNING AV EBML – EN FALLSTUDIE .....</b>	<b>29</b>
6.1	BESKRIVNING AV STOCKHOLMS CENTRAL SERVICE AB.....	30
6.2	EXEMPLIFIERING AV VERKSAMHETSMODELLER.....	31
6.2.1	<i>Tillämpning av en konceptuell modell.....</i>	<i>31</i>
6.2.2	<i>Tillämpning av processmodellering i EBML.....</i>	<i>32</i>
6.2.3	<i>Tillämpning av en rollmatris.....</i>	<i>35</i>
6.2.4	<i>Tillämpning av en statisk processmodell.....</i>	<i>36</i>
6.2.5	<i>Tillämpning av arvshierarki.....</i>	<i>38</i>
6.2.6	<i>Tillämpning av en målmodell.....</i>	<i>39</i>
6.2.7	<i>Tillämpning av förändringsanalys.....</i>	<i>41</i>
6.2.8	<i>Tillämpning av modellering av förändrade processer.....</i>	<i>41</i>
<b>7</b>	<b>UTVÄRDERING .....</b>	<b>44</b>
7.1	ANLEDNING OCH UTFÖRANDE.....	44
7.2	PROCESSMODELLERINGSSPRÅKET EBML.....	45
7.2.1	<i>Dynamisk processmodell.....</i>	<i>45</i>
7.2.2	<i>Statisk processmodell.....</i>	<i>45</i>
7.2.3	<i>Arvshierarkisk modell.....</i>	<i>46</i>
7.3	ARBETSSÄTT FÖR EBML.....	46
7.3.1	<i>Nulägesbeskrivning.....</i>	<i>46</i>
7.3.2	<i>Förändringsanalys.....</i>	<i>47</i>
7.4	DISKUSSION.....	48
<b>8</b>	<b>SLUTSATSER .....</b>	<b>49</b>
<b>9</b>	<b>FRAMTIDA FORSKNING .....</b>	<b>49</b>
<b>10</b>	<b>TACK.....</b>	<b>50</b>
<b>11</b>	<b>REFERENSER OCH KÄLLFÖRTECKNING .....</b>	<b>50</b>

## APPENDIX 1. GRUNDLÄGGANDE BEGREPP

## APPENDIX 2. UPPLÄGG FÖR UTVÄRDERINGSFRÅGOR

# 1 Introduktion

## 1.1 Bakgrund

Utvecklingen av IT-system som underlättar kommunikation och samarbete över avdelningsgränser, affärsenheter och företagsgränser tillför nya förutsättningar för tjugohundratalets företag. Genom integration av IT-system kan företag snabbare och mer flexibelt reagera på förändringar i kunders efterfrågan, automatisera manuella rutiner och samverka med leverantörer och kunder elektroniskt via så kallade marknadsplatser eller extranet. Exempel på teknologier som underlättar integration är CORBA, DCOM, Enterprise Java Beans (EJB), XML, meddelandemäklare (eng. Message Broker) och processmäklare (eng. Process Broker). De brukar med ett gemensamt namn kallas EAI-teknologier (EAI står för Enterprise Application Integration).

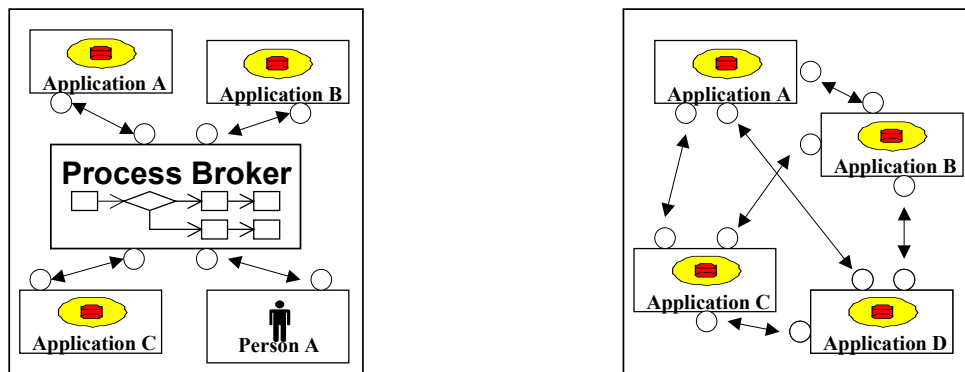
EAI-teknologier underlättar för en organisation att frångå en funktionsindelad och mer traditionell verksamhetsstruktur och i stället anta ett processororienterat förhållningssätt. Det betyder att avdelnings- och funktionsgränser suddas ut och affärsprocesserna som löper över avdelningsgränser blir centrala. En affärsprocess är en samling av organisationsaktiviteter som tillsammans skapar ett värde för kund (Davenport, 2000). Med andra ord, genom processororienteringen hamnar kunden och dess behov i fokus på ett helt annat sätt än tidigare. Ur ett tekniskt perspektiv innebär EAI att de individuella avdelningarnas IT-system lättare kan överföra information mellan varandra och att denna överföring också lättare kan automatiseras.

Med integration av IT-system växer också behovet av metodologier som stödjer och underlättar själva integrationen. En viktig del av en sådan metodologi är ett processmodelleringsspråk, som används för att kunna konstruera modeller som specificerar affärs- och informationsprocesser (Johannesson & Perjons, 2000).

BML är ett nytt modelleringsspråk som står för Business Modeling Language. Språket är ett meddelandeorienterat processmodelleringsspråk som används både för att beskriva och exekvera affärs- och informationsprocesser (Johannesson & Perjons, 2000). Språket är utvecklat av ViewLocity, som använder det i TradeSync Process Manager, en produkt som bygger på så kallad processmäklarteknologi. En processmäklare är ett mellanprogram (eng. middleware) som underlättar flexibel kommunikation mellan olika applikationer<sup>1</sup> genom att för det första samla integrationslogiken mellan applikationer centralt på ett ställe (se Figur 1). Därmed kan en användare lättare förändra vilka applikationer som ska integreras, i vilken ordning de ska kommunicera och vilka typer av meddelanden som ska skickas. För det andra möjliggör processmäklaren konvertering av meddelanden från ett dataformat till ett annat. Det gör det möjligt för en applikation att skicka ett meddelande till en annan applikation, fastän den mottagande applikationen kräver att meddelandet ska ha ett helt annat dataformat och struktur. Processmäklaren i ViewLocitys produkt består av verktyg för modellering och design av processer, verktyg för konvertering av olika dataformat och verktyg för simulering och exekvering av de specificerade processerna (ViewLocity, 2001).

---

<sup>1</sup> Termerna applikation och IT-system används synonymt i uppsatsen.



Figur 1. Användandet av en processmäklare (till vänster) vid applikationsintegration minskar kopplingarna mellan programmen och samlar all processlogik centralt på ett ställe. Jämför med punkt-till punkt integrationen (till höger).

BML beskriver hur kommunikation i form av sändande och mottagande av meddelanden flödar mellan aktörer, som människor och IT-system. BML lämpar sig därför väl som grund för specifikation, exekvering och automatisering av processer i en organisation.

BML är utformat så att intressenter med olika kunskapsnivå ska kunna förstå BML-modellerna. Verksamhetsansvariga, utan teknisk kunskap, men med kunskap om affärslogiken, ska kunna delta i designen av processer, tekniska designers ska kunna validera och implementera de framtagna BML-modellerna och operatörer ska kunna övervaka och analysera exekveringen av BML-modellerna samt kunna anpassa processmodellerna så att exekveringens prestanda ytterligare förbättras (ViewLocity, 2001).

Det medför att möjligheter öppnas att använda BML genom hela systemutvecklingsprocessen. BML måste då kunna användas i tidiga faser av ett systems livscykel, som till exempel analysfasen. Eventuellt skulle BML också kunna användas som grund för att implementera system även i fall när en processmäklare inte är tänkt att tillämpas. BML-modellerna skulle då kunna ges till tekniska designers som utifrån dem bygger system med hjälp av programmeringsspråk som Java eller C++.

## 1.2 Problem

Ett problem med BML i verksamhetsanalys är dock att det endast beskriver interaktionen i form av meddelanden som skickas mellan olika applikationer och mellan applikationer och människor, medan manuella aktiviteter inte kan beskrivas. Skälet till det är att de specificerade processmodellerna i BML måste kunna exekveras. För att använda BML i analysfasen, där kartläggning av även manuella aktiviteter är viktig, måste BML kompletteras med modelleringssymboler som beskriver manuella rutiner. Kombinationen av ett meddelandeorienterat och ett sådant aktivitetsorienterat processmodelleringsspråk öppnar bland annat möjligheter att kartlägga vilka manuella aktiviteter som skulle kunna förändras och automatiseras. BML skulle därmed kunna utökas till att användas även för verksamhetsmodellering i samband med förändringsanalys.

Förändringsanalys är emellertid en komplicerad procedur och måste innefatta information om hur organisationen ser ut och verkar i aktuell situation, vilka målsättningar och framtida tillstånd som organisationen eftersträvar och hur organisationen avser att uppnå de uppsatta målen (Bubenko, 1998).

Även i det här avseendet är problemet med BML i verksamhetsanalys att språket begränsar sig till kartläggningen av informationsflöden i en organisation och mellan dess system.

Språket saknar instrument för att kartlägga och strukturera information som till exempel vilka mål en organisation har och eftersträvar. Att komplettera BML med en målmodell skulle generera bättre förutsättningar för välgrundade resonemang kring möjliga förändringar i analysfasen.

En möjlig lösning till påvisade problem är att utveckla en utökad version av BML, Extended BML (EBML), som kan tillämpas även i verksamhetsmodellering i analysfasen. I samband med utvecklingen av EBML bör också ett arbetssätt eller metod för hur språket ska användas i analysfasen utvecklas eftersom EBML är helt nytt och ingen formell arbetsmetod existerar.

Genom att använda ett och samma modelleringspråk, genom hela utvecklingsprocessen skulle kommunikationen mellan en organisations intressenter, som verksamhetsansvariga, tekniska designers och operatörer, förbättras radikalt. Verksamhetsansvariga skulle också lättare kunna styra utvecklingen så att de får system de önskar sig. Affärslogiken skulle med andra ord styra IT-systemen och inte tvärtom.

### **1.3 Syfte och mål**

Syftet med den här uppsatsen är att utveckla en utökad version av BML så att det meddelandeorienterade processmodelleringspråket ska kunna användas för verksamhetsmodellering och analys i ett systemutvecklingsprojekt.

Målet är att presentera en utökad version av BML, EBML, presentera ett arbetssätt för hur EBML ska kunna användas i verksamhetsmodellering, tillämpa EBML i en verklig fallstudie, samt att göra en utvärdering av det presenterade EBML.

### **1.4 Uppdrag**

Uppsatsen är resultat av ett uppdrag som formulerats dels av IT-företaget BlueFront AB, dels av Institutionen för data- och systemvetenskap (DSV) vid Stockholms Universitet/KTH. BlueFront AB är ett företag inom mjukvarubranschen som tillhandahåller helhetslösningar för företag främst inom fackhandeln. Uppdraget bestod i att göra en verksamhetsanalys av en serviceverkstad som var i behov av ett nytt IT-system. Vi skulle kartlägga användningen av det befintliga systemet, samt dokumentera informationsflöden, verksamhetsprocesser och rutiner som möjligen skulle kunna förbättras eller automatiseras i ett nytt system. Uppdraget från DSV var att undersöka i vilken utsträckning BML kunde användas vid den här typen av verksamhetsmodellering.

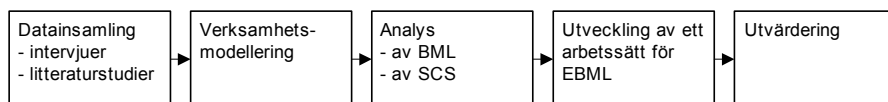
### **1.5 Avgränsningar**

Vår utgångspunkt är att använda BML för processinriktad modellering av en verksamhet. Det exkluderar andra typer av processmodelleringspråk eller närbesläktade modelleringspråk som kan användas för den här typen av ändamål, såsom UML och rutinskisser. Avsikten är att öka kunskapen om BML och möjligheten att använda BML i andra avseenden än enbart design och specifikation av automatiserade processer. Vi begränsar oss till att lösa de begränsningar som BML uppvisat i samband med verksamhetsmodellering och förändringsanalys, vilket också exkluderar undersökning av användning av BML i andra faser av ett systems livscykel.

## 1.6 Arbetsmetodik

Vår arbetsmetodik kan delas in i ett antal distinkta delfaser: förarbete, datainsamling, verksamhetsmodellering, analys, utveckling av EBML och ett arbetsätt för EBML, utvärdering av EBML och det framtagna arbetssättet samt slutsatser.

Figur 2 visar en modell som sammanfattar vår metodik. Faserna kan förefalla ske sekventiellt, men arbetsgången är genomgående iterativ eftersom arbetet i respektive fas i flera avseenden påverkade varandra. Det medförde att vi utförde flera faser parallellt i det fortlöpande arbetet.



Figur 2. En modell av vår arbetsmetodik.

### 1.6.1 Datainsamling

Datainsamlingen bestod i huvudsak av två olika arbetsformer, varav den första innefattade intervjuer och besök på Stockholms Central Service AB för att få ett empiriskt underlag till själva verksamhetsmodelleringen och analysen. Arbetet var av konsultkaraktär, då vi intervjuade verksamhetspersonal på varje nivå från kärnverksamhet upp till ledningsnivå.

Den andra formen av datainsamling byggde på litteratursökning och studier av relevant vetenskaplig litteratur för att bredda och fördjupa vår kunskap om verksamhetsmodellering, processmodellering och formella modelleringsspråk. Båda formerna av datainsamling gjordes fortlöpande och parallellt med annat arbete under uppsatsen gång.

### 1.6.2 Verksamhetsmodellering

Vår arbetsmetod baserades på ett modellcentrerat arbete. Genom att modellera en verksamhet, struktureras kunskapen om den och därmed ökas förståelsen för den (Bubenko, 1998). Vi började med att göra en konceptuell modell av verksamheten som senare följdes av en rollmatris och målmodell. I samband med att kunskapen om verksamheten på verkstaden ökade övergick arbetet till identifiering av affärsprocesser och fokus på modellering av dessa med hjälp av BML. Processmodelleringen skedde iterativt i samband med att informationen om serviceverkstaden med tiden uppdaterades genom nya besök. Under modelleringsarbetet blev också problemen med att använda BML för verksamhetsmodellering uppenbara, vilket gav upphov till analys av språket och de första försöken till att skapa en utökningsmekanism, EBML.

### 1.6.3 Analys

I analysen utgick vi från de modeller vi skapat i föregående modelleringsfas för att på så sätt undersöka i vilken utsträckning BML som modelleringsspråk kunde användas för att illustrera den befintliga verksamheten. Det gav oss möjlighet att experimentera med BML-modellerna och bedöma tilläggs relevans och användbarhet och till fullo utveckla EBML. Utifrån EBML-modellerna, rollmatrisen och målmodellen kunde vi sedan identifiera processer och rutiner i organisationen som kunde förbättras eller automatiseras.

### 1.6.4 Utveckling av ett arbetsätt för EBML

Föregående faser gav oss relativt goda insikter i användningen av ett modellcentrerat arbetsätt med hjälp av BML och EBML. Vår tanke var att nyttja den metodik som vi själva



använt oss av, kombinerat med rådande teorier om metoder vid verksamhetsmodellering, för att utforma ett förslag till ett arbetssätt för EBML.

### **1.6.5 Utvärdering**

Utvärderingen av EBML och det arbetssätt som vi föreslagit, genomfördes i form av strukturerade intervjuer om EBML:s styrkor och svagheter. Intervjuerna utfördes på personer från serviceverkstaden och systemutvecklare från BlueFront AB.

## **1.7 Uppsatsens struktur**

Uppsatsen består av nio kapitel, varav detta kapitel introducerar ämnet. Nästa kapitel innehåller hänvisningar till forskning och teori som på olika sätt är besläktat med arbetet och som påverkat utformningen av vårt arbete. Kapitel tre och fyra definierar och presenterar BML respektive EBML. Kapitlen fokuserar i första hand på hur språken är uppbyggda formellt och hur syntaxen ska tillämpas. Kapitel fem är indelat i tre avsnitt. Det första beskriver bakgrunden till hur vi utvecklat arbetssättet, medan de andra två avsnitten steg för steg beskriver arbetssättet för hur EBML kan tillämpas i verksamhetsmodellering respektive förändringsanalys. I kapitel sex visas exempel på hur EBML tillämpats i en verklig verksamhetsanalys av en serviceverkstad. I kapitel sju redovisas resultatet av en utvärdering av EBML som gjorts genom strukturerade intervjuer med systemutvecklare och anställda på den serviceverkstad som utgjorde fallstudie i uppsatsen. Slutsatser och en avslutande diskussion återfinns i kapitel åtta och i kapitel nio ges förslag på vidare forskning. Som bilagor finns definitioner på grundläggande begrepp som används i uppsatsen samt upplägg för utvärderingsfrågor.

## **2 Forskning inom besläktade vetenskapsområden**

I det här kapitlet presenterar vi teori och forskning inom besläktade vetenskapsområden. Kapitlet är upplagt så att de första fyra avsnitten presenterar besläktade teorier. Presentationen av dessa teorier syftar till att ge en bredare bakgrund och kan läsas kursivt. Teorierna som presenteras i de två sista avsnitten är däremot av större betydelse eftersom de ligger till grund för vårt eget arbete och kommer att återkopplas till resonemang i senare kapitel.

### **2.1 Systemspecifikation och design**

Att nyckeln till ett framgångsfullt systemutvecklingsprojekt ligger i en noggrann och väl genomtänkt systemspecifikation och design, är ett faktum som idag är vida accepterat. En specifikation av ett system beskriver ett system i termer av dess statiska struktur och dynamiska beteende. Den ska ge en god översikt av ett komplext system, vilket också är fördelaktigt i jämförelse med en beskrivning gjord i ett programmeringsspråk. När ett system specificeras med hjälp av ett specifikationsspråk används oftast modeller, som representerar viktiga egenskaper i systemet. Tanken med ett specifikationspråk är vidare att det ska innehålla en väl definierad uppsättning av begrepp och generera otvetydiga och tydliga specifikationer, som ska utgöra grund för analys. (Belina, 1991)

Användningen av ett specifikationsspråk gör det också möjligt att analysera och simulera alternativa systemlösningar vilket i praktiken är svårt när enbart ett rent programmeringsspråk används. Rätt använt underlättar ett specifikationsspråk en diskussion kring lösningsförslag och möjligheten att producera en lösning på problemet. Ett specifikationsspråk bör även kunna ange hur implementationen ska kunna anpassas till specifikationen och hur dator-

baserade verktyg ska utnyttjas för att skapa, underhålla, analysera och simulera specifikationer. (Belina, 1991)

## **2.2 Petri nät**

Ett flertal språk har utvecklats för modellering av system. Många processmodelleringspråk har sin grund i Petri nät, som utvecklades redan under 1960-talet. Petri nät utvecklades med avsikten att skapa en konceptuell och teoretisk bas för att på ett enhetligt och exakt sätt beskriva alla fenomen som generellt kunde kopplas till informationsöverföring och informationstransformation. Språket är i dessa avseenden lämpat för specifikation och design av datorbaserade informationssystem. Petri nätens styrka kan också i huvudsak tillskrivas det faktum att språket både har en grafisk representation och en väl definierad semantik som tillåter en formell analys. (Jensen, 1992 och Reisig, 1985)

Flera varianter av Petri nät existerar. Det finns alltid en motsättning mellan modelleringskraft och beslutskraft, där modelleringskraft innebär uttrycksfullhet och beslutskraft innefattar möjligheten att algoritmiskt bestämma en modells egenskaper. Ju större modelleringskraft, desto mindre beslutskraft och tvärtom. Stor modelleringskraft innebär att det går att modellera många saker. Stor beslutskraft innebär möjligheter att formellt granska modellerna för att upptäcka formella fel i modellerna. En typ av Petri nät som har mindre modelleringskraft men större beslutskraft än klassiska Petri nät är så kallade ändliga tillståndsmaskiner (eng. Finite State Machines), som kännetecknas av ett input och ett output till varje begrepp eller symbol. Klassiska Petri nät tillåter flera input och output till varje begrepp eller symbol. Ändliga tillståndsmaskiner kan dock ges större modelleringskraft genom utökningsmekanismer. Ett språk med en sådan utökad tillståndsmaskin är SDL. (Peterson, 1977)

## **2.3 Specification and Description Language (SDL)**

SDL (Specification and Description Language) är ett standardiserat processmodelleringspråk för specifikation och beskrivning av system, som till skillnad från traditionella tillståndsmaskiner också kan innehålla variabler och timers (Svantesson, 2000). SDL används idag främst inom telekommunikation och datakommunikation, men kan även tillämpas i realtidssystem, interaktiva system och distribuerade system (Belina, 1991).

