

TRIAD Arbetsrapport

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 Datamodelleringsbegreppen

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

TRIAD-projektet drivs gemensamt av Televerket, Posten, Statskontoret, Ericsson Data Services och SISU, Svenska Institutet för Systemutveckling.

För vidare information kontakta SISU Informationscentrum, tel 08-752 16 00, fax 08-752 68 00, Box 1250, 164 28 Kista.

Innehåll

IBM Repository Manager: Datamodelleringsbegreppen

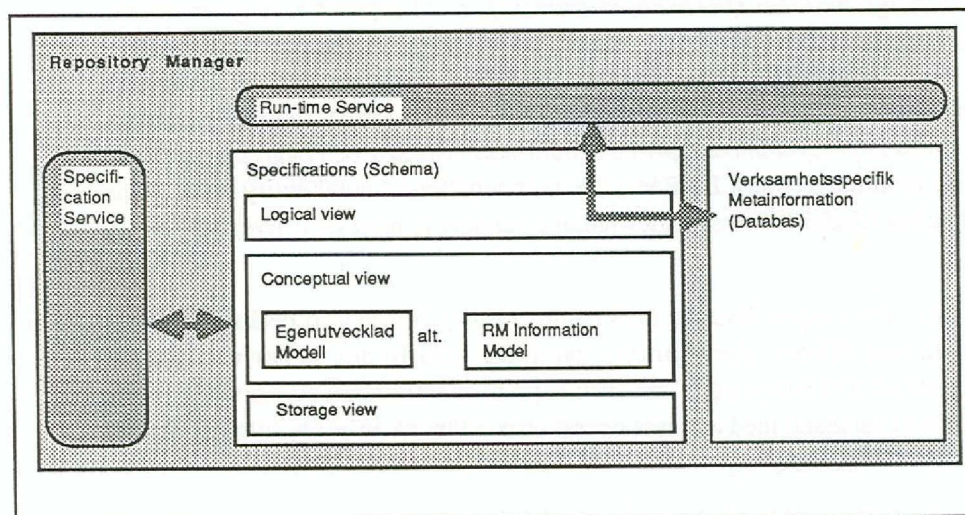
1. Inledning	1
2. Översikt över modelleringsbegreppen	2
2.1. Entity type, attribute type.....	2
2.2 Relationship type.....	4
2.3. Dependent och non-dependent entity types	7
2.4. Aggregation type	10
3. Överensstämmelse med modelleringsbegreppen i Information Model?	13
4. Sammanfattning	14

IBM Repository Manager: Datamodelleringsbegreppen

1. Inledning

En introduktion till Repository Manager (RM) ges i rapporten [Triad K nr 4] . Där identifieras ett antal specifika områden att titta på mer i detalj. Ett av dessa områden är den begreppsapparaten, som används i RM för att diskutera kring och uttrycka ett schema i RM. Begreppen kommer att användas flitigt bla i samband med beskrivning av RMs funktionalitet och vid genomgång av RMs Information Model. Det är därför viktigt att läsaren av kommande rapporter förstår dessa begrepp både till namn och innebörd.

Figur 1 indikerar att begreppsapparaten är en central del av RM som helhet (den ljusare gråtonen). En god förståelse av begreppen är inte minst central för att korrekt kunna tolka syntax och semantik hos gränssnittsspråken (den mörkare gråtonen).



FIGUR 1

I förhållande till IRDS-terminologi svarar RMs begreppsapparat mot nivå 4 (IRDS Definition Schema Level), dvs mot schemat i det tredje nivåparet (IRD Definition Level Pair). Se vidare rapporterna [Triad K 1 och 2].

2. Översikt över modelleringsbegreppen

Detta avsnitt introducerar de olika begreppen i RMs begreppsapparat. För att åskådliggöra hur de är relaterade till varandra använder vi en grafisk presentationsteknik enligt Telmod-notation, en notation framtagen i samarbete mellan Televerket och SISU. Allteftersom nya begrepp introduceras expanderas grafen. Till slut har vi på så vis åstadkommit en heltäckande begreppsmodell.

Obs, att sambandsnamnen inte är hämtade från IBM-dokument. De har figurerat i olika tappningar i lästa dokument och ska här uppfattas enbart som semantiska förtydliganden. Se alltså dessa grafer som begreppsmodeller, inte som exakta schemata för hantering av RMs schema.

Vi har valt att använda den av IBM använda engelska terminologin. I konsekvensens namn har vi använt engelska begrepp även för sambanden.

2.1. Entity type, attribute type

Begreppet *entity* förekommer i repositorysammanhang exv som beteckning på en viss programmodul (*beräkna radsumma*), en viss process (*registrera ny kund*), ett visst informationsbehov (*beställda artiklar*), ett visst begrepp (*order*). Entiteter som har tillräckliga likheter (subjektiv bedömning) abstraheras/klassificeras till en gemensam typ, en viss **entity type**. Exempel på entity types är *programmodul*, *process*, *informationsbehov* och *begrepp* (hämtade ut första meningen ovan). Ett RM-schema innehåller ett antal entity types. RMs Information Model tex, innehåller redan i version 1, release 1, över 100 entity types.

