

SISU analys

Nr. 3

GRAFISKT BASERADE
DATORSTÖD
FÖR SYSTEMBESKRIVNING

SISU – Svenska Institutet för Systemutveckling

Box 515
182 15 Danderyd
08 - 755 29 30

Norra Krokslättsgratan 2
412 64 Göteborg
031 - 83 02 50

GRAFISKT BASERADE DATORSTÖD FÖR SYSTEMBESKRIVNING

ISSN: 0282-9924

Copyright
SISU – Svenska Institutet för Systemutveckling
Februari 1986

INNEHÅLLSFÖRTECKNING

| | |
|--|----|
| FÖRORD | 6 |
| SAMMANFATTNING | 8 |
| 1. GRAFISKT BASERADE DATORSTÖD FÖR SYSTEMBESKRIVNING -- EN INTRODUKTION | 10 |
| 1.1 Typer av datorstöd | 10 |
| 1.2 Metodik | 12 |
| 1.3 Effekter på systemutvecklingsproce- sen och för förvaltning av system | 13 |
| 1.4 Olika användningssituationer för modelleringsstöd | 14 |
| 2. PRESENTATION AV NÅGRA EXISTERANDE DATORSTÖD | 15 |
| 2.1 DAMBI (Datorstöd för MBI) | 15 |
| 2.2 Data Designer | 16 |
| 2.3 DDEW (Database Design and Evalution Workbench) | 17 |
| 2.4 Design Manager | 18 |
| 2.5 EXCELERATOR | 19 |
| 2.6 GRAPHDOC | 21 |
| 2.7 JSP-TOOL | 21 |
| 2.8 LID (Living In a Database system) | 22 |
| 2.9 Sembase | 23 |
| 2.10 Tools for Supporting Structured Analysis | 24 |
| 2.11 USE (User Software Engineering) | 25 |
| 3. BESKRIVNING AV RAMATIC | 27 |
| 3.1 Bakgrund till utvecklingen av RAMATIC | 27 |
| 3.2 Några intentioner för RAMATIC | 28 |
| 3.3 Yttre egenskaper | 28 |
| 3.4 Inre egenskaper | 34 |
| 3.5 Framtida egenskaper | 36 |
| 4. NÅGRA ERFARENHETER AV DATORSTÖD | 39 |
| 4.1 EXCELERATOR - en mångsidig verktygslåda | 39 |
| 4.2 Några erfarenheter av ett datorstöd för MBI | 42 |
| 5. TANKAR KRING DATORSTÖD | 43 |
| Inledning | 43 |
| Allmän reflektion över området | 44 |
| Ett synsätt på datorstödsstruktur | 44 |
| Sammanfattning | 46 |
| 6. KAN DATORSTÖD FÖRBÄTTRA VERKSAMHETSANALYSEN? .. | 47 |
| 6.1 Två paradoxer | 47 |
| 6.2 Systemteoretiska utgångspunkter | 47 |
| 6.3 Verksamhetsanalysens och datorstödet syften | 48 |
| 6.4 Vad kan förbättras med datorstödd verksamhetsanalys? | 48 |
| 6.5 Sammanfattning | 50 |

| | | |
|-----|---|----|
| 7. | KRAV PÅ MODELLERINGSSTÖDSSYSTEM | 52 |
| 7.1 | Introduktion | 52 |
| 7.2 | Problemspecifika krav | 52 |
| 7.3 | Generella krav på MSS | 53 |
| 7.4 | Hur kan dessa generella krav uppfyllas? | 54 |
| 8. | MER ATT LÄSA — REFERENSER | 56 |
| 8.1 | Mer att läsa | 56 |
| 8.2 | Referenser | 56 |
| | APPENDIX: UTKAST TILL EGENSKAPSRAM | 59 |

FÖRORD

Svenska institutet för Systemutveckling (SISU) ger ut två tidskrifter. Den ena av dessa, SISU informa, beskriver på ett populärt sätt vad som löpande händer i och omkring SISU och dess intressentföretag.

Den andra skriften, den av vilken Du nu har ett nummer framför Dig, kallas SISU-analys och behandlar varje gång ett tema -- ett aktuellt problemområde, som belyses litet djupare och ur olika aspekter. De två tidigare utkomna numren har behandlat konceptuell modellering (nr 85/1) och kontorsinformationssystem (nr 85/2).

Temat för detta nummer

Temat denna gång är "GRAFISKT BASERADE DATORSTÖD FÖR SYSTEMBESKRIVNING".

Det finns många metod- och modelleringsansatser inom systemutvecklingsområdet. I flera av dem, kanske t o m i merparten av dem, används, tillsammans med andra tekniker, beskrivningstekniker som är grafiskt orienterade. För att underlätta arbetet med att skapa och hantera graferna utvecklas allt oftare datorstöd med möjlighet till interaktiv grafisk editering, speciellt som tillräckligt kraftfull hårdvara, skärmar samt mjukvarukomponenter nu har blivit tillgängliga som möjliggör detta. Ibland använder man begreppet modelleringsstöd och avser då datorstöd, som stödjer modelleringsarbete i samband med systemutveckling på ett "djupare" sätt genom att bistå med relevanta och användbara analyser, kontroller och utsökningsmöjligheter.

Vi koncentrerar oss alltså på datorstöd som är grafiskt baserade, detta för att ämnesområdet skall bli någorlunda hanterbart med avseende på antalet datorstöd och ansatser som vi kan ta upp. Gränsen för vad som skall räknas till grafiskt baserade datorstöd är dock långt ifrån skarp. Många datorstöd har en grafisk del, men av mindre omfattning, beroende t ex på att de beskrivningsaspekter som man vill ta upp inom ramen för en modelleringsteknik inte alltid är särskilt lämpade att beskriva i grafisk form.

Typer av datorstöd som vi inte kommer att behandla är applikationsgeneratorer, experimentsystemverktyg, etc. Däremot kommer relationer, ihopkopplingsmöjligheter och samverkan med dessa typer av verktyg att belysas.

Två nummer av SISU-analys

Vi kommer successivt att ta fram mer material som rör aktuell grupp av datorstöd. Allteftersom arbetet med föreliggande nummer har fortskridit, har vi stött på fler och fler datorstödsprodukter och vår ambition är att återkomma i ett framtida nummer av SISU-analys med en mer systematisk och fullständig genomgång. Ett utkast till vilka egenskaper som kan vara intressanta och givande att titta på hos de olika datorstöden återfinns dock som ett appendix redan i detta nummer. Dessutom har, i kapitel 2, gjorts korta sammanfattningar av 11 st olika datorstödssatser som vi har träffat på.

