TRIAD

Rapport K 24

HYBRIS IA/DA

– en IA-protortyp vid Telia

Björn Norén

TRIAD-PROJEKTET DRIVS GEMENSAMT AV TELIA, POSTEN, STATSKONTORET, ERICSSON OCH SISU, SVENSKA INSTITUTET FÖR SYSTEMUTVECKLING. FÖR VIDARE INFORMATION KONTAKTA SISU, ELECTRUM 212, 164 40 KISTA, TEL 08-752 16 00, FAX 08-752 68 00.

Rapporterna beställs från:
SISU • Electrum 212 • 164 40 Kista • Fax 08-752 68 00
Rapporterna är endast tillgängliga för TRIAD-parterna och är avgiftsfria.

Innehåll

1. Sammanfattning

Det har visat sig vara svårt att fastställa kraven på ett Repository utifrån ett IA-perspektiv. Syftet med aktiviteten har varit att med hjälp av en prototyp verifiera de informationskrav som har funnits dokumenterade i en meta-modell och komma fram till vilka övriga krav som kan ställas på ett Repository.

För att ge struktur åt och avgränsa problemet presenterar vi ett ramverk. För den avgränsade delen visas översiktligt den verksamhetsprocess som ska stödjas. Aktiviteten har bedrivits som en sidoaktivitet. Därför har bara principerna testats.

Det har fungerat att föra över data från ADW till en Oracle-databas. Från Oracle-databasen har det också varit möjligt att ta fram speciella rapporter och göra enklare sökningar. Ett av syftena med aktiviteten var att undersöka hur Hybris kan användas för spontana sökningar från ett Repository och att redovisa svagheterna i denna hantering.

För att åstadkomma ett användbart Repository krävs att projektet får fortsätta med resursinsatser som är jämförbara med dem för ett konventionellt systemutvecklingsprojekt.

2. Bakgrund

Aktiviteten är en fortsättning på Triad-aktiviteten "IA-prototyp, K16" som redan rapporterats. Målet var att praktiskt testa IA:s arbetsprocess med hjälp av en prototyp och ta fram en specifikation av kraven på ett Repository. Informationsbehovet består av diverse spontana sökningar och framtagning av speciella rapporter. Vi ville undersöka om Hybris kunde användas för att åstadkomma dessa.

3. Projektmål

Målet för projektet var att skapa ett Repository med ett grafiskt navigeringsstöd, med hjälp av en databas för lagring av begreppsmodeller. Navigeringsstödet skulle ge möjlighet till spontana sökningar och specialrapporter.

Det skulle också vara möjligt att mata in och hålla isär begreppsmodeller med hjälp av prototypen. Begreppsmodellerna kan avse såväl projektmodeller för olika revisionslägen som modeller för olika organisationsnivåer t ex företagsmodeller, divisionsmodeller etc.

4. Ramverk

För att få en översikt över problemet är det bra att ha ett ramverk eller en struktur. Vi har byggt vårt ramverk på en allmänt accepterad arkitektur för utveckling av informationssystem, nämligen John Zachmans "Framework for Information Systems Architecture" från 1987. Han har jämfört utvecklingen av informationssystem med ingenjörs arbete, t ex att bygga fastigheter.

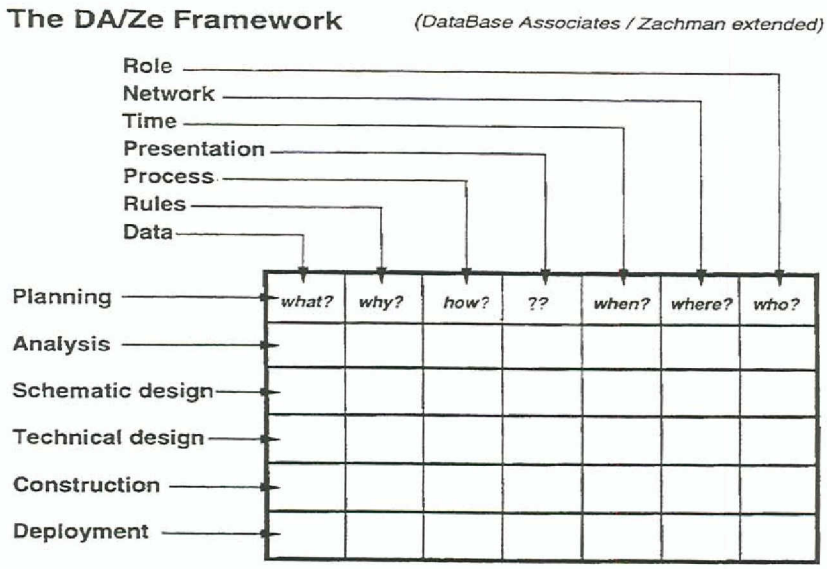
Arkitekturen visar utvecklingen av informationssystem nivå- eller fasindelad utifrån de inblandade parternas olika behov. Den översta nivån ses ur kundens perspektiv. Under denna kommer systemerarens och sedan konstruktörens perspektiv. Kunden vill ha en ändamålsenlig produkt och gör en beskrivning av den. Kundens beskrivning översätts sedan till systemerarens perspektiv etc. En vanlig missuppfattning är att tro att de olika perspektiven blir mer och mer detaljerade. Det stämmer inte. De skiljer sig bara åt när det gäller innehåll och semantik.

Arkitekturen visar olika aspekter på verksamhets- och systemutveckling med hjälp av en matris med tre kolumner och fem rader. Raderna motsvarar de olika parternas perspektiv och kolumnerna är en uppdelning på aspekterna. I sin utbyggda form – Zachman Extended – innehåller matrisen sex kolumner och fem rader.