För entity types anges, förutom dess namn (*Name*), bla en kort beskrivningstext på några ord (*Short description*) samt en utförligare beskrivningstext (*Description*). Till entity types kan även kopplas specifikation av aktiviteter, som ska utföras i samband med att vissa operationer utförs på en entity tillhörig aktuell entity type (*Trigger policy*).

Entities beskrivs med hjälp av **attributes** (exv *beräkna radsummas 'antal rader kod'*). På typnivån får vi **attribute type** (exv *programmoduls 'antal rader kod'*). En attribute type är definitionsmässigt alltid lierad med viss entity type. I RMs Information Model förekommer regelmässigt ca 4 - 10 attribute types per entity type.

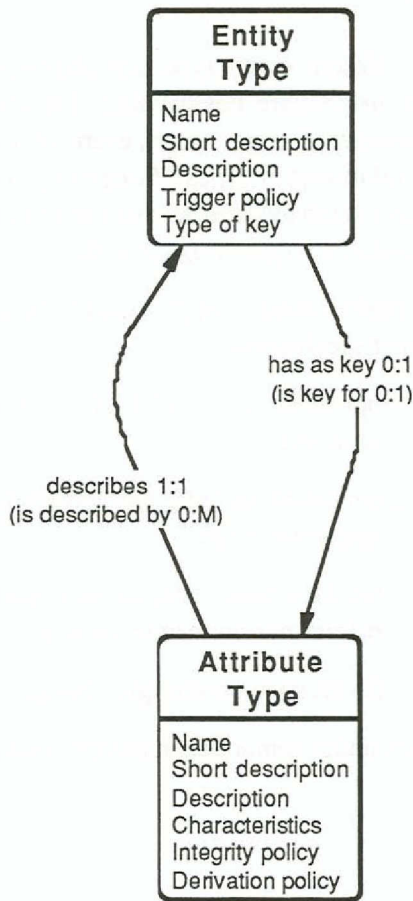
Attribute types har också ett namn, en kort beskrivning och en utförligare beskrivning. Därutöver anges ett standardformat (*Characteristics*), villkor för acceptabla värden (*Integrity policy*) samt härledningsregler i form av algoritmer för beräkning av värden (*Derivation policies*).

En viss entity inom en entity type identifieras alltid via en och endast en attribute type. Två alternativ står till buds. Finns en lämplig attribute type bland de som "modellören" definierat, tas detta. I annat fall skapar RM en internt genererad attribute type för detta specifika ändamål (jmf begreppen internal, surrogate mfl i litteraturen). Den går under beteckningen SYSKEY.

Beteckningen för dylika identifierare är key attribute type. För entity type *programmodul* kanske attribute type *modulbeteckning* kan användas medan en entity type *informationsbehov* kanske unikt kan refereras till endast genom en kombination av dess namn och namnet på den process som initierar behovet. För *informationsbehov* måste därför en RM-genererad attribute type skapas.

Kravet på enkla key attributes är ingen modelleringsteknisk finess utan sannolikt en eftergift för enkelhet i den interna databaslösningen.

Den grafiska beskrivningen återfinns i figur 2.



FIGUR 2

Relationship type

Relationship types sammanbinder två typer av företeelser. I normalfallet sammanbinds två entity types (se dock nedan för alternativ). En relationship type anger alltid sambandet som riktat, dvs det finns en startpunkt (**source**) och en målpunkt (**target**). Generellt sett kan ju ett samband alltid ses ur båda inblandade entity types' perspektiv. Vilken entity type som ska vara source och vilken som ska vara target är helt och hållet upp till modellören. För fullständighets skull namnges båda perspektiven, men med distinktionen att den ena riktningen är huvudriktning (**primary**) och den andra inversriktning (**inverse**). Huvudriktningen är perspektivet hos source.

I vissa ER-ansatser tillåts n-ställiga samband. Som framgått ovan definierar RMs version av relationship types endast binära samband.

RMs Information Model release 1, version 1, innehåller ca 260 relationship types. En i sanning komplex modell!

