

SISU **rappport**

nr 8:2

DataAdministration

VAD
DA

SISU

Svenska Institutet för Systemutveckling
Box 1250, S-164 28 Kista, Sweden

Innehåll

1. Funktioner och arbetsuppgifter	1
1.1 En klar konvergens mot dataadministration är märkbar	1
1.2 Uppgifterna i dataadministration	2
1.3 Förändringsarbete och förvaltning	3
2. Metoder och beskrivningstekniker	13
2.1 Metoder/modellering	13
2.2 Krav och önskemål på modellering	17
2.3 Syfte med datamodellering	18
2.4 Vad är konceptuell modellering	20
2.5 Konceptuell modellering - grundbegrepp	23
2.6 Egenskaper hos modelleringsansatser	25
2.7 Typer av modelleringsansatser	26
2.8 Framtagning och tillämpning av konceptuella modeller	28
2.9 Övergång från konceptuell modell till databasmodell	31
2.10 Kort presentation av några modelleringsansatser	32
3. Dataadministration och datorstöd	51
3.1 Bakgrund till intresset för datorstöd	51
3.2 Historik och nuläge	52
3.3 Dataadministration - vilken användningssituation har vi?	55
3.4 Verktyg för dataadministration	56
3.5 Olika typer av datorstöd och dessas samband med datakatalogsystem	57
3.6 Effekter av datorstöd	59
3.7 Innehåll och representation	61
3.8 Gränssnitt människa-maskin	62
3.9 Tänkbara grupper av verktygsfunktioner	64
3.10 Integration och distribution av datorstöd	67
3.11 Arvet av gamla applikationer	69
3.12 Standardapplikationer och dictionary	71
3.13 Framtidens datorstöd	71
4. Organisatoriska aspekter	77
4.1 Inledning	77
4.2 Organisatoriska aspekter	79
4.3 Dataadministratörens uppgift i projekt	84
4.4 Dataadministratörens organisatoriska placering	88
5. Bemanning och kompetensprofiler/kompetensbeskrivning	91

Förord

SISU:s tillämpningsprojekt ger en unik möjlighet att förena kunskaper inom forskningsområdet med erfarenheter av praktiska problem från företag och myndigheter. De ger också deltagande företag och myndigheter ett värdefullt forum för att diskutera erfarenheter.

Som ett resultat av förslag från medlemsföretagen i SISU definierades projektet DATAADMINISTRATION. Det startade 1987-01-01.

Projektet har drivits av SISU och projektledare har varit Anders Persson (Volvo Data AB), som även är huvudförfattare till del 1 i denna rapport. Ansvariga från SISU, som också ingått i projektledningen, har varit Lars-Åke Johansson, Mats-Roger Gustafsson samt Björn Nilsson.

En referensgrupp samt ett antal delområden/delprojekt har knutits till projektet. De deltagande medlemsföretagen har ställt personella resurser till projektets förfogande. Inom varje delprojekt har diskussioner förts i arbetsgrupper och dessa har framställt preliminära resultat till rapportens olika avsnitt.

De preliminära resultaten har sedan bearbetats av projektledningen varefter de slutligen diskuterats i hela referensgruppen. Därefter har olika medlemsföretag kortfattat beskrivit hur dataadministration tillämpas i den egna organisationen vilket redovisas i rapportens del 3.

Projektet har bemannats av 27 personer från företag och myndigheter som har gedigen erfarenhet av ämnesområdet. Därför ger rapporten en bred och förankrad bild av dataadministration och tar främst fasta på praktiska erfarenheter. Förutom resultatet i form av en rapport har mycket värdefull kunskap inhämtats och utvecklats genom att de olika specialisterna arbetat samman under koncentrerade former.

Deltagande personer och företag:

Bertil Axelsson	ADB-kontoret, Göteborg
Jonas Leffler	Data Logic
Stefan Johansson	Data Logic
Christer Dahlgren	Ericsson information systems
Kurt Bauersfeld	Ericsson
Karl-Erik Lundahl	Försvarets fabriksverk (FFV)
Birgit Norén	Försvarets fabriksverk (FFV)
Rolf Björkenwall	Försvarets materialverk (FMV)
Magnus Jonegård	Programator
Håkan Wall	SAAB SCANIA, flygdivisionen
Jan-Erik Ekenhill	SAS
Kurt Edvardsson	Skandia data
Sven Håkansson	SKF
Staffan Nilsson	SKF
Mats-Roger Gustafsson	SISU
Lars-Åke Johansson	SISU
Stig Johansson	SISU
Björn Nilsson	SISU
Bengt Carnö	Statskontoret
Bror Norén	Vattenfall
Börje Länn	Volvo BM
Per Kangevall	Volvo Data
Håkan Lövgren	Volvo Data
Anders Persson	Volvo Data
Tore Altenstedt	Volvo Lastvagnar
Magne Källström	Volvo Lastvagnar
Lars Lundgren	Volvo Personvagnar

Bakgrund

Begreppet dataadministration anknyter både till den traditionella ADB-världen och till den roll som informationsbehandlingen spelar i verksamheten.

Syftet med projektet DATAADMINISTRATION är att ge data dess rätta förankring i verksamheten. Detta har inte endast att göra med samspelet mellan ADB-sidan och verksamheten utan även med nya synsätt och tekniska miljöer vilka tillåter användaren att utveckla och kontrollera sin informationsbehandling på nya, ibland genomgripande sätt.

Företagets verksamhet, mål, styrning och resultat måste vara ledstjärnan för alla delar i företaget. Det finns en inbyggd konflikt mellan å ena sidan ADB-teknik som är trög att ändra och kräver speciell kompetens och, å andra sidan verksamhetens behov av flexibilitet och kontroll över resurser som utnyttjas och krav på användaren.

Miljön för många företag och myndigheter kan betecknas som mycket turbulent. Man står ofta inför ett antal vägval som både har att göra med verksamhetens förändring och användning av ny teknik i kombination med gammal beprövad. Rapporten försöker att bringa ordning bland begreppen, fastslå inriktningar inför framtiden och anvisa de steg som måste tas.

Rapportens delar

Rapporten från projektet Dataadministration består av tre delar: DA-VARFÖR, DA-VAD och DA-HUR.

DA-VARFÖR är en argumentationsskrift där dataadministrationen motiveras och sätts in i sitt sammanhang.

DA-VARFÖR är avsedd att användas av dataadministratörer som en källa till argument och förklaring av den egna verksamheten. Dessutom är det en förhoppning att den skall vara nyttig läsning för dem som nyttjar DA-tjänster.

DA-VAD är avsedd för dem som arbetar med DA eller som deltar i uppbyggnad av DA-verksamhet. Även systemutvecklare som samarbetar med DA bör ha utbyte av rapporten.

Här diskuteras och analyseras lite djupare de frågor som projektdeltagarna uppfattat som kärnfrågor för DA-verksamhet.

DA-HUR är avsedd för alla som arbetar med DA eller använder DA-tjänster.

Här beskrivs kortfattat hur man praktiskt arbetar med DA i olika organisationer. Infallsvinklarna är olika varför rapporten ger en bred belysning av DA-verksamhet i praktiken.

1. FUNKTIONER OCH ARBETSUPPGIFTER

Vilka funktioner och arbetsuppgifter ingår i vår tolkning av Dataadministration? Är det bara ett stöd för verksamheten eller ingår delar som också stödjer systemutveckling och förvaltning? Låt oss beskriva vad vi menar.

Dataadministration är ingenting fristående, utan måste för att få effekt, inlemmas i någon fungerande struktur. Den struktur som är naturlig är AU/ADB-arbete där vi särskiljer de olika delarna:

- verksamhetsutveckling och AU-planering
- förändringsarbete inom det administrativa området
- förvaltning av administrativa rutiner med tillhörande informationssystemstöd.

1.1 EN KLAR KONVERGENS MOT DATAADMINISTRATION ÄR MÄRKBAR

Arbetet inom AU-planering, förändringsarbete och förvaltning kan formaliseras i mer eller mindre strikta arbetssätt eller modeller (systemutvecklingsmodeller, AU-modeller etc). Orsaken till formalismen är att det numera finns en relativt stor erfarenhet inom området och att det också finns klara behov av att nå enhetlighet eftersom många personer med olika kompetenser behöver en gemensam referensram att utgå ifrån. Vågen av datorstöd, de s k CASE-verktygen, driver också på formalismen.

Studerar vi de mest förekommande s k "nya" modellerna som finns på marknaden märks en klar tendens till att de börjar likna varandra. Skillnaderna är, om man tar större eller mindre hänsyn till helheten inom det totala AU-området eller om man har byggt in projekt- och kvalitetsstyrning etc.

Signifikant är att de flesta utgår från en avgränsning, funktions- och datamodellering. Dessutom vill man nyttja det som skapats i ett tidigare steg till ett senare, d v s bygga på ett resultatorienterat arbetssätt skapat av metodkedjor.

Återanvändningsprincipen gör att det är möjligt att datorisera större delar av AU/ADB-processen med verktyg som primärt stöttar det resultat

som skall förädlas. Det krävs dock en klar eftertanke om vad man önskar lagra och återanvända, eftersom varje lagring kräver förvaltning (som i sin tur kostar pengar).

Aktiviteterna som genomförs vid exempelvis utveckling av ett informationssystem bör därför vara relativt bestämda om vi utgår från samma grundsyn nämligen data, funktioner och gränssnitt. Endast paketering, styrning, pedagogik och erfarenhetsexempel ger avarterna i olika modeller.

1.2 UPPGIFTERNA I DATAADMINISTRATION

Vi har valt att beskriva **Dataadministrationens ingående funktioner och arbetsuppgifter** efter det synsätt och den arbetsmodell som i stort används av Volvokoncernen, ASEA Brown Boveri, Data Logic, ADB-kontoret i Göteborg m fl eftersom den liknar strukturen samt att det finns praktisk erfarenhet att ta tillvara. Valet betyder inte att andra arbetsmodeller utesluts eller är mindre användbara vid tillämpning av Dataadministration. Valet tjänar därför endast som referens.

Beslut om utredning	Beslut om projekt	Beslut om förändring	Beslut om förvaltning
Förstudie	Projektering	Genomförande	Förvaltning
Beskriv nödiga Analysera problem Formulera förändringsmål	Analysera projekteringsbehov	Analysera genomförandebestånd	Applikators utnyttjande
Generera förändringsidéer Utforma lösning Värdera effekter Planera projekt	Analysera verksamheten Utforma lösning	Skapa organisation Dataformlösning Anskaffa teknik Rekrytera lösning Integrera lösning	Korrigera fel Anpassa lösning Förbättra lösning Sanera systemet
	Anpassa organisation Utforma arbetssituation Dimensionera teknik Värdera lösning Planera och kalkylera genomförande	Utbilda personal Införa lösning Avsluta projekt	Värdera nyttan
"PROBLEM"	"FUNKTIONER"	"KONSTRUKTION"	"NYTTA"

För varje del i arbetsmodellen beskrivs vilka funktioner och vilka arbetsuppgifter som Dataadministration har. Beskrivningen är uppdelad så här:

- Med ett **syfte** som anger roller för Dataadministration.
- Vilka **uppgifter** som konkret behandlas. Beskrivningen är utformad enligt gängse nomenklatur för befattningsbeskrivningar och som sedan kan utnyttjas för det egna företaget/myndigheten. Exempel på nomenklatur är **ansvara för, tillse att, medverka vid och utverka**.
- **Resultatet** av genomförda uppgifter.

Ingen hänsyn har tagits till vilka befattningar eller vilka organisatoriska grupper som genomför arbetsuppgifterna. Detta kan vi göra i efterhand med hjälp av exempelvis ansvars- och kompetensmatriser samt befattningsbeskrivningar. Det viktiga är först att klargöra, vad som skall göras.

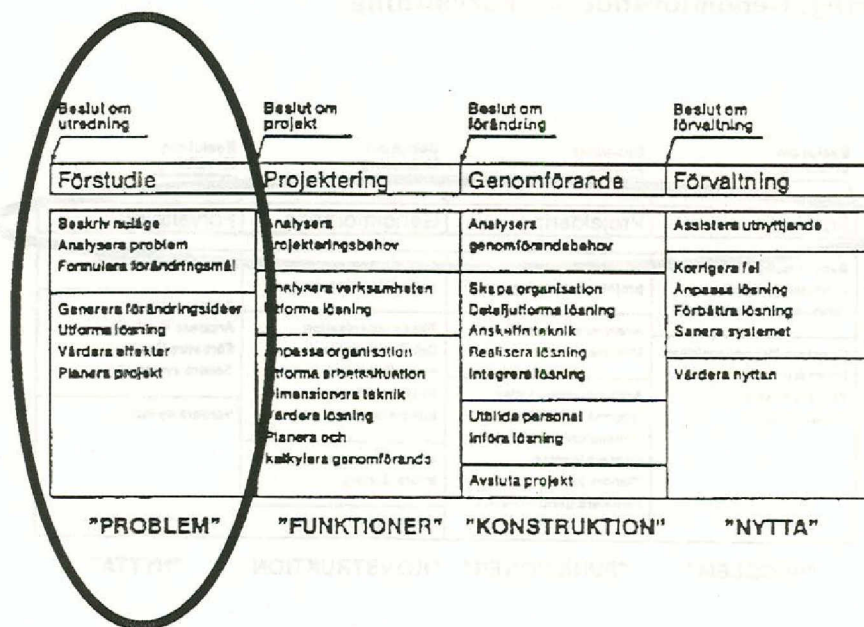
1.3 FÖRÄNDRINGSARBETE OCH FÖRVALTNING

Vår valda modell för förändringsarbete inom det administrativa området och för förvaltning är uppbyggd enligt de fyra faserna **Förstudie**, **Projektering**, **Genomförande** och **Förvaltning**.

Beslut om utredning	Beslut om projekt	Beslut om förändring	Beslut om förvaltning
Förstudie	Projektering	Genomförande	Förvaltning
Beskriv nu läge Analysera problem Formulera förändringsmål	projekteringsbehov Analysera verksamheten Utforma lösning	genomförandebhov Skapa organisation Detaljforma lösning Anskaffningsteknik Realisera lösning Integrera lösning	Kontrollera utgående Korrigera fel Anpassa lösning Förbättra lösning Sanera systemet
Generera förändringsidéer Utforma lösning Värdera effekter Planera projekt	Anpassa organisation Utforma arbetsituation Dimensionera teknik Värdera lösning Planera och kalkylera genomförande	Utbilda personal Införa lösning Avsluta projekt	Värdera nyttan
"PROBLEM"	"FUNKTIONER"	"KONSTRUKTION"	"NYTTA"

Stor hänsyn tas för att uppnå en samverkande helhet av arbetsuppgifter, den teknik som brukas, den kompetens och bemanning som behövs och hur den formella organisationen är uppbyggd.

FÖRSTUDIEFASEN



BESKRIV NULÄGE

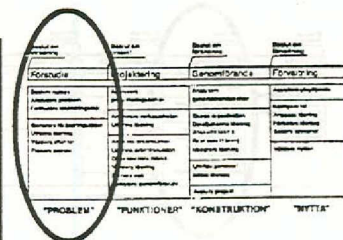
Syfte	Skapa gemensam förståelse för verksamhetens information.
Uppgift	Ansvara för att beskriva verksamhetens begreppsapparat. Medverka vid funktionsanalys med tonvikt på informationsbehoven i verksamheten.
Resultat	Grov datamodell med eventuell komplettering av volymer, ingående termer, frekvenser, ansvar etc. Avstämnda informationsbehov.

ANALYSERA PROBLEM

Syfte	Finna problemens bakomliggande orsaker vilka kan relateras till begreppen.
Uppgift	Medverka vid problemanalys.
Resultat	Prioriterad problemlista.

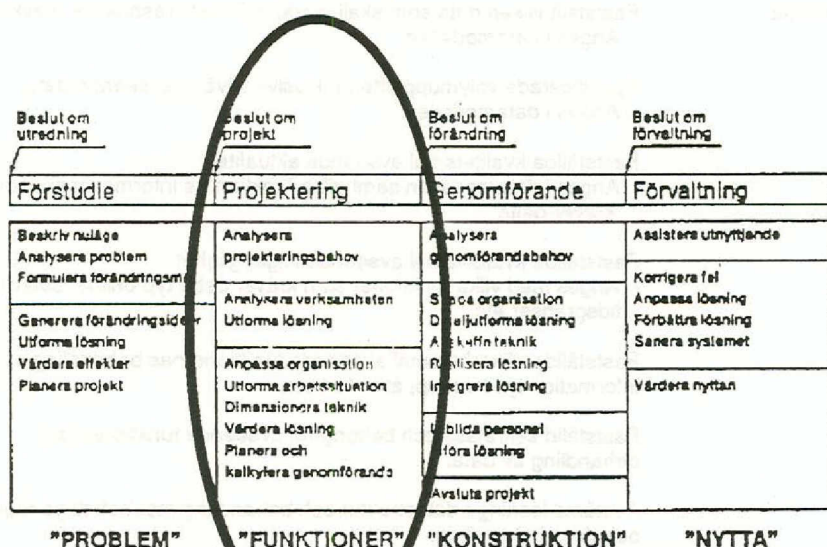
UTFORMA LÖSNINGSFÖRSLAG

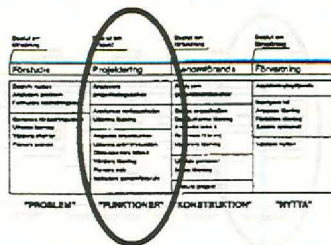
Syfte	Skapa en gemensam referensram för hantering av information i den förändrade verksamheten.
Uppgift	<p>Ansvara för att beskriva den förändrade verksamhetens begreppsapparat.</p> <p>Medverka vid funktionsanalys med tonvikt på informationsbehoven och dess kvalitet.</p> <p>Medverka vid framtagning av förslag på befattningar och arbetsorganisation vad gäller behörighet och hantering av data / information.</p> <p>Medverka vid framtagning av förslag på teknisk lösning.</p>
Resultat	<p>Grov datamodell med eventuella kompletteringar av volymer, ingående termer, frekvenser, kvalitet, ansvar etc.</p> <p>Avstämnda informationsbehov med avseende på innehåll och kvalitetskrav.</p> <p>Ansvarsmatriser avseende befattningar, arbetsorganisation, behörighet.</p> <p>Förslag på teknisk lösning.</p>



Ingen hänsyn har tagits till vilka befattningar eller organisationer som genomför uppgifterna. Detta kan i efterhand beskrivas med ansvarsmatriser.

PROJEKTERINGSFASEN





ANALYSERA PROJEKTERINGSBEHOV

Syfte	Stämma av projektaktiviteter mot angränsande delar av verksamheten med avseende på information.
Uppgift	Tillse att projektets aktiviteter överensstämmer med de riktlinjer som gäller för verksamheten.
Resultat	Granskade projektplaner.

ANALYSERA VERKSAMHETEN

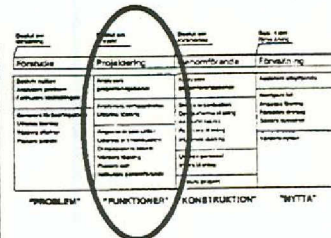
Syfte	Säkerställa en god informationsstruktur i framtida uppgifter så att verksamheten når uppsatta mål.
Uppgift	<p>Ansvara för att framtagna struktur av data överensstämmer med sido- och överordnade verksamhetsdelar.</p> <p>Medverka vid funktionsanalys med tonvikt på informationsbehoven.</p> <p>Tillse att begrepp som dokumenterats i datastrukturen används på ett korrekt sätt.</p>
Resultat	Datamodell som överensstämmer med funktioner inom verksamhetsområdet samt externa funktioners krav på information. Datamodellen är dessutom avstämd mot överordnade och angränsade datamodeller.

UTFORMA LÖSNING

Syfte	Utforma ett bra samspel mellan människa och teknik för att lösa specificerade uppgifter.
Uppgifter	<p>Medverka vid utformning av lösning vid såväl manuella som maskinella delar med tonvikt på hur data används.</p> <p>Tillse att behandling av data sker på effektivt och betryggande sätt samt att detta följer de riktlinjer som fastställts.</p> <p>Medverka vid utvärdering av "standardpaket".</p>
Resultat	<p>Fastställt vilken data som skall lagras manuellt respektive maskinellt.</p> <p>- Anges i datamodellen.</p> <p>Specificerade volymuppgifter, inklusive tillväxt, avseende data.</p> <p>- Anges i datamodellen.</p> <p>Fastställda kvalitetsmål avseende aktualitet.</p> <p>- Anges i datamodellen samt vilka funktioners informationsbehov kräver detta.</p> <p>Fastställda kvalitetsmål avseende tillgänglighet.</p> <p>- Anges med vilka funktioner som kräver detta typ online, satsvis, tidsgränser etc.</p> <p>Fastställda säkerhetsmål avseende funktionernas behandling av information typ backup, återstart etc.</p> <p>Fastställd sekretess och behörighet avseende funktionernas behandling av data.</p> <p>Avstämd lösnings datastruktur och behandling mot befintliga register och dess behandling.</p>
som	

ANPASSA ORGANISATION

Syfte	Genom en bra organisation skapa förutsättningar för en effektiv hantering av information.
Uppgift	Medverka vid utformning av befattningar och arbetsorganisation med tonvikt på hur data används. Tillse att behandling av data sker på ett effektivt och betryggande sätt samt att detta följer de riktlinjer som fastställts. Utverka regler och riktlinjer avseende säkerhet och sekretess avseende data.
Resultat	Volymer, frekvenser och bearbetning av data givet de befattningar som skisserats. - Anges i datamodell samt i skisserade rutinskisser. Reviderade kvalitetsmål avseende tillgänglighet givet de befattningar som skisserats. Reviderade mål avseende sekretess och behörighet givet de befattningar som skisserats.



UTFORMA ARBETSSITUATION

Syfte	Skapa en god arbetssituation avseende gränssnitt mellan människa och maskin.
Uppgift	Medverka vid utformning av gränssnitt. Medverka vid framtagning av prototyper. Ansvara för "sökvägsanalys" av lagrad data.
Resultat	Skisserad dialog. Sökväg i lagrad data.

DIMENSIONERA TEKNIK

Syfte	Skissera på fysisk lagring och åtkomst av data.
Uppgift	Ansvara för att skissera och dimensionera fysiska register. Ansvara för att värdera vilka datahanteringssystem som tillfredsställer lösningens behov. Görs endast då detta är en kritisk faktor för lösningen.
Resultat	Skisserade och dimensionerade fysiska register. Föreslagna databashanteringssystem.

VÄRDERA LÖSNING

Syfte	Granska framtagna lösningar med avseende på informationsbehandling.
Uppgift	Tillse att framtagna lösningar uppfyller de kriterier som utverkats.
Resultat	Granskade lösningar.

GENOMFÖRANDEFASEN

Beslut om utredning	Beslut om projekt	Beslut om förändring	Beslut om förvaltning
Förstudie	Projektering	Genomförande	Förvaltning
Beskriv rådliga Analysera problem Formulera förändringsmål	Analysera projekteringsbehov	Analysera genomförandebehov	Analysera utnyttjande
Generera förändringsidéer Utforma lösning Värdera effekter Planera projekt	Analysera verksamheten Utforma lösning Anpassa organisation Utforma arbetssituation Dimensionera teknik Värdera lösning Planera och kalkylera genomförande	Skapa organisation Detaljutforma lösning Anskaffa teknik Realisera lösning Integrera lösning Utbilda personal Införa lösning Avsluta projekt	Kontrollera fel Anpassa lösning Förvaltningslösning Samtliga systemet Värdera nyttan
"PROBLEM"	"FUNKTIONER"	"KONSTRUKTION"	"NYTTA"

ANALYSERA GENOMFÖRANDEBEHOV

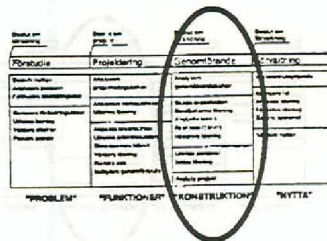
Syfte	Stämma av projektaktiviteter med avseende på informationsbehandling.
Uppgift	Tillse att lösningsspecifikationer uppfyller de kriterier som utverkats. Utverka standards, regler och riktlinjer av lösningen avseende informationsbehandling.
Resultat	Granskade projektplaner. Dokument "Handledning för genomförandet".

DETALJUTFORMA LÖSNING

Syfte	Utforma och konstruera lösningar med avseende på informationsbehandling.
Uppgift	Ansvara för utformning, konstruktion och dokumentation av logiska register. Ansvara för val av den för lösningen lämpligaste databashanteraren. Ansvara för utformning, konstruktion och dokumentation av fysiska register. Medverka vid utformning och konstruktion av program. Medverka vid utformning och konstruktion av gränssnitt.
Resultat	Fastställd detaljspecifikation avseende register, dvs konstruerade fysiska register. Dokumenteras i Data Dictionary (ADB-del).

ANSKAFFA TEKNIK

Syfte	Anskaffa den teknik som krävs.
Uppgift	Ansvara för att anskaffa, installera, testa och granska den teknik som krävs. Här avses såväl utvecklings-, utbildnings- som produktionsmiljö.
Resultat	Anskaffad, installerad och granska teknik.



REALISERA LÖSNING

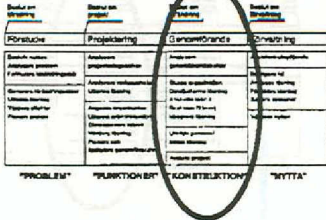
Syfte	Realisera (skapa, ladda, testa, dokumentera och granska) fysiska register. Realisera (dokumentera) manuella rutiner.
Uppgift	Ansvara för att realisera fysiska register. Ansvara för att realisera styrrutiner till fysiska register typ reorganisation, backup och återstart. Tillse att manuella rutiner och befattningar erhåller den säkerhet och behörighet som krävs. Tillse att ansvar avseende data tydliggörs och dokumenteras.
Resultat	Realiserade fysiska register. Realiserade styrrutiner. Fastställd säkerhet och behörighet. Fastställda ansvarsmatriser.

INTEGRERA LÖSNING

Syfte	Få lösningens olika delar att fungera tillsammans.
Uppgift	Ansvara för att skapa och ladda register i utvecklingsmiljön. Ansvara för att testa de styrrutiner som ingår. Ansvara för system- och driftsdokumentation med avseende på register och därtill hörande styrrutiner för reorganisation, backup och återstart. Medverka vid integrationstest.
Resultat	Granskade och dokumenterade fysiska register med därtill hörande styrrutiner. Merparten dokumenteras i Data Dictionary (ADB-del). Granska system- och driftsdokumentation.

UTBILDA PERSONAL

Syfte	Utbilda berörd personal.
Uppgift	Ansvara för att skapa och ladda utbildningsmiljön med avseende på register.
Resultat	Färdigställd utbildningsmiljö.



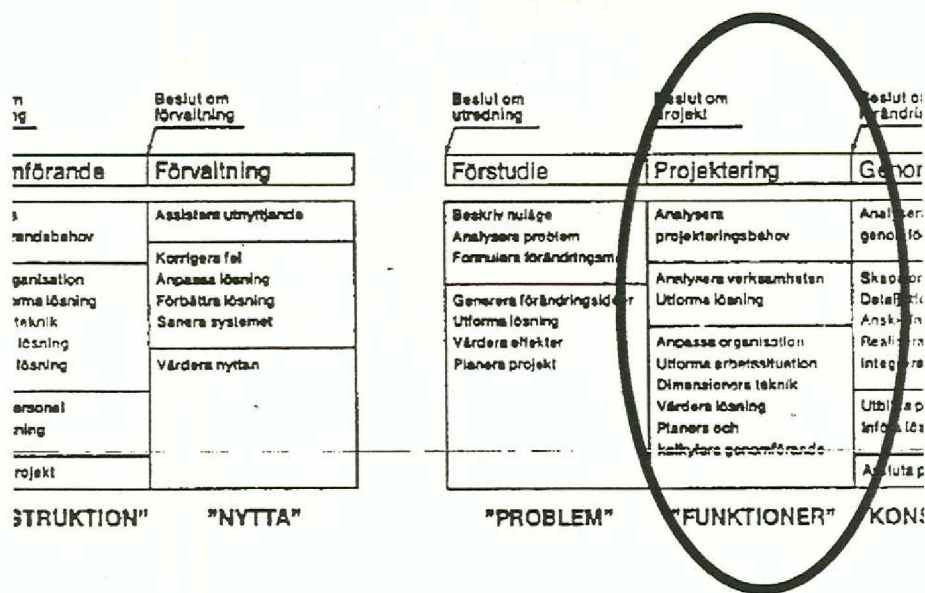
INFÖR LÖSNING

Syfte	Införa den nya lösningen, avveckla den gamla.
Uppgift	Ansvara för produktionsmiljön.
	Ansvara för att skapa och ladda fysiska register.
	Ansvara för produktionstest avseende register.
	Ansvara för tilldelning av behörighet till fysiska register.
	Ansvara för produktionstest avseende behörighet i verksamheten.
	Ansvara för trimning av register.
Resultat	Godkända, införda och trimmade register.
	Tilldelad behörighet.
	Konverterat registerdata, avveckling av gammalt registerdata.

AVSLUTA PROJEKT

Syfte	Förbered förvaltning av register.
Uppgift	Tillse att förvaltning av register kan ske på betryggande sätt.
	Tillse att behörighetsrutiner etableras i driftsorganisationen.
	Tillse att behörighetsrutiner etableras i linjeorganisationen.
Resultat	Utverka och dokumentera standards, regler och riktlinjer för förvaltning av register.
	Överlämnade register.
	Fungerande driftsverksamhet.
	Fungerande rutiner kring behörighet.
	Standards, regler och riktlinjer dokumenterade i "Handledning för förvaltning".

FÖRVALTNINGSFASEN



Eftersom förvaltningsfasens aktiviteter ej ännu fastställts, kan därför endast ett förslag på uppgifter lämnas. Dessa uppgifter är:

- Ansvara för att kvaliteten på lösningens detaljspecifikation vidmakthålls. Dokumenteras i Data Dictionary (ADB-del).
- Ansvara för att kvaliteten på verksamhetens datastruktur vidmakthålls. Dokumenteras i datamodeller och Data Dictionary (Verksamhets-del). Här ingår objekt, relation, termer, säkerhet, sekretess, behörighet, kvalitet etc.
- Ansvara för dokumentation avseende fysiska registers datortekniska uppgifter såsom, maskinamn, skivstation etc.
- Ansvara för prestanda, tillgänglighet, säkerhet, behörighet av fysiska register.
- Utverka standards, regler och riktlinjer avseende verksamhetsbeskrivande data och dess behandling.
- Utverka standards, regler och riktlinjer avseende konstruktion av ADB-register och dess behandling.
- Utverka standards, regler och riktlinjer avseende bruk av databashanterare.
- Tillse att "1000-mils-service" genomförs.
- Medverkar vid verksamhetsutveckling.
- Medverkar vid ändringsverksamhet i ADB-system.
- Medverkar som stöd till "Informations-center"-verksamhet.
- Ansvarar för utformning och konstruktion av "IC"-register.
- Ansvarar för realisering och införande av "IC"-register.
- Ansvara för anskaffning, installation, och test av aktuella databashanteringssystem.
- Ansvara för respektive databashanteringssystemers tillgänglighet och vidareutveckling.