- Data (what?)
- Funktion eller Process (how?)
- Nätverk (where?)
- Människor (who?)
- Tid (when?)
- Motivation (Rules) (why?)

Följande exempel visar vad som beskrivs i cellerna:

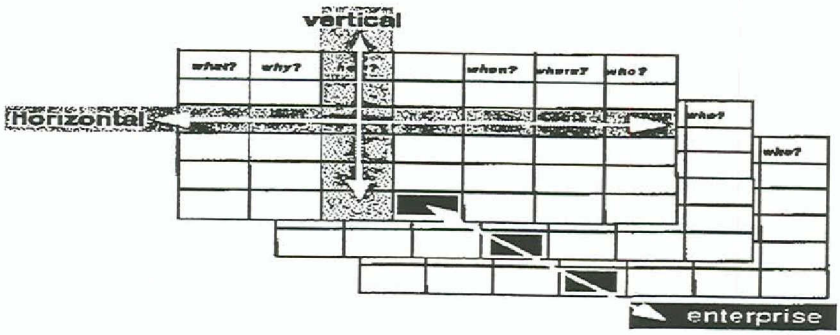
- **Människor**: organisation, användargränssnitt och säkerhet
- **Tid**: verksamhets- och systemhändelser
- **Motivation**: mål, kritiska framgångsfaktorer och kunskapsrepresentation



Figur 1 – Matrisen för Zachmans Extended innehåller totalt 35 celler

Över den här arkitekturen kan man lägga olika aspekter på integrering:

- **Horizontal**, t ex integrering mellan funktion och data.
- **Vertical**, t ex integrering av livscykeln.
- **Enterprise**, t ex integrering och återanvändning av olika systems konceptuella datamodeller (begreppsmodeller). Det är detta som är kärnan i informationsadministration eller dataadministration.
- **Control**, t ex likartad funktionalitet i verktyg.
- **Visual**, t ex likartad presentation.



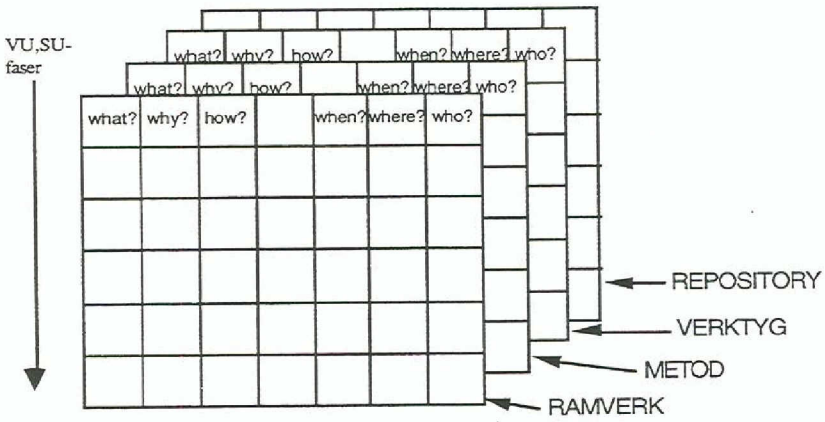
Figur 2 – Tre typer av integrering

Vi använde Zachmans modell och lade till tre plan bakom den (se figur 3). De horisontella raderna visar de olika faserna i verksamhetsutveckling (VU) och systemutveckling (SU).

Varje cell i ramverket definieras med hjälp av en s k referensmodell (metamodell, modermodell). Referensmodellen definierar begreppen för den metod som används för att beskriva cellen. Referensmodellen kan också visa kopplingen till de omgivande cellerna.

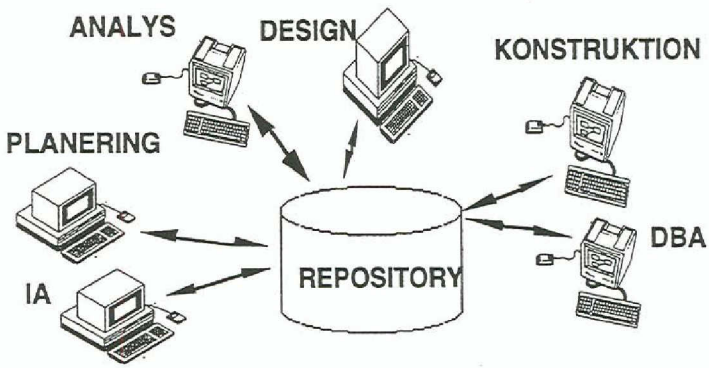
Referensmodellen är samtidigt en specifikation över vad som ska lagras i de verktyg som stödjer metoden och det Repository (datakatalogsystem, resurskatalog) som stödjer verktygen.

De vertikala kolumnerna i ramverket visar den utvecklingsprocess som stöds av olika samverkande metoder (cellerna) som i sin tur stöds av olika verktyg enligt figuren nedan.