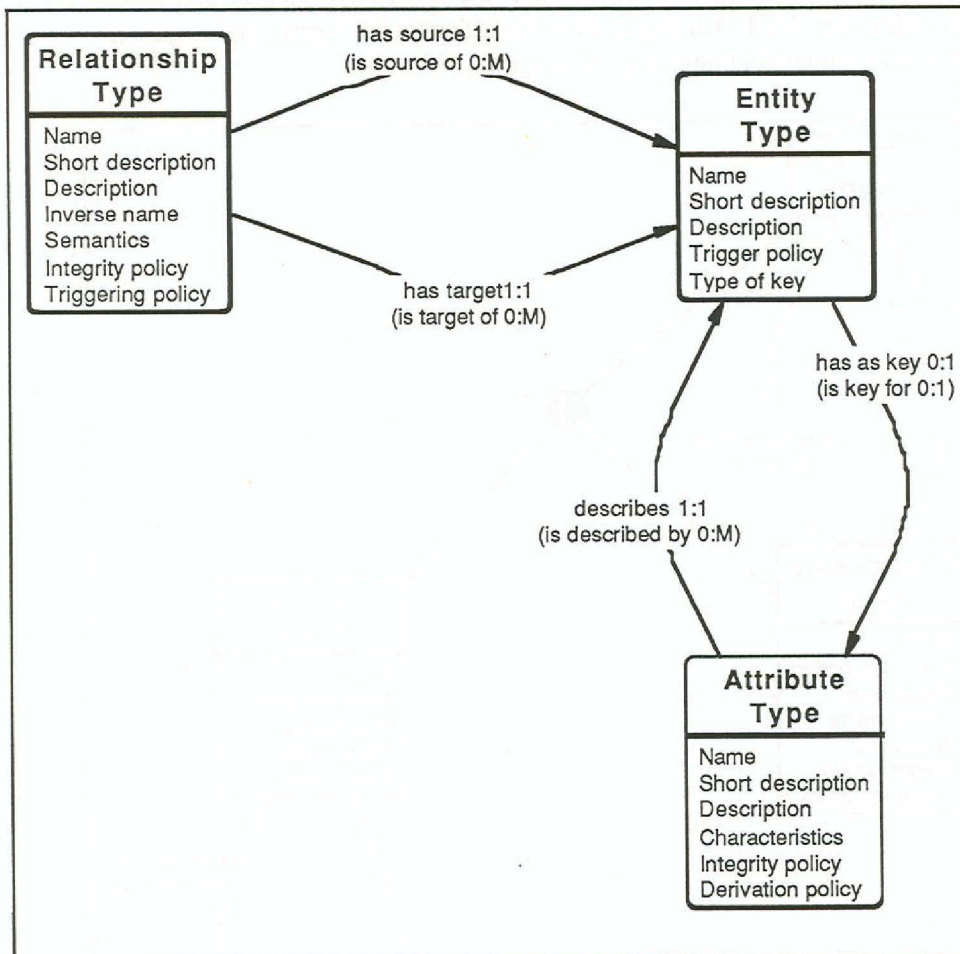
En relationship type kännetecknas, liksom entity type och attribute type, av namn, en kort beskrivning och en utförligare beskrivning. Därtill kommer specifikation av namnet på inversriktningen (*Inverse name*), eventuella villkor för exv acceptabla kombinationer av relationship types (*Integrity policy*) samt eventuella aktiviteter, som ska utföras i samband med att vissa operationer utförs på en relationship tillhörig aktuell relationship type (*Trigger policy*).

Sist, men inte minst, har vi den beskrivning som går under beteckningen *semantics*. Där kan följande egenskaper specificeras:

- *cardinality* om det är ett 1:1, 1:M, M:1 eller M:M samband
- *mandatory* (för varje riktning) om sambandet måste finnas för source respektive target
- *controlling* (för varje riktning) om även source respektive target ska raderas när sambandet raderas
- *ordered* för att styra sorteringsordning.

Ytterligare en aspekt kan specificeras under semantics, nämligen *owning*. Dess innebörd beskrivs i nästa avsnitt.

Begreppsmodellen utvidgas enligt figur 3.