SDL har utvecklats för specifikation av ett systems beteende och struktur, vilket betyder att språket specificerar interaktionen mellan systemet och dess miljö men också ett systems interna struktur. Abstraktionsnivån i SDL är flexibel och specifikationen kan variera från att vara en översikt till att vara en specifikation på instansnivå. Språket representerar viktiga egenskaper hos ett existerande eller påtänkt system, framför allt i form av egenskaper som innefattar beteende, det vill säga vad systemet gör och under vilka förhållanden. (Belina, 1991)

## **2.4 Unified Modeling Language (UML)**

Ett annat standardiserat modelleringspråk är Unified Modeling Language (UML), som används för att specificera, visualisera, konstruera och dokumentera ett system. UML är egentligen en samling av statiska och dynamiska diagram. UML aktivitetsdiagram och UML tillståndsdigram, är båda exempel på dynamiska diagram, som båda har sin grund i Petri nät. UML har, trots sin till synes kompletta verktygslåda av olika modeller, begränsningar. Användare kommer därför att vilja lägga till funktionalitet. Utvecklarna bakom UML har varit medvetna om detta och därför skapat en utökningsmekanism. Genom att erbjuda denna hoppas de att egenmäktiga och radikala tillägg i UML undviks, vilket skulle försvåra UML:s

mål att fungera som en standard. Upphovsmännen betonar emellertid att utökningsmekanismen bör användas med varsamhet, eftersom risken för att producera en egen dialekt fortfarande föreligger. (Fowler, 1997 och Rumbaugh, 1999)

## **2.5 Enterprise Knowledge Development (EKD)**

Enterprise Knowledge Development (EKD) är en ansats som utnyttjar olika verksamhetsmodeller för att systematiskt analysera, förstå, utveckla och dokumentera en verksamhet och dess komponenter. Syftet med EKD är att ge en tydlig bild av hur verksamheten fungerar i given situation, vilka anledningarna och kraven är för förändring, vilka alternativ som kan föreslås för att möta förändringskraven samt vilka kriterier och argument som kan användas för att utvärdera de föreslagna alternativen. (Bubenko, 1998)

Tillämpning av EKD kommer att resultera i en enhetlig verksamhetsbeskrivning som består av ett antal submodeller som var och en representerar en viss aspekt av verksamheten, men som påverkar varandra. EKD kommer att förenkla kommunikation mellan deltagande intressenter eftersom de kommer att tala samma språk. (Bubenko, 1998)

## **2.6 Arv av processer**

Processmodellering innebär också att processerna i en given situation måste representeras på ett sätt som underlättar förståelsen för dem. För att kunna föreslå förbättringar för en process i en given situation måste representationen vidare ge möjlighet för användaren att kunna se och föreslå alternativa processmodeller. Dessa krav medför att processer bör representeras på ett sätt som explicit illustrerar likheter och skillnader mellan en samling av relaterade processer. (Malone, 1999)

Många processmodelleringstekniker innebär att processer presenteras endimensionellt, det vill säga att processen delas upp i olika delar, representerade av olika entiteter eller aktiviteter. En tvådimensionell representation visar även processer i form av generalisering och specialisering. Genom att använda tanken bakom specialisering, som baseras på idéer om arv i den objektorienterade programmeringsmetodiken, går det att representera processer på olika abstraktionsnivå. Det gör att man explicit kan visa likheter och skillnader mellan processer. (Malone, 1999)

## **3 Business Modeling Language (BML)**

I det här kapitlet presenteras BML – Business Modeling Language. Kapitlet är uppdelat i fyra huvudavsnitt. Först ges en generell beskrivning av språket i avsnitt 3.1, därefter presenteras symbolerna i språket i avsnitt 3.2. De två huvudtyperna av BML-modeller presenteras sedan kort - de dynamiska i avsnitt 3.3 och de statiska i avsnitt 3.4. Kapitlet baseras i huvudsak helt på Johannesson & Perjons (2000), i annat fall anges källan.

### **3.1 Generell beskrivning**

BML är ett kommunikationsorienterat processmodelleringsspråk som utvecklats av företaget ViewLocity. Språket används för att beskriva interaktionen mellan applikationer och mellan applikationer och människor, men även för att beskriva datorsystem. Det är alltså möjligt att med BML beskriva ett datorsystems funktionalitet. Språket liknar SDL men syntaxen är anpassat för processmäklare och applikationsintegration. Det centrala i BML är därför skickandet och mottagandet av meddelanden.

ViewLocity använder språket i sin egen produkt TradeSync Process Manager. I denna går det att generera kompillerbar kod från BML-modeller, vilket gör det möjligt att exekvera modellerna, eller mera korrekt, instanser av modellerna (ViewLocity, 2001).

BML har en egen datamodell med vilken det bland annat går att beskriva exakt vad meddelandena som skickas innehåller för data. Vi har valt att inte ta upp den då en sådan detaljnivå inte är nödvändig för den typ av verksamhetsmodellering som uppsatsen fokuserar på. Det finns även olika klassificeringar av processer och meddelanden som inte kommer att behandlas här. Orsaken till det är att klassificeringen av processerna är specifika för applikationsintegration och klassificeringen av meddelandena, som är baserade på talaktsteori, är antingen specifika för applikationsintegration eller för abstrakta och krångliga att använda vid bland annat verksamhetsmodellering. BML:s meddelandeköer kommer inte heller att beskrivas.

## 3.2 Symboler i BML

BML innehåller symboler för de entiteter som kan skicka och ta emot meddelanden, för start-, stopp- och vänttillstånd, för in- och utmeddelande, för val, aktiviteter och tidsbegränsningar. Dessa presenteras nedan.

### 3.2.1 Person

Person (se Figur 3) är en av de entiteter som är kapabel att skicka och ta emot meddelanden.



Figur 3. BML-symbolen för en person.

### 3.2.2 Applikation

Applikation (se Figur 4) är en annan entitet som är kapabel att skicka och ta emot meddelanden.



Figur 4. BML-symbolen för en applikation.

### 3.2.3 Process

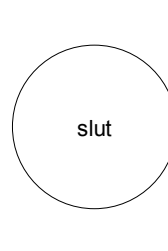
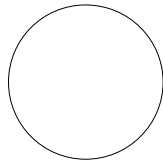
Process (se Figur 5) är den sista typen av entitet som kan skicka och ta emot meddelanden.



Figur 5. BML-symbolen för en process.

### 3.2.4 Start- och sluttillstånd

En process börjar alltid med ett starttillstånd (se Figur 6) och följs alltid av ett inmeddelade (se 3.2.6). Starttillståndet illustreras med en tom cirkel och är ett specialfall av Vänta på händelse (se 3.2.5). En process avslutas alltid med ett sluttillstånd (se Figur 7).



Figur 6. BML-symbolen för ett starttillstånd.    Figur 7. BML-symbolen för ett sluttillstånd.

### 3.2.5 Vänta på händelse

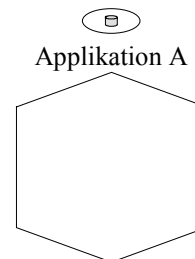
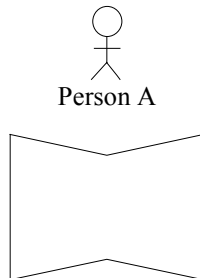
Vänta på händelse-symbolen indikerar att processinstansen står och väntar, antingen tills ett inmeddelande mottages eller att en timer (se 3.2.9) går ut.



Figur 8. BML-symbolen för ett tillstånd av typen Vänta på händelse.

### 3.2.6 In- och utmeddelande

Ett inmeddelande mottages alltid från antingen en person, applikation eller process. På samma sätt sänds utmeddelande alltid till någon av dessa entiteter. I båda fallen visas sändaren eller mottagaren ovanför meddelandesymbolen (se Figur 9 och Figur 10). Ett utmeddelande följs ofta av en Vänta på händelse och därefter av ett inmeddelande.



Figur 9. BML-symbolen för ett inmeddelande inklusive avsändaren, i det här fallet en person.    Figur 10. BML-symbolen för ett utmeddelande inklusive mottagaren, i det här fallet en applikation.

### 3.2.7 Automatiserad aktivitet

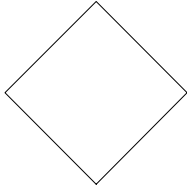
En automatiserad aktivitet (se Figur 11) beskriver operationer som ska utföras på data som följt med ett inmeddelande. Denna symbol behandlas inte närmare eftersom BML:s data-modell inte beskrivs närmare.



Figur 11. BML-symbolen för en automatiserad aktivitet.

### 3.2.8 Automatiserat val

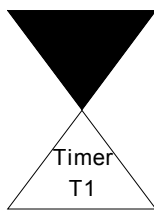
Från ett automatiserat val (se Figur 12) kan flödet gå till en av flera beslutsvägar. Vilken riktning det blir bestäms av verksamhetens affärsregler. Till exempel kan en orderbekräftelse antingen kan vara positiv eller negativ beroende på en affärsregel, som att en order måste överstiga en viss summa för att expedieras.



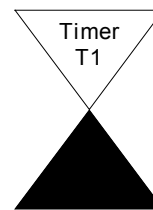
Figur 12. BML-symbolen för ett automatiserat val.

### 3.2.9 Timer (start av och utgången)

I applikationsintegration är tidsaspekten viktigt och timers används ofta för synkronisering. En startad timer (se Figur 13) ges ett tidsintervall. Om tiden gått ut utan att önskad händelse inträffat, aktiveras det flöde där den utgångna timern (se Figur 14) finns och felhantering tar vid.



Figur 13. BML-symbolen för start av timer.

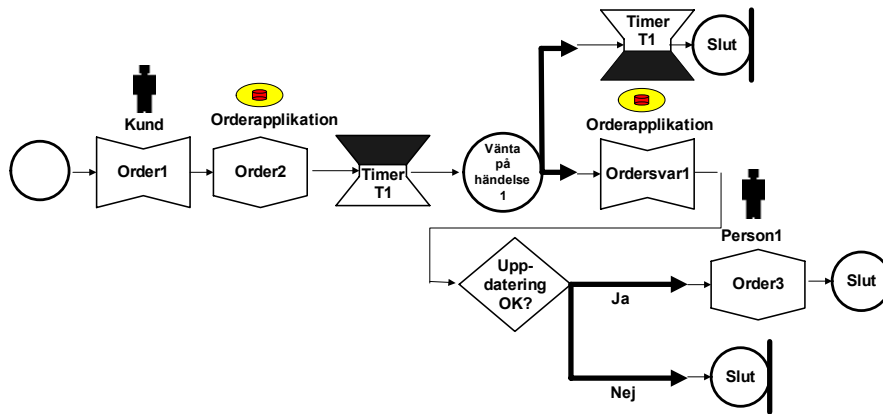


Figur 14. BML-symbolen för en utgången timer.

## 3.3 Dynamisk processmodell i BML

De dynamiska processmodellerna beskriver ett systems beteende, det vill säga i vilken ordning processer skickar och tar emot meddelanden. Dessa modeller kan ses som processmallar eller typer. För varje sådan mall finns det ett antal instanser, som kan vara igång samtidigt.

Processmallen i Figur 15 initieras och instantieras av att den tar emot ett meddelande; Order1, från en Kund. Ett utmeddelande, Order2, skickas till en applikation, Orderapplikation, som uppdateras med information om den nya ordern. Efter det startas en timer, Timer T1, innan processinstansen hamnar i ett väntande tillstånd, Vänta på händelse1. En processinstans befinner sig antingen i ett väntande tillstånd, vilket visas med en Vänta på händelse-symbol, eller i transition på väg från ett väntande tillstånd till ett annat. Det som startar en transition är antingen ett inmeddelande eller att tiden hos en startad timer går ut. I exemplet i Figur 15 går antingen Timer T1 ut eller så inkommer ett meddelande, Ordersvar1, från Orderapplikation, med information om huruvida uppdateringen gått bra eller inte. Har uppdateringen gått bra, vilket testas i ett automatiserat val, skickas ett nytt meddelande, Order3, till en person, Person1, för att informera att en order inkommit. Om uppdateringen däremot inte gått bra eller att Timer T1 gått ut innan Ordersvar1 inkommit, stoppas instansen.

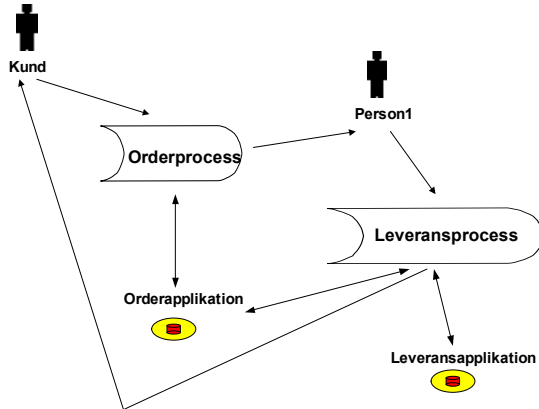


Figur 15. Ett exempel på en dynamisk processmodell<sup>1</sup>.

### 3.4 Statisk processmodell i BML

Statiska processmodeller beskriver ett systems struktur. De beskriver hur flera olika processer interagerar med sin miljö, det vill säga med personer och applikationer samt andra processer (se Figur 16). Processernas kontrollflöde, till exempel i vilken logiska ordning meddelanden skickas, visas dock inte i denna modell, till skillnad från BML:s dynamiska modell, beskriven i avsnitt 3.3. Det är fullt möjligt att från flera dynamiska processmodeller skapa en statisk processmodell. Det går däremot inte att göra tvärtom.

Pilar mellan personer, applikationer och processer representerar meddelanden som skickas. En dubbelriktad pil används vid dubbelriktad kommunikation. Den dynamiska processen som beskrivs i Figur 15 motsvaras av Orderprocessen i Figur 16.



Figur 16. Ett exempel på en statisk processmodell i BML.

## 4 Extended BML (EBML)

I det här kapitlet presenteras EBML – Extended BML. Kapitlet är uppdelat i fem huvudavsnitt. Först ges en generell beskrivning av språket i avsnitt 4.1, vilken följs av en symbolgenomgång i avsnitt 4.2. Därefter kommer de modifierade dynamiska och statiska processmodellerna att presenteras, i avsnitt 4.3 respektive 4.4. Slutligen kommer EBML:s arvshierarki att presenteras i avsnitt 4.5. Både den statiska processmodellen och arvshierarkin

<sup>1</sup> På grund av det modelleringsverktyg vi valt skiljer sig de symboler vi använder för personer och timers mot de som används av Johannesson & Perjons (2000). De är dock semantiskt likvärdiga.

innehåller nya symboler. Dessa presenteras integrerade med respektive underkapitel, det vill säga avsnitt 4.4 och 4.5, och tas inte med i den första symbolgenomgången.

## **4.1 Generell beskrivning**

EBML är en utökning av BML som vi utvecklat för modellering av mänskliga rutiner, vilket gör det möjligt att använda EBML vid verksamhetsmodellering. Eftersom EBML är meddelandeorienterat faller det sig naturligt att se processerna modellerade ur en viss aktörs perspektiv. Det är aktörer som skickar och tar emot meddelanden och utför andra aktiviteter, inte abstrakta entiteter. Därför har samtliga dynamiska processmodeller i EBML en så kallad huvudaktör, ur vilkens perspektiv processen modelleras. Denne huvudaktör kan vara en människa eller en applikation.

Resultatet av denna ansats är att de dynamiska processmodellerna konkret beskriver vilka aktiviteter varje aktör utför i verksamheten. Detta gör processmodellerna enklare att förstå för de människor som utför aktiviteterna i verksamheten, vilket i sin tur förenklar kommunikationen mellan dem och verksamhets- och systemutvecklare.

Nytt i EBML är även att det finns stöd för att skapa arvshierarkier av processer, vilka ska underlätta bland annat förändringsanalyser.

### **Riktlinjer för tillägg till BML**

Det Fowler (1997) skriver om tillägg till ett existerande modelleringsspråk kan sammanfattas som minimera tilläggen samtidigt som uttryckskraften maximeras. Inspirerade av detta utvecklade vi följande riktlinjer för hur vi skulle hantera de nya tilläggen i EBML:

- Att inte medvetet bryta mot existerande BML-regler. Om något kan modelleras med ett tillägg istället för en förändring av reglerna är det förra alternativet att föredra.
- Att göra så få tillägg som möjligt. Om något kan modelleras i vanlig BML utan att modellering blir alltför omständlig ska detta göras, istället för att en ny symbol läggs till. Detta för att minska komplexiteten och undvika onödiga tillägg.
- Då nya symboler ändå läggs till ska de, om det ökar förståelsen, likna existerande symboler i BML.

## **4.2 Symboler och andra notationsaspekter i EBML**

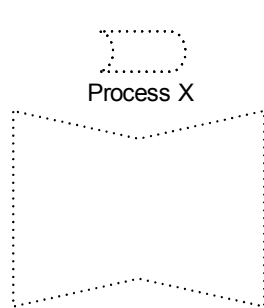
Tilläggen nedan kan delas upp i två huvudkategorier: de som möjliggör modellering av manuella rutiner samt de som lagts till på grund av andra användbarhetsaspekter. Den första kategorin innehåller symboler och regler för artificiella processmeddelanden, slumpmässiga starttillstånd, manuella aktiviteter och val. Den andra för anteckningar, numrering, versionshantering, aktörssymboler och för-varjemönster.

### **4.2.1 Artificiellt processmeddelande (APM)**

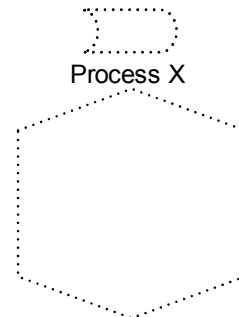
I BML är det möjligt för en aktör att skicka meddelanden till en annan aktör vars process tar emot och hanterar meddelandet. Om aktören är en applikation, det vill säga att aktören är datoriserad, kan den även skicka meddelande till sig själv. Däremot verkar det underligt att en mänsklig aktör skickar meddelanden till sig själv. En person meddelar inte sig själv att påbörja en annan process. Vid modellering utifrån aktörsperspektivet beskrivet ovan i avsnitt 4.1 är det dock ibland nödvändigt att en mänsklig aktör ska kunna skicka meddelanden till sig själv. Det kan till exempel finnas en reparations- och en reservdelsbeställningsprocess, där samma person som utför reparationen kan beställa reservdelar under reparationsprocessens gång (se 6.2.2).



Vi löste detta med att skapa ytterligare symboler för just denna typ av icke-existerande meddelanden, artificiella process-meddelanden (APM). Dessa ser ut som vanliga in- och utmeddelanden i BML, med skillnaden att deras konturer är streckade och att avsändare respektive mottagare alltid är processer (se Figur 17 och Figur 18). På detta sätt kan dessa typer av processer fortfarande grafiskt sett kommunicera med varandra.



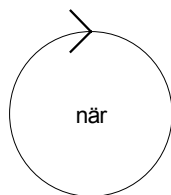
Figur 17. EBML-symbolen för ett artificiellt processinmeddelande inklusive sändaren, vilken alltid är en process.



Figur 18. EBML-symbolen för ett artificiellt processutmeddelande inklusive mottagaren, vilken alltid är en process.

#### 4.2.2 Slumpmässigt starttillstånd

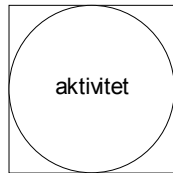
I BML följs ett starttillstånd alltid av ett inmeddelande (se 3.2.4). Återigen medför detta inget problem med datoriserade aktörer. Med mänskliga sådana uppstår dock problem. Vissa processer startas när aktören anser att det är dags att starta en process. Sådana starttillstånd följs i dessa fall inte alltid av ett inmeddelande, vilket bryter mot BML. För att lösa detta skapade vi ytterligare en typ av starttillstånd - ett slumpmässigt starttillstånd. Symbolen (se Figur 19) för detta är i huvudsak densamma som för ett vanligt starttillstånd. Det som skiljer är en pil i den övre delen av cirkeln. Dessutom anges det inuti symbolen när processen startas. Denna symbol kan också användas för processer som startas rutinmässigt av mänskliga aktörer, till exempel om fakturering sköts vissa datum varje månad. I sådana fall anges tidsintervallet inuti symbolen.



Figur 19. EBML-symbolen för ett slumpmässigt starttillstånd.

#### 4.2.3 Manuell aktivitet

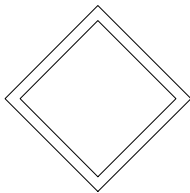
BML har, eftersom det syftar till att modellera datoriserade processer, ingen symbol för manuella aktiviteter. Med manuella aktiviteter menas oftast fysiska aktiviteter som utförs av mänskliga huvudaktörer. Exempel på en manuell aktivitet är flyttandet av ett fysiskt föremål från en plats till en annan. I EBML har vi följaktligen lagt till en symbol för detta, vilken visas i Figur 20. Inuti symbolen står en kort beskrivning av aktiviteten. Att kunna visa manuella aktiviteter är egentligen det mest centrala tillägget i EBML. Utan detta skulle det inte vara möjligt att visa processer med människor som huvudaktörer då sådana processer ofta innehåller manuella aktiviteter.



Figur 20. EBML-symbolen för en manuell aktivitet.

#### 4.2.4 Manuellt val

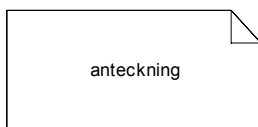
Eftersom huvudaktörer i BML är datoriserade blir alla val de gör automatiserade. Med mänskliga huvudaktörer följer att det finns val som inte är datoriserade, det vill säga manuella val. Initialt använde vi samma BML-symbol för både manuella och automatiserade val. Problemet uppstod dock eftersom både manuella val och automatiserade val kan, även om det inte sker ofta, inträffa i en och samma samma BML-modell. Det kan då i en del fall vara svårt att med säkerhet avgöra vilket typ av val som sker. Som följd av detta lade vi till en ny symbol för manuellt val (se Figur 21). Symbolen är densamma som den för automatiserat val, med skillnaden att den är dubbelt inramad. Samma sak som för automatiserade val gäller för manuella, från dem leder flera beslutsvägar och flödet går via en av dem. Vilken beslutsväg som kommer att väljas bestäms av verksamhetens affärsregler. Vid manuella val finns dock en möjlighet för den mänskliga aktören att göra en egen tolkning av reglerna.