Figur 3 – Zachmans arkitektur med tre bakomliggande plan

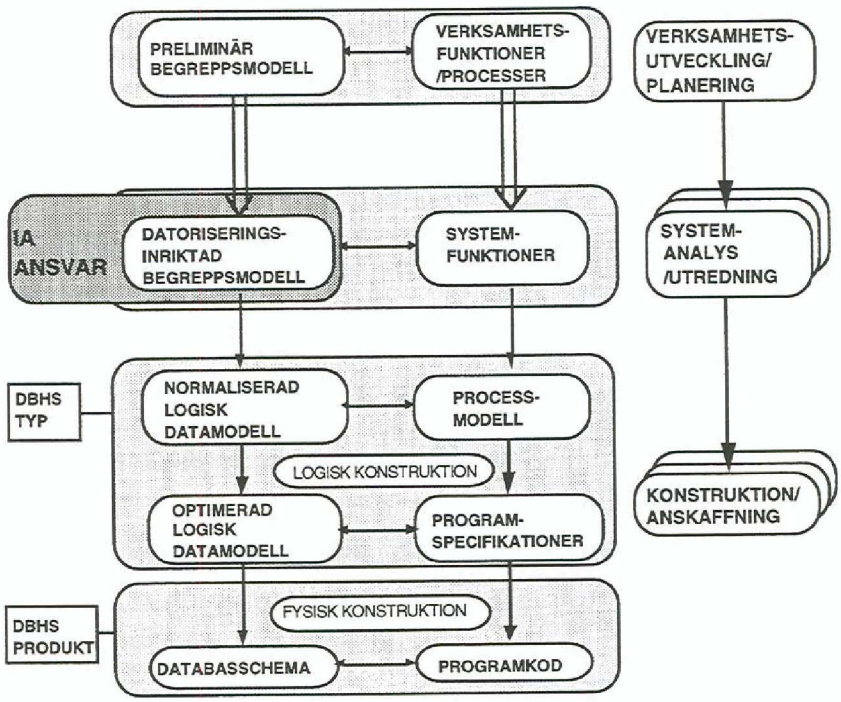
Verktyg för olika faser eller användare samverkar genom Repositoryt enligt nedanstående figur.



Figur 4 – Ett Repository samverkar med flera olika verktyg

5. Ramverk för division NätTjänster på Telia

På Telias division NätTjänst arbetar vi med en systemutvecklingsmodell som är en variant av Zachmans arkitektur.



Figur 5 – Projektet NSTÖD-IA:s variant av Zachmans ramverk

Modellen beskriver *vad* som ska göras d v s de arbetssteg (faser) som modellen föreskriver för utveckling av informationssystem. För varje arbetssteg visas förutsättningarna för arbetssteget och vilket resultat det ska ge. Resultatet från ett arbetssteg används i det efterföljande arbetssteget.

Hur arbetsstegen ska utföras beskrivs av arbetsstegets metod. Metoderna stöds av CASE-verktyg där resultatet från de olika arbetsstegen lagras. I konstruktionsfasen kan vissa arbetssteg automatiseras helt eller delvis. De sammanlagda resultaten från olika projekt lagras i en katalog, ett Repository.

Man väljer metod och verktyg utifrån de arbetssteg som systemutvecklingsmodellen definierar. Vid olika målmiljöer, RDBMS, OODBMS, Edifact-meddelande, Q-gränssnitt, olika programmeringspråk etc, krävs det olika metoder för arbetsstegen inom konstruktionsfasen.

När man väljer eller byter metod är det viktigt att bibehålla gränssnittet till tidigare, efterföljande och omgivande arbetssteg.

Ambitionen är att i framtiden eliminera vissa arbetssteg i konstruktionsfasen genom att bygga in komplexiteten i målmiljön. Därigenom blir det möjligt att generera applikationer direkt från specifikationerna.

För de tidiga faserna kan antalet metoder begränsas. När det gäller verksamhetsanalys bör en metod vara tillräcklig, t ex samma metod för Process Management som för strategisk planering av informationssystem (ISP).

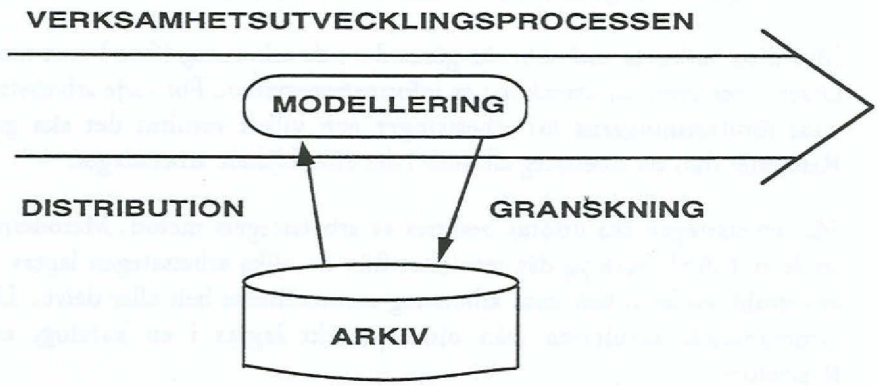
5.1 IA-processen på division NätTjänster

I NSTÖD-IA:s systemutvecklingsmodell avgränsades IA till att gälla cellen "Datoriseringsinriktad begreppsmodell". Eftersom resurserna för IA var begränsade var beslutet mer organisatoriskt än en definition av IA. IA kan i stället definieras så att begreppet omfattar såväl metoder och administration av alla arbetssteg (celler) som utförandet av arbetet i de olika arbetsstegen. Begränsningen enligt NSTÖD-IA gäller i dagsläget också för SIA:s[1] arbete.

Referensmodellen för "Datoriseringsinriktad begreppsmodell", beskrivs i dokumentet "Division NätTjänsters krav på begreppsdefinitioner", NMS-92:001. Syftet med att ta fram begreppsdefinitioner är enligt IP90[2] *återanvändbarhet* och *enhetlighet.*

För att uppnå *återanvändbarhet* måste begrepps definitionerna vara begripliga. För att begreppsdefinitionerna ska vara *enhetliga* måste de fastställas av verksamhetsledningen.