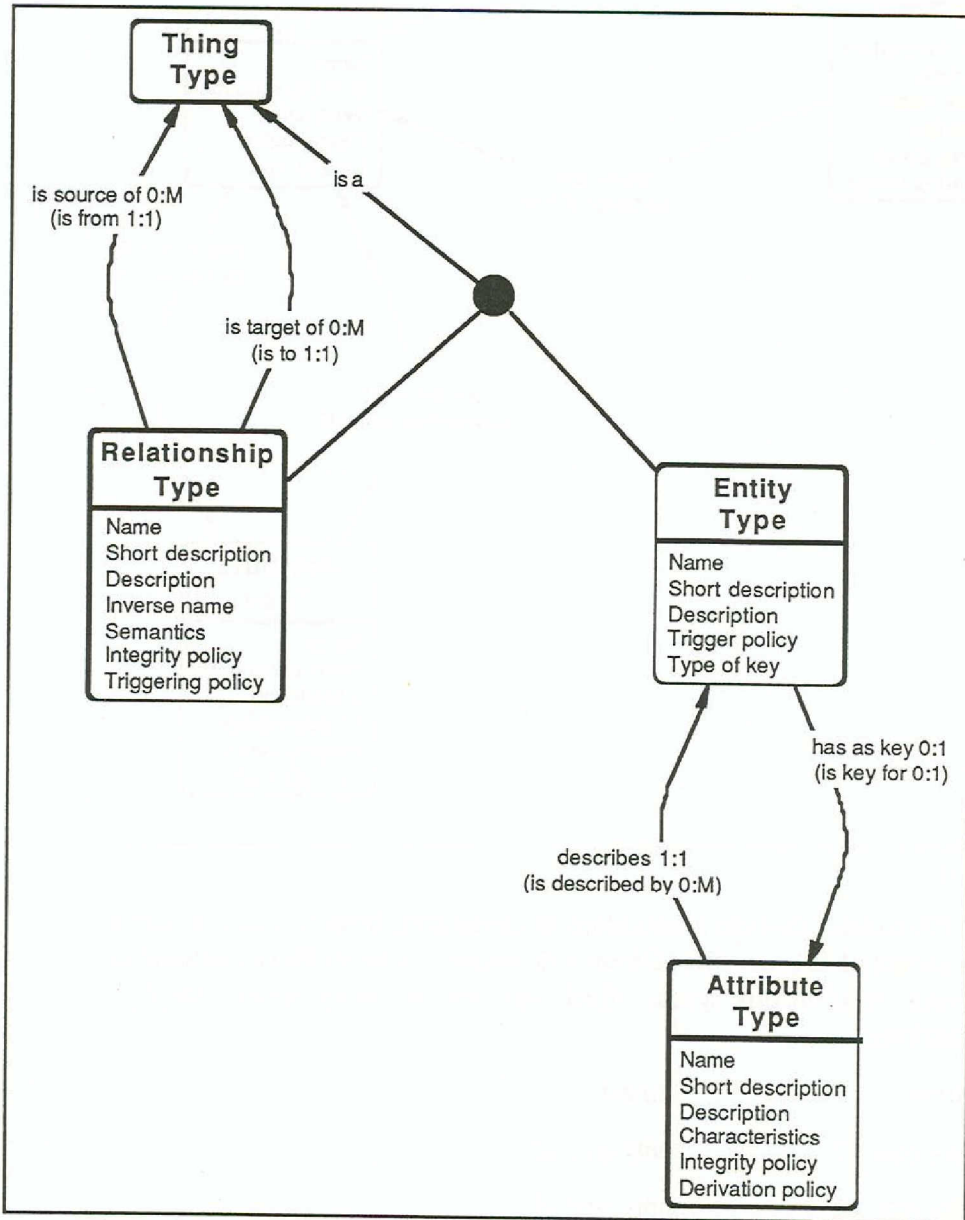
FIGUR 3

I vissa ER-ansatser får även relationships ha egenskaper. Detta är dock inte tillåtet i RMs variant på ER-modellen. Däremot får source och target för en relationship type vara relationship types, alltså inte enbart entity types. Möjliga kombinationer:

Source	Target
entity type e1	entity type e2
entity type e1	entity type e1
entity type e1	relationship type r1
relationship type r1	entity type e1
relationship type r1	relationship type r2
relationship type r1	relationship type r1

Som synes tillåts även självreferenser.

Den grafiska modellen behöver modifieras för att återspegla detta nya faktum. **Thing type** har tillkommit som en generalisering av relationship type och entity type. En thing type är alltså antingen eller. Förhållandet indikeras i grafen genom en liten rund sammanbindningsring. Se figur 4.



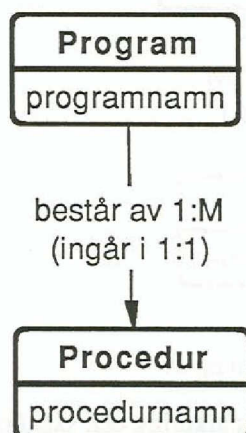
FIGUR 4

2.3. Dependent och non-dependent entity types

(Den som endast vill bilda sig en översiktlig uppfattning om RMs modelleringsspråk, kan hoppa över detta avsnitt.)

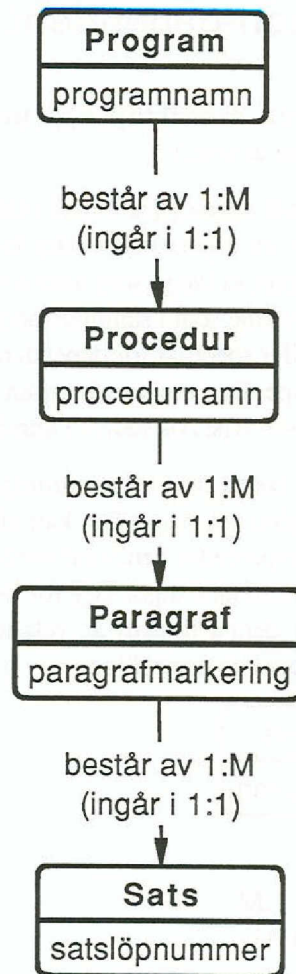
När en relationship type indikerats som *owning* (se ovan) betyder det att dess target är en speciell typ av entity type, en **dependent entity type**. Innebörden av detta är bl.a. att den endast kan existera så länge som dess owner relationship type existerar. Owner kallas den entity type som i sammanhanget figurerar som source. Förutom existensberoendet gäller specifikt för dependent entity types att de inte har någon egen key attribute type. Den unika referensen sammansätts av en "egen" attribute type samt key attribute type hos source i kombination.