Uppläggning

Uppläggningsen av föreliggande nummer är som följer. Kapitel 1 introducerar och ger en överblick över området. I kapitel 2 återfinns, som ovan nämnts, kortfattade presentationer av några datorstöd. Inom programområde 3 i SISU pågår utveckling av ett grafiskt baserat modelleringsstöd kallat RAMATIC. I kapitel 3 beskrivs detta utförligt.

Flera mycket läsvärda bidrag från datorstödserfarna och datorstödsintresserade personer inom SISU's medlemsföretag har flutit in. I avsnitt 4.1 redovisar Lars Axelsson, Kommundata, erfarenheter av användning av datorstödet Excelerator inom sitt företag och Christer Fall, Programator, sammanfattar, i avsnitt 4.2, några erfarenheter från användning av ett MacIntosh-baserat datorstöd för MBI. Anders Persson, Data Logic, reflekterar över området i sitt bidrag betitlat: Tankar kring datorstöd, vilket utgör kapitel 5, och Mats-Åke Hugoson, Programator, ställer, i kapitel 6, frågan: Kan datorstöd förbättra verksamhetsanalysen?

Ulf Sundin, SYSLAB, har bidragit med kapitel 7, som har rubriken: Krav på modelleringsstödsystem. I kapitel 8 återfinns ett "mer-att-läsa"-avsnitt och en referenslista. Det appendix, som nämnts ovan och som innehåller ett utkast till en egenskapsram för karaktärisering av datorstöd, avslutar numret.

Bidragsgivare

Det är enligt riktlinjerna för denna skrift att olika synsätt och behov kommer till tals genom att flera författare deltar med bidrag. Författarna kommer dels från intressentföretagen, dels från högskoleforskning och är dels mera fast anknutna medarbetare till SISU.

Vi vill från område 3 i SISU och som har ansvaret för detta nummer, rikta ett speciellt tack till följande personer för mycket värdefulla bidrag:

Lars Axelsson, Kommundata
Christer Fall, Programator
Mats-Åke Hugoson, Programator
Anders Persson, Data Logic
Ulf Sundin, SYSLAB

samt

Kristina Elestedt, Data Logic och SISU
Kerstin Karlsson, Programator och SISU

De båda sistnämnda arbetar en del av sin tid inom område 3 i SISU. Kristina Elestedt har medverkat i kapitel 2 och 3, och Kerstin Karlsson i kapitel 2.

I övrigt har Lars-Åke Johansson svarat för kapitel 1 och stora delar av kapitel 3, medan resterande avsnitt (kap 2, 3, 8 samt Appendix) i samarbete har framställts av Roland Dahl, Mats R Gustafsson, Lars-Åke Johansson och Håkan Torbjär, samtliga SISU.

SAMMANFATTNING

I den introduktion till grafiskt baserade datorstöd för systembeskrivning, som ges i kapitel 1, grupperas dessa verktyg i: 1) Ritorienterade datorstöd, 2) Datorstöd som ger möjlighet till analys och kontroll av modellstrukturer (modelleringsstöd), 3) Modelltypsoberoende verktyg, 4) Dictionary-liknande verktyg, 5) Verktyg som kan hantera dynamik, och 6) Verktyg som stöder prototyping. Varje grupp betonar en speciell egenskap och grupperna utesluter på intet sätt varandra. Det hävdas här att de flesta datorstöd, som vi hittills har sett, till stor del är ritorienterade, dvs avsedda att stödja skapande och manipulering av grafer, ofta enbart för en viss beskrivningsteknik. Man skapar en bild på en grafisk skärm och ofta sker t ex ingen kontroll av att man "ritar rätt" enligt de regler som gäller för tekniken. Detta gäller åtminstone för stöd som kommit till sedan datorgrafik blev mer tillgänglig att använda. Tidigare datorstöd, som inte använde sig av datorgrafik, hade ibland så svåränvänd interaktion att de inte blev använda av denna anledning.

Utvecklingen ser ut att gå mot datorstöd, som i högre utsträckning än nu kan fås att 'förstå' den skapade bilden på skärmen och lagra en beskrivning av denna struktur. På detta sätt kan datorn i ökad utsträckning bidra med analyser, kontroller och utsökningar i dessa strukturer. Utvecklingen går alltså från ritorienterade stöd mot modelleringsstöd. En annan mycket intressant utvecklingsriktning, som indikeras av grupp 5) ovan, är den att låta datorstöden "agera dynamiskt" t ex genom att olika dynamiska förlopp kan animeras.

I kapitel 1 tas, vidare, kopplingen mellan datorstöd och metod upp och de effekter som datorstödanvändning kan leda till som t ex att ge mer tid för metoden och dess intentioner och att högkvalitativa modeller åstadkoms. Effekter på systemutvecklingsprocessen och för förvaltning av system diskuteras, liksom ett antal olika användningssituationer för modelleringsstöd, t ex lokal användning i projektgrupp, som fleranvändarstöd, distribuerad användning, etc. Önskvärda egenskaper för datorstöd som anges är bl a att det bör vara möjligt att använda dem tillsammans med andra former av datorstöd för systemutveckling, som t ex datorstöd för projektstyrning, applikationsgeneratorer och olika typer av stöd för informationssystemförvaltning (Data Dictionary-system).

Ett antal datorstödsprodukter, av den typ vi begränsat oss till, finns nu på marknaden och flera andra befinner sig på utvecklings- och prototypstadiet. I kapitel 2 presenteras ett antal av dessa. Här återfinns såväl något mer bekanta datorstöd som Design Manager, GraphDoc, EXCELERATOR, JSP-TOOL m fl, som några i Sverige mindre välbekanta stöd som t ex Database Design and Evaluation Workbench (DDEW) och datorstöd för User Software Engineering (USE).

RAMATIC kallas det modelleringsstöd, som vi håller på att utveckla inom SISU, Göteborg. I kapitel 3 beskrivs detta modelleringsstöd, som det ser ut nu, och som det är tänkt att se ut i framtiden. Bakomliggande intentioner och krav på stödet tas fram, varav mycket kan relateras till det som behandlats i kapitel 1.