2. METODER OCH BESKRIVNINGSTEKNIKER

Vi skall i följande avsnitt titta litet närmare på ett antal modelleringsansatser, som kan vara användbara vid dataadministrativt arbete.

Att arbeta med dataadministration innebär bl a att man behöver beskriva relevanta aspekter av verksamheten samt 'fenomen' i verksamheten om vilka man vill hålla information. Ytterst är det fråga om att beskriva information som finns eller hanteras i olika informationssystem och att beskriva hur denna information är uttryckt.

Modern dataadministrativ verksamhet beskriver inte bara, som förr, olika termer som används i ett informationssystem. Man vill också beskriva begrepp som ligger närmare verksamheten i sig, så att de begrepp som hanteras i informationssystemen kan "hängas upp" i ett sammanhang och därigenom kunna definieras och förklaras bättre.

I en modern DA-verksamhet vill man alltså även beskriva begrepp som verksamhetsfunktion (inom vilken en objektvärld kan existera samt där informationssystem kan existera), informationssystem (till vilket databaser kan höra), objekt och "semantiska" relationer mellan objekt (till vilka attribut och termer kan höra), etc.

För dataadministration, liksom för andra typer av 'förvaltande' verksamheter, har vi behov av att beskriva saker på ett ändamålsenligt och exakt sätt. Vi vill kunna beskriva saker som verksamheten uppfattar dem (eller borde uppfatta dem).

Modellering har på många håll blivit uppmärksammat bland systemutvecklare och kunskapen om hur man skapar modeller för verksamheter och förvaltningar har ökat. Detta gäller även internationellt.

2.1 Modeller/modellering

Inom många områden har man en tradition av att bygga modeller av det man tänker skapa innan konstruktion och byggande påbörjas.

Modeller skapas för att belysa olika aspekter av det som skall skapas. Det kan t ex gälla skalmodeller inom bil- och varvsindustrin och olika former av mer abstrakta modeller (symbolmässiga modeller) inom byggsektorn och kraftindustrin.

I den senare formen använder man ofta **symboler** för att representera aspekter som man är intresserad av. Symboler kan t ex vara matematiska uttryck eller grafiska symboler. Grafiska symboler kan vara mer eller mindre "ikoniska", dvs mer eller mindre likna originalet.

Inom informationsbehandlingsområdet har man behov av att beskriva en mängd abstrakta företeelser som verksamhetsdelar, 'information', program, olika former av datalager, informationsflöden, operationer, sekvenser mellan operationer, etc. Detta ställer speciella krav på modelleringsansatserna, inte minst måste de ha modelleringsspråk, som är tillräckligt kraftfulla för att uttrycka det man är intresserad av.

2.1.1 Kategorier av modeller

Några vanliga kategorier av modeller är:

- Ikoniska modeller
- Analoga modeller
- Symboliska modeller
- Verbala modeller

Ikoniska modeller

Ikoniska modeller liknar originalet på ett konkret och påtagligt sätt. De ser ut som den verkliga förlagan och har samma egenskaper som denna, men har ofta annan storlek - en annan skala. De kallas därför ibland också för skalmodeller.

Ikoniska modeller kan vara 2-dimensionella (kartor, fotografier, layouter, etc), 3-dimensionella (skalmodeller av bilar, båtar, etc) och innehålla tiden som en dimension (radar, film, modelljärnväg, etc).

Några av de uppenbara fördelarna med ikoniska modeller är att de är lätta att uppfatta och förstå. Genom likheten med förlagan är representationsreglerna lätta att sätta sig in i.

Däremot är de ofta arbetskrävande att framställa och förändra. De är också specifika till sin natur.

Konceptet bakom ikoniska modeller - likhet med originalet - ligger också bakom pilotanläggningar, testmarknader, etc och inom informationsbehandlingsområdet används ofta prototyp- och experimentsystem.

Analoga modeller

Vid analoga modeller låter man en egenskap i modellen motsvara en annan egenskap i verkligheten. Man använder sig av strukturer som på något sätt liknar de relationer, som valts ut för avbildning, men som inte konkret liknar dem.

Olika kurvor och diagram är analoga modeller, som använder geometriska storheter för att representera egenskaper och deras samband.

Hydrauliska system kan användas som analoger för elektriska system, etc, höjdlinjer på en karta är lutningsanaloger, etc.

Analoga modeller är mindre konkreta än ikoniska modeller och därmed inte så tydliga och lätta att förstå. De inkluderar bara ett begränsat antal aspekter, vilket leder till att antaganden måste göras. De är dock lättare att manipulera och är inte lika specifika som ikoniska modeller.

Symboliska modeller

Symboliska modeller använder sig av bokstäver, tal och andra typer av symboler för att representera en verklighet. De är de mest abstrakta modellerna, men de mest vanliga.

De kan vara kraftfulla hjälpmedel för att manipulera och analysera relationer, men behandlar avsevärt färre aspekter än de ikoniska modellerna.

För att sätta sig in i en symbolisk modell, måste man känna dess representationsregler. Dessa kan vara underförstådda eller uttryckligen formulerade.

Det kan vara lätt att glömma modellens begränsningar - faran ökar för att "man lägger in för mycket" i modellen.

Verbala modeller (beskrivning)

Modeller som använder verbal text - naturligt språk.

Fördelarna med verbala modeller är att de kan vara innehållsrika och kan hantera extremt komplexa situationer. Naturligt språk ger också en flexibel användning. Verbala modeller är svåra att manipulera och kan vara oprecisa.

2.1.2 Modelleringsansats/modelleringspråk

Med en modelleringsansats brukar man mena:

- ett "modelleringspråk", och
- en metodik - ett tillvägagångssätt att formulera, framställa och förändra ett schema/en modell

Ett modelleringspråk består av:

- en uppsättning modelleringsbegrepp
- en grafisk "notation"
- en verbal/textuell notation som kan vara mer eller mindre strukturerad och formell.

I en grafisk notation, som har uppenbara fördelar ("en bild säger mer än tusen ord"), kan man dock i regel bara uttrycka vissa enkla regler och villkor.

Om man vill uttrycka mer komplicerade och omfattande regler och villkor ökar behovet av en mer "uttrycksfull" verbal/textuell notation i tillägg till den grafiska beskrivningen, och som, beroende på önskad grad av precision, kan vara mer eller mindre formell.

2.1.3 Referensramar och kunskaper

Modelleringsansats bör väljas utifrån vem som skall använda den och till vad. Olika människor har olika referensramar när det gäller att förstå uttrycksformen för en viss modelleringsteknik.

Det är alltså inte bara fråga om att tala om uttrycksförmåga, utan dessutom måste man fråga för vem det skall uttryckas?

Man måste alltså beakta ett antal aspekter när det gäller att diskutera lämplighet hos modelleringsansatser:

- Vilken kunskap och erfarenhet har de som skall modellera av att abstrahera och modellera över huvudtaget?
- Vad behöver uttryckas och vilken uttrycksförmåga behöver modelleringsansatsen därmed ha?
- Vilka referensramar har de som skall arbeta med modellerna. Vilka typer av modelleringstekniker är man van vid (skalmodeller, symbolmodeller, olika typer av modelleringsbegrepp, etc)?
- Vilka behov av överblick har man? Hur kan detta balanseras med behovet av att se detaljer?

Om man vill att informationssystemanvändare i ett företag skall vara med och beskriva önskat informationsinnehåll i ett nytt system, så bör de begrepp som används i modelleringstekniken för att beskriva informationsinnehåll vara väl anpassade till det speciella ändamålet och till personerna ifråga, så att man inte tröttnar ut de som medverkar i utformningsarbetet med besvärliga och olämpliga beskrivningssätt.

2.1.4 Användningsområden för modellering

Vi har ställt upp ett antal krav på modelleringsansatser som vi därvid kan relatera till:

- dels har dataadministrativ verksamhet speciella krav på modelleringsansatser och deras lämplighet
- dels så skall de modeller som skapas inom ramen för dataadministrativ verksamhet kunna användas inom andra områden som är starkt kopplade till dataadministrativ verksamhet och tvärtom

Modellerna skall vara återanvändbara inom andra till dataadministration intilliggande verksamhetsområden. Det torde i själva verket vara så, att för att man skall få effekter av dataadministration så måste dessa synergieffekter uppstå, annars erhålls inte TILLRÄCKLIGA effekter.

Sådana angränsande områden kan t ex vara:

- Verksamhetsmodellering på övergripande nivå.

Här beskrivs ofta dels vilka verksamhetsfunktioner som är aktuella, men också vilka viktiga objekt som finns eller kommer att finnas i verksamheten, och omkring vilka information behöver kunna hanteras.

- Informationsanalys.

Inom systemutveckling bör de informationsbehov som identifierats i en verksamhet brytas ner i detalj så att man kan se exakt hur denna information är uppbyggd, vilka verksamhetsobjekt informationen refererar till och hur informationen skall representeras.

- Datamässiga avgöranden och design avseende hur önskad information skall vara representerad i de informationssystem som byggs.

Således vilka 'värdemängder' eller vilka domäner som finns eller behöver kunna användas för att uttrycka information.

- Beskrivning av hur data skall ligga i olika databaser.

Hur data skall distribueras på lämpligt sätt för att bli kunna vara effektiva att hitta och uppdatera i enlighet med de prestandakrav som ställs mot aktuellt system ifråga om svarstider och tillgänglighet.

2.2 Krav och önskemål på modellering

Den typ av modellering som har beskrivits här kallas ofta konceptuell modellering. Här följer en uppräknig av vilka krav och önskemål som finns på konceptuell modellering.

Konceptuell modellering bör/skall utgöra:

1. Ett hjälpmedel gemensamt för slutanvändare, dataadministratörer och systemutvecklare, dvs ett och samma hjälpmedel för utveckling, konstruktion, drift och användning av systemet.
2. Ett hjälpmedel för storskalig informationshantering, dvs för att överblicka data på alla nivåer i mycket stora organisationer.
3. Ett hjälpmedel som kan medverka till ingenjörsmässigt, standardiserat och likformigt underhåll, dvs ett industriellt sätt att se på informationshantering och systemutveckling.
4. Ett hjälpmedel som utgör den centrala filosofin i ett informationshanteringsverktyg - inget glapp mellan analys och konstruktion.
5. Ett hjälpmedel som är uppbyggt på ett sådant sätt att det är pedagogiskt och självinstruerande för att söka vidare i ett datalager; välj först ett adekvat datalager och sök sedan djupare i detta.
6. Ett hjälpmedel som gör det möjligt att helt skilja logisk struktur från fysisk struktur av data. kan då ställas på datalagrets logiska struktur, medan effektivitetskrav kan ställas på datalagrets fysiska struktur. Med ingen kompromiss mellan överblickbarhetskrav och effektivitetskrav på datalagret.
7. Ett hjälpmedel (eller en del i ett sådant) för att kunna integrera system och därigenom skapa förutsättningar för att distribuera kvalitetsdeklarerade data.

8. Ett hjälpmedel som bildar grunden i en datakatalog.
9. Ett hjälpmedel som ger stöd för att anpassa företagets informationsbehandling vid omstrukturering av verksamheten (från central till decentraliserad organisation och vice versa).

2.3 Syfte med datamodellering

2.3.1 Inledning

Skillnaden mellan att kartlägga och representera verksamhetsorienterad kunskap och att realisera en lösning med hjälp av ADB-teknik är ett dilemma. Hur väl man lyckas att harmonisera ett verksamhetsorienterat och utvecklingsorienterat angreppssätt har avgörande betydelse för datoriseringen.

En ADB-verksamhet brukar delas upp i olika delar av praktiska skäl. När man vill veta hur allt fungerar spelar dessa gränser mindre roll. Helheten och de grundläggande problemen vid datorisering är det som fångar intresset.

Utvecklingsmodeller, arbetsfördelning, ansvarsfördelning, projektstyrning, etc har dock en indirekt påverkan på hur vi uppfattar problemet med datorisering. Det är svårt att skilja på dessa två aspekter. Ändå är det nödvändigt att titta på de grundläggande problemen för att pröva hållbarheten hos nya metoder.

Hur ska man uttrycka och representera kunskap om en verksamhet som underlag för datorisering? Hur skall denna kunskap hållas levande? Hur skall man skapa effektiva redskap eller verktyg för att hantera denna kunskap? Hur skall denna kunskap komma till uttryck i tekniska lösningar?

2.3.2 Tolkning och modell

Att göra en modell innebär många saker. För det första skiljer man på sakområdet (det som modelleras) och modellen. Modellen är en abstraktion och kategorisering av det som modelleras och detta i sin tur har styrts av hur väl man förstår sakområdet och det bakomliggande syftet eller vad man skall använda modellen till. En viktig egenskap hos modellen är att dokumentera överenskommelser, ett gemensamt synsätt.

Eftersom modellen skall representera ett sakområde så måste den utgå från objekt och företeelser som kan identifieras, avgränsas och definieras ("Objektmodellering"). Rent filosofiskt är inte sakområdet objektivt givet utan vi vet att vår kunskap om sakområdet påverkar vad vi ser och hur vi modellerar.

Till varje modell hör en tolkning av modellen t ex namn i modellen kopplas till företeelser och objekt i sakområdet. Tolkningsföreskrifter kan finnas dokumenterade vid sidan av modellen.

Modellens stora värde ligger i att den ger en struktur som utsäger inom sakområdet kan hängas upp på. Strukturen bygger på en kunskap om sakområdet och är därför ytterst värdefull.

Vid processtyrning är det självklart att modellen är grunden för alla tekniska lösningar. Annars skulle inte processtyrningen fungera exakt som man vill. För administrativa sakområden har det hittills inte varit lika självklart att utgå från en dokumenterad kunskap om sakområdet i form av en modell. Mindre formella och approximativa metoder har ofta använts.

Varje datoriserad lösning bygger dock på en modell och den praktiska användningen bygger på en tolkning av samma modell.

2.3.3 Resurs och system

Hur stor del av detta sakområde skall modelleras? Kan det göras på olika sätt (olika urval, olika detaljering, etc)? Detta är grundläggande frågor som varit upphov till dagens problem med att integrera system som utvecklas på olika avdelningar i ett företag.

Genom urval kan många kombinationer uppstå som synbart är nya modeller. De bygger dock på en gemensam större modell för hela sakområdet (även om den inte finns konstruerad) som "försörjer" olika möjliga vyer.

Att skapa ordning och reda mellan ett antal system kräver att vi går tillbaka till verksamheten och en modell av denna. Detta bygger på två viktiga insikter, dels att verksamhetens modell är viktig för förståelse av ett system, dels att verksamhetens modell kan användas för att integrera flera system (gemensam referensram).

2.3.4 Begrepp och data

Datamodellering omfattar den del av en verksamhetsmodell som behövs för att beskriva begrepp och data.

Begrepp skapas i en konceptuell modell och står för företeelser eller objekt i sakområdet. Data är informationsbärare för begrepp. För att tolka data behövs tolkningsinstruktioner eller beskrivning av data. En definition av data innehåller förklaring av det begrepp som data skall representera.

Den konceptuella modellen syftar både till att skapa överblick över begrepp och att skapa entydighet mellan begrepp. Begrepp förekommer i "sammanhang" och måste alltid kvalificeras med detta sammanhang t ex begrepp som används för att uttrycka egenskaper hos ett objekt.

Begrepp konstrueras inte isolerade från varandra. Tvärtom skall begreppen kunna stödja varandra. Man kan skilja på generiska begrepp (typ datum) och applikationsbegrepp (typ leveransdatum). De generiska begreppen bildar en bas vid definitionen av applikationsbegrepp.

De minsta dataenheter som definieras - dataelementtyper - kan förtecknas och beskrivas i en dataelementtypförteckning. Hur data används tillsammans kan förtecknas och beskrivas i en meddelandetyppförteckning och en meddelandesyntax. Hur begrepp hör ihop med objekt i verksamheten visas i den konceptuella modellen.

2.3.5 Sammanfattning av syftet

Det är viktigt att inte blanda ihop datamodellering med organisatoriska åtgärder för att införa dataadministration. Datamodellering som teknik fungerar oavsett hur administrativa regler sätts, t ex om modellering görs per system, per verksamhetsområde eller för hela verksamheten. Ordning och reda behövs både i enskilda delar och för alla delar tillsammans.

En konceptuell modell kan både utnyttjas som en karta över verksamheten och dela upp den i lämpliga "affärsområden" eller göras tillräckligt detaljerad för ett område för att tjäna som konstruktionsunderlag. Att modellen kan nivåändelas är en väsentlig egenskap liksom att de olika nivåerna kan kopplas samman.

Syftet är att skapa stabila strukturer som data kan hängas upp på och som kan fungera som en bas (ev en tänkt bas) för försörjning av data till system med sitt informationsbehov (en "vy"). Modellens stora fördel är att dess struktur och begrepp är tillräckliga för att svara mot ett sakområde. Det svarar mot sakområdets komplexitet.

I synsättet ligger en stor möjlighet att utveckla data som en resurs. Det är den första utvecklingsmetod som inte är inriktad mot systemutveckling utan mot resursutveckling och mot försörjning av ad-hoc behov.

Metoder för systemutveckling har äntligen blivit något annat än att utveckla isolerade system för att uppfylla tillfälliga behov som låses fast i system. Verksamhetens struktur uttryckt i en konceptuell modell är den stabiliserande gemensamma faktorn. Den konceptuella modellen ger ett disciplinerat sätt att representera denna kunskap.

2.4 Vad är konceptuell modellering?

Inom området datamodellering/konceptuell modellering finns en omfattande flora av begrepp för modeller av olika slag. Man talar bl a om:

- datamodeller
- databasmodeller
- semantiska datamodeller
- objektmodeller
- begreppsmodeller
- konceptuella modeller

Alla har det gemensamt att de, med hjälp av en viss, utvald normativ uppsättning av modelleringsbegrepp beskriver eller avbildar vissa aspekter av den del av en verksamhet som ett informationssystem försörjer.

Skillnader som kan finnas mellan de olika typerna av modeller kan beröra två aspekter:

- omfattningen av modellerna dvs vilka aspekter av verksamheten/verkligheten som skall beaktas och avbildas i modellen
- hur pass implementerings-/datoroberoende de använda modelleringsbegreppen är.

2.4.1 Konceptuell modell

Man gör i denna en distinktion mellan modell och schema.

ISO (Internationella StandardiseringsOrganisationen), har tagit fram en rapport som behandlar begrepp och språk för konceptuell modellering ("Concepts and Terminology for the Conceptual Schema and the Information Base").

Det man vanligen brukar kalla konceptuell modell eller datamodell motsvarar med ISO-terminologi ett konceptuellt schema. En konceptuell modell skall här också innefatta en s k informationsbas, som innehåller registrerade fakta och händelser i verksamheten/verkligheten.

Enligt ISO skall ett konceptuellt schema inte bara innehålla en strukturbeskrivning av innehållet i informationsbasen utan också ange regler och villkor för hur informationsbasen får förändras dvs vilka tillägg, borttag och ändringar som är giltiga.

Vid konceptuell modellering, enligt ISO:s synsätt, syftar man till att åstadkomma en utökad beskrivning jämfört med den man vanligen gör vid "traditionell" datamodellering. Det finns riktlinjer för vad denna utökade beskrivning skall omfatta i form av två principer som kallas 100 %-principen och konceptualiseringsprincipen.

Det kan vara intressant att här återge innehållet i dessa två principer:

- 100 %-principen

I det konceptuella schemat skall alla relevanta, generella, statiska och dynamiska aspekter, dvs alla lagar, regler och villkor som finns i verkligheten/verksamheten, beskrivas. Ett informationssystem definieras av de regler och villkor som beskrivs i det konceptuella schemat och inte av regler som beskrivs på annat sätt, t ex i applikationsprogram.

- Konceptualiseringsprincipen

Ett konceptuellt schema skall bara omfatta konceptuellt relevanta aspekter, statiska och dynamiska, av verksamheten/verkligheten, och därmed exkludera alla aspekter som har med datarepresentation, dataorganisation, åtkomst, lagring, format, presentation, etc, att göra.

I ett databehandlingssystem ligger viktiga antaganden om den bakomliggande verksamheten inbyggda i datastrukturer, programkod och lagringsstrukturer.

Konceptuell modellering syftar till att göra dessa antaganden om verksamheten explicita och uppfattbara.

Det viktiga att konstatera här är den starka betoningen av det bara är **konceptuellt relevanta** aspekter som skall tas med - dvs alla implementerings- och datorberoende aspekter skall "silas bort" - att **alla regler** i verksamheten skall avbildas och ingå i modellen - och att **dynamiska** aspekter skall beaktas.

Det säger sig självt att en modelleringsansats för konceptuell modellering i denna mening måste förse modellaren med en rikare uppsättning modelleringsbegrepp och ha ett mer utvecklat modelleringsspråk för t ex beskrivning av regler, än en modelleringsansats som syftar till att beskriva en delmängd av detta.

En avgränsning som görs i många modelleringsansatser är att enbart beakta de statiska aspekterna och därmed förknippade regler av enklare typ.

2.4.2 De tre "klassiska" datamodellerna

Man brukar tala om tre "klassiska" datamodellerna, tre huvudtyper av datamodeller som de flesta databashanterare som finns tillgängliga bygger på:

- CODASYL-(nätverks-)datamodellen
- den hierarkiska datamodellen
- relationsdatamodellen

Var och en av dessa ansatser har sin egen uppsättning struktureringsbegrepp för att strukturera data i en databas. De brukar därför också ibland kallas för databasmodeller.

Nätverksmodellen t ex strukturerar data i posttyper och set-typer, den hierarkiska modellen (en begränsad nätverksmodell) i poster, segment, relationsmodellen i tabeller.

De databasorienterade ansatserna använder modelleringsbegrepp som har sitt ursprung i begrepp som används för att strukturera data i filer och databaser på datorers sekundärminnen.

De anses därför allmänt numera inte fristående vara idealiska modelleringsansatser på den konceptuella nivån, där med ISO-terminologi "endast konceptuellt relevanta aspekter skall beaktas".

Av dessa har dock relationsansatsen med sina enkla struktureringsprinciper och välutvecklade matematiska förankring nått stor spridning och användning och olika påbyggnader har gjorts i syfte att kunna göra datadefinitionerna semantiskt rikare.

2.4.3 Datamodellering/semantiskt datamodellering

När man idag talar om datamodellering menar man egentligen inte modellering enbart med hjälp av någon av de tre klassiska datamodellerna.

Om den renodlade relationsansatsen används så görs det alltid i kombination med någon objektorienterad ansats. Den objektorienterade modelleringen föregår den "tabellorienterade" modelleringen.

I en systemutvecklingsprocess måste en övergång göras från den problemorienterade konceptuella modellen till en databasmodell - ett schema för en databas. Beroende på hur pass rik och "konceptuellt inriktad" den konceptuella modellen är, blir denna övergång mer eller mindre komplicerad.

2.5 Konceptuell modellering - grundbegrepp

2.5.1 Objekt

De olika företeelser i en verksamhet som man vill kunna göra olika typer av utsagor om representeras i en konceptuell modell av objekt (entiteter). Objekt kan ha samband - associationer.

Kategorier av objekt

Ofta är det dock så att en modelleringsansats har fler modelleringsbegrepp än så - man kategoriserar objekt i t ex objekt/entitet, attribut och förhållande/relation. På engelska: Entity - Attribute - Relationship, vilket leder fram till de s k EAR-modelleringsansatserna.

Om en modelleringsansats använder sig av flera kategorier av objekt uppstår genast frågan vid modellering, vad som skiljer dessa olika kategorier åt - när skall man representera en företeelse som ett objekt, som ett förhållande, som ett attribut?

Skillnaden definieras ofta med hjälp av regler för objektens existens i modellen. Ett objekt/en entitet kan ha en av andra objekt oberoende existens i modellen, medan ett förhållande bara kan existera om de objekt som förhållandet gäller mellan redan existerar.

Andra modelleringsansatser, som även syftar till att fånga dynamiska aspekter, kan ha ytterligare kategorier av objekt, t ex händelse.

Modelleringsansatser som använder samma modelleringsbegrepp, definierar dem inte alltid på samma sätt. Olika restriktioner kan gälla, t ex måste ett attribut i en EAR-modell vara "funktionellt" dvs det får bara ha ett värde. Vill man t ex representera att en person (entitet) kan ha flera telefonnummer så får man antingen, om det är semantiskt riktigt, introducera flera attribut, t ex bostadstelefonnummer och arbetstelefonnummer, är båda arbetstelefonnummer får man introducera arbetstelefonnummer-1 och arbetstelefonnummer-2 eller kanske t o m introducera telefon som en entitet och representera det med hjälp av förhållanden mellan person-entiteter och telefon-entiteter. Andra modelleringsansatser åter tillåter "mångvärda" attribut.

Typer och klasser av objekt

I en konceptuell modell vill man i regel inte representera enskilda objekt och enskilda förhållanden utan istället typer eller klasser av objekt och förhållanden.

Att klassificera objekt och bestämma vilka objekt som skall tillhöra en viss objektclass är en viktig modelleringsaktivitet. Att klassificera innebär att bestämma klass och typ för ett objekt eftersom man brukar säga att en objektclass definieras av en objekttyp.

Ett enskilt/individuellt objekt sägs vara en instans/förekomst av en objekttyp alternativt en vara medlem i en objektclass.

Om typbegreppet inte används explicit görs ändå motsvarande distinktion genom att man talar om objekt respektive objekt förekomst.

Objekt och namn på objekt

I många modelleringsansatser skiljer man mellan objekt och namn på objekt. Namn motsvarar här term. 'ABC 378' kan sägas vara ett namn för ett bilregistreringsnummer, '412 99' ett namn för ett postnummer, etc.

Objekten kallas då "icke-lexikaliska objekt" medan namnen kallas "lexikaliska objekt".

I NIAM (Nijssen Information Analysis Method) som är ett exempel på en binär, objektorienterad modelleringsansats, används modelleringsbegreppen icke-lexikalisk objekttyp (NON Lexical Object Type = NOLOT) och lexikalisk objekttyp (Lexical Object Type = LOT).

I CMOL (Conceptual MODelling Language) kallas lexikaliska objekttyper för datatyper, i EMOL (Ericsson MODelling Language) för domäner eller värdemängder.

Generalisering/specialisering

Generalisering innebär att man, genom abstraktion, bildar mer generella begrepp. De samband som gäller mellan de speciella begreppen och det generella begreppet kallas generiska samband eller ISA- (is a/är ett)-samband. Motsatsen till generalisering kallas specialisering.

Om man säger att begrepp, eller generiska objekt, karaktäriserar objekttyper, kan vi som exempel säga att objekttypen PERSON är en generalisering av objekttyperna MAN och KVINNA, alternativt att MAN och KVINNA är specialiseringar av PERSON.

I en generisk hierarki ärvs egenskaper och attribut som de generiskt överordnade objekten har av de specialiserade objekten. Har personer ett personnummer så har också män och kvinnor personnummer. Generiska samband används i många modelleringsansatser. Terminologin varierar men den grundläggande andemeningen är densamma. T ex talar man ibland om superobjekt och subobjekt. Så t ex är i exemplet ovan PERSON ett superobjekt och MAN och KVINNA subobjekt.

Aggregering

Aggregering innebär att ett antal attribut aggregeras eller grupperas till ett objekt. Attributen ses som komponenter eller delar till objektet. PERSON kan t ex ses som ett aggregat av namn, adress, telefon, etc.

Komplexa objekt

Komplexa objekt är objekt som har en "innesluttande" struktur - som består av andra objekt.

Komplexa objekt kan vara användbara för två syften: dels för att under analysens gång inte behöva göra modellen detaljerad ner till lägsta nivå omedelbart, utan kunna avvakta med att "dekomponera" objekt i den till ett senare skede. På så sätt skulle man kunna tillämpa ett hierarkiskt nedbrytningsförfarande vid modellering.

De kan också vara användbara för att dölja detaljer i den detaljerade modellen vid presentation för olika användar- och intressentkategorier, dels under modelleringens gång och dels för att tjäna som kartor över databasens innehåll när systemet är i drift.

Det krävs dock att sematiken i och notationen för komplexa objekt är väl definierad för att de skall kunna användas på ett stringent sätt och utgöra en användbar "abstraktionsmekanism".

2.5.2 Regler

Vilka olika typer av regler kan man vilja urskilja i en konceptuell modell och hur kan man beskriva dem?

En del regler byggs in i de använda modelleringsbegreppen. En del regler kan beskrivas i den grafiska notationen. Andra regler beskrivs i annan verbal - informell, formell form.

En del regler kan vara lokala i den meningen att de gäller och kan knytas till en viss objekttyp t ex, andra regler återigen är av mer global natur - de involverar fler objekttyper och samband som kan förekomma mellan dem.

En uppdelning kan göras i restriktionsregler och härledningsregler. Restriktionsregler anger vilka begränsningar som skall gälla för modellen och bildar grunden för att kunna avgöra om ett påstående om objektsystemet är korrekt eller rimligt.

Restriktionsregler som förekommer i de flesta modelleringsansatser och som ofta ingår i den grafiska beskrivningen är:

- avbildningsrestriktioner för relationer/attribut
- optionalitet eller totalitet/partialitet för relationer/attribut

2.6 Egenskaper hos modelleringsansatser

Det finns många egenskaper som skulle kunna användas för att göra typindelningar och jämförelser mellan modelleringsansatser. Nedan listas några exempel på grupper av egenskaper.

Egenskaperna är i huvudsak karakteriserande och i sig inte värderande, dvs de efterfrågar bara om vissa saker behandlas eller finns med i en modelleringsansats. Men det är förmodligen bra om det finns kriterier för kvalitetsmässigt goda modeller, det är bra om det finns koppling till metodik för databaskonstruktion, etc.

1. SYSTEMUTVECKLINGSBEGREPP OCH KOPPLINGAR

Vad säger ansatsen om t ex modelleringsnivåer? Vilka nivåer - hur relateras de till varandra? Förhållande till problem- och verksamhetsanalys, motsvarande modeller? Förhållande till databaskonstruktion? Finns flera distinkta arbets- och utvecklingssteg för att ta fram en modell?

