

Krav på IA
Nästa Generation Modellering
Avancerad utbildning för handledare
Katalogprinciper
Uttagsystem
Informationspridning

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

Rapport N nr 2: Generering av konceptuella modeller från policydokument

Rapport N nr 3: Espritprojektet Tempora

Rapport N nr 4: Prövning av regelbaserad metodik inom Posten

Rapport N nr 5: En kokbok i remodellering - utkast

Rapport N nr 6: Datorstöd för modellintegration

**Rapport N nr 7: Kokbok för modellering**

## **Kokbok för modellering**

**Stig Berild**  
**SISU**

### **Spridningsförbehåll:**

Denna rapport får endast spridas och användas inom de organisationer som deltar som parter i TRIAD-projektet. © TRIAD-parterna juni 1992

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

# Kokbok för modellering

Inledning .....	1
Kokbokens vidare liv.....	3
Alternativa sätt att grafiskt presentera en modell .....	5
Antingen - eller diskussion.....	9
Domän eller objekttyp?.....	16
Hur gör man med härledbar information?.....	20
M:M-sambandstyp eller objekttyp? .....	22
Olika modelleringsansatser.....	24
Många eller få domäner?.....	29
Typ/förekomst enligt ovan contra mängd .....	34
Namngivningsprinciper .....	37
Närbesläktade attributtyper eller "typifierad" objekttyp?.....	39
Slå ihop eller dela upp objekttyper? .....	43
Sammansatt domän eller objekttyp? .....	49
Distinktion mellan typ och förekomst.....	52



# Inledning

Modellering är ofta ett mödosamt arbete där lyhördhet och kompromissvilja avsevärt gynnar kvalitet på resultat. Problemen finns först och främst att hämta i den verklighet som ska modelleras. Verkligheter är komplexa företeelser, som inte utan vidare låter sig beskrivas i en överblickbar, lättfattlig modell. Det vid första påseende relativt enkla blir i allmänhet mycket snart betydligt mer komplicerat när precision och detaljer behöver tillföras modellen. Dessutom har inblandade parter ofta lätt att komma överens i det stora och sällan speciellt förpliktigande (problemen "sopas under mattan"). Detaljer tillåter däremot inga dimridåer, modeller har en erkänd förmåga att avslöja oklarheter. Överensstämmande uppfattningar på detaljnivå erhålls ofta först efter träget arbete.

Detta om verkligheten. En viktig förutsättning för lyckade resultat är i de flesta arbetssituationer tillgång till bra verktyg. Vid modellering är det avgörande verktyget ett modelleringsspråk och en notation med vars hjälp verklighetens abstraktioner kan formuleras på ett sätt som känns naturligt att arbeta med (lagom avvägning mellan enkelhet och uttryckskraft). Det ska vara möjligt att åstadkomma en modell, som upplevs vara en god avspeglning av det modellerade verkligheten. Ett datorstöd för att skapa, lagra, ändra samt inte minst presentera modeller i olika former och detaljeringsgrad underlättar givetvis också arbetet avsevärt.

Tillgång till kvalificerade modellerare (modelleringsledare) är också en viktig förutsättning för ett kvalitativt acceptabelt förslag. På något sätt verkar det svårt att inom modellering tillägna sig en fast och solid kunskaps- och erfarenhetsbas. Visserligen lär sig en modellerare efterhand vissa handgrepp och sätt att angripa en problemsituation. Påtagligt ofta räcker inte de egna handgreppen till. Varje modelleringssituation är påfallande ofta unik i något avseende.

Lika ofta dyker det under modelleringen upp typsituationer, företeelser i verkligheten som har stora likheter med varandra. Kanske ligger skillnaden endast i små detaljer. I dessa sammanhang ligger det nära till hands att använda en enhetlig modelleringsansats - såvida man kommer ihåg hur man gjorde förra gången. Tänk om det fanns en lättillgänglig erfarenhetsbas som stöd.

Också erfarna modellerare klagat ofta på att de visserligen ibland ser "ljuset" (i form av en hägring?) men lika ofta känner den totala tomheten - behovet av att kasta alla hittillsvarande förslag i en papperskorg och börja om från "scratch" igen. Åtminstone finns ofta ett behov av att få tips och rekommendationer från andra som befunnit sig i liknande

problemsituationer. Tänk om det fanns en lättillgänglig erfarenhetsbas som stöd.

Återanvändning ligger ju i tiden. Produktiviteten ökar i arbetet. Enhetlighet ökar läsbarhet. Igenkännande ökar chans till förståelse även gentemot icke modelleringsexperter, osv. Enhetlighet underlättar personbyten i en modelleringsgrupp. Introduktionstiden för nya personer blir kort. De kan koncentrera sig på sakfrågor och inte på olika "dialekter". Tänk om det fanns en lättillgänglig erfarenhetsbas som stöd.

Mycket talar för en receptsamling kring modelleringserfarenheter, en modelleringskokbok.

Denna kokbok har till uppgift att vara ett opretentiöst forum för diverse modelleringsproblem och deras tänkbara lösningar. Eftersom det i dessa sammanhang inte finns något rätt eller fel, ska kokboken givetvis inte betraktas som en "lagbok". Frågan är om det överhuvudtaget är realistiskt att komma fram till någon typ av gemensam uppfattning kring hur olika situationer ska lösas. Kokboken kan åtminstone vara en samlingsplats för olika lösningar och erfarenheter, en förslagslåda var och en fritt får botanisera i. Förhoppningsvis kommer det att visa sig att vissa lösningar får en sådan stabilitet och acceptans att de successivt kan upphöjas till vedertagen policy.

Kokboken ska på sikt förhoppningsvis fylla ett behov hos såväl förstagångsmodellörer som erfarna modellerings-"rävar".



# Kokbokens vidare liv

För att underlätta påfyllnad av nya recept föreslås kompletteringar tills vidare kunna ske under fria former. Krav på genomdiskutering och samordning ska inte behövas. Vi vet ju vilken tid det tar för modellörer att bli överens. Just nu är det viktigt att få in så mycket material som möjligt. Många personer har stor modelleringserfarenhet. Skriv ner och delge andra. Känn inga som helst ambitionshinder. Tar författandet för lång tid, blir inget skrivet. Mao, kvantitet framför kvalitet! Det enda krav som ställs gäller vissa formalia enligt nedan. Så småningom kan vi, baserat på tillägnade erfarenheter, diskussioner mm "plocka russinen ur kakan", dvs formulera mer allmängiltiga riktlinjer och rekommendationer med betoning på kvalitet framför kvantitet.

Kanske kommer vi att parallellt föra en arbetsversion och en slutversion av kokboken. Just nu gäller arbetsversionen.

## Alltså:

a. Var och en som vill informera om och ge spridning åt en intressant eller erfarenhetsbaserad lösning på ett avgränsat modelleringsproblem, fattar tangentbordet och skriver sitt recept. 1-5 sidor borde räcka? Grafer bör om möjligt tillverkas med hjälp av Business Modeler.

Den som önskar revidera ett tidigare inlägg formulerar den reviderade texten.

b. Recept, reviderat recept eller begäran om borttag av recept lämnas till ansvarig samordnare (vem?).

c. Med jämna mellanrum utges en ny release av kokboken (kvantitetsversionen).

d. Kokboken ska finnas i pappersform. Inget hindrar att därutöver någon typ av datorstöd byggs upp för praktisk distribution och sökning. Behöver utredas.

### Formalia:

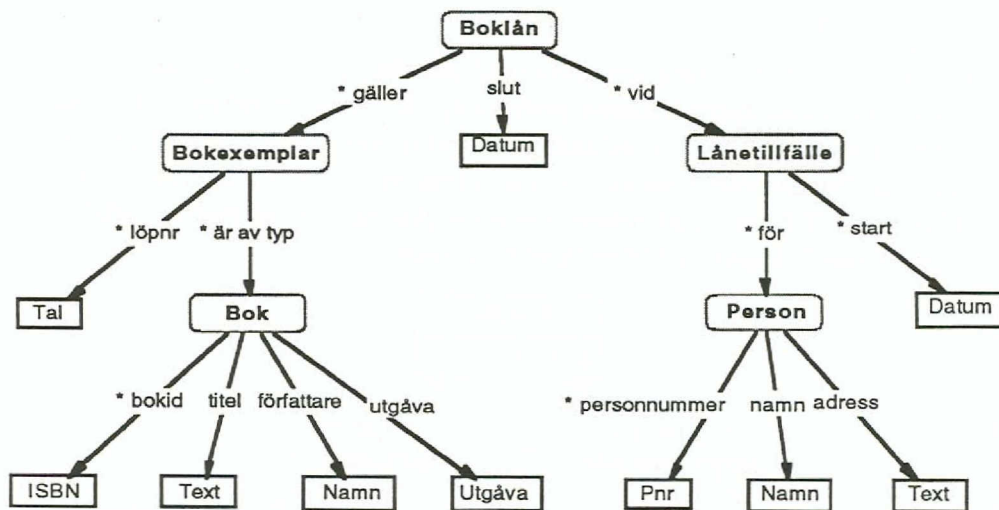
För varje recept anges

- Rubrik
- Texten.
- Receptlämnare (namn, tjänsteställe). För att förstå varför det kanske finns flera mycket olika recept på delvis samma problem. För att veta vem man ska diskutera med.
- Tillverkningsdatum.
- Om revidering, anges "Reviderad" efter datumet.
- Nyckelord. Eftersom recepten fylls på i datumordning behövs efterhand någon typ av klassificeringssystem för att underlätta sökningar. Exv kunde resp. Telmodbegrepp (objekttyp, sambandstyp, uttömmande, .... ) vara en initial uppsättning. Behöver utredas innan nyckelorden börjar användas.

# Alternativa sätt att grafiskt presentera en modell

Används ett datorstöd, sker presentation i enlighet med dess grafik och tillgängliga varianter. Diskussionen nedan är datorstödsberoende.

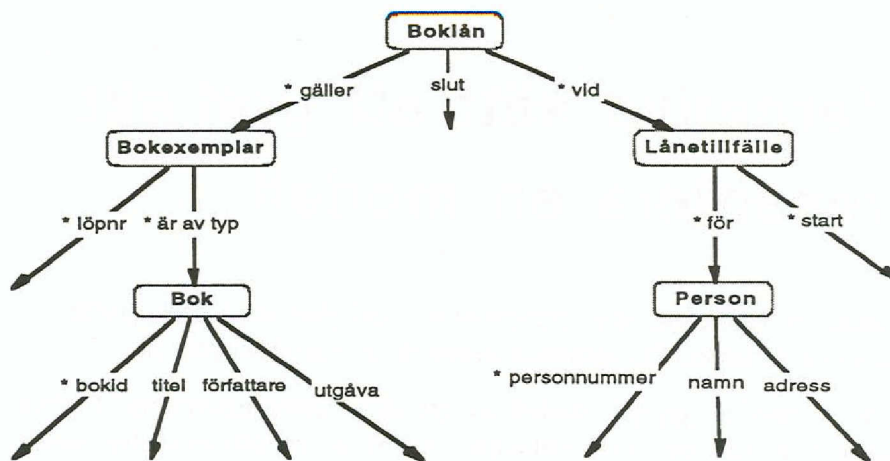
I figur 1 visas en grafisk modell av ett tänkt bibliotekssystem. I större tillämpningar kan en sådan ritteknik snabbt ge en överskådlig mängd objekt, domäner och samband med tillhörande namn, "man ser inte skogen för bara träd". Dess användbarhet inskränker sig till mindre snabbskisser.



Figur 1

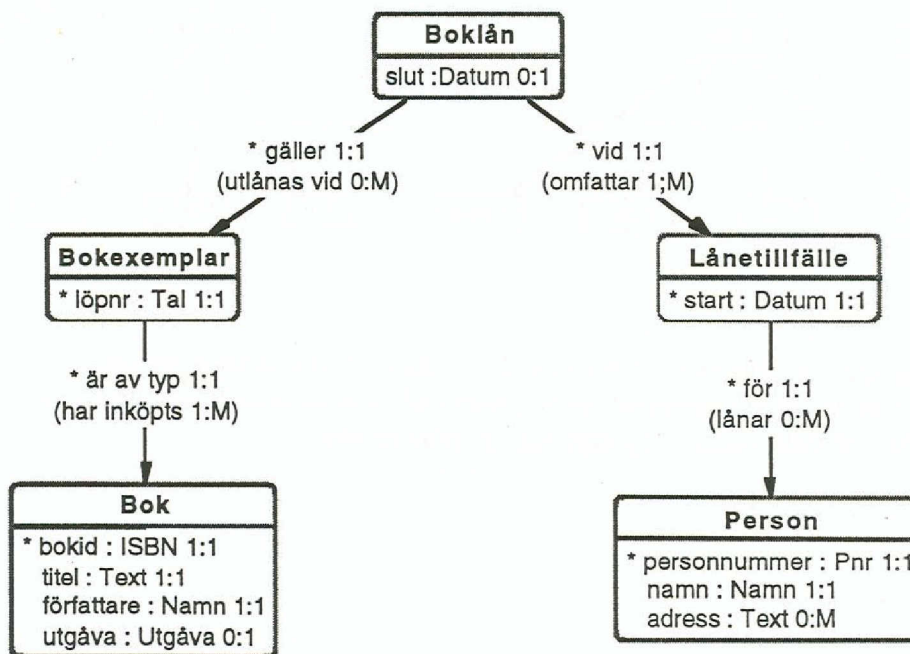
Antagligen är den enklare formen enligt figur 2, där domän utelämnats, lättare att ta till vid spontana behov. Distinktion mellan domän och objekttyp kan ju i ett visst modelleringsläge vara av mindre vikt.