IA-funktionens arbetsprocess beskrivs i IP90. Den kan förenklas till nedanstående bild. En något utförligare bild visas i figur 7.



Figur 6 – Arbetsprocessen för IA-funktionen

[1] SIA = Telias samordningsgrupp för IA

[2] IP90 = Policy för informationshantering inom Telia

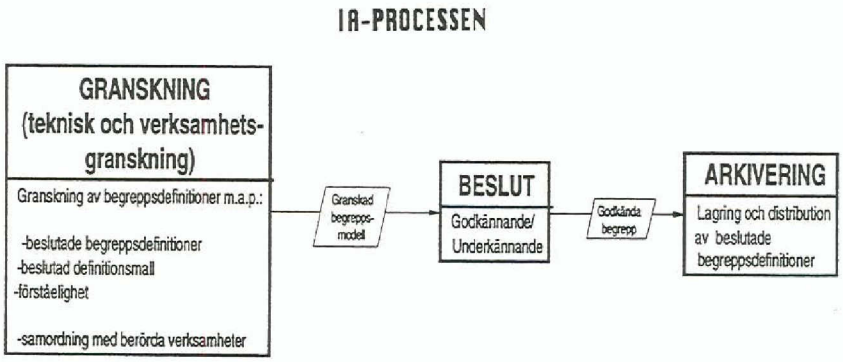
En av IA:s främsta uppgifter i dag är att kvalitetsgranska framtagna modeller.

Innan man kan gå över till konstruktionsfasen (TollGate 2) överlämnas begreppsdefinitionerna till IA för kvalitetsgranskning. Granskningen består av en teknisk del och en verksamhetsdel. Den tekniska granskningen består av:

- Granskning av syntaktisk korrekthet[3], vilket innebär att begreppen är komplett beskrivna enligt "Division NätTjänsters krav på begreppsdefinitioner", NMS-92:001.
- Jämförelse med redan definierade begrepp.
- Granskning av språket så att beskrivningarna är begripliga.
- Kontroll av att de berörda verksamheterna har fått påverka modellutformningen (att samordning skett).

Verksamhetsgranskningen består av en presentation och genomgång av modellen med representanter från SIA, regionerna, berörda projekt etc.

När begreppsdefinitionerna har godkänts överlämnas gemensamma begrepp till SIA som vidarebefordrar dem till koncernens ledningsgrupp (KLG). Ledningsgruppen fastställer sedan begreppen.



Figur 7 – IA-funktionens arbetsprocess

[3] Se Triad-rapport N8, "Modellkvalitet"

6. Systemkrav

Ett Repository är ett stödsystem för bland annat IA-funktionen. Kraven på ett Repository bör därför formuleras med hjälp av konventionell systemutvecklingsmetodik. Det innebär att kraven beskrivs i flera modeller:

- funktionsmodell
- dialogmodell
- datoriseringsinriktad begreppsmodell

Kraven bör sedan verifieras löpande med hjälp av en prototyp. För denna typ av tillämpning har dialogmodellen utgått eftersom dialogen främst består av spontana sökningar.

Normalt utvecklar man inte ett eget Repository utan köper en färdig produkt. Produktens egenskaper ger möjligheter som jämförs med kraven samtidigt som kraven modifieras med hänsyn till de möjligheter produkten ger. I bilagan finns en lista över några Repository-produkter. Att gå igenom egenskaperna för en produkt är dock ett omfattande arbete som inte ingår i den här rapporten.

Figur 8 – Kraven på ett Repository jämförs med dess möjligheter

6.1 Funktionskrav

6.1.1 Allmänt

Systemets funktionskrav kan beskrivas med en funktionsmodell. Med en funktion menas här

- ett antal operationer på data (begrepp, entiteter) som man i systemeringsarbetet valt att hantera tillsammans
- en aktivitet som utförs av systemet i enlighet med kravställarnas önskemål
- en i tiden sammanhängande mängd aktiviteter som är till stöd för användaren i en begränsad arbetsuppgift

Syftet med en funktionsmodell är att

- ge en överblickbar hierarkisk struktur över systemets funktioner
- ge en beskrivning över funktionernas uppgift i systemet
- genom successiv nedbrytning komma fram till funktioner på lägsta nivå som beskriver vad som ska göras
- vara underlag för en beskrivning i dialogsteg som beskriver **hur** funktionen ska utföras
- ge referenser till den datoriseringsinriktade begreppsmodellen

Funktionsmodellering utförs tillsammans med begreppsmodellering i en iterativ process.

6.1.2 IA:s funktionskrav

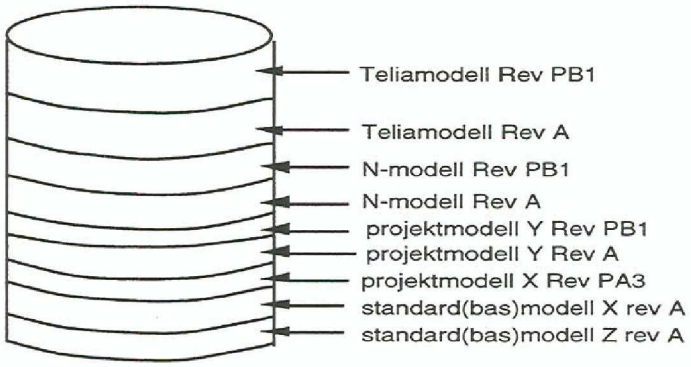
Det är svårt att fastställa funktionskrav för den här typen av system. Därför var ett av syftena med prototypen att komma fram till vilka funktionskrav som bör ställas på ett Repository. Av den anledningen finns bara en del av de tänkbara funktionskraven med i listan nedan. Det ska finnas möjlighet att bland annat