Kapitel 4 inrymmer två inslag, som båda redovisar erfarenheter från användning av datorstöd. Det första inslaget, som behandlar erfarenheter av datorstödet EXCELERATOR, är författat av Lars Axelsson, Kommundata. Här ges först bakgrunden till valet av EXCELERATOR och en komprimerad beskrivning av dess utformning. Med reservationen att verktyget befinner sig i ett införandeskede inom Kommundata och att användningen ännu så länge har varit begränsad, pekar författaren på en hel del erfarenheter som man gjort. Erfarenheterna är såväl positiva (de kraftfulla ritfunktionerna, värdet av det projekt-dictionary som ingår i verktyget, den enkla mus-, meny- och meddelandetekniken) som negativa (hänvisningen till ett antal givna ritsymboler, priset, uteslutning av vissa hårdvarumiljöer). Lars Axelsson sammanfattar med att uttala att ett grafverktyg kan stödja oss i arbetet att tänka klart tillsammans, men att tanken likväl måste tänkas först — kompetens och praktiska metodsteg är fortfarande helt avgörande för hur lyckat

ett projekt blir. Det andra inslaget i kapitel 4 svarar Christer Fall, Programator, för. I samband med verksamhetsanalys enligt MBI-metoden inom kriminalvården, har man prövat ett MacIntosh-baserat datorstöd kallat DAMBI (Datorstöd för MBI) i kombination med ordbehandlare. Inledande dokumentation från blädderblock o dyl har överförts till datorstödet och inför seminarier har overhead-kopior på graferna i DAMBI framställts. I sitt bidrag sammanfattar Christer Fall ett antal praktiska och konkreta erfarenheter av både positiv och negativ natur från denna användningssituation.

Kapitel 5: "Tankar kring datorstöd", är författat av Anders Persson, Data Logic. I ett förslag till datorstödsstruktur urskiljer han grova skisseringsstöd, mer 'sofistikerade' modelleringsstöd samt stöd av Data Dictionary-typ. Olika datorstöd riktar sig till olika målgrupper och har olika funktionell betydelse allteftersom förändrings- och utvecklingsarbetet fortskrider. Behovet och vikten av att kunna transportera modelleringsresultat mellan olika modelleringsstöd och mellan modelleringsstöd och Data Dictionary poängteras. Stora samverkans effekter och nya utvecklingsmöjligheter skulle kunna uppnås därmed, menar författaren. Men för att detta skall kunna realiseras, måste vi, betonar han, definiera upp området och enas om vilka begrepp vi vill hålla information om dvs vi måste "beskriva vad som skall ingå i vårt dataregister".

Mats-Åke Hugoson, Programator, påpekar i sitt bidrag: "Kan datorstöd förbättra verksamhetsanalysen?" (kap 6), att processen att datorstödjat arbetet med verksamhetsanalys måste ske enligt vissa premisser annars kan det finnas risk att datorstödsanvändning kan vara en hindrande faktor i arbetet inom detta metodområde. Han betonar att datorstöden måste utgöra ett direkt stöd för de verkliga användarna när deras verksamhet analyseras och anger ett antal krav och restriktioner för detta. Om dessa inte iakttas, menar han, kan hjälpmedlen istället försvåra användarmedverkan och direkt störa verksamhetsanalysen.

Ulf Sundin, SYSLAB, urskiljer i sitt inlägg: "Krav på modelleringsstödssystem" (kap 7), dels problemspecifika krav (man använder viss systemutvecklingsmodell, vissa modellerings-språk etc) på datorstöd och dels generella krav, som t ex det att ett modelleringsstödssystem inte bara skall vara ett verktyg för konstruktion och manipulation av modeller, utan det skall också ha kapacitet att analysera och utvärdera modeller, samt att utföra resonemang. Ett av problemen här är, menar han, att de flesta typer av modeller som idag används vid systemutveckling inte har en formell syntax och/eller en väldefinierad semantik. Detta gör det svårt att veta vad det är som kontrolleras och vad en slutsats betyder.

I kapitel 8 återfinns en lista över referenser, och i ett avslutande appendix, skisseras en egenkapsram med vars hjälp datorstöd kan karaktäriseras.

Kapitel 1

GRAFISKT BASERADE DATORSTÖD FÖR SYSTEMBESKRIVNING — EN INTRODUKTION

1.1 Typer av datorstöd

För att kunna diskutera egenskaper för grafiskt baserade datorstöd för systembeskrivning, för att kunna jämföra dem och kunna avgöra vart utvecklingen är på väg, vill vi först ange och beskriva ett antal, som vi tycker, viktiga grupper inom denna kategori av verktyg. Det finns naturligtvis flera alternativa indelningar som man kan göra. Vi har valt ett sätt, där varje grupp betonar en speciell egenskap, men där gränserna mellan grupperna på intet sätt är skarpa.

Grupperna är:

- Ritorienterade datorstöd
- Verktyg som har mer av analys- och kontrollmöjligheter av modellstrukturerna ('intelligens')
- Modelltypsoberoende verktyg
- Dictionary-liknande verktyg
- Verktyg som kan hantera dynamik
- Verktyg som också stöder körning av informationsmodeller (prototyping).

Som man direkt kan förmoda så finns flera datorstödsansatser i flera av grupperna, vilket inte bör störa det följande resonemanget.

Ritorienterade stöd

Många datorstöd för systembeskrivning som finns på marknaden idag kan rubriceras som ritstöd. Bakgrunden till att de har utvecklats är, att man har haft problem med att hantera de grafer som skall skapas i många modelleringstekniker för systemutveckling. Man har haft behov av stöd för att rita och modifiera dessa grafer. Speciellt har modifiering av grafer upplevts som arbetssamt att klara av manuellt. Att man har funnit "att det blev så mycket grafer att hålla reda på", kan vara en orsak till att många modelleringstekniker inte har blivit accepterade och använda i en vidare krets.

Man har därför, som ett led i att förbättra marknadsföringsmöjligheterna för modelleringsteknik och eventuell åtföljande metodik, skapat dessa datorstöd.

I vissa fall har man skapat speciella versioner av CAD-verktyg för att hantera sina grafer, där man dessutom haft tillgång till god upplösning på skärmarna.

De ritorienterade verktygen har ofta förblivit just ritorienterade, och eftersom de inte har varit avsedda att vidareutvecklas så har man nöjt sig med att enbart skapa ritstödsfunktioner. Dessa datorstöd innehåller därför inte någon modelleringsmässig representation av vad som har beskrivits på skärmen, vilket innebär att datorstödet inte 'förstår' och inte kan göra några analyser av modellen. Det är ofta fråga om att datorstödet endast lagrar en "bitmap"-representation av vad som finns på skärmen (tända och släckta punkter på skärmen).

