

---

Krav på IA

---

Nästa Generation Modellering

---

Avancerad utbildning för handledare

---

Katalogprinciper

---

Verktyg

---

Informationspridning

---

Rapport K nr 1: IRDS

---

Rapport K nr 2: IRDS Modeller och modellnivåer

---

Rapport K nr 3: Koppning begreppsmodell - relationsmodell

---

Rapport K nr 4: IBM:s Repository Manager- en Introduktion

---

Rapport K nr 5: IBM:s Repository Manager: Datamodelleringsbegreppen

---

Rapport K nr 6: IBM:s Repository Manager: Begreppsmodellering i Information Model

# IBM:s Repository Manager Begreppsmodellering i Information Model

## Stig Berild

Rapporten är skriven i och för TRIAD  
delprojekt Katalogprinciper.

### Spridningsförbehåll:

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

© TRIAD-parterna 1991

# Innehåll

## IBMs Repository Manager: Begreppsmodellering i Information Model

1.	Inledning .....	1
2.	Notationsregler .....	2
3.	Översikt över Information Model.....	5
3.1.	Inledning .....	5
3.2.	Uppbyggnad .....	6
4.	Schema för att hantera begreppsmodeller.....	8
4.1.	Entitetstyper och egenskaper.....	8
4.2.	Samband .....	9
4.3.	Identifiering.....	12
4.4.	Specialisering.....	12
4.5.	Domän-begreppet .....	16
5.	Kompletterande beskrivning av vissa entity type förekomster.....	19
5.1	ENTITY TYPE .....	19
5.2	ATTRIBUTE TYPE .....	21
5.3	RELATIONSHIP LINK .....	22
5.4	RELATIONSHIP TYPE .....	23
5.5	IDENTIFIER .....	23
5.6	SUBTYPE SET .....	24
6.	Sammanfattning.....	25



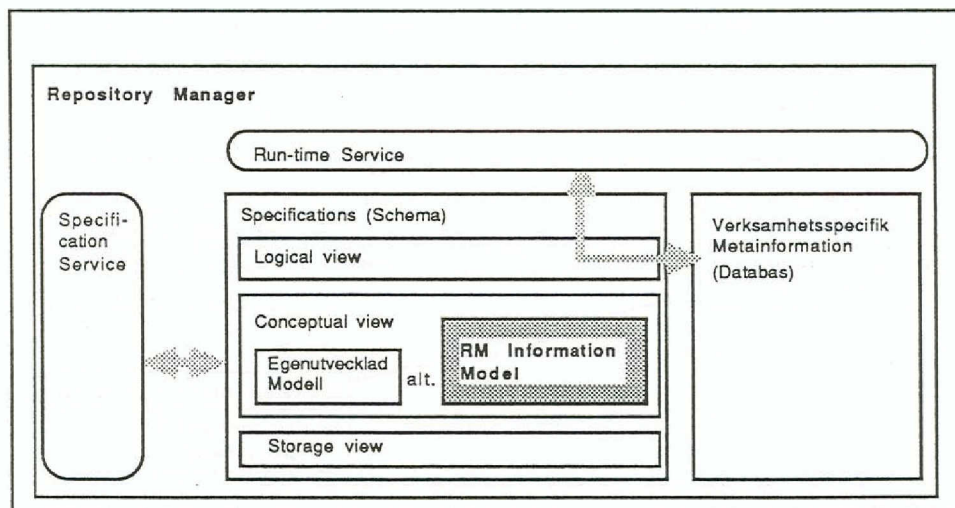
# IBMs Repository Manager:

## Begreppsmodellering i Information Model

### 1. Inledning

En introduktion till IBMs Repository Manager ges i rapporten [Triad K nr 4]. Där identifieras ett antal specifika områden att belysa mer i detalj. Ett av dessa områden är det schema, som levereras med produkten. Schemat går vanligtvis under beteckningen **Information Model**. I IBM-manualer betecknas den **Supplied ER Model**. Dess plats i översikt bilden nedan är gråtonad.

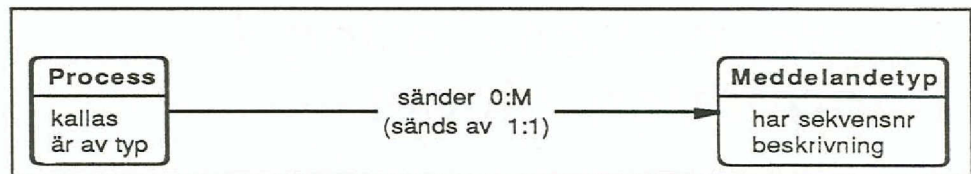
Modellen är i sig mycket omfattande och kan inte avhandlas i sin helhet i en kort rapport. Vi har valt att här fokusera på den del av schemat som hanterar begreppsmodeller (datamodeller). Anledningen är att datamodellering är en central aktivitet vid all systemutveckling och den av de olika delmodellerna, som bedöms vara mest känd och debatterad. Många hävdar dessutom att begreppsmodellen är att se som navet eller centralpunkten i Information Model eftersom andra delmodeller ofta uttrycker samband med begreppsmodelleringsdelen.



FIGUR 1

## 2. Notationsregler

Den grafiska presentationen av schemata och exempel görs med hjälp av en Telmod-liknande notation, en notation framtagen i samarbete mellan Televerket och SISU. Figur 2 visar ett exempel på ett schema.



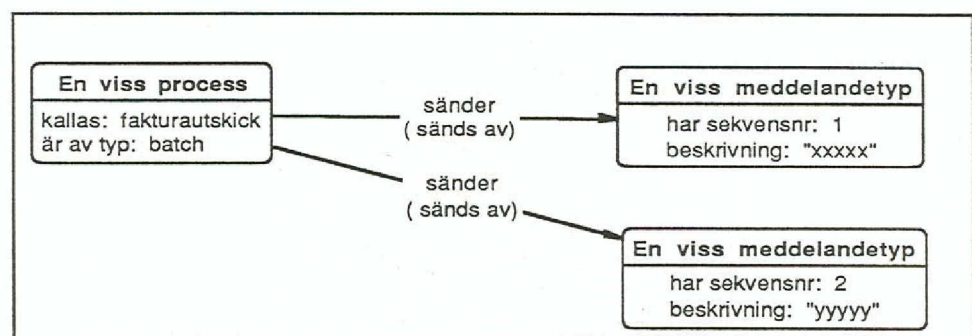
FIGUR 2

I enlighet med RMs begreppsapparat (se vidare rapport [Triad K nr 5]) är *Process* och *Meddelandetyptyp* exempel på två **entity types**. *Process* har **attribute types** *kallas* och *är av typ* medan *Meddelandetyptyp* har **attribute types** *har sekvensnr* och *beskrivning*. Mellan dessa två **entity types** finns ett **relationship type** som ur *Process*'s synvinkel benämns *sänder* och ur *Meddelandetyptyp*'s synvinkel benämns *sänds av*. **Huvudriktningen** är *sänder* (utan parenteser) medan *sänds av* är **inversriktningen** (med parenteser).

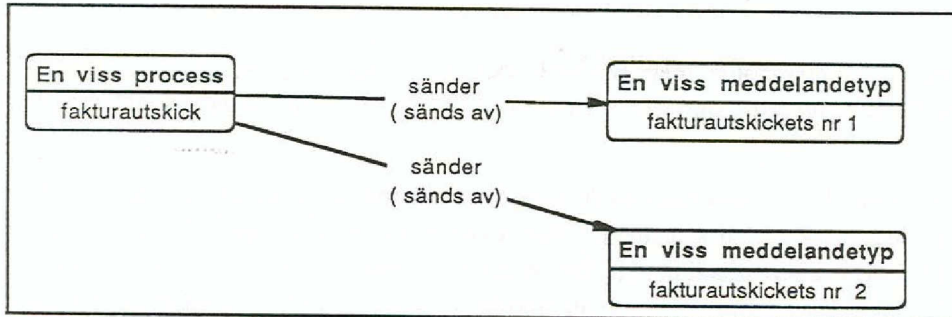
En process får sända ingen (0) eller flera (M) meddelandetyper, vilket anges efter *sänder* (0:M). Enligt samma tolkning måste en meddelandetyptyp sändas av minst en (1) och högst en (1) process (1:1). Motsvarande uppgifter skulle kunna anges för **attribute types**.

En möjlig förekomstmodell i enlighet med schemat i figur 2 visas i figur 3. Prefixet **En viss** indikerar att det är fråga om en förekomst.

Är vi inte intresserade av att presentera attributen för en viss entitet, placeras endast ett tänkt namn på förekomsten, under strecket i den grafiska symbolen. Se exempel i figur 4.



