

| |
|-------------------------------------|
| Krav på IA |
| ■ Nästa Generation Modellering |
| Avancerad utbildning för handledare |
| Katalogprinciper |
| Uttagssystem |
| Informationspridning |

Rapport N nr 1: Modelleringsansatser för begrepps- och datamodellering: – Beskrivning och försök till jämförelse

Rapport N nr 2: Generering av konceptuella modeller från policydokument

■ Rapport N nr 3.2 : Modellering enligt Tempora

Modellering enligt Tempora

Ulf Persson
Benkt Wangler
Rolf Wohed
SISU

Spridningsförbehåll:

Denna rapport får endast spridas och användas inom de organisationer som deltar som parter i TRIAD-projektet.

© TRIAD-parterna feb 1992

Rapporten är skriven i och för TRIAD delprojekt Nästa Generation Modellering.

Innehåll

Modellering enligt Tempora.

| | |
|---|----|
| 1. Inledning | 2 |
| 2. Modelleringsspråk | 6 |
| 2.1. Datamodell | 6 |
| 2.1.1. Grundläggande begrepp | 6 |
| 2.1.2. ISA-relationer | 7 |
| 2.1.3. Komplexa objekt | 9 |
| 2.1.4. Tidsdimensionen | 10 |
| 2.2. Funktionsmodell | 12 |
| 2.3. Regelmodell | 13 |
| 2.3.1. Regelspråket CRL | 15 |
| 3. Metodik | 19 |
| 3.1. Målanalys | 20 |
| 3.2. Analysens syfte | 20 |
| 3.3. Verksamhetsregler | 21 |
| 3.3.1. Samspel mellan begrepp och regler | 22 |
| 3.3.2. Formalisering av regler i formellt språk | 23 |
| 3.4. Avgränsning och detaljering | 24 |
| 4. Datorstöd | 25 |
| 5. Tempora | 27 |
| 6. Referenser | 28 |

1. Inledning

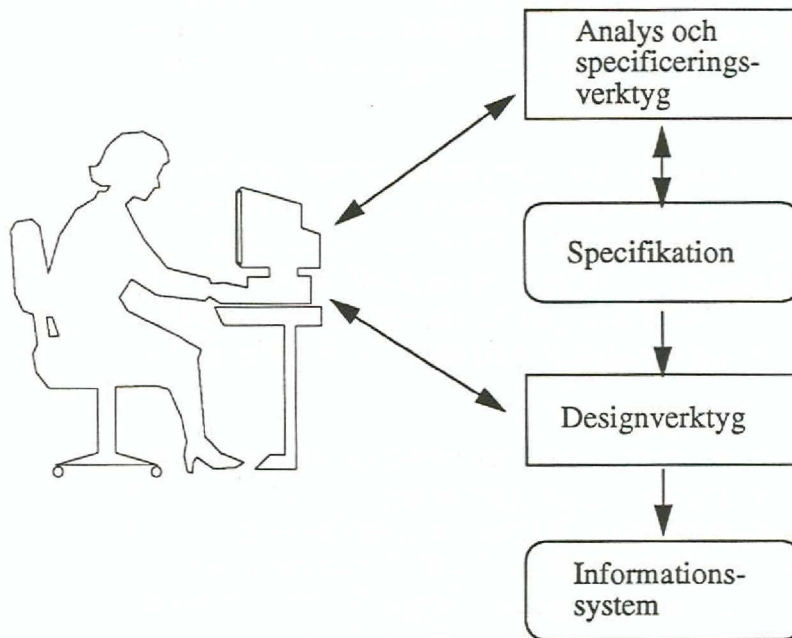
Under senare år har det märkts en ökad efterfrågan på funktionalitet hos informationssystem som går utöver traditionell datahantering. Det har också argumenterats för att utveckling av nästa generations informationssystem kommer att kräva en ansats som bättre knyter ihop verksamhetsregler och systemfunktionalitet. Ett antal frågeställningar sprungna ur dessa krav behandlas i Esprit-projektet Tempora [3], som utgör grunden för det som rapporteras i denna rapport. Behovet av utökad funktionalitet behandlas i Tempora genom användning av en formalism för konceptuell modellering som speciellt hanterar verksamhetsregler, tid och komplexa objekt. Detta språk stöds på databasnivå av en utökad version av relationsmodellen med tidssemantik och en exekveringsmekanism som tillhandahåller aktiv databasfunktionalitet.

Temporas utgångspunkt är att specificering av ett informationssystem ska ses som processen att utveckla eller förändra en befintlig kunskapsbas, som beskriver en verksamhets policy, och på vilken det vidare utvecklingsarbetet sedan baseras. I Tempora används denna kunskapsbas för att lagra definitioner av grundläggande fakta och operationer på dessa tillsammans med de regler som styr operationerna och upprätthåller konsistensen hos fakta. Figur 1 visar huvudkomponenterna i den tänkta systemutvecklingsmiljön och hur dessa avses samverka.

De strukturella egenskaperna hos en applikation uttrycks i termer av en semantisk datamodell. Det modelleringsspråk som utvecklats för detta benämnes *ERT*, vilket står för Entity-Relationship-Time. Funktionskomponenten har att göra med definitionen av operationer och utgöres av dataflödesdiagram av tämligen konventionellt snitt. Denna modelltyp benämnes i Tempora *PID* (Process Interaction Diagram). Beteendet hos systemet beskrivs i termer av regler. I princip kan två generella typer av regler urskiljas: konsistensregler som har att göra med samstämmighet hos innehållet i databasen och aktivitetsregler som beskriver hur operationer mot systemets databas utlöses och kontrolleras.

Denna artikel innehåller en tämligen detaljerad beskrivning av *ERT*-modellen och regelspråket. Funktionsmodellen beskrivs endast översiktligt, eftersom denna inte i sig inte innehåller några nyheter utan endast har intresse som underlag för dekomponering av verksamheten och för utformning av aktivitetsregler.

ERT-modellen utgår från Entity-Relationship-ansatsen [1], som utvidgats med utökad semantik och grafik i, väsentligen, två riktningar: modellering av tid och modellering av komplexa objekt.



Figur 1. Systemutveckling enligt Tempora.

1

Införandet av tidsdimensionen motiveras av att man i många applikationer inte bara är intresserad av sakernas nuvarande tillstånd utan också har behov av att bevara information om historiska händelser. Behovet av att hantera tid mer handfast dök upp i början av 70-talet inom medicinska informationssystem, där en patients historia är av särskilt intresse. Sedan dess har mycket forskning utförts om egenskaper hos tid i datorbaserade informationssystem och om hanteringen av tidsaspekten hos data.

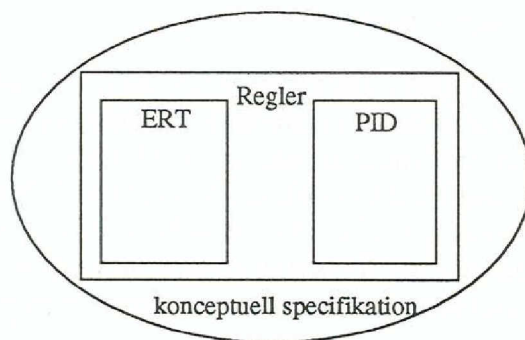
Komplexa objekt har införts för att ge möjlighet att abstrahera bort onödigt komplexitet i en modell. Detta uppnås genom att vissa objekt betraktas som komponenter av andra och därigenom kan döljas i dessa.

Ett problem med dagens specifikationspråk är att verksamhetsregler inte kan representeras explicit. Istället bakas dessa in i programkod och sprids kanske ut över ett antal applikationer. Det betyder att det blir svårt att underhålla systemet eftersom verksamhetsreglerna inte är dokumenterade. Man kan t ex inte lägga till en ny applikation och veta säkert att den kommer att fungera tillsammans med de befintliga.

I Tempora finns ett regelspråk i vilket man kan representera verksamhetsregler på ett deklarativt sätt. Därigenom uppnås att man kan koncentrera sig på vad som måste gälla i tillämpningen snarare än på att definiera hur systemet ska gå till väga för att tillgodose dessa verksamhetsregler.