2. KONSTRUKTIONSMETODIK OCH KVALITETSSTYRNING

Vilka principer rekommenderas för modellkonstruktion: från allmän diskussion om objektsystemet, från specificerade informationskrav, kombination? Finns metodik för konceptualisering och integration? Hur valideras modellen? Hur kontrolleras fullständigheten? Hur kontrolleras konsistens? Finns det kriterier för kvalitetsmässigt 'goda' modeller? Finns koppling till eller metodik för databaskonstruktion?

3. PRAKTISKA ASPEKTER OCH VÄRDERINGAR

Är modelleringsansatsen beroende av tillämpningsområde? Hur bedöms användarnas (utbildade resp. icke utbildade) möjligheter att medverka, förstå och kontrollera specifikationsarbetet? Erforderlig utbildningsvolym? Modelleringsansatsens flexibilitet, 'modelleringsfrihet', 'beteckningsfrihet', etc? Finns datorhjälpmedel? För vad finns datorstöd?

4. TILLÄMPADE MODELLERINGSBEGREPP

Entitet/entitetstyp eller motsvarande? Förhållande/förhållandetyp eller motsvarande? Händelse/händelsetyp - dynamiska aspekter? Attribut: funktionella, flervärda, strukturerade? Globala attribut, "mängdattribut"? Distinktion mellan lexikaliska och icke-lexikaliska objekt (dvs "names of things" och "things themselves")? Regler och villkor? Hantering av dynamiska samband: hur hanteras tidsdimensionen och referens till data vid olika tidpunkter?

5. MODELLERINGSSPRÅKET

Finns formellt definierad syntax för arbete med begrepp enligt 4) ovan? Speciellt: hur definieras regler och villkor? Finns grafisk notation? Vad kan/kan inte uttryckas med den notationen? Uttrycksfullhet ("expressive power"/"richness")?

Det är självklart inte möjligt, och inte heller syftet, att i denna rapport göra en jämförelse mellan olika modelleringsansatser på alla dessa punkter.

2.7 Typer av modelleringsansatser

Man kan möjligen diskutera ett antal grova typer av modelleringstekniker som är tänkbara för dataadministration:

- Term(namn-)baserade modelleringstekniker
- Enkla associativa objekt- och begreppsmodelleringstekniker
- Binära objektorienterade modelleringsansatser
- Objektorienterade modelleringsansatser av EAR-typ
- Avancerade konceptuella modelleringsansatser med språk för definition av statiska och dynamiska regler

2.7.1 Term(namn-)baserade modelleringstekniker

I dessa modelleringstekniker används inte något objektbegrepp. Man relaterar i dessa modelleringstekniker enbart informations- eller dataenheter till varandra.

Ett exempel är att komponentanalys bedrivs. Man bryter ner större informationsenheter till mindre enheter - dataflöden i deldataflöden och vidare till dataelement. Man bryter ner informationsbehov eller informationsmängder i delinformationsbehov/mängder, meddelandetyper, termgrupper och enkla termer.

Den renodlade relationsansatsen kan ses som en exemplifiering av denna typ av modelleringsteknik. Man beskriver relationer mellan dataenheter i form av sk funktionella beroenden. På grundval av dessa kan man sedan

dekomponera en "universal"-relation i mindre relationer eller man kan utgående från en uppsättning funktionella beroenden mellan dataenheter/attribut/termer syntetisera fram en mängd relationer för ett databas-schema.

2.7.2 Enkla associativa objekt- och begreppsmodelleringstekniker

Dessa modelleringsansatser, som i regel används i tidiga skeden av en systemutvecklingsprocess, analyserar och beskriver samband mellan objekt eller begrepp. Sambanden är semantiska dvs de har en innebörd som ges av de namn man ger dem.

Man gör här i regel ingen distinktion mellan olika kategorier av objekt (entitet, förhållande, händelse) eller skiljer mellan begrepp som refererar till objekt av icke-lexikalisk eller lexikalisk typ. Syftet är att fånga begrepp, att diskutera kring begreppen, att notera begreppen, att vara en "inkörsport" till fortsatt modellering.

De hålls medvetet enkla i den meningen att de använder få modelleringsbegrepp och har en enkel notation. Beskrivningssättet är i regel uteslutande grafiskt.

En viktig fråga beträffande dessa modelleringstekniker är huruvida de innehåller eller är kopplade till en vidare konceptuell modellering eller om de stannar vid detta.

En risk som måste poängteras när man stannar på alltför grov och överskådlig nivå är att man kommer till en enighet kring begrepp, en samstämmighet som vid fortsatt detaljering av modellen inte visar sig hålla. En alltför grov beskrivning kan alltså leda till en "falsk" begreppsgemenskap. Man trodde att man var överens, men i själva verket var man det inte.

2.7.3 Objektorienterade modelleringsansatser

Objektorienterade modelleringsansatser använder modelleringsbegrepp som har kunskapsteoretisk, filosofisk och lingvistisk bakgrund och som inte har någon motsvarighet i datorers sätt att arbeta.

Typiska begrepp är objekt/entitet, attribut/egenskap, förhållande/relation/association, händelse.

Beroende på vilka "delmängder" av de "objektorienterade" modelleringsbegreppen som används, brukar man urskilja:

- Entity-Attribute-Relationship (EAR)-ansatser
- Binära ansatser

EAR-ansatserna använder, som namnet utsäger, begreppen objekt/entitet, attribut och förhållande (relationship). Ett attribut är en namngiven association mellan en entitet, eller ett förhållande, och ett attributvärde. En begränsning som gäller är att attribut skall vara "en-värda" (funktionella). Attributvärdens existens beror av existensen av entiteter och förhållanden.

De binära ansatserna använder sig av begreppen objekt och binärt förhållande (association). Ett binärt förhållande råder alltid mellan två

objekt. I NIAM, som är ett exempel på en binär ansats, skiljer man mellan lexikaliska objekt (värden) och icke-lexikaliska objekt. Ett binärt förhållande kan definieras mellan två objekt eller mellan ett objekt och ett värde (lexikaliskt objekt).

2.8 Framtagning och tillämpning av konceptuella modeller

2.8.1 Funktions- och datamodellering

Inom systemutveckling är det brukligt att arbeta med modeller av olika slag. En modell är ju en förenkling av en verklighet (det som modelleras - ofta kallat objektsystemet) gjord ur ett visst perspektiv, ett visst syfte.

Olika modeller beskriver olika aspekter av objektsystemet. Vanligt i de tidiga faserna av en utvecklingsprocess är att utnyttja dels någon form av "funktionsmodellering", där viktiga och relevanta funktioner i verksamheten analyseras, funktioner som kommer att vara berörda av det framtida informationssystemet. De olika funktionernas samband kan också beskrivas, samband av informations- och/eller materiell karaktär.

Vanligt numera är också att objektsystemet modelleras ur begrepps-, informations- och datasynpunkt så att företeelser och regler i en verksamhet modelleras i form av objekt, relationer mellan objekt samt attribut för objekt och relationer.

De olika modellerna, funktionsmodellerna och de konceptuella modellerna, fångar alltså olika aspekter av den bakomliggande verkligheten varit det system som skall byggas skall verka.

I olika ansatser kan de olika inriktningarna betonas mer eller mindre, dvs en ansats prioriterar datamodellering och tonar ner funktionsmodelleringen och vice versa.

I IRM-tanken, som prioriterar datamodelleringen, ligger som grundläggande princip, att systemen bör byggas utifrån en stabil grund, som ej så lätt förändras med tiden. Uppgiften är alltså att hitta de basobjekt i verksamheten som existerar under lång tid och där förändringar snarare berör egenskaper/attribut för dessa objekt. Olika krav på information som olika delar av en verksamhet kan ha är betydligt mer utsatta för förändringar, de kan variera starkt över tiden.

Andra skolor betonar starkare funktionsmodelleringens roll, att avgränsa och analysera funktioner i verksamheten, dess uppgifter och ansvar, beslut som tas i funktionerna på grundval härav, samt resulterande krav på informationsförsörjning.

Man kan alltså bekänna sig till och hävda den ena eller andra skolan och respektive ansats fördelar, eller man kan, som har provats i ett antal företag och organisationer, erkänna att båda modellerna behövs och att de i samspel kan uppvisa en del synergieffekter.

Funktionsmodellering studerar och avgränsar problem, verksamhet och informationsbehov och avgör vad den konceptuella modellen skall omfatta. Den ger också en bakgrund till att tala om delmodeller, lokala konceptuella modeller eller "vyer", lokala modeller på olika nivåer.

Konceptuell modellering upprättar en vokabulär så att de funktionsinriktade modellerna, med hjälp av denna, kan diskutera, beskriva och analysera sina resultat på ett enhetligt, tydligt och begripligt sätt för alla iblandade. De konceptuella modellerna definierar, mer eller mindre omfattande beroende på modelleringsansats, vad informationssystemet skall handla om.

Detta är en sida av systemutveckling - att tillgodogöra sig synergieffekter av "parallell" modellering av olika typer inom faser av en systemutveckling.

2.8.2 En nivåindelad modelleringsteknik

En viss modelleringsansats har ofta ett mindre antal viktiga kärnbegrepp omkring vilka hela modelleringstekniken är byggd. Det är ofta kring de grundläggande kärnbegreppen, som antaganden är byggda omkring 'godheten' av ansatsen för olika användningsområden. Om man skall lära känna en modelleringsansats så är det ofta en god idé att verkligen lära känna dessa viktiga modelleringsbegrepp för att förstå hela modelleringsansatsen och dess potentialer.

Vi kan tala om ett antal aspekter hos en modelleringsansats eller en beskrivningsteknik:

- Antalet modelleringsbegrepp, mängden samband mellan modelleringsbegrepp
- Exaktheten i definitionen av varje begrepp, inklusive utsagor avseende varför det är viktigt att utnyttja dem för beskrivning och på vilket sätt. Detta speciellt för grundbegreppen i ansatsen.
- Att, inom ramen för de aspekter som en modelleringsteknik verkligen försöker sikta in sig på, vilka fasetter finns av de grundbegrepp som modelleringstekniken tillhandahåller. (Exempel är super-subrelationer, och komplexa objekt för grundbegreppet 'objekt'.)

Förutom att en viss ansats har en viss uppsättning grundmodelleringsbegrepp, som kan vara ganska olika emellan ett antal olika ansatser för modellering, så kan antalet övriga begrepp som man sedan bygger på med variera. Dessutom kan djupet i definitionen variera starkt för de olika begreppen.

Dåligt definierade begrepp kan ofta leda till icke önskade tolkningar av begreppen samt missförstånd av hur man skall använda dem. Ett till synes enkelt begrepp kan vid användning i litet mer komplexa modelleringssituationer blir svåränvänt och leda till tveksamheter hur det skall användas. (Exempel: att inte ha tillgång till super-sub-uttrycks-möjligheten för objekt.)

En väl definierad modelleringsansats kan ofta uppfattas som krånglig och svår att komma in i. Likaså kan en ansats som har en mångfald av olika begrepp att använda sig av uppfattas som svår att använda eftersom det finns så mycket att välja bland.

En modelleringsansats där dess olika modelleringsbegrepp är väl definierade och där det finns ett antal begrepp att använda sig av upplevs som mer kraftfull när det råder fler komplexa samband och förhållanden i verkligheten.

Det finns dessutom ett antal olika personer (intressentgrupper) i en verksamhet som har anledning att bli inblandade i arbetet med att ta fram eller att ta del av modeller. Beroende på vad för typ av intressent dessa personer är, när det gäller att arbeta med modellerna, så kan man bli irriterad om det finns en massa detaljer i modellerna som inte är relevanta för den intressentgrupp som man representerar, till exempel för en företagsledning.

Om modellerna för en viss modelleringsansats därför omfattar en mängd detaljer och dessa alltid visas för samtliga intressenter kan modelleringsansatsen få ett rykte om sig att vara svår. För vissa intressenter till modellerna måste det ju dock finnas stor detaljering i modellerna för att dessa skall bli kvalitativa och för att rätt information skall kunna överföras till en viss annan kategori läsare. Ett exempel kan vara en databas-konstruktör som skall skapa en databas eller person som skall skriva ett program för till exempel en viss arbetsrutin (som modellerats).

Om man alltså både har kravet att kunna göra översiktliga modeller för vissa kategorier intressenter och samtidigt för andra kunna uttrycka en hög grad av detaljrikedom och exakthet, förefaller det som att man istället har behov av att kunna visa olika **versioner** av en modell med en detaljrikedom och en 'projicering' på vissa typer av modelleringsbegrepp så att modellen är relevant till vad läsarna behöver ha reda på.

Man visar alltså **rätt version** av modellen för **rätt intressentgrupp**. Man kan tala om följande dimensioner av en modell:

- **DETALJERINGSNIVÅ** för varje modelleringsbegrepp, t ex för begreppet entitet hur mycket av attribut för dessa samt restriktioner tas med i modellen.
- **PROJICERING**, vilka begrepp har tagits med för modellen och vilka har undertrycks.
- **AGGREGERINGSNIVÅ**, för vilka begrepp har man slagit samman flera instanser av ett modelleringsbegrepp under särskilda modelleringsbegrepp som används för översiktliga modeller. Exempel: komplexa objekt och 'verkliga, exakta' objekt.

På detta sätt kan man arbeta med en modelleringsansats, som dels är kraftfull där stor uttrycksfullhet och detaljrikedom krävs, dels att modellerna också kan vara översiktliga och kunna presentera modeller som har exakt rätt detaljeringsdjup och fasettering i begreppen så att det passar t ex en företagsledning.

För att kunna använda en modelleringsteknik till olika ändamål och att samma begrepp kan användas så långt som möjligt kan man tänka sig att dela in modelleringstekniken i **nivåer** så att man bara 'ser' de delar som är nödvändiga för att modellera de aspekter som är relevanta för en viss användningssituation.

Det gäller ofta att det är fråga om att en viss detaljering är användbar för en viss användningssituation och att övriga begrepp upplevs som överflödiga och som ibland kan leda till irritation och synpunkter om relevans.

1. Användning: verksamhetsmodellering

Begrepp: Objekt, attribut, relationer, verksamhetsfunktioner, grova informationssamband, mål, verksamhetsuppgifter.

2. Användning: Begreppsmodellering i samband med projektstart.

Begrepp: Objekt, men ibland är klassificering som attribut, objekt, relation inte nödvändiga att göra här. Funktion, informationsbehov (ibland grova flöden).

3. Användning: Informationsanalys

Begrepp: Objekt, relationer, attribut, domäner, restriktioner på värdemängder, beroenden, härledning, funktioner, flöden, grova processer, begrepp som används i olika processer.

4. Användning: Förvaltning, dataadministration

Begrepp: Objekt, relationer, attribut, informationssystem, verksamhetsfunktioner, databaser, domäner, härledning, databaser, ansvarsområden, (objekt, relationer, attribut, domäner i olika informationssystem och databaser), etc.

5. Användning: Systemutformning och fysisk realisering

Begrepp: Objekt, relationer, sökvägar, kardinalitet, uttags- och uppdateringsfrekvenser. Formell beskrivning av behandlingsregler och kontroller.

En slutsats av ovanstående resonemang är att de modelleringstekniker som används bör ha en sådan uppsättning av begrepp, och att dessa begrepp kan ha en sådan detaljrikedom, att de kan tas fram och användas för respektive användningsområde. Man har alltså möjlighet att uttrycka detaljer när man behöver detta.

2.9 Övergång från konceptuell modell till databasmodell

Ett krav på ansatser för konceptuell modellering är att det inte skall finnas "något glapp mellan analys och konstruktion".

En modelleringsansats bör alltså inrymma eller i varje fall ha koppling till en metodik för hur man vidareför modellen inom konstruktionssteget.

Konstruktionsprocessen innebär att från en problemorienterad kravspecifikation utforma ett schema för en databas med tillhörande processer. Enligt (ref Bub & Lind) kan denna process principiellt indelas i två steg: i det första steget görs en övergång till ett preliminärt databasschema strukturerat på det sätt som aktuell databashanterare föreskriver.

Är databashanteraren av relationstyp, skapas ett preliminärt relationsdatabasschema som definierar vilka relationstabeller som är aktuella och deras innehåll, etc, är databashanteraren av nätverkstyp beskrivs schemat med hjälp av posttyper, settyper, etc. Syftet är att "utveckla ett eller ett antal alternativa databasscheman som 'logiskt' ligger så nära det konceptuella schemat som möjligt".

Det andra steget syftar till att ta fram ett optimerat databasschema, där omstruktureringar görs som syftar till att ta hänsyn till effektivitetsmässiga aspekter. Denormalisering, återinföra redundans, definiera index för snabbare åtkomst, etc.

Den detaljerade konceptuella modellen skall utgöra bästa möjliga underlag för detta steg där design- och konstruktionsbeslut av annan art än under specifikationsarbetet tas.

Viktiga frågor att ställa sig när man betraktar en viss modelleringsansats är alltså:

- är den konceptuella modellen tillräckligt detaljerad för att utgöra ett gott konstruktionsunderlag? Dvs finns det i denna ALLA uppgifter av "problemorienterad natur" som konstruktören behöver eller måste han/hon tillfoga egna antaganden om sådana.
- finns det en god och glappfri koppling mellan metodik för de olika stegen? Finns det detaljerade regler för hur uppgifter i det konceptuella schemat skall/kan representeras i aktuellt databasschema.

2.10 Kort presentation av några modelleringsansatser

I detta avsnitt presenteras kort några modelleringsansatser för konceptuell modellering/datamodellering.

2.10.1 CMOL - Conceptual MOdelling Language

Bakgrund:

CMOL är utvecklat inom SYSLAB (Systemutvecklingslaboratoriet), Stockholms universitet.

Modelleringsbegrepp:

Grundbegreppen är entitet, förhållande, händelse, värde (lexikaliskt objekt) samt typer av entiteter, händelser och värden (datatyper). Samband mellan entiteter eller mellan entiteter och värden kallas attribut. Delmängdsförhållanden (generiska samband) mellan entitetstyper och mellan händelsetyper anges. Entiteter, förhållanden och händelser är s k icke-lexikaliska objekt, värden är lexikaliska objekt.

I ett CMOL-schema definieras de olika typerna genom att olika formella egenskaper anges för dem. För en entitetstyp anges t ex namn, beskrivande text, delmängder, ev. mängdattribut dvs attribut som hör till mängden som sådan, händelser som inför respektive avlägsnar element i mängden, attribut, vilka attribut eller kombinationer av attribut som unikt identifierar en entitet i mängden. För varje attribut anges avbildningsrestriktioner ("mapping", totalitet/partialitet) för antalet element som är associerade till entiteten via detta attribut och dito för dess invers, av vilka händelser eller enligt vilken regel attributet tilldelas, om attributet kan ändras och i så fall av vilka händelser, etc.

Liknande definitionsmallar finns föreskrivna för förhållande-, händelse- och datatyper.

Beskrivningsteknik/notation:

I den grafiska notationen anges enbart entitetstyper, förhållandetyper, datatyper samt attributs- och delmängdssamband.

I den textuella notationen anges förutom de samband som också finns grafiskt beskrivna ytterligare ett antal egenskaper för entitetstyper, händelsetyper, datatyper och attribut, enligt de definitionsmallar som exemplifierats ovan.

Systemutveckling, metodik och datorstöd:

Ett CMOL-schema kan utvecklas genom gradvis detaljering. I tidiga verksamhetsanalysfaser kan ett konceptuellt schema på grov nivå och som upptar de viktigaste objekttyperna och deras samband utvecklas. Fördjupad funktionsanalys introducerar successivt alltmer detaljerade informationskrav och det konceptuella schemat kan kompletteras med nya objekttyper, samband och regler. Det är på detaljerad informationsanalysnivå som hela "batteriet" av modelleringsmöjligheter utnyttjas fullt ut.

Lokala konceptuella scheman för informationsbehov kan upprättas och sedan, vid behov, integreras. Alternativt kan lokala scheman upprättas för varje avgränsad funktion/process och avse funktionens hela "informationssfär", dvs dess in- och utgående informationskrav.

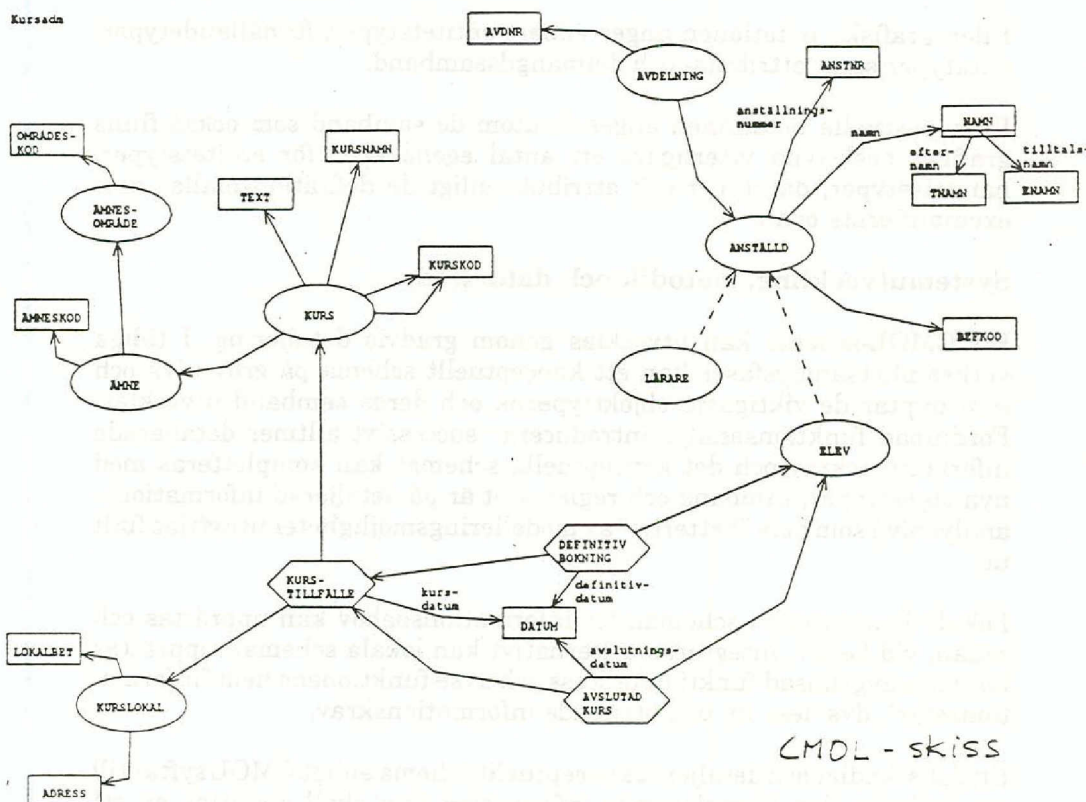
Ett fullständigt och detaljerat konceptuellt schema enligt CMOL syftar till att beskriva den formaliserade information som skall omfattas av ett informationssystem samt hur den produceras, dvs härleds från "externa" händelser/transaktioner. Det konceptuella schemat är ett "dictionary" för den information och de regler som systemet omfattar.

Implementering av ett konceptuellt schema kan tänkas ske i två steg - en preliminär transformation som överför det konceptuella schemat till ett eller flera alternativa databasscheman enligt aktuell databasmodell/typ av DBHS (relationsdatabas, hierarkisk databas, CODASYL/nätverksdatabas). Vissa riktlinjer för en sådan transformation till relations- och CODASYL-databas finns. De flesta regler och villkor i ett CMOL-schema måste implementeras med hjälp av procedurer. I nästa steg sker en omstrukturerad i effektiviserande syfte utifrån operationella krav och data (tidskrav, belastningar, volymer, frekvenser, etc).

CMOL finns preliminärt definierad i datorstödet RAMATIC. I nuläget är användningen dock begränsad till hantering av beskrivningar i den grafiska notationen.

Referenser:

Bubenko, J.& Lindencrona, E.: "Konceptuell modellering - Informationsanalys, Studentlitteratur, 1984.



2.10.2 EMOL - Ericsson MOdelling Language

Bakgrund:

Utvecklad i samarbete mellan Ericsson Telecom och SISU.

Bygger på CMOL (Conceptual Modelling Language, se ovan) men är förenklat och mindre formellt.

Modelleringsbegrepp:

Grundbegreppen är objekt och värde samt typer av objekt och värden, de senare kallade domäner. Samband mellan objekt eller mellan objekt och domän kallas attribut. Delmängdsförhållanden (generiska samband) mellan objekttyper beskrivs. Objekt är s k icke-lexikaliska objekt, värden är lexikaliska objekt.

I ett EMOL-schema definieras de olika typerna genom att olika formella egenskaper anges för dem. För en objekttyp anges t ex namn, beskrivande text, delmängder, funktioner/dialogsteg som inför respektive avlägsnar objektforekomster, attribut, vilka attribut eller kombinationer av attribut som unikt identifierar en objektforekomst. För varje attribut anges bl a undre och övre gräns för antalet element som är associerade till entiteten via detta attribut och dito för dess invers, av vilka dialogsteg eller enligt vilken regel attributet tilldelas. Liknande definitionsmallar finns föreskrivna för domäner.

Beskrivningsteknik/notation:

I den grafiska notationen anges enbart objekttyper och domäner samt attributs- och delmängdssamband. I de verbala beskrivningarna anges förutom de samband som också finns grafiskt beskrivna ytterligare ett antal uppgifter för objekttyper, domäner och attribut (kardinaliteter, värdeförråd, härledningsregler, etc).

Systemutveckling, metodik och datorstöd:

Datamodellering med hjälp av EMOL ingår som en del i aktiviteten Applikationsmodellering. Applikationsmodellering inrymmer dessutom Funktions- och dialogmodellering, och de tre komponenterna är avsedda att användas tillsammans inom aktiviteten. Under applikationsmodelleringen modelleras funktioner (informationssystemfunktioner) och data tillsammans, först översiktligt, och senare under dialogmodelleringen, detaljerat.

Datamodellering med EMOL kan även användas fristående t ex för att beskriva informationsbehov (verksamhetsfunktioners informationssfärer) i samband med verksamhetsbeskrivningar.

Funktionsmodellen består av en informationsstruktur samt verbala beskrivningar av de funktioner som skall stödja verksamheten. Datamodellen består av en graf och av ett antal verbala beskrivningar, som tillsammans beskriver den informationsstruktur som behövs för att uppfylla funktionernas informationsbehov. Dialogmodellen består av ett antal grafer som, för varje funktion på lägsta nivå, beskriver en dialogstruktur (dialogsteg och hur de följer efter varandra) samt verbala beskrivningar av varje dialogsteg.

Att analysera och specificera informationssystemet är en iterativ process, där datamodellen växer fram under nedbrytning och specificering av systemets funktioner. Det finns en direkt koppling mellan datamodellen och funktions-specifikationerna respektive dialogstegen. Datamodellen måste "motiveras" så att varje "element" som förs in i datamodellen svarar mot ett informationsbehov i någon funktion eller något dialogsteg.

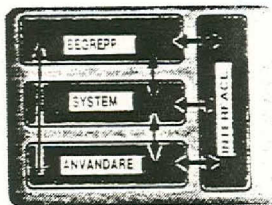
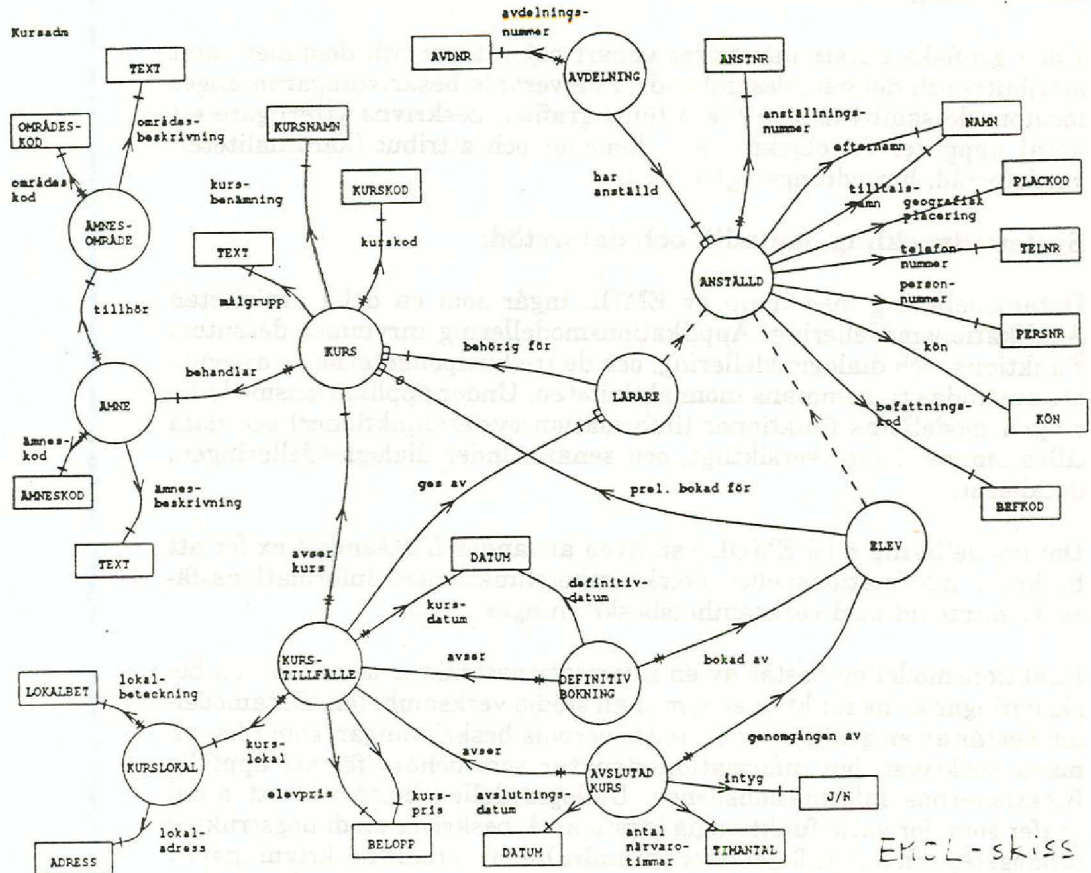
Specificeringsarbetet förutsätter en förstudie som anger hur verksamhetens informationsbehov förändras, ett viktigt underlag för att få till stånd en stabil datastruktur i informationssystemet.

Implementering av en datamodell sker i två steg - en preliminär transformation som överför datamodellen till ett eller flera alternativa databas-scheman (företrädevis en relationsdatabas) Riktlinjer för en sådan transformation till relationsdatabas finns. I nästa steg sker en omstrukturering i effektiviserande syfte utifrån operationella krav och data.

Applikationsmodellering (funktionsmodellering, datamodellering, dialogmodellering) finns definierad i datorstödet RAMATIC.