Verktyg med kontroll- och analysmöjligheter. Modelltypsoberoende verktyg

En del datorstöd har bl a haft det kravet på sig att vara oberoende av modelleringsteknik. Det innebär att ett sådant stöd skulle kunna rubriceras som ett **generellt** modelleringsstöd.

Detta mål ställer betydligt större krav på utformningen av datorstödet. Man behöver en flexibel grund-design för att se till att egenskaper hos olika typer av modelleringstekniker kan representeras och stödjas. Exempelvis är en del grafiska beskrivningstekniker hierarkiska till sin karaktär, medan andra resulterar i en modellstruktur som ligger "i ett och samma plan".

De flesta datorstöd skapas för att enbart stödja (och marknadsföra) en viss beskrivningsteknik och då behöver ovanstående krav inte beaktas. I andra fall har man antagit denna ökade ambitionsnivå, som t ex i den datorstödsutveckling som bedrivs inom SISU (RAMATIC).

Om man har ett datorstöd av generell typ, så måste man också kunna **definiera** olika be-

skrivningstekniker för datorstödet. Det kan ske på ett mer eller mindre specialiserat sätt. Vill man att icke-programmerare skall kunna klara detta så får man också skapa ett språk som denna användarkategori har möjlighet att förstå.

Denna typ av verktyg kan också ge möjlighet att efterhand utvidga beskrivningstekniker med nya symboler. Om datorstödet i övrigt kan klara detta, kan man också bygga på med olika typer av kontroller och analyser.

Dictionary-liknande verktyg

En del datorstöd för systembeskrivning är inriktade på att nå fram till varianter av konventionella data dictionaries. De tidiga formerna av data dictionaries hade egentligen bara som mål att lagra, systematisera och beskriva termer och dataelement och att hålla reda på var de användes i olika procedurer, moduler, program, etc. Genom bland annat dåliga interfaces till systemen, har dessa många gånger ej fått den användning som man hoppats på från ansvarigt håll i organisationerna.

På senare tid har ett ökat intresse för modellering, inte minst för så kallad konceptuell modellering, haft den inverkan att man också vill kunna beskriva hur de olika termerna hänger ihop och relaterar sig till varandra på en "högre", mer "namn-oberoende" nivå (termer är namn, men namn på vad?). På detta sätt kommer nyare former av dictionaries att innehålla också beskrivningar av de objekt/entiteter som de olika termerna "hänger i". Därigenom får den bakomliggande (eller inkluderade) modellen till dictionaryet ett större semantiskt innehåll och ett större uttrycksförmåga. Om man funderar på att införa en ny term i ett nytt informationssystem och tittar i sitt dictionary om det redan finns någon liknande term där, så finns det större möjligheter att upptäcka och avgöra detta om objekt och relationer som 'förklarar' befintliga termer också finns med.

Under den allra sista tiden har uttryckts intresse, inte minst bland medlemsföretagen i SISU, att man skall utvidga dictionaries till att också innehålla modeller som inte bara beskriver data, utan som beskriver funktions- och organisationsstrukturer i verksamheter m m. Bland planeringsgrupperna inom SISU förefaller det finnas ett antagande om, att om dessa typer

av modellering används under utvecklingen så utgör de resulterande modellerna också bra instrument i samband med förvaltning och framtida utveckling och förändring.

Verktyg som kan hantera dynamik

Idag har vi inte så många modelleringsansatser som aktivt modellerar händelser t ex för att i en verksamhetsmodell kunna analysera hur en viss verksamhet fungerar. Detta skulle kunna ge bättre möjligheter att tänka sig in i t ex hur en ny funktionsstruktur klarar de händelser som är, och bör, vara aktuella. Genom händelsebegreppet skulle man också kunna få en koppling mellan objektorienterad och funktionsorienterad modellering. Man kan resa frågan om kanske en anledning till att händelseorienterad modellering inte har vunnit större användning är, att man saknat effektiva sätt att hantera denna typ av modellering, både vad gäller att skapa och att förändra sådana modeller.

En möjlighet är därför att modelleringsstöd, förutom att kunna hantera modellering av statisk typ, också stöder modellering av skeenden t ex genom någon form av animering. Man kan tänka sig att i en sekvens av grafer beskriva hur en "extern händelse" till en verksamhet orsakar att olika aktiviteter utlöses i olika funktioner. Beroende på olika inlagda villkor kan olika delar av funktionsstrukturen med ingående aktiviteter bli inblandade. Att modellera ett ärende genom en kedja av händelser ligger här nära till hands.

Projektet MODES (MODELing Support) inom SYSLAB Göteborg bedriver forskningsverksamhet som bl a inriktar sig på frågeställningar av ovanstående typ.

Verktyg som stöder prototyping

En annan grupp av verktyg utgörs av sådana datorstöd som kan bistå med att göra en informationsmodell körbar. Detta kan vara aktuellt om vi har samlat informationskrav och har modellerat informationsinnehållet med hjälp av informations- och datamodeller. Det har visat sig vara mycket intressant att också, åtminstone till vissa delar, kunna "köra" modellen för användargruppen. Mer utvecklade sådana stöd närmar sig så kallade applikationsgeneratorer el-

ler experimentssystemverktyg.

Man kan här dels tänka sig att lägga till prototypingfunktioner till ett modelleringsstöd, dels att skapa sk exportfunktioner för överföring av modelleringsresultat till en applikationsgenerator och via denna snabbt få till stånd en körbar informationsmodell.

En annan intressant typ av verktyg, som kan nämnas i detta sammanhang och som möjligen skulle kunna betraktas som en mellanform till dessa typer av datorstöd, är sådana som förutom att kunna hantera modeller på typnivå också kan lagra instanser enligt dessa. Det innebär att man för användarna kan exemplifiera instanser t ex av ett visst objekt (eller kanske av en viss objekttyp). Det gör att ej så modelleringsvana personer lättare kan "känna sig hemma".

1.2 Metodik

I detta avsnitt tar vi upp några aspekter, som avser kopplingen mellan metod och datorstöd.

Hur kan de olika metodiska arbetsätten påverkas och förbättras genom tillkomsten av mer avancerade verktyg, och av nya typer av verktyg, inom området?

Det är viktigt att först påpeka att datorstöden knappast i sig själva bidrar till ett förbättrat arbetssätt och till förbättrade resultat vad avser systemutveckling och förvaltning. Att använda datorstöd för någon typ av modellerings-teknik, och inte ha en systematisk metodik att arbeta efter, är ett angreppssätt som förmodligen inte kommer att ge så goda resultat.