Regelspråket, CRL (Conceptual Rule Language), har en tvådelad roll. För det första kan det beskriva villkor på element i ERT och härledning av ny information från existerande dito. För det andra åskådliggör det sekvensiering av operationer och villkor. Språket har utformats för att vara så generellt att alla typer av regler kan beskrivas. Begränsad redundans tillåts om det ökar läsbarheten men inte så att det leder till tvetydighet.

En målsättning för utvecklingen av CRL har också varit att språket ska kunna förstås av alla i kravspecificeringen inblandade parter med ett minimum av träning och expertis. CRL har ett väldefinierat gränssnitt till de andra modelleringspråken, ERT och PID, för att möjliggöra referenser och för att vissa metodologiska synergieffekter ska uppnås. Resultatet är en effektiv, integrerad modelleringsformalism, med hög uttrycksstyrka och alla nödvändiga struktureringsmekanismer.



Figur 2. Konceptuell modell enligt Tempora.

2

Den grundläggande representationsformen för en specifikation i Tempora är regler. Alla utsagor om verksamheten kan representeras i regelmodellen. Begreppsmodellen och funktionsmodellen är projektioner av delar av regelmodellen som gör det lättare att överblicka och förstå en specifikation (fig. 2). Anledningen till att inte alla utsagor kan uttryckas i de grafiska språken ERT och PID, är att de inte har tillräcklig uttrycksstyrka för att alla krav skall kunna uttryckas. Detta trots att Temporas modelleringspråk är mer generella än traditionella grafiska

modelleringsspråk. Det finns också en stor risk i att försöka uttrycka allt i grafiska formalismer genom att man överbelastar den grafiska representationen så att den inte längre bidrar till förståelsen utan tvärtom försvårar tolkningen av modellen.

2. Modelleringsspråk

Ett av de viktiga inslagen i Tempora är konceptuell modellering, d v s en aktivitet där systemanalytikern tillsammans med verksamhetsexperten skapar en konceptuell specifikation som är oberoende av realiseringsmiljö, såväl hårdvara som typ av databashanterare och programmeringsspråk. Den konceptuella modellen skall beskrivas i de termer som är viktiga i verksamheten. Utmärkande för Tempora är också att den denna modell innehåller de regler som styr verksamheten på ett motsvarande implementationsoberoende sätt.

I Tempora skiljer man på tre huvudfaser i utvecklingsarbetet:

- *Verksamhetsinriktad modellering*
- *Informationssysteminriktad modellering*
- *Informationssystemutformning*

I de båda första av dessa aktiviteter beskrivs verksamhetens respektive det önskade informationsstödet begrepp, funktioner och regler med hjälp av de tre språken ERT, PID och CRL. I det tredje steget utformas det önskade systemet utgående från informationssystemmodellen. En närmare beskrivning följer i kapitel 3.

2.1 Datamodell

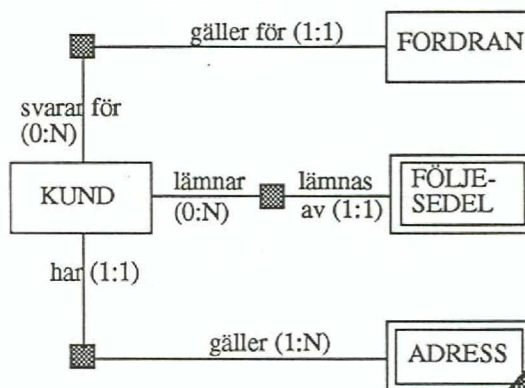
2.1.1 Grundläggande begrepp

ERT skiljer, i likhet med ER-ansatsen, klart på objekt och samband mellan dessa och ger dessutom möjlighet till uttrycklig modellering av tid, taxonomiska hierarkier och komplexa objekt.

Det mest grundläggande begreppet i ERT är *klass*, vilket definieras som en mängd av individuella objekt med gemensamma egenskaper, d.v.s. objekt som är av samma typ. I ett ERT-schema specificeras endast klasser av objekt. Dessa är av två slag, nämligen entitetsklasser och värdeklasser. En *entitet* är en unikt urskiljbar företeelse i verkligheten, som inte enbart utgöres av en teckenföljd. Det sistnämnda objektslaget benämnes *värde*.

Förutom objektklasserna finns klasser av objektsamband. Varje sambandstyp ses som ett rollpar där varje roll beskriver hur en enskild entitet eller ett enskilt värde ingår i ett samband. På detta sätt uttrycks var-

je samband med två meningar, en i varje riktning. Meningarna är syntaktiskt olika men semantiskt ekvivalenta. För varje roll kan man också ange min- och max-värden för avbildningens kardinalitet, d v s för hur många objekt som via denna sambandsroll är associerade med ett objekt i den klass som är subjekt i rollen.



Figur 3. ERT-modell.

3

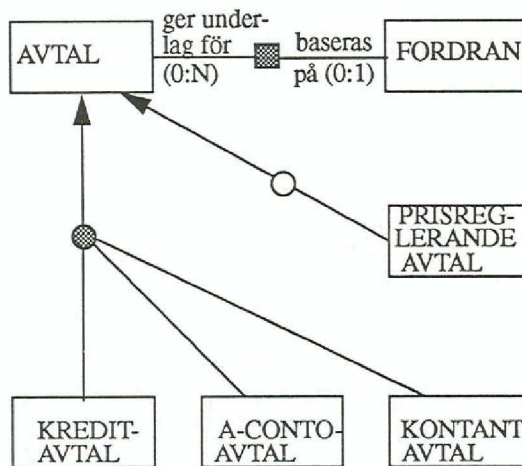
Figur 3 visar ett mycket enkelt exempel på en ERT-modell. I denna modell förekommer bl.a. entitetsklasserna "kund" och "fordran" och värdeklassen "adress". De mellanliggande objektsambandens avbildningskardinaliteter anges av de med parenteser omgivna uttrycken. Sålunda gäller att en kund svarar för minst ingen (0) och högst godtyckligt (N) många fordringar, att en kund har exakt en adress (minst 1 och högst 1), att en fordran gäller för exakt en kund och att en adress gäller för minst en och högst godtyckligt många kunder.

Som framgått av inledningen konstrueras också i Tempora ett verktyg med vars hjälp man kan rita ERT-modeller. Till denna ERT-editor hör också en uppsättning formulär där ytterligare detaljer om klasserna kan specificeras. Bland annat gäller att värdeklasser och samband mellan en entitetsklass och en värdeklass också kan beskrivas i sådana formulär, d v s förutom den möjlighet man har att definiera dessa grafiskt.

2.1.2 ISA-relationer

ERT ger också möjlighet att definiera generaliseringssamband (ISA-relationer) mellan entitetsklasser. Ett sådant samband anger enkelt uttryckt att en entitetsklass står i ett delmängdsförhållande till en annan. I

figur 4 är sålunda entitetsklassen "avtal" specialiserad i dels "prisreglerande avtal" och dels utgående från betalningsform.



Figur 4. Generalisering/specialisering.

4

Varje cirkel i grafen motsvarar ett sätt att specialisera. Klasser som är subklasser till samma superklass, men enligt olika specialiseringskriterier, som t ex "prisreglerande avtal" och "kreditavtal", kan överlappa varandra d v s ha gemensamma element.

Om ett antal subklasser är förbundna med samma cirkel innebär detta att de inte kan överlappa varandra utan att de är disjunkta. I Figur 4 är detta fallet med "kreditavtal", "a-conto-avtal" och "kontantavtal", d v s ett avtal kan bara tillhöra *någon* av subklasserna. Om cirkeln är fylld gäller dessutom att specialiseringen är uttömmande, d v s subklasserna utgör tillsammans hela superklassen. Med andra ord gäller för figur 4 att det inte finns några avtal som inte tillhör någon av de ovan uppräknade tre subklasserna.