FIGUR 3



FIGUR 4

Det är viktigt att i fortsättningen hålla isär den terminologi eller begreppsapparat som används för att förklara egenskaper, struktur, regler mm hos Information Model (IM) och de begrepp som används i IM. Tyvärr tillämpar IBM en delvis överlappande terminologi för de två behoven.

I IM förekommer tex begreppen process, info flow, requirement, entity type, relationship type. När vi "pratar om" dessa företeelser är de var och en exempel på de entity types, som finns i IM.

Mellan process och info flow finns samband, mellan entity type och relationship type också. När vi "pratar om" dessa samband är de var och en exempel på de relationship types, som finns i IM.

De egenskaper som exv används för att beskriva en process, eller generellare, en viss entity type är exempel på attribute types i IM. Se vidare diskussioner och exempel på olika modellnivåer i rapporten [Triad K nr 2].

En annan källa till förvirring i modelleringssammanhang är distinktionen mellan typ och förekomst.

Följande regler gäller fortsättningsvis.

Begreppen vi använder för att "prata om" företeelserna i IM skrivs utan speciell redigering (exv attribute type). Talar vi allmänt om förekomster av dessa anger vi helt enkelt enligt "en förekomst av xxx". INFO FLOW är exempel på en förekomst av entity type. Som synes anges varje sådan förekomst med stora bokstäver.

För att allmänt tala om en förekomst av ett begrepp i IM, anges begreppet med små bokstäver omgärdat med <> (exv <info flow>). Specifik förekomst anges kursiverat. Följande två meningar följer dessa regler.

Exempel på förekomst av entity type är PROCESS. Exempel på förekomst av entity type PROCESS är *fakturauskick*, medan entity type ENTITY TYPE exv kan ha en <entity type> *person*.

I grafema tillåter oss notationen att ange företeelserna lite friare och mer läsvänligt samt utan risk för missförstånd. Således anges <info flow> som "Ett visst info flow" över rutans streck medan *fakturauskick* anges okursiverat under strecket. Förekomster av entity types och relationship types anges i IM schemata med små bokstäver.

## 3. Översikt över Information Model

### 3.1. Inledning

Ett repository (informationsresurskatalog) kan innehålla en mångfacetterad beskrivning av en verksamhet och verksamhetens informationsstöd. Dit hör exv modeller över verksamhetens funktioner och deras informationsbehov, informationservice i form av applikationer och data, hård- och mjukvarukarakteristik. Case-verktyg stödjer hantering av denna typ av information. Varje verktyg har sin specialitet och hanterar därmed delvis olika modelltyper och med sin profil. Trenden (kundkraven) går i riktning mot en alltmer överensstämmande uppfattning om hur de olika modelltyperna ska struktureras.

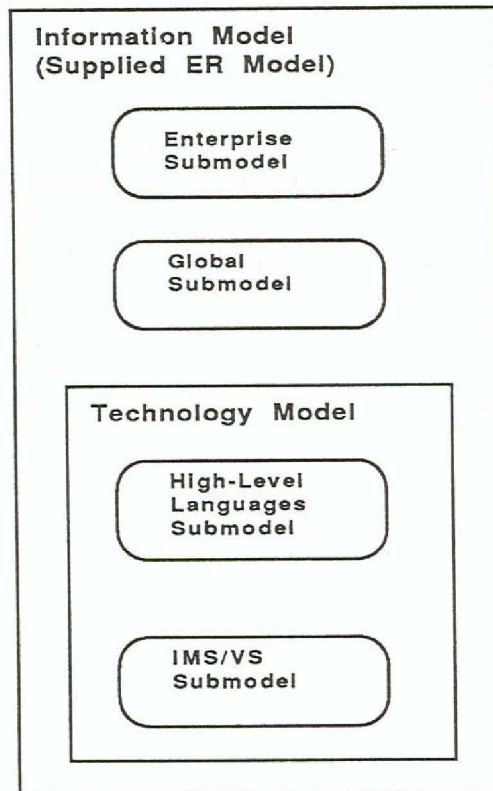
I IBMs AD/Cycle-strategi ligger, som en bärande idé, just tanken på en enhetlig plattform, en enhetlig syn på uppbyggnaden av informationen i ett repository. Information Model är schemat för denna information.

Med IRDS-terminologi svarar Information Model mot nivå 3 (IRD Definition Level), dvs mot schemat i det andra nivåparet (IRD Level pair). Se vidare rapporterna [Triad K nr 1 och 2].



### 3.2. Uppbyggnad

Information Model kommer att vidareutvecklas under lång tid. Kommer en sådan generell modell någonsin att bli "färdig"? Nya behov, nya synsätt, nya förutsättningar, ny teknik, ökad erfarenhet mm kommer nog mer eller mindre kontinuerligt att ställa nya krav. Vad som redovisas i denna rapport avser Repository Manager, Version 1, Release 1. Uppbyggnaden redovisas i figur 5.



FIGUR 5

**Enterprise Submodel (ENT)** stödjer och omfattar de modeller man normalt opererar på i analys- och designfasen. Där kan bla begrepp, begreppsmodeller, dataflöden, processer och processstrukturer beskrivas. Det är inom denna modell vi hittar de delar, som är av intresse för föreliggande rapport.

**Global submodel** hanterar företeelser som är gemensamma för flera delmodeller. Dit hör generell hantering av text och även den del som hanterar "object", något som har likheter med mekanismer inom objektorientering.

**Technology Model** är en gruppering av implementeringsnära modeller. Av dessa används **High-Level Languages Submodel** för att bygga datastrukturer för programmeringsspråk. En del är programspråksberoende, en

annan del är specifikt orienterad mot Cobol-data definitioner. Även andra språk kommer säkerligen att inkluderas allteftersom. På motsvarande sätt beskriver IMS/VS Submodel IMS-datastrukturer. Inom denna kategori förväntas snart även submodellen för DB2.

Det bör poängteras att de enda kopplingar, som för närvarande finns mellan submodeller, går mot global submodel. Den som exv vill koppla en begreppsmodell beskriven genom Enterprise submodel till vad den resulterat i i form av en IMS datamodell måste definiera egna utvidgningar av Information Model.

Det kan här vara på sin plats att notera att den formella angivelsen av begrepp i IM inkluderar ett trebokstavsprefix, som står för begreppets submodell-tillhörighet. ENT är tex prefixet för Enterprise Submodel-begrepp. Underscores används istället för blanka. Fullständigt entydig beteckning på en förekomst av relationship type inkluderar namnet på de båda sammanbundna entity type-förekomsterna, med den entity type-förekomst pilen utgår ifrån som prefix och mål-entity type-förekomsten som suffix. ENT\_INFO\_FLOW är beteckningen på en entity type- förekomst i Enterprise Submodel. ENT\_ATTRIBUTE\_TYPE\_DESCRIBES\_ENTITY\_TYPE är beteckningen på en relationship type-förekomst i Enterprise Submodel, nämligen den som talar om att ENT\_ATTRIBUTE\_TYPE beskriver ENT\_ENTITY\_TYPE.

För att inte tynga ner framställningen i onödan följer vi de under avsnitt 2 beskrivna reglerna.

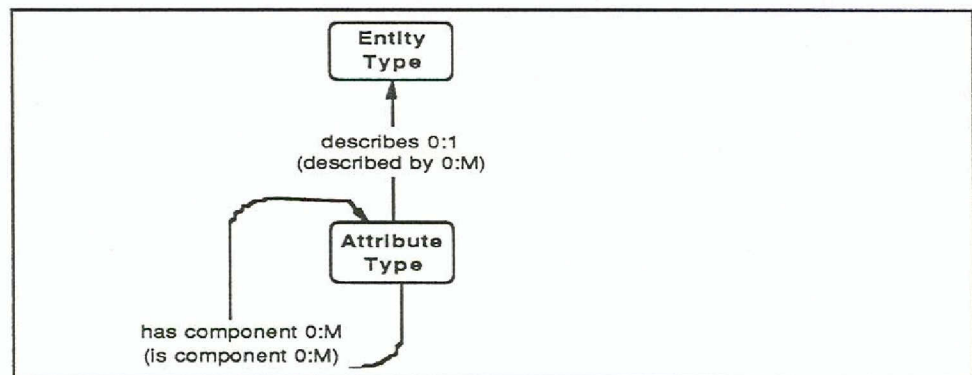
## 4. Schema för att hantera begreppsmodeller

Under avsnitt 4 kommer vi att successivt bygga upp den del av Enterprise Submodel (ENT), som hanterar begreppsmodeller.