Referenser:

Metodhandboken för utveckling av informationssystem, ERICSSON TELECOM Avdelningen för konstruktionsstöd system- och programvara



2.10.3 Lindqvistkonceptet

Av Birgit Norén och Bror Norén

Bakgrund:

I början av 1980-talet var man inom ramen för den dataadministrativa verksamheten SIV (Standardiseringsnämnden för Informationsbehandling inom Vattenfall), inom Vattenfall intresserad att få fram en strukturerings teknik som kunde användas för att analysera komplexa informationssamband. Ett sådant hjälpmedel, betecknat "Lindqvistkonceptet", utvecklades av Birgit Norén (numera FMV) som ett fritidsarbete i form av en trebetygssuppsats vid Stockholms Universitet, 1984 (se referenslista).

Inspirationskälla för detta arbete var därvid John Lindqvist som tidigare var chef för driftteknisk forskning och utveckling inom Vattenfall. Denne, som också fått ge namn till konceptet, avgav redan i början av 1970-talet principerna för katalogstyrd datalagringsteknik i samband med utvecklingen av Vattenfalls datoriserade driftövervakningssystem KK-info.

Lindqvistkonceptet innefattar i dag:

- dels konceptuell strukturerings teknik som hjälpmedel för konstruktion av ADB-baserade datakatalogstyrdatalager
- dels konceptuell teknik för successiv nedbrytning och kartläggning av informationssamband i "nivåer".

Konceptet innefattar principer för successiv nedbrytning och kartläggning av objektverkligheter i olika nivåer. Såväl konceptuell modellerings-teknik som katalogiseringsteknik i konceptet kan modulariseras i nivåer efter eget önskemål. Detta gör att även mycket stora och komplexa informationssamband kan hanteras utan att överblickbarheten förloras.

Den konceptuella modell som utvecklas under en analysfas utgör tillsammans med den likaledes framväxande datakatalogen, en informationskarta där varje enskilt elementarmeddelande kan urskiljas. Varje term (attribut) registreras och hanteras där med hjälp av en "longitud-och-latitudkod" som id-begrepp. På detta sätt får man ett hjälpmedel (informationskarta och datakatalog) där man kan urskilja och hantera såväl "fjäll som mindre stenar".

Modelleringsbegrepp:

De modelleringsbegrepp som används i Lindqvistkonceptet kan sammanfattas såsom

- **objekt** - objektindivid eller objekt förekomst "som man vill informera om"
- **objektklass** - gruppering av objekt (dvs objektindivider) med samhörande egenskaper. Objektgrupper kan uttryckas enligt mängslärens principer dvs som delmängder, snittmängder etc.

Genom samspelet mellan modell och katalog tillsammans med att varje enskild term kan identifieras "på kartan" genom sin kod kan sådana grupperingar vara av godtycklig komplexitet.

- **term** - egenskap till objekt (synonymt begrepp; attribut)
- **i s a-förhållande** - delmängd(er) av en objektpopulation i en objektklass som har den omslutande klassens alla attribut plus ytterligare ett antal delmängdsspecifika sådana attribut
- **association** - "förhållande" mellan objektgrupperingar. Associationer (pilar i modellen) kan på grupperingar över objektklassnivå innebära grupperingar av associationer. En pil på högre objektgrupperingsnivå kan alltså innebära ett antal olika associationer i praktiken. Den exakta innebörden av en sådan associationspil löses upp när man "kommer till" objektklassnivån i ett analysarbete.

Beskrivningsteknik/notationer:

Den grafiska notation som används i Lindqvistkonceptet är hämtad från mängdläran. För objektgrupperingar används sålunda grafiska mängdsymboler (ringar) i olika kombinationer (se bif. modellexempel). För associationerna används pilar.

Systemutveckling, metodik och datorstöd:

Den grafiska datamodellens främsta uppgift i Lindqvistkonceptet är att tjäna som hjälp för konstruktion av konceptuellt strukturerade datakataloger. Väl att märka är dock att modelleringshjälpmedlen i detta koncept utgörs av en grafisk modell och av en datakatalog, dvs man kan "modellera" lika väl i katalogen som med hjälp av den grafiska notationen. Den grafiska modellen och katalogstrukturen utgör ju också en och samma

struktur. Grafisk notation på objektklassnivå används sålunda framförallt för att

- "bena upp" komplexa objektverkligheter
- pedagogiskt åskådliggöra olika detaljerade informationssamband
- grafiskt visualisera informationsstrukturer.

Färdiga kataloger kan sedan tjäna dels som uppslagsverk över information och informationssamband och dels som helt integrerade logiska databaser, dvs databasscheman i ADB-baserade datalager. För databasrealisering krävs dock helst objektorienterad databasteknik i botten.

Lindqvistkonceptet har i detta sammanhang endast beskrivits i rollen modellering. I konceptet inryms dock också möjligheter av typen

- integration mellan objektstruktur och andra strukturer
- möjligheter att skapa "kataloger över kataloger", dvs totalt överblickbara och dataadministrativt underhållbara grupperingar av ADB-system (systemområden)
- knytning av strukturella och datakvalitativa metadata (data om data).

Referenser:

Norén Birgit: "Varför är databasen i Vattenfalls TIDAS-system så stabil och flexibel? Ett försök till formalisering av databasens konceptuella struktur", Stockholms Universitet 1984-06-05

Norén Bror: "Dataadministration - en principstudie av kriterier och bakgrundskrav för dataadministrativ verksamhet" 1987-11-24, Vattenfall

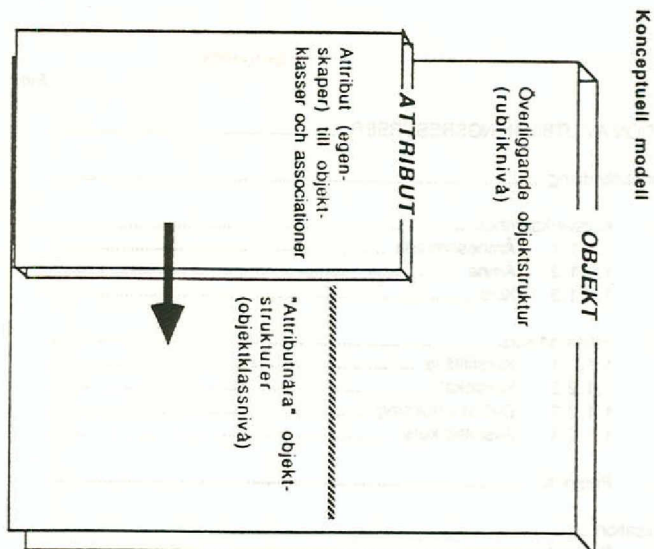
Norén Bror: "Lindqvistkonceptet - en integrerad strategi för informationshantering" 1987-09-19, Vattenfall

Norén Bror: "Resurssyn på data - erfarenheter från Vattenfall" föredragsunderlag Stadskontoret 1988-01-06, Vattenfall

Kommentarer:

Denna teknik skapar en datakatalog som har samma konceptuella struktur som motsvarande grafiska modell. Genom att sedan låta katalogen utgöra logisk databas i en katalogstyrd "objektsorienterad" databasansats uppnår man ett "ett till ett-förhållande" mellan modell och databasstruktur. Genom att modellen i princip utgör en "informationskarta" kan man också tala om ett sådant "ett till ett-förhållande mellan i detta fall verksamhet, modell och databas.

Detta samt att godtycklig mängd metadata kan specificeras för respektive dataelementtyp i katalogen (logiska databasen) ger stora möjligheter till dataadministrativ överblick över såväl enskilda ADB-system som samverkande ADB-system i systemområden eller i distribuerad miljö.

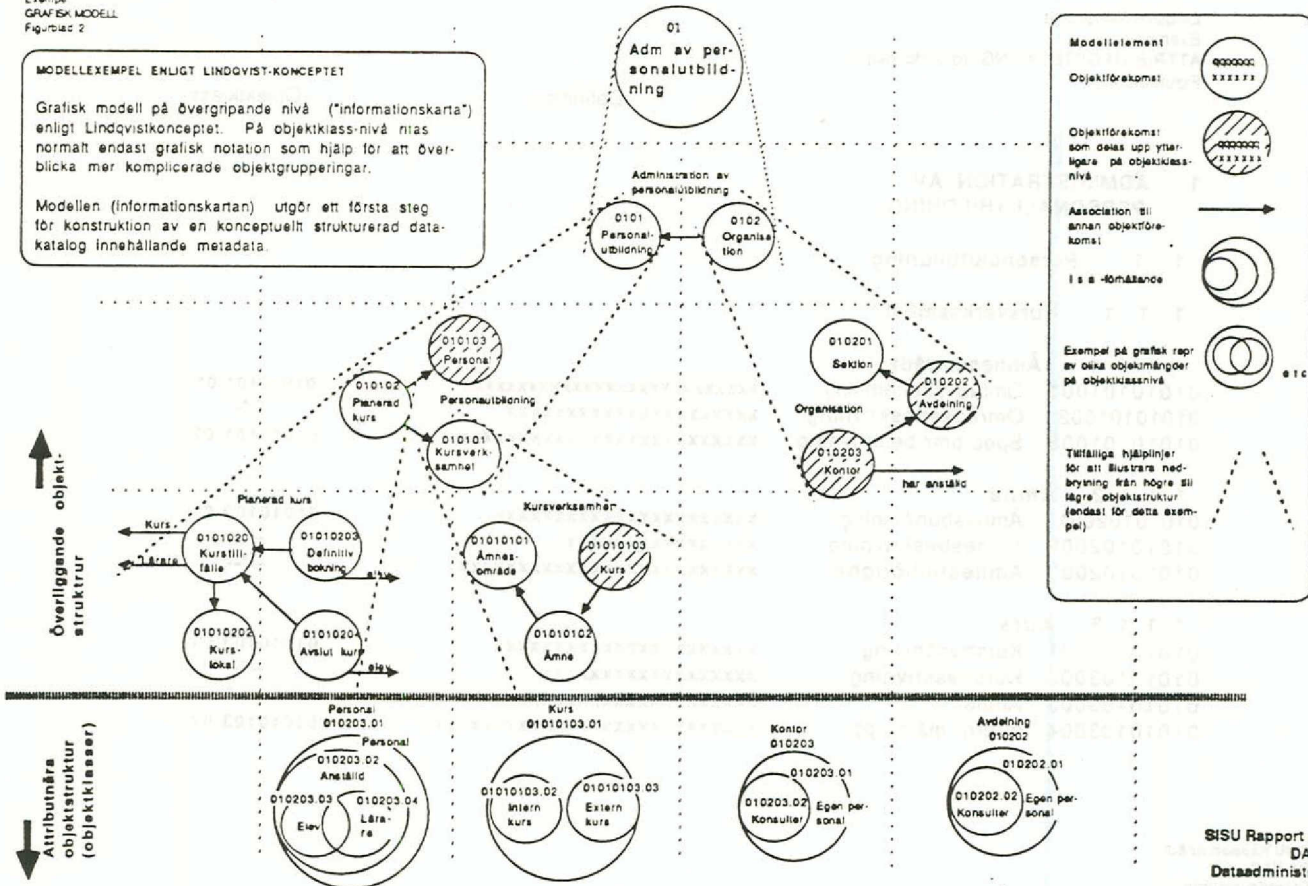


Lindqvistkonceptet
 Exempel
 GRAFISK MODELL
 Figurblad 2

MODELLEXEMPEL ENLIGT LINDQVIST-KONCEPTET

Grafisk modell på övergripande nivå ("Informationskarta") enligt Lindqvistkonceptet. På objektklass-nivå ritas normalt endast grafisk notation som hjälp för att överblicka mer komplicerade objektgrupperingar.

Modellen (informationskartan) utgör ett första steg för konstruktion av en konceptuell strukturerad data-katalog innehållande metadata.



Modellelement

- Objektföräkomst:
- Objektföräkomst som delas upp ytterligare på objektklass-nivå:
- Association till annan objektföräkomst:
- Is-a-förhållande:
- Exempel på grafisk representation av olika objektmängder på objektklass-nivå:
- Tillfälliga hjälplinjer för att illustrera nedbrytning från högre till lägre objektklasser (endast för detta exempel):

Lindqvistkonceptet
 Exempel
 RUBRIKNIVÅ
 Figurbild 3

Sid

- 1. ADMINISTRATION AV UTBILDNINGSRRESURSER
 - 1.1 Personalutbildning
 - 1.1.1 Kursverksamhet
 - 1.1.1.1 Ämnesområde
 - 1.1.1.2 Ämne
 - 1.1.1.3 Kurs
 - 1.1.2 Planerad kurs
 - 1.1.2.1 Kurstillfälle
 - 1.1.2.2 Kurslokal
 - 1.1.2.3 Definitiv bokning
 - 1.1.2.4 Avslutad kurs
 - 1.1.3 Personal
 - 1.2 Organisation
 - 1.2.1 Sektion
 - 1.2.2 Avdelning
 - 1.2.3 Kontor

Lindqvistkonceptet
 Exempel
 ATTRIBUTFÖRTECKNING (grundprincip)
 Figurbild 4

Definition

Objektklass

1 ADMINISTRATION AV
 PERSONALUTBILDNING

1 1 Personalutbildning

1 1 1 Kursverksamhet

1 1 1 1 Ämnesområde

01010101001	Områdesbenämning	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	01010101.01
01010101002	Områdesbeskrivning	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	--"
01010101003	Spec.omr.beskrivning	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	01010101.02

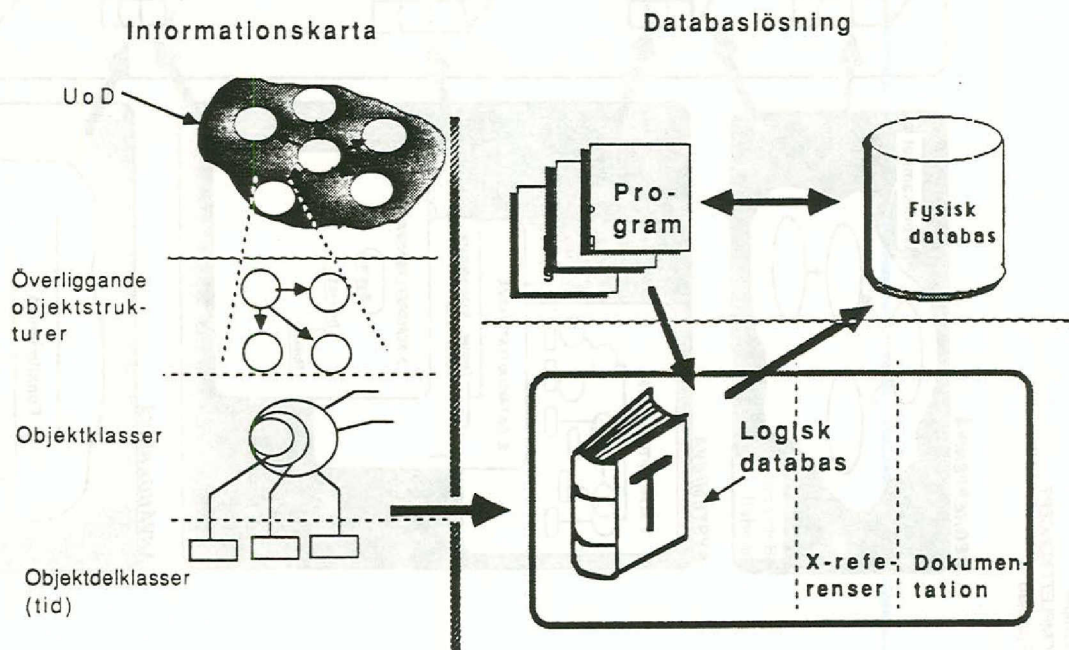
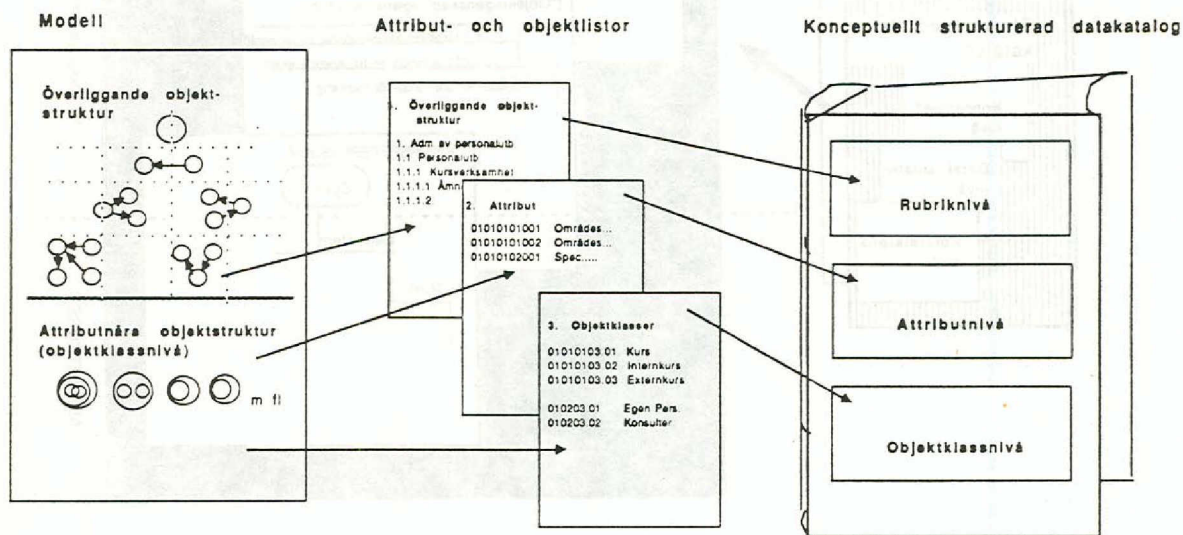
1 1 1 2 Ämne

01010102001	Ämnesbenämning	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	01010102.01
01010102002	Ämnesbeskrivning	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	--"
01010102003	Ämnestillhörighet	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	--"

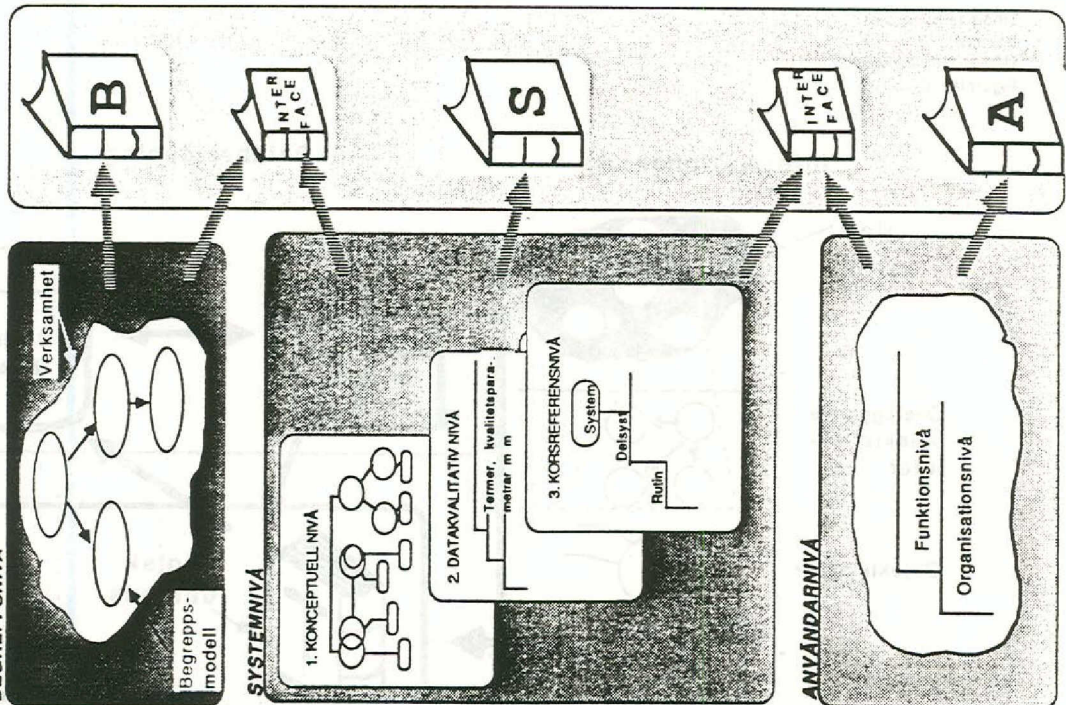
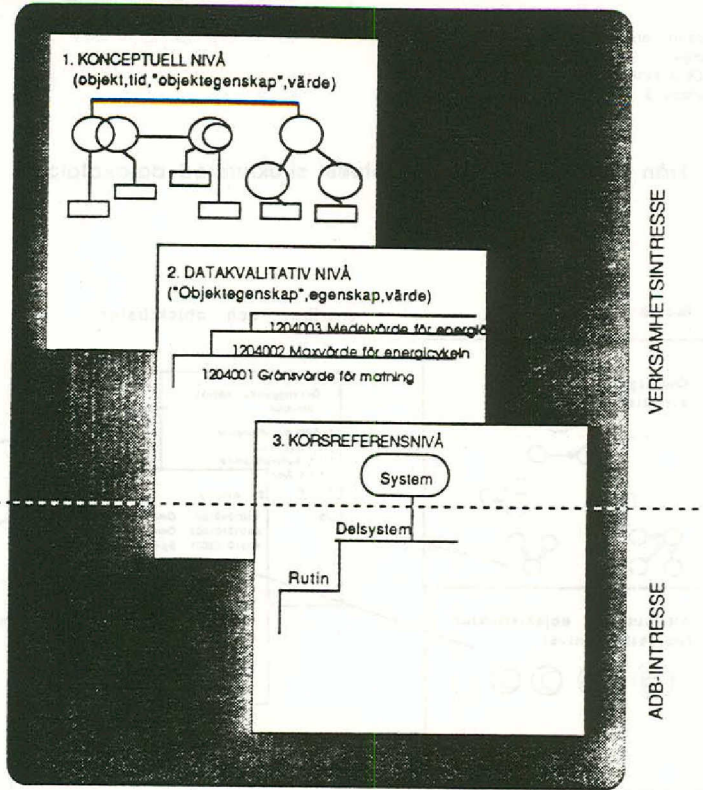
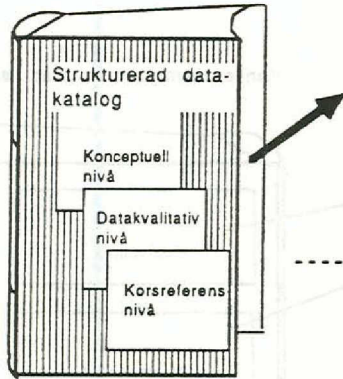
1 1 1 3 Kurs

01010103001	Kursbenämning	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	01010103.01
01010103002	Kursbeskrivning	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	--"
01010103003	Ämne	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	--"
01010103004	Intern målgrupp	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	01010103.02

Från datamodell till konceptuellt strukturerad datakatalog



Konceptuellt strukturerad
 datakatalog / logisk databas



2.10.4 NIAM - Nijssens Information Analysis Method

Bakgrund:

Utvecklad av G.M. Nijssen m fl vid vid Control Data Belgium, Inc. .

Modelleringsbegrepp:

I NIAM:s konceptuella modellering skiljer man mellan icke-lexikaliska och lexikaliska objekt. Lexikaliska objekttyper kallas LOTs (beskriver klasser av lexikaliska objekt som namn, titel, etc) och icke-lexikaliska objekttyper kallas NOLOTs (beskriver klasser av icke-lexikaliska objekt som person, stad, bil, etc.).

En association (binär relation) mellan icke-lexikaliska objekt kallas IDEA, en association mellan ett icke-lexikaliskt och ett lexikaliskt objekt kallas BRIDGE och en association mellan två lexikaliska objekt kallas PHRASE.

För associationer anges roller i båda riktningar (t ex PERSON bor-i STAD och STAD är-bostadsort-för PERSON). Subtyps-indelning används. Vidare anges ett antal olika typer av s k restriktioner (se nedan).

Ett konceptuellt schema kallas i NIAM en "conceptual grammar".

I NIAM:s funktionsmodellering beskrivs funktioner, informationsflöden (strömmar av meddelanden), informationsbaser (lagringsplatser för meddelanden) och omgivande enheter ("environment"). Funktioner bryts ner i delfunktioner. Procedurbeskrivningar för formaliserbara funktioner ingår i den "konceptuella grammatiken".

Beskrivningsteknik/notation:

Det finns en grafisk notation där NOLOTs, LOTs, IDEA TYPEs, BRIDGE TYPEs och subtyp-förhållanden anges.

Det finns också möjligheter att i graferna ange en hel del olika typer av restriktionsregler, t ex identifieringsregler (motsv s k "mapping"-regler dvs 1:1, 1:M, etc), delmängdsregler mellan roller, ömsesidigt uteslutande ("exclusive") subtyper, unikhetsregler (att en kombination av roller unikt identifierar ett icke-lexikaliskt objekt), etc.

Graferna kallas ISDs (informationsstrukturdiagram). Funktionsmodelleringen beskrivs också i grafisk form. Graferna kallas IFDs (informationsflödesdiagram). I tillägg till den grafiska notationen finns en formell notation kallad RIDL (Reference and IDEa Language). I detta kan alla IFDs och ISDs specificeras samt ytterligare ett antal regler. Även procedurer för funktionerna kan specificeras.

Systemutveckling, metodik och datorstöd:

NIAM föreskriver i sig ingen speciell systemutvecklingsmodell. Inriktar sig som namnet säger på informationsanalys.

Gången som föreslås är komprimerat följande: lista alla funktioner som informationssystemet avser stödja, bryt ner funktionerna i delfunktioner till en nivå där informationsflödena och och bearbetningen i funktionerna klart kan beskrivas. En hierarki av IFD:s blir resultatet. Låt varje

informationsflöde ge upphov till ett informationsstrukturdiagram (alltså ett lokalt konceptuellt schema för varje informationsflöde). Beskriv regler och funktioner formellt i RIDL. De lokala schemana överlappar sannolikt - integrera till ett schema för informationssystemet. som bara upptar viktiga NOLOTs och IDEA TYPEs kan användas.

Datorstöd finns utvecklade. Med hjälp av ISDIS (Information Systems Design and Implementation System - CDC-produkt) kan det konceptuella schemat lagras, uppdateras och kontrolleras. ISDIS kan också generera förslag till databasschema för relationsdatabaser och Codasyl-databaser.

En stor användare av NIAM är Dept of National Defence, Canada. Hur utbredd användningen i övrigt är internationellt och inom Sverige är oklart.

Referenser:

Verheijen & Van Bekkum: NIAM: An Information Analysis Method, i Information Systems Design Methodologies: A Comparative Review, North-Holland, 1982.

Concepts and Terminology for the Conceptual Schema and the Information Base, ISO TC97/SC5/WG3, 1982.

2.10.5 SASMO

Bakgrund:

Ursprungligen utvecklad inom SAS. Vidareutvecklad inom IRM Consult. Ingår som del i IRMA- och SVEA-metodiken. Använd i ett stort antal projekt i ett 50-tal skandinaviska verksamheter.

Modelleringsbegrepp:

Centrala modelleringsbegrepp är objekt och relation. Ett enskilt objekt och en enskild relation kallas objekt förekomst respektive relations förekomst.

Relationer kan vara av ordning två eller högre, dvs relatera två eller flera objekt. Relationer av högre ordning än två kan, om lämpligt, objektifieras dvs istället modelleras som ett objekt.

Objekt och relationer beskrivs förutom av sina relationer till andra objekt och relationer också av termer. För objekt väljs en av dessa termer (om termvärdena unikt utpekar förekomster av objektet) som identifikations- eller nyckelterm. övriga termer kallas egenskapstermer. Relationer lånar sina nyckeltermen från de objekt de relaterar.

Avbildningsrestriktioner anges för relationer (1:1, 1:M, M:M). I samband med datamodelleringen exemplifieras objekt- och relationsförekomster i form av tabeller. Sju huvudregler för övergång från datamodell till tabeller (s k tabellprecisering) finns definierade.

M:M-relationer namnges med substantiv i singular form. 1:M-relationer kan, men behöver inte namnges. Relationerna är oriktade - inverser används inte.

Modelleringsbegreppen är medvetet få och enkla i avsikt att inte komplicera begreppsapparaten. I varianter av SASMO-ansatsen kan begrepp som optionalitet, uteslutande relationer, generiska samband (super-/subobjekt), etc, användas.

Beskrivningsteknik/notation:

I den grafiska notationen symboliseras objekt av rektanglar och relationer med linjer som förbinder de objekt som relateras.

Avbildningsrestriktioner för relationer, dvs maxangivelse symboliseras med "gaffel" på relationslinjen i respektive relationsända.

Om optionalitet anges noteras detta med hjälp av nollor och ettor på relationerna. Ibland kan min-, medel-, maxangivelser noteras i anslutning till relationerna.

Uteslutande relationer kan indikeras genom att en båge dras över dem.

Generiska samband kan beskrivas genom i varandra inskrivna rektanglar, där det yttre objektet är superobjekt och de inre är subobjekt (specialiseringar av det mer generella superobjektet).

Systemutveckling, metodik och datorstöd:

Som tidigare nämnts ingår SASMO-ansatsen som en del i IRMA-metodiken (IRMA = Information Resource Management Architecture). IRMA-metodiken syftar till att framställa en "stadsplan" som underlag för planering och utbyggnad av en infrastruktur för en verksamhets informationsbehandling.

IRMA-metodiken innehåller ett antal arbetssteg, vari en datamodell tas fram, verksamheten indelas i funktioner och funktionerna relateras till datamodellens komponenter i form av en s k IRM-matris, vari framgår användning av och ansvar för data i funktionerna.

IRM-matrisen utgör sedan den karta varifrån, i de efterföljande arbetsstegen, en ideal utvecklingsplan tas fram, som analyseras i anslutning till de kritiska framgångsfaktorer, som utarbetas tillsammans med ledningen.

Arbetet utmynnar i en stadsplaner rapport och förslag till och beslut om fortsatt arbete innehållande aktiviteter som berör etablering av IRM-/dataadministrationsfunktion, teknologifrågor, säkerhetsaspekter, etc.

SVEA-metodiken används vid utveckling av nya system enligt den stadsplan som tagits fram. I denna används också datamodellering enligt SASMO men nu med inriktning på den aktuella tillämpningen.

Referenser:

Datamodellering - begrepp och definitioner. IRM Consult.

Databasanpassning. IRM Consult.

Axelsson, L. & Ortman, L.: Utvecklingshandboken SVEA. Modell och metoder för administrativt utvecklingsarbete, Kommundata, 1985.

IRM Consult säljer stadsplaneringsidén, SISU informa, Nr 88/3-4.