Däremot kan datorstöd hjälpa till med att åstadkomma ett genombrott för en metodik. Det kan ske på flera sätt. Dels kan riktlinjer, rekommendationer och råd avseende metoden och dess användning finnas lätt tillgängliga att "slå upp" i datorstödet. Dels kan man i datorstödet implementera kontroll- och stödfunktioner som ger sig tillkänna när man bryter mot olika viktiga regler i metoden eller som gör påpekanden när inte "tillräckligt goda" eller "fullständiga" resultat har uppnåtts i något arbetsmoment.

På detta sätt kan man räkna med att få bättre genomslag för de metoder och de modelle-

ringsansatser som man vill använda, eller som man i en organisation vill skall bli använda och uppskattade.

Några tänkbara effekter som kan uppnås genom användning av datorstöd är:

1) Mer tid för metoden och dess intentioner

Genom att verktyget ger stöd så att man effektivare kan förändra modellstrukturer och se till att ändringar och konsekvensändringar får genomslag och blir korrekt utförda, så kan mer tid användas till det egentliga modellerings- och analysarbete, som avses i metodens olika steg. Man kan ägna mer tid åt att lösa problemen med informationsförsörjningen i en verksamhet etc.

Kan man sedan också slussa de resulterande modelleringsresultaten vidare maskinellt till t ex en applikationsgenerator och till separata "förvaltnings-dictionaries", så kan man spara mycket inmatningstid och undvika många fel, och snabbare komma vidare i utvecklingsarbetet.

2) Korrekta modeller

Ett datorstöd kan hjälpa till med att bevaka att man inte skapar modeller, som bryter mot de olika syntaktiska och semantiska regler som gäller för modelltypen. Symboler får inte kopplas till varandra hur som helst, namn måste vara unika i viss angiven utsträckning, etc. Motstridigheter eller tvetydigheter i modellerna kan upptäckas och indikeras. T ex kanske det gäller i en viss modelltyp att objekt och verksamhetsfunktioner inte får ha samma namn eller t o m inte bör ha namn som är alltför lika. Genom att söka efter likheter i namn eller delar av namn, kan datorstödet upptäcka och indikera detta. Vidare kan datorstödet ge viss hjälp med att finna att det t ex redan finns ett objekt i modellen som skulle kunna vara detsamma som det man just håller på att introducera. Också här kan textmatchning användas. Ett annat sätt skulle kunna vara att granska objekts omgivning t ex att titta på vilka attribut som olika objekt har för att urskilja vilka objekt som har många gemensamma attribut. Sådana indikationer från stödets sida kan också resultera i att "modelleraren" skapar en "generisk relation" mellan aktuella objekt där ett antal av

attributen är gemensamma för de båda objekten. Man kan tänka sig fler exempel av detta slag.

3) Fullständiga modeller

Fullständighet av olika slag är också en viktig egenskap i anslutning till modellers korrekthet. Ett datorstöd kan kontrollera att man skapat sin modell till viss fullständighetsnivå. Det kan bl a gälla att alla symboler av en viss typ måste ha minst en "ingående relation" och minst en "utgående relation" för att anses vara fullständigt specificerad. Enklare former av fullständighet kan vara att all obligatorisk information av "dictionary-typ", som krävs av modelltypen, har fyllts i för de olika typer av fenomen som beskrivits i modellen.

1.3 Effekter på systemutvecklingsprocessen och för förvaltning av system

Om vi använder oss av välutvecklade och exakta modelleringsansatser under utvecklingen, så bör vi också utnyttja de framtagna modellerna för att på ett effektivt sätt införa önskade förändringar i systemen när de är i drift. Vi kan på detta sätt bli bättre på att förvalta och underhålla våra system.

Likaså bör det vara intressant att utgå från de modeller som används för förvaltning när vi påbörjar en nyutveckling. Det kan innebära att vi utgår både från existerande beskrivningar av verksamheten och från beskrivningar av informationssystem när vi påbörjar förändringsarbetet. Man behöver då t ex inte spendera så mycket tid på att genomföra modellering av nuläget. För övrigt läggs nog många gånger för mycket tid ner på nulägesbeskrivningar till förfång för nya goda idéer om hur man skall gå möjligheter och problem till mötes.

Många upplever att det är en stor och arbetsam process att lära sig en ny modelleringsansats. Detta är sannolikt en stor anledning till att man ofta arbetar vidare med de tekniker man använde tidigare och som man är "inkörd på".

Om ett modelleringsstöd kan göra det lättare att lära sig och komma igång med en ny ansats skulle det därför vara en stor fördel. Men det gäller då att verktyget verkligen stöder model-

leringsansatsen i dess olika delar, vad gäller symboler, syntax och andra regler, att det ger tillgång till hjälp i olika lägen, etc. I mer avancerade framtida datorstöd kanske man har tillgång till s k lärandefunktioner, vilket innebär att datorstödet kan känna av hur det används och ge instruktioner till användaren, instruktioner som är anpassade till dennes kunskapsnivå.

Som nämnts i föregående avsnitt, kan man räkna med bättre kvalitet på resultaten från utvecklingsarbetet om man har tillgång till datorstöd som kan bidra med goda och välutformade kontroller för validering och verifiering i modelleringsarbetet.

Det troliga är att om modelleringsstöd av denna typ skall vinna nämnvärd acceptans och användning så måste man koppla användningen till andra former av stöd. Ett exempel är stöd för det projektadministrativa arbetet.

Man använder sig ofta av faser, etapper och aktiviteter för strukturering av projekt. För att genomföra projektet åtgår mantid och resurser. Tid för aktiviteter skall reserveras och planeras, resurser skall kalkyleras, resursförbrukning och tidsåtgång skall följas upp, etc. Mötesprotokoll, korrespondens m m skall dokumenteras. Det finns idag på marknaden och i användning många projektplaneringsstöd kanske dock huvudsakligen inriktade på ren tids- och resursadministration, och det förefaller finnas behov av mer utvecklade och anpassbara stöd för det projektadministrativa arbetet.

Mycket av det som berörs i det projektadministrativa arbetet är också saker som hanteras i modelleringsstödet. De beskrivningar som görs och de modeller som skapas med hjälp av modelleringsstödet, utgör, eller ingår i, etapp- och aktivitetsresultat, som skall avrapporteras, kvalitetsgranskas och dokumenteras. Dessa kopplingsmöjligheter mellan modelleringsstöd och projektstyrningsstöd borde kunna utnyttjas effektivare, dels för att definiera vilken kvalitet modelleringsresultatet skall ha när det är färdigt och dels för att kunna gå direkt in och titta hur långt resultatet avancerat.