Som exempel kan vi ta program som består av en uppsättning procedurer. En procedur har bara unikt namn inom givet program. Det kan alltså finnas flera program som har en egen version av en procedur benämnd *beräkna lagervärde*. Programnamn och procedurnamn tillsammans utgör key för dependent entity type *procedur*. Automatiskt följer av denna filosofi att relationship type har semantics enligt 1:1 eller 1:M, mandatory och controlling.



FIGUR 5

Det är inget som hindrar att en dependent entity type i en relationship type samtidigt kan vara source och owner i en annan relationship type. På så vis går det att bygga upp valfritt djupa hierarkier av dependent entity types. Vi kan exempelvis bygga på exemplet ovan med indelningen av procedurer i paragrafer och indelningen av paragrafer i satser.



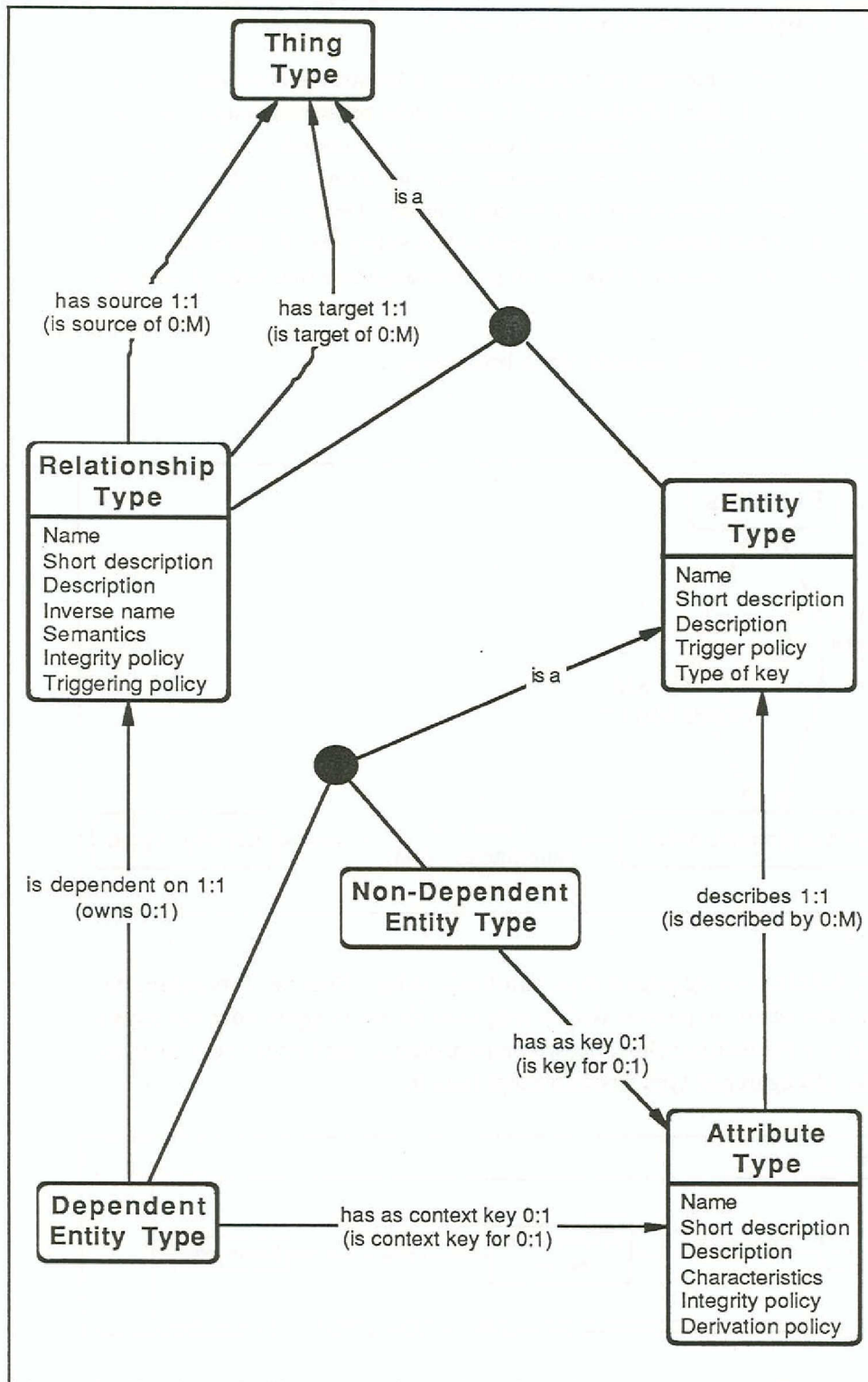
FIGUR 6

En *sats* får en nyckel som är en sammansättning av

programnamn, procedurnamn, paragrafmarkering och satslöpnummer.