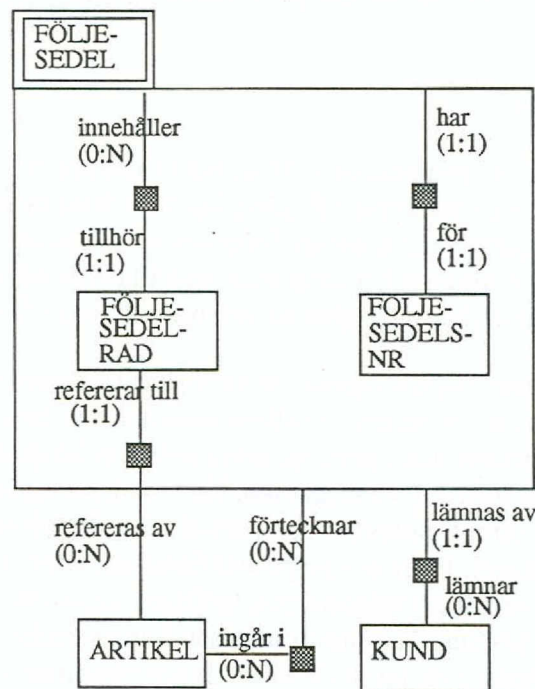
Innebörden i en ISA-relation är som tidigare påpekats att mängden av entiteter som hör till subklassen är en delmängd av mängden entiteter som hör superklassen. Med andra ord är mängden av "kreditavtal" en delmängd av mängden av "avtal".

Det är viktigt att förstå att en ISA-relation inte är ett samband mellan entiteter utan mellan entitetsklasser. Det är alltså inte så att ett kreditavtal är relaterat till ett avtal. Innebörden i ISA-relationen är i stället att om en entitet är ett kreditavtal så är den på en och samma gång också ett avtal. Den är ett element både i mängden av kreditavtal och i mängden av avtal.

Alla de samband som superklassen har med andra objektclasser gäller också för subklasserna. Man brukar tala om att superklassens egenskaper *ärvs* av subklasserna. Så gäller t ex att "avtal ger underlag för fordran" oavsett till vilken av subklasserna avtalet hör. I databasen kommer prisreglerande avtal likaväl som andra slag av avtal att vara relaterade till en "fordran" i den ovannämnda sambandstypen.

2.1.3 Komplexa objekt

För att ge möjlighet att abstrahera bort oönskad komplexitet i en bild har abstraktionsmekanismer benämnda (något oegentligt) "komplex entitetsklass" resp "komplex värdeclass" införts. Entitetsklassen "följesedel" i figur 3 är exempel på en komplex entitetsklass, medan "adress" är en komplex värdeclass. Om "följesedel" öppnas erhålles bilden i figur 5 som visar de komponenter som en följesedel består av.



Figur 5. Öppet komplext objekt.

Komponentklasserna representeras av de inom den stora fyrkanten befintliga symbolerna. Mellan sambanden "följesedel innehåller följesedelrad refererar till artikel" och "följesedel förtecknar artikel" råder ett samband i så motto att den mängd av artiklar som förtecknas i en följesedel är den mängd artiklar som refereras av följesedelrader som hör till denna följesedel. Sambandet "följesedel förtecknar artikel" kan således

i princip härledas ur "följesedel innehåller följesedelrad refererar till artikel". Beroenden av denna typ representeras i specifikationen av CRL-regler och realiseras i informationssystemet antingen som en restriktion (det "härledbara" sambandet lagras i databasen) eller som en härledningsregel. I ERT finns möjligheten att i informationssystemmodellen visa att ett samband är härlett genom att representera det med en streckad linje. Även entitetsklasser kan vara härledda och representeras i så fall också genom att symbolen är streckad.

Ur databassynpunkt gäller att en komplex entitetsklass är ett objekt i sig och således inte utgörs av ett aggregat av komponenter utan endast är relaterat till dessa med normala objektsamband. Dessa samband benämnes ofta "består av/ingår i" eller "har del/är del av".

Komplexa värdeklasser har motsvarande uppbyggnad. Så består t ex den komplexa värdeklassen "adress" av komponenterna *gatuadress* och *ort* (redovisas ej i bild).

2.1.4 Tidsdimensionen

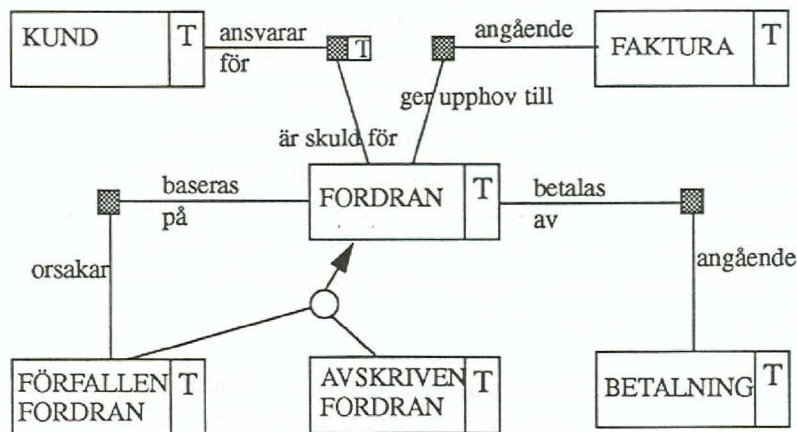
För att göra det möjligt att se information i ett utsträckt tidsperspektiv och att representera regler som refererar till tiden, har den databas som hör till ett informationssystem skapat enligt Tempora en temporal dimension. Detta innebär både att man har möjlighet att hålla reda på historisk information och att man har möjlighet att referera till framtiden.

Närmare bestämt så har varje enskilt faktum (entitet eller samband) en viss validitetsperiod, dvs en period under vilken detta faktum betraktas som sant. Validitetsperioden ska, i princip, svara mot den tidsperiod då motsvarande faktum är giltigt i verkligheten. En viss kundentitets validitetsperiod ska alltså direkt motsvara den period då vederbörande betraktas som kund i företaget. Efter denna period finns entiteten kvar som en historisk entitet som går att referera och möjligen att återuppliva, men som inte är giltig just nu. På motsvarande sätt kan en entitet eller ett samband ha en giltighet som helt och hållet ligger i framtiden.

I databasen representeras validitetsperioden av en *tidsstämpel* som anger dess början och slut i termer av de *verkliga* tidpunkter då vederbörande faktum blev sant respektive falskt. Det som representeras i databasen är med andra ord de tidpunkter då något inträffat i verkligheten (händelsetid) inte då det blivit känt i databasen (transaktionstid).

Eftersom man inte är intresserad av att hålla reda på historien för alla klasser av entiteter och samband, så har man möjlighet att i modellen markera de klasser för vilka man vill detta, med ett T-märke. Figur 6 visar ett enkelt konceptuellt schema, där vissa klasser markerats med ett

T. Innebörden i detta är att de entiteter och samband som svarat mot dessa klasser kommer att tidsstämplas medan alla övriga anses vara, implicit, giltiga under hela databasens livstid. Då ett icke tidsstäplat faktum görs falskt kommer det att avlägsnas ur databasen, d v s man förlorar kunskapen om att det en gång har varit giltigt, medan motsvarande operation för ett tidsstäplat faktum endast innebär att dess giltighetstid upphör.



Figur 6. ERT-modell med T-markeringar.

6

T-markeringar kan införas i en modell av åtminstone två skäl:

- Entiteten eller sambandet är "tidsvarierande" och vi vill hålla reda på dess historia.
- Entiteten eller sambandet är i och för sig inte tidsvarierande, men vi använder T-markeringen som ett sätt att "modellera" start- och slutpunkt i stället för att införa särskilda datumattribut för detta.