Figur 21. EBML-symbolen för ett manuellt val.

#### 4.2.5 Anteckning

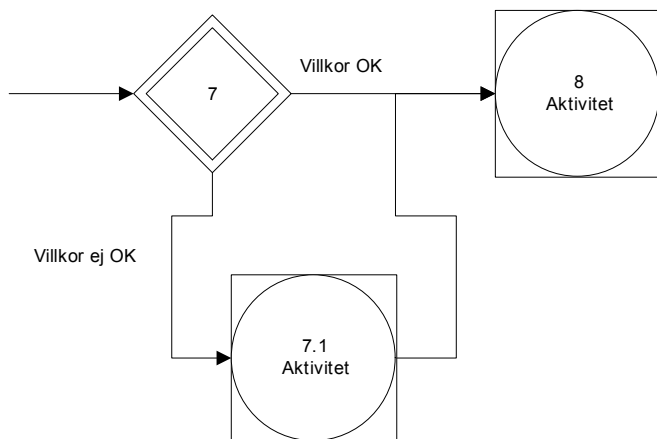
För att kunna kommentera företeelser direkt i EBML-modellerna valde vi att lägga till UML:s symbol för anteckningar (se Figur 22). Anteckningar ska användas sparsamt. Då de används är dess främsta syfte att öka förståelsen av modellen. Ett annat användningsområde för anteckningar, vilket vi konsekvent använde oss av, är att notera var oklarheter fanns. Detta förenklar vid genomgång av respektive modell då det är mindre risk att dessa oklarheter glöms bort under efterföljande diskussioner.



Figur 22. EBML-symbolen för en anteckning.

#### 4.2.6 Numrering

En BML-modell bör alltid åtföljas av en textuell beskrivning. För att förenkla både skrivandet och läsandet av denna har vi i EBML lagt till numrering av alla symboler i modellerna. En numrering identifierar unikt varje objekt i en modell (se Figur 23). Detta förenklar diskussionen av modellerna då det eliminerar risken för missförstånd. Vi slår inte fast exakt vilken typ av numrering som ska användas, till exempel om siffror, bokstäver och olika typer av nivåer ska användas, i stället får detta avgöras från projekt till projekt.



Figur 23. Ett exempel på hur numrering kan användas.

#### 4.2.7 Versionshantering

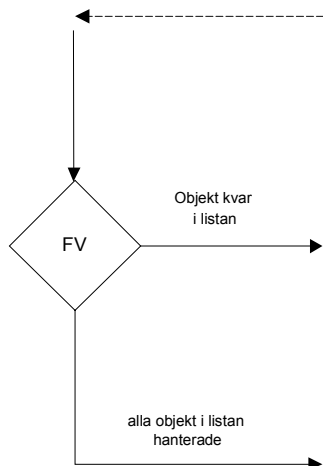
Att numrera symbolerna räcker dock inte fullt ut – i två olika versioner av samma process kan samma objekt ha olika nummer. För att unikt kunna identifiera objektet behövs numrering på en annan nivå – versionshantering. Denna kan skötas manuellt eller med hjälp av ett datoriserat verktyg. Oavsett sätt, bör varje modell, eller åtminstone varje uppsättning av modeller, märkas med versionsnummer och datum för att på så sätt undvika missförstånd vid identifiering av objekten. Inte heller i det här fallet bestämmer vi exakt vilken typ av numrering som ska användas för versionerna, utan det får avgöras från projekt till projekt.

#### 4.2.8 Aktörssymboler

I BML är de aktörer som finns antingen människor eller applikationer och för dessa två finns det redan symboler. När processer med mänskliga aktörer modelleras kan det uppkomma nya typer av icke-mänskliga aktörer, till exempel pärmar. I EBML har vi valt att tillåta användandet av vilken symbol som helst för dessa typer av nytillkomna aktörer. Det är även tillåtet att byta ut symbolerna för applikation och person. BML stöder redan allt detta, om än implicit. Vi har valt att explicitgöra det.

#### 4.2.9 För-varjemönster

Ytterligare ett tillägg i EBML är för-varjemönstret. Detta är inget direkt tillägg till BML, i detta fall handlar det snarare om ett specifikt sätt att använda det. Vi valde att explicitgöra detta eftersom vi upptäckte att det var ett ofta förekommande mönster när vi modellerade. Mönstret används då en lista eller dylikt ska gås igenom och samma sak ska göras för varje objekt i listan. Detta modelleras i EBML med ett val, antingen finns det fortfarande kvar objekt i listan, eller så är hela listan igenomgången. I det första fallet följer sedan ett ospecificerat antal symboler, som exempelvis utmeddelanden, som alltid slutar med en återgång till valet. I det andra fallet fortsätter flödet vidare. För-varjemönstret kan vara antingen ett automatiskt eller manuellt val. Inuti valsymbolen ska det alltid stå FV för att förtydliga att det är ett för-varjemönster. Ett exempel på mönstret visas i Figur 24.



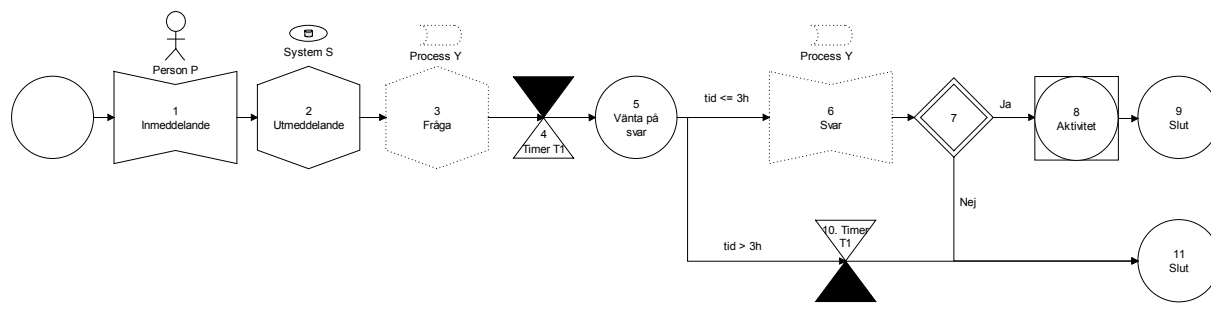
Figur 24. Ett exempel på för-varjemönstret i EBML.

### 4.3 Dynamisk processmodell i EBML

En dynamisk processmodell i EBML skiljer sig en del från dess motsvarighet i BML (se 3.3). Detta är en följd av de nya symbolerna som presenterats ovan. Själva flödeslogiken är dock densamma som tidigare. Processen utförs, som tidigare nämnts, av en huvudaktör, det vill säga processen beskriver en huvudaktörs agerande. Namnet på denna ska stå under processens namn.

Processen X i Figur 25 startar med att Huvudaktör H tar emot ett meddelande, Inmeddelande, från Person P. Ett utmeddelande, Utmeddelande, skickas till en applikation, System S. Därefter skickas ett ut-APM, Fråga, till Process Y. Efter det startas en timer, Timer T1, innan processen hamnar i ett väntande tillstånd, Vänta på svar. Antingen går Timer T1 ut eller så kommer ett in-APM, Svar, från Process Y. Om svaret är positivt utförs en manuell aktivitet, Aktivitet. Aktivitet.

Process X  
Huvudaktör H

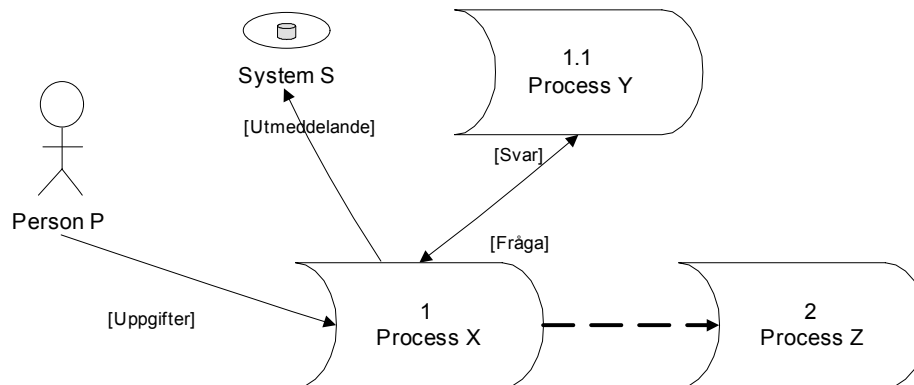


Figur 25. Exempel på en dynamisk processmodell i EBML.

### 4.4 Statisk processmodell i EBML

Den statiska processmodellen i EBML är i huvudsak densamma som för BML. Den enda skillnaden är en följd av omöjligheten i BML att illustrera implicita kopplingar mellan processer. Med detta menas då process nummer två inte startas av ett meddelande från process ett, men process ett ändå alltid måste ske före process två. Exempel på detta visas i Figur 26. Process Z kan inte starta innan Process X är klar. I BML skulle denna koppling inte synas, processerna skulle till synes vara fristående från varandra. I EBML illustreras en implicit

koppling med en streckad pil, vilket visas i Figur 27. Den dynamiska processen som beskrivs i Figur 25 motsvaras av Process X i Figur 26.



Figur 26. Ett exempel på en statisk processmodell i EBML.



Figur 27. EBML-symbolen för en implicit koppling mellan två processer i en statisk processmodell, där destinationsprocessen inte kan starta utan att den föregående processen avslutats.

## 4.5 Arvshierarkiska modeller i EBML

Arvshierarkier av processer har inte någon motsvarighet i BML. Syftet med att skapa en arvshierarki av processer är att strukturera verksamhetens processer i hierarkiska modeller för att underlätta arbetet i förändringsanalysen. Skapandet av arvshierarkin gör att de modellerade processernas skillnader och likheter explicitgörs. Modellen vi tagit fram använder sig av det från Malone (1999) givna tvådimensionella sättet att representera olika typer av processer (se 2.6). Modellen lånar också från UML:s sätt att representera arv (Schmuller, 1999).

För att enklare förstå arvshierarkin i EBML presenteras först ett antal koncept som härrör från objektorienterad programmering. Därefter presenteras kort de två olika vyerna som används för att beskriva en arvshierarki: processhierarkin och processmodeller med arv.

### 4.5.1 Arv

En process kan ärva sitt innehåll från en annan överordnad process. Det betyder att allt som görs i superprocessen även görs i subprocessen. Subprocessen är en specialisering av superprocessen som alltså är en generalisering av subprocessen. Subprocesser kan sedan i sin tur agera som superprocess för andra processer.

### 4.5.2 Abstrakthet

En process kan vara abstrakt. Med detta menas här att den inte kan instantieras, det vill säga att den inte "finns" i verkligheten. Superprocesser är ofta abstrakta processer – de innehåller basflödet för de subprocesser som ärver från dem. Även processobjekt i en process kan vara abstrakta. Ett processobjekt är ett samlingsnamn för de entiteter som en dynamisk processmodell består av och som representerar syntaxen i EBML, det vill säga start-, slut-, och vänttillstånd, meddelanden, APM, val, manuellt val, manuell aktivitet samt timer. En process

som innehåller abstrakta processobjekt är per definition även den abstrakt då den omöjlig kan utföras i verkligheten. Abstrakthet illustreras med att processens eller processobjektets namn skrivs med kursiv text i modellen, vilket vi lånat från UML (Schmuller, 1999). Exempel på detta visas i Figur 28.

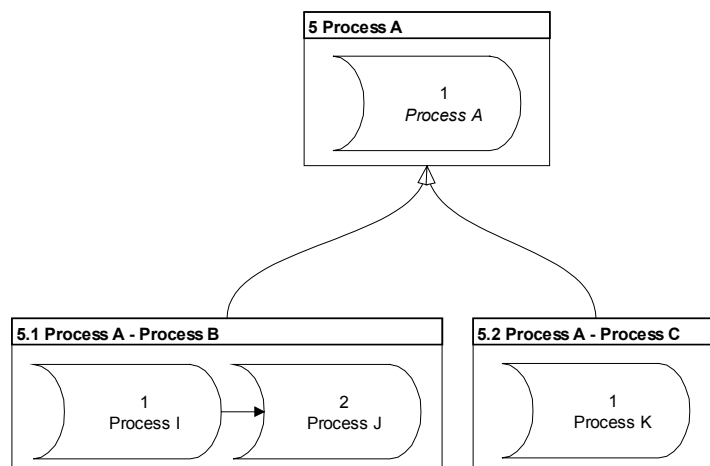
### 4.5.3 Ersättning

Subprocesser som ärver från en process med abstrakta processobjekt måste, för att inte också vara abstrakt, ersätta (eng. override) dessa. Ersättning innebär att processobjektet i subprocessen förändrats, det är inte längre detsamma som i superprocessen. En process kan även ersätta icke abstrakta processobjekt. Det existerar således ingen form av skydds nivåer som till exempel public, private och protected, som det gör i vanlig objektorienterad programmering. För att tydliggöra ersättning har vi valt att använda Malones (1999) färgkodning för att visa vilka processer och processobjekt som är ersatta alternativt nedärvda utan förändring. Ersatta processer och processobjekt illustreras med att de är vitmarkerade. Om de istället är nedärvda utan förändring är de gråmarkerade.

### 4.5.4 Processhierarki

I processhierarkin ingår alla processer som modellerats i de dynamiska processmodellerna. Utöver dessa tillkommer de nya, oftast abstrakta, superprocesser som skapas för att innehålla det gemensamma för subprocesser som liknar varandra. Pilarna pekar från subprocessen till superprocessen, det vill säga att dessa är en specialisering av superprocessen. Även på denna nivå bör processerna numreras. Återigen bestämmer vi dock inte något fast format på numreringen.

I Figur 28 är processerna I, J och K de ursprungliga processerna. Process A är en ny superprocess som skapats för att innehålla det gemensamma från dem. Process A är skriven i kursiv text, vilket visar att den är abstrakt och därmed inte kan instantieras.



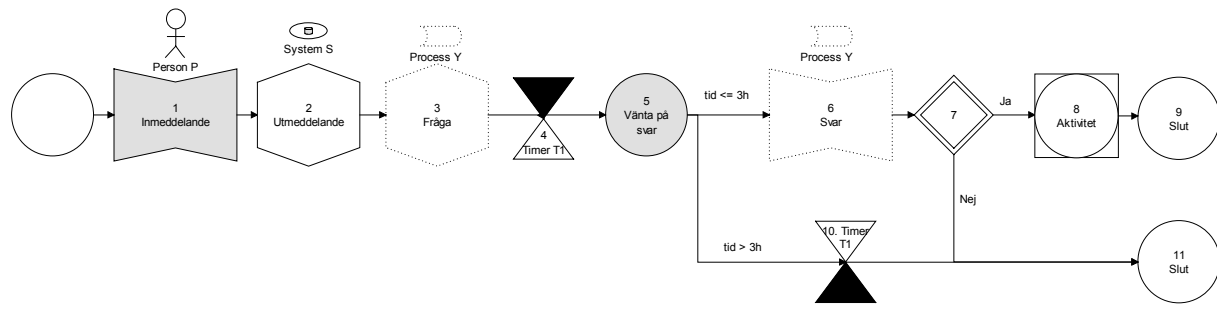
Figur 28. Ett exempel på en hur en arvshierarki modelleras i en processhierarki.

### 4.5.5 Processmodell med arv

Varje process som finns i processhierarkin har en motsvarande dynamisk processmodell (se 4.3). Då hierarkin är färdig ska samtliga processmodeller som har en superprocess modifieras för att visa vilka av deras processobjekt som ärvt. Dessutom måste processmodeller skapas för de nya superprocesserna.

Figur 29 visar ett exempel på en processmodell som kompletterats med arvsinformation, det vill säga att processobjekten har färgkodats för att visa om de ärrvts eller inte. De gråmarkerade processobjekten är nedärvda utan förändring från. De vitmarkerade processobjekten är antingen ersatta från superprocessen eller helt nya för subprocessen.

Process X  
Huvudaktör H



Figur 29. Exempel på en processmodell i EBML som kompletterats med arvsinformation.

## 5 Att arbeta med EBML

I det här kapitlet presenteras ett förslag till ett arbetssätt för hur en användare steg för steg kan arbeta med EBML i verksamhetsmodellering och förändringsanalys. Kapitlet är uppdelat i tre huvudavsnitt. I det första avsnittet beskrivs bakgrunden till det arbetssätt som vi föreslår och presenterar. I det andra avsnittet beskrivs steg som bör ingå i verksamhetsmodelleringsfasen och i det tredje beskrivs arbetsstegen i en förändringsanalys.

### 5.1 Bakgrunden till föreslaget arbetssätt

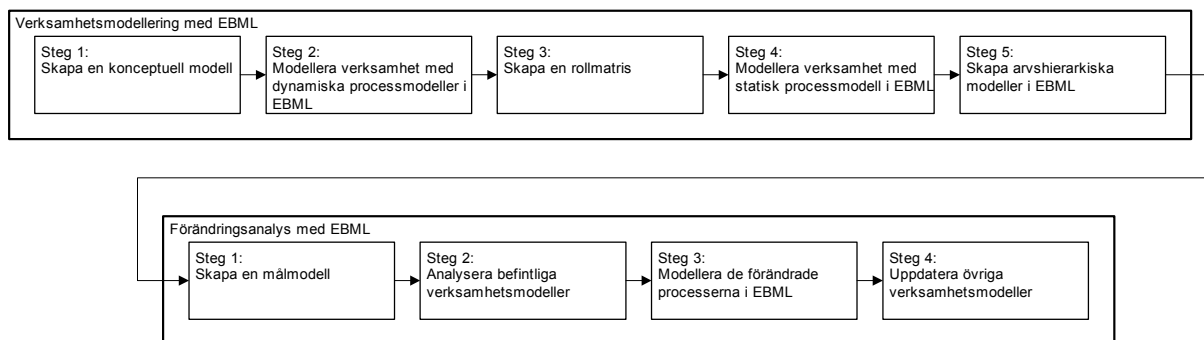
För att åstadkomma en så komplett bild som möjligt av en organisation och dess olika delar krävs att organisationen beskrivs på ett strukturerat sätt. Strukturerade beskrivningar i form av modeller gör det möjligt att se hur olika delar av en organisation interagerar och påverkar varandra. Modellerna ger en överskådlig och koncis beskrivning och är därför en god utgångspunkt för diskussion och analys. (Bubenko, 1998)

Ett sätt att underlätta arbetet kring verksamhetsmodellering och analys i ett systemutvecklingsprojekt är att upprätta och tillämpa ett arbetssätt. Ett arbetssätt är ett förslag på vilka steg som bör ingå en specifikations- och designprocess (Fowler, 1997). Tillämpning av ett arbetssätt bör resultera i en god dokumentation av verksamheten och underlätta kommunikation mellan deltagande intressenter. Vidare bör nämnas att ett arbetssätt som kan återanvändas bör leda till ökad färdighet och till att onödiga misstag undviks i framtiden (Andersen et al, 1994).

Vårt förslag på ett arbetssätt för EBML har sin utgångspunkt i de erfarenheter vi fick under det iterativa arbetet med fallstudien, men också i de litteraturstudier vi bedrivit under uppsatsarbetets gång.

I vårt förslag på arbetssätt kompletteras processmodellerna i EBML med andra typer av modeller, som konceptuell modell, målmodell och rollmatris. Kompletteringen syftar till att ge en bättre helhetsbild och därmed generera en bättre förutsättning för ett välgrundat resonemang kring möjliga förändringar i analysfasen.

I Figur 30 visas en modell som sammanfattar de olika arbetsstegen i arbetssättet för EBML.



Figur 30. En modell som visar de steg som ingår i arbetssättet för EBML.

## 5.2 Verksamhetsmodellering med EBML

Verksamhetsmodelleringsfasen består i princip av konstruktion och dokumentation av modeller som representerar de aspekter av verksamheten som är av betydelse för analysen och förväntade resultat. Arbetsgången delas in i ett antal steg, där ordningen bestäms av att vi går från en generell till en mer specifik och detaljerad verksamhetsbeskrivning. Det iterativa inslaget i arbetsgången är dock starkt.

### 5.2.1 Steg 1: Skapa en konceptuell modell

#### Syftet med en konceptuell modell

Den konceptuella modellen används för att identifiera och definiera entiteter och fenomen som återfinns i verksamheten och dess omgivning. Huvudsyftet med en konceptuell modell är att ge en generell bild av verksamheten och de entiteter och fenomen som närmare utreds i de andra modellerna. En konceptuell modell fungerar därför som en typ av ordlista eller kontrollista för diskussion om verksamheten.

Vår ursprungliga tanke var att göra den konceptuella modellen till ett obligatoriskt delsteg i verksamhetsmodelleringsfasen. Arbetet med att modellera verksamheten med hjälp av EBML innebar dock att vi idag är av en annan åsikt. Vi upptäckte med tiden att informationen som finns i den konceptuella modellen tydligt kom att representeras och upprepas i processmodellerna gjorda i EBML. Vår rekommendation är därför att snarare använda den konceptuella modellen i ett initialt skede för att öka och strukturera kunskapen om verksamheten, och senare använda den som en form av verifierings- och kontrollmekanism.