- mata in och hålla isär begreppsmodeller från olika projekt med olika revisionslägen
- lagra gemensamt definierade begrepp (N-modell respektive Telia-modell) separat men i samma databas som olika begreppsmodeller från olika systemut vecklingsprojekt
- lista en modells innehåll med valbara egenskaper, t ex att lista ett enskilt objekt[4] med valbart antal egenskaper: attribut (egenskaper och relationer), domäner, beskrivningar etc
- söka efter samma namn på objekt, attribut etc i olika modeller för att hitta de som är gemensamma i olika begreppsmodeller (det ska också finnas möjlighet att söka på synonymer vilket är till hjälp vid integreringar)
- 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 version
- söka på vad ett ej specialiserat objekt i en modell motsvaras av i form av ett specialiserat objekt i en annan modell
- lista alla objektnamn och i vilken modell de finns samt vilken status objektet har
- lista de uppgifter som saknas i modellen (exempel på en modellgransknings uppgift som bör kunna automatiseras)

---

[4] Objekt i denna skrift är synonymt med entitet (entitetstyp).

Möjligheten att lagra olika typer av modeller med olika revisionslägen beskrivs schematiskt i nedanstående bild:



Figur 9 – Olika modeller med olika revisionslägen i samma databas.

6.2 Informationskrav

Informationskraven fanns redan beskrivna i en datoriseringsinriktad begreppsmodell från Triad-aktiviteten IA-prototyp[5] där den användes som underlag för databaskonstruktion. Modellen gjordes med utgångspunkt i dokumentet "Division NätTjänsters krav på begreppsdefinitioner" från 1992. Dessa krav har under tiden reviderats men för att undvika att göra ändringar i databasdesignen fördes inte ändringarna in.

För att man ska kunna lagra olika typer av modeller med olika revisionslägen finns objektet "Administrativa uppgifter" eller "Modelluppgifter" med i metamodellen. Objektet identifieras av "modellnamn" (tidigare "projekt/modell") och "revisionsläge".

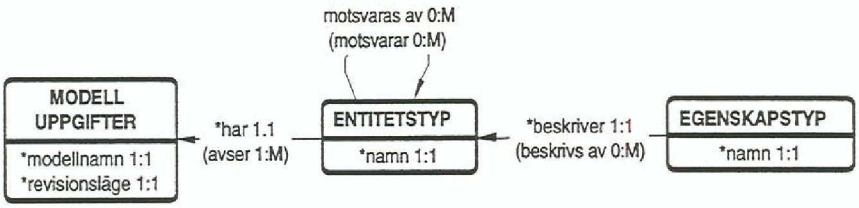
Detta är till hjälp vid integreringar eftersom man ser om samma objekt förekommer i flera modeller. Relationen "motsvaras av" för "Objekt" och "Domän" kan också användas som stöd vid integreringar.

Till en början räcker det att knyta "Administrativa uppgifter" till objektet "Objekt", eftersom det minskar komplexiteten i prototypen.

---

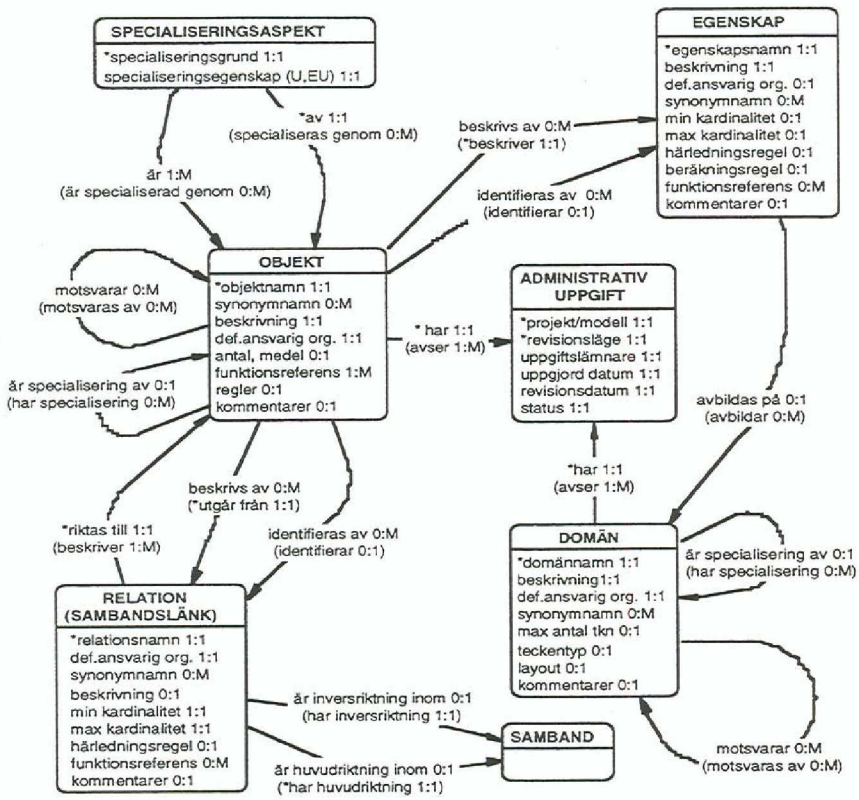
[5] Beskrivs i Triad-rapport K 16.

En förenklad metamodell visas i figuren nedan. Den beskriver hur man kan lagra olika modeller.



Figur 10 – Förenklad metamodell

I figur 11 visas den modell som realiserades i Telias IA-prototyp.