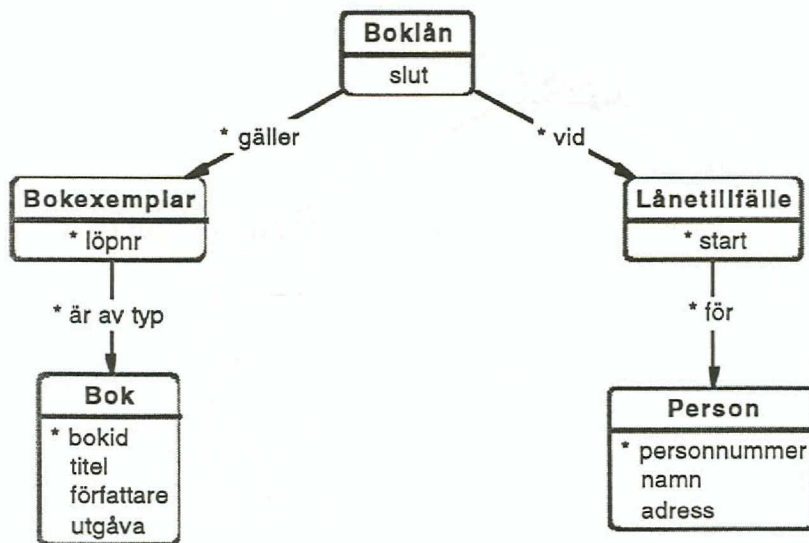
Figur 2

Ett sätt att kombinera hög detaljnivå med god läsbarhet är att gruppera in en objekttyps attributtyper i samma grafiska symbol som objekttypen. Då finns även utrymme för avbildningsrestriktioner och uppgifter om inverssambandsnamn utan att det ser trångt ut (figur 3). Denna notation tillämpas i Telmod. Kokbokens exempel presenteras normalt i Telmod-notation enligt någon av nedanstående varianter.



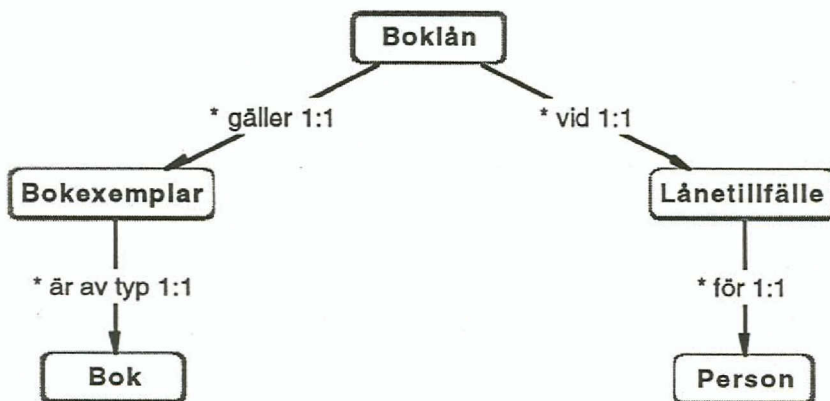
Figur 3

Figur 2, presenterad kompakt, blir enligt figur 4.



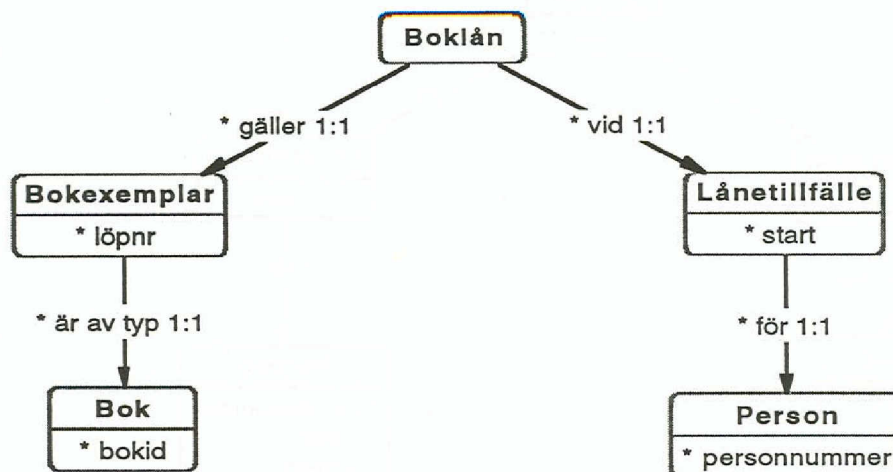
Figur 4

Överskådlighet kan ökas på bekostnad av detaljrikedom om endast objekten och deras inbördes kopplingar tas med. Fokus inriktas mot sambandstyperna, vilka representerar den kanske viktigaste semantiken. Figur 5 visar exempelmodellen i förenklat skick. I den mån mapping redan utretts bör den framgå av grafen. Den tar mycket lite plats och ger mycket klagörande information. Onödiga missförstånd kan kanske undvikas eller snabbt uppdagas och klaras ut. Däremot kan det ibland upplevas jobbigt att formulera inversnamnen. Semantiken framgår ofta klart av huvudriktningens namn. Mer bekymmersamt är att inversmappingen inte kommer med.



Figur 5

En kompromiss är att, utöver objekten, även ta med de attributtyper som ingår i identifieringen av objekttyperna för att ge en extra förklaring/precisering av respektive objekttyps innebörd.



Figur 6

Välj den grafiska presentationsteknik som tilltalar dem som arbetar med, respektive har att förstå modellen. Om ringar, rektanglar e.dyl används spelar inte så stor roll, men det man valt ska användas konsekvent. Arbetar man med ett datorstöd bör datorstödet notation användas även i manuellt ritade grafer för att uppnå enhetlighet. Detaljrikedom väljs beroende på syfte och vem som ska läsa/förstå grafen. Den tvehågsne rekommenderas använda ritteknik enligt figur 3. Att avstå detaljrikedom för att det innebär mycket skrivarbete brukar sällan vara en god strategi. Tiden att ta fram uppgifterna överstiger vida tiden att sätta dem på pränt.



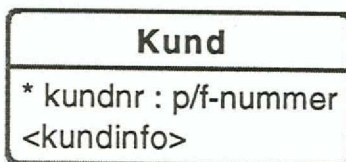
# Antingen - eller diskussion

Antag att vi vill beskriva att en Kund antingen kan vara en person eller ett företag.

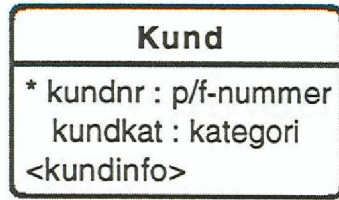
Vi kan dela upp problemet i tre del-problem med olika förutsättningar. Alternativ a är ett specialfall av alternativ b som i sin tur är ett specialfall av alternativ c. I fallet a är samtliga angivna alternativ i princip acceptabla. Med b-förutsättningen väljer man inom b och c medan c-förutsättningen kräver c-lösning. A- och b-alternativen har tagits med som exempel på ett "teknik-orienterat" synsätt på modelleringen. Dock är det inte osannolikt att i en modelleringssituation hamna i den typen av resonemang på grund av ett medvetet eller omedvetet hänsynstagande till de speciella förutsättningarna under modelleringen. Resonemangen kan därför vara rimliga att inkludera här, men ska knappast ses som en rekommendation.

## a. Identifierare för respektive objekt är "kund-unik".

Genom att man kan se på identifieraren vilken objekt-tillhörighet som gäller, uppstår aldrig problemet med att avgöra om en kund är en person eller ett företag. Skapa ett kundnummer-attribut under objekttypen Kund som om det vore ett vanligt kundnummer. Attributet kundnr måste noga uttryckas till sin innebörd eftersom det tjänstgör som en ersättning för beskrivning av antingen/eller-identifierare (något som inte finns i begreppsapparatus uttrycksrepertoar). P/f-nummer står för värdeförrådet "person- och företagsnummer". Att på detta sätt, som i figur 1, utnyttja den inbyggda semantiken i kundnumret avråds bestämt från. Kontroll av identifierarens uppbyggnad för att avgöra kundtyp, kan undvikas om man i stället kompletterar med ett typ-attribut där de möjliga värdena är PERSON eller FÖRETAG, figur 2.



Figur 1

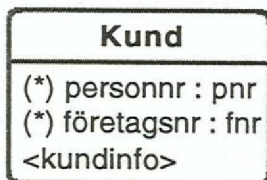


Figur 2

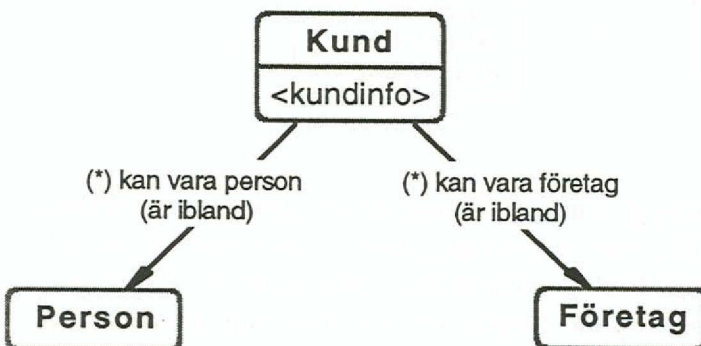
Ett annat alternativ är att explicit ha två attribut, ett för varje typ av kundnr, figur 3. Vilket attribut som anges blir beroende på vilken typ av kund som är aktuell.

Finns alla tre objekten i schemat kan man istället komplettera med sambandstyp till Person och Företag från Kund, figur 4. Vi förutsätter att kundnumret är detsamma som respektive objekttyps identifiering varvid behovet av specifikt kundnr bortfaller. Identifieringen återfinns istället under beskrivningen av Person respektive Företag.

Eftersom inte antingen/eller-identifierare finns bland modelleringsbegreppen (skulle kunna läggas till) måste detta förhållande noteras separat tillsammans med alla andra typer av villkor och omständigheter som inte heller bestäms grafiskt. En mer explicit notering av förhållandet kunde vara att sätta asterisken inom parentes så som i figurerna 3 och 4. (Har man sammansatta identifierare får man ta till numreringar e dyl.) Den renlärige tillåter helt enkelt inte alternativen enligt figur 3 och 4 förrän begreppsapparaten utvidgats med motsvarande konstruktioner.



Figur 3



Figur 4

Varför inte enkelt lösa problematiken med ett generaliserings samband (is\_a)? En kund är ju alltid antingen en person eller ett företag. Visserligen, men det villkoret reglerar endast om det är fråga om en uttömmande eller icke uttömmande generalisering. En viktigare förutsättning är att den specialiserade objekttypen också alltid figurerar i den mer generella rollen. Någon sådan förutsättning finns inte given här. Det kan exv mycket väl finnas personer som inte är kunder.

Generaliserings samband är, rätt använda, kraftfulla instrument men kräver eftertanke! Tänk efter extra noga vid situationer där inte de specialiserade objekttyperna ärver sin identifiering från överordnad objekttyp.

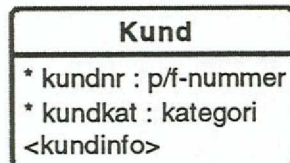


**b. Unika identiteter men inte möjligt att syntaktiskt särskilja.**

Välj något av alternativen i figurerna 2 - 4.

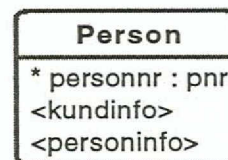
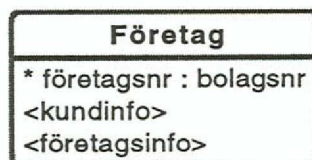
**c. Ej säkert helt olika identitetsmängder.**

Detta är den normala situationen. Alternativen a och b är som tidigare nämnts specialfall som istället mycket väl kan och bör modelleras enligt nedan. Existerar inte person och företag i modellen förs kundkategori in i identifieringen.



Figur 5

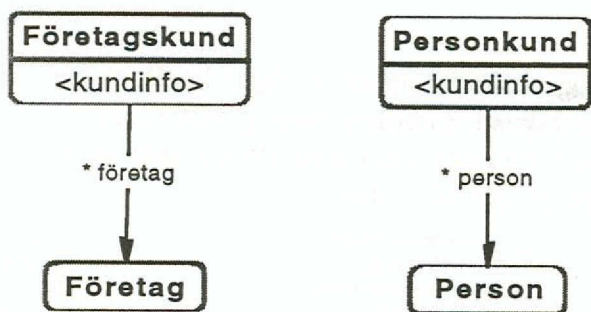
Existerar däremot person och företag finns flera alternativa lösningar. I den första placerar vi in kundinformationen under respektive objekttyp.



Figur 6

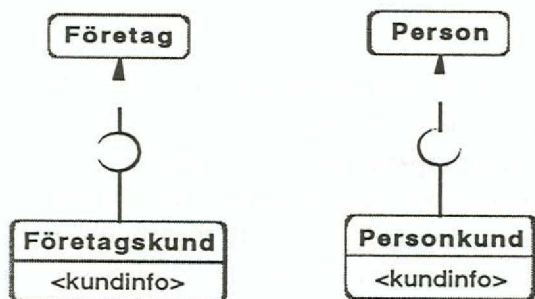
Lösningen är inte helt "snygg" eftersom samma typ av information återfinns under flera objekttyper. Har man delvis olika kundinfo beroende på om det är person eller företag kan förslaget accepteras, speciellt om tillämpningen kommer att hantera resp. kundinfo i olika formulär eller rapporter, dvs om synen på informationen ändå är uppdelad.

En snyggare modell med dessa förutsättningar är nog ändå figur 7, nedan. Den lokaliserar kundinfo endast till de företag och personer som verkligen är kunder och är därmed mer precis i sin beskrivning, speciellt om de, som är kunder endast utgör en begränsad del av respektive population.