Om det finns en god förkunskap om verksamheten och det som ska modelleras kan det vara ett onödigt steg att konstruera en konceptuell modell. Konstruktion av den konceptuella modellen är därför frivillig och snarare av kompletterande karaktär för dokumentationen.

#### Tumregler

Notationen kan variera beroende på vilken betydelse den konceptuella modellen har. Ju större betydelse desto kraftfullare notation. Eftersom den konceptuella modellen inte är av primär betydelse har vi valt en lättförståelig och enklare form av notationsspråk. Det är vanligt att börja med en enklare form, såsom vi föreslår här och sedan övergå till en mer avancerad notation om ett sådant behov växer fram med tiden (Bubenko, 1998).

En annan faktor som kan påverka modelleringsprocessen är identifiering av entiteter. Att identifiera entiteter kan uppfattas som svårt. Det finns emellertid ett antal frågor som kan underlätta identifieringen, exempelvis:

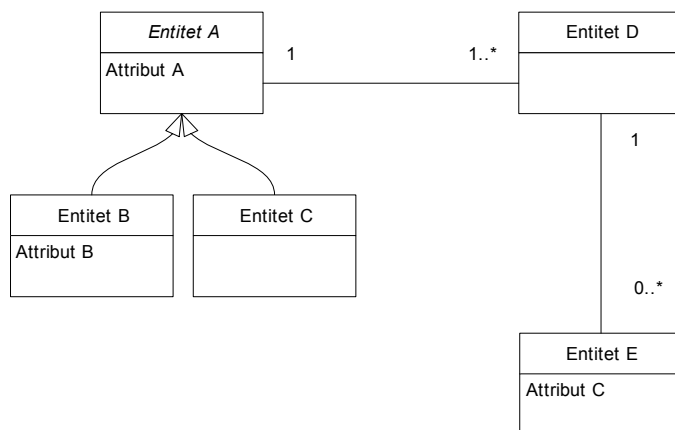


- vilka huvudentiteter kan identifieras?
- vilka aktörer och verksamhetsfunktioner existerar?
- hur är entiteter relaterade till varandra?
- hur många och vilka instanser av entiteten existerar?
- vilka attribut finns?

### Exempel på en konceptuell modell

Den konceptuella modellen består av komponenter som entiteter, relationer och informationsbärande attribut. Entiteter kan kopplas till varandra genom olika typer av relationer som association, generalisering och specialisering (ISA/ arv) och aggregering (del av). Användandet av dessa gör att abstraktionsnivån kan variera beroende på hur komplext problemområdet som modelleras är och beroende på vilken nivå som användaren önskar representera den verkliga situationen. (Schmuller, 1999)

Figur 31 visar ett exempel på hur en del av en konceptuell modell kan konstrueras för att representera någon form av problemområde som är av intresse. Modellen beskriver objekt i form av entiteter, objektens egenskaper i form av attribut och de olika typer av relationer som kan existera mellan objekten. I exemplet visas också avbildningsregler eller så kallade kardinalitetsregler för objekt förekomster. Dessa regler bestämmer hur många objektinstanser som kan ingå i en relation (Schmuller, 1999). Kardinaliteten anges med en minimum- eller maximumnotation och kan avläsas från båda riktningar.



Figur 31. Exempel på en konceptuell modell.

## 5.2.2 Steg 2: Modellera verksamhet med dynamiska processmodeller i EBML

### Syftet med dynamiska processmodeller i EBML

I en processororienterad verksamhetsanalys är kärnan verksamhetens affärsprocesser. En affärsprocess är en samling av organisationsaktiviteter som tillsammans skapar ett värde för kund (Davenport, 2000). Processmodeller konstrueras för att analysera affärsprocesser och flöden av information och material i verksamheten (Bubenko, 1998). Vi vill i det avseendet betona skillnaden mellan processmodellering i form av verksamhetsmodellering och specifikation av datoriserade system. Verksamhetsmodellering är vårt fokus och syftet med EBML, medan syftet med BML i första hand är systemspecifikation och därför ligger på en annan abstraktionsnivå. Nedan beskrivs verksamhetsmodellering med EBML, dock med vissa undantag eftersom verksamhetsmodellering inte utesluter specifikation av system. Det finns

med andra ord inget som hindrar att både verksamhet och IT-system modelleras med EBML eftersom BML inkluderas i EBML.

Processmodeller visualiserar en organisations eller systems dynamiska beteende. Identifieringen av vilka processer som ska modelleras beror på syftet med modelleringen och vilken abstraktionsnivå som efterfrågas. Ytterligare en viktig aspekt är att processmodellering även kan beskriva processerna ur olika perspektiv beroende på om syftet är att presentera dåtid, nutid eller framtid.

Det kan vara problematiskt att identifiera och angripa rätt processer vid processmodellering. I verksamhetsmodellering kan det vara särskilt svårt om verksamheten är stor. Varje organisation har emellertid en unik samling av verksamhetsprocesser och i de flesta fall är nyckelprocesserna ganska få till antalet, men hur de ser ut beror på verksamhetsbransch och vad som produceras. En nyckelprocess representerar en komplett affärsprocess. En nyckelprocess i EBML innefattar därför alla de processer som tillsammans resulterar i ett värde för kund, till exempel den kedja av processer som sker från att en kund beställer en vara till att varan levererats och fakturan betalats. (Hammer, 1996)

En nyckelprocess utgörs ofta av flera på varandra följande aktiviteter och kan därför delas in i ett antal distinkta delprocesser. Dessa delprocesser delas i vår definition in i två typer av delprocesser som kallas primära och sekundära processer.

Primära processer kännetecknas av att de nödvändigtvis måste ingå i en nyckelprocess för att den ska fullföljas. I orderhantering, som kan ses som en nyckelprocess som omfattar både ordermottagning, leverans av beställd produkt och fakturering, måste exempelvis primärprocessen leverans av beställd produkt ingå för att en organisation ska lyckas leverera det kundvärde som eftersträvas i en processororienterad organisation.

En sekundär process kännetecknas av att de inte nödvändigtvis behöver ingå i en nyckelprocess för att den ska kunna fullföljas, utan kan i viss mån ses som en typ av stödprocess. Det är viktigt att betona att sekundära processer är underordnade primärprocesser. En sekundär process är med andra ord beroende av en eller flera primärprocesser, vilket innebär att en sekundär process endast kan initieras av den primärprocess som den relateras med. Ett kostnadsförslag exempelvis, görs kanske inte rutinmässigt varje gång en order hanteras utan initieras endast under vissa förhållanden som när kunden explicit begärt ett kostnadsförslag. Orderhanteringen, som i det här fallet är en nyckelprocess, kan fullföljas även om inget kostnadsförslag läggs.

En dynamisk processmodell modelleras, som tidigare nämnts, alltid ur en specifik aktörs perspektiv. En konkret fysisk person kan dock utföra olika processer, även sådana som ligger utanför hans eller hennes huvuduppgifter. Det betyder att en huvudaktör för en process inte alltid behöver vara en enskild fysisk person. För att hantera detta bör ett samlingsnamn skapas för personer som utför processen – en så kallad roll. För att fortfarande behålla kopplingen mellan fysiska personer och roller skapas en rollmatris. Hur detta går till beskrivs i avsnitt 5.2.3.

## **Tumregler**

Identifieringen och namngivningen av en verksamhets processer är ett avgörande steg. Några tumregler är att processer som identifieras ska kunna relateras till entiteter, attribut eller relationer i den konceptuella modellen om en sådan skapats, men processerna ska även relateras till den rollmatris som tas fram i samband med processmodelleringen och som närmare beskrivs i nästa delsteg. Rollmatrisen visar till exempel vilken roll som utför och ansvarar för en viss process och därmed utgör processens huvudaktör. Det är också väldigt

viktigt att namnge de identifierade processerna eftersom det underlättar all vidare kommunikation och dokumentation.

Frågor som kan underlätta identifieringen av processer som ska modelleras är exempelvis:

- vilka är organisationens huvudsakliga funktioner och nyckelprocesser?
- vilka processer är primära respektive sekundära processer?
- hur är processerna relaterade till varandra?
- vem är processens huvudaktör?
- vilka personer kan utföra processerna och vara huvudaktörer?

### **Exempel på en dynamisk processmodell i EBML**

Vi har valt att utelämna den tekniska beskrivningen av den dynamiska processmodellen eftersom den finns utförligt beskriven i föregående kapitel (se 4.3).

### **5.2.3 Steg 3: Skapa en rollmatris**

#### **Syftet med en rollmatris**

Syftet med en rollmatris är att definiera de typer av roller som kan identifieras i verksamheten, samt relatera rollerna till de olika anställningstitlar som existerar i organisationen. En rollmatris utvecklas i samband med att processerna identifieras och utgör därför ett parallellt delsteg i processmodelleringen.

Med en anställningstitel medföljer ett primärt ansvar för vissa uppgifter (processer) i verksamheten. Det behöver dock inte innebära att personen enbart är begränsad till dessa uppgifter. Det är fullt möjligt att personen kan utföra uppgifter utöver sitt ordinarie ansvarsområde. I en flexibel organisation är detta särskilt tydligt, det vill säga att personer med olika titlar kan utföra ett flertal uppgifter som inte nödvändigtvis ingår i deras ordinarie uppgifter. Till exempel kan en tekniker hoppa in och utföra den lageransvariges arbete när det är brist på personal i lagret.

I EBML är en roll något som dessa personer antager då de utför en viss uppgift. En roll är således detsamma som en mänsklig huvudaktör i en dynamisk processmodell och en gemensam benämning för de personer som kan utföra flera processer i verksamheten. En rollmatris visar följaktligen vilka roller en anställd med en viss anställningstitel kan ha.

#### **Tumregler**

Tumregeln vid konstruktion av rollmatrisen är att fokus bör riktas till att hitta rollbeteckningar som inte kan missförstås och som täcker in möjliga anställningstitlar.

Frågor som kan underlätta konstruktionen av rollmatrisen är bland annat:

- vilka anställningstitlar existerar i verksamheten?
- vilka huvudaktörer finns enligt de dynamiska processmodellerna?
- vilka roller kan en viss anställningstitel ha?
- vilka uppgifter är ordinarie för en viss titel?
- vilka uppgifter, som inte är ordinarie, kan en viss titel ha?

#### **Exempel på en rollmatris**

I Figur 32 visas ett exempel på en rollmatris. Matrisen visar rollfördelningen mellan de anställda och varje kryss anger roller per anställningstitel. Längs x-axeln definieras

anställningstitlar och längs y-axeln definieras de roller eller huvudaktörer som identifierats i verksamhetens processer. I matrisens rutor sätts kryss för varje uppgift som en anställd genom sina roller kan ha. För att tydliggöra en anställds ordinarie uppgifter görs krysset fetstilt, medan icke-ordinarie uppgifter sätts inom parentes.

	Titel 1	Titel 2	Titel 3	Titel 4
<b>Roll A</b>	<b>X</b>	(X)		
<b>Roll B</b>		<b>X</b>		
<b>Roll C</b>			<b>X</b>	
<b>Roll D</b>				<b>X</b>
<b>Roll E</b>	(X)	<b>X</b>		

Ordinarie uppgift                   **X**  
 Icke-ordinarie uppgift           (X)

Figur 32. Exempel på en rollmatris.

## 5.2.4 Steg 4: Modellera verksamhet med statisk processmodell i EBML

### Syftet med en statisk processmodell i EBML

Syftet med en statisk processmodell i EBML är att visualisera en organisations hela processtruktur. Genom att konstruera en statisk processmodell generas en helhetsbild, som beskriver alla processer och hur de interagerar med sin miljö.

Att skapa en statisk processmodell är fördelaktigt av flera anledningar. Först och främst ger modellen information om hur de enskilda processerna interagerar och kommunicerar med varandra och sin omgivning. För det andra är den statiska modellen viktig eftersom modellen ger en översikt av alla dynamiska processer som identifierats. För det tredje underlättar modellen arbetet med de arvshierarkiska modellerna som sker i nästa arbetssteg (se 5.2.5).

### Tumregler

Storleken på den statiska processmodellen är beroende av verksamhetens storlek, vilket innebär att ju större verksamhet och ju fler processer som existerar, desto större och mer komplex kommer den statiska processmodellen att bli. Det är därför viktigt att arbeta iterativt och ta en bit i taget för att underlätta modelleringsarbetet.

Genom att titta på hur de dynamiska processmodellerna interagerar går det att härleda en statisk processmodell. Det enklaste sättet att konvertera dynamiska processmodeller till en statisk processmodell är att utgå från respektive nyckelprocess och börja från början i processen för att sedan arbeta sig framåt i sekventiell ordning. På så sätt byggs den statiska processmodellen upp, process för process.

Frågor som kan underlätta konstruktionen av den statiska processmodellen är exempelvis:

- vilka är verksamhetens nyckelprocesser?
- vilka processer utgör primärprocesser?
- vilka processer utgör sekundärprocesser?
- hur är processerna relaterade till varandra?
- hur interagerar en viss process med sin miljö, det vill säga med vilka andra processer, aktörer eller applikationer?
- vilka är ut- och inmeddelandena som kopplar två eller fler processer med varandra eller sin externa miljö?

- existerar några processer som startar slumpmässigt och som är beroende av att föregående processer först slutförts, det vill säga existerar några implicita kopplingar mellan processer?

### **Exempel på en statisk processmodell i EBML**

Vi har valt att utelämna den tekniska beskrivningen av den statiska processmodellen eftersom den finns utförligt beskriven i föregående kapitel (se 4.4).

## **5.2.5 Steg 5: Skapa arvshierarkiska modeller i EBML**

### **Syftet med en arvshierarkiska modeller i EBML**

Syftet med att skapa en arvshierarki av processer är att strukturera verksamhetens processer i hierarkiska modeller för att underlätta arbetet i förändringsanalysen. De hierarkiska modellerna bygger på principer utvecklade av bland annat Malone (1999). (se 2.6)

Genom att visa processerna i hierarkiska arvsmodeller tydliggörs processernas uppbyggnad och därmed likheter och skillnader. Modellerna ger en överskådlig bild av en grupp av relaterade processers relation, men visar samtidigt varje enskild process innehåll. Som en följd tydliggör arbetet med arvshierarkin ytterligare definitionen av verksamhetens nyckel-, primär- och sekundärprocesser, vilket i sin tur konkretiserar bilden av verksamheten.

Skapandet av en arvshierarki bör påbörjas i samband med att processerna identifierats och sker därefter iterativt allt eftersom ny information tillkommer med tiden. Arvshierarkin bör emellertid vara klar innan förändringsanalysen påbörjas.

### **Tumregler**

En tumregel är att tänka objektorienterat vid modelleringen av arvshierarkier. Skillnaden är att det är processer i stället för klasser som modelleras. För att skapa en hierarki bör de dynamiska processernas egenskaper, det vill säga delar eller aktiviteter i form av processobjekt jämföras. Därefter kan de processer som har gemensamma flöden grupperas. Utifrån dessa grupper kan sedan generella processer, det vill säga abstrakta superprocesser skapas. Dessa abstrakta superprocesser ska innehålla generella beteenden som kan ärvas till specialiserade processer. (Malone, 1999)

Vi rekommenderar att arbeta med en grupp av processer i taget. Det kan också vara en fördel att arbeta med en specialiseringsnivå i taget. Genom att arbeta iterativt och steg för steg avverka grupperingar och specialiseringsnivåer blir specialiseringsnivåerna konsekventa.

Frågor som kan underlätta modelleringen av arvshierarkin är exempelvis:

- vilka nyckelprocesser består av samma typer av primärprocesser?
- är flöden och processobjekt i primärprocesserna likadana i alla avseenden?
- är de gemensamma flödena generella för flera processer?
- vilka är skillnaderna mellan en grupp av relaterade processer?

### **Exempel på arvshierarkiska modeller i EBML**

Vi har valt att utelämna den tekniska beskrivningen av de arvshierarkiska modellerna eftersom de finns utförligt beskrivna i föregående kapitel (se 4.5).

### **5.3 Förändringsanalys med EBML**

Verksamhetsmodelleringsfasen bör ha genererat en god förståelse för organisationens aktuella tillstånd och befintliga processer. Förändringsanalysen tar projektet ett rejält steg framåt och kommer att fokusera kring specifikation av organisationens framtida tillstånd, det vill säga hur organisationen och dess processer ska verka i framtiden.

Analysfasen består av fyra steg som sker sekventiellt (se Figur 30.). Först ska en målmodell skapas som beskriver verksamhetens mål. I det andra steget sker själva förändringsanalysen som följs av ett tredje steg, modifiering och uppdatering av processmodellerna i EBML. I det fjärde steget sker uppdatering av övriga verksamhetsmodeller, i enlighet med de förändringar som analysen producerat.

#### **5.3.1 Steg 1: Skapa en målmodell**

##### **Syftet med en målmodell**

En målmodell används för att mer konkret beskriva en verksamhets målsättning och de inre och yttre faktorer som påverkar möjligheten att uppfylla målen.

Enligt Bubenko (1998) är syftet med en målmodell att skapa ett ramverk för att se vilken relevans verksamhetens befintliga processer har i förhållande till dess målsättning. Genom att sätta processerna i relation till målsättningen förklarar målmodellen varför eller varför inte vissa processer och verksamhetskrav existerar, vilket i sin tur underlättar arbetet i förändringsanalysen. (se 2.5)

Utöver detta ger målmodellen också insikt om verksamhetens prioriteringar och affärsregler. I en förändringsanalys där effektivisering och förbättring ofta är syftet kan föreslagna förändringar baseras på fakta som återges i målmodellen. Det ger föreslagna förändringar en god förankring i verksamheten. Vidare kan konstruktionen av en målmodell också ge information om vilka resurser som finns tillgängliga eller kan göras tillgängliga i en förändringsprocess, vilket också är fördelaktigt och underlättar förändringsanalysen.

##### **Tumregler**

För att skapa en målmodell kan man samla deltagare till ett idémöte, vid vilket alla deltagare ska ha möjlighet att yttra sig och alla åsikter ska beaktas. Det första utkastet av en målmodell blir därför ofta något ostrukturerad, men genom att arbeta iterativt och inkludera strukturering och klassificering av målen kommer modellen att successivt förbättras. Det är också viktigt att formulera målen så klart och tydligt som möjligt så att de inte kan missförstås. Bubenko (1998) anger att målen bör beskrivas i hela meningar och som något mätbart. Genom att konsultera anställda och någon ur verksamheten med befogenhet att verifiera modellen bör resultatet också bli tillfredsställande för alla parter (Andersen et al, 1994).

Frågor som kan underlätta modelleringsprocessen kan vara:

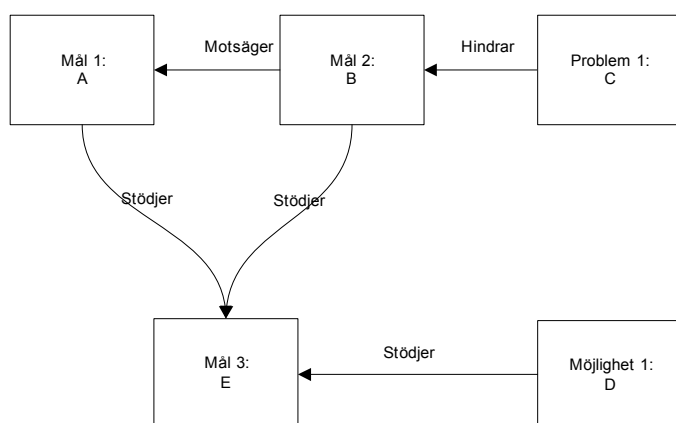
- vad vill verksamheten uppnå med verksamhetsmodelleringen och förändringsanalysen?
- vad har organisationen för mål i övrigt?
- vilka strategier existerar?
- vilka uttalade mål finns?
- finns några relevanta outtalade mål?
- existerar det några problem som hindrar något mål?
- vad orsakar problemet?
- existerar några möjligheter som är användbara?
- hur kan ett specifikt mål uppnås?

## Exempel på en målmodell

En målmodell består huvudsakligen av komponenter som representerar mål, problem och möjligheter. Ett mål representerar ett önskat verksamhetsmål eller tillstånd som organisationen vill uppnå. Det ska helst uttryckas som någonting kvantifierbart. Ett problem används för att visa på något i miljön som är icke önskvärt, som kan hindra eller hota någon målsättning. En möjlighet representerar någon resurs som kan underlätta att uppnå ett visst mål. En möjlighet kan också ge upphov till helt nya mål.

Komponenterna i målmodellen binds samman genom tre typer av enkelriktade relationer. En relation där en komponent stödjer en annan kallas stödjande relation. En komponent som hindrar eller negativt påverkar en annan komponent visas genom en förhindrande relation och är motsatsen till en stödjande relation. En situation där en målsättning är i konflikt med en annan visas med en motsägande relation.

Figur 33 visar exempel på en målmodell. Det är tillåtet och kan ibland vara motiverat att lägga till ytterligare komponenter i modellen som kommentarer, begränsningar och antaganden som gör modellen tydligare, men vi har här valt att enbart visa den grundläggande strukturen.



