

Krav på IA

Modellering

Utbildning

Katalogprinciper

Uttagssystem

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

Rapport K nr 7: IBM Repository Manager: Attribut- och värdemodellering i Enterprise Submodel

Rapport K nr 8: Navigering i Repository

Rapport K nr 9: TRIAD Newsletter – IRDS inom ISO. Dagsläget

Rapport K nr 10: TRIAD Newsletter –ISO/IRDS. Händelseutvecklingen 91/92

Rapport K nr 11: Samverkan mellan resurskataloger – visioner eller behov

Rapport K nr 12: AD/Cycle I Information Model – Processer och informationsflöden mellan processer

Rapport K nr 13: AD/Cycle I Information Model – Info Flows inom Processmodellen

Rapport K nr 14: AD/Cycle I Information Model – Relationsdatabasmodellering

Rapport K nr 15: AD/Cycle I Information Model – Härledningsspecifikationer i begreppsmodellen

Rapport K nr 16: IA-prototyp

IA-prototyp

Björn Norén, Televerket Nätdivisionen

Spridningsförbehåll:

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

Rapporten är skriven i och för TRIAD delprojekt Katalogprinciper.

1 Sammanfattning Lägesrapport

Division Nätjänsters krav på modelleringsbegrepp har realiserats i en Oracle databas med enkla in-utmatningsbilder enligt delaktivitet 1. Funktionaliteten har testats med testdata. Principerna fungerade i stort.

Underlag till delaktivitet 2, "IA-arbetsätt" kom inte fram från TRIAD's kravgrupp och delaktiviteten kom inte igång . Däremot har Televerkett nu kommit fram till ett första förslag till "IA-rollspel". Delaktivitet 3 är fortfarande ej aktuell pga resursbrist. Ett mindre försök med att göra en brygga från ADW har gjorts. Detta misslyckades p.g.a. för låg resursinsats. Delaktiviteten har övergått till 523, HYBRIA IA/DA som delaktivitet. Aktiviteten 523 kan ses som en fortsättning på denna aktivitet.

2 Bakgrund

2.1 Allmänt

Behovet av TRIAD-aktiviteten härrör utifrån Division Nätjänsters målmodell för IA-funktionen från slutet av 1990. En förenklad modell visas i figur 1 nedan. Målen utgår från IP90. De bryts ned till olika delmål eller åtgärder.

En av åtgärderna var att utarbeta en standard för begreppsdefinitioner. Standarden togs fram genom att vidareutveckla IA-projektets referensmodell. Vid utarbetandet av standarden användes modellering och resultatet presenterades bland annat i form av en konceptuell datamodell. Den utgör en del av en kravspecifikation för ett datakatalogsystem och att den kan verifieras praktiskt mha en prototyp.

Ett datakatalogsystem kan ses som ett konventionellt informationssystem. Då kan också konventionell systemutvecklingsmetodik användas. Riktigt konventionellt är det kanske inte, men man bör för den skull inte frågå traditionell systemering.

2.1 Första förslaget på en prototyp

De första idéerna på en prototyp kom upp som ett aktivitetsförslag i uppdrag-specifikationen till NSTÖD-IA. NSTÖD var ett strategiprojekt för Division NätTjänster som pågick våren 1991. Aktiviteten kom inte att ingå i NSTÖD-utredningen. Aktivitetsförslagen redovisas nedan som bakgrundsmaterial till TRIAD-aktiviteten.

2.1.1 Mål

De preliminära kvalitetskrav på modelleringsbegrepp som finns konkretiseras genom realisering i en datakatalogprototyp. Prototypmiljö Oracle enligt SKAL's¹ rekommendationer. Denna ger en konventionell 4GL/RDB-miljö utan avancerade grafiska gränssnitt. Syftet med prototypen är att få praktiska erfarenheter och återföra dessa till "IA-referensmodell", krav på datakatalogers funktionalitet och krav på systemutvecklingsverktyg.

¹ SKAL - Standard och Kvalitet vid Applikationsutveckling Landsomfattande - var ett projekt som arbetade på uppdrag av I-enheten i syfte att ta fram riktlinjer och standarder för utveckling och förvaltning av informationssystem inom Televerket.

2.1.2 Aktivitetsförslag

Prototyparbetet kan delas in i olika aktiviteter där aktivitet 1 är en grundaktivitet.

Aktivitet 1

Oracle finns för olika datormiljöer så att den första testen kan ske i en Mac enanvändarmiljö. Inmatning mha enkla hypercardprogram. Utmatning via det grafiska gränssnittet i Hybris. SQL används för specialoperationer. I samband med denna aktivitet kan datakatalogens grundfunktionalitet specificeras.

Aktivitet 2

Begrepp som kommer fram i aktiviteten "framtagningen av övergripande begrepp inom TeleNät" kommer att ligga i Business Modeler.

Aktiviteten består i att prova överföring av begrepp från Business Modeler till Oracle/RDB.

Aktivitet 3

Många begrepp är i dag definierade i Ramatic. Aktiviteten består i att prova överföring av begrepp från Ramatic till Oracle/RDB.

Aktivitet 4

CIA's (Centrala Informationsadministrationsens) begrepp ligger i DEFT och benämns SKATT.