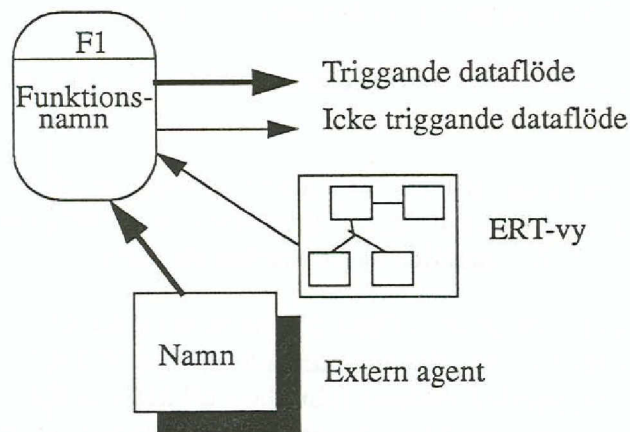
Syftet med de T-märken som införts i Figur 6 är att man i detta fall vill hålla reda på under vilken tid de olika entiteterna är giltiga. Vad som är att betrakta som validitetsperiod varierar från klass till klass och definieras av de regler som, för varje klass, beskriver hur entiteter uppstår och görs ogiltiga. För fordran gäller t ex att den är giltig från det att den uppstår till dess att den är betald. Sambandet "fordran är skuld för kund" har en T-markeringen på grund av att en kund kan överta fordran från en annan och att vi vill hålla reda på vilken kund som varit betalningsansvarig under olika tidsperioder. T-markeringarna på "förfallen fordran" och "avskriven fordran" innebär att vi har information om under vilka tidsperiod en fordran ev intar något av dessa båda tillstånd.

Validitetsperioderna för en fordrans tillhörighet till dessa klasser kan dock inte överlappa varandra eftersom de mot klasserna svarande mängderna i varje ögonblick ska vara disjunkta.

2.2 Funktionsmodell

Som tidigare nämnts använder Tempora funktionsmodeller av traditionellt snitt för att beskriva en del av den dynamiska aspekten. Denna formalism benämnes PID (Process Interaction Diagram) och beskriver beteendet hos en verksamhet eller ett informationssystem som ett antal samarbetande funktioner. Samverkan beskrivs i termer av dataflöden. Varje funktion kan i sin tur beskrivas som ett antal samverkande funktioner. PID-delen av den konceptuella specifikationen består alltså av hierarkiskt nedbrutna verksamhetsfunktioner i ett antal diagram där varje diagram beskriver en viss del av verksamheten. Funktionsmodellen är en grafisk beskrivning men på lägsta nedbrytningsnivån beskrivs varje process i termer av ett antal regler.

PID i sig själv är en traditionell funktionsmodell men med den skillnaden att man skiljer på triggande och icke triggande dataflöden. Ett triggande dataflöde startar en bearbetning i en funktion. Således behövs minst ett triggande dataflöde till varje process. Icke triggande dataflöden representerar en funktions informationsbehov och visar därmed vilka data som en funktion bearbetar.



Figur 7. Funktionsmodell enligt PID.

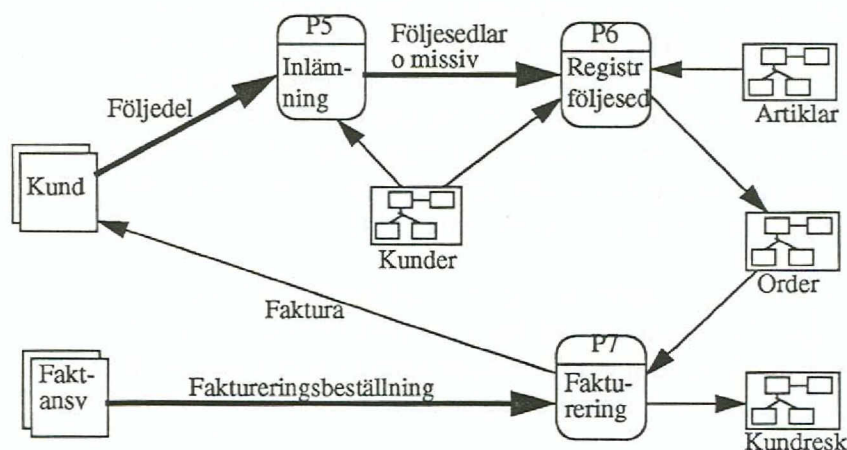
7

ERT-vyer motsvarar det som i traditionella dataflödesdiagram kallas datalager. Benämningen ERT-vy har införts för att betona det faktum att vi har en enda 'konceptuell' informationsbas i vilken funktioner kan ha behov av att läsa eller skriva viss information. Med andra ord är de

aldrig intresserade av hela det konceptuella schemat utan bara av vissa delar av detta (extern vy i tre-schema-terminologi).

För att en verksamhet eller ett informationssystem ska kunna fungera tillsammans med sin omgivning måste den kunna kommunicera med den. I PID sker kommunikationen med dataflöden till eller ifrån externa agenter. Externa agenter ligger alltså utanför systemgränsen.

Figur 8 visar ett enkelt exempel på en PID innehållande de externa agenterna "Kund" och "Faktureringsansvarig" och funktionerna "Inlämning", "Registrera följesedel" och "Fakturering" med involverade dataflöden och ERT-vyer. De pilar som är bredare anger triggande flöden. .



Figur 8. Process Interaction Diagram

8

Diagrammet i Figur 8 ger en starkt förenklad bild av Postens system för fakturering av kreditkunder.

2.3 Regelmodell

Regelmodellen i Tempora är till för att man ska kunna fånga verksamhetsregler som inte kan uttryckas vare sig i begreppsmodellen eller funktionsmodellen. Ofta finns det krav på verksamheten som pga brister i modelleringsspråkens uttryckskraft inte kan fångas med traditionella modelleringsspråk. Det kan t ex vara fråga om policyregler och regler för hur verksamheten bedrivs.

Man skiljer på abstrakta och formaliserbara regler i Tempora. Abstrakta regler är högnivåregler som snarare uttrycker specifika målformule-

ringar än regler, t ex "företaget ska gå med vinst". Dessa kan inte i sig själva formuleras i Temporas formella regelspråk utan beskrivs i naturligt språk. Däremot kan man utifrån sådana abstrakta regler bestämma mer konkreta regler som 'implementerar' dessa. Efter ett antal sådana steg når man så småningom en konkretiseringsnivå där regler går att beskriva formellt.

Att en regel är formaliserbar betyder inte att den måste uttryckas formellt. Tvärt om, alla regler som beskriver verksamheten bör först uttryckas i naturligt språk. Även om regelformalismen är förhållandevis abstrakt så är den inte direkt konstruerad för att förstås av användare. Den vänder sig i första hand till systemanalytikern, som ett kommunikationsmedel till dem som utvecklar systemet. Dvs den uttrycker så precist som möjligt de krav som verksamheten har på systemet, utan att uttala sig om hur de ska lösas. Även om man tror sig ha formulerat en regel precist i naturligt språk så är det först när man har formaliserat den som man vet om den verkligen är precis. Det kan dölja sig många oklarheter under vackra formuleringar i naturligt språk. Detta är något som framkommit mycket tydligt under den fallstudie som utförts inom ramen för Temporaprojektet och som redovisas i [4].

Det visar sig också att det under arbetet med att formulera regler även framkommer nya eller dolda krav på begreppsmodellen och funktionsmodellen. Det gäller inte bara vid formaliseringen av regler utan även i viss mån när man preciserar abstrakta regler i naturligt språk.

Som tidigare antytts kan det också finnas relationer mellan regler. Framför allt gäller att många regler står i ett motivförhållande till andra regler. Som framgår av beskrivningen av begreppsmodellen ovan har varje förhållande i Tempora två roller. Detta gäller även för förhållandet i regelmodellen. I ena riktningen heter förhållandet "motiveras av" och i andra riktningen "motiverar". Det är alltså frågan om en sorts komponentförhållande som beskriver vilka regler som preciserar en annan mer abstrakt regel. Resultatet blir en hierarkisk struktur där varje nod kan ha gemensamma subträd med andra noder (regler).

Varje regel kan också associeras med objekt i ERT. Det är i första hand fråga om objekt som regeln uttalar sig om, men även förhållanden som uttrycker annan semantik kan förekomma. Ett typiskt exempel på associationer till ERT utgöres av härledda komponenter i ERT, eftersom varje härledd komponent behöver en härledningsregel som talar om från vilka objekt informationen härleds och hur den beräknas.