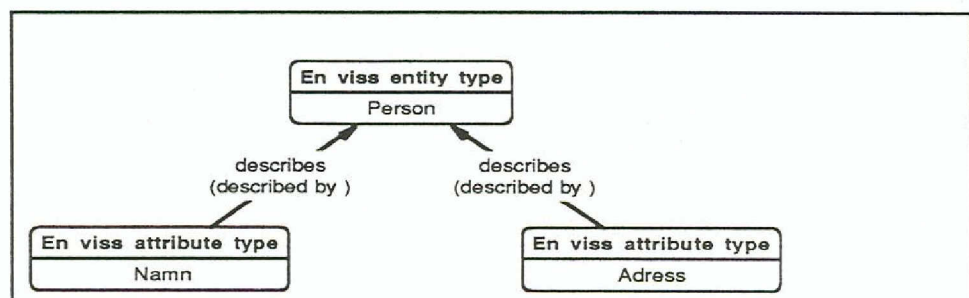
### 4.1. Entitetstyper och egenskaper

En förekomst av typen ENTITY TYPE (en <entity type>) är en företeelse för vilken man vill uttrycka egenskaper och/eller samband. Exempel är *order*, *person*, *artikel*, *bil*.

Detta uttrycks i ENT genom att ENTITY TYPE relateras till ingen, en eller flera typer av egenskaper eller ATTRIBUTE TYPEs. Schema, så långt, visas i figur 6 och exemplifieras i figur 7.



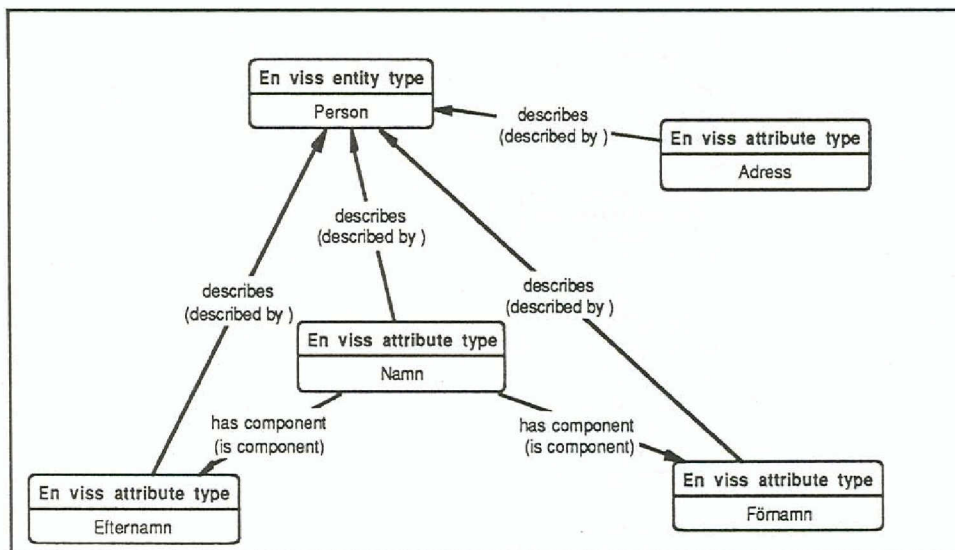
FIGUR 6



FIGUR 7

En ATTRIBUTE TYPE är aldrig relaterad till mer än en ENTITY TYPE. *Namn* på *person* och *namn* på *artikel* redovisas alltså som två ATTRIBUTE TYPES. Den vaksamme läsaren noterar att DESCRIBES med 0:1 verkar underligt. En ATTRIBUTE TYPE är ju ENTITY TYPE specifik men kan ändå "leva" som fristående företeelse?! Förklaringen ligger i att ATTRIBUTE TYPE även används till annat än att beskriva ENTITY TYPES. Efter vad vi uppfattat kan bli en användning vara som variabel i ett Cobol-program (i Working-Storage Section). En <attribute type> kan i alla händelser etableras innan den har någon <entity type> att beskriva. Denna uppdelade användning känns inte helt naturlig.

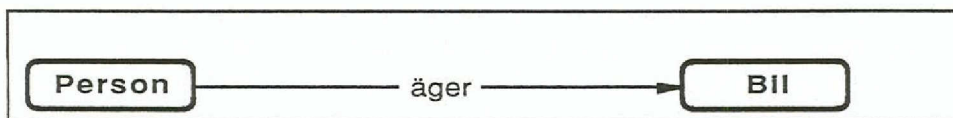
ATTRIBUTE TYPES kan vara enkla eller sammansatta ner till valfritt djup. Detta regleras med relationship type HAS COMPONENT. Se figur 6 för schemat och figur 8 för exempel.



FIGUR 8

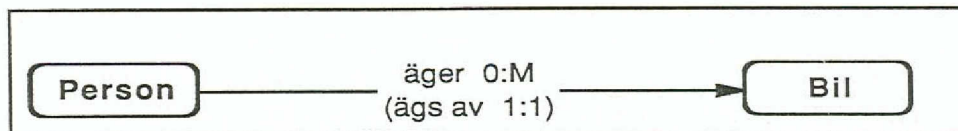
## 4.2. Samband

I en verksamhetsbeskrivning vill man kunna hantera samband mellan olika företeelser, likaså i en datamodell. Kanske är ägar-samband mellan personer och bilar av intresse. Se figur 9.



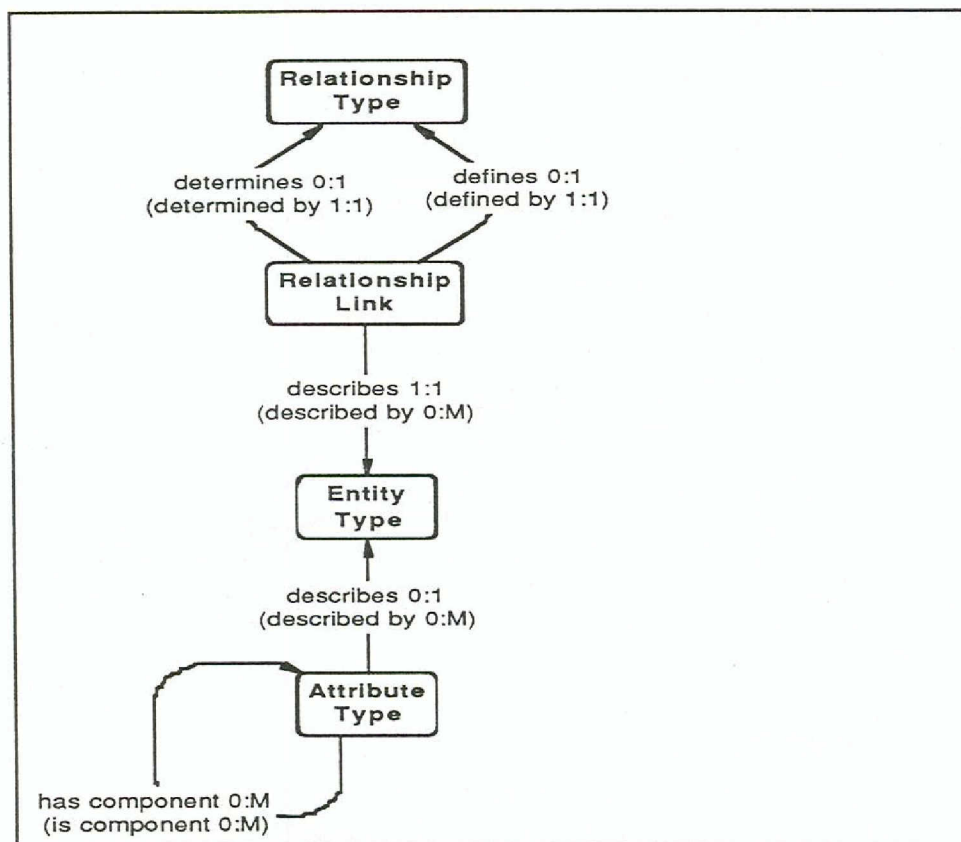
FIGUR 9

Sambandet kan helt enkelt ses som en hoplänkning av två <entity types>. Samma samband kan också upplevas i respektive inblandad <entity type>'s perspektiv. En viss *person* anser säkert att han *äger bilen* medan *bilen* (om man hade kunnat fråga) känner att den *ägs av* eller ställer sig till förfogande för en viss *person*.



FIGUR 10

Båda perspektiven är lika riktiga. De är just två perspektiv på samma samband. Denna distinktion är väsentlig eftersom det finns behov att beskriva både sambandet som neutral företeelse och respektive perspektiv. Avsnitt 5 redovisar exempel på detta. ENT använder begreppet RELATIONSHIP TYPE för den neutrala synen och RELATIONSHIP LINK för perspektiv-synen.