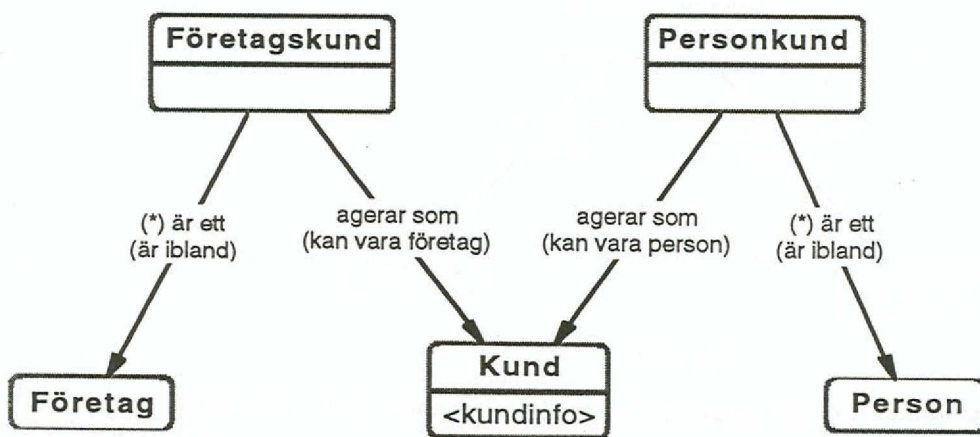
Figur 7

Här kan antagligen generaliserings samband användas som ett alternativ. Se figur 8. {terigen krävs vaksamhet. Kund-aspekten måste avse en generell roll hos företag respektive person, dvs att 1:1-förhållande gäller. Står exv Företagskund för ett företag som kund hos viss leverantör, uppstår ju en Företagskundsförekomst för varje leverantör där företaget är kund. Vi har ett M:1-samband som omöjliggör upprättandet av ett generaliserings samband.



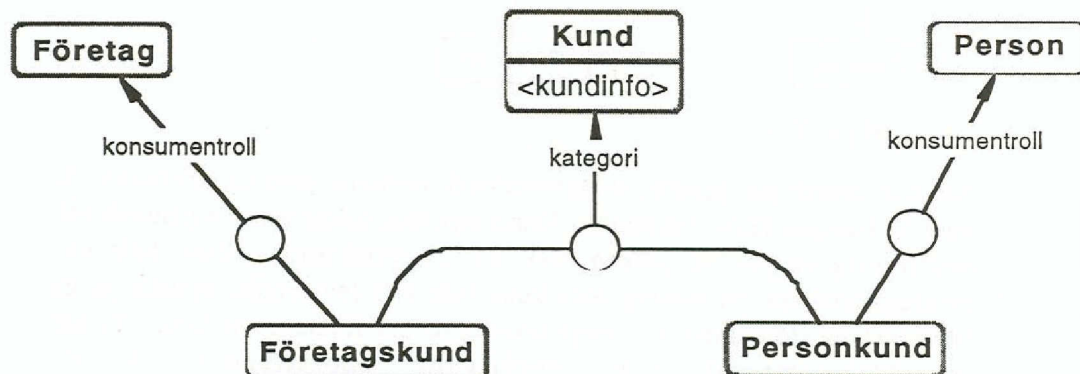
Figur 8

Hanteras alla kunder i stort sett lika bör det också finnas ett kund-objekt dit kundinformationen hänförs.



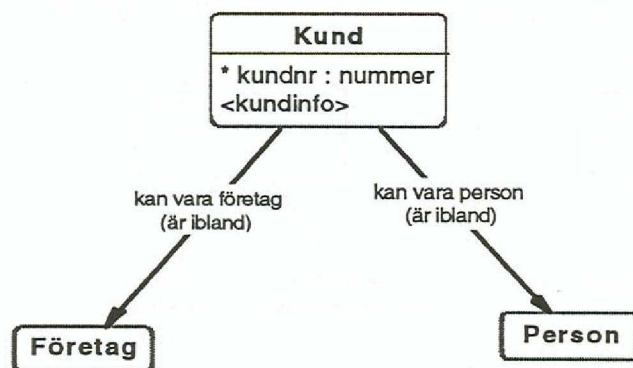
Figur 9

Givetvis kan även här generaliseringssamband upprättas, figur 10. Dock har vi nu två supertyper för exv Företagskund, nämligen Företag och Kund. Vi får så kallade "multipla arv", vars för- och nackdelar brukar diskuteras livligt. Vi ser här inga argument för att inte tillåta dem.



Figur 10

Problemet är hur detta fristående kundobjekt ska identifieras. Där finns nu all information om en kund, förutom identifieringen av vem som har kundkaraktistiken ifråga. Finns ingen separat kundidentifikation, typ kundnummer, måste tillräcklig delmängd av beskrivningen tillsammans utgöra identifieraren. Detta kan vara anledningen till att man i dylika situationer väljer att etablera en separat kundidentifierare. Finns det kvar separat företagskund- eller personkund-information kan det vara lämpligt att ha kvar dessa objekttyper, i annat fall kan de ersättas med attribut direkt till företag resp. person varvid vi kommit åter till figur 4. Skillnaden är att kundnummer nu finns som specifikt attribut för kund-objektet.



Figur 11

Krångla inte till datamodellen enligt alternativen a och b, dessa bygger på förutsättningar som mycket väl kan komma att ändras under tillämpningens livstid. Välj den lösning under c som är lämpligast med hänsyn till hur modellen för övrigt är uppbyggd.



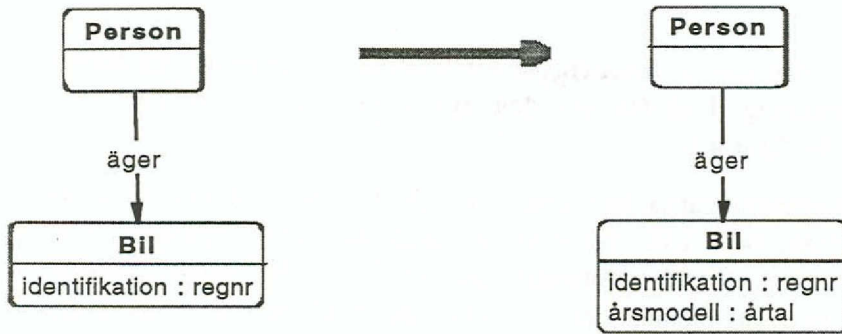
## Domän eller objekttyp?

I en tillämpning där man endast är intresserad av exempelvis en bils registreringsnummer och inget mer, ligger det nära till hands att representera bilarna i form av endast domän i stället för som en objekttyp Bil med den enda attributtypen regnr. Under pågående applikationsmodellering kan det vara svårt att avgöra om det kommer att "dyka upp" fler än den hittills enda attributtypen. Arbetar man helt behovsorienterat bör således domänen regnr gälla tills ytterligare attribut för bil efterfrågas, varvid Bil uppstår med två (eller flera) attribut. Ägar-attributtypen blir en sambandstyp. På detta sätt skulle ett antal objekttyper kunna genomlöpa "förstadiet" domän. Figur 1.



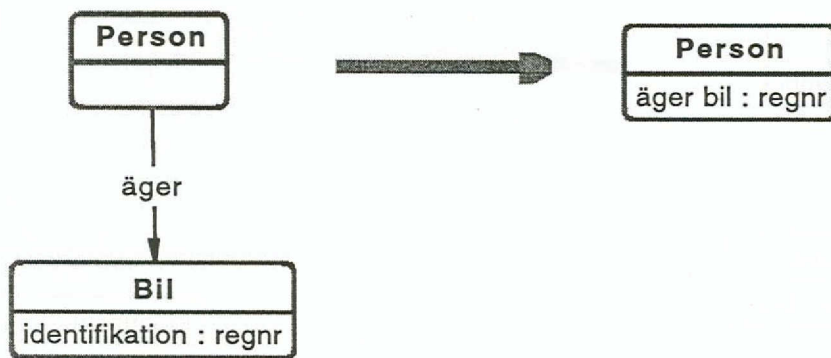
Figur 1

I praktiken uppstår sällan problemet. Man vet ganska väl vad som behöver bli objekttyp och vad som kan kvarstå som domän, även om de exakta informationsbehoven ännu ej presenterats. Vägledande kan inte minst namngivningen vara. Talar man om BILar under modelleringen tänker man sig nog dessa som objekt medan REGNR utan tvekan är domän. Observera den i andra modeller förekommande distinktionen mellan lexikala och icke-lexikala objekt (domän - objekt). Man får aldrig glömma att modeller aldrig är neutrala, de är beroende på aktuellt perspektiv och behov.



Figur 2

Skulle det vid modelleringens slut visa sig vara ett misstag är det både enkelt att hitta (objekt med endast ett attribut) och enkelt att åtgärda. Man tar bort attributtypen och låter domänen ta objekttypens plats.



Figur 3

Var dock uppmärksam på följande situation:

En bilfirma vill dokumentera både information om genomförda reparationer på en bil och genomförda bilbesiktningar. Man är bara intresserad av bilens registreringsnummer. Bilen har därmed bara en egenskap varvid det ligger nära till hands att modellera enligt figur 4 a.

Därmed har man inte sagt någonting om att samma domänvärde betyder samma bakomliggande objekt. Referensen går endast till ett värde. Ändrar man reparationens attributvärde kan det bero på

att det gällde en annan bil,

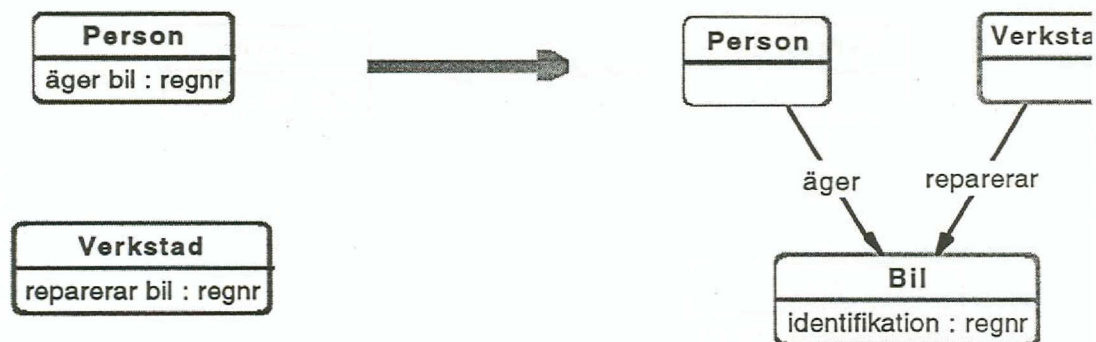
samma bil men att bilnumret var fel tidigare och nu rättats,

(i det generella fallet) att man valt ett annat alternativt registreringsnummer på samma bil.

Man kan aldrig referera till en företeelse genom en domän, bara till ett värde. Om vi alltså ändrar domänvärdet för visst attribut påverkar det inte alls andra attributreferenser till (det tidigare) värdet. Dessa kvarstår

oförändrade, vilket inte är avsikten om företeelsens identifierare ska ändras av någon anledning. I så fall ska den nya förutsättningen ju gälla alla referenserna. Se figur 4 b.

Antag att förutsättningarna inte indikerar något som helst behov av att uppleva objekttypen Bil. Persondata hanteras helt separerat från Verkstadsdata men via andra sambandstyper motiveras de i samma modell. Antag också att vi med kännedom om verksamheten vet att det i båda fallen är referenser till samma typ av företeelse Bil. Vad göra? Visserligen ska en modell endast spegla avgränsat problemområde, men det verkar ligga närmast till hands att välja alternativ 4 b, om inte annat för att till kommande användare av modellen förmedla kunskapen om att det är fråga om samma typ av företeelse.



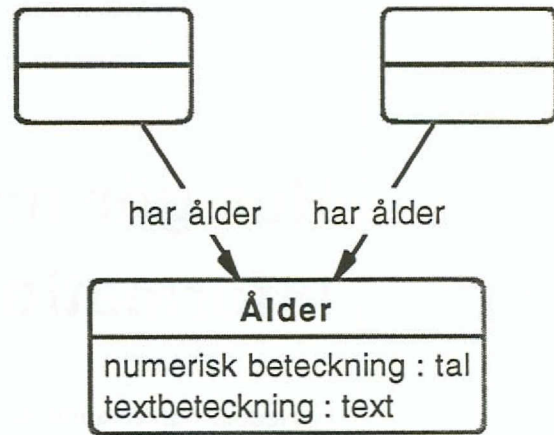
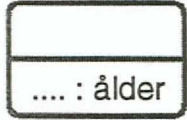
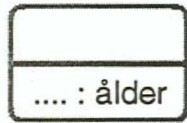
Figur 4 a      Figur 4 b

Vad vi hittills berört är bedömningen av om en typ av företeelse inom en tillämpning ska representeras som domän eller objekttyp. Vi konstaterade att behoven får avgöra. Vid tveksamhet, se det som objekttyp (tills vidare).

Det finns även en annan aspekt, som ibland dyker upp vid modelleringar: Kan domäner ha egenskaper?

Antag att vi har Ålder, vars värden vi ibland vill ange som tal, ibland som text. Oavsett vilket vill vi kunna se det som samma värde. Strängen "4" pekar på samma punkt som "Fyra". Refererar vi till viss ålder numeriskt, kan vi få fram dess övriga attribut, exv dess text-motsvarighet. Ålder har plötsligt blivit objekttyp med attributtyperna numerisk beteckning och textbeteckning. Eller?





Figur 5

Figur 6

I moderna utvecklingsverktyg finns ofta olika typer av redigeringsfunktioner för domänvärden som tal, datum, personnummer mm. Hur värdet uttrycks ligger där i användargränssnittet.

Numer brukar man också skilja på domän som representerande anstrakta värden och till domänen kopplade symboltyper vilka var och en representerar ett konkret sätt att uttrycka värdet. Oavsett om åldern uttrycks som tal eller text är det samma värde som ligger bakom. Alltså har vi här snarast en domän Ålder och två symboltyper Num-ålder och Text-ålder.

**Domäner kan inte beskrivas med attribut i datamodellen. Finns behovet måste domänen övergå till objekttyp. Se även diskussion kring sammansatta domäner.**

## Hur gör man med härledbar information?

Härledbar information är information som aldrig direkt uppdateras. Den efterfrågas i samband med olika typer av rapporter, statistik eller andra typer av sammanställningar. Inom en tillämpning kan det finnas en stor mängd sådana uppgifter. Sannolikt är dessa även föränderliga, dvs härledd information blir till sin konstruktion inaktuell eller uppstår i nya skepnader oftare än icke-härledd information. I grunden är all härledd information temporär, den kan sägas uppstå ur annan information i samband med ett informationsbehov. Att härledd information ändå ibland lagras permanent beror på prestandakrav.