Regler används också för att uttrycka hur processer i funktionsmodellen utlöses och vad som sker i processen. De villkor som utlöser en process benämnes triggers.

Som exempel på motiveringssamband kan den tidigare nämnda regeln "företaget ska gå med vinst" användas. Denna regel kan bli motiverad av preciseringen "vi ska ha betalt för våra kundfordringar så snabbt som möjligt", vilket i sin tur kan preciseras som "skicka ut kundfakturer senast en dag efter beställning". Den här regeln är nu tillräckligt precis för att kunna uttryckas formellt. Men innan vi går in på hur det kan se ut, ska vi först beskriva några allmänna egenskaper hos det formella språket.

Den formella notationen i Temporas regelspråk är deklarativ, dvs vi beskriver inte hur saker och ting ska utföras i systemet utan bara vad som ska göras. Språket bygger på formell logik och kan tack vare det kontrolleras i högre grad än vad som annars hade varit möjligt.

För att förstå hur en regel fungerar vill vi påminna om hur tid hanteras i Tempora. Det finns två oberoende tidsaxlar i Tempora. Den ena av dem kallas händelsetid, den andra transaktionstid. När man mäter händelsetiden registrerar man när händelser inträffar i verksamheten. Transaktionstiden har att göra med när information om händelserna registreras i systemet. Händelsetiden är det som talar om när en viss uppgift är sann i verksamheten. Det är också framför allt den vi använder i Tempora.

2.3.1 Regelspråket CRL

En regel i Tempora består av tre delar: *när*, *om* och *så*. Dvs *när* något precis inträffat *om* något villkor gäller *så* gäller något tredje påstående. *När*-delen och *om*-delen är valfria och om båda saknas i en regel, skrivs inte *så* ut explicit. Det ger oss fyra syntaktiska grundtyper:

| | |
|---|---|
| <i>När</i> p1 <i>om</i> p2 <i>så</i> p3 | — <i>När</i> påstående p1 just inträffat <i>om</i> påstående p2 är sant <i>så</i> gäller påstående p3 |
| <i>När</i> p1 <i>så</i> p2 | — <i>När</i> påstående p1 just inträffat <i>så</i> gäller påstående p2 |
| <i>Om</i> p1 <i>så</i> p2 | — <i>Om</i> påstående p1 är sant <i>så</i> gäller påstående p2 |
| p | — Påståendet p är sant |

En regel som är definierad på fig. 3 skulle t ex kunna se ut så här:

Om KUND lämnar FÖLJESEDEL *så* KUND svarar_för FORDRAN

R 1

Den betyder att om en kund lämnar en följesedel så svarar kunden också för en fordran.

Varje regel kan uttryckas i vilken av dessa typer som helst och ändå betyda samma sak. Det är viktigt att veta eftersom varje typ leder tankarna till ett visst sätt att implementera regeln. Utan träning är det ofta enklare att uttrycka sig proceduriellt (imperativt, hur) än deklarativt (vad). Det är en av anledningarna till att man i Tempora valt att införa just dessa typer.

För att förebygga missförstånd är det viktigt att alla inblandade parter tolkar regeln på samma sätt. Därför är det bättre att reglerna återspeglar sättet att tänka och därmed leder tankarna till en viss lösning (hur de utförs) än att användarna har olika sätt att förstå reglerna. Det går ändå att utföra dem på andra sätt än de som antytts i reglerna. Vi kommer dock inte här att gå in på hur man översätter en typ till en annan. Det är en fråga om användning av logiska transformationer som skulle föra beskrivningen för långt in på detaljer.

Regeln i föregående exempel kan implementeras på tre olika sätt:

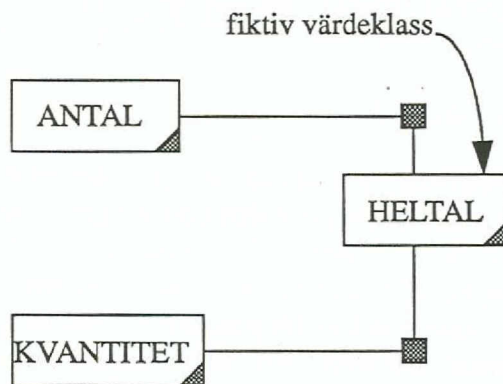
- *Handling*: dvs om en följesedel registreras så skapas automatiskt en fordran för kunden att svara för.
- *Härledning*: fordran som kunden svarar för lagras inte utan härleds vid behov från uppgifter i följesedel.
- *Restriktion*: följesedel och fordran registreras för sig. Regeln ser till så att samma kund avses i 'kund lämnar följesedel' och 'kund svarar för fordran'.

I det första fallet ser vi till att lagra information om fordran för alla kunder med följesedlar. Det är alltså frågan om att utföra en handling, till exempel att begära in information från omgivningen. I det andra fallet beräknar vi fordran för kund varje gång den efterfrågas genom att leta upp den text från ett attribut i följesedeln. I det tredje fallet genereras ett felmeddelande om en kund som har en följesedel inte också har en fordran.

Som vi redan sett refererar regler till objekt i ERT. För att kunna beskriva villkor på och selektioner av ERT-data så behöver vi ett tillräckligt generellt språk som klarar av att uttrycka komponenterna i modellen. Som vanligt kan man använda både typer, variabler och konstanter i regeluttrycken. I föregående exempel använde vi oss bara av typer. Ett regeluttryck kan användas för att göra selektioner av instanser från ERT. Man kan selektera både enskilda instanser och mängder av instanser.

Man kan även namnge tupler av instanser, de kallas då predikat. Det finns också operatörer inbyggda i språket som gör att man kan bygga upp sammansatta uttryck. Det finns förstås vanliga logiska och aritmetiska operatörer och en uppsättning för hantering av tid.

För att värden tillhörande olika värdeklasser skall kunna jämföras med varandra måste de vara av samma grundtyp. Det finns ett antal sådana elementära typer i Tempora. Det kan t ex vara intressant att jämföra kvantitet i en följesedel med ett minsta antal för en rabattsats. Man kan då tänka sig modellen utvidgad med en fiktiva typ som visas i figuren nedan (fig. 9).



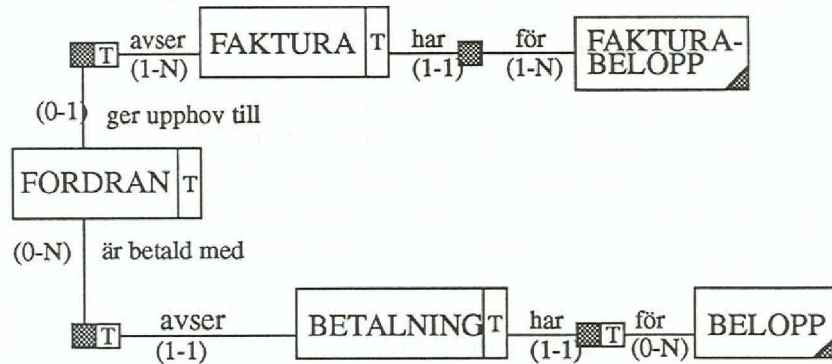
Figur 9. Mental utvidgning av modellen med fiktiva värdeklasser. 9

Det kan nu vara dags att se på en mer komplex regel. Låt oss betrakta en regel som säger att fakturan skall tas bort när den är fullständigt betald. Motsvarande del av ERT finns i fig. 10.

När BETALNING.B

om BETALNING.B avser FORDRAN ger_upphov_till FAKTURA.F och
 SUMMA {X för_villka_någon_gång_i_dåtid
 (FAKTURA.F avser FORDRAN är_betald_med BETALNING har BELOPP.X)} ≥
 FAKTURABELOPP för FAKTURA.F

så alltid_i_framtiden icke(FORDRAN ger_upphov_till FAKTURA.F) R 2



Figur 10. Exempel-ERT för regel R 2.