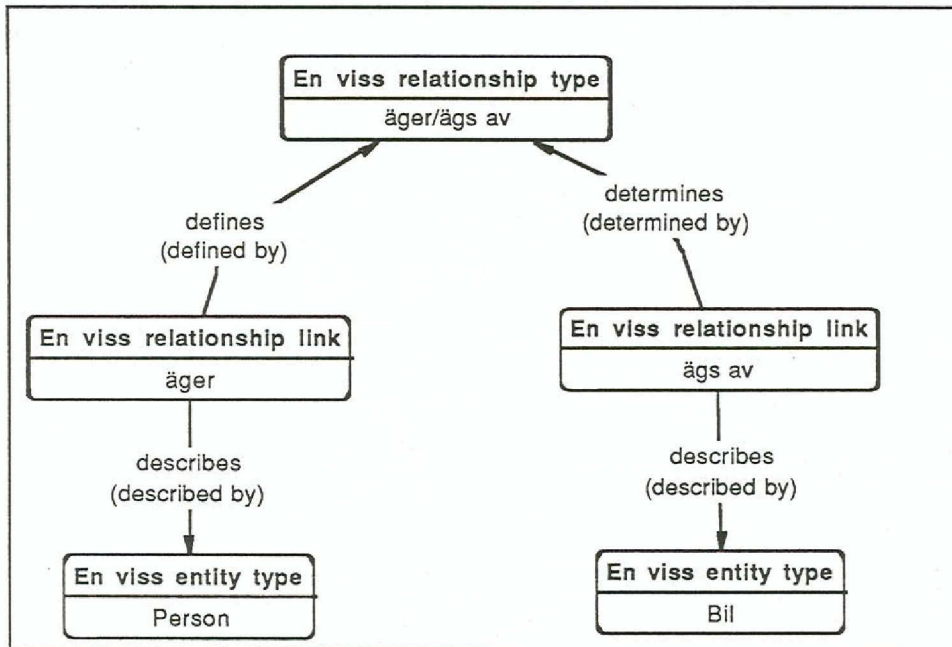


FIGUR 11

Som synes i figur 11 sammanförs de båda perspektiven till sambandet via två olika relationship types. Anledningen till detta är att man inte bara vill koppla perspektiven till rätt samband. Det finns ett kompletterande behov av att ange vilket av perspektiven som är huvudriktning och vilket som är så kallad inversriktning. DEFINES används för huvudriktningen och DETERMINES för inversriktningen.

Intressant att notera är att Televerkets referensmodell gör samma distinktion genom begreppen "samband" respektive "relation" samt mellan huvud- och inversriktning. RMs begreppsapparat skiljer också på huvud- och inversriktning som synes av schema-graferna. Däremot delas inte sambandet upp i type respektive link.

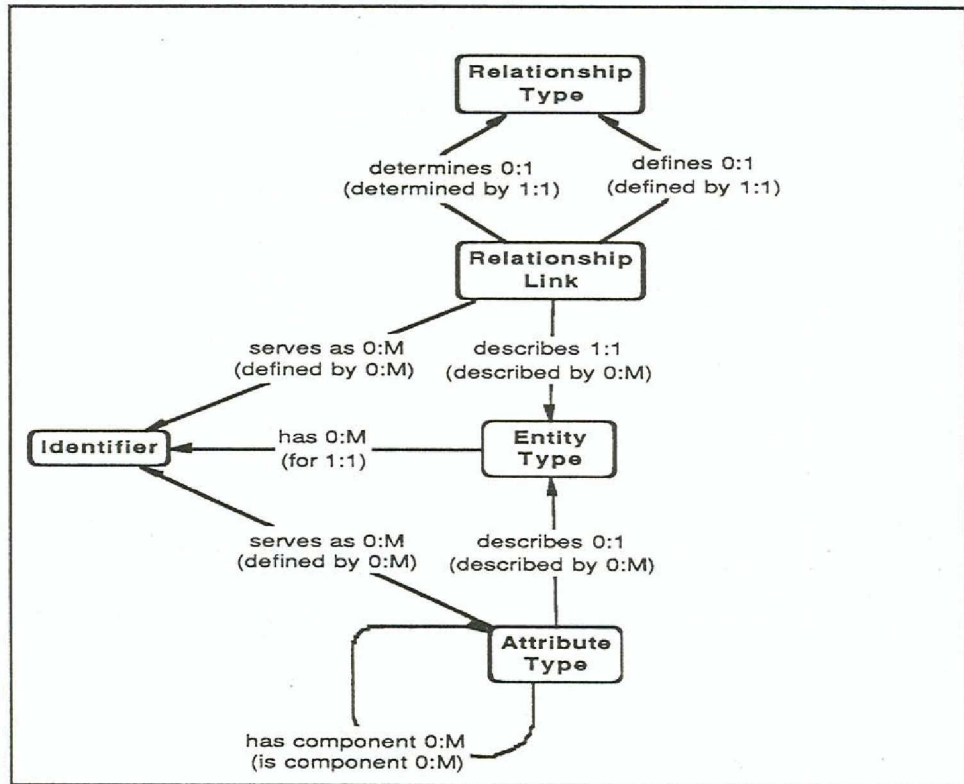
Innehållet i figur 10 ovan, strukturerat i enlighet med schemat i figur 11, blir (vi bortser här från eventuella <attribute types>) enligt figur 12.



FIGUR 12

### 4.3. Identifiering

Referens till viss <entity type> antas ske med hjälp av en kombination av <attribute types> och/eller <relationship links>. (Även entity type COMPOUND FACT kan vara en sådan beståndsdel, men vi bortser tills vidare från detta.) Unika referenser beskrivs i ENT under entity type IDENTIFIER. En <entity type> måste minst ha en <identifier>, men får ha flera. Schemat utökas enligt figur 13.



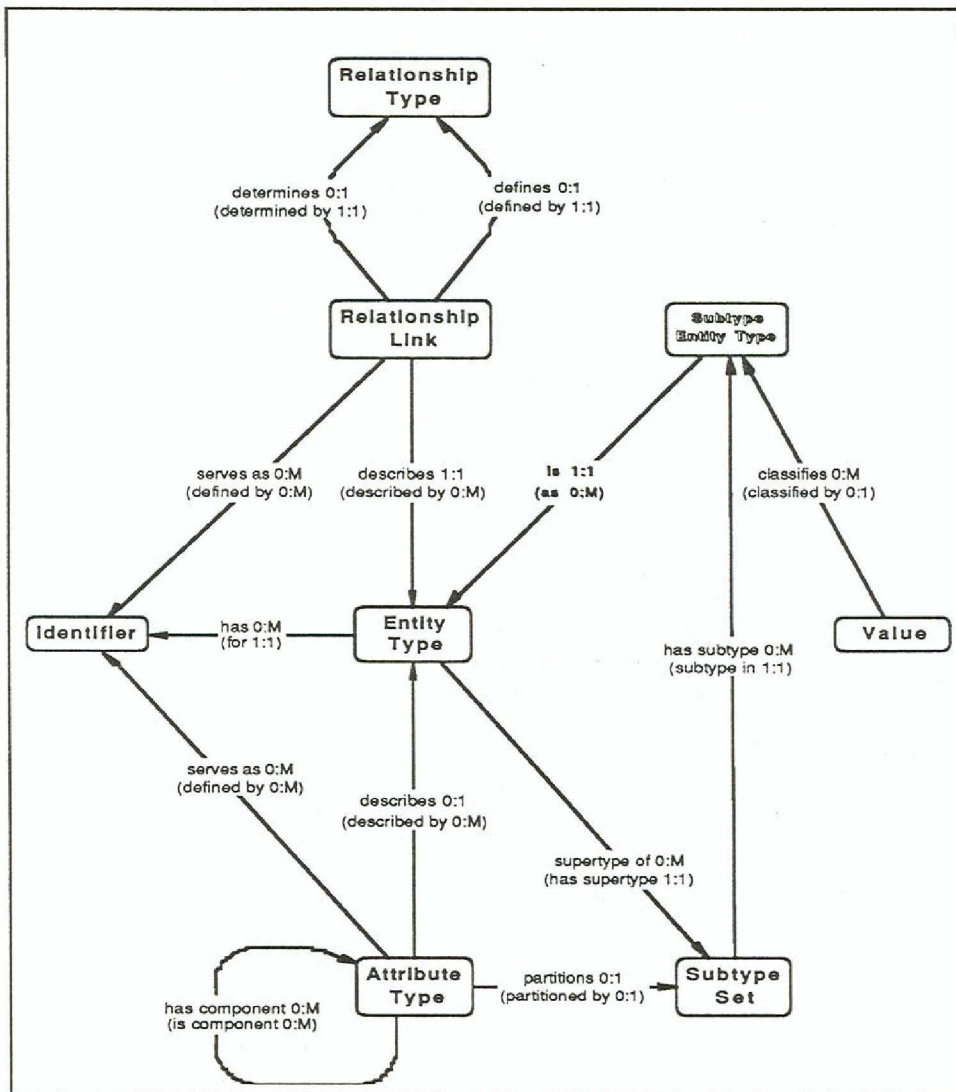
FIGUR 13

### 4.4. Specialisering

Specialiseringar och generaliseringar av ENTITY TYPE kan formuleras inom ENT. Man talar i dessa sammanhang ofta om specialiserings- eller arvshierarkier. Renodlade hierarkier (enkla arv) innebär att en <entity type> kan specialiseras i flera sub-<entity types> men att en sub-<entity type> bara kan vara specialisering av högst en <entity type>. ENT stipulerar inga sådana begränsningar. Multipla arv tillåts, dvs resultatet behöver inte bli en hierarki, även nätverk accepteras.