Databas: dabas
 Bild : Kursadm

Tabellnummer:
 Namn : ANSTALLE
 Beskrivning :

Typ	Namn	Format
Id	Anställningsnummer	C(5)
Eg	Arbetsplatsnummer	C(5)
Eg	Efternamn	C(24)
Eg	Förnamn	C(15)
Eg	Geografisk placering	C(18)
Eg	Telefonnummer	C(10)
Eg	Personnummer	C(10)
Eg	Kön	C(1)
Eg	Befattningskod	K(4)

Tabellnummer:
 Namn : AVDELNING
 Beskrivning :

Typ	Namn	Format
Id	Anställningsnummer	C(5)

Tabellnummer:
 Namn : AVSLUTAD KURS
 Beskrivning :

Typ	Namn	Format
Id	Anställningsnummer	C(5)
Id	Områdeskod	C(3)
Id	Ämneskod	C(3)
Id	Kurskod	C(2)
Id	Lokalbet	C(10)
Id	Kursdatum	D(8)
Eg	Analysningsdatum	M(6)
Eg	Marvaxrollnummer	M(4)
Eg	Intygs	C(2)

Tabellnummer:
 Namn : DEFINITIV BOKNING
 Beskrivning :

Typ	Namn	Format
Id	Anställningsnummer	C(5)
Id	Områdeskod	C(3)
Id	Ämneskod	C(3)
Id	Kurskod	C(2)
Id	Lokalbet	C(10)
Id	Kursdatum	D(8)
Eg	Definitivdatum	D(8)

Databas: dabas
 Bild : Kursadm

Tabellnummer:
 Namn : ELEV
 Beskrivning :

Typ	Namn	Format
Id	Anställningsnummer	C(5)

Tabellnummer:
 Namn : KURS
 Beskrivning :

Typ	Namn	Format
Id	Områdeskod	C(3)
Id	Ämneskod	C(3)
Id	Kurskod	C(2)
Eg	Kursbenämning	C(15)
Eg	Kursgrupp	C(15)

Tabellnummer:
 Namn : KURSDAG
 Beskrivning :

Typ	Namn	Format
Id	Kursdatum	D(8)

Tabellnummer:
 Namn : KURSLOKAL
 Beskrivning :

Typ	Namn	Format
Id	Lokalbet	C(10)
Eg	Lokaladress	C(55)

Tabellnummer:
 Namn : KURSLÖSARE
 Beskrivning :

Typ	Namn	Format
Id	Anställningsnummer	C(5)
Id	Områdeskod	C(3)
Id	Ämneskod	C(3)
Id	Kurskod	C(2)
Id	Lokalbet	C(10)
Id	Kursdatum	D(8)

Databas: dabas
 Bild : Kursadm

Tabellnummer:
 Namn : KURSTILLFALLE
 Beskrivning :

Typ	Namn	Format
Id	Områdeskod	C(3)
Id	Ämneskod	C(3)
Id	Kurskod	C(2)
Id	Lokalbet	C(10)
Id	Kursdatum	D(8)
Eg	Kurspris	N(6)
Eg	Elektris	M(6)

Tabellnummer:
 Namn : LARAARBEPICHT
 Beskrivning :

Typ	Namn	Format
Id	Anställningsnummer	C(5)
Id	Områdeskod	C(3)
Id	Ämneskod	C(3)
Id	Kurskod	C(2)

Tabellnummer:
 Namn : LARAAE
 Beskrivning :

Typ	Namn	Format
Id	Anställningsnummer	C(5)

Tabellnummer:
 Namn : PRELIMINAR BOKNING
 Beskrivning :

Typ	Namn	Format
Id	Anställningsnummer	C(5)
Id	Områdeskod	C(3)
Id	Ämneskod	C(3)
Id	Kurskod	C(2)

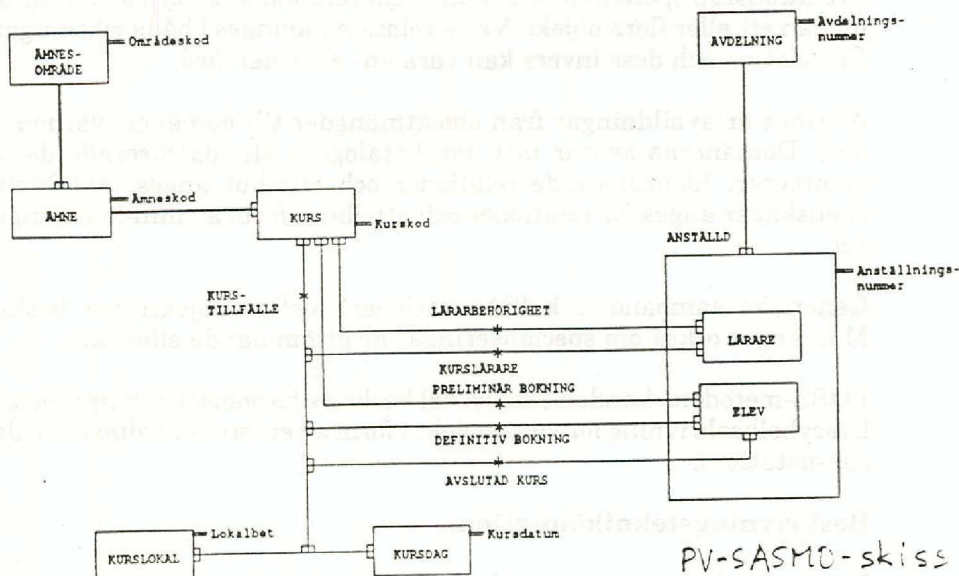
Tabellnummer:
 Namn : AMBT
 Beskrivning :

Typ	Namn	Format
Id	Områdeskod	C(3)
Id	Ämneskod	C(3)
Eg	Ämnesbeskrivning	C(25)

Databas: dabas
 Bild : Kursadm

Tabellnummer:
 Namn : AMBESOMRADE
 Beskrivning :

Typ	Namn	Format
Id	Områdeskod	C(3)
Eg	Områdesbeskrivning	C(55)



2.10.5 TELMOD

Bakgrund:

Utvecklad i samarbete mellan Televerket, TeleLOGIC och SISU som ett förslag till modelleringsansats för det sk IA-projektet (IA= Informations-Administration).

Ansatsen är till alla delar ännu inte helt färdigutvecklad. Det kan vara intressant att återge de krav som ställdes upp inför utvecklandet av modelleringsansatsen:

- det skall vara möjligt att ange domän för
- 'lexikala' attribut relationer mellan objekt skall uttryckas som verb, där så är möjligt (undantag 'lexikala' attribut)
- relationers riktning skall framgå på ett väldefinierat och tydligt sätt
- en relations båda riktningar skall namnges
- avbildningsrestriktioner skall (kunna) anges
- om M:M-relationer 'objektifieras' bör man vända gafflarna åt rätt håll
- generiska relationer skall få användas, dvs det skall finnas en notation för detta
- någon form av notation för beskrivning av ett objekts livscykel bör övervägas

Modelleringsbegrepp:

De modelleringsbegrepp som används i TELMOD är objekttyp, relation ("relationship"), attribut och domän. En relation är en typ av förhållande mellan ett eller flera objekt. Varje relation namnges i båda riktningarna. En relation och dess invers kan vara en- eller flervärd.

Attribut är avbildningar från objekt mängder till domäner (värdemängder). Domänerna svarar mot datakatalogens standardiserade dataelementtyper. Identifierande relationer och attribut anges. Avbildningsegenskaper anges för relationer och attribut i form av min-, maxangivelser.

Generiska samband (s k ISA-relationer) mellan objekttyper beskrivs. Man anger också om specialiseringar är uttömmande eller inte.

I OBS-metodens händelseanalysdel beskrivs händelser och upprättas en livscykelbeskrivning för varje objekt i form av ett strukturdiagram (Jackson-notation).

Beskrivningsteknik/notation:

Den använda notationen är i huvudsak grafisk. I grafen beskrivs objekttyper och relationer mellan objekttyper. Relationslinjerna är riktade och namngivna - relationens namn och inversens namn anges, det senare inom parentes. Avbildningsrestriktioner noteras med "gafflar" (max) och nollar och ettor (min - ibland kallat optionalitet).

Identifierande relationer markeras med asterisk (*), liksom identifierande attribut, som kan skrivas in i objekttypssymbolerna. Övriga attribut anges inte i grafen utan skrivs in i en attributlista, där domän och attributets avbildningsrestriktioner anges.

Systemutveckling, metodik och datorstöd:

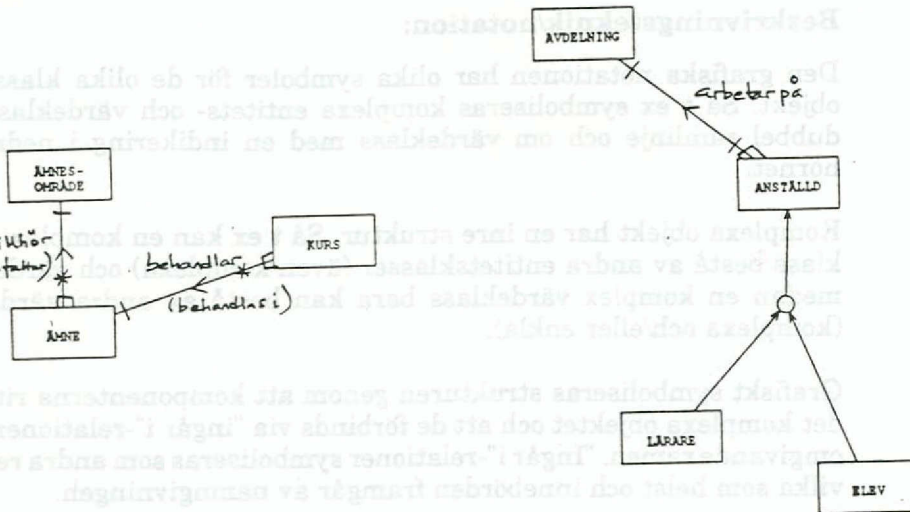
TELMOD är avsedd att användas för modellering inom den systemutvecklingsmetod som utvecklats inom Televerket och som kallas OBS-metoden.

TELMOD kommer inom en snar framtid att datorstödjas i RAMATIC.

Referenser:

Wangler, Benkt: "Förslag till modelleringsansats för IA-projektet, SISU, 12 sep 1988

Berghem, A & Jussila, K.: "TelMod, modelleringsansats för Televerket, S tlo, TeleLOGIC Sv, 1989-07-17



TELMOD-skiss

2.10.6 TEMPORA-ERT

Bakgrund:

Utvecklad inom TEMPORA-projektet, ett samarbetsprojekt som ingår i det s k ESPRIT-programmet. I projektet deltar BIM (Belgien), Hitec (Grekland), Imperial College (England), LPA (England), SINTEF (Norge), SISU (Sverige), University of Liege (Belgien) och University of Manchester Institut of Science and Technology (England).

I TEMPORA-projektets ansats för informationssystemutveckling ingår många komponenter, men beskrivningen koncentreras här till ERT (Entity-Relationship-Time) ansatsen för konceptuell modellering. På specifikationsnivån finns i övrigt två andra modeller, en regelmodell och en processmodell.

ERT-modellen är en utvidgad E-R modell, utvidgad i den meningen att den innefattar en tidsdimension. Dessutom skiljer den sig från den ursprungliga E-R-modellen i det att alla associationer mellan objekt ses som relationer ("relationships") och att därmed ingen distinktion görs mellan relationer och attribut ("attributeships").

Projektet befinner sig i ett relativt tidigt skede, men ERT-modellen är den modell som har kommit längst och som nu börjar stabilisera sig.

Modelleringsbegrepp:

I ERT-ansatsen används modelleringsbegreppen entitet, relation (förhållande), värde, tidsperiod och komplext objekt samt klasser av dessa, dvs entitetsklasser, relations-(förhållande-)klasser, värdeklasser, tidsperiodklasser och komplexa objekt-klasser. Värdeklasser kan också vara komplexa, dvs innehålla andra värdeklasser som komponenter.

Relationer/förhållanden kan gälla mellan två entiteter eller mellan en entitet och ett värde.

Beskrivningsteknik/notation:

Den grafiska notationen har olika symboler för de olika klasserna av objekt. Så t ex symboliseras komplexa entitets- och värdeklasser med dubbel ramlinje och om värdeklass med en indikering i nedre högra hörnet.

Komplexa objekt har en inre struktur. Så t ex kan en komplex entitetsklass bestå av andra entitetsklasser (även komplexa) och värdeklasser, medan en komplex värdeklass bara kan bestå av andra värdeklasser (komplexa och/eller enkla).

Grafiskt symboliseras strukturen genom att komponenterna ritas inuti det komplexa objektet och att de förbinds via "ingår i"-relationer till den omgivande ramen. "Ingår i"-relationer symboliseras som andra relationer vilka som helst och innebörden framgår av namngivningen.

Entitetsklasser och förhållanden som är härledda (dvs vars existens kan härledas ur andra förhållanden i modellen) markeras som streckade. Entitets- och förhållandeklasser kan tidsstämplas och aktuell tidsresolution anges i separat "ruta" i anslutning till huvudsymbolen.

Generiska samband (ISA-relationer) symboliseras på samma sätt som i TELMOD (se ovan).

Alla "vanliga" förhållanden är binära och namnges i båda riktningar. Dock finns ingen riktning angiven för förhållanden. Förhållanden kan också vara "unära" och används för att modellera förhållanden mellan en entitet och ett värde, där värdet är "sant eller falskt".

Avbildningsrestriktioner anges för alla förhållanden i båda riktningar i form av min-, maxangivelser.

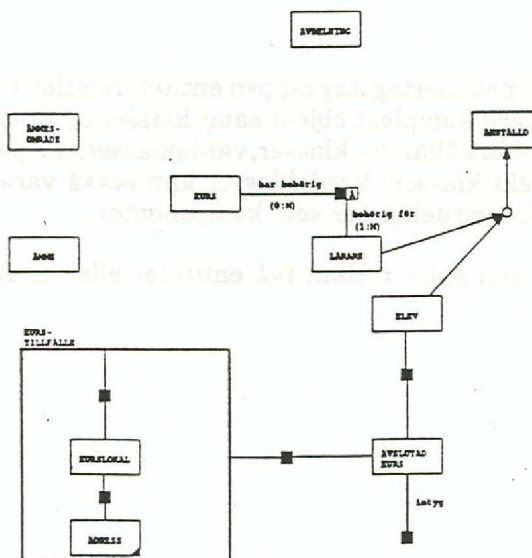
Systemutveckling, metodik och datorstöd:

Som ovan nämnts ingår också modeller för specifikation av regler och processer. Dessa ansatser är dock i nuläget inte helt utarbetade. Detta gäller också metodikdelen, dvs tillvägagångssätt, riktlinjer och procedurer för utvecklingen av en TEMPORA-modell.

En första enkel implementering av datorstöd för ERT-modellen finns inlagd i RAMATIC.

Referenser:

The TEMPORA Concepts Manual, TEMPORA-projektet, Juni 1989.



3. Dataadministration och datorstöd

3.1 Bakgrund till intresset för datorstöd

I detta avsnitt analyseras och diskuteras behovet av datorstöd för att bedriva dataadministration och vilka egenskaper dessa datorstöd skall ha.

Dataadministration är en verksamhet som praktiskt går ut på att hålla reda på beskrivningar av verksamheten, den information som hanteras i verksamheten, vilka informationssystem som finns samt vilken information dessa system hanterar. I syftet för dataadministration ligger att utnyttja dessa beskrivningar på ett ändamålsenligt sätt.

Man kan fråga sig om denna verksamhet kräver användning av datorstöd. En reell analys av behov bör göras innan man lås sig för uppfattningen att datorstöd krävs för denna typ av verksamhet. I det följande genomförs en diskussion om vilka krav som i allmänhet brukar finnas.

En allmän hypotes: Om det finns stora beskrivningar i en verksamhet och det finns ett större antal människor i verksamheten som behöver ta del av dessa beskrivningar vid olika tillfällen, har det visat sig att det har varit just avsaknaden av lämpliga datorstöd som har orsakat att dataadministration ej slagit igenom.

Svårigheter med att

- hantera olika modeller,
- att hålla dem aktuella,
- att hitta i dem,
- att få överblick i modellerna,
- att snabbt söka och kontrollera i modellerna samt
- att hålla modellerna snabbt tillgängliga,

har ofta varit anledningar till att dataadministration ej haft erforderliga effekter i verksamheterna. De ovan angivna hanteringsuppgifterna är just vad datorstöd är bra på, särskilt som datorstöd NUMERA också har betydligt bättre grafiska och andra presentationsmöjligheter så har effektiviteten i att använda datorstöd ökat.

Nedan skall vi gå in på vilka egenskaper olika datorstöd bör ha för att passa i olika dataadministrativa situationer. Vi skall också se på hur olika datorstöd kan se ut som används till andra närliggande uppgifter, t.ex. systemutveckling, databaskonstruktion, informationssökning. Ofta är det vettigt att ha samma datorstöd till flera sådana situationer.

3.2 Historik och nuläge

Datorstöd för dataadministrativ verksamhet har varit ganska primitiva. Det gäller både deras förmåga att representera mer uttrycksfulla beskrivningar och även deras gränssnitt. Sådana verktyg har hittills blivit använda i begränsad omfattning, företrädesvis i större organisationer. Detta medan datorstöd för systemutveckling på senare tid börjat bli mer kraftfulla.

Mer kraftfulla verktyg för dataadministrativ verksamhet verkar dock vara på väg, och verktyg som kan stödja dessa båda områden verkar nu vara inom räckhåll.

Dataelementkataloger

I databehandlingens barndom var de datorbaserade systemen små och isolerade i verksamheterna. Man såg då inga behov av att separat beskriva de dataelement som varje applikation använde sig av i något som skulle kunna sägas vara en dataelementkatalog eller liknande. Programmeringsspråken, t ex Cobol, hade inte några faciliteter för något som skulle kunna kallas dataelementkataloger, t ex copyfunktioner, utan register och arbetsareor beskrevs i varje enskilt program.

Så småningom blev antalet system och register så pass omfattande att man fick problem att ha tillräcklig ordning och reda på framförallt dataelement som beskrev registerinnehåll. Olika program kunde ha olika längd på en variabel, vilket ledde till fel när dessa skulle samverka. Samma dataelement benämndes olika på olika rapporter, något som förvirrade läsaren.

Cobol och andra programmeringsspråk fick med tiden en copyfunktion som underlättade programmerarens arbete, eftersom registerbeskrivningarna gjordes en gång och kopierades in i respektive program och man fick färre fel. Ett behov fanns att registrera dataelementen endast en gång med en definition och format. Många företag byggde därför sin egen (eller köpte) datakatalog (rättare sagt dataELEMENTkatalog). Vanligen var den helt fristående och passiv i förhållande till programmen. De mera avancerade katalogerna kunde till och med generera copytexter för ett eller flera programmeringsspråk. Ibland tillkom en befattning som var ansvarig för innehållet i dataelementkatalogen och som kontrollerade att definitioner och format var rimliga och att man i möjligaste mån återanvände befintliga dataelement i stället för att registrera synonymer. Det var dock svårt för dessa personer att göra ett bra jobb eftersom de ofta betraktades som byråkrater, poliser och flaskhalsar. Det var lätt för en programmerare att låta bli dataelementkatalogen eftersom den inte var tvingande.

På universitet och högskolor experimenterades det med mera avancerade kataloger ("fipotek") men dessa användes knappast av näringslivet.

Dataelementkatalogerna blir datakataloger

De kommersiella datakatalogerna fick fler och fler funktioner. Från uppgifter i katalogen kunde man generera databasscheman, klassificera definitionerna och dokumentera applikationernas olika beståndsdelar i ett nätverk. Fråge- och rapportmöjligheter tillkom för korsreferenser och ingår-i-analys.

Fortfarande spelade datakatalogen en ganska passiv roll och många företag misslyckades att få dess innehåll i fas med de applikationer och databaser de skulle beskriva.

Datakatalogen var i allmänhet ADB-avdelningens verktyg och betraktades med oförståelse av linjeverksamheten. Detta berodde delvis på att dialog och terminologi var alltför tekniskt orienterad. Att satsa på datakatalogsystem var en stor och kostsam uppgift som skulle ge utdelning i form av snabbare systemutveckling och billigare förvaltning först efter ett antal år. Eftersom det var svårt att övertyga linjechefer och i vissa fall AU-chefer om det riktiga i en sådan satsning bedrevs detta arbete många gånger av metodintresserade "eldsjälar" mer eller mindresom en "underground"-verksamhet.

Metoderna ligger steget före hjälpmedlen

På metodsidan började man vid denna tid att arbeta med flödesorienterade modelleringstekniker, t ex ISAC och Yourdon. Ansatser för datamodellering befann sig i sin linda. Datakatalogsystemen började anpassas för att kunna dokumentera enligt dessa ansatser, inkluderande även beskrivningar av kopplingar till implementerade system och databaser. Grafiska notationssätt var emellertid en viktig del av de aktuella beskrivningssätten och dessa fick matas in i textform i verktygen. Primitiva rapporter kunde tas ut på radskrivare. Man kan säga att hjälpmedlen inte längre höll jämna steg med metoderna.

Integrerade datakataloger

Parallellt med dessa fristående datakataloger arbetade programvarhusen med att tillämpa samma teknik. Rapportgeneratorer försågs med ett directory över dataelement och filer så att specificering av rapportinnehåll förenklades.

Denna idé utvecklades vidare i de mera kompletta utvecklingsmiljöer som programvarhusen lanserade. Dessa bestod av databassystem, programmeringsspråk (ofta på högre nivå än Cobol), skärmbildsmålare och slutanvändarfunktioner, t ex rapportgenerator, frågespråk mm. Hjärtat i dessa system var den integrerade datakatalogen där databaser, dataelement, bildlayouter, program m m beskrevs och dessa uppgifter hämtades till de olika funktionerna.

Fjärde generationens verktyg

Dessa miljöer siktade mot att höja produktiviteten i systemutvecklingsarbetet genom att uppgifter om dataelement m m inte behövde beskrivas i program, bildskärmar, rapport- och frågespråk eftersom dessa uppgifter redan fanns i den integrerade datakatalogen. Verktygen innehöll också programmeringsspråk av högnivåtyp som gjorde vissa automatiska antaganden om lagring etc. (defaults). Dessa i vissa avseenden produktiva

hjälpmedel kallades ofta för "fjärde generationens verktyg". Datakatalogen i dessa verktyg innehöll vanligtvis de uppgifter som krävdes för att beskriva databaser och program. Beskrivningar av konceptuell karaktär, t ex funktionshierarkier, dataflödesdiagram, och datamodeller kunde normalt inte beskrivas i dessa verktygs datakataloger.

Denna grupp verktyg fick dock ej de stora effekter och den genomslagskraft inom systemutveckling och förvaltning som många i av branschen ett tag hoppats.

CASE-verktyg

De senaste åren har det vuxit fram en ny typ av hjälpmedel. Dessa avser bl a att stödja användningen av olika konceptuella beskrivningar, vilka bl a kan utgöra specifikationen av en verksamhet och de informationssystem man vill skapa för denna. Dessa specifikationer kan (om de är tillräckligt detaljerade) utgöra underlag för att mer eller mindre automatiskt generera det informationssystem man vill skapa.

CASE-verktygen har börjat användas i olika faser av specifikationsprocessen. Det gäller även mycket tidiga faser av en specifikationsprocess där man skisserar lösningar för verksamhet och informationssystem.

CASE-verktygen är maskinellt vanligen baserade på persondator eller arbetsstation och utnyttjar dessas möjligheter med grafisk bildskärm, fönsterteknik och mus. Grafiska beskrivningar, t ex datamodeller och flödesmodeller kan ritas upp, manipuleras och knyts samman och registreras automatiskt i dictionary, eller "encyklopedia". Det kan ligga lokalt i PC:n eller på någon 'central' belägen större maskin. De flesta av dessa s k CASE-verktyg (Computer Aided Software Engineering) kan köras fristående på arbetsstationen men det finns en trend också att integrera dessa verktyg med en större värddators datakatalog.

Många CASE-verktyg kan också från specifikationsmodeller i varierande grad generera fungerande databasscheman och programkod. Programkoden kan vara mer eller mindre fullständig.

CASE-verktygen kommer troligen att 'driva' på intresset för modellering genom att dessa verktyg kräver kunskaper av denna typ. Intresset för CASE-verktyg drivs också på av att man kommer att kunna skapa sitt system på specifikationsnivån genom att i stort sett all beskrivning av systemet sker på denna nivå. Förändringar av systemet kommer också att kunna ske på denna nivå. Betydelsen av konventionell programmering i stora volymer minskar på sikt.

CASE-verktygen av idag är vanligen till stor del metodberoende. Detta tenderar att styra innehållet och strukturen i datakatalogen. Nya typer av CASE-verktyg kan dock komma runt detta metodberoende, som styrs hårt av verktygsleverantören.

Det tycks som om CASE-verktygen driver på utvecklingen kring dataadministration eftersom datakataloger med ett så omfattande innehåll som ett resultat av en utvecklingsinsats med nödvändighet bör "vårdas", vidareutvecklas och samordnas. För att man skall uppnå de potentiella effekterna med CASE-teknologi så krävs det att de specifikationer som skapas under utvecklingen också förvaltas. De kan då användas till vidare förändringsarbete. Man slipper starta ifrån "noll" ifråga om hur saker "ser ut" och vilka samband som råder.

En trend är att många företag vill styra upp sin långsiktiga planering av information, system och datorteknologi med starka kopplingar till verksamhetens struktur och mål. Även här växer fram datorstöd av CASE-typ med datakatalog som kan integreras med andra datakataloger. De modeller som växer fram under detta arbete måste kunna kopplas till de modeller som tas fram under informationssystemutvecklingsarbetet.

3.3 Dataadministration - vilken användningssituation har vi ?

Det är svårt att diskutera datorstöd och egenskaper hos sådana om man inte har en uppfattning om hur den användningssituation ser ut som man vill använda datorstöden i.

En analys av användningssituation är, eller bör utgöra, en del av den strategi som man använder sig av för att bedriva dataadministration.

Den valda strategin bör grundas på de mål man ytterst har för sin dataadministrationsverksamhet. Målen kan vara mer eller mindre uttalade, men man bör sträva efter att formulera dem klart.

Delgruppen i projektet "Dataadministration", som arbetade med datorstöd kom snabbt in på olika möjliga typer av STRATEGIER för dataadministration. Detta eftersom man eljest hade svårt att formulera önskade egenskaper för datorstöd.

Nedan skall vi kort beskriva fyra typer av strategier för datorstöd. Strategierna kommer att implicera olika behov av funktionalitet för datorstöd som kan ingå i de olika strategierna. De olika strategityperna har bland annat stor betydelse för vilka krav som skall ställas på verktygens förmåga att representera modeller och modelldelar som hänger samman med varandra.

För att man kan ska få stora effekter av dataadministration i en verksamhet är det viktigt att det sätt på vilket dataadministration tillämpas stämmer med de principer för verksamhetsstyrning som finns. Dataadministration skall stödja det sätt på vilket verksamheten styrs. Dataadministration är mycket nära kopplad till verksamhetsstyrning. Olika styrformer för dataadministration ställer skilda krav på hur denna bör bedrivas. En offensiv dataadministration bör t. ex. hela tiden se till att man förändrar det sätt varpå dataadministration bedrivs så att den anpassas till de styrprinciper som utvecklas i verksamheten.

Nya styrformer i verksamheterna blir alltmer vanliga i förvaltningar och företag, t.ex. olika varianter av decentraliserad styrning.

Att koppla ihop dataadministration med aktuell form för verksamhetsstyrning är ganska nytt. En utveckling härvidlag drivs dock troligen på av erfarenheter från fall där konventionella sätt att bedriva dataadministration har använts, fastän nya former för styrning av verksamheterna har börjat tillämpas. Effekterna av dataadministration har då varit små, om några alls.

Nedan följer fyra typer av verksamhetsstyrning och de krav dessa ställer på modeller i samband med dataadministration.

global modell

1) Starkt centraliserad verksamhet, där man dataadministrationsmässigt strävar efter att skapa EN datamodell som gäller för hela organisationen. Effekterna blir speciellt markerade av denna strategi för mycket stora organisationer. Vissa organisationer i Sverige har provat denna strategi.

lokal modell

2) En mycket decentralt styrd organisation eller verksamhet där man har en datamodell i VARJE delverksamhet men där man har mycket litet kommunikationsbehov mellan delarna. Följaktligen behöver man inte lägga så mycket tid på att beskriva information som flyter mellan delverksamheterna.

kärnmodell

3) En verksamhet är relativt decentral till sin karaktär och man vill att de olika delverksamheterna skall ha stort lokalt ansvar, vilket resulterar i att man skapar datamodeller över resp delverksamhet. Emellertid har man vissa centrala begrepp som man av samordningsskäl, vill skall gälla över hela organisationen. Man definierar en liten central 'KÄRNMODELL', vilken man vill skall vara en del av varje delverksamhets modell och skall beaktas för alla nya projekt. Kärnmodellen skall lagras i centralt datorstöd (dictionary).

sambandsmodell

4) Man har en mycket decentral verksamhet, men där man har ett relativt stort kommunikationsbehov mellan delverksamheterna. Varje delverksamhet har sin egen datamodell. För att klara kommunikationen mellan delverksamheten har man dock speciella beskrivningar för att beskriva den information som kommuniceras. Då den information som kommuniceras beskrivs i helt separata modeller, och där man har speciella 'mappningar' till de olika lokala modellerna, så har man stor frihet att också ändra i sina lokala modeller utan att beskrivningar av den information som kommuniceras behöver påverkas. Resultat blir då en hög lokal frihetsgrad för de lokala delarna.

Nedan återkommer vi till hur dessa verksamheter med sina respektive styrstrukturer kommer att kräva att olika strategier för dataadministration måste tillämpas. Dessutom krävs datorstöd med olika egenskaper. De verktyg som skall användas för dessa olika strategier måste bl.a. vara försedda med olika representationsmöjligheter som svarar mot de olika inriktningarna för dataadministration.

3.4 Verktyg för dataadministration

Att arbeta med dataadministration kan underlättas med hjälp av datorstöd. Dessa bör ha en lämplig funktionalitet och en lämplig interaktionsform för ändamålet. Det har visat sig att alltför triviala och allmänna hjälpmedel ger ett otillräckligt stöd.

Arbetsgruppen för datorstöd i detta projekt har försökt uttrycka en definition av datorstöd för dataadministrativ verksamhet:

"Maskinellt hjälpmedel för att representera och hantera beskrivningar av data, informationssystem och verksamheter."

Då verksamheten är av någorlunda storlek och om informationssystemen är till viss del datorbaserade så visar erfarenheten att det är mycket svårt att upprätthålla tillförlitliga och korrekta beskrivningar utan lämpliga hjälpmedel.