10

Regeln uttalar sig om vad som ska gälla omedelbart efter det att en betalning (B) sker. Då provar man först om just den betalningen är relaterad till en fordran som i sin tur är relaterad till en viss faktura (F). En punkt följt av stor bokstav, betyder att bokstaven är en variabel som innehåller en viss instans av entiteten framför punkten. I nästa steg kontrolleras om summan av betalningar som avser denna faktura överstiger fakturabeloppet. Om dessa villkor är uppfyllda förpassas fakturan till historien. Operatorerna *någon_gång_i_dåtid* och *alltid_i_framtiden* är exempel på temporala operatorer.

3. Metodik

Ett datoriserat informationssystem utvecklas enligt Tempora av systemutvecklare och domänexperter tillsammans genom en successiv detaljering av en specifikation över systemet. Specifikationen över datorsystemet grundar sig på detaljerade verksamhets- och informationssystemanalyser.

I Tempora delas systemutvecklingsprocessen in i tre huvudfaser:

- *Verksamhetsanalys* där alla begrepp, funktioner och verksamhetsregler av betydelse för verksamheten i stort ska ingå i modellerna. Syftet med verksamhetsmodellen är att beskriva verksamheten ur ett helhetsperspektiv som ska vara relevant i förhållande till syftet med beskrivningen.
- *Informationssystemanalys* där man med verksamhetsmodeller som grund utvecklar mer detaljerade informationssystemmodeller. I denna fas avgränsas informationssystemet från den allmänna verksamheten och endast det som är relevant för informationssystemet behandlas. I informationssystemmodellen försöker man minimera redundans och tvetydighet.
- *Design* där man direkt koncentrerar på den fysiska realiseringen av informationssystemet. Här avgörs till exempel hur olika funktioner som ska realiseras med hjälp av datorer, hur databasschemat ska se ut och hur reglerna ska tolkas.

I verksamhets- och informationssystemanalysen sker utvecklingen av begrepps-, regel- och funktionsmodeller parallellt. Iteration är ett nyckelord i modelleringsarbetet, dvs det är alltid möjligt, och nödvändigt, att gå tillbaka och ändra i redan skapade modeller.

Tempora-metoden har utvecklats inom projektet och baseras på traditionella metoder som utökats och anpassats för att kunna hantera Temporas modelleringsprimitiver och informationsbehov. Till exempel har ABC-metoden [5] använts som grund för det praktiska genomförandet av analysarbetet i de tidiga faserna.

En vidareutveckling av Tempora-metoden pågår som bäst (december 1991) inom Tempora-projektet. Syftet är att göra metoden mera konkret och enklare att tillämpa. Erfarenheter från fallstudier, bl a [4] har gett värdefulla tillskott till metodutvecklingen, speciellt vad gäller specifikation av verksamhetsregler.

3.1 Målanalys

Själva analysarbetet tillsammans med projektdeltagarna inleds med en enkel analys av verksamhetens/projektets mål. Syftet med målanalys är att få deltagarna att själva skapa ramarna för projektet och att komma överens om projektets fokus och innehåll.

Målanalysen resulterar i en målmodell som handlar om verksamhetens mål, ambitioner och medel för att uppnå dem. Målmodellen är en informell grafisk struktur över hur några av verksamhetens viktigaste mål hänger ihop. Sambanden mellan målen kan ha namn som *motiveras av*, *underlättas av*, *är ett medel för*, *uppfyller* etc. Det finns inga höga krav på fullständighet eller korrekthet för målmodellen vilket innebär att endast lite tid ska läggas på denna aktivitet. En typisk målmodellering kan ta 1-2 timmar med normalt aktiva deltagare.

Förutom att ensa deltagarna om ett fokus för det vidare arbetet kan målformuleringarna användas för att hitta de första grundbegreppen för begreppsmodellen. Eftersom målen uttrycker utsagor om verksamheten använder de många av verksamhetens begrepp. En genomgång av målen kan därför ge en flygande start i begreppsanalysen.

3.2 Analysens syfte

Syftet med verksamhetsanalysen är utveckla en modell som ger en helhetsbild av verksamheten. En verksamhetsmodell är ett viktigt verktyg för att alla i projektgruppen ska kunna förstå verksamheten och enas om en allmängiltigt synsätt. Begreppsstrukturen i en verksamhetsmodell är en modell över verksamhetens begrepp och inte över databasobjekt. Det innebär att en verksamhetsmodell kan innehålla begrepp som man klart inser inte kommer att ingå i det framtida informationssystemet men som ändå kan vara nödvändiga för att förklara och motivera relaterade begrepp.

En verksamhetsmodell innehåller ofta redundant information, dvs samma sak finns uttryckt på olika sätt i modellen. Redundans är vanligen något vi vill undvika i ett informationssystem men är inte alls något problem i verksamhetsmodellen. Detta beror på att de begrepp som används i verksamheten ofta är redundanta och att de dessutom kan användas för att ge kompletterande beskrivningar av olika fenomen. I informationssystemanalysen bör man dock försöka reda ut all redundans, dvs försöka avgöra om till exempel två olika samband mellan samma begrepp verkligen uttrycker olika samband eller om de visar två olika representationer av samma samband.

Arbetet med verksamhetsmodellen bör pågå till dess att den är tillräckligt detaljerad och begriplig för att alla deltagare ska vara överens om modellens innehåll. Vi vill poängtera vikten av att inte i ett för tidigt stadium fortsätta från verksamhetsanalys till informationsanalys. Resurser som satsas på att utveckla kompletta och precisa verksamhetsmodeller under de tidiga faserna är väl investerade och kommer att bidra till att arbetet går smidigare under resten av systemutvecklingsarbetet. Om de grundläggande strukturerna finns klart definierade redan i verksamhetsmodellen minskar behovet att stora modellförändringar i ett senare läge.

Ett annat viktigt avgörande är om modellerna ska visa den nuvarande verksamheten eller den framtida, önskade, verksamheten. Det finns ett klart värde i att starta med en analys av nuläget, d v s att försöka definiera det aktuella tillståndet i verksamheten. Med hjälp av kunskap om den nuvarande situationen är det möjligt att diskutera verksamheten och föreslå förändringar för att komma till rätta med eventuella problem.

Det kan vara svårt att hålla fokus på det nuvarande tillståndet eftersom deltagarna gradvis kan vilja "förbättra" modellerna genom att introducera önskade förändringar. En flytande övergång från analys av nuläge till definition av det önskade läget är inte principiellt felaktig men ställer stora krav på den systemutvecklare som leder arbetet för att inte övergången blir omedveten och sker okontrollerat. Det är extremt viktigt att hela tiden vara klart medveten om vad som modelleras.

3.3 Verksamhetsregler

I Tempora är verksamhetsregler en mycket viktig del av den konceptuella modellen tillsammans med begrepp och funktioner. Under analysprocessen läggs det därför stor vikt vid specificering av regler. I de tidiga faserna av analysen uttrycks verksamhetsreglerna informellt på vanlig svenska för att sedan under den senare delen av informationssystemanalysen översättas till ett formellt regelspråk.

I analysarbetet används välkända drivande frågor och metoder för specifikation av begrepps- och funktionsmodeller som utvecklats under lång tids användning och erfarenhet av dessa modelltyper. När det gäller regelmodellering saknas denna bakgrund och inom Tempora har det utvecklats förslag till drivande frågor för regelmodellering. Några exempel är:

- Vad finns det för allmänna restriktioner i verksamheten (även oskrivna lagar och förordningar)?

- Vad finns det för olika riktlinjer för verksamheten?

Givet en viss regel:

- Vilka mål bidrar regeln till att uppfylla? Hur realiseras detta mål?
- Är regeln en snävare variant av en annan regel?
- Vilka regler gäller för ett visst begrepp?
- Hur skapas en instans av ett begrepp? Hur försvinner den?
- Om en regel gäller för en viss begreppsklass som är en specialisering av ett generellt begrepp, gäller den även för det generella begreppet?

Det stora antalet regler i en verksamhet beror huvudsakligen på de otaliga undantag till huvudreglerna som måste definieras, där varje undantag resulterar i en ny regel.