Ibland behöver man kunna definiera flera olika uppsättningar specialiseringar för en och samma <entity type>. Orsaken är att indelningen görs utgående från en specifik förutsättning, en specifik aspekt. En person kan specialiseras inom

aspekten *kön* i *man* och *kvinna* men samtidigt även inom aspekten *nationalitet* i *svensk* och *norrman*. I ENT ger varje sådan indelningsgrund upphov till ett <subtype set> av typen SUBTYPE SET. Ett <subtype set> pekar ut de <entity types>, som är specialiseringar inom den givna indelningsgrunden, med hjälp av HAS SUBTYPE. Se det utökade schemat i figur 14 nedan.



FIGUR 14

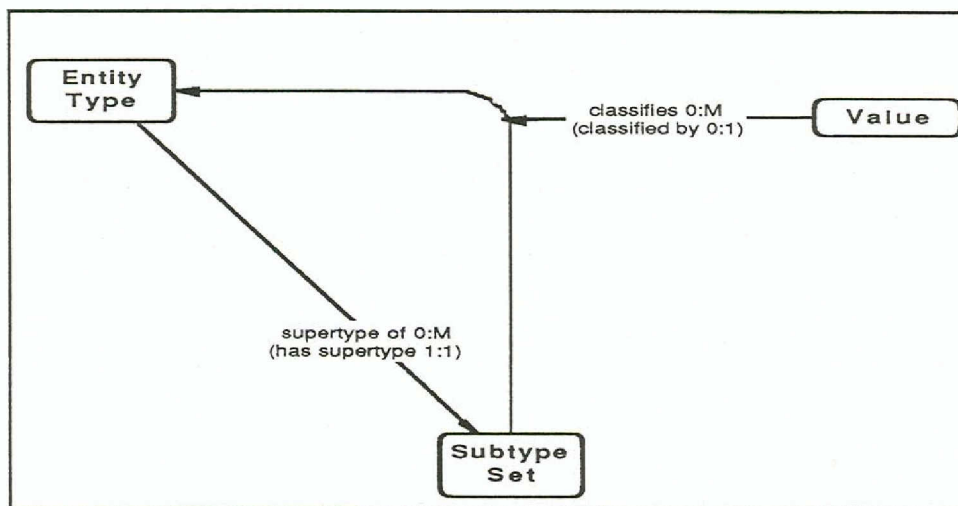
Om indelningsgrunden är kopplad till värdet på ett visst <attribute type>, anges detta genom relationship type PARTITIONS. Vet man till och med vilket värde, som ger upphov till vilken sub-<entity type>, kan även detta anges genom att associera VALUE över CLASSIFIES till SUBTYPE ENTITY TYPE.

RMs begreppsapparat tillåter att relationship types sammanbinder både entity types och relationship types. Specialisering uttrycks med dessa förutsättningar enligt figur 15. Se vidare rapporten [Triad K nr 5]. Som vi redan klarat ut i



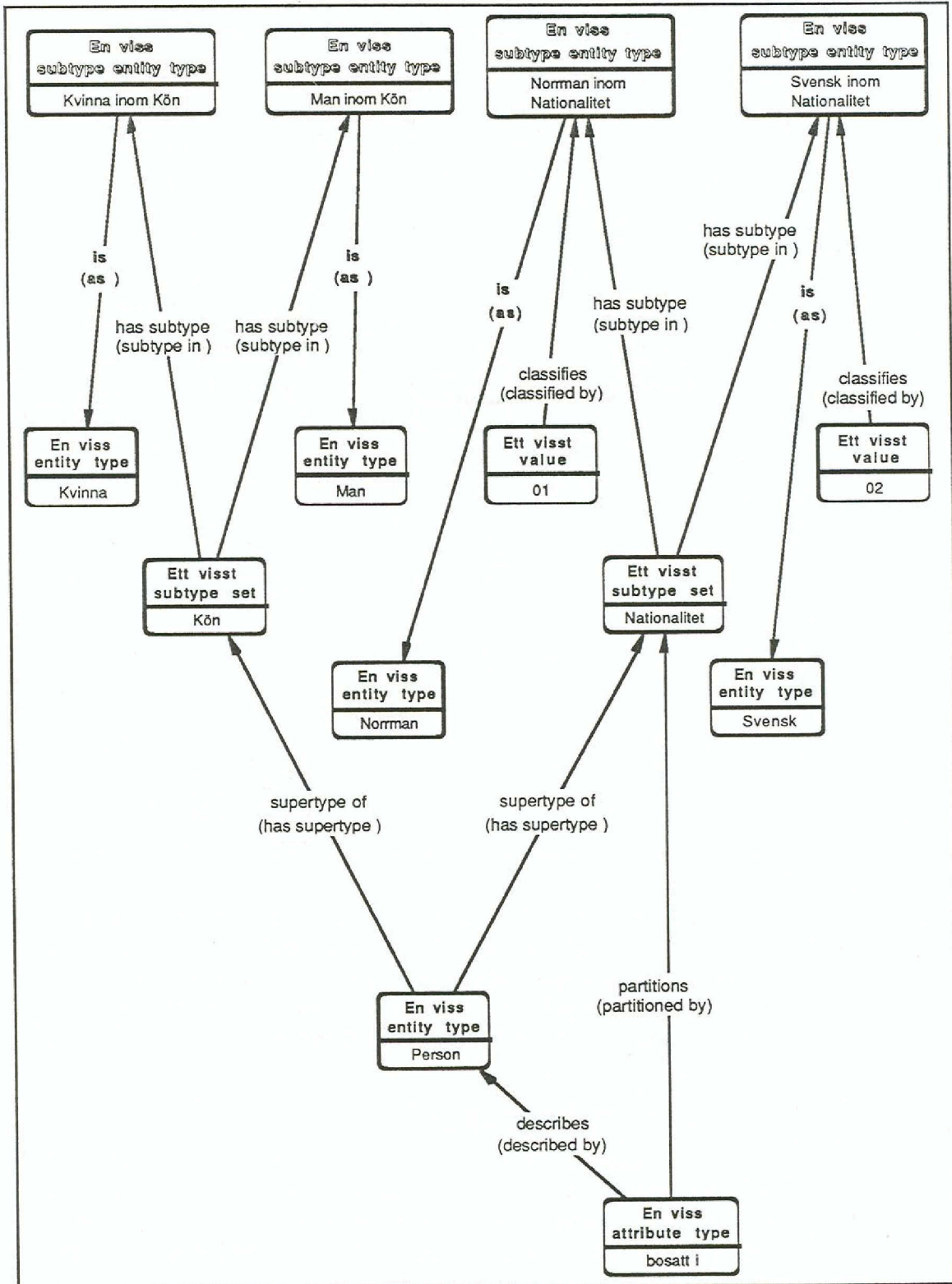
avsnitt 4.2 accepterar inte ENT sådana samband. I ENT sammanbinder ett <relationship type> endast två <entity types> med hjälp av två <relationship links>. IBM har alltså, av någon anledning, valt två olika principer på nivå 4 och på nivå 3 (i IRDS-terminologi).

Eftersom all verksamhetsspecifik information kommer att struktureras med hjälp av den begreppsapparat som definieras på nivå 3, dvs ENT-nivån, är det principerna enligt denna nivå som CASE-verktygsanvändare kommer i kontakt med (om någon). Av samma anledning har vi i denna rapport valt att presentera ENT-schemat enligt samma principer som ENT själv kan hantera verksamhetsspecifika schemata. Dock indikerar vi avvikelserna med skuggad text eftersom de inte ingår i den officiella vokabulären. Vi visar också ett utsnitt av avvikelserna i form av den officiella versionen.