Behovet av dependent entity types känns oklar. Man gör avsteg från principen om ett key attribute för en entity type. Vid behov kan ju systemet generera lämplig nyckel (SYSKEY). Controlling- och mandatory-aspekterna under semantics (se ovan) klarar existensvillkoren. Varför inte generera SYSKEYs för procedur, paragraf och sats ovan?

I alla händelser uttrycker vi denna aspekt grafiskt i figur 7.



FIGUR 7

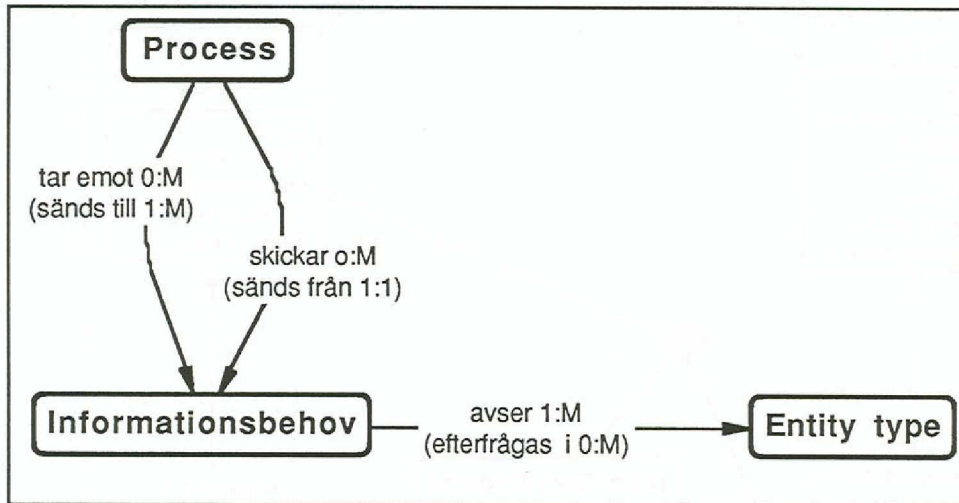
2.4. Aggregation type

(Den som endast vill bilda sig en översiktlig uppfattning om RMs modelleringsspråk, kan hoppa över detta avsnitt.)

Som ett tillägg till de normala ER-begreppen finns en konstruktion som kan liknas vid en enklare typ av så kallat komplext objekt. Den går under benämningen *aggregate type*. En aggregate type består av en uppsättning entity types och relationship types ordnade i en hierarki med en entity type som startpunkt eller *root*. Från root utgår en eller flera relationship types. Hierarkin kan alltså ha valfri "bredd". Deras target kan sedan vara source för en ny uppsättning relationship types och så vidare ner till valfritt djup. Varje gren i form av en relationship type har en positionsangivelse som anger dess plats i trädet.

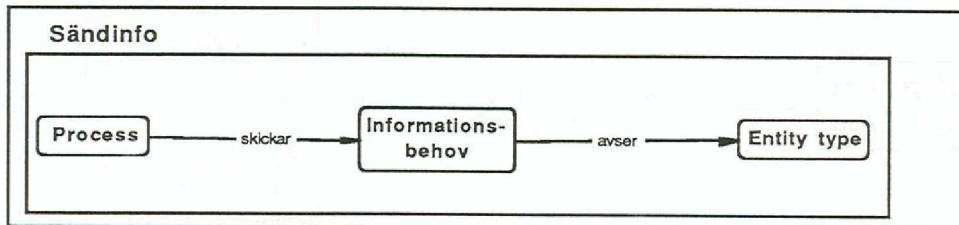
Varje aggregate type åsätts ett namn och en beskrivning.

Antag en modell enligt figur 8.