Inom parentes sagt måste lagrad, härledbar information hanteras med största varsamhet i databaser så att inkonsistens aldrig uppstår mellan det lagrade härledbara och de basuppgifter och algoritmer ur vilka det härledbara tas fram. En grundregel bör vara att undvika lagring av härledbar information utom då mycket särskilda krav föreligger. Detta understryks dessutom av moderna systemutvecklingsmiljöer, ex.vis 4 GL-system, och moderna typer av datorstöd, ex.vis kalkylprogram, där härledning ingår som naturliga delar i specifikationen av önskade operationer. Vem tänker på att lagra orderradssumma i en faktura? Å andra sidan kanske det inte är helt ovanligt att notera fakturabelopp i en modell, även om man vet att den är härledbar. Ackumulerad lön i ett lönesystem är ett annat exempel.

En anledning till att dessa tas med kan vara begreppets starka förankring i verksamheten eller dess vitala betydelse för verksamheten i något avseende. En annan orsak kan vara en intuitiv bedömning av att dessa värden är komplicerade att beräkna och dessutom tillräckligt ofta efterfrågade för att det ska vara mödan värt att lagra dem.

Vid tveksamhet om uppgiften är härledbar eller ej bör attributtypen tas med i modellen och så fort härledbarheten upptäcks detta noteras verbalt i anslutning till beskrivningen av attributet. Man ska däremot akta sig för att tynga ned modellen med härledbara uppgifter av mer tillfällig natur, där det är helt uppenbart att lagring inte blir aktuell, eller där man laborerar med i grunden samma typ av information men varierar förutsättningarna genom olika parameterkombinationer eller dylikt.



Ett extremalternativ är att överhuvudtaget inte notera härledbar information. Modellen blir "torftig". Dess motsats är en modell innehållande all efterfrågad information inklusive den härledbara. Den modellen blir "fullmatad" men sannolikt svår att överblicka. Tilläggsymboler e.dyl. till grundsymbolerna skulle kunna tala om att det är fråga om någonting härledbart. Däremot ryms ändå inte härledningsreglerna i grafisk notation. Dessa måste beskrivas på annan plats. En praktisk ansats är att gå en medelväg, se slutsatser nedan.

Några exempel som belyser svårigheten att på formella grunder klassa in informationen i en tillämning under korrekta begrepp (bör inte läsas av den som vill ha klara besked om hur varje problem ska lösas):

En användare efterfrågar "ackumulerad lön för viss person". Är detta något som ska beskrivas som ett informationsbehov i användargränssnittet eller som ett attribut till person i datamodellen eller både och? Om en användare frågar efter "lönebesked för viss person" så är det säkert ingen som ifrågasätter klassningen av detta som ett informationsbehov. Varför då inte som härledbart attribut? Beror det på att man tycker att lönebesked är en sammanställning av en mängd värden (bruttolön, skatt, nettolön, uttagen semester ....)? Får något som i modellen klassas som härledbart bara vara ett enkelt värde? Varför då i så fall? Adress klassar säkert alla som attribut till t.ex person. Vad händer om man får behov av att hantera gatuadress, postnummer och ort för sig. Försvinner adress helt, får det vara kvar som härledbart attribut eller övergår det till att bli informationsbehov (ex.vis genom att det används till adressetiketter)? Tänk om den som gillar adress påstår att det är gatuadress, postnr och ort som är härledbara från adress. Är det fel och varför? Självklart är det sammansatta sammansatt av beståndsdelarna, säger nog de flesta och alltså är det adress som är härledbart om inte ännu snarare ett informationsbehov. Hur många av dessa personer har inte i samma modell ett antal datum där säkerligen det i några fall finns intresse av att se både dag, månad och år separat, ex.vis för att översätta 881025 till "tisdagen den 25 oktober 1988"? Vad är härledbart till vad? Kanske datum egentligen är ett informationsbehov?

Jmf även diskussionen kring sammansatta domäner.

#### **Slutsats:**

**I grunden bör detta problem lösas med en god portion sunt förnuft.**

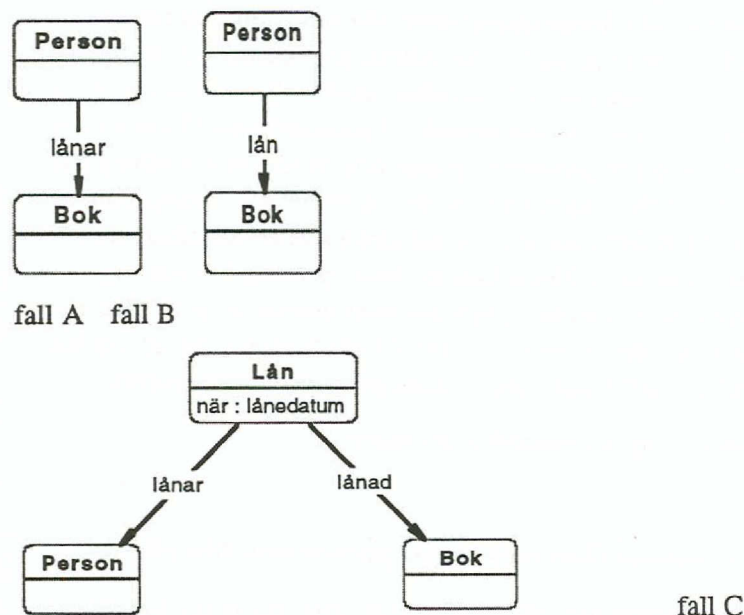
**Ta med härledbar information om den svarar mot centrala begrepp i verksamheten eller om den bedöms vara efterfrågad i flera olika sammanhang och inte helt naturligt löses inom ramen för respektive informationsbehov. Undvik sådana härledbara attribut som "tvingar fram" nya objekttyper, dvs objekttyper som annars inte skulle behövts i modellen.**



## M:M-sambandstyp eller objekttyp?

När bör ett M:M-attribut ersättas med ett objekt och två 1:M- eller M:1-attribut? När bör man gå den motsatta vägen?

Låt oss ta ett exempel ur ett lånesystem. Om vi bara vill ha information om vilka personer som lånar vilka böcker, kunde det räcka med ett samband mellan person och bok (fall A i figur 1). Även om vi vill uppleva sambandet som ett lån kan attributkopplingen kvarstå men med ändrat namn (fall B). Så snart vi vill hantera information om sambandet, dvs lånet, så måste sambandet med ett objekt som håller informationen ifråga (fall C).

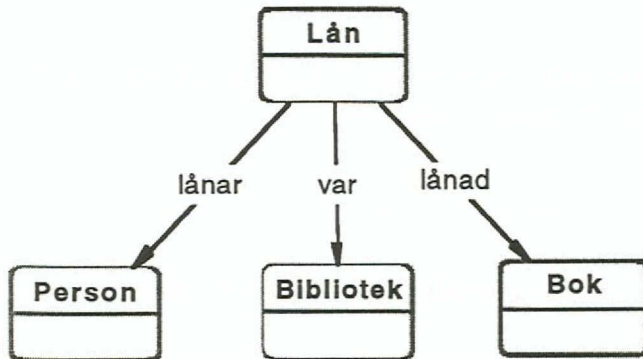


Figur 1

Skulle det kunna vara en idé att göra objektifieringen även utan det extra informationsbehovet? Ja, om LÅN står för ett centralt begrepp i applikationen och därför kan förmodas få "egenskaper" längre fram. Man är då förberedd och behöver varken göra omdefinitioner i modellen eller inom operationer som redan implementerats.

Antag nu att vi har objekttyper som inte har "egna" attribut andra än kopplingarna till de objekttyper som är del-identifierare. När kan dessa

ersättas med ett M:M-attribut? Jo, enligt samma resonemang som ovan, dvs med förutsättningen om tvåställighet. Om ett lån anses vara en bok lånad vid ett visst bibliotek av en viss person gäller treställighet. Lånet måste bli ett objekt även om inga övriga attribut anses behövas (figur 2). Obs, att kravet om tvåställighet gäller oavsett om sambandet är huvud- eller inversriktning, sett ur den aktuella objekttypens perspektiv.



Figur 2

Med god kännedom om den verksamhet som modelleras kan man nog i de flesta fall snabbt bedöma om sambandstypen behöver objektifieras eller ej. Vid osäkerhet är det bättre att skapa objekttypen och, om det senare inte visar sig behövas, då ersätta den med en sambandstyp.

## Olika modelleringsansatser

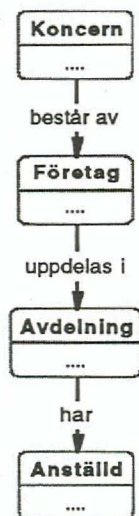
Låt oss ta ett exempel:

a. En koncern består av ett antal företag som vart och ett är uppdelat i ett antal avdelningar i vilka arbetar ett antal personer.

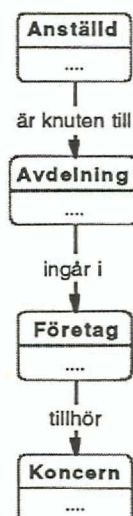
Uttryckt på ett annat sätt:

b. Varje person är knuten till en viss avdelning som ingår i ett företag som tillhör en koncern.

De båda synsätten ger sannolikt upphov till olika datamodeller. Se figur 1 och figur 2. Alternativ 1 redovisar ett hierarkiskt synsätt, där varje objekt förekomst pekar på ett antal andra objekt förekomster. Exvis kan koncernen "k1" peka på var och en av företagen "f1", "f2" och "f3"; "f2" kan ha avdelningarna "a1" och "a2" osv. Man upplever att det man modellerar är 1:M samband i huvudriktningen, inte sällan därmed från enklare till mer sammansatta eller abstrakta objekttyper. Avdelning "a1" är sannolikt inte vilken "a1" som helst utan "a1" som ingår i f1 som ingår i k1" till skillnad från ex.vis "a1, f2, k1". Observera att den här ansatsen inte nödvändigtvis leder fram till rent hierarkiska strukturer (se exv figur 3). Beteckningen används för att ange att modellen skapats utifrån en hierarkisk syn på den bakomliggande verkligheten.



Figur 1



Figur 2



Alternativ 2 angriper istället modelleringsproblemet "inifrån och ut". Vilka egenskaper (attribut) har eller beskriver en anställd? Förutom diverse övriga attribut framgår att en anställd hör till en viss avdelning. Observera att "är knuten till" bara pekar ut en enda avdelning. Man har här ett funktionellt synsätt, arbetar med 1:1- och M:1-samband. Ibland brukar man säga att det hierarkiska alternativet står för ett makro-perspektiv medan det funktionella står för ett mikro-perspektiv.

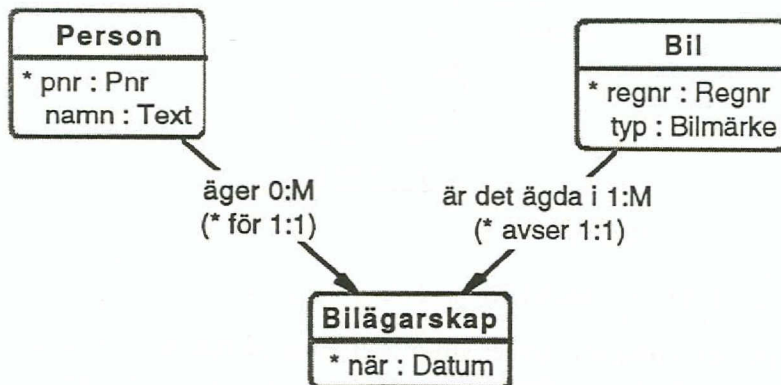
Vilken ansats kan vara att föredra? Den kritiske invänder genast att det inte spelar någon roll eftersom en god modell alltid ska ha benämningar på sambandens båda riktningar. Vi anser dock att rittekniken med pilar fokuserar mer på huvudriktningen än på inverteringen. Läsaren av modellen upplever den olika beroende på vad som placeras som huvudriktning. Dessutom är valet i första hand fråga om en ansats, ett sätt att angripa modelleringsproblemet.

Ytterligare ett exempel belyser skillnaderna.

Antag ett bil exempel med personer, bilar och ägarskap. En person har personnummer och namn plus några ytterligare attribut som vi utelämnar här. På samma sätt ser vi bara till bilens registreringsnummer och typ. "Ägarskap" står för en persons ägarförhållande till en viss bil. Om två personer äger samma bil har vi alltså två ägarskap. Samma sak gäller om en viss person äger två bilar. Av intresse är att hålla reda på när en viss person blev (del)ägare till viss bil.

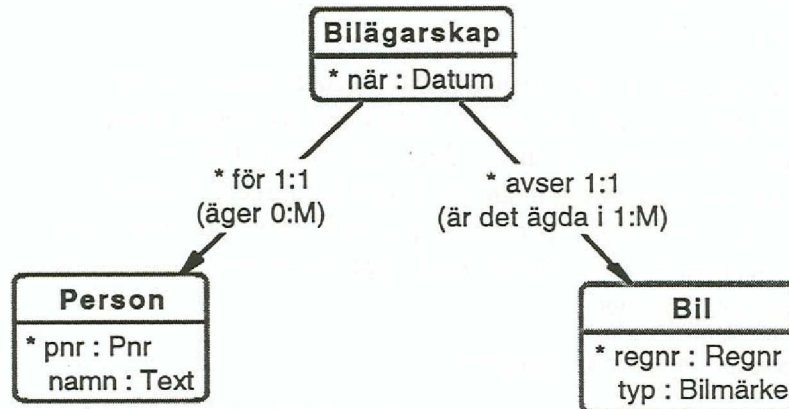
Med en hierarkisk ansats ser vi verkligheten så att en person äger ett antal bilar eller mer precist, har ett antal bilägarskap. På samma sätt utifrån bilens perspektiv ägs den av ett antal personer eller mer precist ingår i ett antal bilägarskap. Därutöver har varje objekttyp två attribut. Se figur 3.