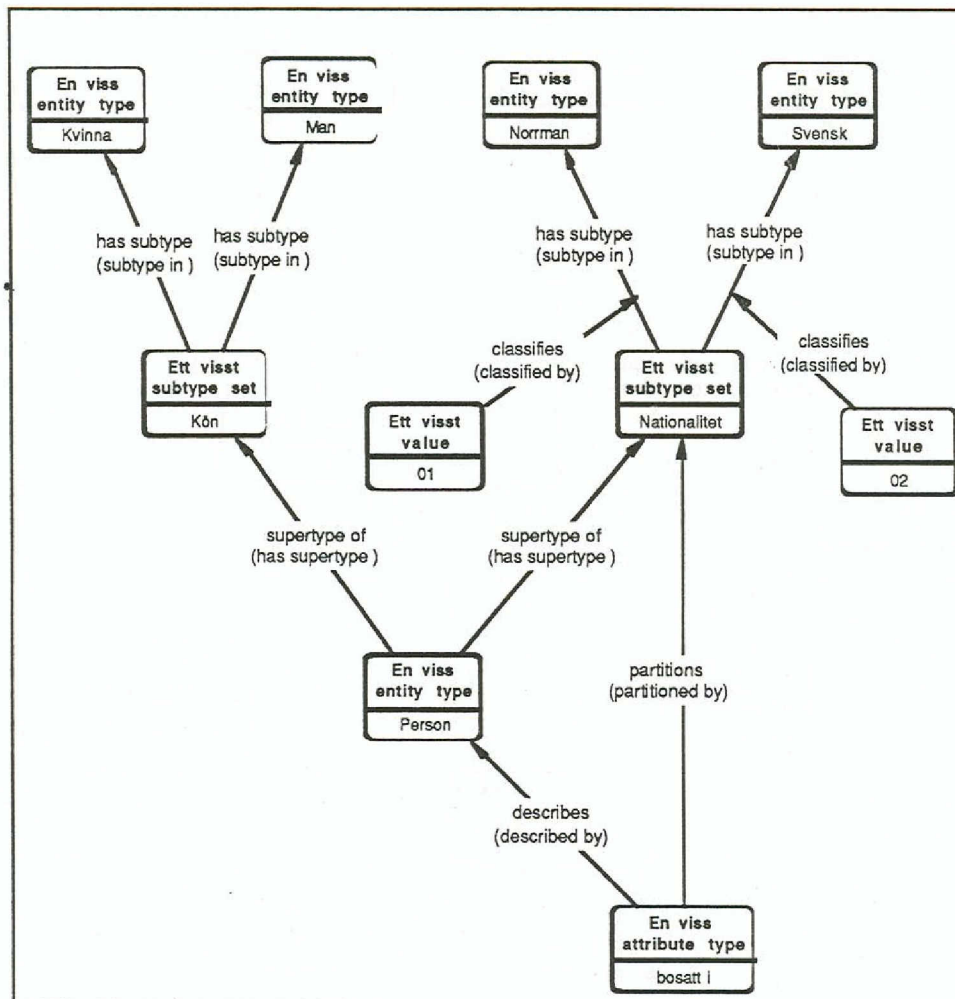


FIGUR 15

Antag att *person* indelas i specialiseringar med avseende på *kön* och *nationalitet* och att <attribute type>*bosatt i* har värdet 01 för Norge och 02 för Sverige. Figur 16 blir resultatet. Ett utdrag av samma sak enligt RMs modelleringsspråk visas i figur 17.



FIGUR 16



FIGUR 17

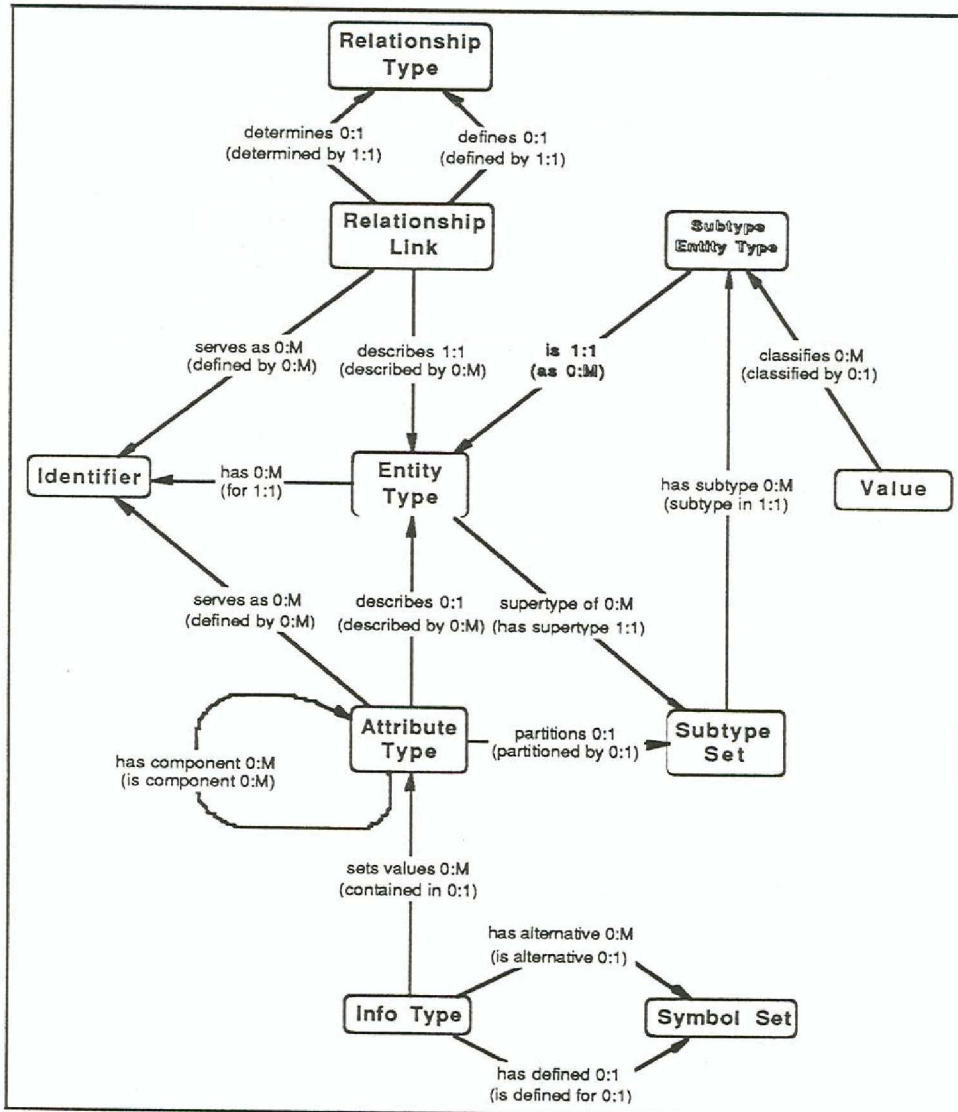
Observera att begreppet arv här endast används för att specificera en omständighet, som är en tillräcklig förutsättning för en möjlig arvsmechanism. Om, och i så fall hur, denna kunskap kommer att utmynna i arvsmechanismer i en run-time-miljö, är en fråga för de som har att realisera en applikation utgående från given specifikation. (Den kanske aldrig ens är tänkt att vara grunden för en realisering).

## 4.5. Domän-begreppet

Enterprise submodel (ENT) har en mycket ambitiös och detaljerad begreppsapparat för att beskriva de värden och symboler ATTRIBUTE TYPE kan uttryckas i. Vi noterar i detta sammanhang endast två begrepp, men kommer i en separat rapport att titta på de olika aspekterna av domän-begreppet mer i detalj.

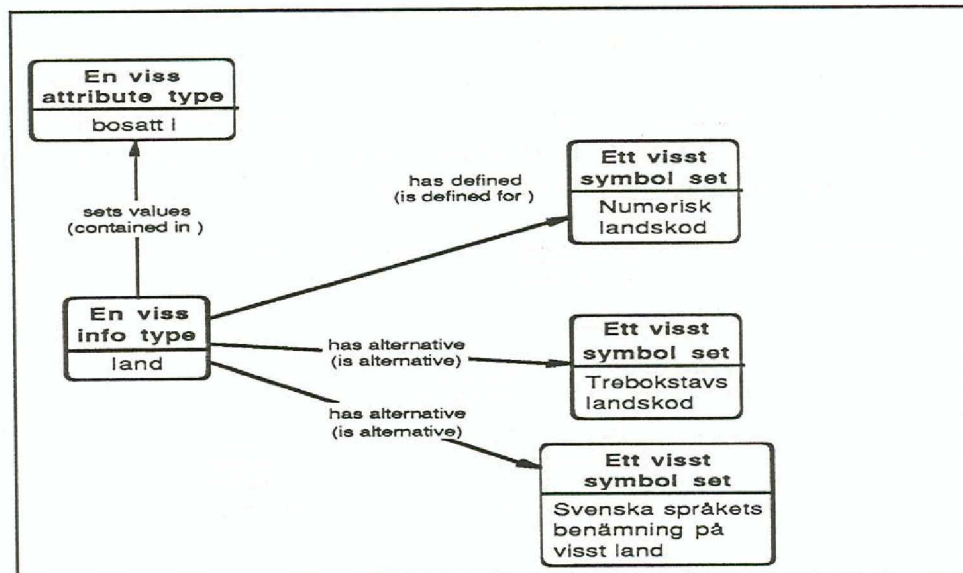
Det i andra sammanhang använda domän-begreppet är här förfinat till två begrepp, INFO TYPE och SYMBOL SET, med varsin roll. En <info type> står

för en uppsättning värden (datatyp) som används i en verksamhet och som har en vedertagen uttolkning. En <info type> uttrycks eller representeras med hjälp av ett <symbol set>. Finessen är att en <info type> har kopplat till sig ett standard-<symbol set> (HAS DEFINED) men kan alternativt uttryckas även med hjälp av andra <symbol sets> (HAS ALTERNATIVE). Man skiljer mao på dataelement och dess representation. Figur 18 visar den kompletterade schema-versionen.