Figur 33. Exempel på en målmodell

### 5.3.2 Steg 2: Analysera befintliga verksamhetsmodeller

#### Syftet med analys av befintliga verksamhetsmodeller

Analysfasens andra steg är den viktigaste delen eftersom analysen kommer att leda till att organisationens framtida tillstånd definieras och ger underlag till beslut om förändring. Tanken med förändringsanalysen är alltså att öka kunskapen om organisationen i syfte att effektivisera dess verksamhet och processer. I analysen ingår därför två steg som stödjer detta.

I det första steget ska EBML-modellerna sättas i förhållande till målmodellen. Analys av denna relation klargör vilka förändringsbehov som finns inom verksamheten, eftersom målmodellen anger vilken relevans en verksamhets processer har i förhållande till sin målsättning. Därefter ska en plan för det fortsatta utvecklingsarbetet formuleras. Många systemutvecklare tror att de flesta problem inom en verksamhet låter sig lösas med hjälp av ett bättre informationssystem. Om en person är expert på systemutveckling är det lätt att se systemutveckling som lösningen på de flesta problem, liksom organisationsutvecklare vill se lösningen i organisationsutveckling eller rationaliseringsexperten i rationalisering. Genom

förändringsanalysen ska emellertid verksamhetens reella problem fastställas samt vilken typ av åtgärder som bör vidtas för att lösa dem (Andersen, 1994).

I det andra grundläggande steget i analysfasen ingår att mer konkret att föreslå förbättringar, vilket enligt Malone (1999) förutsätter att processerna i en given situation presenteras på ett sätt som underlättar förståelsen för dem. För att kunna föreslå förbättringar för en process i en given situation måste representationen ge möjlighet att kunna se och föreslå alternativa processmodeller. Dessa krav medför att processer bör presenteras på ett sätt som explicit visar likheter och skillnader mellan en grupp av relaterade processer. I det avseendet blir de arvshierarkiska modellerna i EBML aktuella, eftersom de uppfyller denna funktion.

### **Tumregler**

Analysen kan göras vid ett eller flera idémöten. Diskussionen vid mötet bör fokusera på de problem som finns definierade i målmodellen och de problem som framkommer under analysens gång. Mer specifikt, vad som orsakar problemen och hur de skulle kunna åtgärdas, det vill säga vilka alternativa processer som är att föreslå.

Ett antal frågor som är användbara i analysfasen är till exempel:

- vilka förändringar är önskvärda?
- vilket är förhållandet mellan målmodellen och processmodellerna, det vill säga stöds varje mål av en process som finns modellerad i EBML? Om så inte är fallet, varför?
- finns processer som inte är kopplade till mål i målmodellen? I sådant fall, varför?
- borde det finnas processer som stödjer ett specifikt mål?
- hur bör processen i sådant fall se ut?
- finns det grupper av processer som har stora likheter, men som utförs på mycket olika sätt?
- bör en sådan grupp av relaterade processer utföras under mer enhetliga former och riktlinjer?
- finns överflödiga processer?
- på vilket sätt är de överflödiga och kan de elimineras, ersättas eller förändras på något vis?
- vilka resurser kräver en viss process?
- bör organisationen investera i nya resurser? I sådant fall, vilken typ av resurser?

### **5.3.3 Steg 3: Modellera de förändrade processerna i EBML**

Syftet med att modellera de förändrade processerna är att tydliggöra de förändringar som analysen gett upphov till. Genom att konkretisera de önskade förändringarna i nya processmodeller blir det i efterhand enkelt att jämföra de ursprungliga processerna med de nya och se vilka skillnader som en framtida implementation kan medföra. De nya modellerna utgör alltså underlag till uppföljande diskussioner och kontroll om förändringarna verkligen är genomförbara, men möjliggör även en kontroll av om alla processer analyserats.

### **Tumregler**

Modelleringen av de nya processmodellerna bygger på den information som genererats i samband med förändringsanalysen.

Vi rekommenderar emellertid att modelleringsarbetet baseras på de modeller som visar de befintliga processerna och att de återanvänds så långt som möjligt. Det sparar tid, såvida analysen inte resulterat i enbart nya processer.



Det är viktigt att komma ihåg att spara de gamla och ursprungliga processmodellerna så att det i efterhand enkelt går att jämföra skillnaderna mellan de ursprungliga och framtida processerna. Det är också viktigt att spara alla modeller för att bibehålla en genuin verksamhetsdokumentation, som av olika anledningar kan bli aktuell att återgå till någon gång i efterhand.

### **Exempel på en förändrad process i EBML**

Förändringarna som eventuellt uppkommit i samband med analysen modelleras i enlighet med syntaxen för de olika typer av EBML-modeller, som finns utförligt beskrivet i föregående kapitel (se 4).

### **5.3.4 Steg 4: Uppdatera övriga verksamhetsmodeller**

#### **Syftet med att uppdatera övriga verksamhetsmodeller**

Syftet med att uppdatera övriga verksamhetsmodeller, det vill säga rollmatrisen och den konceptuella modellen, följer av två anledningar. Den första anledningen är att underhålla all dokumentation som modelleringsarbete och förändringsanalys gett upphov till, eftersom den inte bara är användbar i det aktuella projektet utan också kan vara användbar i andra projekt vid senare tillfällen.

Den andra anledningen är att om förändringsanalysen skulle resultera i förändringar som till exempel påverkar antalet roller i rollmatrisen, bör rollmatrisen göras om i en ny och uppdaterad version. Genom att jämföra den uppdaterade modellen med modellen som beskriver det ursprungliga nuläget, kan det bli lättare att se vilka skillnader önskade förändringar kan komma att medföra. Modellerna ger med andra ord ett bra underlag till uppföljande diskussioner eftersom de ger möjlighet att mer konkret se effekterna av föreslagna förändringar.

#### **Tumregler**

Precis som för modellering av förändrade EBML-processer i föregående arbetssteg gäller att uppdateringen av övriga verksamhetsmodeller baseras på den information som förändringsanalysen gett upphov till.

Vi rekommenderar också att återanvända de befintliga och ursprungliga modellerna i den mån det går. Det sparar tid i de fall analysen inte resulterat i helt nya uppgifter.

#### **Exempel på uppdaterade verksamhetsmodeller**

Uppdatering sker i enlighet med den notation som angivits för konceptuell modell (se 5.2.1) respektive rollmatris (se 5.2.3).

## **6 Tillämpning av EBML – en fallstudie**

I det här kapitlet presenterar vi hur vi tillämpat EBML och föreslaget arbetsätt i en verklig verksamhetsanalys av Stockholms Central Service AB.

Kapitlet består inledningsvis av en sammanfattande och textuell beskrivning av Stockholms Central Service AB och verksamhetens nuvarande situation. Därefter förklarar vi med exempel hur vi tillämpat oss av verksamhetsmodelleringens och förändringsanalysens olika delsteg för att kartlägga och analysera företaget med hjälp av processmodelleringsspråket

EBML och kompletterande verksamhetsmodeller. En mer utförlig beskrivning av företaget och fullständiga modeller finns i Pettersson & Wärynen (2001).

## **6.1 Beskrivning av Stockholms Central Service AB**

SCS står för Stockholms Central Service AB och är ett serviceföretag i Upplands Väsby som arbetar med service och reparation av hemelektronik. Till hemelektronik räknas främst ljud- och bildprodukter. SCS utför även service på datorer och vitvaror. Företaget har en liten filial i Enskede.

På SCS i Upplands Väsby arbetar sammanlagt sexton personer, varav två stycken är delägare med ansvar för administrativt arbete för hela företaget. En person är heltidsanställd i kundmottagningen. Tretton stycken är servicetekniker, varav en person innehar titeln servicechef och är arbetsledare för övriga tekniker. På SCS i Enskede arbetar endast en person som är servicetekniker, men som också betecknas som servicechef.

SCS ingår i Assist, som är en nordisk samarbetskedja bestående av tjugoen stycken serviceverkstäder. Samarbetet har medfört att verkstäderna slagit samman vissa administrativa funktioner. Exempelvis skickas alla fakturor via e-post till Assistedjans huvudkontor i Piteå.

SCS kunder består av återförsäljare, leverantörer och privatkunder. Återförsäljare utgör den största kunden och står för omkring nittio procent av reparationerna. Utbudet på tjänster är samma för de tre olika kundgrupperna. Däremot skiljer företaget på kundgrupperna eftersom olika priser, villkor och avtal gäller beroende på om kunden är en återförsäljare, leverantör eller privatperson.

En reparationstjänst registreras som en order. En order kan antingen vara en så kallad serviceorder eller utserviceorder. Typen av order bestäms generellt av vilken form av leverans, reparation och kund som ingår i processen. En serviceorder omfattar alla kundgrupper och innebär att kunden själv lämnar in och hämtar serviceobjektet. Reparationen sker på verkstaden. En utserviceorder omfattar enbart privatkunder och innebär att SCS hämtar och levererar serviceobjektet på beställning av kund. Reparationen kan ske antingen på verkstaden eller på plats hos kund.

Orderhantering är SCS viktigaste affärsprocess och därmed en nyckelprocess. Orderhanteringsprocessen består av fyra primära subprocesser, som alla finns i flera varianter, beroende på typ av leverans, kund och reparation:

- inlämning av serviceobjekt
- reparation av serviceobjekt
- utlämning av serviceobjekt
- fakturering av tjänst

Inlämning respektive utlämning av ett serviceobjekt sker idag på olika platser i verkstadens lokaler. Kundmottagning för privat- och direktkunder finns i den ena av verkstadens lokaler, medan lagermottagning för leveranser med lastbil från återförsäljare och leverantörer finns i den andra lokalen.

Orderhanteringen stöds idag av ett datoriserat informationssystem, men bara till en viss grad då verksamheten inte utnyttjar datorsystemet till fullo. Parallellt med datorsystemet används ett pärmsystem, i vilket bland annat order-, leverans- och beställningsdokument förvaras.

## 6.2 Exemplifiering av verksamhetsmodeller

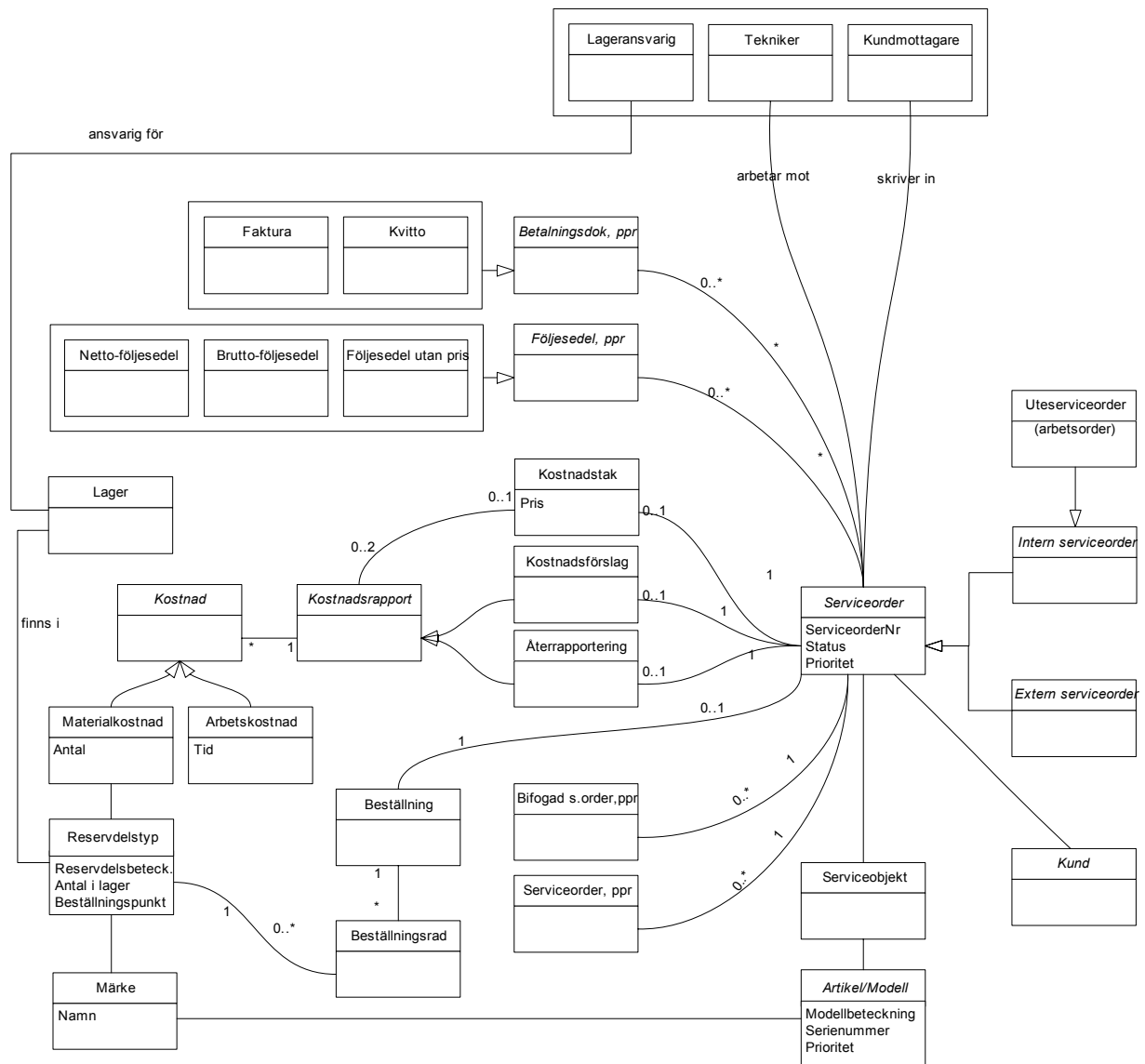
Nedan beskrivs och exemplifieras det tillvägagångssätt som beskrivits ingående i kapitel fem. Flera av modellerna är väldigt stora och i dessa fall har vi därför för översiktens och förståelsens skull ibland valt att endast visa delar av modellerna för att exemplifiera modelleringen.

### 6.2.1 Tillämpning av en konceptuell modell

Den konceptuella modellen används för att generellt beskriva verksamheten genom att identifiera och definiera entiteter och fenomen som finns i verksamheten och dess omgivning. Modellen fungerar därför som en typ av ordlista för det som närmare beskrivs i andra verksamhetsmodeller.

Orderhantering är SCS viktigaste nyckelprocess. Den del av den konceptuella modellen som finns avbildad i Figur 34 representerar de entiteter och fenomen i verksamheten som är mest centrala i orderhanteringsprocessen. All orderhantering bygger på registreringen av en serviceorder, som görs i det befintliga datorsystemet när serviceobjekt lämnas in av en kund, eller registrering av en utserviceorder om det är en privatkund som beställer en reparation.

Figur 34 visar att en serviceorder är en central entitet i verksamheten. I modellen nedan visas hur en utserviceorder ärver en serviceorder, eftersom en utserviceorder består av samma egenskaper, men har speciella tillägg för hemservice. Varje serviceorder är unik och identifieras genom sitt attribut serviceordernummer. Till serviceordern kopplas information om kund, serviceobjekt och ansvarig tekniker som finns definierade som separata entiteter. Vidare visar modellen övrig dokumenthantering som sker i samband med en specifik serviceorder. Entiteterna Beställning, Återrapportering, Kostnadsförslag, Följesedel och Betalningsdokument hör med andra ord till en specifik instans av en serviceorder. Det visas genom relationer mellan dessa entiteter och den kardinalitet som anges i relationen.



Figur 34. En konceptuell modell som representerar vilka entiteter och de relationer som är mest centrala i orderhantering på SCS.

## 6.2.2 Tillämpning av processmodellering i EBML

En dynamisk processmodell i EBML beskriver informationsflöden, aktiviteter, och processer i en organisation. Processmodellen visualiserar organisationens dynamiska beteende. Identifieringen av vilka processer som ska modelleras beror av den processorienterade ansatsen.

När vi identifierade processerna verksamheten utgick vi från den datainsamling som vi gjort i form av besök och intervjuer på företaget. Vi använde oss av två sätt att informera oss om verksamheten, dess funktioner och produktion. För det första intervjuade vi anställda på ett strukturerat sätt och för det andra talade vi informellt med anställda under besöken. Genom att från början medvetet rikta informationssökningen till processidentifikation underlättades det efterföljande arbetet med att identifiera processerna.

En av de viktigaste delprocesserna i SCS verksamhet är reparationsprocessen, som utgör en del av nyckelprocessen orderhantering. Reparationsprocessen betraktas som en primär process, eftersom den måste ingå i orderhanteringsprocessen för att den ska kunna fullföljas

och för att SCS ska kunna lyckas med att leverera det kundvärde som den processororienterade ansatsen syftar till.

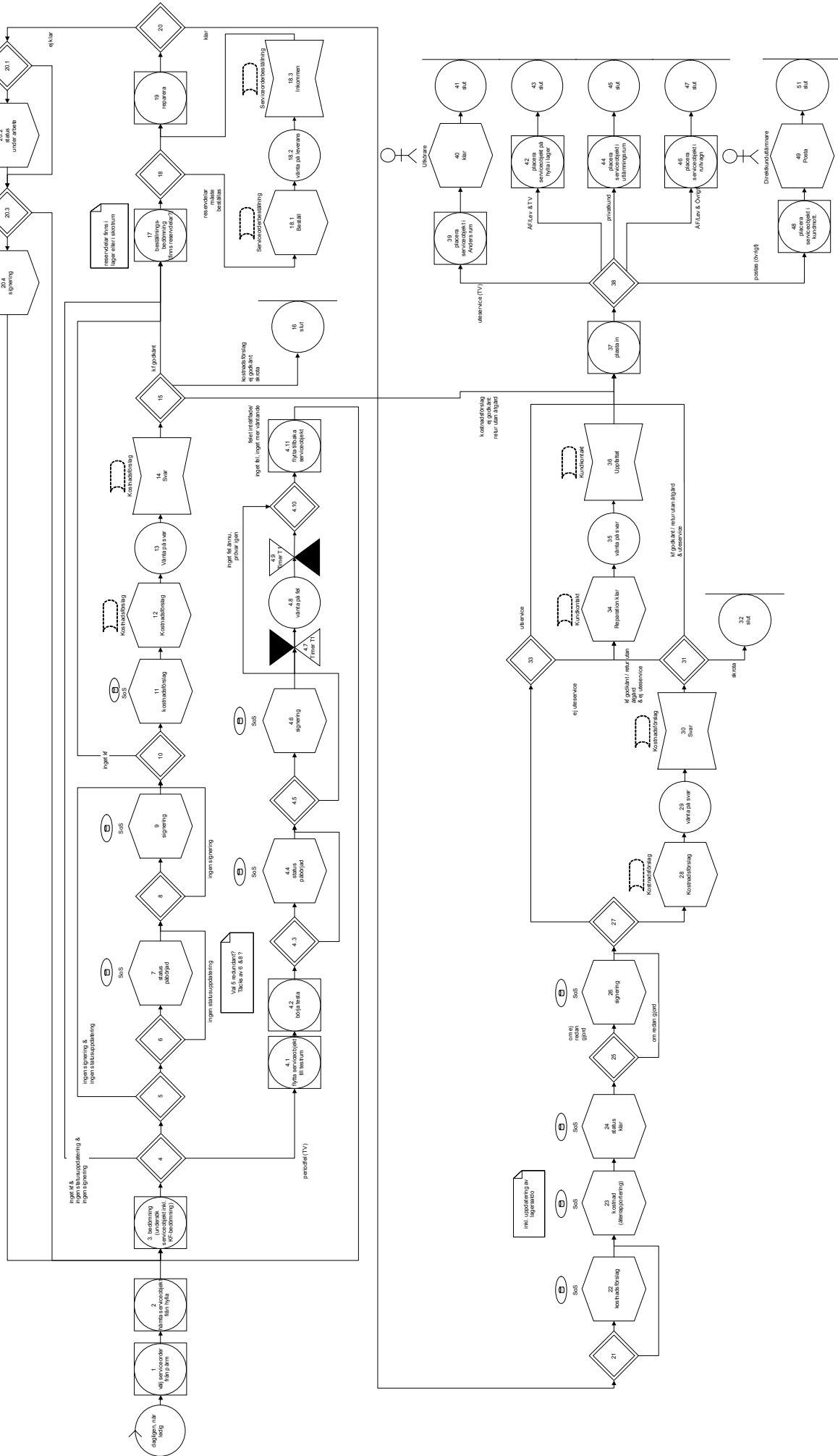
Figur 35 visar hur hela reparationsprocessen i nuvarande situation ser ut när den modelleras i modelleringspråket EBML. Modellen visualiserar i vilken ordning som meddelanden sänds och tas emot och i vilken ordning aktiviteter sker, vilket därför representerar reparationsprocessens kontrollflöde.

Reparationsprocessens huvudaktör utgörs av rollen reparatör och den är en av de processer som startar slumpmässigt. Reparation sker dagligen och initieras så fort en reparatör är ledig och denne vill initiera processen. Processen börjar med tre manuella aktiviteter som visar att reparatören väljer ut en serviceorder, hämtar serviceobjektet och därefter gör en första bedömning av reparationen som kommer att krävas. Beroende på utslaget av bedömningen karaktäriseras reparationsprocessen därefter av att det finns tre beslutsvägar beroende på typ av reparation. Den första typen är en enkel trettiominuters reparation som görs på direkten. Den andra typen är en reparation som innebär att serviceobjektet måste sättas i ett så kallat långkörningstest för att undersöka vad som är fel. Detta långkörningstest tidsbegränsas av en timer. Den tredje typen är en normal reparation, vilket innebär dels att reparatören vet vilket felet är, dels antar att den uppskattade reparationstiden är längre än trettio minuter.