Observera att modellen inte totalt sett är hierarkisk. Både person och bil "pekar ju på" ägarskap. Beteckningen står endast för hur man upplever sambanden på förekomstnivå utgående från viss startposition (viss objekttyp). Sett ur en viss persons synvinkel kan den alltså ha ett antal ägarskap. Samma gäller för bil.



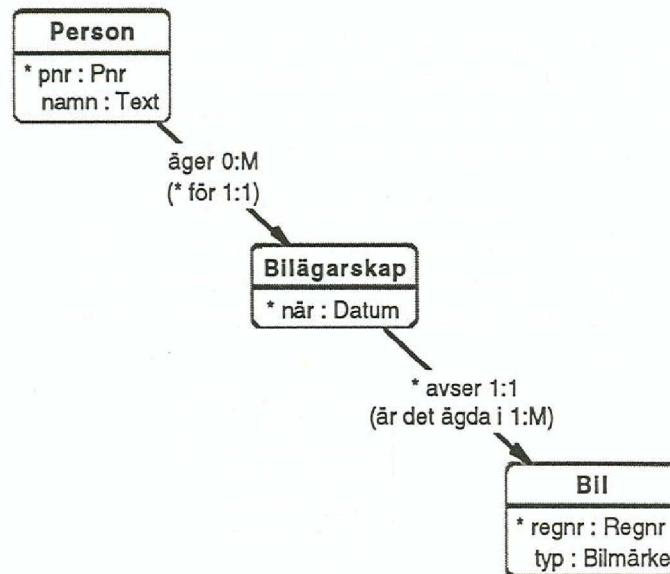
Figur 3

Med en funktionell ansats försöker vi uttrycka vad som beskriver ett visst objekt. Person beskrivs genom personnummer och bil genom registreringsnummer. Ägarskap beskrivs genom vem som är ägare, vilken bil som ägs samt när ägarskapet startade.



Figur 4

I en blandad ansats tar man inte hänsyn till någon enskild grundprincip. Det aktuella resonemanget får avgöra. Vi kanske ser att personer kan äga fler bilar, ha flera ägarskap. Ägarskap i sin tur "pekar ut" både vilken bil det är och när personen blev ägare till den bilen.



Figur 5

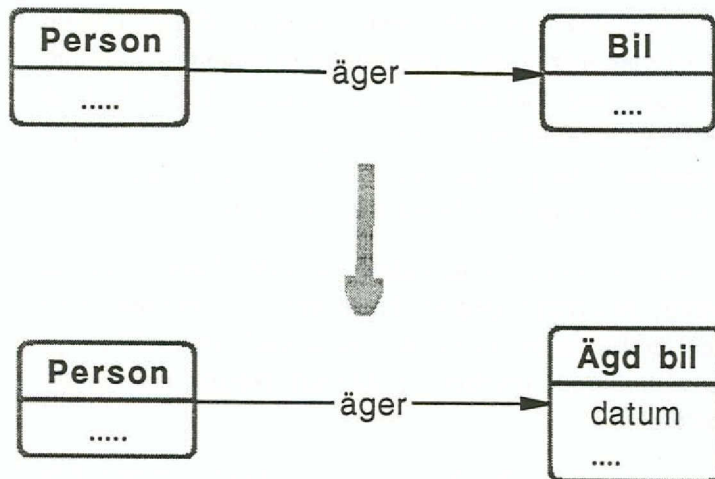
Några synpunkter på för- och nackdelar med de olika ansatserna:

- \* Det första och viktigaste kravet vid modellering på den här nivån är att modellen kan förstås av samtliga inblandade, dvs att den verkar vettig. Annars blir det lätt missförstånd.



\* Den hierarkiska ansatsen återspeglar sannolikt mer direkt hur data kommer att presenteras från eller uppdateras i det färdiga systemet. Ex.vis är frågan "Presentera för en viss person alla bilar personen äger samt datum för ägarbyte" inte speciellt långsökt.

\* Den funktionella ansatsen fokuserar mer på den exakta innebörden av ett objekt eftersom man per objekt koncentrerar sig på att åstadkomma en så fullödig beskrivning av den som möjligt. Det är sannolikt att den "hierarkiske" i en första ansats kommer fram till en modell liknande den i figur 6. Ägarskapet ligger i sambandet. När sedan ägartidpunkt dyker upp glider bil över till att bli "ägd bil" (ägarskap) dit således ägartidpunkten kopplas. Övriga egenskaper hänger kvar. Vi har så omärkligt hamnat i en modell, som felaktigt avbildar både bil och bilägarskap i samma objekt.



Figur 6

Eftersom man här i första hand tittar på förekomstsamband utifrån ett givet objekt, löper man risken att uppleva M:M-samband som 1:M. Den "personfixerade" tycker sig se ett 1:M-samband både före och efter introduktionen av ägartidpunkt. En (1) person kan ju äga många (M) bilar liksom ha många (M) bilägarskap. Alltså ingen anledning till ändring av mapping. Att i själva verket ägd bil nu svarar mot en persons (del)ägarskap i en bil (det är ju inte säkert att varje del-ägare till en bil blir del-ägare samtidigt) noteras inte. Sambandet har "omärkligt" gått över från ett M:M-samband till ett 1:M-samband.

Hade detta upptäckts, hade det även automatiskt indikerat att betydelsen av bil förändrats, att en översyn var på sin plats.

Motsvarande misstag kan ju även "funktionalisten" göra men genom att denne oftare ställer sig frågan "vad är det egentligen som beskriver det här objektet" ökar sannolikheten att misstagen upptäcks eller undviks.

\* Både den funktionella och den hierarkiska ansatsen måste tillåta sambandstyper, som är naturligt M:M. Om exempelvis ägartidpunkt inte skulle vara av intresse kan även objekttypen ägarskap utgå och ersättas av



sambandstypen äger, om man inte av rent förståelsemässiga skäl anser att objekttypen bör vara kvar.

\* Funktionell ansats kan vara stelbent för attributtyper. Antag att personen har flera telefonnummer. Inte skulle det kännas rimligt att behöva skapa ett objekt telefonnummerabonnent (om inte tillämpningen kretsar kring telefonabonnemang, förstås).

\* I den funktionella ansatsen sker all beskrivning, precisering av objekttyper i sambandstypsriktningen. Det är därmed lätt att läsa sådana modeller.

\* Viktigare är dock att de attributtyper, som tillsammans utgör identifieringen av en objekttyp återfinns bland beskrivningsattributen, dvs de "utgående pilarna". Att asterisk-märka en inkommande sambandstyp kan ge tolkningsproblem eller åtminstone förvilla. När en sambandstyp och även dess invers ingår som (del)identifierare av vart och ett av de två sammanbundna objekttyperna finns ju inget val. Dock kan man utgå ifrån att det i så fall är fråga om någon typ av sällsynt 1:1-förhållande.

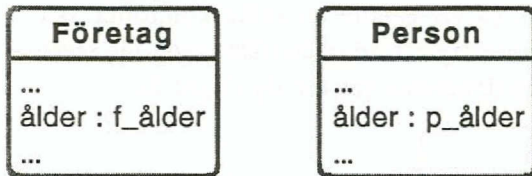
\* Relationsmodellens 3:e normalform, vilken normalt eftersträvas vid logisk databasutformning, uttrycker endast funktionella samband. Om datamodellen uttrycker motvarande typer av samband underlättas ju översättning. Att systemerare och realiserare därmed lättare kan förstå varandra är ju ingen nackdel.

\* Vid små tillämpningar eller i början av ett modelleringsarbete kan man tillåta sig att skissa utan alltför mycket hänsyn till regler men annars bör man sträva efter att arbeta efter överenskommen modelleringsprincip.

**Avgörande ska vara det som känns naturligt och rätt för modelleringsgruppen. Finns inga påtagliga preferenser rekommenderar vi att man strävar efter att använda den funktionella ansatsen men med tillåtelse att dels ha flervärdiga attributtyper, dels sambandstyper av typen M:M.**

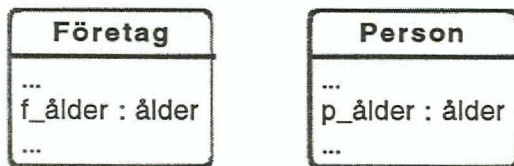
## Många eller få domäner?

Varje attributtyp tillhör entydigt sin objekttyp. På samma sätt skulle till varje attributtyp kunna kopplas en unik domän. Det är skillnad på Ålder (för person) och Ålder (för företag). Kanske gäller olika villkor för dem (personer blir 1-120 år medan företag testas för >0). De blir sannolikt domänerna p\_ålder resp. f\_ålder.

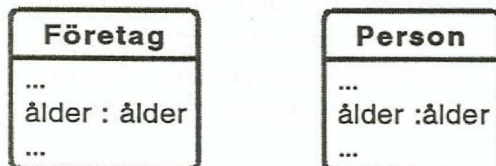


Figur 1

Kanske räcker det att uppfatta Ålder på en högre generaliseringsnivå, exv endast med kravet att det ska ha ett värde större än noll. Hur attributtypen benämns är objekttypens "ensak".

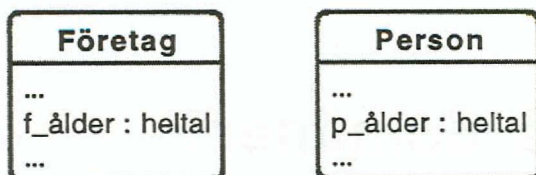


eller



Figur 2

En extrem variant är att helt avstå från applikationsorienterade domäner och endast referera till ett fördefinierat antal generella domäner, ex.vis Heltal, Heltal\_5, Text\_30, Pnr, Datum.

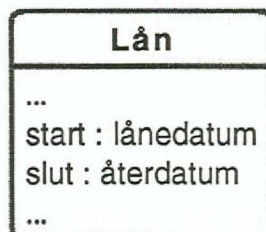


Figur 3

Vid större modelleringar bör man sträva efter att etablera välgenomtänkta och för det aktuella behovet specifika domäner. Jämför med låne- och återdatum. Under ett skisskede kan datum mycket väl anses tillfyllest (om någon domän överhuvudtaget). Se figur 4. Så snart modellen börjar preciseras, bör precisionen även avse domänerna. Vill man i något avseende separatbehandla låne- och återdatum, ex.vis genom att olika kontroller ska utföras på dem (lån får bara ske på måndagar medan återlämnande får ske i boklådan varje dag), delas de upp och definieras var för sig (figur 5).



Figur 4



Figur 5

För- och nackdelar:

- En förfinad uppdelning innebär många namn att hitta på på domännivå. Ofta blir det samma namn som för attributet (eftersom det då sannolikt bara är en attributtyp som refererar till domänen). Risk för sammanblandning, (talar man om attributtypen eller domänen?) för den som är nybörjare. För den vane och i större projekt med stort schema är det snarare en fördel att veta att domän och attribut har samma namn (förutom ev skillnader i versaler och gemena). Det kan t.ex. underlätta underhåll och sökning i dokumentationen.

- Viktigare är att ju mer uppdelat desto mer exakta kontrollprocedurer kan kopplas till domänen, eller, om man så vill, en procedur per "roll".

Har man å andra sidan för grova kontroller kopplade till schemat och/eller för få domäner, kompletterar man säkert med andra kontrollprocedurer i andra delar av en tillämpning, exvis direkt i formulär- eller rapportdefinitionerna. Observera, att sådan komplettering därmed

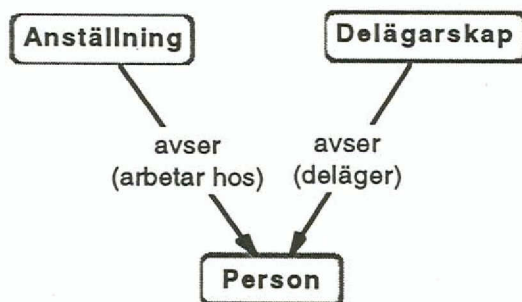


måste göras på varje plats där domänen refereras, vilket rimligtvis måste ses som en nödlösning när tillgänglig domänhantering är otillräcklig.

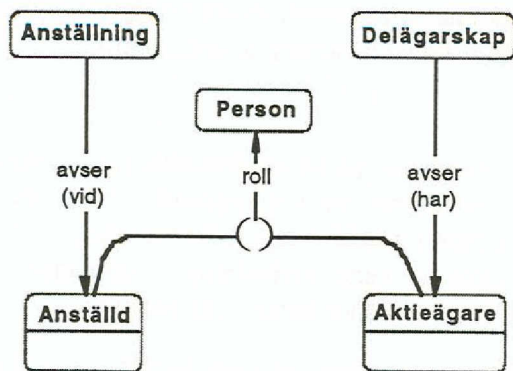
- Vid få domäner måste man tänka på att en förändring av syntax för representationen, det kan vara editering, ny uppbyggnad - exvis ett nytt prefix, bokstavskombination i stället för siffror mm - kräver eftertanke. Den nya syntaxen kanske inte ska gälla för varje användning av domänen? Mao kan i det läget en uppdelning i flera domäner komma att behövas.

- Det är principiellt inget som hindrar att man lägger kontroller på attributtyperna. Detta minskar behovet av unikheter på domännivå. Kan då inte alla kontroller ligga på attributtypsnivå? Nej, praktiskt sett inte, eftersom varje attributtyp är objekttypsunik och därigenom samma kontroll kan få lov att upprepas för varje objekttyp trots att kontrollen är av mer generell natur. Ett antal allmänt definierade domäner löser problemet. Obs, att referens till domän bara är referens till värde och syntax (lexikal företeelse). Om flera samband refererar till samma domäner finns risk för att, vid ändring av domänspecifikationen, detta kan passa vissa attributtyper men skapa inkompatibilitet för andra. Ett problem som kan undvikas om specifikationen ligger på attributtypen eller på alltid unika domäner.

- Jämför hur man normalt modellerar objektsamband, figur 6. Man skapar inte en ny objekttyp för varje roll, ofta bara om rollen i sig innehåller specifika beskrivningar. Vill man i dessa lägen applicera kontroller finns bara sambanden att tillgå. En anställd måste vara minst 18 år medan man får vara aktieägare redan som nyfödd. Inte skapar man figur 7 bara för detta?