Under verksamhetsanalysen ges undantagen relativt lite uppmärksamhet till förmån för huvudreglerna i verksamheten. Hantering av undantag får sedan ökad betydelse allteftersom vi närmar oss designfasen. Det finns stora vinster att göra genom att koncentrera intresset mot huvudreglerna och att dessutom försöka ge dessa en positiv formulering. Regler med positiv formulering kan ge datorsystemet helt nytt beteende. Regeln:

Hindra oauktorerad åtkomst

ger ett restriktivt system som försvårar för användaren, medan regeln:

Se till att användaren får den information han behöver

leder till ett system som hjälper användaren att hitta den information som han behöver (t ex genom att läsa in nödvändig information i primärminne när användaren loggar in i systemet).

3.3.1 Samspel mellan begrepp och regler

Begreppsmodell och verksamhetsregler har ett mycket stort beroende av varandra och är nära sammankopplade på grund av att begreppen utgör det språk med vilket vi uttrycker utsagor om verksamheten. Formulering av regler stärker begreppsmodellen och gör den mer precis eftersom reglerna uttryckes med hjälp av begreppen. Användningen av verksamhetsregler påverkar inte oväntat begreppsmodellens struktur eftersom reglernas komplexitet starkt beror av valet av begreppsstruk-

tur. En verksamhetsregels formulering är till mycket stor del beroende av begreppsmodellens detaljer.

Trots att regelformulering är ett viktigt inslag i att skapa en korrekt begreppsmodell är det paradoxalt nog så att det i många fall krävs en väldefinierad begreppsmodell för att *kunna* formulera regler. I de tidiga faserna av analysarbetet förändras modellerna snabbt. Begrepp byts ut, får annan betydelse eller sammanbinds i nya strukturer. Under dessa förhållanden kan det också vara svårt att hitta stabila regelformuleringar.

Man kan grovt urskilja två typer av regler, modellförklarande regler som preciserar begreppsmodellens betydelse samt policy-regler som bör anses som de "riktiga" verksamhetsreglerna. En vanlig användning av modellförklarande regler är att beskriva förhållanden mellan redundanta begrepp. De modellförklarande reglerna härrör alltså inte direkt från verksamheten utan är helt beroende av hur verksamheten modellerats i begreppsmodellen.

De modellförklarande reglerna kan sägas vara mer orienterade mot databaser än mot konceptuella modeller vilket gör att dessa bör utelämnas i verksamhetsmodellen. De har sin plats i informationssystemmodellen där de bidrar till att minska redundans och tvetydighet. I verksamhetsmodellen ska man däremot fokusera på de regler som är av mera konceptuell karaktär, de sanna verksamhetsreglerna.

3.3.2 Formalisering av regler i formellt språk

Översättning av verksamhetsregler uttryckta i naturligt språk till ett formellt regelspråk är en viktig del i arbetet eftersom det innebär en ordentlig genomgång av reglerna, som måste vara mycket väldefinierade för att kunna uttryckas i en formell syntax. Som ett resultat av översättningen kan begrepps- och funktionsmodellerna behöva förändras eftersom översättningen från informella till formella regler leder till en bättre förståelse av verksamheten.

För att en regel ska kunna uttryckas i det formella språket krävs det att alla begrepp som refereras i regeln i förväg är definierade i begreppsmodellen. Det som sagts ovan om beroendet mellan begreppsmodell och regler gäller i än högre grad när reglerna ska uttryckas formellt. Det är helt enkelt inte möjligt att uttrycka formella regler som refererar till ej definierade begrepp.

Vi arbetar huvudsakligen med relativt stringenta formuleringar av informella regler i naturligt språk som sedan översätts av Tempora-experten. För både informella och, i synnerhet, formella regler gäller dock att

läsligheten ökar om de inte är stora och komplexa utan hellre uppdelade i flera enkla regler.

I de senare faserna av informationssystemanalysen sjunker antalet nyupptäckta regler. Översättningen av de informella reglerna bör lämpligen påbörjas i detta ganska stabila läge även om man fortfarande måste vara beredd att ompröva delar av modellen.

3.4 Avgränsning och detaljering

Temporas verksamhetsanalys följs av informationssystemanalys. Detta innebär att arbetet fokuseras på de delar av beskrivningarna som avses ingå i det önskade informationssystemet. En avgränsning till begrepp och funktioner relevanta för systemet görs, detaljerad modellering genomförs inom avgränsningen, beskrivningarna formaliseras och onödig redundans elimineras.

Härledningsregler kan införas i denna fas för att avspegla informationsbehov som det framtida systemet måste tillfredsställa. Processerna i funktionsmodellen specificeras med funktionsbeskrivningar och definitionsblanketter för objekten i modellerna fylls i.

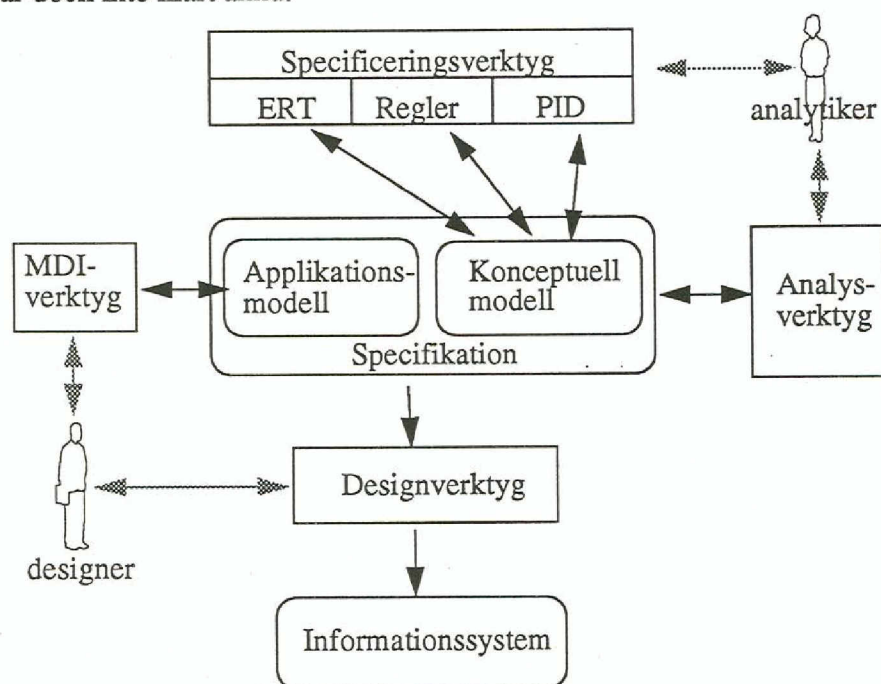
Det är ytterst viktigt att resultatet från det sista arbetssteget i informationssystemanalysen håller tillräcklig precision och kvalitet eftersom det ska kunna användas som underlag för design av informationssystemet.

Tidsmarkering av begrepp och samband ingår som en aktivitet i informationssystemanalysen men redan under verksamhetsanalysen kan olika typer av tidsoperatorer utnyttjas i regeluttryck. Detta behövs i de fall då regeln innehåller någon form av tidsreferens. Detta innebär att samtliga begrepp i begreppsmodellen underförstått ska anses som tidsmarkerade i denna fas. Regler som innehåller en sådan temporal dimension kan dessutom användas som ett medel för att under informationssystemanalysen identifiera behov av historisk information.

Vi vill påpeka att informationsbehov som dyker upp spontant inte ska försummas, även om det råkar vara i "fel" analysfas. De bör noteras när de dyker upp för att undvika att informationsbehov glöms bort.