Aktiviteten består i att prova överföring av begrepp från DEFT till Oracle/RDB.

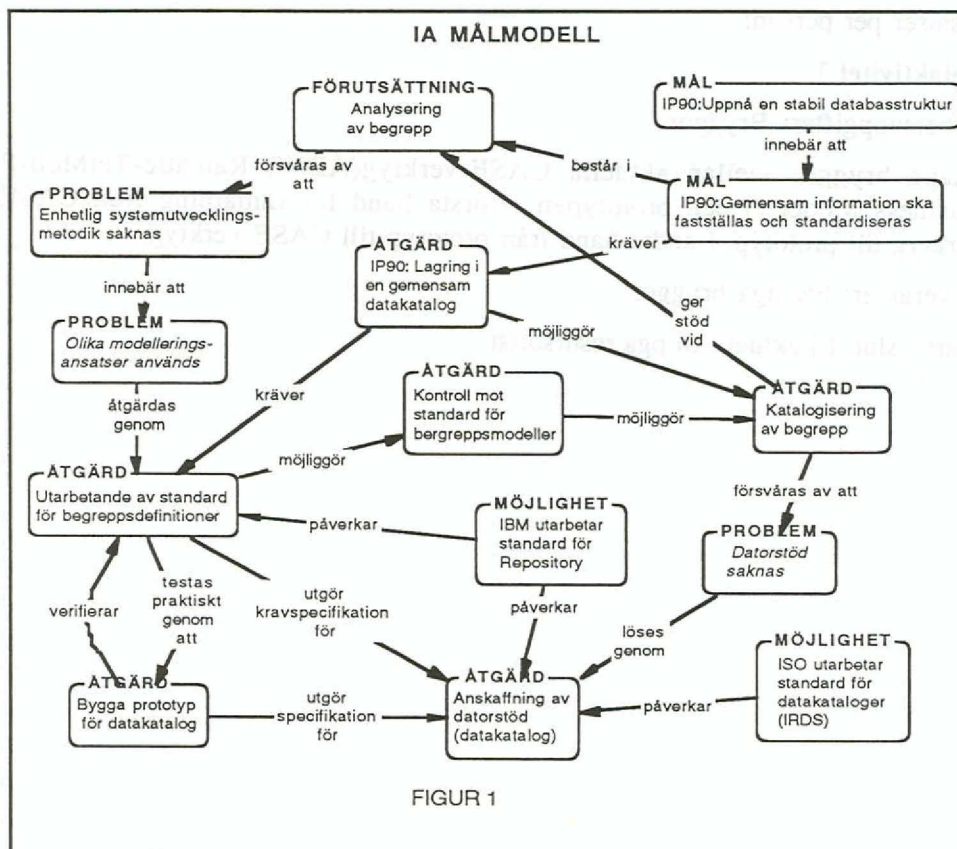
Aktivitet 5

Överflyttning till fleranvändarmiljö åtkomlig för lokala IA-funktioner och systemerare inom TeleNäts intresseområde. Detta ställer krav på den framtida systemstrukturen som behandlas i NSTÖD-SY.

Aktivitet 6

För TMOS används metoden ObjectOry. Aktiviteten består i att prova överföring av begrepp från ObjectOry till Oracle/RDB.

4 Aktivitetsplan, Tvt's IA-Prototyp



Aktivitetsidentifikation: 512a

Aktivitetsrubrik: IA prototyp

Delprojekttillhörighet: Katalogprinciper

Mål, nyttoeffekter:

Få en praktisk test av IA's arbetssätt mha en prototyp för att åstadkomma en specifikation av ett datakatalogsystem för stöd åt IA.

Arbetets bedrivande

Delaktivitet 1

Arbetsuppgifter: Prototyp.

Realisering av Tvt/Nät's IA-funktions krav på begreppsmodeller i en Oracle DB. Skapa enkla inmatningsbilder för inmatning av en begreppsmodell. Därefter lägga in testdata och undersöka behov av olika rapportuttag mha SQL.

Leverabler: Enkel prototyp enligt ovan.

Start-slut: V9107 - V9114

Delaktivitet 2

Arbetsuppgifter: IA-arbetsätt.

Beskriva olika scenarior för att komma fram till IA's arbetsätt. Införa och testa prototypen i verkliga projekt.

Leverabler: Arbetsmodell för IA. Kravspecifikation på datorstöd.

Start - slut: Ej uppstartat pga resursbrist.