Om man har ambitionen att ha korrekta beskrivningar av data, informationssystem och verksamheter i en större organisation så ökar behovet av en viss uppsättning av beskrivningsbegrepp för att göra modellerna tillräckligt uttrycksfulla och fullständiga. Därav stiger den totala volymen av beskrivningsuttryck snabbt och då ökar också behovet av datorstöd för hantera dessa beskrivningar effektivt.

3.5 Olika typer av datorstöd och dessas samband med datakatalogsystem

Olika representationsnivåer i verktyg

Olika verktyg har olika förmåga att hantera olika typer av beskrivningar. När det gäller datorstöd och dessas förmåga i detta avseende kan vi se fyra nivåer.

Meta-Meta-nivå	Basbegrepp tillgängliga i ett verktyg, när det gäller att hantera modeller över huvudtaget i ett verktyg. Dessa grundbegrepp kan man sedan använda för att sedan bygga upp sin hantering av de modelleringsbegrepp man är intresserad av att hantera i sitt verktyg. (Dessa aspekter är speciellt intressant för s.k. CASE-skala.)
Meta-schema-nivå	De begrepp som vi väljer ut att ingå i olika modelltyper och hur dessa modelleringsbegrepp förhåller sig till varandra.
Modellnivå	Modeller som vi skapar för att beskriva en viss verksamhet med dess ingående verksamhetsaspekter, informationssystem, data, etc.
Instansnivå	Instanser av en verklighet modellerade i en viss modell och lagrade i ett verktyg, t ex olika instanser av en viss objektclass som ingår i en viss datamodell (modelltyp).

Nu övergår vi till att se på olika typer av verktyg bl.a. sett mot ovanstående nivåer.

Datakatalogsystem

Datakatalogen har normalt generella funktioner för hantering av informationskomponenter och dessas datamässiga representation, och utgör ett "datakatalogsystem". Det kan finnas dialogfunktioner med menyhantering, panelhantering och kommandon för inmatning eller ändring av information i datakatalogen. Vanligtvis finns någon form av frågespråk för analyser och korsreferenser samt funktioner för rapportgenerering. Dessa och övriga funktioner i ett datakatalogsystem behandlas mera i detalj i ett senare avsnitt.

Delverktyg runt ett dictionary (encyklopedia)

Man kan betrakta datakatalogen som kärnan bland datorstöd för dataadministration. Till denna kärna har olika leverantörer och företag kopplat verktyg och andra programsystem. ISO/IRDS talar om "service interface", som gör det möjligt att koppla katalogen till andra programsystem t ex:

- andra datakataloger
- in- och utmatning via t ex kommandon eller paneler
- databassystem
- kompilatorer
- datorstödda utvecklingshjälpmedel
- 4:e generationens verktyg, programgeneratorer
- analys-, modellerings- och designstöd
- applikationsprogram

Dessa service-interface indelas enligt ISO i grupperna:

- Panel interface
- Command Language interface
- A class of other man machine interfaces
- Command Language Application Program Interface
- Import/Export File interface

Modelleringsstöd

Modeller av olika slag (t ex datamodeller, verksamhetsmodeller), blir allt viktigare inom dataadministration och vi kan se allt fler datorstöd som kan rita, lagra och bearbeta modeller. De flesta modelleringsstöd som finns på marknaden är avsedda för persondator eller arbetsstation. Modellerna manipuleras i grafisk form på bildskärmen men lagras i en datakatalog, encyklopedia eller liknande lokalt i PC:n. Vissa av dessa modelleringsstöd är helt fristående medan andra kan kommunicera med andra datakataloger och programsystem. Det förekommer även icke-grafiska hjälpmedel för att hantera och analysera modeller.

Databassystem

Konceptuella informationsmodeller i datakatalogen kan i vissa datorstöd översättas till logiska register och databasscheman för olika databashanterare. I extremfallet är databashanteraren och datakatalogen så intimt kopplade att katalogen är en nödvändighet för databassystemet. Normalt genereras databasscheman från katalogen när databasen skapas men det finns exempel på databashanterare som konsulterar datakatalogen för varje enskild databasåtkomst.

Beroende på databashanterare kan det finnas funktioner för generering av "views" mot databasen, t ex för ett visst applikationsprogram. Datakatalogen kan också tänkas innehålla uppgifter som används av databashanterarens säkerhetssystem.

Utvecklingsverktyg

Verktyg för utveckling av applikationer kan vara knutna till datakatalogen på en olika sätt. Vanligt är att generella beskrivningar av register kan omvandlas till databeskrivningar i aktuellt programmeringsspråk, t ex i form av copytexter.

Moderna utvecklingsverktyg är många gånger integrerade med datakatalogen på ett sådant sätt att det är tillräckligt att ange namnet på en programvariabel, ett fält på en bildskärm eller en term i ett register eller en rapport. Verktuget hämtar sedan övriga uppgifter som är associerade med variabeln/termen från katalogen. Det kan röra sig om lagringsformat, längd, typ, rubrik, tillåtna värden m m.

I så kallade CASE-verktyg har integrationen drivits längre genom att hela eller delar av applikationen specificeras och lagras i datakatalogen. Program, skärmbilder och register kan sedan genereras mer eller mindre automatiskt.

Slutanvändarverktyg

Rapportgeneratorer, frågespråk, grafikprodukter m m av slutanvändarkaraktär kan ibland dra nytta av databeskrivningar i datakatalogen.

System- och dataplaneringsverktyg

Strategisk planering av ett företags information och applikationer kan dra nytta av datorstöd. Med dessa verktyg kan man dokumentera övergripande modeller över företagets funktioner, data, mål, problem, framgångsfaktorer och annat, ofta i form av grafiska modeller och matriser. Dessa uppgifter lagras i en datakatalog som kan vara integrerad med andra datakataloger eller som kan kommunicera med sådana. Exempelvis kan en övergripande datamodell utgöra en utgångspunkt för en mera detaljerad modell över en delverksamhet.

3.6 Effekter av datorstöd

Datorstöd kan ge effekter i styrning, kvalitet och produktivitet, men dessa är ofta förstärkningseffekter. Bakom verktyget måste finnas genomtänkta och accepterade principer och modelleringskunnande. Vi måste därför granska de principer som ligger bakom den möjliga användningen av datorstöd.

Styreffekter

I ett företags systemutvecklings- och systemförvaltningsorganisation är det eftersträvarvärt att hålla en enhetlig standard för uttrycksformer och fullständighet i modelleringsarbetet per lämpligt verksamhetsområde. Detta är ofta grunden för att få effekter av datorstöd i arbetet. Accepterade standards och principer i beskrivningsarbetet ökar möjligheterna för att flera personer skall kunna kommunicera och samarbeta och det underlättar byte av arbetsuppgifter och inte minst underlättar det vid byte av systemförvaltningspersonal.

Att åstadkomma ett enhetligt arbetssätt är svårare ju större organisationen är. En heltäckande datakatalog som alla applikationer är knutna mot utgör ett hjälpmedel för att stödja ett enhetligt arbetssätt också över en mycket stor organisation. En sådan katalog gör det också möjligt att successivt vidareutveckla det enhetliga arbetssättet i väl definierade steg. Genomförandet av stegen kan dessutom kontrolleras maskinellt via datakatalogen.

Kvalitet

Uppfyllandet av kvalitetskrav för ett visst arbetsresultat kan kontrolleras. Detta om kvalitetskriterier i tillräcklig omfattning definierats. Detta är dock mycket närliggande då en modelleringsstradition börjar sprida sig i en organisation. Genom bl a en datakatalog ges frågemöjligheter som kan utnyttjas för kvalitetskontroller vilka t.ex. kan göras i stickprovsform på ett smidigt sätt.

När det gäller modeller som är dokumenterade i datakataloger så kan också kvalitetskontroller göras av annan typ t.ex. vad gäller motsägelser i modellerna, s.k. konsistenskontroller.

Produktivitet i systemförvaltning och systemutveckling

Systemförvaltning

I systemförvaltning är det vi vet att datakatalogen ger produktivitetsvinster. Drift av dessa verktyg drar förhållandevis låga kostnader. De ger sökmöjligheter som underlättar analyser och de har en styrande effekt så att ändringar görs i enlighet med etablerat gemensamt arbetssätt. Resursbesparing i hantering och manipulering av förvaltade system är effekter härvidlag.

Systemutveckling

Inom systemutveckling bör man skilja på utveckling av permanenta system som innehåller databaser och uppdatering av dem, från sådana system som är av informationsåtervinningskaraktär, kanske rent av engångskaraktär.

Vid nyutveckling ger datakatalogen större nytta ju större omfattning det nya systemet har. Det beror på att man via katalogen får hjälp med att hålla inbördes konsistens mellan systemets komponenter då det inträffar ändringar i systemets förutsättningar under utvecklingstiden. I takt med att vi allt mer närmar oss en situation då system utvecklas via "CASE" så kommer ett datakatalogsystem att ge en klar produktivitetsökning.

I de fall då man gör ett system som är av karaktären informationsåtkomst ur befintliga databaser så har man nytta av katalogen för att finna sina data. Vid utveckling av andra systemtyper har man mer nytta av datakatalogen när det gäller att definiera den information som systemet lagrar som ny och omvandlar.

Det är en fördel om man särskiljer de här båda typerna av system med klara gränslinjer, så att man kan särbehandla dem också med avseende på hur de skall relateras till en datakatalog.

Ökad livslängd på system

Datakatalogen ger överblickbarhet och en ökad kvalitet genom en bibehållen **konsekvent standard** i de tillägg och ändringar som successivt görs i ett system under dess livslängd ofta av flera "generationer" systemansvariga. Det ger en längre livslängd på systemet i de fall då det är systemets inre kvalitet som bestämmer dess livslängd.

I de fall där systemets livslängd främst beror på **ändrade förutsättningar i verksamheten**, snarare än dess inre kvalitet, är det ännu mera så att en datakatalog ökar systemets livslängd. Man kan med datakatalogens hjälp bättre analysera och överblicka vilka ändringar i systemet som en viss verksamhetsförändring skulle kräva. Genom att man har den överblicken "vågar" man satsa på lösningar som innebär omfattande ändringar i befintliga system. Utan den överblicken skulle man tvingas att säga helt nej till ombyggnadsmöjligheten och därmed endast ha kvar alternativet att bygga ett helt nytt system då verksamheten kräver större ändringar. Den högre kvaliteten i systemen och överblickbarheten som en datakatalog ger, ger därmed en högre ändringsberedskap.

I det fall man inte kan erbjuda en hög ändringsberedskap inför en verksamhetsändring så får man lätt den situationen att en grupp användare som behöver systemstöd för en ny verksamhet, men som inte kan få det, tvingas göra någon lösning i form av att man kommer överens om att använda några delar av det befintliga systemet på ett icke avsett sätt, men på ett sätt som kanske löser det aktuella verksamhetsproblemets systemstöd.

Den typen av lösningar blir normalt inte väl dokumenterade och om systemet har andra användare så kan den "nödlösningen" ställa till stora problem som man måste åtgärda genom ytterligare underhållsinsatser på systemet för att det överhuvudtaget skall fungera för de andra användarna.

Bristande ändringsberedskap gentemot verksamhetsändringar gör att nödlösningar måste forceras in i systemen med ökat underhåll och förkortad livslängd som följd. Datakatalogen bidrar till att förbättra den situationen.

Om man i alla fall tvingas till att göra den här typen av nödlösningar på grund av att man inte hinner anpassa systemet på ett riktigt sätt, så ger katalogen ändå ett positivt bidrag genom att den ger överblick så att man får bättre underlag för att välja en så "snäll" nödlösning som möjligt. Katalogen kan dessutom användas för att dokumentera och kommunicera ut vilka nya eller delvis nya betydelser som lagts in i befintliga fält i systemen. Då är det viktigt att katalogen inte är alltför stel i sina former utan att den kan användas också för att dokumentera inte fullt "renläriga" lösningar.

Bättre effekt genom flexibla verktyg

Det är viktigt att man kan kopiera ett befintligt system och användarkopian som bas för ett nytt system, dvs man gör anpassningar i det kopierade systemet ungefär som man normalt måste anpassa ett utifrån köpt systempaket. Kanske vill man använda ett och samma system för att stödja flera skilda verksamheter, inte genom att man kopierar systemet och skapar två instanser, utan genom att man systematiskt utnyttjar en och samma systemkomponent på flera logiskt skilda sätt, parallellt.

3.7 Innehåll och representation

En viktig aspekt med verktyg för dataadministration, systemutveckling och förvaltning är att man kan använda de modelleringsbegrepp man är intresserad av så att man kan skapa de modeller man behöver på ett korrekt sätt.

Det är mindre lämpligt att behöva göra 'konstgjorda' representationer i verktyget för att beskriva begrepp som man är intresserad av.

Vi samlar ett antal aspekter som vi vill beskriva avseende 'verkligheter', formulerar modelleringsbegrepp för dessa och vi har därmed skapat en modelleringsteknik. Av överblickbarhetsskäl samlar vi inte för många aspekter i en modelleringsteknik utan formulerar i dessa fall hellre flera modelleringstekniker.

Moderna systemutvecklingsmetoder har ofta flera modelleringstekniker. Moderna encyklopedier och dictionaries måste också klara att hantera detta.

Om det finns flera modelleringstekniker behöver man kunna definiera samband mellan dessa eftersom begreppen i modelleringsteknikerna har samband. Det är därför också av stort intresse att i ett modernt verktyg kunna representera samband mellan begrepp i olika modelleringstekniker. Exempel är att man för flöden i en funktionsmodelleringsteknik vill beskriva hur innehållet i flödena ser ut med hjälp av referenser till attribut i en datamodelleringsteknik.

När det gäller datorstöd finns två viktiga grupper:

- 1) Datorstöd som är skapade för att stödja en viss modelleringsteknik eller en grupp av sådana.
- 2) Datorstöd som är konstruerade för att anpassas till en viss eller en grupp av modelleringstekniker. Det innebär att hjälpmedlet kan definieras eller 'ställas in' så att det lär sig en viss modelleringsteknik, dess begrepp och dess syntax.

Flertalet av de datorstöd som finns marknaden är av typ 1.

När det gäller den senare gruppen så avses inte bara att en symbol kan ändras till form och till färg, utan att man kan definiera nya modelleringbegrepp, symboler för dessa, syntax-regler, kontroller, lämpliga utsökningar, referenser till modelleringbegrepp i andra modelleringstekniker.

Representation av successiv utveckling/detaljrikedom

En vanlig egenskap hos datorstöd, som många är intresserade av, är att man successivt kan detaljera grova utgångsbegrepp så att man får en alltmer exakt modell genom att ytterligare delbegrepp successivt skapas. Man kan till exempel till slut få fram körbara program. Exempel är grova funktioner som delas upp så att man till sist skapar exakta programspecifikationer för minsta delfunktioner i en funktionsstruktur.

Ett annat fall kan vara att initiala modelleringbegrepp efter förfining kan övergå till ett annat begrepp. Då vill man kunna representera denna övergång till det nya modelleringbegreppet, och man vill också kunna använda sig av denna referens när man sedan förändrar och uppdaterar ihopkopplingen av begreppen. Exempel kan vara s k initialt infångade koncept (begrepp) i en verksamhet som senare under mer exakt datamodellering övergår i antingen entiteter, attribut eller domäner.

3.8 Gränssnitt människa-maskin

Det har visat sig bland många av de företag som är intressenter till SISU att användningen av datakataloger har en begränsad användning hitintills, även om man som policy satt upp som riktlinje att man skall använda denna typ av verktyg för utveckling och förvaltning. Förmodligen är en anledning till detta att verktygen har haft olämpliga gränssnitt mellan maskin och människa. Verktyg för utveckling och förvaltning måste ha goda gränssnitt för att bli attraktiva.

Kommandoorientering

Många verktyg har huvudsakligen varit kommandoorienterade. I vissa verktyg har detta varit den enda interaktionsformen.

Med kommandoorienterade menas här att verktygen körs genom att uttryck skrivs radorienterat på en terminal enligt viss syntax för att exekvera olika funktioner i verktyget. Det kan gälla att mata in vissa tillägg i modellen etc.

I vissa sammanhang betraktar man denna typ av gränssnitt som lite gammaldags, men kommandoorienterade gränssnitt kan vara mycket lämpliga för avancerade användare till ett verktyg.

Grafiska gränssnitt

På senare tid har datorer och bildskärmar utvecklats till att kunna arbeta med grafisk presentation, t.ex. med grafiska symboler.

Eftersom det visat sig att grafer och symboler i många fall kan användas för att göra det lättare för användaren att uppfatta strukturer så har grafiska gränssnitt snabbt blivit populära.

Modeller kan matas in i ett verktyg med hjälp av att placera ut symboler i en graf. Man kan manipulera modellen genom att förändra grafen på olika sätt. Man kan ibland också låta verktyget självt rita upp en graf över en struktur som sökts ut eller som kanske importerats ifrån ett annat verktyg. Ibland har man också möjlighet att utföra funktioner i verktyget genom att peka på vissa symboler (ikoner) på skärmen.

Man behöver alltså inte skriva så mycket på terminalen, att peka i en meny kan vara tillräckligt för att utföra avancerade och omfattande operationer i verktyget. Grafiska gränssnitt har ökat intresset för verktyg som stödjer modellering. Nya grupper av användare av verktygen har kunnat utnyttja dessa utan större utbildningsinsatser på verktygen i sig.

Det grafiska gränssnittet innebär att kommunikationstödseffekten av modellering och verktyg ökat. Inte minst direkt vid skärmen genom att man kan sitta invid verktyget när det t ex gäller verksamhetsmodellering. Verksamhetsrepresentanter och modelleringsexperten kan då sitta direkt vid skärmen och diskutera en graf på terminalen. I dessa fall är det naturligtvis en fördel om skärmytan är stor och god detaljåtergivning.

Helt nya gränssnitt

I framtiden kan man också räkna med att "inmatning" och "utmatning" till/från verktygen kan ske genom helt nya gränssnitt såsom t ex 'voice input' (taligenkänning) och genom att vissa resultat presenteras via syntetiskt tal.

Flerfönstertechniker

På senaste åren har också de grafiska gränssnitten förfinats genom att man kan betrakta och arbeta med flera fönster på en stor grafisk skärm.

Detta innebär att man kan arbeta med flera bilder eller modeller samtidigt utan att få långa omritningstider för varje 'skärmbyte' man vill göra jämfört med en konventionell terminalsession. De olika fönstren kan 'läggas över' varandra helt eller delvis, kan dras undan så att underliggande syns, etc. Med fönsterhanteringsteknik kan ett stort djup av undermenyer användas effektivt. De 'kan dras ner' i olika fönster etc.

I verktyg för modelleringsstöd kan flerfönstertekniker vara särskilt lämpade. Man kan bl.a. ha olika typer av dokument i vissa fönster, olika grafiska fönster för olika modelltyper i var sitt fönster etc. Man kan genom att i verktyget öppna olika dokument och grafter förstärka olika arbetsätt och därmed sammanhängande metod.

3.9 Tänkbara grupper av verktygsfunktioner

Detta avsnitt behandlar funktioner man kan finna eller skulle önska i ett datakatalogsystem, d v s den programvara som hanterar datakatalogen. Många av funktionerna kan också vara som önskvärda för CASE-verktyg vilka ju också ofta innehåller någon form av datakatalog, dictionary eller encyklopedia.

Schemadefinition

En grundläggande funktion är att kunna definiera och generera datakatalogens inre struktur d v s databasschemat för katalogen. (Jfr avsnittet om representationsnivåer i verktyg.) Vissa datakatalogsystem har en given struktur och saknar möjlighet att påverka denna struktur. I ISO/IRDS termer kallas denna beskrivning för "Information Resource Dictionary Definition".

Datakatalogens struktur, vilka objekt/medlemstyper som skall finnas, vad de ska heta och vilka relationer som finns mellan dessa, specificeras i ett särskilt språk. Beskrivningen kan också omfatta vilka attribut/termer som är aktuella för respektive objekt. Efter syntaxkontroll kan en schemadefinition genereras och en datakatalog skapas.

Utsökning och projicering

Att kunna söka ut en önskad delmängd ur datakatalogen och få resultatet presenterat tillhör datakatalogens viktigaste funktioner. Datakatalogen i en fungerande dataadministration innehåller en stor mängd information om företagets begreppsvärld, verksamhet, ADB-system och register. Det kan röra sig om konsekvensanalys vid ändring av ADB-system, ingår-i och består-av frågor, framställning av översikter, dokumentation m m.

För att kunna söka ut den delmängd som man är intresserad av behövs ett frågespråk där man kan specificera sitt urval. Frågespråket kan t ex vara kommandoorienterat eller ha stöd av menyer och paneler eventuellt kombinerat med grafiska tekniker.

Behörighet/sekretess

Det omfattande innehållet i en datakatalog måste kunna skyddas mot obehörig åtkomst. Vissa uppgifter kan vara av känslig eller hemlig natur. Övriga uppgifter bör skyddas så att inte någon av oförstånd eller illvilja ändrar eller förstör datavärden.

Ett behörighetssystem bör innehålla ett register över alla användare samt vilka operationer som de är behöriga att göra mot olika delmängder i katalogen. Olika delmängder av katalogen tillhör "ägare" som kan utdela olika slags behörighet till användare. Eventuellt kan ägaren delegera utdelning av behörighet till andra personer.

En användare kan t ex vara behörig att läsa vissa delar av katalogen och uppdatera andra delar. Det är önskvärt att kunna ange åtkomstbehörighet på så detaljerad nivå som möjligt, t ex att någon har rätt att uppdatera endast den verbala definitionen av en viss term, men inte dess lagringsformat.

Versionshantering

Innehållet i datakatalogen är under ständig revidering. Ett företags verksamhet, begrepp och ADB-verksamhet utsätts hela tiden för förändring. Fastställda beskrivningar av verksamhet och data måste bevaras i datakatalogen medan olika projekt och arbetsgrupper förändrar och vidareutvecklar beskrivningarna. Här finns ett behov av att kunna hantera olika versioner av innehållet i datakatalogen.

En avancerad versionshantering tillåter att en grundversion vidareutvecklas parallellt av flera projekt. Varje projekt kan i sin tur tillåta underprojekt att vidareutveckla nya versioner. Det innebär att en version av en beskrivning i datakatalogen kan baseras på en annan version som bygger på ytterligare en version i en trädliknande struktur. Systemet måste kunna hantera sammansmältning av utvecklingsversioner med grundversioner, upptäcka fel och konflikter. Parallella projekt måste skyddas från att störa varandra genom att de förändrar överlappande versioner.

Det krävs också att verktyget erbjuder en smidig administration av versionshanteringen, fråge- och analysmöjligheter m m.

Kontroll och regler

Datakatalogsystemet kan vara försett med olika möjligheter att specificera regler och kontroller för innehållet. Kontrollerna kan t ex röra fullständigheten i beskrivningen för respektive typ av objekt eller sambandet mellan olika beskrivningar. Det kan t ex vara väsentligt att kunna ange vilka uppgifter som är obligatoriska, vilka som skall var numeriska eller alfnumeriska, kontrollera rimligheten av värden, att namnstandard följs o s v. I det enklaste fallet gäller detta generellt för alla användare, medan det i vissa fall kan finnas behov av speciella kontroller/restriktioner för vissa användarkategorier, exempelvis så tillåter man linjefolk att författa termdefinitioner medan ADB-personal ansvarar för lagringsformat. Exempel på kontroller av samband mellan beskrivningar kan vara att termer måste ha en relation till ett objekt eller att en term inte får ha relationer.

Kontrollerna kan utföras direkt vid inmatningstillfället eller vid ett senare tillfälle, eller en kombination av dessa. Rena felaktigheter bör givetvis fångas omedelbart om detta är möjligt medan det kan vara praktiskt att tillåta icke kompletta och delvis felaktiga beskrivningar under vissa skeden av ett projekt.

Historik och loggning

En variant av versionshantering är att alla förändringar "loggas". Med hjälp av en sådan logg kan man rekonstruera beskrivningar bakåt i tiden, t ex en beskrivning av en databas, i ett visst ögonblick. För att detta skall vara en praktiskt användbar metod måste detta kunna göras selektivt.

Säkerhet, underhåll, backup

Datakatalogen bör betraktas som en komplett databas. Den bör ha funktioner som garanterar integriteten vid olika former av störningar, t ex strömavbrott, diskproblem och programavbrott. Normalt innebär detta funktioner för säkerhetskopiering (back-up), loggning av förändringar, rekonstruktion av databasen till utseendet vid viss tidpunkt, automatisk backning av förändringar vid programavbrott etc.

Portabilitet och distribution

Ett portabelt datakatalogsystem innebär trygghet och flexibilitet. Katalogen kan flyttas till annan hårdvara och till andra operativsystem utan större arbetsinsats. En datakatalog kan kopieras och placeras i flera miljöer.

Distribution av datakataloger innebär att en katalog fysiskt kan vara spridd på flera olika platser medan den ur användarens synpunkt uppträder som en enda logisk katalog.

Export/Import

Import till datakatalogen innebär att innehållet i katalogen kan hämtas från externa källor, t ex en annan datakatalog, ett datorstöd för modellering eller liknande. Export innebär motsatsen dvs att kunna hämta ut delar av innehållet för bearbetning i t ex en rapportgenerator, överföring till CASE-verktyg eller till en annan datakatalog etc.

Det är naturligtvis viktigt med en rigorös kontroll vid import så att inte felaktigheter förs in i datakatalogen. Detta krav blir speciellt starkt, när man i det importerande verktyget redan har en modell som utökas genom importen.

Utökade import/export-funktioner innehåller regelverk som kan översätta innehållet i en datakatalog till en annan. Ett specialfall av import är funktioner som automatiskt kan dokumentera befintliga ADB-system och register genom att tolka databasschema, programkod etc.

Systemdokumentation

Ett datakatalogsystem innehåller vanligen mer eller mindre avancerade rapportfunktioner. Dessa rapportfunktioner kan utnyttjas för att framställa kompletta beskrivningar av t ex ADB-system, verksamheter etc.

I dessa rapportfunktioner kan ingå ytterligare funktioner som ordbehandlingsfunktioner med möjlighet att ange olika typsnitt, inkluderande av bilder och diagram av olika slag etc.

Handledning och hjälp

Olika former av handledning och hjälpfunktioner kan ingå i ett datakatalogsystem. Hjälpinformation kan finnas direkt åtkomlig vid terminalen där man kan bläddra och söka instruktioner och förklaringar om kommandon, syntax, felmeddelanden m m. Hjälpsystemet kan även beskriva standards, riktlinjer etc.

Om datakatalogsystemet har anpassats till den egna organisationen med egna paneler, menyer, kommandon och egen intern struktur (metamodell) så bör också hjälpsystemet omfatta även dessa anpassningar.

Det är en fördel om hjälptexter etc också finns på svenska. Det är mindre lämpligt om datakatalogsystemets hjälptexter är på engelska medan tillägg etc skrivs på svenska.

Man kan tänkas sig mer intelligent hjälp genom att verktyget 'serverar' hjälp om det märker att man gör fel av en viss typ.

Manipulation av datakatalogens innehåll

Datakatalogen innehåller funktioner för att mata in, ändra och ta bort olika beskrivningar. Olika typer av dialog har beskrivits i tidigare avsnittet "gränssnitt människa-maskin".

En princip vid inmatning är att all indata skrivs i fri form med en editor enligt en viss syntax med nyckelord. En annan princip är formulärorienterad där olika fält på en bildskärm eller fönster med fält eller dylikt fylls i.

Kontroller för att hålla strukturen korrekt och fullständig kan göras i olika skeden i ett inmatnings- eller manipulationspass. När det är lämpligt att utföra kontrollerna kan bero på vilka typer av kontroller det är fråga om. Det kan också bero på vilka svarsreaktioner man vill ha av verktyget. Det kan exempelvis vara så att man vill skissa på modeller ganska ostört utan att få en mängd kontroller utförda omedelbart vilket kan 'trasa sönder' ett kreativt arbete. På särskilt kommando kan det emellertid vara mycket användbart att ha tillgång till ett antal mycket kraftfulla kontroller som både kontrollerar syntax och olika former av kvalitet i modellerna. Vissa kontroller bör dock alltid göras direkt, t ex av att regler för hur man får koppla två typer av modelleringsbegrepp inte bryts.

Eftersom datakatalogen innehåller en hel del löpande text är det viktigt med en bra editor, gärna med ordbehandlingsfunktioner.

3.10 Integration och distribution av datorstöd

Strategier för datasamordning

Olika typer av verksamheter med olika former för verksamhetsstyrning bör åtföljas av olika strategier och målsättningar för datasamordning. Samband mellan olika delar av verksamheten behöver finnas beroende av kommunikationsbehov men kan etableras enligt olika principer. Dessa principer leder till olika lösningar för samordning av terminologin / begreppsapparaten, uppgifter för dataadministrationsfunktioner etc inom verksamhetsområdet. Därför är det viktigt att klarlägga och definiera samordningsstrategin innan man söker lösningar i form av organisation, datakataloger, distribution av datorstöd etc. Nedan visas några exempel på möjliga målsättningar och strategier för datasamordningen. Här skall vi titta lite närmare på strategier ur dataadministrationens funktionella synvinkel:

- Enhetlig och global strategi
- Lokal strategi
- Kärnmodellstrategi
- Kommunikationsstrategi

a. Enhetlig och global strategi

Målsättningen är ambitiös och går ut på att åstadkomma en enhetlig begreppsapparat som täcker hela verksamheten. Denna strategi leder till stora möjligheter att integrera olika delar av verksamheten, dels mellan olika funktioner inom ett företag, men även mellan olika företag. Den ställer krav på en stark central DA-funktion inom organisationen och eventuellt även på lokala DA-funktioner beroende på vald lösning med hänsyn till distribution av dictionaries.

Samordningen kan åstadkommas på flera olika sätt. Man kan tänka sig lösningar med ett centralt dictionary eller olika kombinationer med ett central master dictionary som kopplas till distribuerade dictionaries. Vid distribuerade lösningar är det viktigt att uppdateringsansvar definieras strikt, och att de tekniska lösningarna, som installeras, ger stöd härför.

b. Lokal strategi

Då delar av verksamheten bedrivs utan samband och utan krav på kommunikation kan målsättningen vara att tillåta olikartade lokala begreppsapparater att existera inom ett antal "öar". Inom varje "ö" kan krav finnas att uppnå enhetlighet. "Öarna" kan utgöras av funktioner inom ett företag eller hela företag i en större organisation. I detta fall ställs inga krav på någon central DA-funktion, utan lokala DA-funktioner torde i regel vara tillräckliga. Möjligen kan centralt metodstöd tänkas för lokala DA-funktioner.

c. Kärnmodellstrategi

En målsättning kan vara att skapa en global enhetlig begreppsapparat inom ett begränsat område inom varje lokal "begreppsö" samt att acceptera lokala ej globalt samordnade begreppsapparater utanför det enhetliga området. För att uppnå denna målsättning krävs en begränsad global kärnmodell som kommunicerar med lokala modeller. Det ställer krav på en stark DA-funktion som ansvarar för kärnmodellen samt lokala DA-funktioner.

d. Kommunikationsstrategi

Ingen överordnad samordning förekommer av begreppsapparaterna som existerar inom lokala "öar". I den mån kommunikationsbehov förekommer mellan olika "öar" skapas ett enhetligt sambands- och kommunikationsschema. Inom de lokala "öarna" kan den lokala begreppsapparaten samordnas. Krav ställs på en central DA-funktion som ansvarar för sambandsmodellen.