Figur 6



Figur 7

Rekommendation:

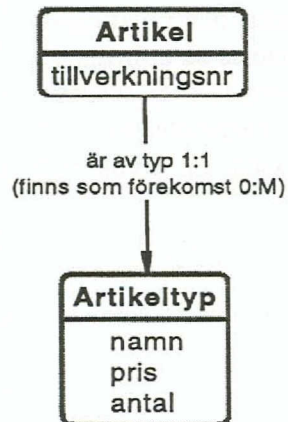
I mindre projekt, under experimentella faser eller i tillämpningar med överblickbar datastruktur, kan man tjäna tid och enkelhet vid specifikationer och ändringar genom att laborera med få domäner. I större sammanhang är tiden att specificera resp att underhålla en datamodell med många attributtyper försumbar i förhållande till övrig projekttid. Det viktiga är att skapa klarhet, finna begrepp och samband som fungerar bland användarna. Om det sedan är genom många eller få domäner kan vara en smaksak.

Varför inte sträva efter att finna en uppsättning generella domäner som kan ingå i företagets basuppsättning lexikala begrepp (domäner) och så stringent som möjligt använda dessa. Varje ny domän måste definieras så utförligt att felanvändning undviks. Därutöver måste framgå dels vem som är ansvarig för domänen, dels med ansvaret vidhängande förpliktelser och befogenheter. Jämför med den övergripande processen att finna och underhålla en verksamhets centrala objekttyper.



## Typ/förekomst enligt ovan contra mängd

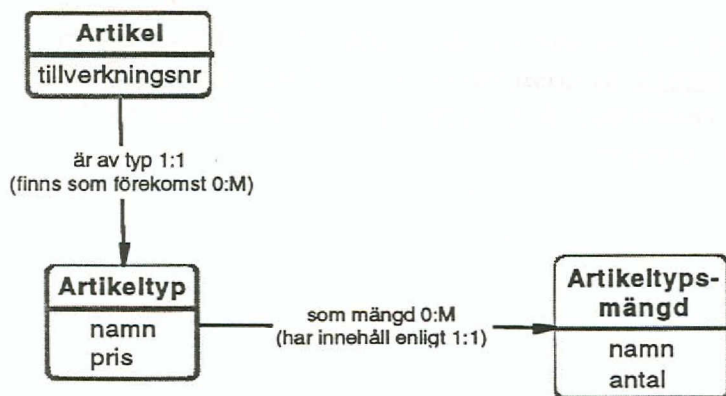
Artikeltyp står för en generell beskrivning av vad som gäller varje artikel. Varje artikel som pekar på en viss typ har det pris och det namn som finns beskrivet för typen.



Figur 1

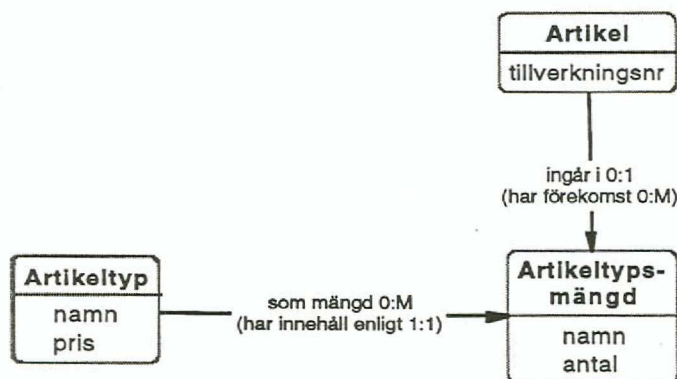
Ett parallellt fall är förhållandet mellan data och datamodell (förekomst och typ). Om vi påstår att (varje) anställd har en ålder och en lön betyder det att varje enskild anställd har en specifik ålder och specifik lön. Säger vi däremot att de anställda har en total lön och att de uppgår till ett visst antal är detta en beskrivning av mängden anställda, inte något som kan hänföras till varje enskild anställd. Båda aspekterna kan behöva finnas företrädda i en och samma datamodell.

Tillbaka till vårt artikel-exempel. Att *pris* och *antal* gäller som typbeskrivning har vi redan konstaterat. Men *antal* är ju en klar mängdegenskap. Obs, skillnaden i namn "cykel" som typnamn och "cyklar" som mängdnamn. En mer preciserad modell kan då se ut enligt figur 2.



Figur 2

Beroende på tillämpningens informationsbehov kan man alternativt formulera sambandstyperna enligt figur 3, eller ha med alla tre sambandstyperna. Dock måste man hela tiden ha i minnet att mängdegenskaper inte sällan är härledbara (som fallet är i exemplet) och att, om alla egenskaper är härledbara, mängdobjektet kan utgå helt (om man så önskar).



Figur 3

Att antalsuppgiften utan större bekymmer hamnade under typ-objekttypen beror sannolikt på att typ- och mängdobjekten står i ett 1:1-förhållande till varandra om man med en viss Artikeltypsmängd menar en generell mängd där samtliga förekomster av viss typ ingår. Ingen större skada har inträffat.

Står en viss Artikeltypsmängd för endast en delmängd av förekomsterna (sambandet med artikeltypen beskriver en restriktion) eller kan det finnas flera mängder mot samma typ måste mängd-objekttypen finnas i modellen.

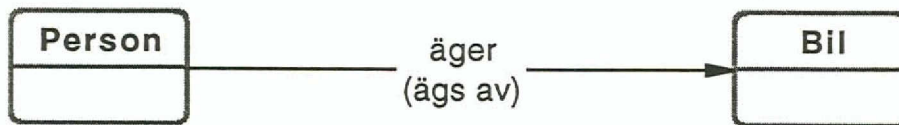
Objekt som i vissa avseenden beskrivs på samma sätt kan upplevas vara av samma typ. Det finns ibland behov av att kunna särskilja sådan typinformation från förekomstinformation i datamodellen. Man refererar i dessa fall från förekomster bla till en, för ett antal förekomster, gemensam beskrivning (typ-objekt). Det som finns i den gemensamma delen gäller var

och en av de förekomster som refererar dit. I andra fall vill man istället kunna beskriva en mängd av förekomster. De attribut som anges för mängden gäller som beskrivning av mängden och kan aldrig hänföras till något enskilt element i mängden.



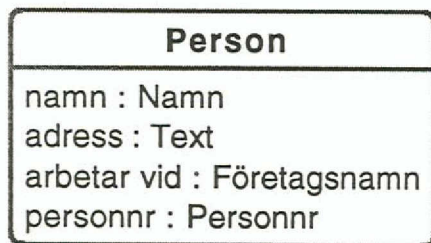
# Namngivningsprinciper

Sambandstypsnamn tenderar bli verb. På så vis blir det naturligt att formulera kompletta utsagor med hjälp av de båda objekttyperna som subjekt respektive ackusativobjekt och sambandstypsnamnet som predikat. Exv kan ur figur 1 utläsas att "Person äger Bil" med det underförstådda behovet av att kunna uttrycka att "En viss person äger en viss bil".



Figur 1

Attributtyper blir ibland också verb, exv 'heter' eller "har adress". I andra lägen blir det enklare att använda substantiv för att markera en klassificering. Istället för "har adress" är det antagligen lika informativt att bara ange "adress". Ofta får attributtypen samma namn som den domänen definieras genom, exv "personnr" för Person. Se några olika varianter i figur 2.



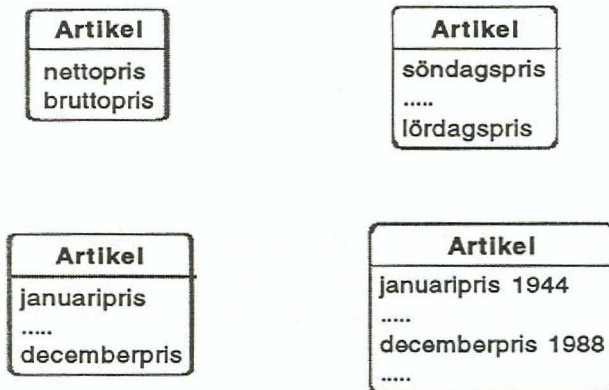
Figur 2

Vissa ansatser reglerar även användning av versaler och gemena. Exv kan stipuleras att objekttypsnamnet ska börja med versal med resterande tecken som gemena, attribut- och sambandstyper endast vara gemena och domäner enbart vara versaler. Frågan är om dylika detaljregler bidrar till klarhet och stringens eller snarare blir en syntaktisk börda.

Vilka principer som väljs är av mindre vikt så länge som terminologin är förankrad hos användarna. Av praktiska skäl bör man inte välja för långa namn. Alltför långt gående förkortningar, som inte finns etablerade i tillämpningsmiljön, ska också undvikas. 5-15 tecken borde i normalfallet räcka.

## Närbesläktade attributtyper eller "typifierad" objekttyp?

Antag att vi har en artikel som beskrivs med Brutto- och Netto-pris. Det är då sannolikt att man gör två attribut av dessa. Men om man har ett pris som varierar med veckans dagar? Gör man 7 attribut då? Förmodligen. Var och en av 12 månader då? Antagligen även då. Men om man vill hålla reda på de priser som gällt under varje månad under varje år? Dels så har vi nu många fler saker att beskriva, dels så blir det nya allteftersom tiden går.

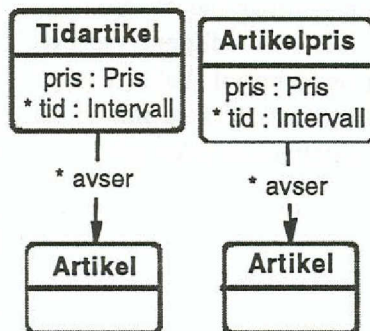


Figur 1

I det läget är det dags att gå ifrån attributalternativet och istället "objektifiera". Vi skapar en objekttyp som pekar på vilken artikel som är aktuell, på aktuell tidpunkt eller tidsintervall och vilket pris som gäller då. Priset är beroende av (en funktion av) artikel och tid. I datamodellen kommer därför objekttypen att identifieras med vilken artikel det är frågan om och vilket tidsintervall. Istället för intervall kan givetvis en tidsutdräkt även uttryckas genom månad, vecka eller annan lämplig konvention.

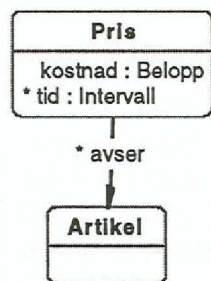
Hur namnges då lämpligen denna nya objekttyp, vad står den för? Ibland kan objekttypen ta ett namn som redan finns i verksamheten, varvid problemet inte uppstår. I annat fall gäller det att hitta på ett namn som är tillräckligt distinkt och samtidigt beskrivande. "Artikels pris vid viss tid" kan tänkas men verkar långt. "Pris vid viss tid" är kortare och med samma innebörd om man kan ha "artikel" som underförstådd. Mer kärnfullt med samma innebörd är "Intervallpris". Inte sällan gör man en kombination av namnen på de samband och/eller egenskaper som tillsammans utgör identifieringen. I exemplet är det ju artikel som, tillsammans med tiden, pekar ut ett pris. "Artikeltid" eller Tidartikel" är då alternativen. Här måste

man dock noga vaka på att ordningsföljden dels ger ett rimligt beskrivande begrepp dels inte ger en felaktig semantisk innebörd. Exempelvis är inte "Taktegel" detsamma som "Tegeltak"! I vårt exempel kan man uppleva att de tre beskrivningselementen inte väger jämt. Sannolikt ligger synfältet på objekttypen utifrån artikel-perspektivet (via den enda sambandstypen), man vill veta priset för artikeln. Orsaken till hela problemet är att man har att ta hänsyn till att priset varierar över tiden. Men det viktiga är ändå fokuseringen på artikelns pris. Beteckningen "Artikelpris" ligger då nära till hands. Även det neutrala "Tidartikel" ligger nära till hands.



Figur 2a      Figur 2b

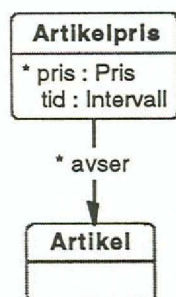
Nå, varför inte rätt och slätt "Pris" enligt figur 3? Det är ingen skillnad i sak mot ovan, endast i namngivningen av objekttypen och en attributtyp. I en större modell är sannolikt inte "Pris" tillräckligt informativt. Det kan ju finnas andra saker än artiklar som har pris.



Figur 3

Ett annat synsätt på pris är att upphöja detta till att, tillsammans med artikel, peka ut de tider eller tidsintervaller under vilket priset gäller/har gällt. Mot en artikel och ett pris kan då svara ett antal tider.

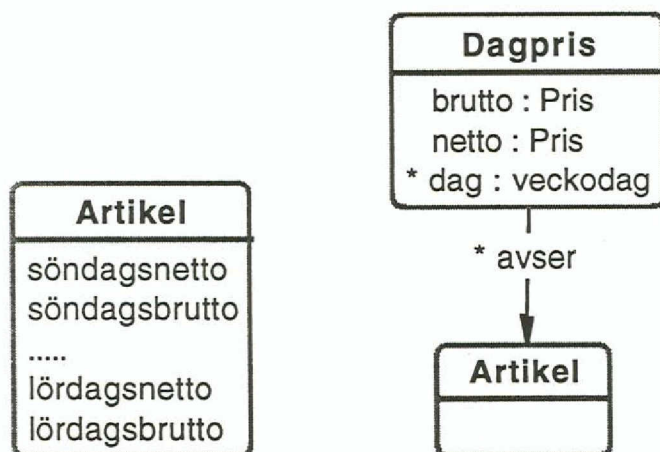




Figur 4

Att figur 4 inte känns lika bra kan bero på att priser är individuella medan intervaller antagligen är artikeloberoende och därmed mer naturliga att använda vid referenser till objekten ifråga. I alla händelser måste vi vara vaksamma på att Artikelpris i figur 4 inte är samma Artikelpris som i figur 2b.