Insatser per person:

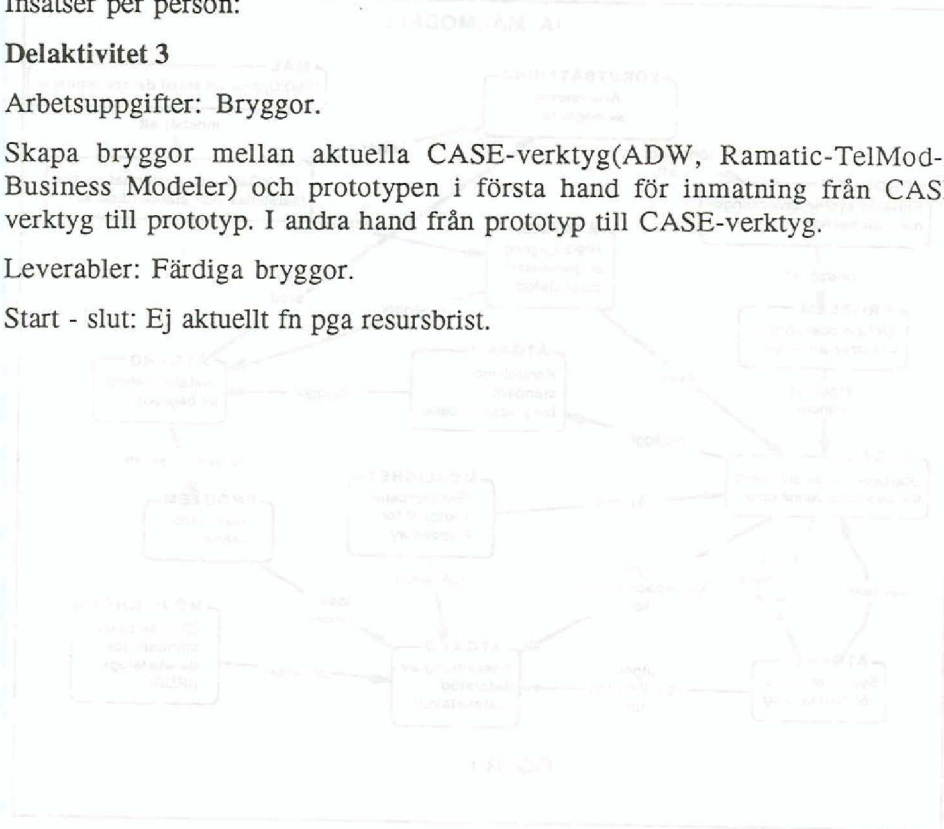
Delaktivitet 3

Arbetsuppgifter: Bryggor.

Skapa bryggor mellan aktuella CASE-verktyg(ADW, Ramatic-TelMod-B, Business Modeler) och prototypen i första hand för inmatning från CASE-verktyg till prototyp. I andra hand från prototyp till CASE-verktyg.

Leverabler: Färdiga bryggor.

Start - slut: Ej aktuellt fn pga resursbrist.



5 Aktivitetsplan, Postens IA-Prototyp

Posten har en annan referensmodell och avsåg att även testa den mha en prototyp. Det generella användningen (systemfunktionerna) ansågs dock vara lika.

Utdrag ur Postens aktivitetsbeskrivning:

Aktivitetsidentifikation: 512a

Mål - Nytt

- ge DA's centrala stödgrupp en interimistisk resurskatalog för informationsbeskrivningar och datamodeller.
- ge DA inom ansvarsområde/projekt en resurskatalog för registerbeskrivningar och realiseringsmodeller.

Avgränsningar enligt MIPS-SU.

5.1 Exempel på utsökningar från IA-prototyp

Samtidigt med detta arbete pågick en aktivitet inom Tvt med syfte att fastställa hur olika IA-funktioner skulle arbeta. Denna aktivitet slutfördes aldrig (Den slutfördes under 1992 under arbetsnamnet "IA-rollspel"), ej heller TRIAD's kravgrupps motsvarande aktivitet., ROA Roller soch ansvar.

Någon fullständig funktionsmodell har därför inte gjorts upp. Det finns bara några personliga tankar om vilka exempel på utsökningar som ska vara möjliga att utföra.

Resultatet av utsökningarna ska kunna fås i en relationsdatabastabell eller på en lista.

Det ska finnas möjlighet att:

- mata in, och hålla isär, begreppsmodeller från olika projekt med olika revisionslägen.
- lagra gemensamt definierade begrepp(CIA-begrepp) separat men i samma datorstöd som olika begreppsmodeller från olika systemutvecklingsprojekt.
- lista enskilt objekt med valbart antal egenskaper, t ex attribut (egenskaper och relationer), domäner, beskrivningar, etc.
- lista en modells innehåll med valbara egenskaper.
- söka efter samma namn på objekt, attribut, etc i olika modeller för att hitta de som är gemensamma i olika begreppsmodeller. Även möjlighet att söka på synonymer.

Ger underlag för eventuell samordning.

- kunna ha olika objektnamn för samma objekt i olika modeller och kunna se vilka objekt i en modell som *motsvaras av* ett objekt i en annan modell.
- få fram vad en äldre versions objekt *motsvaras av* i en senare versions.
- söka på vad ett ej specialiserat objekt i en modell *motsvaras av* ett specialiserat i en annan modell.

Skall detta även gälla för attribut?? (finns ej med i Tvt/Näts krav!).

- lista alla objektnamn och i vilken modell de finns och i vilken status objektet är.