I modellen följer sedan ett antal meddelanden, val och aktiviteter som bland annat beror på om reparatören signerat och skrivit in status i datorsystemet och därmed uppdaterat den datoriserade serviceordern. Denna rutin utförs dock inte på samma sätt och vid samma tillfälle bland de olika teknikerna på SCS, vilket resulterar i en komplex EBML-modell eftersom modellen anger vid vilka tillfällen som teknikerna kan välja att kommunicera med datorsystemet.

Vidare finns i modellen ett antal in- och ut-APM, som beror på att det är reparatören som är huvudaktör i de anropade processerna. I praktiken startas processerna därför av honom själv och därför används APM. APM sänds till processerna Kostnadsförslag, Serviceorderbeställning och Kundkontakt (numrerade 12, 18.1, 28 och 34). När väl reparationen är klar är det också reparatörens ansvar att plasta in och placera serviceobjektet på sin speciella utlämningshylla. Placeringen beror av typ av service och illustreras av valet och de avslutande manuella aktiviteterna i EBML-modellen.

2.1.1 Reparation (TV, Kamera & Övriga apparater)  
Reparatör



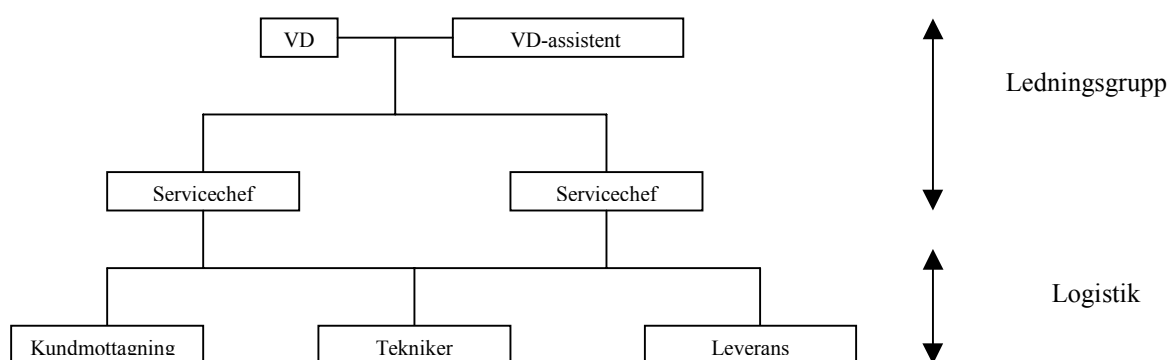
Figur 35. En dynamisk EBML-modell som visualiserar reparationsprocessen på SCS.

### 6.2.3 Tillämpning av en rollmatrix

En rollmatrix definierar de typer av anställningstitlar och roller som kan identifieras i verksamheten. Rollmatrisen redogör för hur olika anställningstitlar relateras till olika roller. Rollerna representerar de dynamiska EBML-modellernas olika huvudaktörer.

När vi skapade rollmatrisen använde vi oss av EBML-modellerna och en karta över organisationsstrukturen. Kartan skapade vi i samband med sammanställningen av datainsamlingen av SCS. För att få en klarare bild av rollfördelningen valde vi också att vid ett av besöken på verkstaden be en person med chefstitel förklara organisationsstrukturen på SCS, inkluderat anställningstitlar och vilka huvuduppgifter respektive titel innefattade. Vi bad chefspersonen också förklara vilka uppgifter utöver huvudsakligt arbetsansvar som eventuellt kunde utföras av en person med en viss anställningstitel. Detta verifierades sedan i personliga intervjuer med anställda. Processmodellerna gav underlag till definition av huvudaktörer och därmed de roller som var aktuella att knyta till specifika anställningstitlar i rollmatrisen.

I Figur 36 visas kartan över SCS organisationsstruktur, som vi utnyttjade när vi definierade de olika anställningstitlarna i företaget.



Figur 36. Organisationsstrukturen på SCS.

I Figur 37 visas den rollmatrix som representerar roller per anställningstitel. Längs x-axeln definieras de sex olika anställningstitlarna; VD, VD-assistent, servicechef, kundmottagare, tekniker och lageransvarig. Längs y-axeln definieras de fjorton olika roller som vi kunnat identifiera i verksamheten. Anställningstitlarna kan alltså ha flera roller, vilket illustreras av de kryssade rutorna.

En anställds ordinarie uppgifter representeras av fetstilta kryss. Icke-ordinarie uppgifter representeras av kryss inom parenteser. Alla anställningstitlar utom kundmottagaren har minst tre ytterligare roller förutom sin ordinarie roll i organisationen. Det är en följd av att SCS organisationsstruktur är mycket flexibel och att arbetet delegeras så fort det blir överbelastning i någon funktion.

Skapandet av en rollmatrix medförde att det blev enklare att identifiera huvudaktörer i EBML-modellerna eftersom sammanställningen av rollfördelningen i rollmatrisen underlättade överblicken och därmed förståelsen för de olika rollernas funktion och relation till de olika anställningstitlarna som finns på SCS.

	VD	VD-assistent	Servicechef	Kundmottagare	Tekniker	Lageransvarig
Direktkundmottagare	(X)	(X)	(X)	X		
Lagermottagare			(X)		(X)	X
Registrerare	(X)	(X)	(X)	X	(X)	(X)
Reparatör			(X)		X	(X)
Direktkundutlämnare	(X)	(X)		X		
Lagerutlämnare			(X)		(X)	X
Fakturere	(X)	X				
Utservicereparatör					X	
Kundkontaktare		(X)	(X)	X	X	(X)
Arbetsledare	(X)		X			
Ledare	X	X	X			
Beställningsregistrerare			(X)		X	X
Beställningssammanställare						X
Beställningsmottagare			(X)		(X)	X

Ordinarie uppgift                   X  
 Icke-ordinarie uppgift           (X)

Figur 37. En rollmatris över rollfördelningen mellan de olika anställningstitlarna på SCS.

#### 6.2.4 Tillämpning av en statisk processmodell

En statisk processmodell beskriver processerna i ett statiskt tillstånd, som visualiserar systemets struktur istället för dess beteende. Den statiska processmodellen ger en helhetsbild av systemets alla definierade processer genom att modellen visar relationer mellan processer.

Figur 38 visar den statiska processmodell som representerar processtrukturen i SCS verksamhet. Modellen visar hur alla processer interagerar med sin externa miljö, i form av aktörer och andra processer. Genom att titta från vänster till höger i modellen kan flödet, från det att mottagning och inskrivning av ett serviceobjekt till fakturering och betalning av reparationen sker, urskiljas. Notera dock att den logiska ordningen inte alltid går från höger till vänster eftersom tidsaspekten inte representeras i en statisk processmodell. Detta representerar SCS nyckelprocess, orderhantering. Nyckelprocessen består av de primära processerna inlämning, reparation, utlämning och fakturering, vilka i sin tur består av olika varianter beroende på typ av leverans, kund och reparation. De primära processerna stöds av ett antal sekundära processer, som Serviceorderbeställning och Kostnadsförslag. Modellen kan härledas från de dynamiska processmodellerna.





## 6.2.5 Tillämpning av arvshierarki

En arvshierarki illustrerar processmodellerna på olika abstraktionsnivå genom generalisering och specialisering. Syftet med att strukturera verksamhetens processer i hierarkiska modeller är att underlätta arbetet i förändringsanalysen.

I Figur 39 har vi valt att representera en delgruppering av en processhierarkisk modell, som baseras på nyckelprocessen Orderhantering. Processhierarkin i figuren visar att delprocessen Inlämning kan ske på olika sätt på SCS. Vilken variant av inlämningsprocessen som sker beror på typ av leverans i form av mottagning, typ av kund och reparation.

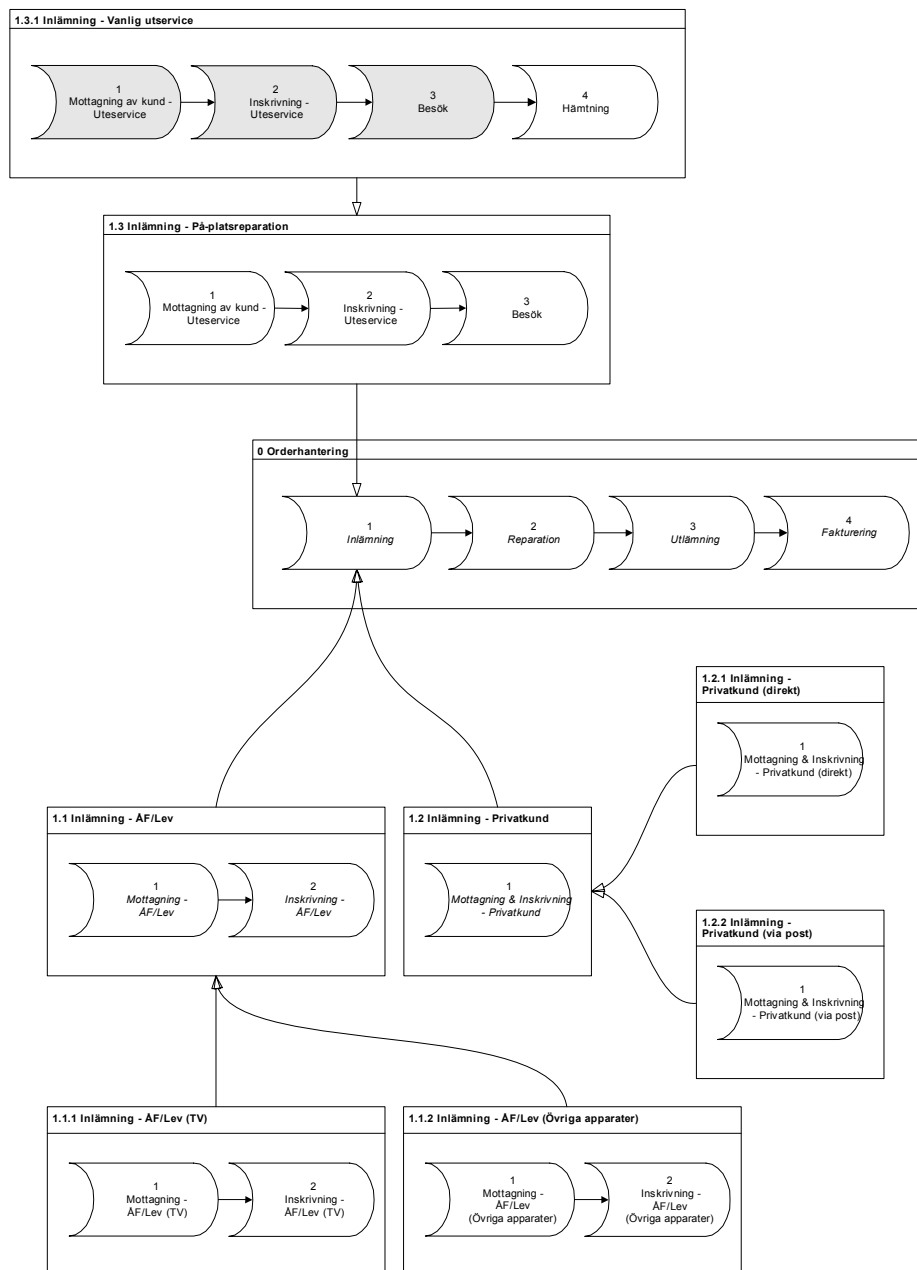
I modellen visas hur den abstrakta processen Inlämning ärvs i form av specialiseringar som består av tre subprocesser. Specialiseringen bestäms av hur den fysiska mottagningen går till, det vill säga om det sker genom leverans till lagret, direkt över kundmottagningsdisken eller om SCS själva hämtar eller reparerar serviceobjektet hos kund genom hembesök. De tre subprocesserna numreras och namnges i modellen som: 1.1 Inlämning – ÅF/Lev, 1.2 Inlämning – Privatkund samt 1.3 Inlämning – På-platsreparation. Dessa har alla i sin tur subprocesser i någon form.

1.1 Inlämning – ÅF/Lev, som sker i lagerlokalen, består av två subprocesser som bestäms av att inlämningsprocessen är olika beroende på om det är TV-apparater eller övriga apparater som levereras till SCS, vilket representeras av entiterna numrerade 1.1.1 respektive 1.1.2. Dessa är specialiseringar av processerna i 1.1 Inlämning – ÅF/Lev vilket är nödvändigt då den är abstrakt.

1.2 Inlämning – Privatkund består också av två subprocesser som bestäms av om kunden lämnar in apparaten personligen eller skickar den via post, vilket representeras av entiterna numrerade 1.2.1 respektive 1.2.2. Dessa är specialiseringar av processerna i 1.2 Inlämning – Privatkund vilket är nödvändigt då den är abstrakt.

1.3 Inlämning – På-platsreparation innebär att kunden besöks och reparationen av serviceobjektet sker direkt på plats. Denna process är superprocess till subprocessen 1.3.1 Inlämning – Pre-reparation – Vanlig uteservice.

1.3.1 Inlämning –Vanlig uteservice innebär att kunden besöks och att serviceobjektet hämtas för att sedan repareras på verkstaden. Själva hämtningen är det enda som skiljer processen från superprocessen 1.3 Inlämning – På-platsreparation. Allt annat ärvs ned utan förändring, vilket visas med gråmarkering.



Figur 39. En representation av hur processhierarkin för den primära processen Inlämning konstruerats enligt de specialiseringar av processen som finns på SCS.

## 6.2.6 Tillämpning av en målmodell

En målmodell konstrueras för att mer konkret beskriva en verksamhets målsättning och de faktorer som påverkar möjligheten att uppfylla målen.

Figur 40 exemplifierar en del av hela den målmodell som vi konstruerade genom analys av den data som vi samlade in under intervjuer med personal på SCS. I figuren finns målmodellens grundläggande komponenter mål, problem och möjligheter representerade. Dessa binds samman av tre olika typer av enkelriktade relationer som visar om komponenterna stödjer, hindrar eller motsäger varandra.

I det här exemplet av SCS målmodell representeras den generella målsättningen Öka intäkter (Mål 26). För att öka förståelsen för målsättningen har vi valt att bryta ner denna målsättning i delmål. Komponenterna som representerar delmål består av två typer av mål, där den första

består av mål som är av strategisk och långsiktig karaktär. Den andra typen består av mål som är av operativ och mer kortsiktig karaktär. Vi har också här valt att lägga till numrering för att ytterligare tydliggöra modellens uppbyggnad.

## Strategiska mål

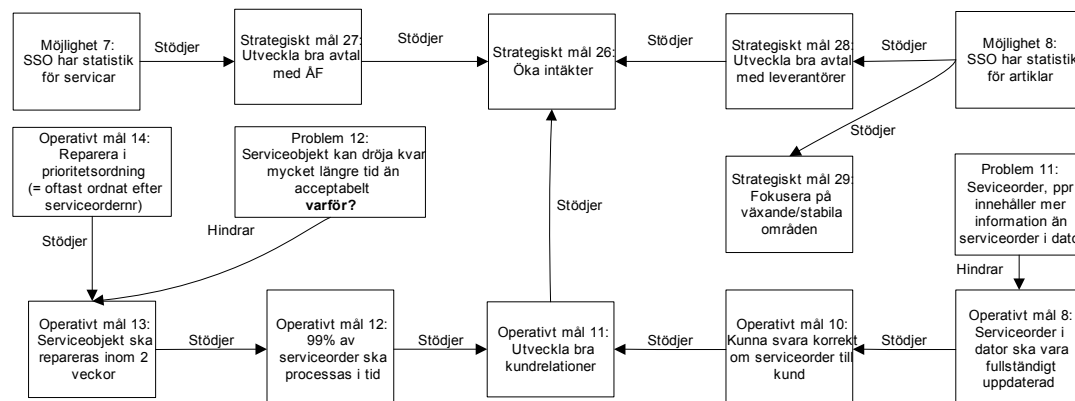
De strategiska målen i modellen består av komponenter som innehåller delmålen för att utveckla bra avtal med återförsäljare respektive leverantörer, som utgör de största kunderna. Till dessa båda mål (Mål 27 och 28) har vi identifierat en möjlighet som kan stödja målet om att utveckla bra avtal. Möjligheten består i alternativet att investera i ett nytt system som kallas StoreSupport Organisation (SSO) som är under utveckling på BlueFront AB. Systemet innehåller funktioner som för statistik över antal tjänster en viss återförsäljare beställt (Möjlighet 7), men även statistik över vilken typ av artiklar som en viss leverantör ofta lämnar in för service (Möjlighet 8). Statistiken ger ett adekvat underlag till formuleringen av avtal, men ger även underlag till målsättningen att kunna fokusera på växande och stabila områden (Mål 29).

Ett tredje strategiskt delmål för att långsiktigt öka intäkterna är målet Utveckla bra kundrelationer (Mål 11), som i sin tur kan brytas ner i ett antal operativa delmål. De operativa delmålen diskuteras nedan.

## Operativa mål

Målet Utveckla bra kundrelationer stöds av ett flertal mer konkreta operativa delmål. I modellen visas ett sådant i det kvantifierbara delmålet som anger att 99% av serviceorder ska processas i tid (Mål 12), som stöds av ett annat operativt mål. Tidsaspekten för att serviceorder ska processas i tid anges som två veckor (Mål 13), som i sin tur stöds av målet att alla serviceorder ska hanteras enligt given prioritetsordning (Mål 14). Två-veckorsgränsen hindras dock av att hanteringen idag tar längre tid och därför är ett problem för att uppfylla målet (Problem 12).

Det operativa delmålet att kunna ge korrekt information om en viss service när kunden kontaktar SCS (Mål 10) förutsätter att kundmottagaren genom datorsystemet får fram en uppdaterad serviceorder, vilket också är ett klart uttalat delmål i verksamheten (Mål 8). Även i det här fallet finns dock ett problem som hindrar målet om att alltid ha fullständigt uppdaterade serviceorder i datorsystemet. Problemet består i att det oftast är det utskrivna pappersexemplaret av en serviceorder som innehåller den uppdaterade informationen, eftersom SCS idag använder sig av både datorsystemet och pärmsystemet i orderhanteringsprocessen (Problem 11).



Figur 40. En del av SCS målmodell som representerar vilka delmål, problem och möjligheter som finns i verksamheten för att uppnå det övergripande målet Öka intäkter.

### **6.2.7 Tillämpning av förändringsanalys**

Syftet med förändringsanalysen är att föreslå förbättringar som ska effektivisera verksamheten och dess processer. I analysfasen ingår två grundläggande steg.

I analysfasens första steg ska processmodellerna analyseras och sättas i förhållande till målmodellen för att klargöra förändringsbehov.

Ett exempel är SCS målsättning att kunna ge korrekt information om en serviceorder till kunder som kontaktar företaget. Samtidigt visar de dynamiska processmodellerna i EBML att det inte uppfylls. Målet förutsätter att serviceorderns status konsekvent uppdateras i datorsystemet, vilket inte är fallet idag då det kan ske vid olika tillfällen i processen och dessutom snarare görs på serviceorderns pappersexemplar. Genom att bortrationalisera alla valmöjligheter för tekniker att föra in status och signering i ett sent skede i processen och istället ange specifika tillfällen för datoriserad inskrivning av signatur och status går det att i högre grad garantera uppdaterad och korrekt statusinformation i datorsystemet.

I analysfasens andra steg ska de arvshierarkiska modellerna analyseras för att kunna föreslå förbättringar som kan kopplas till de likheter och skillnader som en grupp av relaterade processer kan uppvisa.

Ett exempel är de olika typer av inlämningsprocesser som existerar i nuläget. De två typer av processer som avser mottagning av serviceobjekt från återförsäljare eller leverantörer respektive privatkunder visas i processhierarkin som specialiseringar (Subprocess 1.1 respektive 1.2 i Figur 39), men är i själva verket nästan identiska. Skillnaderna ligger i ett antal manuella aktiviteter som tillkommer i de dynamiska processmodellerna för mottagning av leveranser från återförsäljare och leverantörer. Mottagningen sker i det fallet i lagerlokalen, vilket idag kräver ett antal extra förflyttningar mellan de båda mottagningslokalerna då inskrivningen av alla serviceorder i datorsystemet sker i kundmottagningen och inte på lagret.

I den fullständiga målmodellen finns även effektiviseringsmål angivna. För att stödja dessa, föreslår vi en omorganisering. Genom att konsekvent använda en av lokalerna för all typ av mottagning, förslagsvis lagerlokalen som är större, effektiviseras inlämningsprocessen för att den sker på ett och samma ställe och onödiga förflyttningar undviks. Ett annat alternativ är att byta lokaler. Förslaget innebär att inlämningsprocessen blir mer enhetlig, vilket i sin tur skulle förenkla den processhierarkiska modellen.