Alternativa tillvägagångssätt finns här. Antingen kan man utvidga modelleringsstöden med projektadministrativa funktioner eller så kan

man koppla ihop dessa två typer av datorstöd. Viktigt är dock att de båda stöden kan komma åt varandras strukturer.

1.4 Olika användningssituationer för modelleringsstöd

Lokal användning

En viktig användningssituation för modelleringsstöd är den, då en hel projektgrupp använder sig av stödet i sitt analysarbete och i sitt arbete med att skapa lösningar. Det torde här vara en stor fördel om datorstödet är lätt flyttbart och om man, exempelvis vid internat, kan använda sitt datorstöd lokalt med fulla prestanda på någon kursgård.

Enkla stöd för förstudier

En användningssituation, som möjligen kan ses som en variant av den förra, är att man vill använda datorstöd för begränsade insatser inom ramen för en förstudie. Man har då inte behov av avancerade modelleringsstöd, utan klarar sig i allmänhet med ett enklare ritorienterat verktyg för skisser. Denna typ av önskemål kunde vi notera inom planeringsgruppen "Datorstöd inom AU-processen" under dess arbete med att generera projektidéer inom ramen för SISUs planeringsverksamhet.

Fleranvändarstöd

Vid stora utvecklingsprojekt med många delprojekt, kan det vara en fördel om ett modelleringsstöd kan köras av flera användare eller grupper av användare samtidigt. Det kan också vara en stor fördel om projektmedlemmar under tiden mellan sina projektarbetspass kan köra modelleringsstödet från sin arbetsplats, och inte bara från den "grupparbetsstation" där man arbetar med hela projektgruppen.

Här kommer också vissa säkerhetsaspekter in som t ex att man kan få ha tillgång till modellstrukturer hos grannprojektet medan modifiering bara får ske av de modeller som man hanterar i det egna delprojektet.

Distribuerad systemutveckling

För många projekt gäller att projektmedlem-

marna fysiskt är placerade ganska långt ifrån varandra, varför man tänker sig för innan man samlar projektgrupper och referensgrupper. Man kanske, vilket är vanligt i större organisationer och koncerner, har tillgång till system för elektronisk post. Det kan då vara av intresse att också använda dessa postsystem för att löpande skicka ut resultat och utkast från projektet och dess medlemmar till berörda för att snabbt få synpunkter. Ett problem har ju hittills varit den låga svarsfrekvensen på remisser som sänds ut från arbetsgrupper, ofta i form av "tjocka luntor".

Kommunikationsstöd

En viktig effekt av användning av grafiska modelleringsstöd är att de skall bidra till förbättrad kommunikation t ex mellan medlemmarna i en projektgrupp.

Detta kan ske på flera sätt:

- genom att framställningarna är grafiska
- genom att de har en "snygg" utformning även om man har ändrat mycket i dem
- genom att man snabbt kan växla mellan grafer när man har kommit på att man måste titta i andra grafer
- genom tillgång till datorstöd och sådana terminaler som tillåter hela grupper att se flera grafer samtidigt
- genom att man snabbt kan förstora en viss symbol till en ny bild
- genom att man snabbt kan förändra modellerna, så att intresset för, och koncentrationen på det som skall modelleras kan upprätthållas. Tar manipulering för lång tid, hinner intresset svalna och koncentrationen att minska
- genom möjligheten att då och då plotta ut grafer på overhead-film för diskussion i plenum.

Kapitel 2

PRESENTATION AV NÅGRA EXISTERANDE DATORSTÖD

I detta avsnitt presenteras kortfattat följande datorstöd:

1. DAMBI (Datorstöd för MBI)
2. Data Designer
3. DDEW (Database Design and Evaluation Workbench)
4. Design Manager
5. EXCELERATOR
6. GRAPHDOC
7. JSP-TOOL
8. LID (Living In a Database system)
9. Sembase
10. Tools for Supporting Structured Analysis
11. USE (User Software Engineering)

Presentationerna är informella, i den meningen att de inte följer någon i förväg uppgjord strikt mall. Avsikten är att de skall ge en första idé om de olika stödets huvudsakliga egenskaper och funktioner och illustrera några av de olika typer av datorstöd som finns.

Ovanstående förteckning gör inte heller några som helst anspråk på att vara fullständig. Allteftersom arbetet med detta nummer av SISU-analys har fortskridit, har vi fått kännedom om ytterligare ett antal datorstöd, både ur den synpunkten att de är avsedda för systembeskrivning och att de är grafiskt orienterade. Istället för att redan nu förteckna dessa datorstöd och göra grunda beskrivningar av dem, har vi valt att avvakta med en mer genomarbetad och fullständig behandling av dem till ett framtida nummer av SISU-analys. Där kommer också den kriterie- eller egenkapsram, vartill ett utkast återfinns sist i detta nummer som ett appendix, att komma till användning.

Med dessa reservationer beträffande urval, uppläggning och fullständighet gjorda, är det dags att presentera de ovan uppräknade 11 datorstöden. Presentationerna görs i bokstavsordning och ingen innebörd skall läggas i att de varierar något i längd. I slutet av varje presentation anges det underlag, i form av manualer, artiklar, intervjuer etc, som använts för att upprätta beskrivningen. Referenser hänvisar till referenslistan i avsnitt 8.2.

2.1 DAMBI (Datorstöd för MBI)

Datorstöd för MBI är en applikation som byggts upp kring programvaran Filevision från Telos Software i USA. Datorstödet avser att stödja verksamhetsanalys enligt MBI-metoden /HUG-83/, med bland annat:

- * Grafiskt ritstöd som minimerar iterativt ritarbete
- * Vissa analys- och utsökningsmöjligheter
- * Dokumentationsstöd
- * Metodstöd

Datorstöd för MBI är utvecklat av datakonsultföretaget AB Programator under våren 1985. Datorstödet används främst inom Programator-koncernen men distribueras även till andra intresserade, som använder MBI-metoden.

Datorstödet är gjort med hjälp av Filevision, en programvara till Apple Macintosh, som kombinerar grafik och databas. Filevision skulle kunna stödja många andra metoder än MBI-metoden eftersom den är enkel att anpassa efter olika behov.