6 Konceptuell datamodell över Tvt's IA-prototyp

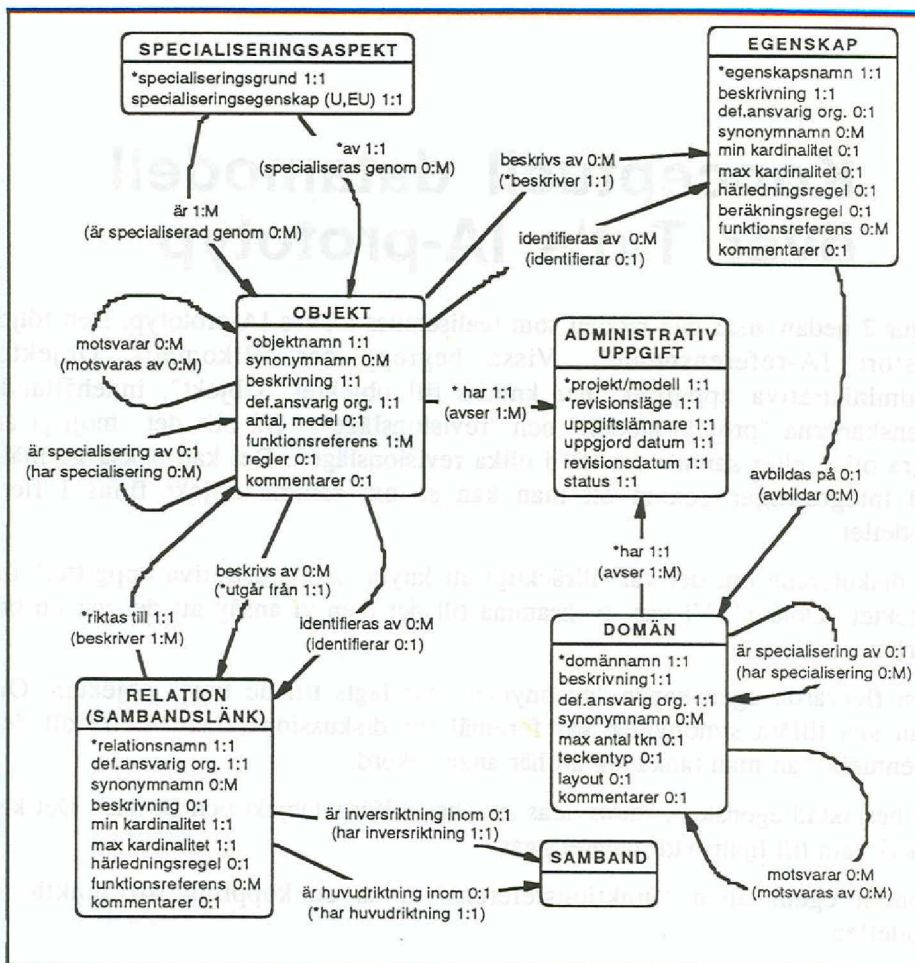
Figur 2 nedan visar den modell som realiserades i Tvt's IA-prototyp. Den följer i stort IA-referensmodell. Vissa begrepp har tillkommit. Objektet "Administrativa uppgifter" har knutits till objektet "Objekt", innehållande egenskaperna "projekt/modell" och "revisionsläge". Då blir det möjligt att lagra olika eller samma modell i olika revisionslägen. Det kan också ge hjälp vid integreringar genom att man kan se om samma objekt finns i flera modeller.

Vi diskuterade om det var tillräckligt att knyta "Administrativa uppgifter" till objektet "Objekt". Vi var tveksamma till det men vi ansåg att det var en bra början.

Den flervärda egenskapen "synonymer" har lagts till de flesta objekten. Om man ska tillåta synonymer var föremål för diskussioner men vi behöll det. Eventuellt kan man tänka sig att här ange sökord.