Figur 11 – Modellen för Telias IA-prototyp

Egenskapen "synonymer", som kan ha flera värden, har lagts till de flesta objekten. Tills vidare kommer synonymer att tillåtas, men eventuellt kan man tänka sig att i stället använda sökord.

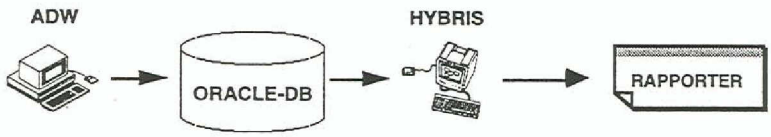
Genom egenskapen "funktionsreferens" finns en koppling till funktionsmodellen.

I framtiden kan man utvidga metamodellen genom att lägga in de begrepp som används vid verksamhetsanalyser. (Telia utför verksamhetsanalyser med hjälp av den s k ISP-metoden.) Det kan ge en lösning på problemet med att hitta definitionsansvariga för begrepp i verksamheten. I ISP beskrivs verksamhetsfunktioner, vem som ansvarar för funktionen, vilka informationsområden verksamhetsfunktionen skapar, läser, uppdaterar eller tar bort (visas med en CRUD-matris) och vilka entitetstyper som hör till informationsområdet. Exempel på informationsområden är linjenätsdata och kunddata. Genom att tillföra detta kan egenskapen "Definitionsansvarig organisationsenhet" utgå för objekt.

Figur 12 – En förenklad metamodell

6.3 Systemstruktur

Begreppsmodellerna tas fram med CASE-verktyget ADW och lagras i filer. Filerna bearbetas med ett program som är skrivet i C och matas in i en Oracle-databas. Med hjälp av Hybris skapas SQL-frågor mot databasen utifrån den grafiska konceptuella metamodellen. Resultatet av Hybris-frågorna överförs till Word, varifrån resultatet skrivs ut.



Figur 12 – Kedjan från begreppsmodell till rapport

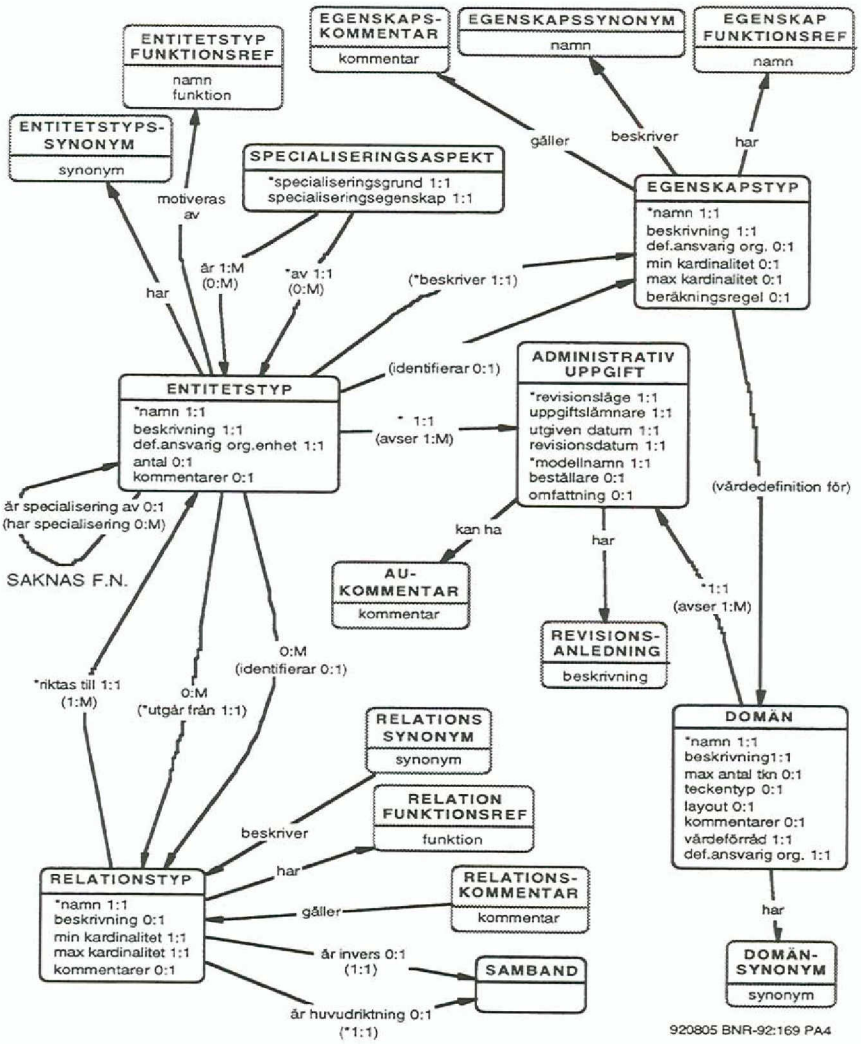
7. Problem

De största problemet under aktiviteten har varit bristen på resurser hos både Telia och SISU. På Telia prioriterades det operativa arbetet och på SISU Intuitive-projektet.