Dataadministrationsrollen och uppgifterna för denna utformas i enlighet med vald strategi, och som framgår ovan kan den variera starkt beroende på vilken samordningsmodell som valts.

Tekniska aspekter

I den mån en distribuerad dictionary-lösning väljes är det viktigt att förhållandet styrande/styrd med tillhörande uppdateringsansvar klart definieras och att de organisatoriska och tekniska lösningarna anpassas härefter.

Vid kommunikation eller samordning mellan heterogena miljöer (t.ex. IBM mainframe/S36) bör gälla att begreppsapparaten skall kunna definieras enhetligt oberoende av den tekniska miljön.

3.11 Arvet av gamla applikationer

Vi utgår från att företag vill ha kontroll över sin maskinellt lagrade information. Detta åstadkoms bäst genom att man håller sina applikationer maskinellt kopplade till en datakatalog som ajourhålls i takt med att applikationerna vidareutvecklas. Via den katalogen kan man söka reda på vilka applikationer och delar av applikationer som berörs då en viss ändring måste göras.

I de flesta företag finns ADB-applikationer som motsvarar en investering som det skulle ta 10 år att åstadkomma med den takt man idag har på nyinvestering. De applikationerna som är i drift kräver också ungefär lika stor insats i årligt underhåll som man i företaget satsar på investering i nya system. Om man skulle välja att introducera en datakatalog i ett sådant företag genom att enbart knyta nya system till datakatalogen så skulle det kanske ta 20 år innan samtliga applikationer vore kopplade till den datakatalogen.

Jämför med ett företag som vill kodifiera sina produkter men bara i takt med att nya detaljer konstrueras. Det kommer då att ta mycket lång tid innan samtliga förekommande detaljer är kodifierade. D v s det kommer att ta mycket lång tid innan man kan nå några effekter på komponent- och system-nivå.

Samma situation uppstår i ett företag om man har delar av sina ADB-applikationer knutna till en datakatalog medan andra inte är det. Då en applikation knutits mot datakatalogen måste arbetssättet kring den, vid vidareutveckling och underhåll ändras beroende på att kopplingen till katalogen kräver vissa format och parametervärden som den "fria" applikationen inte behöver anpassa sig till.

Det innebär normalt ett ganska omfattande arbete att komplettera gamla applikationer med sådana uppgifter som kopplingen till katalogen kräver. För att det över huvud taget skall vara möjligt över en stor mängd applikationer så kräver det att man har möjlighet att göra det åtminstone till stor del via någon form av maskinell konvertering.

En lyckad konvertering beror i mycket hög grad på om det finns någon genomgående standard använd i de applikationer som skall knytas in mot datakatalogen. Självklart är det till stor fördel om samtliga applikationer ligger i samma miljö och är skrivna i samma språk. Eftersom det är fråga om dataprogram så har man ändå kompilatorns standard att luta sig mot om ingen annan skulle finnas. Det förekommer programpaket som läser och strukturerar om gamla Cobolprogram i ny form men med samma logiska resultat. Ett sådant paket skulle kunna kompletteras så att den återskapade koden försågs med format och parametervärden som den skulle kunna utnyttja som koppling gentemot datakatalogen.

Om man ändå tvingas välja att endast ta med en del av sina gamla applikationer vid en knytning mot datakatalogen så blir konsekvensen att man inför två skilda arbetssätt i sin vidare systemutveckling och i sitt systemunderhåll.

Det blir då väsentligt att man väljer den mängden så att man får ett så klart gränssnitt som möjligt mot de övriga applikationerna. På samma sätt om man måste kompromissa och endast ta med vissa typer av komponenter ur applikationerna, att man väljer att göra det på ett enhetligt sätt över alla applikationer.

Ett minimum för en datakatalog som introduceras med syfte att dokumentera ett företags dataresurs är att företagets viktigaste databaser knyts in mot den.

En knytning av en gammal applikation till datakatalog innebär att man förändrar eller åtminstone påverkar arbetssättet för systemutvecklingspersonalen. Det blir då mycket viktigt, om man vill lyckas, att man kan motivera varför ändringen genomförs och att den genomförs efter analys om hur den skall utformas i de detaljer som påverkar användarna, i detta fallet systemutvecklings- och systemförvaltningspersonal.

Kopplingen mellan ett gammalt system och katalogen är normalt bräcklig i ett inledningsskede och den är starkt beroende av att samtliga användare stödjer den. Det åstadkommer man bäst, om man lyckas göra kopplingen så osynlig som möjligt. Det är också viktigt i ett inledningsskede att målet bara sätts till att etablera kopplingen och till att åstadkomma så mycket nytta att datakatalogsatsningens första steg kan motiveras. Ytterligare krav på innehåll i katalogen bör man vänta med tills det nya arbetssättet har satt sig på bredden. D v s om man tar till för mycket är det stor risk att kopplingen misslyckas p g a att ändringssteget blir för stort för användarna som då inte accepterar det och stödjer förändringen.

Då man lyckats etablera en allmänt accepterad nivå på datakataloganknytningen så får man en bas att arbeta vidare från. Det vidare arbetet syftar då till att lyfta informationen i katalogen från den låga nivå som man måste ha från början upp till en nivå som man siktar mot mera långsiktigt. Det arbetet bör ske parallellt över alla de applikationer som finns inknutna mot katalogen. Till dels för att katalogen är ett utmärkt hjälpmedel i arbetet med att få till stånd ett förändrat arbetssätt. Man kan successivt införa allt fler hjälpmedel och kontrollmöjligheter i katalogen gentemot applikationsutvecklingen och därigenom få genomslag för ett förändrat arbetssätt i väl definierade steg. Exempel på generella nivåhöjningar som man kan genomföra på det här sättet är att man använder katalogen för att lagra enhetliga kodsekvenser som man vill ha kontroll över att de utförs på samma sätt i alla program t ex fältkontroller.

En helhet bör ändå eftersträvas, dvs så att samtliga applikationer åtminstone samtliga applikationer i företagets huvudmiljö/er knyts in mot datakatalogen. Det är viktigt även om det finns applikationer som man inte uppfattar som speciellt intressanta p g a att de snart skall avvecklas eller för att de är av ren bearbetningskaraktär, dvs det inte finns egna register inom de applikationerna som är av brett intresse inom företaget. Motivet för helheten är ändå att man slipper stödja två arbetssätt: ett med och ett utan kataloganknytning. Det skulle ge en sämre kvalitet i systemutveckling och systemförvaltning.

Konsekvensen av att knyta in samtliga applikationer på en och samma ambitionsnivå innebär att vi i katalogen får en blandning av information, både sådant som avser mycket väsentliga systemkomponenter och sådant som är ointressant ur verksamhetssynpunkt och som bara finns med av fullständighetsskäl eller av tekniska skäl.

Det är viktigt att det i katalogen går att kategorisera komponenter av samma typ i olika kategorier så att man lätt kan urskilja väsentliga komponenter från mindre väsentliga och att man kan ställa olika krav på hur detaljerad dokumentation som det skall finnas om dem i katalogen så att man maskinellt kan kontrollera att dokumentationsnivåerna efterlevs, åtminstone till formen.

3.12 Standardapplikationer och dictionary

Hur hantera köpta standardapplikationer?

När en standardapplikation nu och även i framtiden bedöms arbeta utan krav på samband med övriga delar av verksamheten kan man ifrågasätta om insatserna för en datasamordning med den övriga verksamheten är motiverade.

Om skäl finns för samordning kan den lösas på olika sätt med olika ambitionsnivåer.

- Man kan koncentrera sig på data och innehållet i databaser och viktiga datafiler.
- Man kan inskränka sig till data som är föremål för kommunikation med den övriga verksamheten.

Vid en samordning föreligger bl.a. följande uppgifter:

- Identifiera gemensamma data även om namn och terminologi i standardapplikationen och verksamheten i övrigt skiljer sig ifrån varandra.
- Lösa problematiken med olika namn på samma dataelement.
- Lösa problematiken med "snarlika" definitioner på data.
- Dokumentera standardapplikationens data i dictionary.

Om standardapplikationen har eget dictionary måste relationerna mellan ett "centralt" dictionary och standardapplikationens dictionary definieras och eventuellt krävs att bryggor byggs emellan dem. Det är viktigt att definiera var och av vem den styrande "master" uppdateringen skall ske. Olika situationer kräver olika lösningar. Exempelvis kan ett "centralt" dictionary vara helt styrande. Vissa standardapplikationer är flexibla och tillåter enkelt tillägg av nya dataelement. För att inte gå miste om vissa fördelar härmed kan standardapplikationens dictionary göras styrande för vissa delar men i övrigt följa det "centrala" dictionary. Sådana arrangemang ställer emellertid stora krav på tekniska lösningar samtidigt som möjligheterna att köpa software som fyller kraven torde vara synnerligen begränsade f.n.

3.13 Framtidens datorstöd

Inledning

Verktygen inom bl a CASE-området kommer att vara föremål för en fortsatt snabb utveckling. Förmågan hos datatekniken och dess gränssnitt mot användaren kommer att ge ökade möjligheter. Egenskaperna hos verktygen kommer att utvecklas i rask takt. Potentiellt kan man således åstadkomma en mängd saker med dessa verktyg.

Det kommer att vara nödvändigt att orientera sig om verktyg och deras potentialer för att planera verktygsstrategier i verksamheterna.

Men, verktygen kommer inte att kunna ersätta kunskap om systemutveckling och förvaltning. Man måste tvärtom kunna mycket om modellering och specifikation av informationssystem för att få effekt av verktygen och för att överhuvudtaget använda dem. Verktygen kommer att ge inga eller mycket små effekter om man inte har kunskap nog för att kunna sköta dem och utnyttja deras potentialer.

Man kommer att behöva lära sig mer om modellering, koppling mellan modelleringstekniker, metodiskt arbete, hur man skall kunna åstadkomma olika typer av kvalitet i modellering av olika slag, hur olika specifikationssteg är kopplade till varandra, etc.

Å andra sidan kommer framtiden också att medge egenskaper i verktygen som hjälper användaren att använda modelleringstekniker etc. på ett kvalitativt bra sätt. Det kan ske med olika typer av kontroller och analyser som kommer att vara tillgängliga. Det kan också innebära att verktyget kan ha lärande-sessioner inbyggda som man kan använda sig av för att lära sig viss metodik. Man kan också tänka sig att ett modernt verktyg rapporterar "intelligenta" instruktioner till sin användare genom att verktygen ger instruktioner eller ledande utsagor som stämmer med den metodik man arbetar med och baserat på det "fel" man har gjort i arbetet.

Modeller för specifikation

Framtidens verktyg kommer att bygga på att man gör specifikationer i stället för att "koda" systemet i alla sina detaljer. Det innebär dock att specifikationerna måste bli mer fullständiga än de är idag. Verktyget kan ge visst stöd i arbetet med att få specifikationer fullständiga.

Redan i dagens 4:e generationens verktyg kan vi se tecken på ett sådant mer "specifikationsorienterat synsätt", även om detta bara utgör en liten del i denna utveckling. Uttryck för att ange restriktioner och för att skapa effektiva lösningar för större tillämpningar kommer att behövas framgent.

Att arbeta specifikationsorienterat innebär inte nödvändigtvis att man måste bli formell med en gång". Ett lämpligt verktyg kan stödja mer skissorienterade utkast till en början men stödjer sedan en successiv detaljering.

Evolution och återanvändning

Att skapa system genom att definiera detaljerade specifikationer innebär på sikt också att förändring av system genomförs på specifikationsnivån och att man därefter "regenererar" system i en ny version.

När man sedan skapar nya system som stöd för t ex andra verksamheter, kan man plocka specifikationsdelar ifrån redan gjorda specifikationer och komplettera dessa med ytterligare eller delvis förändrade delar.

Arbete på specifikationsnivå kan på sikt underlättas av att man inte behöver "slänga allt gammalt" för att man inte har överblick och bygga helt nytt. En högre grad av successiv förändring kan tillåtas och man kan ändå hålla jämnasteg med en snabb förändring av kraven mot ett system.

Nya modelleringstekniker och metoder ställer nya krav på verktygen

Framtidens datorstöd och verktyg för systemutveckling, vidareutveckling och förvaltning kommer att ha en mängd nya funktioner. En orsak till detta är bland annat att framtidens modelleringstekniker kommer att vara mer utvecklade än de som används idag.

Bl a kommer förmågan att kunna beskriva händelser som är kopplade till konceptuell modellering att blir mer vanligt och detta kommer att ställa nya krav på verktygen. Det intressanta är att eftersom vi kommer att få mer kraftfulla hanteringsfunktioner i verktygen, kan olika "utvecklare" använda beskrivningsuttryck som annars betraktades vara lite svåra att utnyttja. T ex kan det gälla möjligheten att beskriva strukturen hos en händelse och koppla denna till objekt och attribut i en konceptuell modell. Teknologin kommer därför att underlätta att mer kraftfulla och avancerade modelleringsuttryck blir mer använda. Detta bidrar till att modellerna blir mer fullständiga och "sanna".

Detsamma gäller möjligheten att definiera och skapa system som bygger på objektorienterade principer, vilket förfaller vara en intressant riktning för framtidens systemutveckling och -förvaltning att gå. Verktygen kommer allt mer att ha stöd för denna typ av modellering.

Likaså kommer det att i framtidens metoder finnas möjlighet att beskriva tidsaspekter av modeller på ett bättre sätt än idag. Det innebär att man bättre kommer att kunna beskriva NÄR ett tillstånd inträffar för t ex ett objekt samt hur länge sådana tillstånd kan finnas. Tillstånd kan bättre kopplas till händelser etc., och exakta beskrivningar kan göras.

Verktygen måste dock ha goda sätt att illustrera tillståndsförändringar hos objekt, detta eftersom dessa aspekter annars kan bli ganska abstrakta och svåra att uppfatta. För att klara detta måste man ha tillgång till goda "projektionsfaciliteter". Man vill se delar (aspekter), av en modell som man för tillfället är intresserad av, grafiskt eller icke-grafiskt. Flerfönsterteknik, ljud, goda 3-dimensionella grafer etc samt olika former av hanteringsfunktionalitet kan också vara "delinstrument" som kan användas för att bättre stödja hanteringen av sådana aspekter av en modell.

Evolution mer är ständig nyutveckling

Som vi tidigare konstaterat kommer allt bättre specifikationsorienterade verktyg att användas i större utsträckning för att definiera system med hjälp av noggranna specifikationer. Specifikationerna kommer sedan att utgöra godutgångspunkt för, eventuellt maskinell, realisering. Dessutom kommer förmodligen olika former av dictionaries och encyklopedier att byggas upp för att hålla fram beskrivningar på ett lättillgängligt sätt för olika syften. Många aspekter av en verksamhet kommer att finnas med i dessa beskrivningar.

Detta kommer att lägga grunden till större grad av återanvändning av beskrivningar som redan finns för att bedriva olika grader av nyutveckling/vidareutveckling. Eftersom man gärna vill kunna ta tillvara delar som redan finns utvecklade och beskrivna kommer utvecklingen mer att ligga mot vidareutveckling än mot nyutveckling.

Återanvändning kommer att omfatta beskrivningar av verksamheter, informationssystem samt informationslager.

För att bygga upp ett bättre återanvändningsförfarande än idag måste man se till att de modeller som skapas med verktyg för nyutveckling idag kommer in i verktyg som skall användas för framtida förvaltning.

Distribution/sammankoppling av hjälpmedel

Man kan utgå ifrån att behovet av att kommunicera modeller, delmodeller och specifikationer mellan arbetsstationer och andra maskiner i olika delar av en verksamhet kommer att öka. Modeller behöver återanvändas på andra ställen än där de skapats och man behöver kunna kommunicera modeller och få respons ifrån intressenter till exempelvis ett visst utvecklingsprojekt etc. Verktyg kommer att användas lokalt i olika delverksamheter utav en större helhet. Verktygen kommer dessutom att vara av olika typ och ibland komma ifrån olika leverantörer. Detta ställer flera krav. Ett är att man måste hitta och bygga olika typer av protokoll för kommunikation mellan verktygen.

Om dessa kommunikationsprotokoll skall kunna vara någorlunda generella så att de kan användas för olika modelleringstekniker (beskrivningar) och för olika syften måste de vara på den ganska hög 'semantisk' nivå eller på annat sätt vara flexibla. Detta kräver välunderbyggda protokoll med stor utbyggbarhet.

Vissa leverantörerna kommer troligen att tillhandha sådana protokoll. Dock kommer en hel del kunskap krävas för att använda dessa protokoll.

I moderna datornät behöver man också ta i anspråk tjänster på "lägre" nivåer än ovan beskivna för kommunikationsändamål. De nya verktygen kommer att "inifrån en session" kunna utnyttja olika kommunikationstjänster som finns som funktioner i de moderna operativsystemen och som är inriktade på kommunikation mellan maskiner. Detta för att kommunicera modeller till olika delar av verksamheten.

Det kan t ex innebära att ett CASE-verktyg på en arbetsstation har inbyggda funktioner som gör att verktyget automatiskt eller på begäran använder grundfunktioner i företagets datornät för att söka upp ett visst dictionary som är aktuellt för en viss del av verksamheten och uppdaterar vissa saker där som just blivit klara i en session gjord av en projektgrupp som utvecklar ett visst delinformationssystem.

Om CASE-verktyg och dictionary-verktyg blir mer använda i verksamheterna kommer detta med säkerhet att ställa ytterligare krav på kommunikation mellan maskiner i verksamheten. Detta ökar i sin tur intresset för olika former av nät och tjänster som kan erbjudas i dessa. De stora datortillverkan som t ex Digital och IBM arbetar med sådana nätkoncept.

Dessa kommunikationsmöjligheter blir ännu mer påtagliga när tillverkarna av verktyg i dessa inför inbyggda funktioner som direkt kan utnyttja tjänster i viktiga nät.

Samband mellan och ihopkoppling av 'begreppsvärldar

Om datorstöd skall användas för att beskriva förhållanden över större delar av en stor organisation kommer det att ställas helt andra krav än som finns realiserade idag.

Datorstöden måste kunna beskriva den komplexitet som finns i en stor organisation. Denna kan vara av olika slag. Dels kan man vilja hantera beskrivningar som gäller vid olika tidpunkter. T ex vill man hantera dels beskrivningar av den situation som gäller vid en viss tidpunkt, dels beskrivningar som gäller ett önskat läge och som man strävar emot. Det innebär att beskrivningen och dess olika uttryck behöver ha någon form av tidsstämpling på sig.

En annan form av komplexitet kan vara att det finns begreppsvärldar som gäller för en viss del av verksamheten och endast där. Det innebär att verktyget måste dels ha en uppfattning om vilka delar av en verksamhet som kan ha begreppsvärldar och dels för varje sådan del vad den aktuella begreppsvärlden består av. Det kan också vara så att begreppsvärldarna är "skiktade" så att vissa begreppsvärldar är "lokala" (enligt ovan) och vissa är sådana att de utnyttjas för kommunikation enligt meddelanden som behöver utväxlas mellan de olika lokalerna.

Dessa strukturer kan hanteras i mycket få av dagens verktyg.

För att klara denna typ av krav måste verktyget kunna klara att exempelvis varje begreppsvärld bildar en separat "modell" inom vilken vissa saker skall gälla, t ex att begreppsnamn måste vara unika.

Kontroller, analyser och slutsatser ifrån verktygen

Man kan också räkna med att olika typer av kontroller och analysfaciliter kommer att bli mer rikhaltiga för att i olika meningar hjälpa till att skapa goda modeller eller beskrivningar. Det finns en mängd olika typer av kontroller och analyser som kan vara tänkbara.

Några exempel:

- Får jag dra en flödespil ifrån en "extern" funktion till en annan "extern" funktion?
- Hur har jag bildat mina funktionssnamn? Finns det aktiva verb i namnen?
- Har jag olika objekt med nästan samma attributstruktur? Vad kan det tyda på?
- Har jag mer än 20 funktioner i verksamhetsområdesöversikten (nivå 0)? Är det en bra modell?
- I en ansamling på 30 rutiner finns det ingen delrutin. Vad kan det tyda på?

Ovan angivna kontrollfrågor och dessas ordning kan mot slutet indikera kontroller i en mer kvalitativ eller semantisk riktning. Den senare typen av kontroller kommer också att bli mer vanlig. Det innebär att olika TYPER av kvalitetsaspekter av en modell kommer att kunna hanteras.

Som en förlängning av avancerade kontroller kommer också att ligga möjligheten att kunna ge olika former av avancerad vägledning hur användaren skall bete sig för att korrigera modellen i olika avseenden. Det innebär att verktygen kan ge vägledning i olika former för att man skall kunna undvika ytterligare fel av samma typ.

Domänkunskap

Framtidens verktyg kommer också att kunna vara försedda med viss domänkunskap. Exempelvis kanske man försöker skapa en speciell verktygsversion lämplig att använda i t ex bilindustrin. Denna verktygsversion kan vara försedd med viss kunskap om denna bransch.

Denna kunskap kan komma till pass när man gör modeller kring den aktuella typen av verksamhet. Verktyget kommer således att ge vissa synpunkter på de modeller som skapas i enlighet med denna grundkunskap. Ett visst företag eller en viss koncern kan bygga ut denna grundkunskap med mer specifik kunskap eller regler som man vill skall gälla för de modeller och de specifikationer som skapas. Ett exempel är hur man skapar identifierare för viktiga objekt.

Kunskap om verktyg och deras tillämpning

Det mest kritiska när det gäller verktyg är dock inte vilka avancerade-funktioner de har, utan mer hur man kan koppla verktyg till ett kunnande i organisationen. Annars kommer troligen inte verktygen att varken spridas eller medföra att man får effekt av dem.

Svårigheten är mest att koppla ett verktyg med lämpliga egenskaper till referensramar och kultur i en organisation och till den dynamiska processen så att kunskapen bland människorna i organisationen hela tiden utvecklas.

4. ORGANISATORISKA ASPEKTER

4.1 Inledning

Avsnittet beskriver huvudproblemen i DA-verksamhet sett ur organisationsaspekt och analyserar vad som är de främsta orsakerna till misslyckande. Därefter presenteras och diskuteras organisationsformer som, rätt hanterade, kan bidra till framgångsrik DA-verksamhet.

Ett antal seriösa försök att sätta igång dataadministration har genomförts på företag och myndigheter. Tyvärr har en del försök tenderat att rinna ut i sanden och dataadministratören har övergått till andra sysslor än vad som en gång var tänkt. Ett fåtal försök har fungerat och anledningen till framgången beror både på organisationsformen och på sättet som dataadministration bedrivits.

Dataadministration har en viktig funktion i hela förändringskedjan från huvudverksamhet till förvaltning av dataregister. En svårighet är att etablera en fungerande formell organisation för DA-verksamheten.

Slutsatsen när det gäller organisation är att dataadministrationsfunktionen har sin naturliga hemvist inom AU-verksamhetens funktion. Den kräver i sin tur att AU-verksamheten har en organisatorisk placering där stabsrollen kan utövas och där det finns en operativ knytning så att funktionen får en stark ställning i förhållande till utvecklingsprojekten.

Problemen

Dataadministration brottas med ett antal problem, eller möter hot och möjligheter, när det gäller funktionens etablering och funktion. De, som i DA-projektet ansetts viktiga är följande:

1. Felaktig verksamhetsform
2. Mognad
3. Insäljning på rätt nivå
4. Stöd från "hjärtat", inte från "plånboken"

5. Uteblivna kortsiktiga effekter
6. Förståelse och satsning på hela kompetenskedjan
7. Ifrågasatt "gökunge"
8. Sättet som DA bedrivs på
9. Starka och svaga individer

När det gäller DA-verksamhetens inkoppling i utveckling och förvaltning angav DA-projektet följande problem/orsaker, utan hänsyn tagen till deras inbördes rangordning:

1. "Vattentäta skott" mellan DA och SU.
2. DA är inte såld uppåt/SU/verksamheten.
3. DA försöker hitta sin roll.
4. SU har inte ställt krav på DA.
5. Datamodellerna ofullständiga för SU.
6. Skapa databaser är ett traditions- och teknikbundet arbete.
7. Det finns ingen gemensam metod för DA och SU.
8. Verktyg saknas (ex ritverktyg, aktiva datakataloger osv).
9. DBA-funktioner (DataBasAdministration) drunknar bland övrig metod- och teknik stöd.
10. Kulturskillnad mellan DA och DBA. DA är verksamhetsinriktad och DBA är teknikinriktad.
11. Helhetssyn saknas över de möjligheter till informationsåtervinning som finns.
12. Ansvar för våra databaser är spritt till personal med varierande kunskap om DBA-området.
13. Externa programvaruleverantörer går förbi DA och SU.

Bland ovanstående punkter har fem huvudorsaker hämtats och av dessa ger främst de tre första en grund för den följande diskussionen om DA:s uppgift i projekt och organisation av DA-arbetet.

1. DA har inte tillräckligt stöd.
 - a) SU ställer inte krav på DA
 - b) I/C arbetar enbart produktinriktat
 - c) DBA arbetar enbart teknikorienterat
 - d) DA hittar inte sin roll.

2. DBA har inte tillräckligt stöd.

- a) DBA ges inte möjlighet att få tillräcklig DBA-kompetens
- b) Metoder saknas

3. SU, DA och I/C samarbetar inte.

- a) Slutanvändarna föredrar en funktion som ger snabbt resultat
- b) I/C saknar helhetssyn

4. Gemensam metod används inte.

- a) Kulturskillnad mellan DA och SU

5. Verktyg används inte.

- a) Ritverktyg saknas
- b) Aktiva datakataloger
- c) Dagens datakataloger används sparsamt

4.2 ORGANISATORISKA ASPEKTER

Problem

Det finns tre problemområden som är förknippade med att starta upp en dataadministration. Problemen härrör från de organisatoriska aspekterna i så måtto att de har sitt ursprung i bristande förståelse och förutsättningar för dataadministrationen. Genom att ge dataadministrationen rätt organisatoriska förutsättningar kan många problem elimineras.

Bristande samspel med övriga funktioner

Ett område handlar om att få funktionen dataadministration att fungera i samspel med övriga funktioner. Det gäller samspelet med verksamhetsansvariga i systemutvecklingsprojekt och med databasadministratörer samt Information Center (IC).

Verksamhetsansvariga har inte tillräcklig förståelse för dataadministratörens uppgifter. Ofta vill man se snabba resultat, medan funktionen har en långsiktig uppgift som ger effekter på sikt. Därför ges inte tillräcklig auktoritet åt dataadministratören att driva igenom sina krav på helhetslösningar.

Dataadministratören har i många utvecklingsprojekt inte haft en naturlig roll. I regel saknas det etablerade former för hur projektet skall driva sina krav mot dataadministratören kontra hur dataadministratören skall driva igenom sina krav på vad projektet skall uppvisa.

Teknikorienteringen hos databasadministratörerna och verksamhetsorienteringen hos dataadministratören medför kulturskillnader. Dataadministratören har svårt att svara upp mot de förväntningar som databasadministratören har i form av stöd för att skapa databaser.

IC-verksamheten är produktorienterad och ger verksamheten snabba resultat. Dataadministratören är verksamhetsorienterad och arbetar med en helhetssyn, där effekterna är långsiktiga. Olikheterna kan skapa samarbetsproblem och gör att dataadministratören hamnar i underläge gentemot den del av verksamheten som har näraliggande behov.

Dåliga förutsättningar

För att en ny funktion skall fungera måste det finnas etablerade rutiner och former i verksamheten. Ett vanligt problem är att det saknas ritverktyg och en aktiv datakatalog, samt metoder som knyter ihop dataadministrationens funktion med andra funktioner.

Det kan också finnas för många verktyg där dataadministratörens verktyg blir en ytterligare tyngd för databasadministratören.

En oklar ansvarsfördelning av register och databaser medför att dataadministratören har svårt att hitta rätt samarbetspartners.

Är styrningen av systemutvecklingsprojekt bristfällig får dataadministratören svårt att verka. Samordningen med övriga data och register blir lidande vilket leder till problem i ett senare skede.

Saknas utvecklade regler och riktlinjer i ADB-verksamheten kommer effekterna av dataadministrationen att utebli. Kan t ex inköpen av program ske fritt utan någon samordning med befintliga system leder det till att dataadministratörens arbete blir verkningslöst.

Brister vid etablering av dataadministrationen

Själva processen att skapa en ny dataadministrationsfunktion i en verksamhet kan uppvisa svagheter i form av bristande stöd, felaktig rekrytering och tillsättning och för höga förväntningar om snabba resultat etc.

Krav för etablering av dataadministration

Från ovanstående problem följer att en etablering av en dataadministratör i en verksamhet måste föregås av:

- att de organisatoriska förutsättningarna klargörs så att dataadministratören får en naturlig roll att verka i.
- att det finns en ordning och ett etablerat arbetssätt att utveckla och förvalta system, så att dataadministratörens verktyg och arbete blir en naturlig och integrerad del.
- att förväntningarna hålls på rimlig nivå och att introduktionen ger möjligheter att bygga upp relationer och former för ett samarbete med berörda funktioner.

4.2.1 ADB-VERKSAMHETENS ROLLER

Stabs- och servicerollen

ADB-verksamheten brukar traditionellt delas in i en stabs och service roll, vilket gör att den verkar under olika förutsättningar.

I stabsrollen skall ADB-verksamheten ha ett långsiktigt perspektiv. Den skall arbeta med hela företagets verksamhet och se till att den görs effektivt. Stabsaktiviteter sker därför på uppdrag av ledningen. Lokala enheters intressen kan då få stå tillbaka för att företaget som helhet skall bli effektivt. Därför är formerna och spelreglerna kring stabsverksamheten ytterst kritisk för att de skall få genomslag i verksamheten.

I servicerollen skall ADB-verksamheten tillgodose kundens eller den lokala enhetens behov. I servicerollen är ADB-verksamheten en stödresurs med specialister där de lokala enheterna prioriterar och beslutar om vilka ADB-resurser de vill köpa. På så sätt är det de lokala enheterna som styr den kompetensmässiga utvecklingen och dimensioneringen av ADB-resurser. Serviceaktiviteter sker inom de regler och riktlinjer som staben utfärdar för att effektivisera helheten.