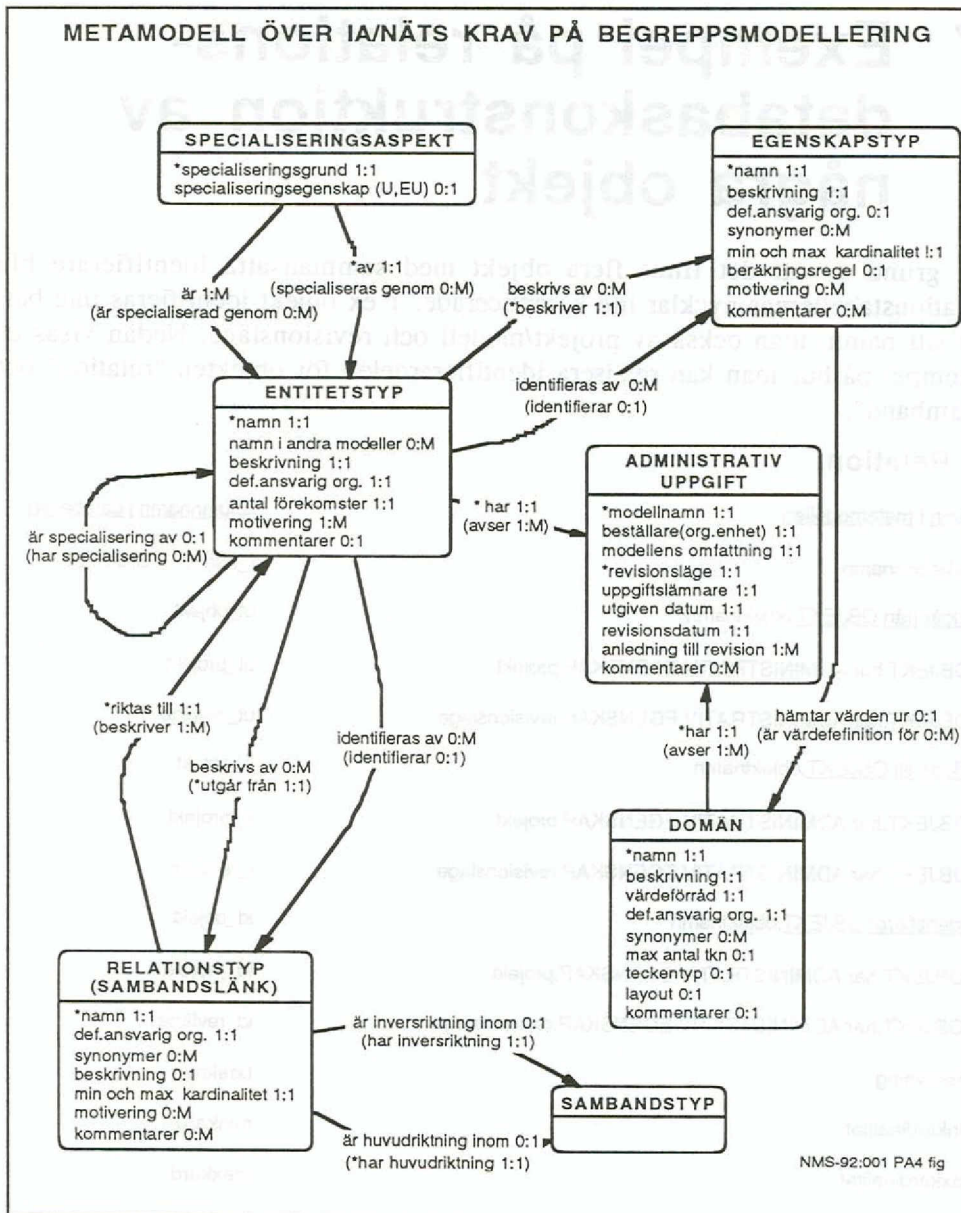
En besläktad egenskap, "motsvaras av", har tillförts Objekt och Domän. Det kan också vara till hjälp vid integreringar.

Genom egenskapen "funktionsreferens" finns en koppling till funktionsmodellen.



FIGUR 2

Våra krav enligt metamodellen i figur 2 har reviderats under senaste året och för information visas den reviderade metamodellen över våra krav i figur 3 nedan.



FIGUR 3

7 Exempel på relationsdatabaskonstruktion av några objekt

På grund av att det finns flera objekt med sammansatta identifierare blir relationstabellernas nycklar lite komplicerade. T ex objekt identifieras inte bara av sitt namn, utan också av projekt/modell och revisionsläge. Nedan visas ett exempel på hur man kan realisera identifierardelen för objekten "relation" och "samband".

Relation	
<u>Namn i metamodellen</u>	<u>kolumnnamn i databasen</u>
* relationsnamn	r_namn CHAR 32
* <u>utgår från</u> OBJEKT.objektnamn	ut_objekt
* OBJEKT.har ADMINISTRATIV EGENSKAP.projekt	ut_projekt
* OBJEKT.har ADMINISTRATIV EGENSKAP.revisionsläge	ut_revläge
* <u>riktas till</u> OBJEKT.objektnamn	ri_objekt
* OBJEKT.har ADMINISTRATIV EGENSKAP.projekt	ri_projekt
* OBJEKT.har ADMINISTRATIV EGENSKAP.revisionsläge	i_revläge
<u>identifierar</u> OBJEKT.objektnamn	id_objekt
OBJEKT.har ADMINISTRATIV EGENSKAP.projekt	id_projekt
OBJEKT.har ADMINISTRATIV EGENSKAP.revisionsläge	id_revläge
beskrivning	beskriv
minkardinalitet	minkard
maxkardinalitet	maxkard
Samband	
* har huvudriktning inom RELATION.relationsnamn	relation
* <u>utgår från</u> OBJEKT.objektnamn	ut_objekt
* OBJEKT.har ADMINISTRATIV EGENSKAP.projekt	ut_projekt
* OBJEKT.har ADMINISTRATIV EGENSKAP.revisionsläge	ut_revläge
* <u>riktas till</u> OBJEKT.objektnamn	ri_objekt
* OBJEKT.har ADMINISTRATIV EGENSKAP.projekt	ri_projekt
* OBJEKT.har ADMINISTRATIV EGENSKAP.revisionsläge	ri_revläge

har inversiktning.RELATION.relationsnamn	invers
<u>utgår från</u> OBJEKT.objektnamn	i_ut_objekt
OBJEKT.har ADMINISTRATIV EGENSKAP.projekt	i_ut_projekt
OBJEKT.har ADMINISTRATIV EGENSKAP.revisionsläge	i_ut_revläge
<u>riktas till</u> OBJEKT.objektnamn	i_ri_objekt
OBJEKT.har ADMINISTRATIV EGENSKAP.projekt	i_ri_projekt
OBJEKT.har ADMINISTRATIV EGENSKAP.revisionsläge	i_ri_revläge

8 Resultat

Jan Ljungberg, SISU, Göteborg, utförde databaskonstruktionen i Oracle och skapade inmatningsbilder i SQL-Forms. Vi testade formulären med testdata och det fungerade i stort sett som det var tänkt. Formulären visas i bilaga 1.