Redan tidigt uppmärksammades dessutom följande svagheter i Hybris:

- Hybris kan bara representera databasens tabeller och inte en konceptuell datamodell. Hybris är utformat efter PULS behov. Ett exempel på detta är att ett attribut som kan anta flera värden i konceptuell datamodellering visas som ett objekt i Hybrismodellen. Modellen i Hybris kommer därför att innehålla fler objekt än den ursprungliga konceptuella datamodellen. Jämför modellerna i figur 11 och 14.
- Självreferenser i konceptuella datamodeller kan inte visas i Hybrismodellen. Därför uteslöt vi egenskapen "motsvaras av" för objekt i prototypen. Det kanske också hade kunnat lösas genom att ett extra objekt hade förts in i modellen.
- Relationerna i Hybris kan bara namnsättas i en riktning. Därför är det i vissa fall svårt att bedöma vad som är huvudriktningen. En pil på relationen hade kunnat lösa det.
- Metamodellens definitioner finns inlagda i Hybris uppslagsbok. Det finns ingen möjlighet att skriva ut metamodellens textbeskrivningar eller ändra dem på ett enkelt sätt.
- Vissa egenskaper i metamodellen, t ex synonymer, saknades i ADW. Detta löste vi genom att ange dem i ADW:s kommentarfält, vilket överföringsprogrammet kunde känna av. Modellnamn och revisionsläge fick vi mata in separat. Domäner registrerades under "global data type" i ADW och registrerades av laddprogrammet tillhörande den modell eller det revisionsläge som de användes av.
- Identifierande attribut anges kryptiskt i ADW. Vi löste inte detta i prototypen.

Slutligen hade vi problemet med att presentera Hybrisfrågan. Vid normal användning av Hybris överförs resultatfilen till Excel men det är olämpligt vid spontana sökningar mot ett Repository. Resultatfilen var inte i läsbart skick så vi gjorde ett försök att skapa en snygg rapport i Word med hjälp av mallar och en Word-programmerare. Detta fungerade i princip. Det fanns några buggar i kedjan ADW – Oracle – Hybris – Word som vi inte lade ned resurser på att rätta till.



Figur 14 – Televerkets Hybrisvariant i juli

8. Resultat

Det är möjligt att göra enklare, spontana sökningar utan kvalitetskrav på rapportlayouten. Mer omfattande rapporter kommer att kräva ytterligare program mering i Word. Å andra sidan är de komplicerade rapporterna få och insatsen begränsad. Programmeringsinsatsen är dock så stor att nyttan av att använda Hybris kan ifrågasättas i dessa fall. Det vore nog enklare att skapa rapporterna direkt med Oracles rapportgenerator.

Att hålla hela kedjan ADW - Oracle - Hybris - Word fri från buggar är ett konstruktionsprojekt som kräver normala konstruktionsresurser. Inom Triad fanns inte tillräckliga resurser för att klara det.

Bilaga

CASE

Det går inte att tala om Repository utan att först nämna CASE. Alla CASE-verktyg har någon metod för intern lagring som ibland benämns Repository. Vad man vanligtvis menar med ett Repository är en central lagringsplats som flera CASE-verktyg arbetar mot.

CASE är inget annat än ett stödsystem för systemutvecklings processen och består av de från IP90 välkända delarna presentation, bearbetning och lagring. Stödsystemet kan vara ett en- eller fleranvändarsystem, ge möjlighet att lagra i filer, relationsdatabaser eller objektorienterade databaser, använda olika typer av plattformar o s v.

Ofta ses CASE som något speciellt, men tillämpar man ett konventionellt synsätt kan man göra problemet enklare eller i alla fall något lättare att strukturera. Konventionell metodik kan t ex tillämpas vid kravspecificering, client/server-baserade CASE-verktyg kan användas vid utveckling av client/server-tillämpningar o s v.

CASE-verktyg utvecklar man sällan själv.

CASE-verktyg brukar indelas i Upper-CASE, Lower-CASE och Integrated-CASE beroende på om det är tidiga, sena eller samtliga systemutvecklingsfaser som stöds. I Lower-CASE inräknas också 4GL-produkter, applikationsgeneratorer och interface-generatorer (GUI-tools). Det talas nu även om Instant-CASE, som direkt från modellerna ska kunna generera applikationer. I CASE-verktyg inräknas också verktyg för testning, reverse engineering, projektstyrning, klassbibliotek, Repository m m. Det finns också meta-CASE-verktyg som kan anpassas till olika metoder.

De flesta CASE-verktyg är främst inriktade på att generera IBM stordator-applikationer i COBOL. Men i dag blir det allt vanligare med distribuerade client/server-lösningar och objektorienterad utveckling. Detta har tvingat in CASE-leverantörerna i en hektisk process för att göra om sina produkter och nyintroducera dem som client/server- och OO-produkter.

CASE-utvecklingen följer plattformsutvecklingen i dess vidaste tolkning. Moderna CASE-verktyg har avancerade användargränssnitt, använder AI-teknik och objektorienterade databaser (OODBMS).

Ett hett begrepp inom informationsteknologin är integrering. Det gäller också inom CASE-området. Integreringsfrågorna behandlas främst inom olika leverantörsgrupperingar och standardiseringsorgan.

Marknaden är i dag mycket dynamisk och instabil. Därför bör utvecklingen följas och analyseras t ex med hänsyn till hur nya CASE-produkter stöder client/server-utveckling och utveckling inom objektorientering. Detta gäller också meta-CASE-produkter. Man bör utgå utifrån de nya utvecklingsmetoder som kommer fram.

Företagsköp och allianser mellan leverantörer bör noteras och konsekvenserna bedömas.

Standardiseringsutvecklingen är intensiv och bör följas, t ex OMG, ECMA/TC33/PCTE, EIA CDIF, IRDS, ISO/IEC JTC1/SC7, ANSI X3H6.