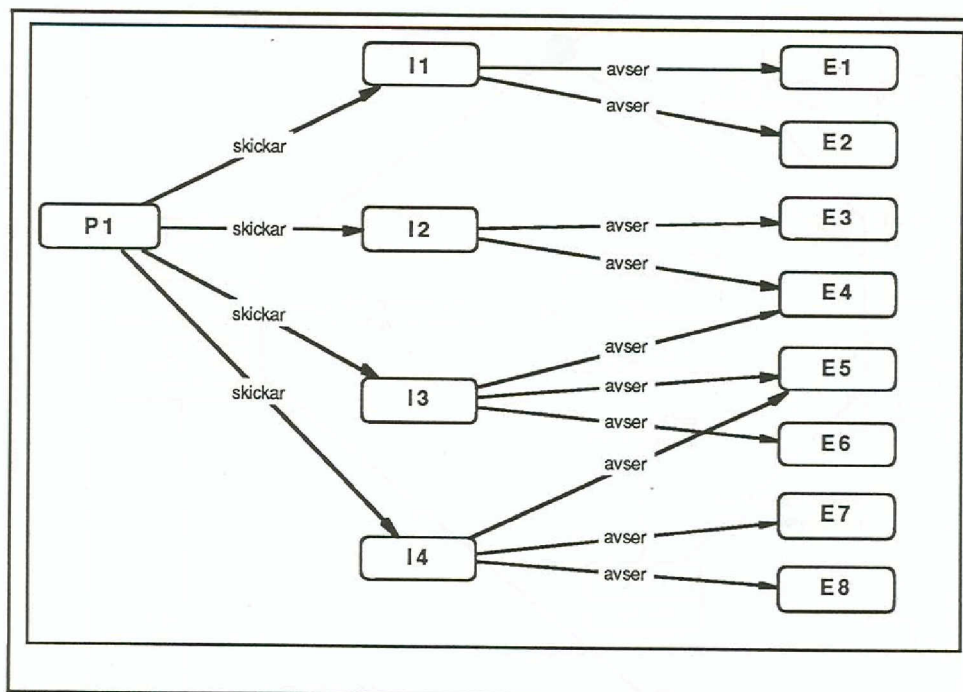
FIGUR 8

Vi vill definiera en aggregate type *sändinfo* enligt följande. Processen, de informationsbehov en process skickar iväg samt de entity types, som refereras i respektive informationsbehov, utgör komponenterna. Med entity type *process* som root blir aggregate type *sändinfo* enligt figur 9.



FIGUR 9

Det primära användningsområdet för aggregate types tycks vara att erbjuda ett högnivåbegrepp för låsningsändamål. Låsning begärs på ett visst aggregate, dvs en förekomst av en aggregate type. En sådan förekomst får sitt innehåll och avgränsning genom en viss entity tillhörig root entity type. Begär vi låsning av sändinfo med root-process P1 blir följden låsning av samtliga lierade entities till P1 enligt definitionen av sändinfo. Exempel ges i figur 10.