För att kunna göra en verklig test behövdes verkliga data. Under sommaren 1991 tänkte vi göra ett försök till att få in verkliga data. Vi upplevde då två problem. Vi hade ingen modell som uppfyllde de krav vi tillställt systemutvecklingsprojektet. Dessa hade också godkänts av ledningen för Division NätTjänster. Kraven bestod kortfattat av att "alla attribut" i metamodellen i figur 2 med minimikardinalitet "lika med ett" skulle vara beskrivna

Systemutvecklingsprojektet ansåg kraven för höga och svåra att förstå. Man hade ofta inte gjort ett ordentligt systemeringsarbete utan koncentrerat sig på konstruktionsfrågor.

Det andra problemet uppstod då vi försökte mata in de ofullständiga modeller som fanns. Det var tråkigt och för omständigt att mata in uppgifter mha skärmbilderna.

Under tiden hade en del modeller samlats i ADW. ADW har vissa fasta rapporter men ingen som stöder dataadministration (DA). DA behöver kunna lista begrepp. Med begrepp menas här en komplett beskrivning över en entitetstyp med dess definition, attributtyper, sambandstyper, etc.. Om det gick att överföra data från ADW till IA-prototypen skulle man kunna skraddarsy rapporterna.

ADW's encyclopedia består av fem filer. I ADW-manualen beskrivs SQL-satserna för att skapa fyra ungefärligt motsvarande tabeller i en relationsdatabas. Encyclopedians fem filer kan importeras till de fyra tabellerna.

Utifrån dessa bör det vara möjligt att flytta data till prototypens tabeller. Tom Göransson, Tvt Nät, gjorde ett försök att prova detta under sommaren 1991. "ADW-tabellerna" har en invecklad struktur. SQL-satserna blir därför komplicerade och kräver djupa kunskaper i språket vilket vi inte hade. Försöken misslyckades därför. Det skulle kräva en större insats vilket vi inte pga andra prioriteringar hade resurser till.

BILAGA 1, Skärmbildslayouter

Administrativ Egenskap	
Projekt	Revisionsläge
Uppgiftslämnare	Status
Uppgjord datum 01-JUN-92	Revisionsdatum
Definitions ansvarig organisation	

OBJEKT FORMULÄR			
Projekt:	Revisionsläge:		
Objekt: Beskrivning:	Medelantal:		
Reglar:			
Objektsynonymer	Objekt/Funktionsreferenser		
Egenskapsnamn	Beskrivning:	Min:	Max:
Ctrl-PgDn för att ge en detaljerad egenskapsbeskrivning.			

ÖVRIGA OBJEKTEGENSKAPER	
Projekt:	Revisionsläge:
Specialiseringsaspekt	
av Objekt:	
Spec. grund:	
Spec.egenskap: U	
Subobjekt:	
Objektrotsvarighet	
Objekt:	Motavaras av:

BILAGA 2, IA-prototypens databasschema

CREATE TABLE MADM_EGENSKAP

(PROJEKT	CHAR(40)	NOT NULL,
REVLGE	CHAR(10)	NOT NULL,
UPPG_LMN	CHAR(40)	NOT NULL,
UPPGJORD_DAT	DATE	NOT NULL,
DEF_ANSV_ORG	CHAR(40)	NOT NULL,
REV_DATUM	DATE	NOT NULL,
STATUS	CHAR(10)	NOT NULL);

CREATE TABLE MDOM

(D_NAMN	CHAR (40)	NOT NULL,
PROJEKT	CHAR (40)	NOT NULL,
REVLGE	CHAR (10)	NOT NULL,
BESKRIV	LONG	NOT NULL,
MAXTECK	NUMBER,	
TECKTYP	CHAR (40),	
LAYOUT	CHAR (40),	
VRD_FRD	CHAR (40),	
KOMMENT	char (240));	

CREATE TABLE MDOM_MOTSV

(DOM	CHAR (40) NOT NULL,
PROJEKT	CHAR (40) NOT NULL,
REVLGE	CHAR (10) NOT NULL,
M_DOM	CHAR (40) NOT NULL,
M_PROJEKT	CHAR (40) NOT NULL,
M_REVLGE	CHAR (40) NOT NULL);