FIGUR 18

Som exempel kan vi ta ett förtydligande av det tidigare använda <attribute type> bosatt i.



FIGUR 19

## 5. Kompletterande beskrivning av vissa entity type-förekomster

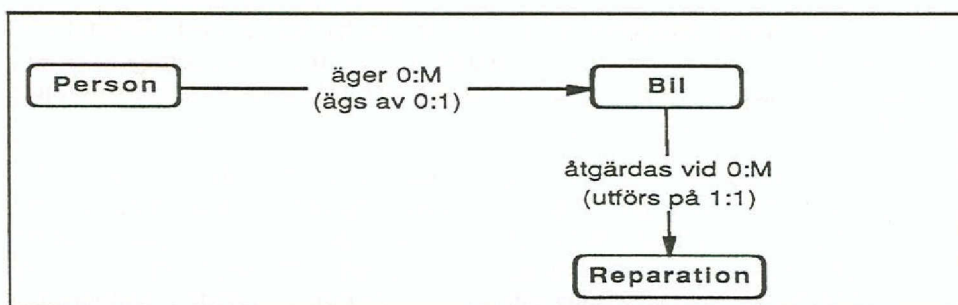
Vi har hittills koncentrerat oss på en allmän förståelse av begreppsmodelleringens entity type-förekomster i IM och deras samband. I detta avsnitt ges avslutningsvis lite mer "kött på benen" i form av exempel på förekomster av attribute types för några förekomster av entity type.

### 5.1 ENTITY TYPE

ENTITY TYPE beskrivs bla genom

name	Namnet på en <entity type>.
description	En verbal beskrivning.
maximum count	Max antal förekomster av viss <entity type> i verksamheten.
purpose	Purpose är värd en något utförligare beskrivning. Den kan anta endera av 3 värden:

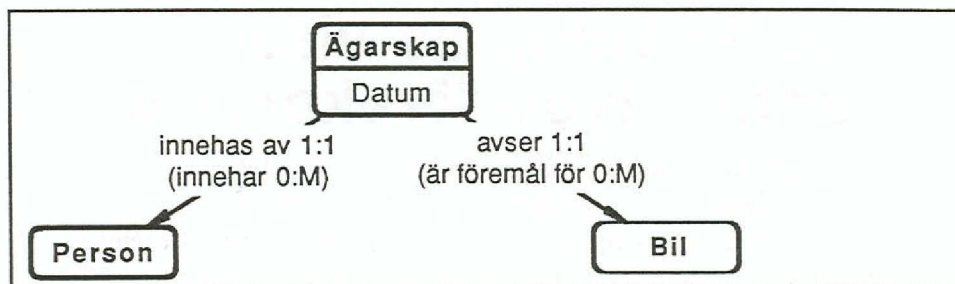
*Fundamental* anges om <entity type> till sin existens är oberoende av hur omgivningen "ser ut". Kallas ibland för "base entity types". I exemplet i figur 20 kan man anta att personer finns kvar även om de inte äger någon bil. Samma sak gäller antagligen bil. *Bil* och *Person* är var för sig exempel på fundamentala <entity types>. Däremot finns det ingen möjlighet att hålla kvar uppgifter om bilens reparationer om bilen själv försvinner.



FIGUR 20

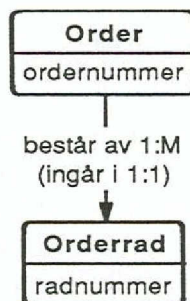
Sambanden uttrycks i ENT alltid binärt, vilket innebär att flerställiga samband måste "objektifieras", dvs representeras genom en sammanbindande <entity type>. Vill vi beskriva datum för ägarskapet (bilköpet) måste en <entity type>

*ägarskap* etableras. Se figur 21. Kopplingen eller associationen mellan *person* och *bil* har blivit en <entity type>. Sådana <entity types> ges beteckningen *associative*. Liksom *reparation* är *ägarskap* beroende av relaterade <entity types> (här *person* och *bil*) för sin egen existens.



FIGUR 21

Den tredje varianten kallas *attributive*. Innebörden är snarlik "dependent entity type" i RMs begreppsapparat och går ofta under beteckningen "weak entity type". En sådan <entity type> står i ett slags beroende till en "parent" <entity type> både i verklighetsuppfattningen och genom att den lånar identifieringen av "parent" som delidentifierare av sig själv. I figur 22 är *order* en "parent" <entity type> och *orderrad* en attributive <entity type>. Orderrad identifieras genom orders identifiering (ordernummer) kompletterad med radnummer.



FIGUR 22

Nyttan med denna distinktion i tre typer är och har länge varit, föremål för kontinuerliga diskussioner mellan företrädare för olika modelleringsansatser. Vissa ifrågasätter om en entydig distinktion enligt ovan överhuvud taget är möjlig att göra.

Man kan förmoda att IBM valt denna lösning för att erbjuda en generalitet i uttryckskraften. Därigenom kan man svara upp mot många olika behov. Exempelvis kan företrädare för flerställiga <relationship types> och förespråkare av att <relationship types> ska kunna beskrivas med <attribute types> och/eller, som i RMs begreppsapparat, ingå som source eller target för andra <relationship types>, anse sig ha behov av den beskrivna klassificeringen.

## 5.2 ATTRIBUTE TYPE

ATTRIBUTE TYPE beskrivs bla genom

<b>name</b>	Namnet på en <attribute type>. Måste vara unikt för den <entity type> som beskrivs. (Nyckeln för en <attribute type> är systemgenererad.)
<b>description</b>	En verbal beskrivning.
<b>minimum_per_entity</b>	Minsta antalet attribut av <attribute type> per entity av <entity type>. Exv 1 för <attribute type> <i>efternamn</i> till <entity type> <i>person</i> .
<b>maximum_per_entity</b>	Största antalet attribut av <attribute type> per entity av <entity type>. Exv 10 för <attribute type> <i>telefonnummer</i> till <entity type> <i>person</i> . Observera att den generella beteckningen "flera" i form av M e dyl inte tycks accepteras. Värdet måste vara null eller ett positivt tal.
<b>derivation_type</b>	Anger hur attributvärden tas fram. Härunder döljer sig några intressanta möjligheter.
Fyra alternativ erbjuds.	
Basic	Den vanliga situationen när värden finns explicit angivna.
Computed	Innebär beräkning av värden enligt en algoritm. Algoritmen finns i detta alternativ definierad under <b>derivation_formula</b> . För att kunna tolka algoritmen behöver man känna till det språk den uttrycks i. Aktuellt språk anges under <b>derivation_formula_language</b> .
Indirect	Värden hämtas via en indirekt "koppling" i modellen. Kopplingen byggs upp av en eller flera <relationship links>. Kopplingen definieras genom de separata entity type förekomsterna COMPOUND FACT och DERIVATION ARGUMENT.
Composite	Vad vi kan förstå är detta en <attribute type> som har en definierad substruktur genom HAS COMPONENT sambandet. Se vidare avsnitt 4.1.

I modelleringssammanhang talar man ofta om härledning av attributvärden men det är sällan det formuleras med denna kombination av exakthet (pga kravet på exekverbarhet) och avancerad uttryckskraft.



## 5.3 RELATIONSHIP LINK

RELATIONSHIP LINK beskrivs bla genom

<b>name</b>	Namnet på en <relationship link>. Måste vara unikt för den <entity type> som beskrivs. (Nyckeln för en <relationship link> är systemgenererad.)
<b>description</b>	En verbal beskrivning.
<b>minimum_per_subject</b>	Minsta antalet förekomster av <relationship link> per förekomst av <entity type>. Exv 0 för <relationship link> <i>äger</i> från <entity type> <i>person</i> .
<b>maximum_per_subject</b>	Största antalet förekomster av <attribute type> per förekomst av <entity type>. Exv 5 för <relationship link> <i>äger</i> från <entity type> <i>person</i> . Observera att den generella beteckningen "flera" i form av M e dyl inte tycks accepteras. Värdet måste vara null eller ett positivt tal.
<b>arm_indicator</b>	Är direkt relaterad med aspekten <i>purpose</i> under ENTITY TYPE. Om <entity type> är klassad som associative och <i>arm_indicator</i> för aktuell <relationship link> är satt till Y(es) betyder det att den är ena sidan av en nu objektifierad <relationship type> och att det för samma <entity type> finns en ytterligare <relationship link> som står för "andra sidan". För <i>innehåller</i> och <i>avser</i> i figur 21 sätts <i>arm_indicator</i> till Y.