Obs, även vid ett fåtal närbesläktade attributtyper blir det snabbt aktuellt att objektifiera om det finns grupperingar av dem. Antag att vi för var och en av veckans dagar vill spara både Brutto- och Netto-priser. Om vi inte objektifierar får vi ju Söndagsbrutto, Söndagsnetto, .... (figur 5a) i stället för DAGSPRIS-objekttypen med egenskaperna Brutto och Netto (figur 5b).



Figur 5a

Figur 5b

Så snart man har en situation där antalet attribut inte är fast, dvs de ändras som en funktion av tiden, som en följd av inträffade händelser etc, bör objektifiering genomföras. Även vid fast antal attribut kan det vara lämpligt att objektifiera, speciellt om man inom ett informationsbehov vill variera mellan attributtyperna som ett slags parametrar, ex.vis ena gången efterfråga "söndagsinfo", nästa gång "onsdagsinfo", o.s.v., men i båda fallen med avsikten att få presentation av samma typ av uppgifter.

En lämplig strävan bör vara efter ett synsätt på den skapade objekttypen, som gör att sambandstyperna blir funktionella, dvs givet en viss förekomst av objekttypen pekar den ut maximalt ett värde/objekt per samband. Exemplet i figur 4 bör av den anledningen undvikas.

Namngivningen överenskommes mellan berörda modellörer. Finns inget givet bra namn från verksamheten brukar en kombination av namnen på de identifierande sambanden duga. Låter det inte bra bör man ta till en beskrivande ordsekvens typ "Artikels pris vid viss tid".

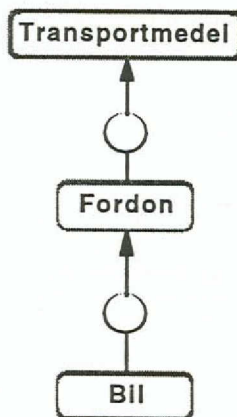
# Slå ihop eller dela upp objekttyper?

När kan det vara lämpligt att dela upp en objekttyp i flera, alt. slå ihop flera till en objekttyp?

En absolut förutsättning är att det råder ett 1:1-förhållande mellan dem.

Påverkande faktorer för övrigt är:

- Naturlig generaliseringsstruktur kan urskiljas, se exempel, figur 1 nedan.
- Attributbeskrivningen är dåligt/väl överlappande.
- Olika/lika användning, ex.vis i olika formulär eller rapporter.
- Beteckningarna (namnen) används (ej) i applikationsmiljön.
- (O)likhet i identifieringen.
- Dåligt/väl överlappande objektmängder.
- Resp. objekttyp förekommer (ej) i delvis/helt separata delar av modellen, dvs andra objekttypers sambandstyper "refererar (ej) till" objekttyperna.



Figur 1

Vi ska diskutera några tänkbara situationer:

## a. Person - pensionär - telefonabonnent

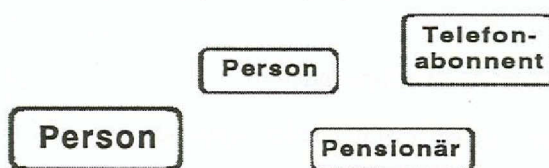
Ska pensionärsinformation ligga under Person eller under separat Pensionär-objekttyp? Det beror på hur separat de refereras till och hur uppdelade attributtyps-, sambandstypsbeskrivningarna är. Sannolikt i stort



överlappande "kringstruktur", varför Person borde räcka. Har de inte speciellt överlappande attributtyper och/eller sambandstyper bör en generaliseringsstruktur övervägas. Eventuellt måste då objekttypen Icke-pensionär tillföras modellen.

Ok, telefonnummer då? Ska man skilja upp de personer som har telefonnummer under Telefonabbonnent? Nej, sannolikt gäller både överlappande "kringstruktur" (sambandstyper) och attributtyper med Person.

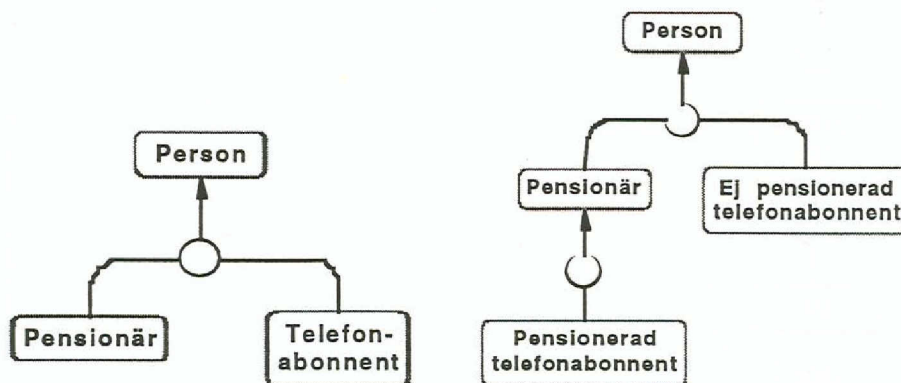
Den enklaste lösningen är figur 2 med endast en Person-objekttyp. Den andra extremen är figur 3 med en uppdelning i tre fristående objekttyper. Förutsättningen för uppdelning är minimal överlappning i attributtyper, "kringstruktur" och mängder.



Figur 2

Figur 3

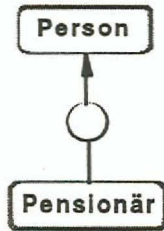
En kompromiss är att arbeta med generaliseringssamband enligt figur 4. Men där gäller det istället att vara vaksam på att strukturen blir korrekt. Figur 4 är inte fel om den beskriver de erforderliga arvsmekanismerna kring subobjekttyper, dvs att telefonabbonnenter ärver person-information. Det som inte finns redovisat är att en del telefonabbonnenter är pensionärer och andra inte. Är det så att de telefonabbonnenter som är pensionärer ska hanteras lite annorlunda än andra, exvis genom att det är intressant att ärva viss pensionärsinformation, bör substrukturen göras om enligt figur 5.



Figur 4

Figur 5

Eftersom detta i det aktuella fallet inte verkar sannolikt och inte heller att telefonabbonnent behövs som separat objekt kan en lämplig kompromiss i slutändan ex.vis bli enligt figur 6.



Figur 6

Utän kunskap om applikationens behov och med synnerligen begränsad kunskap om förutsättningarna, verkar alternativen 2 och 6 ligga närmast till hands.

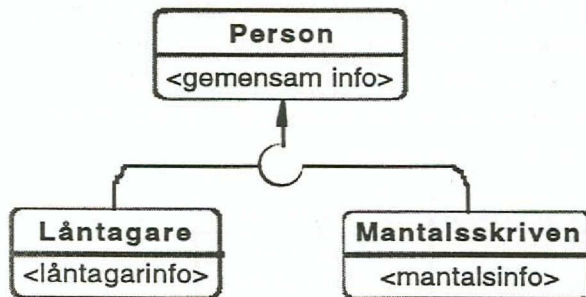
#### b. Person - låntagare - bibliotekarie

Vi går över till en bibliotekssituation. Antag att både låntagare och bibliotekarie finns. De har sannolikt både olika "kringstruktur" och informationsbehov. Dessutom är det en mycket liten överensstämmelse i mängderna (3 st. bibliotekarier mot 10000 låntagare). Figur 7.



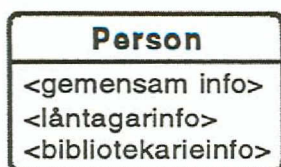
Figur 7

Finns "gemensam nämnare" eller generellare begrepp, som båda är specialiseringar av och där mängderna är rimligt överlappande, kan det vara lämpligt att upprätta exvis Person, och föra över gemensam beskrivning (dvs ren Person-info) dit. Figur 8.



Figur 8

Varför inte ersätta båda med enbart ett Person-objekt? Mantalsskriven- resp låntagareinformationen är ju ändå separerbar genom resp attributnamn. Resp. roll framgår ju dessutom genom attributnamnet från Lån.



Figur 9

Motargumenten är snarlika de som framförts beträffande för få attributtyper, dvs man kan inte enkelt applicera kontroller per objekttyp, begreppen kanske naturligt finns hos användarna, man kanske längre fram väljer olika identifieringsprinciper, man ska eftersträva att lokalisera attributtyperna så precist som möjligt mm.

Slutsatsen blir att med skissade förutsättningar figur 7 och med tveksamhet figur 8 är användbara.

**c. Olika principer för identifiering av objekten.**

En ytterligare anledning till uppdelning i flera objekttyper kan vara varierande identifiering. Antagligen är detta tecken på att generaliseringssamband inte kan upprättas. Åtminstone bör förutsättningarna noga analyseras. Låntagare kanske kan vara även sådana som arbetar men inte bor i kommunen. Där kan trots allt finnas ett delmängdssamband eller en delvis överlappning av intresse. Behöver man kunna relatera en företeelse under den ena objekttypen till motsvarande under den andra, måste samband upprättas i modellen. Antingen beskrivs det genom vanliga sambandstyper eller genom ett nyskapat generaliseringssamband. Den "gemensamma" mängden, som man ju vill ha relateringen inom, kan brytas ut, ex.vis som i alternativ b ovan. Se figur 10a. Finns på så vis ingen gemensam information kan sambandet i stället klaras genom en normal sambandstyp, se figur 10b nedan.



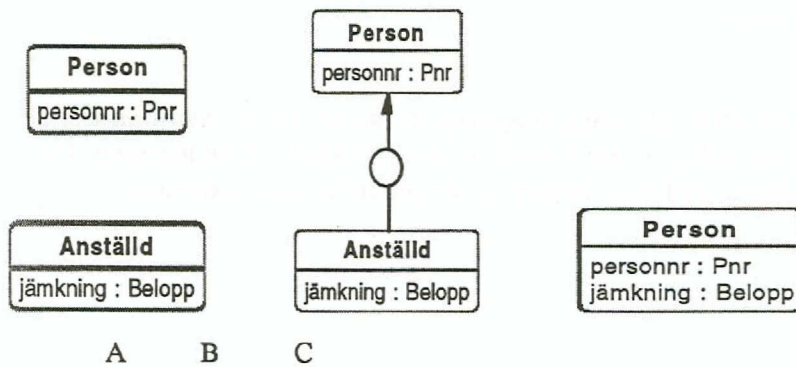
Figur 10 a      Figur 10 b



Lösning beroende på förutsättningar. Se diskussionen, ovan.

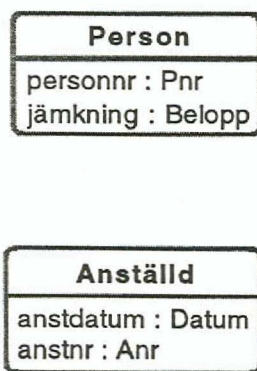
d. Vaksamhet på 1:1-villkoret.

Det som vid första påseende verkar vara ett 1:1-förhållande kanske inte alls är det. Ta fall A i figur 11 nedan. En person kan ha rollen som anställd. Som sådan finns ett anställd-objekt. Personer som är anställda har eller agerar i två roller, som *person* och som *anställd*. Det föreligger här ett 1:1-samband. Om man så finner lämpligt kan både B och C utgöra lämpliga alternativ.



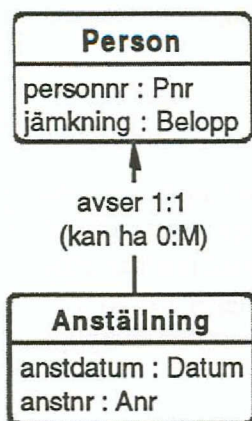
Figur 11

*Anställd* i figur 12 står för en anställningssituation. Om personen har två anställningar (med olika anställningsdatum) finns det för denne tre förekomster av objekt, ett person-objekt och två anställd-objekt. Personen har fortfarande en roll som anställd men befinner sig i den rollen i två situationer. Eftersom figur 12 inte beskriver rollen utan situationen, finns beskriven råder alltså ett 1:M-samband och inte ett 1:1-samband. Att här försöka sig på hopslagning av objekttyperna eller upprättande av generaliserings samband är direkt fel! Jämför med diskussionen vid figur 4.



Figur 12

Problemet skulle nog aldrig uppstå om man valt ett namn som inte direkt associerar till en person, ex.vis *anställning*, se figur 13.



Figur 13

Man bör sträva efter att ange mapping så snart en ny sambandstyp introduceras i modellen. Detta kräver en analys och klok bedömning. En följd blir att misstag undviks. Dessutom blir modellen mer informativ.

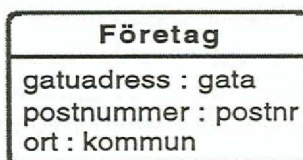
# Sammanfattad domän eller objekttyp?

Det kan ibland vara svårt att bedöma om ett antal värden, som hör ihop, ska kunna få existera som värden i någon typ av sammansatt domän eller om sammanhörigheten måste manifesteras i form av ett "tillhörighets"-objekt. Givetvis är detta en i grunden praktisk frågeställning som behöver ett praktiskt svar. Ta exv begreppet adress och låt oss bortse ifrån att det som företeelse antagligen är mycket komplext.

Adress kan åtminstone ses ur fyra synvinklar, se figurerna 1 - 4.



Figur 1



Figur 2



Figur 3



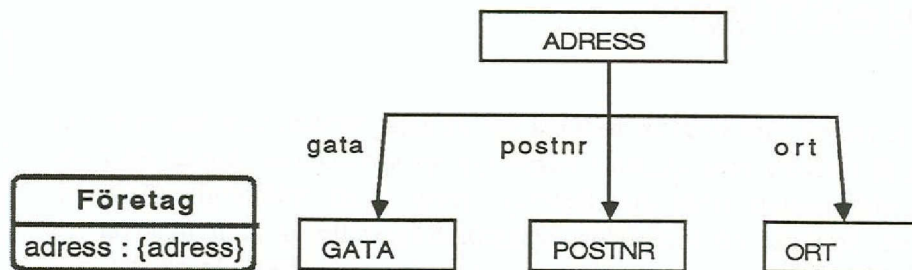
Figur 4

Hur man väljer att modellera blir som vanligt beroende på synsätt och behov. Det enda alternativet som verkar lite farligt är figur 3 där adress är härledbart. Utan klara konsistensmekanismer kan man hamna i en situation där adressen inte stämmer överens med de andra tre komponenterna. Lösningen är tekniskt prestandaanpassad, vilket inte borde vara fallet på den konceptuella nivån. Å andra sidan kanske alla fyra begreppen gata, postnr, ort och adress är använda begrepp i verksamheten. Adress kan i så fall definieras som härledbart ur de andra komponenterna.