4. Datorstöd

I det här kapitlet ska vi ge en kort beskrivning av den uppsättning datorstöd och det runtime-system som utvecklats inom ramen för Tempora. Som framgår av figur 11 tänkes analytikern samverka med både specificeringsverktygen och analysverktygen. De förra är realiserade med hjälp av SISU's CASE-verktyg Ramatic och består enkelt uttryckt av två grafiska editorer, en för ERT och en för PID, kompletterade med en del speciella formulär, bl a för att definiera regler. I dessa editorer sker viss syntaxkontroll. I analysverktyget som är en fristående applikation sker djupare konsistens- och kvalitetskontroller. Även andra analys- och valideringshjälpmedel kan tänkas ingå. Vilken utformning dessa får är dock inte klart ännu.



Figur 11. Temporas systemutvecklingsmiljö.

11

Designern använder ett MDI-verktyg för att utforma användargränssnittet. Han måste också tillföra information till specifikationen för att tala om hur reglerna ska utföras och hur transaktioner ska definieras. I designverktygen ingår också en databasgenerator som delvis automatiskt kan generera ett denormaliserat databasschema för att åstadkomma korta svarstider för dom frågor som kommer att ställas. Den konceptuella modellen lagras i Ramatics databas (CS5) och översätts sedan till en objektorienterad Prolog (Probe) där man också lägger app-

likationsmodellen. Utifrån detta genereras ett runtime-system som består av en 'temporal' regelhanterare skriven i Prolog som i sin tur lagrar information i databashanteringssystemet Sybase.

5. Tempora

Mycket kort syftar projektet Tempora, som framgått ovan, till att utveckla och förbättra metodik, språk och verktyg för specifikation och utveckling av informationssystem (IS) baserat på explicit representation av verksamhetsregler och inkluderande tidsdimensionen. Tempora är ett Esprit-projekt, d v s det finansieras delvis av EG inom det s.k. ESPRIT-programmet. Projektet beräknas pågå till slutet av 1993 och involverar både kommersiella företag och universitet. Deltagare förutom SISU är:

| | |
|---------------------|------------|
| Imperial College | London |
| Université de Liège | Liège |
| UMIST | Manchester |
| LPA | London |
| BIM | Bryssel |
| Hitec | Atén |
| Sintef | Trondheim |

SISU:s medverkan i Tempora sponsras av Nutek (fd STU), Ericsson och Televerket.

Författarna till föreliggande rapport deltar alla i Tempora. Projektresultaten är dock i stor utsträckning resultatet av ett lagarbete, varför författarna ber att få framföra sitt tack till övriga i projektet arbetande personer.

6. Referenser

- [1] P. P. Chen, "The Entity-Relationship Model - Toward a Unified View of Data", ACM Transactions on Database Systems, vol. 1, no. 1, pp. 9-36, 1976.
- [2] TEMPORA Consortium, "Concepts Manual", E2469, UMIST, 1991.
- [3] P. McBrien, M. Niézette, D. Pantazis, A.-H. Seltveit, U. Sundin, C. Theodoulidis, G. Tziallas and R. Wohed, "A Rule Language to Capture and Model Business Policy Specifications", in "Advanced Information Systems Engineering - CAiSE 91", Ed. R. Andersen, J. A. Bubenko jr. and A. Sølvberg, pp. 307-316, Trondheim, Norway, Springer-Verlag, 1991.
- [4] T. Kinnula, U. Persson, R. Wohed, B Wangler, Erfarenheter av modellering enligt Tempora, SISU, Triad, 1991.
- [5] H. Willars, Handbok i ABC-metoden, Plandata Strategi, 1988.

TRIAD utvecklar IA

Televerket har just tagit första steget in i sin nya IA-organisation och Posten håller på att bygga upp sin nya DA-organisation. Båda organisationerna har sett nyttan att inför 90-talet gå vidare tillsammans i TRIAD-projektet som drivs tillsammans med SISU. Statskontoret deltar också i projektet för att på sikt kunna föra ut nya synsätt och hjälpmedel inom den civila statliga sektorn.

Ericsson Data Services deltar med tyngdpunkten i den del som handlar om att utveckla kompetenta modelleringsledare, delprojektet "Avancerad utbildning för modelleringsledare".

Modelleringsmetoder är centrala i bedrivandet av verksamheten inom informationsadministrationen. Därför arbetar ett delprojekt med utvecklandet av "nästa generation modelleringsmetod" som skall sättas i händerna på informationsadministratören. Siktet är att fördjupa och bredda dagens modelleringsmetoder och där hämta in kunskap från pågående forskning och utveckling internationellt. (faktaruta om IAS91).

Som stöd för informationsadministrationen behövs verktyg. Inom TRIAD arbetar man där inom två områden, kataloger och verktyg.

Delprojektet kataloger arbetar dels med att utforma den informationsmodell som måste kunna täckas av en katalog, dels med att granska och följa utvecklingen av produkter inom området t ex IBM:s "Repository" och Digital's "CDD". Dessutom följer man standardiseringen internationellt kring IRDS. För parterna i projektet liksom för andra organisationer är detta ett tungt område både vad gäller kommande investeringar ekonomiskt och vad gäller kompetenta resurser för en kommande övergång till "repository-världen". - Det inledande skedet syftar till att bygga upp en kunskapsplattform, som sedan kommer att kunna utnyttjas för kravställande och planering och genomförande av övergång från dagens kataloghantering till morgondagens.

Den andra verktygshanterande delen inom TRIAD-projektet, delprojektet "verktyg för informationsadministration", syftar till att ta fram verktyg för uttag och dokumentering av modeller. Betoningen ligger på människa datorgränssnitt och i första skedet görs utveckling av HYBRIS-gränssnittet med prototyper för Posten och för Televerket.

För att hålla ett helhetsperspektiv på projektets delar och för att ha inpassningen av funktionen Informationsadministration i organisationens övriga verksamhet arbetar delprojektet "Krav på IA". I delprojektet arbetar man dels med att kartlägga dagens krav på dataadministration och projicera till morgondagens krav på IA. Dessutom skall man skapa en bild av IA-verksamhetens innehåll och organisation. Från detta i sin tur ställer man krav

på övriga delprojekt. Vilka krav skall ställas på kompetens, metoder, hjälpmedel typ kataloger och gränssnitt?

TRIAD projektet är stort

Budgeten för TRIAD-projektet löper på 10 MSEK per år under en treårsperiod som startar vid kalenderåret 1991 års början och som alltså beräknas avslutad vid utgången av 1993.

TRIAD-projektet är ett tillämpningsprojekt

Det innebär att parterna, Televerket, Posten, Statskontoret, EDS och SISU går in med såväl persontidssatsningar som ekonomiska och att STU, Styrelsen för Teknisk Utveckling, bidrar med ett ekonomiskt tillskott som svarar mot ungefär 40 % av den insatta persontiden.

Öppet för fler deltagare

Parterna i TRIAD-projektet vill gärna öka tempot och bredda perspektivet och vill därför gärna ha fler parter in i projektet. Dessa parter får då enligt SISU:s tårtprincip "betala för en tårtbit, men ät hela tårtan", tillgång till projektets resultat med en insats som ger stor "price performance".

Nya deltagare kan gå in i hela projektet eller i det eller de delprojekt som verkar intressantast. En förutsättning är att man framförallt är beredd att satsa kompetent personal. För de flesta intressenter bord detta vara ett utmärkt sätt att driva personalutveckling för personer t ex inom DA-området, samtidigt som man bygger upp beredskapen inför 90-talets IA-verksamhet.

Kompetensutveckling viktigt resultat

En viktig effekt för parterna av deras medverkan i TRIAD är kompetensutveckling. Man satsar på att ta in personer som så småningom eller redan idag arbetar med DA och IA för att ge dem en djup och "frontlinje"-mässig kompetens. Detta skall utnyttjas när man successivt för in resultaten i den egna organisationen. Projektdeltagarna har alltså en viktig roll som kunskapsförmedlare i den egna organisationen. Dessutom ger projektarbetet deltagarna tillfälle till en egen utveckling inom det professionella området som är unik.

Informationsspridning

Det sjätte delprojektet "Informationsspridning" har till uppgift att sörja för att i första hand parterna men också SISU:s övriga intressenter successivt kan följa och tillgodogöra sig resultat från TRIADprojektet. Seminarier, rapporter och referensgruppsverksamhet är led i den verksamheten.