De ekonomiska spelreglerna skiljer sig åt mellan stab och serviceaktiviteter. Staben är ledningens förlängda arm. Den fattar inga beslut, utan bereder ärenden för ledningens beslutsfattande. Stabsaktiviteter finansieras därför från ledningen, eftersom lokala enheter inte kan betala för förändringar där de själva måste få stå tillbaka för att göra helheten effektiv.

I servicerollen är ADB-verksamheten marknadsstyrd. Kunden skall vara beredd att betala för tjänsterna som erbjuds. Graden av marknadsstyrning brukar variera beroende på ADB-verksamhetens totala struktur. I regel skall verksamheten ge ett nollresultat och tjänsterna skall inte belastas med annat än sådant som är direkt kopplad till utförandet.

Det är alltså en avsevärd skillnad mellan ADB-verksamhetens stabs- och serviceroll, då de har olika uppgift, perspektiv, uppdragsgivare, styrning och spelregler. Aktiviteterna kräver därmed olika förutsättningar för att kunna fungera.

Den integrerade och den fristående strukturen

ADB-verksamheter inordnas i en struktur. I princip finns två ytterligheter. Dels den där ADB-verksamheten är integrerad i linjen. Den ingår då som en avdelning i en lokal enhets verksamhet. I ett företag finns då flera ADB-avdelningar. Dels då ADB-verksamheten samlas i en fristående enhet som har ett resultatansvar.

I det förra fallet styrs ADB-verksamheten direkt från den lokala ledningen. I det senare fallet är ADB-verksamheten mer marknadsstyrd med en mer markerad kund - leverantörssituation.

Det finns flera drivkrafter bakom framväxten av strukturerna. Storleken driver fram en mer fristående ADB-verksamhet. När den nått en viss volym separeras den och görs mer självständig för att sedan återigen integreras i linjen. Är ADB en viktig komponent i verksamhetens affärer bör den integreras, är den inte kritisk och främmande kan den separeras. ADB-teknologin som både blir allt mindre och allt mer lättillgänglig driver fram lokala lösningar.

Båda strukturerna har emellertid svårt att hantera ADB-verksamhetens stabsroll. En fristående ADB-verksamhet får inte tillräcklig kraft och kontroll över lokala behov, eftersom det är kunderna som styr. I den integrerade formen saknas helt enkelt en funktion som har överblick.

Verksamheter med stort behov av en stabsroll måste därför söka strukturer som ger förutsättningar för en stab att verka. I dessa strukturer måste ADB-verksamheten vara neutralt placerad för att kunna verka mot lokala enheter. Den måste ha tillgång till ledningen och det måste finnas organ för att förankra och föra ut ledningsbesluten.

4.2.2 DATAADMINISTRATIONENS SYFTE OCH UPPGIFT

Dataadministratörens arbetsuppgift har både stabs- och serviceaktiviteter.

Stabsuppgift

Till stabsuppgifterna hör främst att utveckla regler och riktlinjer kring informationen, samt att granska och göra uppföljningar att dessa följs. Det gäller både i löpande verksamhet och i utvecklingsprojekt. Följande uppgifter är aktuella:

- Att strukturera data- och informationssystem, så att gränssnitten blir klara och att sambandsinformationen blir enhetlig.
- Att kontrollera data och datamodeller i systemutvecklingsprojekten. I denna arbetsuppgift skall dataadministratören kontrollera att verksamhetens regler och riktlinjer för data följs. T ex skall strukturen av data stämmas av mot, framtida uppgifter, sido- och överordnade verksamhetsdelar. I kontrollen ingår även att se till att begreppen används och är dokumenterad på ett korrekt.
- Att se till att det finns uttalade ansvar för aktuella data och register samt att säkerheten och behörigheter att hantera data följs.
- Att bereda beslut om förändringar i begrepp och strukturfrågor.
- Inom stabsuppgiften finns också en kompetensfunktion, att kunna och hålla ordning på verksamhetens begrepp för verksamheten som helhet.

Serviceuppgifter

Dataadministratören är också en specialistresurs som bör kunna göra analyser bl a med hjälp av dataflöden och datamodeller. I serviceuppgifterna ingår utbildning och stöd både till användare och i projekt. Det kan t ex gälla att ta fram funktionsanalyser, datamodeller med existenskrav, volymer och frekvenser, utföra dokumentation av dessa och göra stabilitetsanalyser.

I genomförandefasen handlar det mer om att assistera vid detaljutformningen, dokumentationen, realiseringen och vid test av register och databaser.

Dataadministration i den integrerade strukturen

En dataadministratör som fungerar lokalt kan ge ett lokalt kompetensstöd i utvecklingsprojekt, men har begränsade möjligheter att fungera i en stabsroll eftersom utgångspunkten är lokal. Däremot kan dataadministratören fungera i en lokal stabsroll, eftersom det finns en närhet till ledningen. Potentialen är emellertid mer begränsad. Dessutom torde de problem som tidigare poängterades uppträda.

Man kommer i den lokala enheten att blanda samman dataadministratörens service- och stabsroll. Det leder till oklarheter kring vad som förväntas av dataadministratören och brister i samspelet med övriga funktioner uppstår. Eftersom man arbetar med ett kortare tidsperspektiv längre ned i organisationen, ställs förväntningarna på kortsiktiga effekter. Dataadministratörens uppgifter syftar främst att ge långsiktiga effekter, vilket leder till att uppgifterna och nyttan kommer att ifrågasättas.

Dataadministrationen i den fristående strukturen

I den fristående strukturen kan dataadministrationen fungera som en servicefunktion, men har förutsättningar att fungera i en stabsroll. T ex har dataadministratören inte tillgång till ledningsbeslut och auktoritet att granska att regler och riktlinjer följs. Söker dataadministratören ta på sig sin stabsroll i denna situation kommer personen att finna sig i en frustrerad situation. Dataadministratören ser de klassiska problemen som han/hon är skickad för att lösa, men arbetet kommer att få litet eller inget genomslag. De organisatoriska förutsättningarna saknas.

Däremot kan dataadministratören fullgöra sina arbetsuppgifter i en serviceroll. Problemet kommer att vara att motivera behovet av funktionen för kunden, eftersom denna struktur är resultatorienterad och kommer då att verka relativt kortsiktigt för att kunna visa resultat.

4.2.3 EN FUNGERANDE STABSFUNKTION

I båda strukturerna uppstår problem med att få dataadministrationens stabsroll att fungera. Det är i stabsrollen som dataadministrationen har den största potentialen och det är utifrån den som de organisatoriska förutsättningarna skall beaktas. I servicerollen kan dataadministrationskompetens alltid köpas, vare sig den finns internt eller externt i en fristående ADB-enhet eller på konsultmarknaden.

Riktlinjer för beslutsfattande och arbete

En dataadministratörs främsta uppgift ligger i stabsarbetet. Det innebär att AU-funktionen behöver en ledning som fattar övergripande beslut av samordnande karaktär. Dessa beslut kan vara i form av regler och riktlinjer som sedan skall efterlevas på lokal nivå och i projekten. Dessa beslut behöver ofta förankras lokalt. Därför behövs i regel ett organ av lokala representanter i syfte att nå samförståndslösningar kring de riktlinjer och regler som skall gälla.

På lokal nivå finns ett ansvar att följa de av ledningen fattade besluten. Dataadministratören skall i sin stabsroll kunna tillhandahålla kompetens och information för de lokala enheterna. I stabsrollen ingår också att följa upp och granska att beslutade regler och riktlinjer efterlevs.

Därför måste dataadministratören i stabsrollen placeras neutralt i förhållande till alla lokala enheter. Staben måste också ha en position så att den kan rapportera till ledningen.

Det är inom ledningens ramar och med ledningens beslut som dataadministratören arbetar. Detta arbete skall då också finansieras av och rapporteras till ledningen. En finansiering via lokala enheter medför att dataadministratören får svårt att driva impopulära beslut.

Operativ knytning

Resurs och styrningsmässigt måste dataadministratörens stabs- och serviceaktiviteter skiljas åt. Organisatoriskt kan de hållas ihop, förutsatt att AU-verksamheten har en neutral organisatorisk placering. Syftet är att skapa en tillräckligt stor kritisk massa som möjliggör kompetensutveckling och reducerar sårbarheten.

I projekten är det emellertid viktigt att dataadministratören har auktoritet och befogenheter att gå in och granska projektaktiviteter så att de följer gällande regler och riktlinjer. Denna rapportering skall ske direkt till styrgrupp och sammanfattas till ledningen.

Genom att ha en operativ knytning till dataadministratörens stabsroll elimineras de klassiska linje-stabskonflikterna, samtidigt som dataadministratören får en närhet till projekten och serviceuppgifterna. Då serviceaktiviteter bedrivs inom samma organisation ges möjligheter för dataadministratören att föra ut sin kompetens och sitt synsätt i organisationen, vilket underlättar styrningen. Syftet är ju att dataadministratörens synsätt skall genomsyra såväl AU- som linjeverksamhet.

4.3 DATAADMINISTRATÖRENS UPPGIFT I PROJEKT

4.3.1 Projekt kontra linjearbete

Dataadministratören är en linjebefattning vars huvudsakliga uppgift är att löpande ansvara för verksamhetens begrepp och datastruktur. Dataadministratörens roll i utvecklingsprojekt är beroende av hur man på övergripande nivå fördelar ansvaret mellan projekt och linjefunktion.

Tidigare arbetade utvecklingsprojekt på jungfrulig mark. Då ingick allt i projektet. Det ansvarade för införskaffande av utrustning, skapande av nya register och funktioner. På sikt finns en tendens att begränsa projekten och låta mer och mer linjeansvar/aktiviteter ta över.

Det beror på att det redan finns en uppbyggd teknologi och en befintlig utrustning, att existerande begrepp och register samt funktioner eller sk "business rules" går att återanvända. Därmed reduceras utvecklingsprojektens storlek och ansvarsområde. Det blir således projektets ansvar att köpa in linjekompetens och att följa de regler och riktlinjer som är kopplade till respektive linjefunktions ansvarsområde.

Detta innebär att funktionen skall ha auktoritet över projektet inom det dataadministrativa ansvarsområdet, dvs ha vetorätt om ett projekt t ex använder ett begrepp med felaktig definition. Inom funktionsansvaret skall dataadministratören tillhandahålla och överföra kompetens om verksamhetens begrepp tvärs igenom olika verksamhetsområden, projekt och systemområden. Inom funktionsansvaret skall det finnas specialistkompetens, som kan köpas in i projekt. Den fungerar då i en service-roll med interndebiteringsprinciper. Funktionen har också en kontrollerande uppgift som är kopplad till stabsrollen. Stabsuppgifter finansieras med medel från ledningen.

4.3.2 Befattningar

Med dataadministrationens funktion menar jag hela kompetenskedjan. Ansvar sträcker sig från verksamhetens begrepp till realisering och trimning av register. Befattningshavare som berörs är **dataadministratörer, datakonstruktörer, information center** (för söksystem och gränssnitt) och **databasadministratörer**.

Dataadministratören är en ansvarig stabsfunktion under utvecklingsprojektet. Befattningen har rätt att delegera stabsuppgifter till övriga befattningshavare. Dessa rapporterar då till dataadministratören. Befattningen har också servicefunktioner främst i de inledande skedena i utvecklingsarbetet.

Datakonstruktören arbetar främst på uppdrag av projektet under konstruktionsfasen, vid specificering och utformning av logiska register. IC-verksamheten kommer i framtiden att få en delvis ny roll och ansvara för söksystem och gränssnitt. Även denna funktion kommer främst att arbeta i konstruktionsfasen. Databasadministratören arbetar huvudsakligen med fysiska register i genomförandefasen då det gäller att realisera, testa och trimma register.

Ovanstående befattningar kan också arbeta med stabsuppgifter på uppdrag av dataadministratören.

Projekt- och även förvaltningsarbetet är traditionellt systemindelad. Dataadministrationens uppgift är emellertid funktionsindeldad. Den kommer därmed att skära tvärs över projekt- och förvaltningsområden och den kommer att arbeta med olika typer av system.

Om vi utgår från utvecklingsarbetets faser kommer dataadministrationens funktion både att ha en stabs- och en serviceuppgift.

Dataadministratörens uppgift i förstudien

I förstudien handlar det främst om att se till att de identifierade problemen som berör dataadministrationens ansvarsområde fångas upp. Om de inte blir prioriterade utifrån givna direktiv bör de värderas och ges en prioritet utifrån dataadministratörens funktion.

Vidare finns det både en stabs- och servicefunktion vid data- och funktionsmodellering både i nulägesbeskrivningen och vid utformningen av lösningsförslag, samt fastställande av behörigheter och hantering av data/information t ex med hjälp av ansvarsmatriser. Stabsfunktionen kontrollerar att riktlinjer och kvalitetskrav följs. Servicefunktionen är en specialistfunktion som kanske inte finns i projektet, utan måste köpas in. Vidare har dataadministratören kanske främst en serviceroll i sin medverkan vid framtagningen av teknisk lösning.

Datakonstruktörens uppgift i projekteringen

I projekteringen arbetar datakonstruktören i en serviceroll. Vidare är dataadministratörens stabsroll central eftersom funktionen främst skall se till att projektets aktiviteter överensstämmer med de riktlinjer som gäller för verksamheten. Uppgiften blir att granska projektplaner och följa upp genomförda aktiviteter.

Det gäller att se till att definitioner av begrepp och dokumentation av dessa är korrekt och att framtagna datastrukturer överensstämmer med framtida uppgifter och är anpassade till angränsade verksamheter. Saknas denna kompetens i projektet kan datakonstruktören ingå som en serviceroll och ansvara för att uppgifterna blir utförda.

Vid utformning av lösningen kan dataadministratören dels utforma krav för data. Det gäller, volymer, aktualitet, tillgänglighet, säkerhet i form av back-up, återstart etc, samt sekretess och behörigheter. Dels medverka vid utformningen av såväl manuella som maskinella rutiner vid användning av data.

Dataadministratören kan också bidra med att utvärdera standardpaket, eftersom datamodellen kompletterad med ovanstående krav utgör viktiga utvärderingskriterier. Det är ju lättare att komplettera ett standardpaket med funktioner än att ändra/komplettera begreppen i det.

I samband med att organisationen skall anpassas kan dataadministratören i en renodlad serviceroll hjälpa till med att revidera lösningen med tanke på de organisatoriska aspekterna.

IC-verksamheten bör på sikt utveckla specialistkunskap om gränssnitt och söksystem som utnyttjas vid utformning av den specifika arbetssituationen i samband med utformningen av gränssnitt mellan människa och maskin. Däremot har funktionen en mer renodlad serviceroll vid en eventuell framtagning av prototyper och för att göra sökvägsanalyser för lagrad data.

I dimensioneringen av teknik har funktionen ansvar för fysiska register och för att värdera vilka databashanteringssystem som tillgodoser lösningens behov. Här kommer alltså stabsrollen in i form av granskare, medan det är projektets uppgift att se till att aktiviteten genomförs, eventuellt med hjälp av en datakonstruktör.

Vid värdering av lösning har dataadministratören en stabsroll som kontrollerar att värderingen blir objektiv.

Datakonstruktörens och databasadministratörens uppgift i genomförandet

I genomförandet blir dataadministratörens stabsroll än mer förstärkt. Där skall befattningen dels utveckla standarder, regler och riktlinjer för lösningen, dels skall funktionen se till att lösningsspecifikationer uppfyller utvärderingskriterierna för lösningen.

I servicerollen engageras datakonstruktören i de tidiga aktiviteterna, vid detaljutformning, konstruktion och dokumentation av register i datakataloger. Det gäller även valet av den slutliga databashanteraren, om funktionen används som rådgivare i tidigare aktiviteter. IC kan medverka vid utformningen av program och konstruktion av gränssnitt.

IC-funktionen kan även medverka vid anskaffning av den teknik som krävs om inte detta sköts av den teknikansvarige eller inköpsfunktionen. Däremot är granskning av installation och test av den teknik som krävs vid utveckling, utbildning och produktion en viktig stabsfunktion.

I de senare aktiviteterna i genomförandefasen, vid test och införande av lösning får databasadministratören en tung serviceroll. Det gäller även vid realisering av registrern och styrrutinerna för dessa. Då skall också tidigare fastställda säkerhetskrav och behörighetsrutiner realiserars. Vidare skall ansvaret för data blir tydligt och dokumenteras.

Då lösningen integreras skall funktionen hjälpa till att skapa och ladda register i utvecklingsmiljön och utbildningsmiljön, testa styrrutiner och ta fram system- och driftdokumentation för register och berörda styrrutiner för omorganisation, back-up och återstart. Funktionen kommer också att delta i integrationstesten.

Funktionen har en stabsroll vad gäller uppläggning och genomförande av utbildning av användare och förvaltare av lösningen.

Vid själva införandet av lösningen finns ett övergripande ansvar för registren då dessa skapas, laddas, produktionstestas och trimmas i produktionsmiljön. Här ingår också konvertering och avveckling av gamla register. Vidare skall funktionen se till att behörigheter delas ut.

I samband med projektavslut har dataadministrationen en stabsuppgift att kontrollera registren med avseende på förvaltning, uppgjorda standarder, regler, riktlinjer och behörigheter för användare och driftpersonal. Funktionen skall också utveckla nya riktlinjer för drift och förvaltning av registren.

Dataadministratörens uppgift i förvaltningsfasen

Förvaltningsfasen är liksom projekten traditionellt kopplad till ett specifikt system eller systemområde. Dataadministratörens uppgifter i förvaltningsfasen är dels att i servicerollen ge stöd och hjälpa till vid ändringar av register, dels att sköta uppgifter av mer generell natur som både har inslag av stabsroll och av ett traditionellt linjearbete, då det gäller att vidmakthålla en anpassad data- och lagringsstruktur för verksamheten. Det gäller tvärs igenom alla system.

I förvaltningsfasen har dataadministrationen en funktion att vidmakthålla kvaliteten i befintliga register, tillhörande dokumentation i datamodeller och datakataloger samt datortekniska uppgifter såsom maskinnamn, skivstation etc, utvecklings- och utbildningsmiljö.

I linjeansvaret ingår ett övergripande ansvar för datastrukturen och för att utveckla standarder och regler för:

- verksamhetsbeskrivande data och dess behandling
- konstruktion av ADB-register
- användning av databashanterare
- IC-register

Dataadministrationen har ett samordningsansvar över systemgränserna för dataflödet. De bereder ändringar av behörigheter, begrepp etc som systemägare initierar för att värdera behov och eventuella åtgärder som påverkar över systemägarens ansvarsområde.

4.4 DATAADMINISTRATÖRENS ORGANISATORISKA PLACERING

Det är inte säkert att den organisatoriska placeringen löser dataadministratörens problem. Den ger endast förutsättningar för dataadministratören att verka. Däremot kan den organisatoriska placeringen förhindra en dataadministratör att fungera. T ex krävs en viss placering för att dataadministratören skall kunna ta ett funktionellt ansvar. Vidare kräver stabsrollen vissa organisatoriska förutsättningar.

Dataadministratören i linjen

Dataadministratören är placerad i en samordnande stabsroll i linjen medan IC och databasadministratören är placerad i ADB-avdelningen.

För att dataadministratören (DA) skall fungera krävs att linjeföraren har förståelse för DA:s uppgift och kan ge DA tillräcklig auktoritet gentemot egna linjeförare och systemerare. En DA:s stabsroll kan i denna lösning aldrig sträcka sig längre än till den egna organisatoriska enheten. Vidare förutsätter lösningen att DA har auktoritet mot ADB-avdelningen, som kanske ligger under en annan organisatorisk enhet.

Problemet med denna lösning är att DA blir en ensamvarg inom linjeorganisationen som får svårt att verka i en stabsroll. Istället finns risk för att linjen utnyttjar DA-kompetens för annat stöd inom IC-verksamheten. Spänningar kan uppstå mellan DA och ADB-enheten som vill driva sin linje.

Fungerande lösningar finns där AU-chefen tillika har tagit på sig rollen som DA. Funktionen är placerad i en stab för ledningen medan verksamheten har sin ADB-funktion organiserad som en fristående dataenhet. AU-chefens dataadministrativa uppgifter är främst av stabskaraktär, medan det funktionella och operativa arbetet utförs inom dataenheten.

Dataadministratören som stab i ADB-enheten

Dataadministratören är placerad som en samordnande stabsroll i ADB-avdelningen, medan IC och databasadministratören är placerad under applikationsansvarig i samma avdelning.

Detta är kanske den naturliga lösningen där placeringen gör det möjligt för DA att fungera i en stabsroll för hela verksamheten. Det förutsätter att ADB-enheten är placerad neutralt inom aktuell verksamhet och att den kan verka över hela verksamheten. Är ADB-enheten placerad i en separat organisation, t ex i ett eget bolag har den svårt att fungera i en stabsroll, eftersom den då kommer för långt från verksamheten och den har också en egen ledning.

I den organisatoriska placeringen finns det inget som hindrar att dataadministratören har ett funktionellt ansvar. Alternativt skulle funktionen likaväl kunna placeras i linjen och då få ett delegerat stabsansvar av AU-chefen. Oberoende av om dataadministratören är placerad som en renodlad stabsfunktion eller som en linjefunktion kan servicerollen fullgöras i form av interndebiteringsprinciper.

Databasadministratören och IC får huvudsakligen en service-roll vare sig de är självständiga linjefunktioner i ADB-avdelningen eller om de är placerade under en systemchef. DA har delegeringsrätt vad gäller stabsärenden som berör databasadministratören och IC-området.

Hela funktionen som en stab

Dataadministratören är en stabsfunktion under AU-chefen medan datakonstruktören och databasadministratören är staber under ADB-chefen i samma organisatoriska enhet.

Det är en ytterlighet som särskiljer dataadministrationsfunktionen från andra utvecklingsfunktioner. Därmed kommer placeringen att framhåva stabsfunktionen, vilket medför att den kan hamna vid sidan om den operativa verksamheten. Funktionen kan då få problem med service-rollen. Den blir inte engagerad i utvecklingsprojekten, utan kompetens byggs upp inom systemgrupperna. Risken är alltså att funktionen avskärmas. Därför kan lösningen ifrågasättas.

Harmoni med verksamhetens organisation

Dataadministrationsfunktionen har sin naturliga hemvist inom AU-verksamhetens funktion. Den kräver i sin tur att AU-verksamheten har en organisatorisk placering där både stabsrollen kan utövas och där det finns en operativ knytning så att funktionen får en stark ställning i förhållande till utvecklingsprojekten.

Funktionen bör om möjligt vara relativt sammanhållen. Den interna samordningen underlättas om berörda befattningshavare arbetar inom samma organisation och under lika förutsättningar. Funktionens organisatoriska placering skall harmoniera med verksamheten i övrigt. Är verksamheten utspridd och ADB-verksamheten integrerad i linjen måste också dataadministrationen beakta detta. Samordningen sker då genom att man skapar separata samordningsorgan för funktionen.

På så sätt blir organisationen för den enskilda funktionen något relativt, där man måste beakta verksamhetens organisation och geografiska spridning. Det är under dessa betingelser som förutsättningarna för dataadministrationen kan skapas. En viktig förutsättning finns i dataadministrationens linjefunktion i förhållande till utvecklingsprojekt. En annan handlar om att överföra kompetens om verksamhetens begrepp mellan olika verksamhetsområden. En ytterligare förutsättning är de organisatoriska förutsättningarna för stabsrollen.

5. Bemanning och kompetensprofiler/ kompetensbeskrivning

Det kompetenskrav som måste ställas på en DA-funktion/-funktionär kan formuleras på ett antal sätt med fokusering på olika egenskaper. Kompetenskraven beror bland annat på verksamhetens art, storlek, informationsberoende och "datamognad".

Vilka egenskaper hos en DA-funktion som kan vara aktuella att bedömma/värdera är naturligt nog - bortsett från vissa "givna" - inte särskilt lätta att hitta/definiera, inte minst beroende på att de verksamhetsbeskrivande faktorer som nämns ovan sinsemellan är (kan vara) "motverkande". Som exempel på detta kan nämnas att ett mycket högt informationsberoende inom en verksamhet, vilket säkerligen kräver "sin speciella profil" på en tänkt DA-funktion, inte nödvändigtvis betyder att datamognaden är särskilt hög i denna verksamhet, vilket i sin tur kanske ställer helt andra krav på DA.

Någon entydig relation mellan "behov och profil" i form av ett slags facit kan med andra ord inte anses vara möjligt att fastställa. Dock kan man, vilket vi ansett vara det näst bästa uppslaget, kartlägga och sammanställa en generell profil, som skulle kunna karaktärisera den i alla tänkbara sammanhang kompetenta DA-funktionen. Utifrån denna generella kompetens står det sedan var och en fritt att "själv komponera" den DA-funktion som passar verksamhetens behov.

Vi har sålunda sammanställt följande egenskaper som vi anser lämpliga att inneha i rollen som DA-funktion/-funktionär. Prioriteringsordningen är antydd men förmodligen inte "hållbar" i alla lägen;

- a ADB-bakgrund
- b Säljförmåga
- c Analytisk förmåga - kunna se helheter
- d Kännedom om/känsla för - "ADB från användarhåll"
- e Initiativförmåga

- f AU-inriktning i arbets- och tankesätt
- g Utbildningsintresse
- h Nyfiken på/intresserad av "vad som händer i omvärlden"
- i Verksamhetskunskap/-kännedom
- a ADB-bakgrund

Man kan höra och läsa uttryck som "DA skall vara helt fristående ifrån datorer och ADB-system, att DA inte på något sätt skall skilja på stöd till manuella respektive maskinella system" etc. Detta är naturligtvis rätt såtillvida att DA-insatser för manuella rutiner inte får glömmas bort. Men trots allt så måste DA-funktionens huvudsakliga inriktning anses vara att "stödja och påverka utveckling och förvaltning av maskinella informationssystem" med syftet att erhålla effektivare och bättre ekonomiskt utnyttjande av information.

Funktionellt sett är DA-funktionens "logiska arbetsbeskrivning" relativt flytande med avseende på gränsen eller gränzonen mot den renodlade tekniska hanteringen av data (i form av laddning, trimning etc av databaser). Även den tekniska hanteringen av databaser kan ju benämnas "administration av data". Hur denna zon skissas - om den nu överhuvudtaget behöver nämnas i den specifika organisationen - kan och bör vara beroende av den kompetens som finns tillgänglig samt hur denna är fördelad inom organisationen.

Detta resonemang leder till att kravet på ADB-bakgrund - hellre bred och lite grundare än, smal och djup - måste klassas som ett mycket väsentligt inslag i DA-profilen. Erfarenheter från drift, programutveckling, systemering och förvaltning av informationssystem hamnar således högt upp på kravlistan.

b) Säljförmåga

En egenskap som absolut måste vara väl tillgodosedd hos den DA-ansvarige är säljförmåga. Detta gäller inte minst lanseringen av nyheter, nya produkter, koncept och metoder.

Vad som kanske sätter extra press på säljegenskaper i samband med att vi diskuterar DA-funktionen är, att det så till övervägande del handlar om idéförsäljning - kanske till beslutsfattare/makthavande som

- inte (till en början) kanske ser värdet av DA
- inte tycker att man har några bekymmer som skulle kunna lösas med hjälp av insatser från en DA-funktion
- tycker att det mesta fungerar bra - man har ju tillgång till ADB-resurser och kompetens, det måste väl vara tillräckligt!

De flesta av DA-funktionens arbetsuppgifter kan härledas ur företagets informationsstrategi och/eller tillhörande målformuleringar. En av DA-funktionens viktigaste arbetsuppgifter, vilken direkt kan härledas ur denna informationsstrategi, är att verka för denna strategis tillämpning och fortlevnad. I detta ligger en stor portion försäljningsarbete.

c) Analytisk förmåga - kunna se helheter

Det är viktigt att DA-funktionen/funktionären har förmåga att analysera problem och möjligheter, att han kan tolka utsagor, se var det finns behov av samordningsinsatser som kan finnas eller uppstå i en organisation. Detta innebär att DA måste inneha kompetens att genomföra analyser av utfall av olika slags åtgärder och insatser - egna såväl som andras.

Att se eller känna igen en helhetsbild, att kunna höja "siktet" såpass att helheten framträder, så att "vilselädande" detaljer förpassas till bakgrunden, är en erkänt svår konst, svårare ju mer komplex verksamheten är.

DA skall här kunna visa på och tillhandahålla metoder som stöder detta sätt att se på verksamheten.

d) Kännedom om och/eller känsla för - 'ADB från användarhåll'

Det är lätt att förbise användarens problem att förstå och uttrycka sig med den speciella ADB-vokabulären när man själv är väl hemmastadd i densamma. Dessutom är det sannolikt så att ju mer insatt man själv är, desto svårare är det att förstå den/dem som inte förstår.

Ett annat slags problem i rakt motsatt riktning är de svårigheter som ADB-utvecklare/tekniker har att förstå vad användaren/beställaren av t ex utvecklingstjänster, egentligen vill ha.

Dessa kommunikationsproblem har ställt till med mycket förtret över tiden och kommer så att fortsätta om inget görs åt problemen.

DA-funktionen, som kan betraktas som en länk mellan användaren och ADB-systemutvecklaren, speciellt beträffande ordbruk, begreppsanalys, begreppssamband etc, har här en viktig arbetsuppgift att ta hand om. Detta ställer bland annat krav på kunskaper som närmar sig det lingvistiska området, d v s goda kunskaper och kännedom om svenska språket både teoretiskt och i tal och skrift.

e) Initiativförmåga

En initiativtagare är sannolikt välkommen var han/hon är hamnar, om vi bortser ifrån förhållanden/verksamheter som styrs av den berömda "Jantelagen". Naturligtvis har även Janteorganisationen behov av initiativtagare, kanske i högre grad är andra. "List" är det vanliga botemedlet i dylika situationer.

När vi diskuterar initiativ i samband med DA är det närmast ur perspektivet att detta, dvs dataadministration, ofta anses vara ett redan inmutat och väl omhändertaget område. Många ADB-avdelningar anser sig (många med rätta!) ha en väl fungerande dataadministration. Bortsett ifrån att detta i de flesta fall är ett oriktigt påstående, är det i högsta grad diskutabelt om DA hör till/skall höra till ADB-avdelningen. Mera om detta senare.

Vad som här skall diskuteras avser närmast initiativ till t ex att lyfta frågan om "vikten av god data/informationsadministration" till ledningsnivå vilket är en nödvändighet av flera skäl. Det är en tung fråga som kan röna visst motstånd och behöva mycket stöd!