MBI-metodens steg med arbetsmoment, syften och resultat finns beskrivna i datorstödet i översiktlig form och kan tas fram såsom metodstöd medan man arbetar.

MBI-graferna ritas upp med klassisk Macintosh-teknik inklusive rullgardinsmenyer och mus. Det är enkelt att rita, kopiera, radera, flytta osv i graferna. "Undo"-funktion finns om man ångrar något kommando. Samtliga symboler som MBI-metoden använder finns fördefinierade och de går att kopiera och använda i de egna graferna. Separata kopieringsbilder finns för funktionsstrukturer, beslutsgrafer samt objektmodeller.

För varje symboltyp finns ett bakomliggande formulär definierat. I detta formulär kan man beskriva en viss symbol genom att fylla i formulärets olika fält. På formuläret kan man göra utsökningar och kombinera olika typer av villkor. Från grafen kommer man till en speciell symbols textsida genom att peka på symbolen och "klicka" två gånger med musen.

Olika grafer kan kopplas till varandra hierarkiskt, sekvensiellt och i nätverk. Man kan hoppa direkt från en översiktlig graf till en underliggande mer detaljerad graf och därefter förflytta sig tillbaka igen.

Datorstödet är tänkt att användas av MBI-analytiker i samarbete med användare. Då en MacIntosh är portabel så kan datorstödet enkelt tas med ut i olika användarmiljöer.

Stödet skall användas under verksamhetsanalysen i förstudien och lämna ifrån sig underlag för den fortsatta systemutvecklingen; exempelvis bedriven enligt SAK-metoden.

Utskrifter av bilder samt informationen i formulären kan fås ut på fyra olika sätt. Olika typer av och storlekar på text kan fås. Text kan antingen vara fristående på skärmen eller "tillhöra" en speciell symbol. Olika typer av skuggningar för symbolfyllning samt olika linjetjocklekar kan fås.

Det är möjligt att definiera helt nya symboler med egna informationsformulär, samt att förändra tidigare definierade formulär.

Medan man ritar utför datorstödet inga som helst kontroller.

Om man vill prova datorstödet så måste man anskaffa programvaran Filevision, som säljs hos flera återförsäljare av persondatorsoftware. Generalagent för Filevision är Plan-data. MBI-datorstödsdelen kan rekvireras från Programators Göteborgskontor (ref Kerstin Karlsson). Filevision kostar ca 2500 kr och finns i både svensk och engelsk version. Datorstöd för MBI distribueras till självkostnadspris; ca 200 kr. För att kunna använda datorstödet krävs en MacIntosh med antingen två diskettstationer eller en hårddiskmaskin (MacIntosh XL).

Underlag för beskrivningen

Till programvaran Filevision medföljande engelsk manual.

2.2 Data Designer

Data Designer är ett datorstöd, som är avsett att stödja arbetet med att utforma ett logiskt

databasschema.

Den föreslagna utformningsprocessen med användande av Data Designer innehåller följande steg:

1. Användare och analytiker identifierar, utvecklar och beskriver olika informationskrav eller "user views" av data t ex med hjälp av sk "bubble charting". Dessa grafiska beskrivningar kan dock inte skapas och hanteras med hjälp av Data Designer utan de måste först kodifieras i ett speciellt språk innan de matas in (se steg 3).
2. En relevant standardlista över term-/dataelementnamn extraheras ur existerande data dictionary och läggs i en fil, som bildar input till Data Designer. Om inget data dictionary finns, kan en sådan lista skapas med hjälp av Data Designer.
3. Beskrivna "user views" kodifieras och matas in i Data Designer.
4. "User views" (dvs ingående termnamn) jämförs med namnen i standardlistan och omatchade namn presenteras för vidare åtgärd: antingen tillägg till data dictionaryet eller ändring av namnen så att de ansluter till standardnamngivningen.
5. Data Designer kombinerar nu de så validerade "user views" med hjälp av sk "kanonisk syntes" till ett preliminärt, normaliserat (3NF), logiskt databasschema.
6. Det föreslagna schemat utvärderas och förfinas successivt.
7. Den logiska utformningen transformeras till den databasmodell, som används i aktuellt databashanteringssystem. Operationella krav som t ex uppgifter om responstider och accessfrekvenser kan matas in och presenteras av Data Designer som underlag/krav för den fysiska konstruktionen.

Det syntetiserade logiska databasschemat avses alltså förfinas successivt. Data Designer kan producera ett antal olika rapporter och grafiska presentationer som underlag till denna process. Tillägg och ändringar i "user views" diskuteras fram och matas in i Data Designer, som syntetiserar och genererar ett

nytt schemaförslag; olika alternativa strukturer kan provas etc.

De grafiska presentationerna kan tas ut på printer, på plotter och på grafisk terminal. Tre olika typer förekommer:

1. En "komplett" graf, som beskriver alla "datagrupper" (relationer) och deras samband (funktionella och mångvärda beroenden mellan nyckelattribut). Primär-, sekundär- och konkatenerade nycklar indikeras.
2. En översiktsgraf, som enbart visar datagrupper och deras samband. Alla attribut undertrycks.
3. En detaljgraf, som separat beskriver varje datagrupp och deras resp nyckel- och övriga attribut.

I tillägg till de grafiska beskrivningarna av den logiska strukturen, kan ett flertal olika speciala rapporter genereras.

Data Designer är utvecklat av Database Design Inc (DDI), Ann Arbor, USA.

Underlag för beskrivningen

/DAT-81/.

För ovanstående beskrivning måste reservation göras för ev vidareutveckling av verktyget som kan ha skett under de senaste åren.

2.3 DDEW (Database Design and Evaluation Workbench)

Database Design and Evaluation Workbench (DDEW) kan ses som en grafisk arbetsstation för databaskonstruktörer. DDEW tillhandahåller en interaktiv miljö för att specificera och experimentera med databasstrukturer och konstruktioner. Samtidigt håller DDEW reda på alla de konstruktionsalternativ som undersökts. En enkel och enhetlig åtkomst av en integrerad och utökningsbar samling av verktyg för utvärderingar, analyser, och transformationer finns och täcker hela databaskonstruktionens livscykel.

DDEW är framtagen av ett projekt vid Compu-

ter Corporation of America (CCA), Cambridge, MA. En prototyp skall finnas färdig sedan juli 1985.

På arbetsstationens skärm kan användaren bygga, visa och manipulera objekt av två typer: listor och diagram. Friformat och formaterad text representeras som listobjekt, medan databasscheman och konstruktionshistorien representeras som diagram. Både listor och diagram kan redigeras med DDEWs redigeringsprogram.