### **6.2.8 Tillämpning av modellering av förändrade processer**

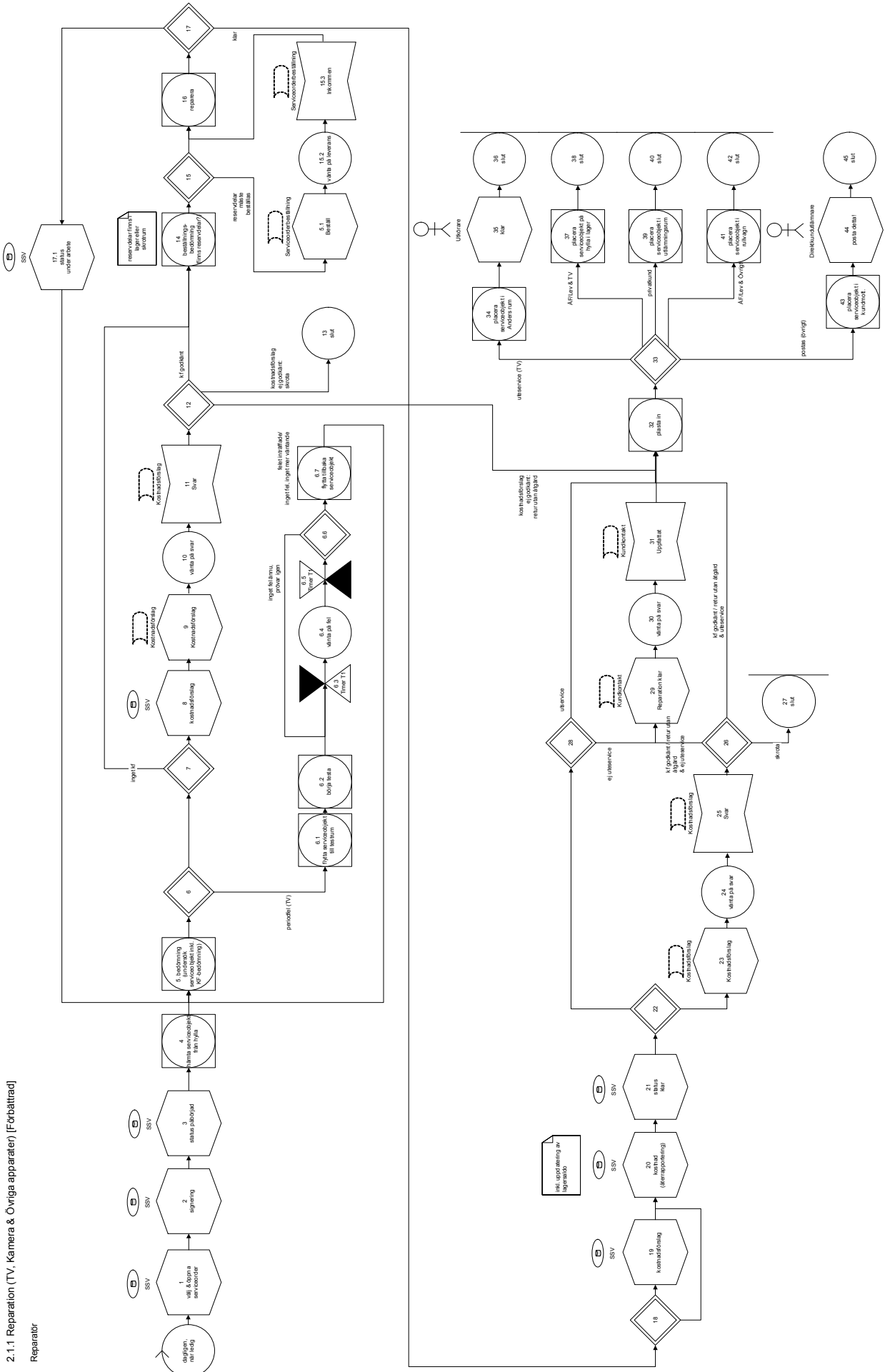
Genom att modellera de förändringar som analysen gett upphov till blir det enkelt att i efterhand jämföra den ursprungliga processen med den nya och se vilka skillnader som en framtida implementation kan medföra.

I Figur 41 visar vi hur vår analys genererade ett antal förändringar av reparationsprocessen, som visas i sin ursprungliga form i Figur 35. Förändringarna resulterade i en mindre komplex reparationsprocess genom automatisering av manuella aktiviteter, men också genom rationalisering som skedde genom eliminering av ett antal överflödiga beslutsvägar. Beslutsvägarna gjorde den ursprungliga processen onödigt komplex och omständlig.

I modellen visas förändringarna redan initialt där vi föreslår att i stället för att service-teknikern väljer vilket serviceobjekt som ska repareras så föreslår applikationen i datorsystemet detta med en lista på serviceorder ordnade i prioritetsordning. Teknikern slipper därmed att manuellt bläddra i pärmar för att välja serviceobjekt. Därefter signerar och statusmärker teknikern den serviceorder som han valt. Det ska ske konsekvent enligt det förslag vi

anger i analysen och i modellen, vilket medför att de onödiga beslutsvägarna som tidigare komplicerade processen försvinner.

2.1.1 Reparation (TV, Kamera & Övriga apparater) [Förbättrad]  
Reparatör



Figur 41. En dynamisk EBML-modell som visualiserar den förändrade reparationsprocessen på SCS.

## 7 Utvärdering

I det här kapitlet redovisas resultatet av utvärderingen av processmodelleringspråket EBML och arbetssättet. Kapitlet är indelat i fyra huvudavsnitt. I det inledande avsnittet förklarar vi varför och hur utvärderingen utförts. I de följande två avsnitten presenteras och sammanfattas svaren på intervjufrågorna om EBML, arbetssätt respektive generellt intryck. I det fjärde och sista avsnittet diskuteras resultatet av utvärderingen.

### 7.1 Anledning och utförande

Anledningen till att vi valt att inkludera en utvärdering i arbetet är att vi ville komplettera resultatet från fallstudien med en återkoppling från personer med extern anknytning till utvecklingen av EBML, det vill säga personer utan förkunskap om språket. Tanken var att dessa personers reflektioner skulle ge förslag på eventuella framtida förändringar av EBML.

Målet med utvärderingen var att verifiera begriplighet respektive användbarhet av EBML och de modeller som ingår i det arbetssätt vi föreslår för kartläggning av en verksamhet. För att kunna göra detta definierade vi först ett antal kriterier för begriplighet respektive användbarhet.

Kriterier för begriplighet gällde framför allt följande punkter:

- Att det enkelt går att förstå symbolerna
- Att det är möjligt att förstå vad en viss modell eller flera modeller tillsammans representerar
- Att det är möjligt att förstå en viss modells funktion i helheten

Kriterier för användbarhet gällde framför allt följande punkter:

- Att all nödvändig information finns representerad
- Att modelleringspråket inte är för komplext i relation till dess användningsområde
- Att det är lämpligt att inkludera modellerna i arbetssättet
- Att modellerna och arbetssättet uppfyller sina syften

Vi valde att utföra utvärderingen i form av strukturerade intervjuer, som innehöll frågor som skulle avgöra om de fördefinierade kriterierna uppfylldes. Vi valde att dela in intervjun i tre delar, som i sin tur delades upp i frågor som först behandlade begriplighet och sedan användbarhet.

Den första delen av intervjun fokuserade på utvärdering av processmodelleringspråket EBML och dess tre modelltyper: dynamisk, statisk och arvshierarkisk modell. Den andra delen behandlade arbetssättet för EBML, inkluderat de modeller som enligt förslaget bör användas vid kartläggning av en verksamhet. Den tredje delen innehöll frågor om hur EBML och arbetssätt uppfattades generellt, det vill säga om modellerna och arbetssättet uppfyller sitt syfte att beskriva en verksamhets nuläge respektive möjligheten att utvärdera eventuella förändringsbehov. För att slippa onödiga upprepningar har vi valt att integrera resultatet från den tredje delen av intervjun med presentationen av resultatet för EBML respektive arbetssätt.

I valet av vilka personer som skulle ingå i utvärderingen bestämde vi oss för att vi ville få synpunkter från personer med god kunskap om verksamheten Stockholms Central Service AB och från personer med goda kunskaper om verksamhetsmodellering. Fördelen med detta var att vi skulle få synpunkter från personer med olika bakgrund och olika kunskapsnivå. I ett projekt av det här slaget är ofta personer med sådana bakgrunder inblandade. Utvärderingen genomfördes med tre nyckelpersoner. Det hade varit önskvärt att ha ett större urval, men på



grund av tidsbrist begränsade vi oss till följande personer: en person med ledarposition på SCS, en modellerare från BlueFront AB samt en programmerare från BlueFront AB. Intervjufrågorna finns återgivna i Appendix 2.

## **7.2 Processmodelleringspråket EBML**

### **7.2.1 Dynamisk processmodell**

Alla tre intervjuade personer svarade att symbolerna i notationen och syntaxen var begripliga, med viss reservation från programmeraren som menade att de enskilda symbolerna först måste läras in. Modelleraren påpekade också att vissa symboler kunde vara svåra att förstå, till exempel de artificiella processmeddelandena (APM). Däremot sade alla att det var bra att antalet symboler var få, eftersom det underlättade inläring och förståelse och därmed förmodligen också användning av EBML.

På frågan om symbolerna var intuitiva svarade representanten från SCS att de var logiska eftersom skillnaderna var tydliga. Programmeraren svarade nej och påpekade återigen inläring som förutsättning för förståelse. Modelleraren svarade att det var bra med likhet mellan symboler som hade samband till varandra. Exempelvis har både manuellt val och manuell aktivitet dubbel inramning, vilket var intuitivt och därmed lättare att lära in, enligt modelleraren. Vidare menade representanterna från BlueFront att förståelsen av den exemplifierade EBML-processmodellen förutsätter viss bakgrundsinformation om verksamheten, men att den dynamiska modellen annars var fullt begriplig.

På frågor om användbarhet saknade ingen av de intervjuade symboler för fenomen eller entiteter som borde vara möjliga att modellera. Däremot ansåg modelleraren att manuella val behöver en explicit förklaring av vad som styr det manuella valet, till exempel genom att fästa en anteckning med en streckad linje till det specifika manuella valet. Ett alternativ skulle kunna vara att införa olika symboler för olika typer av manuella val, det vill säga någon slags kategorisering av manuella val. Programmeraren svarade att han eventuellt saknade någon form att talaktskategorisering av in- och utmeddelanden, men nämnde också att det kanske inte var användbart i alla typer av projekt, eftersom det är en komplex teori.

Ingen av de intervjuade tyckte att den dynamiska processmodellen var onödigt komplext för syftet att kartlägga och dokumentera en organisations processer, främst för att språket innehåller så få symboler. Representanten från SCS sade emellertid att modellerna ger en lättöverskådlig dokumentation till skillnad från en rapport skriven i ett naturligt språk. Han ansåg att modeller är lättare att begripa och därför användbara som diskussionsunderlag.

### **7.2.2 Statisk processmodell**

Alla tre ansåg att det var svårare att förstå den statiska processmodellen än den dynamiska. Både representanten från SCS och modelleraren från BlueFront svarade att modellen var komplex och svår att förstå. Framför allt var det svårare att urskilja flödet i modellen, åtminstone vid en första anblick. Samma personer svarade också att det kanske hade underlättat om den statiska modellen avbildats i ett större format, exempelvis A3, eftersom modellen är omfattande och komplex.

På frågor om användbarhet menade alla att tanken att modellen ska visa hur processerna förhåller sig till varandra är god, eftersom det ger en överblick över alla processer och deras relationer, med viss reservation för att risken för att modellens komplexitet ökar med större verksamheter. Vidare svarade personen från SCS att den statiska modellen dessutom skulle kunna användas som verktyg vid ISO-certifiering. Han menade att varje process skulle kunna

kompletteras med en textuell beskrivning med hjälp av den numrering som identifierar respektive process i modellen. På frågan om de saknade någon modelleringssymbol för fenomen eller entiteter i den statiska modellen svarade alla tre nej.

### **7.2.3 Arvshierarkisk modell**

Den processhierarkiska modellen är begriplig, enligt svaren från de intervjuade. Representanten från SCS tyckte att notationen var tydlig med enkla pilar, vilket underlättade förståelsen för vad som var en specialisering. Såväl modelleraren som programmeraren var sedan tidigare bekanta med arv och den objektorienterade paradigmen, vilket förenklade tanken bakom modellen och dess funktion. Modelleraren och programmeraren var speciellt positiva till färgkodningen, det vill säga att det som ärvs gråmarkeras i modellen och därför tydliggör likheten mellan super- och subprocesser.

Vad gäller användbarhet var alla tre överens om att den processhierarkiska modellen visar arv och specialisering på ett användbart sätt. Personen från SCS svarade att notationen är väldigt grundläggande och inte för komplex. Däremot föreslog han att den processhierarkiska modellen skulle visas före den statiska processmodellen, eftersom han ansåg att den var en slags förenkling av den statiska modellen. Processhierarkin gav enligt honom en bättre första överblick av existerande processer och flöden dem emellan, samtidigt som specialiseringarna illustrerades. Detta skulle även medföra en introduktion och förberedande förståelse för de detaljerade dynamiska processmodellerna. Modelleraren och programmeraren uttryckte dessutom återigen färgkodningens användbarhet för att ytterligare tydliggöra processernas likheter och skillnader. På frågan om de saknade någon modelleringssymbol för fenomen eller entiteter som borde ingå i den processhierarkiska modellen svarade alla tre nej.

## **7.3 Arbetssätt för EBML**

### **7.3.1 Nulägesbeskrivning**

#### **Konceptuell modell**

Alla tre svarade att symbolerna i notationen för den konceptuella modellen generellt sett är begripliga. Representanten från SCS, som till skillnad från de båda personerna från BlueFront inte kände till modelleringssättet sedan tidigare, hade inga problem att förstå notationen. Han tyckte det var bra att det var så få typer av symboler och relationer. Programmeraren svarade som tidigare att språket måste läras in innan det går att förstå. Alla var också överens om att den konceptuella modellen representerar informationen på ett sätt som lätt går att förstå. Modelleraren påpekade emellertid att det är viktigt att förstå vad den konceptuella modellen exakt ska representera innan den skapas.

Ur användbarhetssynpunkt var det ingen av de tre som ansåg att modellen är för komplex vare sig för syftet att enbart kartlägga och dokumentera en verksamhet eller syftet att använda sig av modellen för att utröna eventuella förändringsbehov. Personen från serviceverkstaden svarade att modellen borde vara användbar som en slags koppling mellan de övriga modellerna, det vill säga som en slags kontroll för att de övriga modellerna täcker in det som representeras i den konceptuella modellen. Samtidigt påpekade han att den konceptuella modellen av serviceverkstaden kändes onödigt eftersom han har så god förkunskap om verksamheten. Därför ansåg han också att det var bra att steget att skapa en konceptuell modell i arbetsmetoden är frivilligt. Modelleraren svarade att den konceptuella modellen kan användas som en ordlista för att undvika tvetydigheter i begrepp som används under arbetets gång, eftersom utvecklingen av den innebär att olika entiteter definieras och namnges från

början. Programmeraren sade att abstraktionsnivån tillåts variera beroende på behovet av modellen. Ingen av de intervjuade ansåg sig sakna någon modelleringsymbol i den konceptuella modellen.

### **EBML-modeller**

Begripligheten av de tre typerna av EBML-modellerna finns presenterade i föregående avsnitt, eftersom dessa frågor ställdes under utvärderingen för processmodelleringspråket EBML.

EBML-modellerna ger en god nulägesbeskrivning av en verksamhets processer och är därför användbara för detta syfte enligt de tillfrågade. Personen från serviceverkstaden ansåg att de dynamiska EBML-modellerna var speciellt bra som underlag för syftet att utröna och kartlägga vilka förändringsbehov som existerar, medan den statiska EBML-modellen skulle vara användbar som dokumentationsverktyg i exempelvis en ISO-certifiering. Modelleraren menade att EBML-modellerna generellt ger en bra översikt av en verksamhets existerande processer och därför ett användbart underlag för diskussion av förändringar. Programmeraren sade att de dynamiska EBML-modellerna är speciellt användbara eftersom de ger ett gott stöd för förståelsen av verksamhetens processer, vilket därför är användbart i hans arbete som programmerare.

### **Rollmatris**

Alla tre svarade ja på frågor om rollmatrisen och den information den representerar var begriplig. Representanten från SCS föredrog emellertid ringar och ifyllda ringar i stället för fetstilta kryss respektive kryss inom parentes som markörer för de olika rollernas ordinarie och icke-ordinarie uppgifter. Han ansåg att det tydligare visade skillnaden mellan dem. Modelleraren tyckte kryssen var mycket intuitiva och förstod funktionen utan att ens ha läst beskrivningen för kryssen först.

Kriterierna för användbarhet uppfylldes genom att alla de intervjuade svarade att rollmatrisen är enkel att använda för syftet att dokumentera de roller och deras uppgifter som kan identifieras i en verksamhet. Personen från SCS svarade att den är lätt att anpassa och utveckla efter behov eftersom den tillåter användaren att utforma matrisen efter eget tycke, samt att den är mycket användbar då förändringar blir aktuella och framtida tillägg eller borttagningar av roller därför kan komma att ske. Ingen sade sig sakna något i rollmatrisen.

## **7.3.2 Förändringsanalys**

### **Målmodell**

Svaren till alla fyra frågor om begriplighet av målmodellen var genomgående positiva. Alla tre tyckte att målmodellens symboler och notation var enkla att förstå. Likaså tyckte alla tre att målmodellen visar både målsättning, problem och deras relationer på ett sätt som lätt går att förstå. Personen från SCS tyckte dock att det kunde var önskvärt att kanske använda sig av olika färger eller former för de entiteter som representerar mål, problem respektive möjligheter. Det skulle ytterligare tydliggöra informationen som målmodellen representerar. Han var också mycket positiv till tanken att försöka dela upp målen i strategiska respektive operativa mål, liksom tanken att uttrycka mål och problem i hela meningar och som något mätbart. Både modelleraren och programmeraren diskuterade i samma banor, eftersom dessa faktorer underlättar förståelsen för modellen.

Svaren som gällde användbarhet var också genomgående positiva. Alla tre tyckte att målmodellen beskriver en organisations framtida och önskade tillstånd på ett bra och

användbart sätt. Alla var också överens om att det är lämpligt att använda målmodellen som en utgångspunkt vid diskussion kring och analys av hur existerande problem skulle kunna lösas. På frågan om målmodellen är lämplig att inkludera i arbetsmetoden för verksamhetsmodellering med EBML svarade alla ja. Modelleraren sade att utvecklingen av målmodellen också ger kunskap om en organisations värderingar, vilket kan vara bra för att förankra förslag om eventuella förändringar som konsult.

### **Förändringsanalys**

På frågan om det är lätt eller svårt att förstå förhållandet mellan de olika modellerna, gavs svar som pekade åt samma håll. Representanten från SCS svarade att det åtminstone till en början är svårt att sätta alla modeller i förhållande till varandra, men att det går att urskilja konkreta samband mellan olika modeller. Dessutom påpekade han att det tar mer tid att i efterhand sätta sig in i redan färdiga modeller. Det skulle säkert underlättat om han själv deltagit i modelleringsprocessen. Både modelleraren och programmeraren tyckte att det generellt gick att förstå förhållandet mellan de olika modellerna, men sade precis som representanten från SCS att det är svårare att sätta modellerna i förhållande till varandra i efterhand, utan att själva ha deltagit i modelleringsarbetet. Modelleraren nämnde även att förståelsen för de olika modellernas funktion skulle öka med erfarenhet.

På frågan om förslaget att jämföra modellerna och sätta dem i förhållande till varandra är en användbar utgångspunkt vid identifiering av problem och förändringsbehov gavs varierande svar. Personen från SCS svarade att det berodde på vilken modell som står som utgångspunkt, det vill säga om det var målmodellen eller processmodellerna eller som tredje alternativ, där båda modellerna var likvärdiga och användes i en iterativ arbetsgång. Han föredrog ur användbarhetssynpunkt det iterativa alternativet. Vidare sade han att modellernas användbarhet också berodde på syftet med kartläggning och verksamhetsmodellering, men menade dock att han föredrar modeller framför beskrivningar utförda i ett naturligt språk.

Modelleraren var mycket positivt till förslaget. Att konstruera en målmodell innan själva analysen tar vid, var enligt honom mycket användbart för att öka kunskapen om vilka problem som existerar. Dessutom tillade modelleraren att alla modeller kompletterar varandra och därför ger en god verksamhetsbeskrivning som utgångspunkt vid analys.

Programmeraren svarade liksom modelleraren att modellerna tillsammans ger en bra verksamhetsbeskrivning, men menade att modellernas användbarhet beror på vilka som ska delta i förändringsprocessen. Modelleringen resulterar i många modeller och han frågade sig om de är intressanta för alla deltagare. Han menade att det i så fall skulle vara bättre att använda sig av en enklare syntax eller högre abstraktionsnivå beroende på vem modellerna egentligen vänder sig till. Vidare sade programmeraren att vissa modeller eller delar av modeller kanske inte berör alla deltagare. Endast de delar som en viss projektdeltagare berörs av, kan brytas ut och visas. Exempelvis skulle de mål som berör teknikerna på serviceverkstaden brytas ut för att ge dem ett diskussionsunderlag i enskild grupp.

På frågan om onödig redundans existerar i den information som de olika modellerna representerar svarade alla tre nej.

På frågan om EBML och arbetsmetoden uppfyller sitt syfte, det vill säga beskriva en verksamhets nuläge respektive möjligheten att utvärdera förändringsbehov svarade alla tre ja.

## **7.4 Diskussion**

Resultatet av utvärderingen visar generellt att kriterierna för såväl begriplighet som användbarhet uppfylls för EBML liksom för arbets sättet vi föreslår. Alla tre som intervjuades

var positiva till att använda EBML och arbetssättet vid kartläggning av en verksamhets processer även i andra projekt. Resultatet förstärks dessutom av att representanten från SCS, som inte hade någon tidigare erfarenhet av verksamhetsmodellering, inte hade några större problem att förstå vare sig modelleringspråken eller modellernas funktion.