CREATE TABLE MDOM_SPEC

(DOM	CHAR (40) NOT NULL,
PROJEKT	CHAR (40) NOT NULL,
REVLGE	CHAR (10) NOT NULL,
S_DOM	CHAR (40) NOT NULL,
S_PROJEKT	CHAR (40) NOT NULL,
S_REVLGE	CHAR (40) NOT NULL);

```
CREATE TABLE MDOM_SYNONYM
(
    DOM CHAR (40) NOT NULL,
    PROJEKT CHAR (40) NOT NULL,
    REVLGE CHAR (10) NOT NULL,
    D_SYN CHAR (40) NOT NULL,
    S_PROJEKT CHAR (40) NOT NULL,
    S_REVLGE CHAR (10) NOT NULL);
```

```
CREATE TABLE MEGENSKAP
(
    E_NAMN CHAR (40) NOT NULL,
    B_OBJEKT CHAR (40) NOT NULL,
    B_PROJEKT CHAR (40) NOT NULL,
    B_REVLGE CHAR (10) NOT NULL,
    ID_OBJEKT CHAR (40),
    ID_PROJEKT CHAR (40),
    ID_REVLGE CHAR (10),
    AVBILD CHAR (40),
    BESKRIV LONG NOT NULL,
    MINKARD CHAR (1),
    MAXKARD CHAR (1),
    HREGEL CHAR (240),
    BERREGEL CHAR (240),
    KOMMENT CHAR (240));
```

```
CREATE TABLE MEGENS_FUNREF
(
    E_NAMN CHAR (40) NOT NULL,
    OBJEKT CHAR (40) NOT NULL,
    PROJEKT CHAR (40) NOT NULL,
    REVLGE CHAR (10) NOT NULL,
    FUNKTION CHAR (40) NOT NULL);
```

```

CREATE TABLE MEGENS_SYN
    (E_NAMN CHAR (40) NOT NULL,
    OBJEKT CHAR (40) NOT NULL,
    PROJEKT CHAR (40) NOT NULL,
    REVLGE CHAR (10) NOT NULL,
    E_SYN CHAR (40) NOT NULL);

CREATE TABLE MFUNKTION
    (F_NAMN CHAR (40) NOT NULL);

CREATE TABLE MOBJEKT
    (O_NAMN CHAR (40) NOT NULL,
    PROJEKT CHAR (40) NOT NULL,
    REVLGE CHAR (10) NOT NULL,
    BESKRIV LONG NOT NULL,
    MEDELANT NUMBER,
    REGLER CHAR (240),
    KOMMENT CHAR (240));

CREATE TABLE MOBJ_FUNREF
    (O_NAMN CHAR (40) NOT NULL,
    PROJEKT CHAR (40) NOT NULL,
    REVLGE CHAR (10) NOT NULL,
    FUNKTION CHAR (40) NOT NULL);

CREATE TABLE MOBJMOTSV
    (O_NAMN CHAR (40) NOT NULL,
    PROJEKT CHAR (40) NOT NULL,
    REVLGE CHAR (10) NOT NULL,
    M_OBJEKT CHAR (40) NOT NULL,
    M_PROJEKT CHAR (40) NOT NULL,
    M_REVLGE CHAR (10));

```

```

CREATE TABLE MOBJ_SYNONYM
    (OBJEKT CHAR(40) NOT NULL,
    PROJEKT CHAR(40) NOT NULL,
    REVLGE CHAR(10) NOT NULL,
    O_SYN CHAR(40) NOT NULL);

CREATE TABLE MRELATION
    (R_NAMN CHAR(40) NOT NULL,
    UT_OBJEKT CHAR(40) NOT NULL,
    UT_PROJEKT CHAR(40) NOT NULL,
    UT_REVLGE CHAR(10) NOT NULL,
    RI_OBJEKT CHAR(40) NOT NULL,
    RI_PROJEKT CHAR(40) NOT NULL,
    RI_REVLGE CHAR(10) NOT NULL,
    ID_OBJEKT CHAR(40),
    ID_PROJEKT CHAR(40),
    ID_REVLGE CHAR(10),
    BESKRIV LONG,
    MINKARD CHAR(1) NOT NULL,
    MAXKARD CHAR(1) NOT NULL,
    H_REGEL CHAR(240),
    KOMMENT CHAR(240));

CREATE TABLE MREL_FUNREF
    (RELATION CHAR(40) NOT NULL,
    PROJEKT CHAR(40) NOT NULL,
    REVLGE CHAR(10) NOT NULL,
    FUNKTION CHAR(40) NOT NULL);

CREATE TABLE MSAMBAND
    (RELATION CHAR(40) NOT NULL,
    UT_OBJEKT CHAR(40) NOT NULL,
    UT_PROJEKT CHAR(40) NOT NULL,
    UT_REVLGE CHAR(10) NOT NULL,
    RI_OBJEKT CHAR(40) NOT NULL,
    RI_PROJEKT CHAR(40) NOT NULL,
    RI_REVLGE CHAR(10) NOT NULL,
    INVERS CHAR(40) NOT NULL);

```

```
CREATE TABLE MSPECASPEKT
```

```
    (SPECGRUND CHAR (40) NOT NULL,  
     OBJEKT CHAR (40) NOT NULL,  
     PROJEKT CHAR (40) NOT NULL,  
     REVLGE CHAR (10) NOT NULL,  
     SPECEGEN CHAR (40) NOT NULL);
```

```
CREATE TABLE MSUBOBJEKT
```

```
    (SPECGRUND CHAR (40) NOT NULL,  
     OBJEKT CHAR (40) NOT NULL,  
     PROJEKT CHAR (40) NOT NULL,  
     REVLGE CHAR (10) NOT NULL,  
     SUBOBJ CHAR (40) NOT NULL,  
     SPROJEKT CHAR (40) NOT NULL,  
     SREVLGE CHAR (10) NOT NULL);
```

```
CREATE TABLE MTSYNONYM
```

```
    (SYN CHAR (40) NOT NULL);
```