Gata, postnr och ort "hänger" i dessa alternativ direkt under Företag, som har adressen ifråga. Borde egenskaperna inte istället samlas under en separat Adress-objekttyp? Se figur 6. Innan vi kommer dit kan det vara värt att notera att vissa modelleringsansatser tillåter sammansatta domäner, vilket ger



ett lösningsalternativ "mitten mellan" de två övriga, se 5. Definition och hantering av begreppet "sammansatt domän" verkar ännu något oklar.



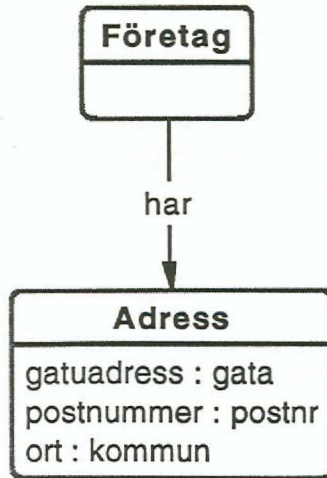
Figur 5a

Figur 5b

Modellering av en företeelse som sammansatt domän respektive som objekttyp innebär konsekvenser. Ett domänvärde uppdateras aldrig i sig, ej heller ett aggregerat värde, bara ett värde kopplat till ett visst objekt via visst attribut. Uppdatering av attributvärde innebär alltså byte av domänvärde. "Bytet" är endast lokalt till visst objekt. Ex.vis, om två personobjekt pekar på "Odengatan 12" och den enes ändras till "Odengatan 13", så bor de i fortsättningen på olika ställen. Om vi objektifierar Adress, förändras dock villkoren. En förändring av ett objekt från att låta sig refereras via "Odengatan 12" till att refereras via "Odengatan 13" betyder att objektet bytt referens eller identifiering. Det är dock samma objekt. Förändringen får genomslag på alla som refererar till objektet.

Vill vi åstadkomma samma resultat som för domän ovan, måste den enes adress sättas att peka på ett nytt objekt, ett objekt som lystrar till "Odengatan 13".

Observera att i ena fallet ändras den lokala beskrivningen av objektet, medan alla pekare från omvärlden ligger kvar. I det andra alternativet byter man en pekare, medan lokal beskrivning ligger fast. Mot domäner finns bara det senare alternativet.



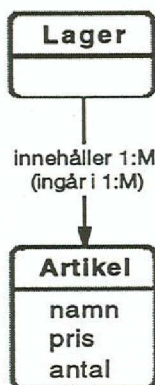
Figur 6

Avser man att referera till en geografisk adress eller postadress, att se adressen som företeelse, är figur 6 en naturlig lösning.

Finns det grupperingar av dessa slag, som på något sätt hänger ihop, står för någonting väsentligt inom applikationen, så bör man göra en närmare undersökning kring om ett nytt objekt eller aggregerad domän ska skapas. Kan man konstatera att mer än ett annat objekt på förekomstnivå kan komma att referera till företeelsen, antingen via samma eller via olika sambandstyper så har man ett klart tecken på att företeelsen har en allmängiltighet utöver det syntaktiska, varför det bör skapas som objekt.

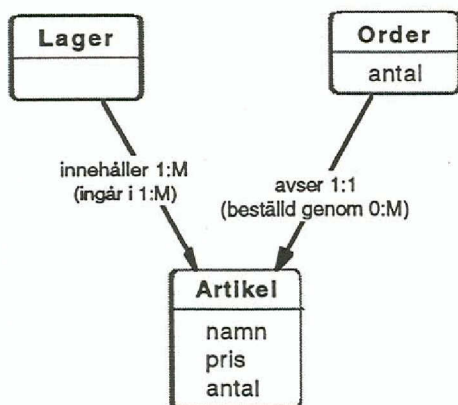
## Distinktion mellan typ och förekomst

Ett postorderföretag säljer en mängd artiklar. Man förvarar artiklarna i flera lager. För varje *artikel* behövs uppgift om *namn*, *pris* och *antal*. En viss typ av artikel kan finnas i flera lager.



Figur 1

Nu vill man även inkludera en mycket enkel orderhantering. Den går ut på att registrera per *order* vilken artikel som beställts och i vilket antal.

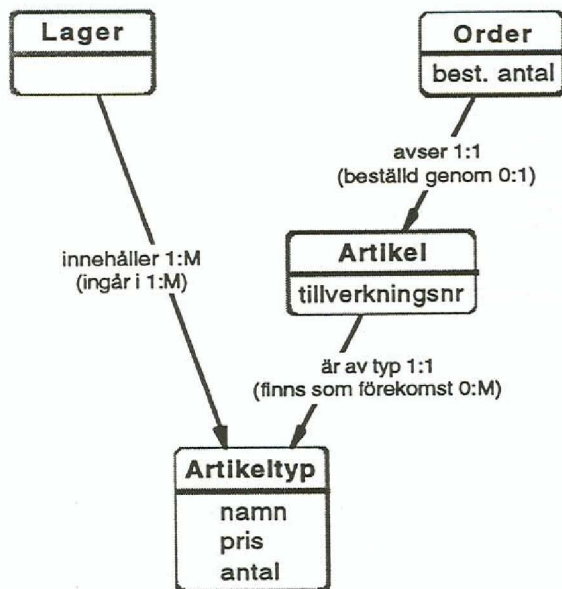


Figur 2

Om vi nu tittar lite närmare på ex.vis artikeln "cykel" är den att betrakta som en *typ* av artikel (vi bortser ifrån att det dessutom kan finnas flera olika typer av cyklar: dam-, herr-, racer-, .....cyklar, som dessutom kanske indelas per fabrikat). Det kan ju sedan i ett visst lager finnas 25 st cyklar, eller, om



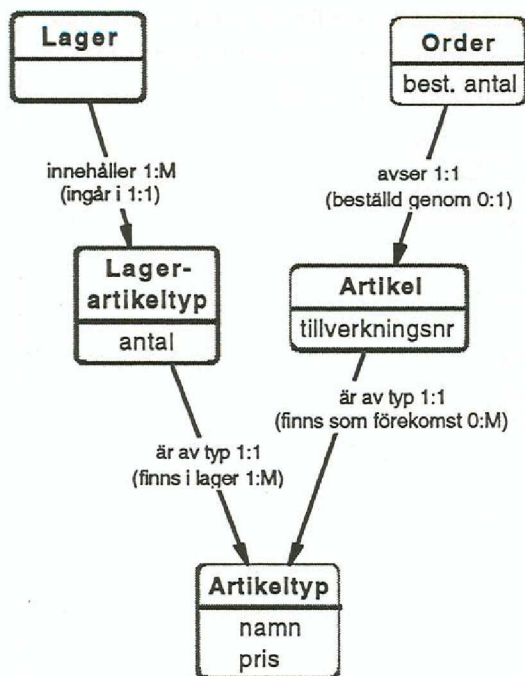
man så vill, förekomster av artikeln. Antag att vi skulle vilja hålla både typ- och förekomstinformation, exvis *pris* (typ) och *tillverkningsnummer* (förekomst) för varje cykel. Hur skulle modellen se ut då?



Figur 3

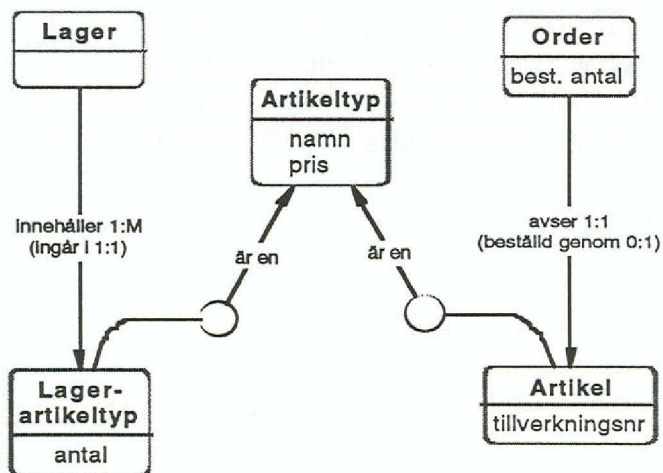
Nå, bör inte typ och förekomst normalt befinna sig på olika modell- eller abstraktionsnivåer? Här blir det någon konstig mix. Nej, det är helt ok om de båda är del av samma modellerade "verklighet", vilket ju är fallet.

Vi har vaksamt gjort åtskillnad på typ och förekomst mot order. Vi måste också precisera vad Artikeltyp står för. Namn och pris är ju detsamma (kan vi på goda grunder anta) oavsett i vilket lager artikeln befinner sig. Däremot antar vi att antalsuppgiften ska vara lager-orienterad. Därför måste vi komplettera med en ny objekttyp dit antalsuppgiften kan placeras. Låt den heta *lagerartikeltyp*.



Figur 4

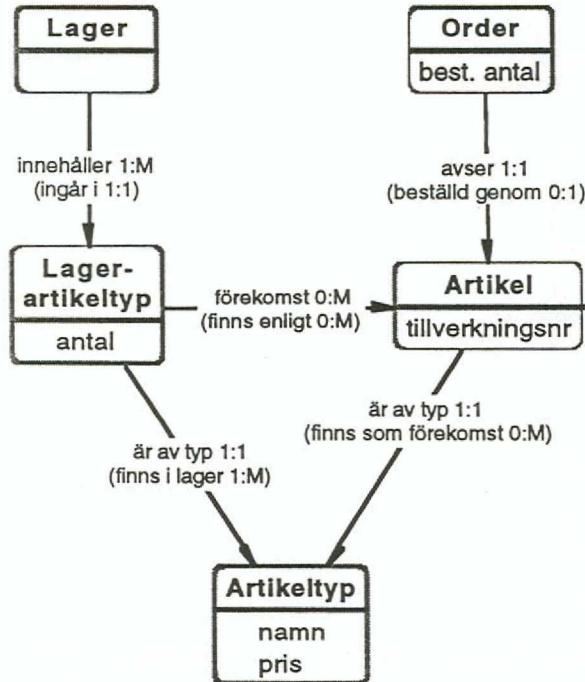
Vi kan snabbt konstatera att varje artikelförekomst en order pekar på har ett namn (exv "cykel") och ett pris (exv 750.-). Samma sak gäller för en viss lagerartikeltyp. Den har ju ett visst namn (exv "cykel") oavsett vilket lager typen befinner sig i. Samma gäller pris. I båda fallen kan namn och pris ärvas från artikeltyp. Varför inte etablera en specialisering enligt figur 5?



Figur 5

Jo, därför att det inte föreligger ett 1:1-samband utan ett M:1-samband, något man måste vara vaksam på. Alltså, glöm figur 5.

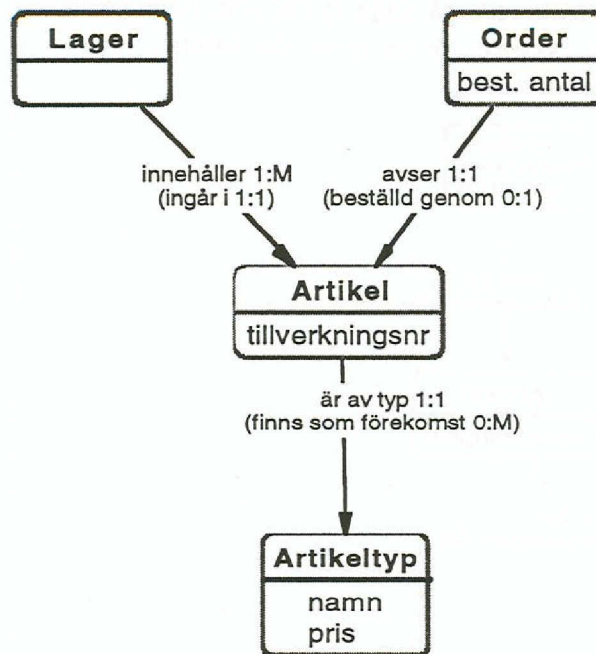
Antag nu att firman vill kunna hålla reda på vilka cyklar som finns i vilket lager, exv för att veta varifrån cyklarna för en order ska hämtas. Figur 6 visar en anpassning till denna förutsättning.



Figur 6

Och därmed borde väl allt vara i sin ordning? Nej, vi har hamnat i en ny omständighet, som sannolikt är ganska vanlig i samband med typ/förekomst-uppdelning, nämligen hantering av härledbar information. Antalet artiklar kan ju beräknas över sambandstypen *förekomst* genom att för en viss lagerartikeltyp se hur många artiklar som *förekomst* pekar på för tillfället. Vilken artikeltyp en viss lagerartikeltyp avser kan ju också härledas via *förekomst*- och *är av typ*-sambandstyperna. Kvar är ett objekt utan egna egenskaper, som bara tjänstgör som sammanbindningspunkt mellan lager och artikel. Varför inte ersätta objekttypen lagerartikeltyp med en sambandstyp mellan lager och artikel?





Figur 7

Hur långt vi egentligen ska hårdra "rensningen" kan bara bedömas utifrån vetenskapen om de totala informationsbehoven och hur central roll den härledbara informationen har där.

Den som vill resonera vidare i dessa banor kan exv komplettera modellen för att täcka in att det i lagret, förutom unika, identifierbara artiklar, även finns artiklar som säljs i kvantiteter, exv cykelkedjor och skruvmejslar. Vi har förberett för detta genom att hålla kvar attributtypen *best. antal* under Order utan att den i figurerna 3 till 7 haft någon roll att fylla (underförstått = 1). Tillåt nu en order att ha ett beställt antal > 1 för både förekomst- och kvantitetsartiklar.

## 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 nytan 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.

### Informations spridning

Det sjätte delprojektet "Informations spridning" 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.