Även för RELATION LINK erbjuds möjligheten att specificera härledningsregler.

**Derivation\_formula** och **derivation\_formula\_language** fungerar här på samma sätt som hos ATTRIBUTE TYPE. Eftersom en link alltid har en direkt motsvarighet i en inverslink är det ännu oklart hur ena riktningens härledningsregel med säkerhet kan fås att ha en motsvarande korrekt invers härledningsregel i andra riktningen. Möjligtvis sköter aspekten *controlling* detta. Om denna sätts till Y(es) för en link betyder det att denna link är styrande inom <relationship type> och att den andra riktningen nödvändigtvis har *controlling* satt till N(o). Antagligen återfinns härledningsreglerna endast hos styrande links. Varför i så fall inte placera härledningsregeln under <relationship type>? Vi återkommer med klarare besked i någon senare rapport. Är för övrigt en härledningsregel, oavsett var den placeras alltid användbar för båda riktningarna?

## 5.4 RELATIONSHIP TYPE

RELATIONSHIP TYPE beskrivs bla genom

name	Namnet på en <relationship type>. Behövs verkligen detta namn? Det är dessutom obligatoriskt. Varje link har ju sitt eget namn. Tillsammans blir det tre namn på i sak samma företeelse!
description	En verbal beskrivning.
derivation_type	Här återkommer ett likartat syfte som för ATTRIBUTE TYPE, dvs att tala om en <relationship type> ska uppfattas som enkel eller härledbar.

Tre alternativ erbjuds.

Basic	Ingen härledning.
Computed	Innebär beräkning av befintliga kopplingar i enligt med villkoren i en algoritm. Den aktuella algoritmen finns i detta alternativ definierad under controlling <relationship link>'s derivation_formula.
Indirect	Korrekta kopplingar (paths) hämtas via en sammansättning av en eller flera <relationship links>. Kopplingen definieras genom de separata entity type förekomsterna COMPOUND FACT och DERIVATION ARGUMENT.

## IDENTIFIER

IDENTIFIER beskrivs bla genom

name	Namnet på en <identifier>. Måste vara unikt för den <entity type> som beskrivs.
description	En verbal beskrivning.
primary_indicator	En <entity type> kan ha flera alternativa identifierare. Sannolikt väljs en av dessa som den som primärt ska användas. Jmf primary och candidate keys i relationsmodellen. För primär <identifier> gäller att primary_indicator har värdet Y(es).

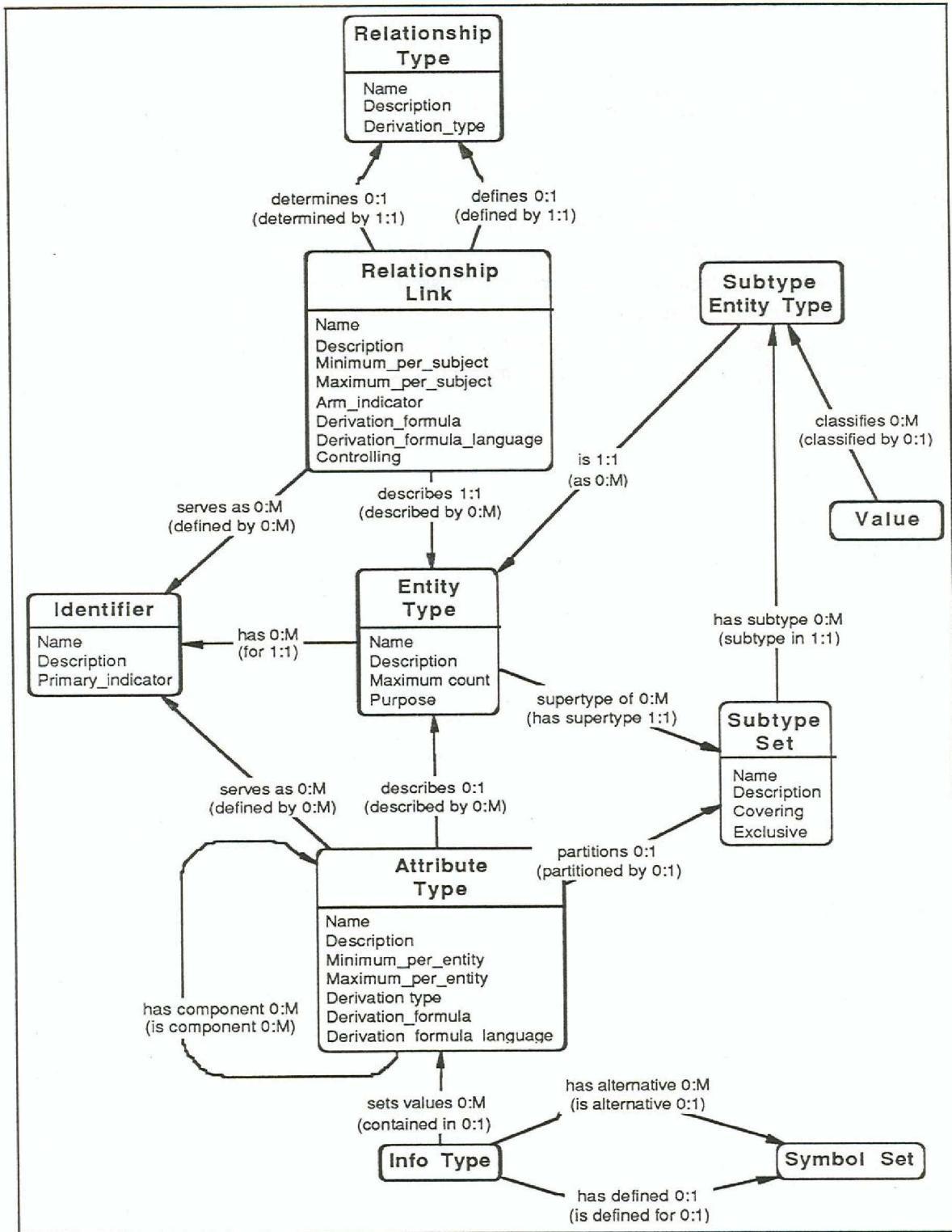
## 5.6 SUBTYPE SET

SUBTYPE SET beskrivs bla genom

<b>name</b>	Namnet på ett <subtype set>. Bör vara unikt för super-<entity type>.
<b>description</b>	En verbal beskrivning.
<b>covering</b>	<p>Om Y(es), betyder det att en förekomst av super-&lt;entity type&gt; måste förekomma även i någon av sub-&lt;entity types&gt;.</p> <p>N(o) ställer inget sådant krav.</p> <p>Covering svarar mot Telmods "uttömmande" respektive "ej uttömmande" specialisering. Indelningen av <i>person</i> i <i>man</i> och <i>kvinna</i> är uttömmande för &lt;subtype set&gt; <i>kön</i> medan indelning i <i>svensk</i> och <i>norrman</i> knappast kan anses uttömmande för &lt;subtype set&gt; <i>nationalitet</i>.</p>
<b>exclusive</b>	<p>Om Y(es), får en förekomst av super-&lt;entity type&gt; bara förekomma i högst en av sub-&lt;entity types&gt;. En <i>person</i> är antingen <i>man</i> eller <i>kvinna</i>, aldrig både och.</p> <p>N(o) ställer inget sådant krav. Har vi ett &lt;subtype set&gt; befattning är det inte osannolikt att en och samma <i>person</i> kan befinna sig i både sub-&lt;entity type&gt; <i>försäljningschef</i>, <i>ordermottagare</i> och <i>lagerförman</i>.</p>

## 6. Sammanfattning

För att hålla rapportens omfång inom rimliga gränser, har vi endast givit en översiktlig beskrivning av den del av ENT som hanterar begreppsmodeller. Rapporten gör alltså inte anspråk på att vara uttömmande. Vi måste också hela tiden hålla i minnet att ENT är under kontinuerlig vidareutveckling. Denna rapport refererar endast faciliteter i RM version 1, release 1. Förhoppningsvis har läsaren fått en allmän förståelse om uttryckskraften i denna del av ENT. Avslutningsvis presenterar vi i figur 23 återigen den diskuterade delen av schemat men nu inkluderande även de under avsnitt 5 presenterade aspekterna.



FIGUR 23.

## 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 TRIAD-projektet. Seminarier, rapporter och referensgruppsverksamhet är led i den verksamheten.