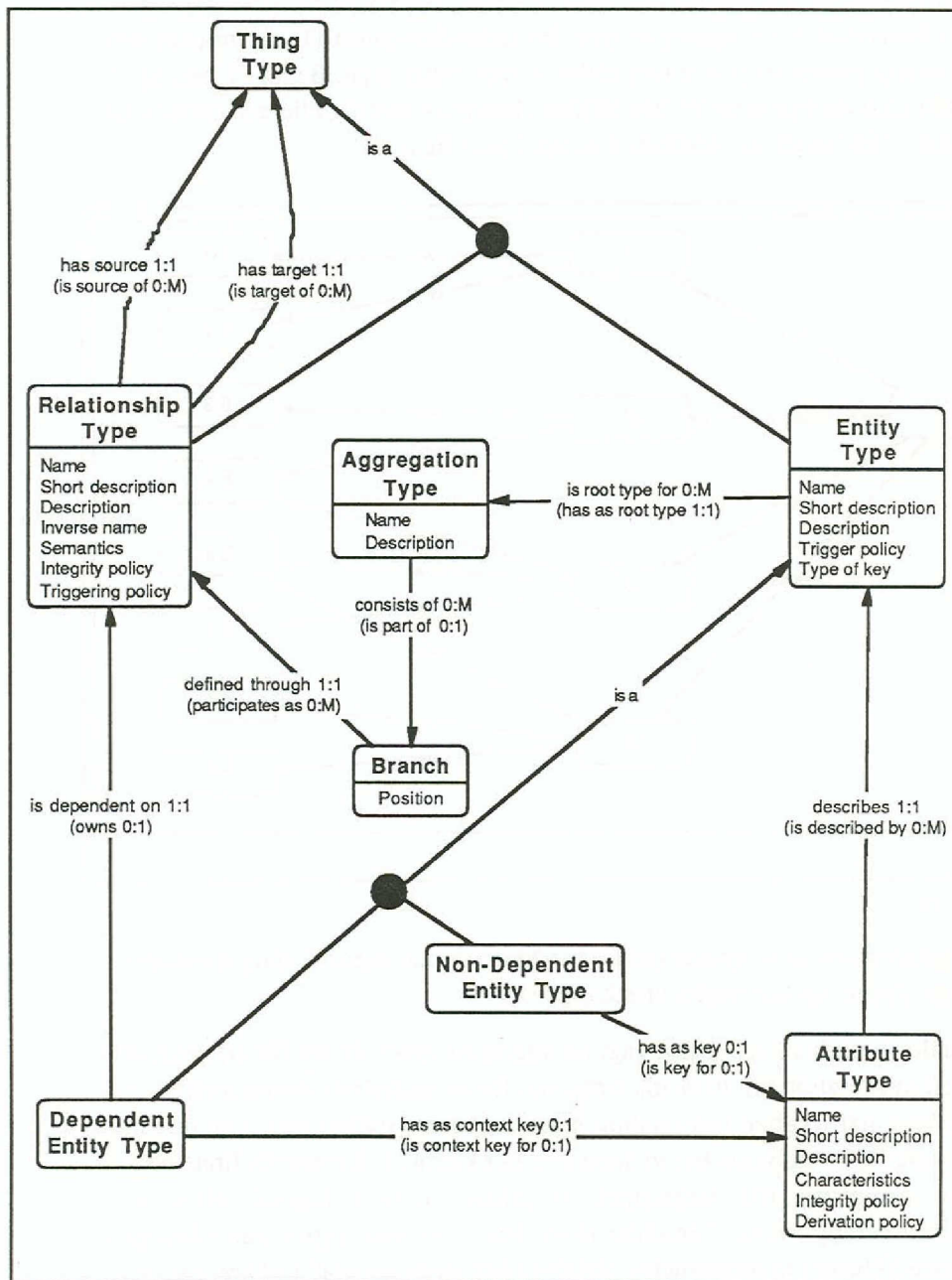


FIGUR 10

Låsningsalternativ är *noupdate* (andra har åtkomst men ingen låsningsmöjlighet) eller *add, delete, update* (andra utestängs helt).

I manualer beskrivs även något vagt att aggregate types är bra för att gruppera relaterad information för att skapa rapporter, för att underlätta dialog mellan RM och verktyg när likartad information behövs på flera ställen mm. Om och hur detta i praktiken realiserats har vi ännu inte fått klarhet i. Eventuellt finns någon typ av koppling till begreppet *template tree* i logical view (se introduktionsrapporten). Template trees och aggregate types har nämligen påfallande likheter i uppbyggnaden. Behövs båda företeelsema? Det kan vara värt att notera att Enterprise Submodel i RMs Information Mode, release 1, version 1, inte innehåller några aggregate types. Övriga sub-modeller omfattar totalt ca 20 aggregate types.

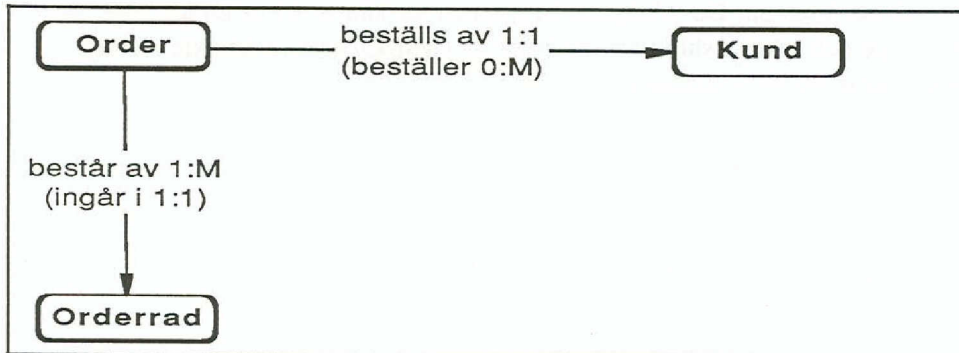
Den utvidgade begreppsmodellen visas i figur 11.



FIGUR 11

3. Överensstämmelse med modelleringsbegreppen i Information Model?

En central del av Enterprise submodel i RMs Information Model är den begreppsapparat som används för att skapa verksamhetsspecifika informationsmodeller. Mycket enkelt exempel visas i figur 12.



FIGUR 12

Den begreppsapparat vi gått igenom i denna rapport har samma syfte, men hanterar en annan typ av informationsmodeller. Se exv figur 7 och 9. Utförlig diskussion kring modellnivåer återfinns i rapport [Triad K nr 2]. Man kan tycka att det ligger nära till hands att använda samma eller åtminstone en snarlik ansats för båda behoven. Så är dock inte fallet. Skillnaderna är markanta. Varför IBM valt denna strategi är oklart. En komplicerande faktor är också att delvis samma begrepp återkommer på de båda nivåerna men med inte säkert samma betydelse. Risken för missuppfattningar är stor.

Helt distinkta begrepp på respektive nivå men med parvis överensstämmelse i innebörd skulle ha underlättat kunskapsinhämtning.

För den som tagit del av Enterprise submodel (ENT), tex i [Triad K nr 6], kan några exempel på skillnader vara av intresse. För ENT gäller specifikt:

- Relationship type har en mer detaljerad definition genom uppdelning i relationship type (sambandet som sådant) och relationship link (respektive riktning).
- Identifiering av entity types kan specificeras med valfri kombination av attribute types och relationship links.
- Relationship types sammanbinder endast entity types.
- Specialiseringsstrukturer kan skapas. Attribute types kan ha en intern struktur.
- Policies finns inte i samma generella form, däremot möjligheter till härledningsspecifikationer.

4. Sammanfattning

Denna rapport beskriver den terminologi/begreppsapparat som används vid formulering och förklaring av innehållet i RM-scheman, dvs när vi behöver prata om de företeelser som är komponenter i ett repository-schema på en generell nivå. Läsare av andra rapporter, manualer eller artiklar om RM kommer ofta att stöta på dessa begrepp. De viktigaste begreppen att känna till är entity type, attribute type och relationship type, de typer av egenskaper som karakteriserar dem samt deras inbördes samband.

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 tårtpincip "betala för en tårbit, men ät hela tårten", 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 borde 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.