BILAGA 3, ADW's fyra tabeller

```

CREATE TABLE OI                                     Object Instance Relation
(OI$ID          NUMBER(11,0) NOT NULL, object instance token
OI$TYPE        NUMBER(5,0) NOT NULL,  object type code
OI$NAME        CHAR(32));                object instance name
CREATE UNIQUE INDEX OI_PK ON OI(OI$ID);
CREATE INDEX OI_TY_FK ON OI(OI$TYPE);          FK för TY.TYPE

```

```

CREATE TABLE AI                                     Association Instance Relation
(AI$ID          NUMBER(11,0) NOT NULL, association instance token
AI$TYPE        NUMBER(5,0) NOT NULL,  association type code
FR$ID          NUMBER(11,0) NOT NULL,  From token
TO$ID          NUMBER(11,0) NOT NULL);To token
CREATE UNIQUE INDEX AI_PK ON AI(AI$ID);
CREATE INDEX AI_TY_FK ON AI(AI$TYPE);          FK för TY.TYPE
CREATE INDEX AI_FR_FK ON AI(FR$ID);
CREATE INDEX AI_TO_FK ON AI(TO$ID);

```

```

CREATE TABLE PI                                     Property Instance Relation
(SUBJ$ID       NUMBER(11,0) NOT NULL, subject token
PI$TYPE        NUMBER(5,0) NOT NULL,  property code
SEQ            NUMBER(5,0) NOT NULL,  repetiton number
PI$NAME        CHAR(72));                property value
CREATE UNIQUE INDEX PI_PK ON PI(SUBJ$ID,PI$TYPE,SEQ);
CREATE INDEX PI_TY_FK ON PI(PI$TYPE);
CREATE INDEX PI_SUBJ_FK ON PI(SUBJ$ID);

```

```

CREATE TABLE TY                                     Translation Relation
(TYPE          NUMBER(5,0) NOT NULL,  subject type code
INAME         CHAR(32),                internal subject name
DNAME         CHAR(32));                external subject name
CREATE UNIQUE INDEX TY_PK ON TY(TYPE);

```

Exempel på "Type Codes":

20007 "Attribute Type" describes "Entity Type"

20044 Relationship type

10007 Entity Type

10003 Attribute Type

30011 Name

30076 Definition

30077 Comment

TRIAD-rapporter per 921130

Verksamhetskrav på informationsadministration

- V 1: IA och verksamhetens krav – erfarenheter från offentlig förvaltning
- V 2: Fallstudie av IA-projektet vid Televerket
- V 3: IA-erfarenheter från företag och myndigheter
- V 4: Frihandel och handlingsfrihet - informationsvägar till 2000-talet (KOMMER jan 1993)

Modellering

- N 1: Modelleringsansatser för begrepps- och datamodellering: – Beskrivning och försök till jämförelse
- N 2: Generering av konceptuella modeller från policydokument
- N 3: Espritprojektet Tempora
- N 4: Prövning av regelbaserad metodik inom Posten
- N 5: En kokbok i remodellering - utkast
- N 6: Datorstöd för modellintegration
- N 7: Modellbaserad kunskapsinsamling
- N 8: Modellkvalitet (NY - nov 1992)
- N 9: Samband mellan dokument och modeller (NY - nov 1992)

Utbildning

- H1 – Handledarutbildning för modelleringsledare, avancerad
- H2 - Slutrapport HUMLA prototyp (NY - nov 1992. 4 sidor)

Katalogprinciper

- K 1: IRDS
- K 2: IRDS Modeller och modellnivåer
- K 3: Kopping begreppsmodell - relationsmodell
- K 4: IBM:s Repository Manager- en Introduktion
- K 5: IBM:s Repository Manager: Datamodelleringsbegreppen
- K 6: IBM:s Repository Manager: Begreppsmodellering i Information Model
- K 7: IBM Repository Manager: Attribut- och värdemodellering i Enterprise Submodel
- K 8: Navigering i Repository
- K 9: TRIAD Newsletter – IRDS inom ISO. Dagsläget
- K 10: TRIAD Newsletter –ISO/IRDS. Händelseutvecklingen 91/92
- K 11: Samverkan mellan resurskataloger – visioner eller behov
- K 12: AD/Cycle i Information Model – Processer och informationsflöden mellan processer
- K 13: AD/Cycle i Information Model – Info Flows inom Processmodellen
- K 14: AD/Cycle i Information Model – Relationsdatabasmodellering
- K 15: AD/Cycle i Information Model – Härledningsspecifikationer i begreppsmodellen
- K 16: IA-prototyp (NY - nov 1992)
- K 17: Repository AD/Cycle (R//AD) - International Users Group (NY - nov 1992)

Uttagssystem

- U1: Hybris i Unix-miljö, förstudie
- U2: DEBRIS (NY - nov 1992)