Som alltid är resultatet av utvärderingar beroende på urvalet; i det här fallet vilka som intervjuades. Det är möjligt att resultatet inte skulle blivit detsamma med andra personer. Samma sak gäller för valet av strukturerade intervjuer som datainsamlingsmetod. Vad gäller de intervjuade personerna är det möjligt att deras åsikter och synpunkter skulle varit annorlunda om de fått ett testfall att modellera, istället för att redan färdiga modeller presenterades för dem. De intervjuades korta erfarenhet av EBML kan ha legat till grund för att samtliga svarade nej på frågorna om någonting saknades i modellerna. Det är möjligt att de med större kännedom om EBML svarat annorlunda.

## 8 Slutsatser

Uppsatsen har presenterat en utökad version av det meddelandeorienterade språket BML så att det ska kunna användas för verksamhetsmodellering och analys i ett systemutvecklingsprojekt. Den utökade versionen kallas Extended BML (EBML) och uppsatsen redogör för notation och syntax i EBML, samt ett arbetssätt för hur EBML ska användas i verksamhetsmodellering och analys.

Uppsatsen presenterar också en utvärdering av EBML och arbetssättet. Denna utvärdering har påvisat vissa svagheter. Det borde bland annat vara enklare att sammankoppla de olika modelltyperna som ingår i arbetssättet och det är svårt att urskilja flödet i den statistiska EBML-modellen, som lätt blir komplex. Vidare gav utvärderingen förslag på förbättringar, till exempel tydligare indelning av målmodellens mål. Även en kategorisering av manuella val skulle kanske underlätta begripligheten av den dynamiska EBML-modellen.

Trots dessa svagheter anser vi att vi att syftet med EBML och dess arbetssätt uppnåtts – att lösa de begränsningar BML uppvisat med avseende på verksamhetsmodellering och förändringsanalys. Vi har bland annat genomfört en verksamhetsmodellering och förändringsanalys av en serviceverkstad och kunnat framställa en dokumentation som är begriplig och givande för olika intressenter. Det har bland annat utvärderingen givit stöd för, då samtliga intervjuade anser att modellerna och arbetssättet uppfyllt sitt syfte.

## 9 Framtida forskning

För framtida forskning är det önskvärt att ytterligare testa EBML i olika typer analysprojekt för att ta reda på om EBML och dess arbetssätt verkligen täcker allt som det avser göra inom verksamhetsmodellering. Det bör även genomföras tester för att se hur väl det faktiskt går att använda EBML tillsammans med BML i ett och samma projekt.

Användandet av EBML och dess arbetssätt skulle förenklas om det fanns datoriserade modelleringsverktyg som stöder dess syntax och semantik. Kopplingar mellan såväl olika modelltyper som inom samma modelltyp skulle kunna definieras exakt och på sätt minskar risken för att något missas.

I presentationen av BML utelämnades, som tidigare nämnts, vissa områden: datamodellering av meddelanden och meddelandeklassificering. Dessa är dock intressanta områden i sig värda vidare forskning. En detaljerad beskrivning av exakt vad varje meddelande innehåller skulle till exempel kunna förenkla vid datorisering av manuella rutiner. Den meddelande-

klassificering som gjordes av Johannesson & Perjons (2000) var specifik för applikationsintegration. Det vore önskvärt att en mer generell klassificering togs fram – en klassificering som är enklare att använda än den ursprungliga från talaktsteorin. En annan inriktning skulle vara att ta fram en speciell klassificering för verksamhetsmodellering.

## 10 Tack

Vi vill rikta stor tacksamhet till vår handledare Erik Perjons på Institutionen för data- och systemvetenskap (DSV), Stockholms Universitet. Eriks råd, stöd och entusiasm har varit ovärderligt under hela uppsatsarbetet.

Vi vill också tacka de personer på BlueFront AB, Stockholms Central Service AB och DSV som ställt upp på intervjuer och bidragit med information som i flera avseenden möjliggjort uppsatsen.

## 11 Referenser och källförteckning

### Tryckta källor

- (Andersen et al, 1994) Andersen, E.S., Grude, K.V., Haug, T. *Målinriktad projektstyrning*, Studentlitteratur Lund, 1994.
- (Andersen, 1994) Andersen, E.S., *Systemutveckling*, 1994, Studentlitteratur.
- (Belina, 1991) Belina, F., Hogrefe, D., Sarma, A., *SDL with applications from protocol specification*, 1991, Carl Hanser Verlag och Prentice Hall International.
- (Bubenko, Brash, Stirna, 1998) Bubenko, J., Brash, D., Stirna, J., *EKD User Guide*, Elektra Electrical Enterprise Knowledge, 1998, DSV, KTH och Stockholms universitet.
- (Davenport, 2000) Davenport, T.H., *Mission Critical*, 2000, Harvard College.
- (Fowler, 1997) Fowler, M., *UML Distilled*, 1997, Addison Wesley Longman, Inc.
- (Hammer, 1996) Hammer, M., *Beyond reengineering*, 1996, Harper Collins Publishers, Inc. Svensk översättning ISL Förlag AB, 1997.
- (Jensen, 1992) Jensen, K., *Coloured Petri Nets, Basic concepts Volume 1*, 1992, Springer-Verlag Berlin Heidelberg New York.
- (Malone et al, 1999) Malone, T.W., Crowston, K., Lee, J., Pentland, B., Dellarocas, C., Wyner, G., Quimby, J., Osborn, C.S., Bernstein, A., Herman, G., Klein, M., and O'Donnell, E., *Toward a handbook of organizational processes*, Management Science 45(3) sid 425-443, mars 1999
- (Peterson, 1977) Peterson, J.L., *Petri nets*, Computing Surveys, vol. 9, no. 3, 1977.
- (Pettersson & Wäyrynen, 2001) Pettersson, P., Wäyrynen, J. *Verksamhetsöversikt - Stockholms Central Service AB*, 2001.
- (Reisig, 1985) Reisig, W., *Petri Nets An Introduction*, 1985, Springer-Verlag Berlin Heidelberg.
- (Rumbaugh, Jacobson, Booch, 1999) Rumbaugh, J., Jacobson, I., Booch, G., *The Unified Modeling Language Reference Manual*, 1999, Addison Wesley Longman, Inc.

- (Schmuller, 1999) Schmuller, J., *SAMS Teach Yourself UML in 24 Hours*, 1999, Sams Publishing.
- (Svantesson, 2000) Svantesson, B., *Dynamic Process Management in SDL to VHDL Synthesis*, 2000, Royal Institute of Technology, Kista Snabbtryck AB.

### Elektroniska källor

- (Johannesson & Perjons, 2000) Johannesson, P. & Perjons, E. *Design Principles for Application Integration*, 2000 i Wangler, B., Bergman L. (Eds.) *Advanced Information Systems Engineering*, 12th International Conference CAiSE 2000, Stockholm, Sweden, June 5-9, 2000, Proceedings. Lecture Notes in Computer Science. Volym 1789, 2000, Springer, sid 212-231, <http://www.dsv.su.se/~perjons/caise226.pdf> (2001-07-22)
- (Svenska akademiens ordbok, 2001) <http://g3.spraakdata.gu.se/saob/> (2001-07-22)
- (ViewLocity, 2001) <http://www.viewlocity.com/> (2001-07-22)

### Muntliga källor

- (Andersson, 2001) Hans Andersson, BlueFront AB. 2001-07-03.
- (Barosen, 2001) Tony Barosen, Stockholm Centralservice AB. 2001-02-07, 2001-03-20, 2001-07-03.
- (Eriksson, 2001) Magnus Eriksson, Stockholm Centralservice AB. 2001-03-20.
- (Jansson, 2001) Kent Jansson, Stockholm Centralservice AB. 2001-02-07.
- (Johannesson, 2001) Paul Johannesson, Institutionen för Data- och Systemvetenskap, Stockholms Universitet. 2001-05-17.
- (Lundevall, 2001) Per Lundevall, BlueFront AB. 2001-07-03.
- (Perjons, 2001) Erik Perjons, Institutionen för Data- och Systemvetenskap, Stockholms Universitet. 2001-01-14 till 2001-05-17.
- (Raage, 2001) Ali Raage, Stockholm Centralservice AB. 2001-02-07.

### Bibliografi

- (Billington, Diaz, Rozenberg, 1999) Billington, J., Diaz, M., Rozenberg, G., *Application of Petri Nets to Communication Networks*, 1999, Springer-Verlag Berlin Heidelberg.
- (Davidsson & Patel, 1994) Davidsson, B., Patel, R., *Forskningsmetodikens grunder*, 1994, Studentlitteratur.
- (Gogolla, 1994) Gogolla, M., *An extended Entity-Relationship Model*, 1994, Springer-Verlag Berlin Heidelberg.

- (Holm, 1999) Holm, P., *En orientering i objekt och Java*, 1999, Studentlitteratur, Lund.
- (Thurén, 1991) Thurén, T., *Vetenskapsteori för nybörjare*, 1991, Liber AB.



## **Appendix 1. Grundläggande begrepp**

### **Affärsprocess**

En affärsprocess är en samling av organisationsaktiviteter som tillsammans skapar ett värde för kund. (Davenport, 2000)

### **Applikationsintegration**

Applikationsintegration motsvarar engelskans Enterprise Application Integration (EAI). EAI innefattar ett antal teknologier som underlättar integration av IT-system, vilket innebär att individuella IT-system lättare kan överföra information mellan varandra. Exempel på EAI-teknologier är CORBA, DCOM, Enterprise Java Beans (EJB), XML, meddelandemäklare och processmäklare.

### **Arvshierarkisk modell**

Den arvshierarkiska modellen är en av de tre typer av grafiska modeller som finns i EBML. Den har ingen motsvarighet i moderspråket BML. Syftet med att skapa en arvshierarki av processer är att strukturera verksamhetens processer i hierarkiska modeller för att underlätta arbetet i förändringsanalysen. Skapandet av arvshierarkin gör att de modellerade processernas skillnader och likheter explicitgörs.

Arvshierarki är ett samlingsbegrepp för de två olika vyerna som används för att beskriva en arvshierarki: processhierarkin (se under Processhierarki) och de dynamiska processmodellerna med arvsinformation (se under Dynamisk processmodell med arv).

### **Delprocess**

Delprocess är ett generellt samlingsbegrepp för primära och sekundära processer, eftersom de utgör delar av en nyckelprocess.

### **Dynamisk processmodell**

Den dynamiska processmodellen är en av de tre typer av grafiska modeller som finns i EBML. Det motsvarar den dynamiska processmodellen i moderspråket BML. Processmodellens kontrollflöde beskriver ett visst beteende. En verksamhets eller ett systems dynamiska beteende beskrivs genom användningen av processmodeller (Johannesson & Perjons, 2000).

### **Dynamisk processmodell med arv**

Den dynamiska processmodellen med arv utgör en av de två vyerna som finns i den arvshierarkiska modellen.

En dynamisk processmodell kan kompletteras med arvsinformation, det vill säga att nedärvda processobjekt gråmarkerats.

### **Huvudaktör**

Varje dynamisk processmodell i EBML modelleras ur en specifik aktörs perspektiv. Denne aktör benämns huvudaktör och kan vara en människa eller en applikation. Vem eller vad som utgör huvudaktör ska återges vid processens namn.

## **Nyckelprocess**

Begreppet nyckelprocess används i samband med diskussioner om verksamhetsmodellering och verksamhetsanalys för att klargöra olika processers funktion i helheten. En nyckelprocess representerar en komplett affärsprocess. En nyckelprocess i en EBML-modell innefattar därför alla de delprocesser som tillsammans resulterar i ett värde för kund. Ett exempel på en nyckelprocess är en kundkontakt som initierar en rad organisationsaktiviteter som producerar någon form av lösning till det problem som kunden önskade få hjälp med.

## **Primära och sekundära processer**

En nyckelprocess utgörs av ofta av flera processer och kan därför delas in ett antal distinkta delprocesser. Dessa delprocesser kan i sin tur delas in i två typer av delprocesser; primära och sekundära processer.

Primära processer kännetecknas av att de nödvändigtvis måste ingå i en nyckelprocess för att den ska fullföljas. I orderhantering måste exempelvis primärprocessen betalning av faktura ingå för att organisationen ska veta om de lyckats leverera det kundvärde som eftersträvas i en processororienterad organisation.

En sekundär process kännetecknas av att de inte nödvändigtvis behöver ingå i en nyckelprocess för att den ska kunna fullföljas, utan kan i viss mån ses som en typ av stödprocess. Det är viktigt att betona att sekundära processer är underordnade primärprocesser. En sekundär process är med andra ord beroende av en eller flera primärprocesser, vilket innebär att en sekundär process endast kan initieras av den primärprocess som den relateras med. Ett kostnadsförslag exempelvis, görs kanske inte rutinmässigt varje gång en order hanteras utan initieras endast under vissa förhållanden som när kunden explicit begärt ett kostnadsförslag. Orderhanteringen, som i det här fallet är en nyckelprocess, kan fullföljas även om inget kostnadsförslag läggs.

## **Process**

En affärsprocess är en samling av organisationsaktiviteter som tillsammans skapar ett värde för kund (se även Affärsprocess). (Davenport, 2000)

I de avseenden som diskussionen inte direkt sammanfaller med processororientering eller processmodellering tillämpas begreppet process i sin generella betydelse, det vill säga ett förlopp bestående av en följd moment som tillsammans bildar ett visst skeende. (Svenska akademiens ordbok, 2001)

## **Processhierarki**

Den processhierarkiska modellen utgör en av de två vyerna som finns i den arvshierarkiska modellen.

En processhierarki visar en grupp av relaterade processer i verksamheten och deras likhet med varandra. För att illustrera detta används arvs-konceptet, lånat från objektorientering. Syftet med att skapa en processhierarki av processer är att strukturera verksamhetens processer i hierarkiska modeller för att underlätta arbetet i förändringsanalysen. Skapandet av processhierarkin gör att de modellerade processernas skillnader och likheter explicit görs.

## **Processmodell**

Processmodell är ett samlingsnamn för de dynamiska och statiska modellerna i EBML.

## **Processmodelleringspråk**

Ett processmodelleringspråk är den grafiska notationen som används för att på ett enhetligt och exakt sätt beskriva alla fenomen som generellt kan kopplas till informationsöverföring och informationstransformation, vilket lämpar sig för specifikation och design i olika modelleringsmetoder (Reisig, 1985). Modelleringspråket är den viktigaste delen i en metod (Fowler, 1997). Ett processmodelleringspråk med en väl definierad semantik tillåter formell analys av det som specifikationer och design representerar (Jensen, 1992).

## **Processobjekt**

Ett processobjekt är ett samlingsnamn för de entiteter/komponenter som en dynamisk processmodell består av och som representerar syntaxen i EBML, det vill säga start-, slut-, och vänttillstånd, meddelanden, APM, val, manuellt val, manuell aktivitet samt timer.

## **Processorientering**

För att lösa problemen som orsakats av det traditionella funktionsorienterade organisations-tänkandet, har det kommit att ersättas av ett processorienterat organisationstänkande där processerna lyfts fram och ges en central roll i företagets verksamhet och administration. Processorientering handlar om att byta perspektiv, innebärande en omkastning av det som tidigare ansågs primärt, det vill säga uppgifter och funktioner och sekundärt, det vill säga processer byter plats. Processerna i en viss verksamhet har med andra ord alltid existerat, men inte givits den uppmärksamhet som processorientering innebär. Främst innebär fokus på en organisations processer att kundvärdet ges nytt utrymme, eftersom processerna korsar avdelningsgränser och knyts samman i en naturlig kedja av aktiviteter som resulterar i ett värde för kund. (Hammer, 1996)

## **Roll**

En roll är detsamma som en mänsklig huvudaktör i en dynamisk processmodell och en gemensam benämning för de personer som kan utföra flera processer i verksamheten.

## **Rutin**

En rutin är en handling eller flera på varandra följande handlingar, som sker periodiskt och som blivit praxis på en arbetsplats eftersom det är ett ofta återkommande göromål. (Svenska akademiens ordbok, 2001)

## **Sekundära processer**

Se under Primära och sekundära processer.

## **Specifikation**

En specifikation uttrycker kraven på en process genom att den explicit beskriver möjliga tillstånd och aktiviteter. (Jensen, 1992)

En systemspecifikation beskriver ett systems beteende med fokus på interaktionen mellan processer och funktionaliteten inuti processerna. (Svantesson, 2000)

## **Statisk processmodell**

Den statiska processmodellen är en av de tre typer av grafiska modeller som finns i EBML. Det motsvarar den statiska processmodellen som finns i moderspråket BML. Modellen beskriver en verksamhets eller ett systems struktur genom att grafiskt visualisera processerna i

ett statiskt tillstånd (Johannesson & Perjons, 2000). Den statiska processmodellen beskriver meddelanden som sänds mellan processerna och meddelanden som sänds mellan processerna och deras miljö. Miljön kan bestå av externa applikationer eller människor.

### **Verksamhetsmodell**

Verksamhetsmodell är ett samlingsnamn för alla modeller som ingår vid verksamhetsmodellering. För EBML inkluderar detta dynamiska, statiska och arvshierarkiska EBML-modeller samt konceptuell modell, rollmatris och målmodell.

## **Appendix 2. Upplägg för utvärderingsfrågor**

### **1. Utvärdering av processmodelleringspråket EBML**

#### **Struktur**

Iterativ genomgång och presentation av EBML, det vill säga språk, notation och typer av modeller enligt:

- Dynamisk processmodell
- Statiskt processmodell
- Arvshierarkisk modell

#### **Intervjufrågor för processmodelleringspråket EBML**

##### **A. Begriplighet**

1. Går det att förstå symbolerna i notationen? (sänd och mottag meddelanden, APM, start-, vänt- och sluttillstånd, val, manuella val och aktiviteter, slumpmässiga starttillstånd, anteckning, versionshantering etc)
2. Är symbolerna intuitiva?
3. Presenterar modellen informationen på ett sätt som enkelt går att förstå?

##### **B. Användbarhet**

1. Är det något som saknas vid modelleringen, det vill säga är det något som inte kan modelleras? Om ja, vad och varför?
2. Är språket för modellen onödigt komplext för syftet att enbart kartlägga och dokumentera organisationen?

### **2. Utvärdering av arbetssätt EBML**

#### **Struktur**

Inledande presentation av arbetssättet som används vid tillämpning av EBML. Arbetssättet består av två faser som går igenom iterativt i intervjun:

- nulägesbeskrivning (konceptuell modell, rollmatris, EBML)
- förändringsanalys (målmodell, analys, uppdatering av EBML, uppdatering av övriga modeller)

#### **Intervjufrågor för nulägesbeskrivning**

- konceptuell modell
- EBML
- rollmatris

## **A. Begriplighet**

1. Går det att förstå symbolerna i notationen?
2. Är symbolerna intuitiva?
3. Presenterar modellen informationen på ett sätt som enkelt går att förstå?

## **B. Användbarhet**

1. Är det något som saknas vid modelleringen, det vill säga är det något som inte kan modelleras? Om ja, vad och varför?
2. Är modellen onödigt komplex för syftet att enbart kartlägga och dokumentera organisationen?
3. Är modellen onödigt komplex för syftet att kartlägga organisationen för att utröna och analysera vilka förändringsbehov som existerar?

## **Intervjufrågor för förändringsanalys**

- målmodell
- analys
- uppdatering av EBML-modeller
- uppdatering av övriga modeller

## **Målmodell**

### **A. Begriplighet**

1. Går det att förstå symbolerna i notationen?
2. Är symbolerna intuitiva?
3. Visar målmodellen organisationens målsättning på ett sätt som är lätt att förstå?
4. Visar målmodellen existerande problem på ett strukturerat och lättförståeligt sätt?

### **B. Användbarhet**

1. Beskriver målmodellen organisationens framtida och önskade tillstånd på ett bra och användbart sätt?
2. Är det lämpligt att använda målmodellen som en utgångspunkt vid diskussion och analys av hur existerande problem kan lösas?
3. Är det lämpligt att inkludera målmodellen i arbetsmetoden?

## **Analys**

### **A. Begriplighet**

1. Är det lätt eller svårt att förstå förhållandet mellan de olika modellerna, exempelvis förhållandet mellan målmodellen och processmodellerna när jämförelsen ska göra i analysfasen?

## **B. Användbarhet**

1. Är förslaget att jämföra modellerna och sätta dem i förhållande till varandra en användbar utgångspunkt vid identifiering och definition av problem och förändringsbehov?

## **Generella intervjufrågor om arbetssättet**

### **A. Begriplighet**

1. Representerar modellerna informationen på ett sätt som generellt sett är lätt att förstå?
2. Var någon modell svårare att förstå än någon annan? I så fall, varför?
3. Är det klart vad de olika modellerna har för funktion i metoden?

### **B. Användbarhet**

1. Är urvalet av modeller användbart? Saknas någon modell eller ingår modeller som inte är nödvändiga? Om ja, varför?
2. Vilka modeller är viktigast? Varför?
3. Existerar onödig redundans i informationen som de olika modellerna representerar? Om ja, varför?
4. Vad är ditt generella intryck av föreslaget arbetssätt?
5. Uppfyller arbetssättet sitt syfte, det vill säga beskriva verksamhetens nuläge respektive möjligheten att utröna förändringsbehov?