För nya teknologier behöver ofta området struktureras. För t ex client/server har Gartner Group gjort en klassificering som är allmänt accepterad. För objektorientering delar man ofta upp området i systemanalys och design, programmering, databas etc. Nedanstående matris är ett sätt att visa mognaden inom området objektorientering:

	Planning	Analysis	Design	Construction
Technology	None	Intro	Varied	Stable
Models	None	Varied	Varied	Stable
Methods	None	Intro	Direction	Stable
Techniques	None	Intro	Varied	Direction
Practices	None	Intro	Varied	Stable

Repository-marknaden

Det här är ingen genomgång av befintliga Repository-produkter utan bara några presscitat över ett fåtal produkter.

Den ISO-standard som behandlar Repositories benämns Information Resource Dictionary Systems (IRDS) och har beskrivits i Triad-rapporterna K1, K9, K10 och K19.

IBM misslyckades med Repository Manager/MVS på grund av dåliga prestanda, högt pris, omogna kunder och dålig koppling till CASE-verktyg. Få insåg komplexiteten i produkten. Konceptet var rätt men marknaden var inte redo. RM/MVS är implementerad i DB2, men en relationsdatabas är inte lämplig för denna typ av applikation. Det blir många JOIN:s och kontroller (av "policies" på IBM-språk). Låsning av databasobjekt sker på för grov nivå. En lösning av versionshanteringen skulle ytterligare försämra redan dåliga prestanda. RM/MVS kan därför inte användas som ett aktivt Repository för CASE-verktyg. RM/MVS är inte nedlagt utan stöds fortfarande.

Bachman, Intersolv med flera har hoppat av ifrån IBM:s "International Alliance for AD/Cycle" och tagit fram egna alternativ. Bachman har sin "Shared Work Manager" på OS/2-baserade servrar i ett NetWare LAN från Novell. Intersolv har sitt LAN-Repository inom Excelerator XLII-konceptet.

AD/Platform, IBM:s nya LAN-baserade utvecklingsstrategi går under AIX och OS/2. Den består av en informationsmodell, ett klassbibliotek för användargränssnitt och tjänster för verktygsgränssnitt samt en objekt-orienterad databas.

Genom att använda en befintlig objektorienterad databas får man en hel del på köpet. Man får en distribuerad databas, säkerhet och en avancerad versionshantering med arvsmekanismen. Objektslåsning kan ske på en mer detaljerad nivå. Det går lättare att bygga upp en struktur med t ex företags-gemensamma data i olika versioner med delmängder på olika nivåer ned till personnivå.

IBM går nu in för standardlösningar genom att både anpassa sig efter och söka påverka standarder. De för in HP:s Broadcast Message Server (BMS), baserar utvecklingen på industristandarden ECMA 149 Portable Common Tool Interface (PCTE), presenterar sin informationsmodell och sin "Object Oriented Tool Integration Service" (OOTIS) i standardiseringsorgan som ECMA/TC33, OMG och ANSI/X3H6.

Vissa delar av AD/Platform skulle bli klara under 1993, nämligen användargränssnitt och meddelandetjänster för verktyg-till-verktyg. Men nyckelkomponenten, det LAN-baserade objektorienterade Repositoryt, kommer inte att bli tillgängligt förrän någon gång under 1994. Till en början kommer bara informationsmodellens "Technology Model" att kunna stödjas.

Data Manager från MSP är den äldsta och mest etablerade resurskatalogen. Den används främst i konventionella IBM-miljöer framför allt IBM:s egna IBM-DD (Data Dictionary). Den används också under MVS, VSE och VM.

DB Excel under MVS är en av de största konventionella Repository-produkterna i USA.

DataDictionary/Solution från Brownstone har liknande egenskaper som DB Excel. Används också under MVS.

Någon tid för att tränga djupare ned i de tre största produkterna, Data Manager, DB Excel och Data Dictionary/Solution finns för närvarande inte.

Alla produkterna går mot en client/server-arkitektur i takt med att denna teknik mognar. Men för tillfället är inte säkerhetsfrågorna lösta på ett tillfredsställande sätt. Man kan eller kommer att kunna få en produkt baserad på LAN eller stordator eller en kombination av båda. Alla har någon form av GUI på OS/2, Windows och/eller UNIX/Motif. Alla har gränssnitt mot ett varierande antal CASE-verktyg.

CaseSpan från InfoSpan Corp används under UNIX, DOS och OS/2. Flera olika CASE-verktyg kan integreras men de måste användas på en specifik plattform.

Rochade från R&O GmbH är ett tyskt Repository som också finns på den amerikanska marknaden. Dess främsta konkurrent är CaseSpan. Rochade integrerar olika CASE-verktyg och stöder olika plattformar – även client/server under DOS, UNIX, MVS, BS 2000, VM, OS/2 och VMS. Klienten kan gå under ett OS och servern under ett annat vilket inte andra produkter, t ex CaseSpan, klarar av. TCP/IP används mellan klient och server därför att det är det enda protokoll som stöds av alla sju operativsystemen. CDIF stöds enligt försäljningsbroschyren men inte i praktiken. R&O deltar i IBM:s Tool Vendor Assistant Program men följer inte deras informationsmodell. Produkten säljs som ett toolkit med Repositoryt i centrum med verktyg för reengineering, process management, projekt management, work management och användargränssnitt (Autopilot). R&O stöder integrering av CASE-verktyg som ADW, IEW, Excelerator, IEF, Bachman och IBM:s CSP.

Det går inte att integrera olika CASE-verktyg utan vidare beroende på olika semantik i verktygens informationsmodeller. Rochade har en egen informationsmodell som de olika CASE-verktygens informationsmodeller i sin tur avbildas mot. Rochade levererar en specifik informationsmodell för varje CASE-kombination.

Sedan KnowledgeWares misslyckades med ett eget Repository har de gått över till att sälja Rochade.