DDEW har en integrerad metodik för databaskonstruktion som stöd dels för ett effektivt utnyttjande av systemet av alla konstruktörer och dels som undervisningshjälp för oerfarna konstruktörer. Nya och förbättrade verktyg kan inkorporeras i DDEW när de blir tillgängliga. En utvidgad version av "Entity-Relationship"-modellen ligger bakom alla konstruktionsfaser i DDEW och denna används vid konceptuell design. Logisk konstruktion görs i relations-, nätverks-, eller hierarkiska modeller.

Sex steg ingår i metoden: kravanalys- och specifikation, konceptuell design, logisk design, distribuerad databasdesign, lokal fysisk databasdesign, samt "prototyping". Metoden är iterativ så tillvida att tidigare beslut alltid kan omprövas och alternativ till dem kan undersökas parallellt. Vid varje steg finns ett antal verktyg tillgängliga. Andra verktyg används för att föra, och eventuellt transformera, resultat från ett steg till nästa.

Allteftersom konstruktionen framskrider för DDEW ett grafiskt protokoll över alternativ som provas och deras inbördes relationer, i form av ett konstruktionsträd. Detta förhindrar att arbete förloras, samtidigt som man kan se hur arbetet fortskrider. Trädet innehåller kommentarer, analysresultat, m m och tillåter att man inspekterar förhistorien till en viss konstruktion.

Underlag för beskrivningen

/REI-84/.

2.4 Design Manager

Design Manager är ett dictionary-baserat system för automatisk generering av logiska databasmodeller. Det har utvecklats av Management Software Products i USA och England.

Design Manager kan köras på alla IBM stordatorer och maskiner som är kompatibla med IBM. Det är körbart interaktivt eller i batch.

Användningsområden

Design Manager har två huvudsakliga användningsområden, dels som ett lagringsmedium för dataelement och "user views" och dels som design-verktyg för att producera logiska databasmodeller. Design Manager arbetar därför i två olika "modes", DICTIONARY MODE och DESIGN MODE.

De som arbetar med Design Manager är i första hand personer som arbetar med datamodellering och logisk design av data inom företaget.

Generering av normaliserade strukturer

Inmatning sker i "dictionary mode" till ett så kallat "modeling dictionary". I "design mode" kan sedan t ex önskade "user views" laddas in i en speciell design-arbetsarea och logiska strukturer, som, enl användarens val, är på 1:a, 2:a eller 3:e normalformen, kan genereras. Från dessa logiska strukturer/relationsscheman kan Design Manager vidare generera förslag till hierarkiska strukturer för IMS-databaser och tabelldefinitioner för dBase II och SQL/DS-databaser.

Design Manager använder sig alltså av normaliseringsteknik för att analysera data och framställa logiska utformningsstrukturer, som, genom att redundans elimineras, kommer att kräva minimalt antal operationer avseende nyupplägg, borttag och uppdatering.

"User views" kontrolleras vid inladdning till arbetsarean med avseende på homonymer och synonymer.

Förutom ovanstående kan även datamodeller göras utifrån "Top-Down"-modellering, som

innebär att man skapar datamodeller över de data som används inom företaget.

Grafiska beskrivningar

Design Manager innehåller i nuläget ingen interaktiv grafisk editor (se dock under "Framtida planer" nedan). Grafiska beskrivningar av logiska strukturer och datamodeller ritas upp utifrån verbalt definierade objekt och relationer. De bilder som skapas går alltså inte att ändra interaktivt. Bilderna med symboler som ritas, är egentligen enbart ett antal textfiler som är skapade.

Rapporter

Förutom att ta fram logiska databasmodeller kan även ett antal rapporter med olika tonvikt på innehållet genereras, t ex rapporter över vilka dataelement som finns lagrade, relationsscheman, etc.

Koppling till andra verktyg

Design Manager kan kopplas till data dictionary-systemet Data Manager. Information kan flyttas mellan de båda systemen. De flesta som kör Design Manager gör det integrerat med Data Manager. I Sverige finns det ingen användare som kör Design Manager enbart utan alla använder det integrerat med Data Manager. De lagrar sina data i Data Manager och hämtar över dem till Design Manager när de vill generera någon rapport.

Framtida planer

MSP håller på att utveckla ett skissverktyg för PC som kommer att bli av typ Excelerator och GraphDoc. Det kommer, enligt uppgift, att hetta X-manager.

PCen använder man fristående eller som en arbetsstation kopplad till en stordator. I stordatorn ligger alla data lagrade och man kan hämta in eller lagra data till/från PCen. Om PCen används fristående så hämtas de data man vill ha först in till PCen där bearbetning sker och därefter kan resultatet föras tillbaka till stordatorn.

Med X-manager blir det möjligt att skapa grafer och sedan överföra strukturen på grafen direkt till Design Manager eller Data Manager. Man kommer även att kunna skapa grafer utifrån en redan lagrad struktur. De skapade graferna kan sedan tas ut på plotter.

Verktyget X-manager kommer att släppas under 4:e kvartalet 1986.

Underlag för beskrivningen

Information från: Management Software Products i Stockholm, Humblegatan 22, 172 39 Sundbyberg, 08/764 56 60.

Manualer till Design Manager.

2.5 EXCELERATOR

Excelerator är ett flerfunktionsverktyg för IBM PC-kompatibla datorer och avser att stödja större delen av systemutvecklingsprocessen; från tidiga förstudieskisser till prototyping. Verktyget är till viss grad generellt genom att det stödjer flera metoder för systemutveckling.

Excelerator är utvecklat av Index Technology Corporation i USA. Företaget har ett hundratal anställda, varav ungefär tio stycken arbetar med utveckling av datorstödet. I Sverige säljs Excelerator av Kommundata, i Norge av ISI A/S och i övriga Europa av James Martin. I Storbritannien och USA finns "hotline"-telefonnummer dit man kan ringa dygnet runt om man får problem med datorstödsanvändningen. Excelerator har 2-3000 användare i USA och IBM har köpt försäljningsrätten till programvaran där. 1983 vann Excelerator programvarutävlingen Million Award.

Huvudfunktionerna i verktyget är:

- * Grafik
- * Data dictionary
- * Skärmbildsgenerator
- * Rapportgenerator
- * Dokumentationsfunktion
- * Servicefunktioner

Grafik

Med Excelerators grafikfunktion kan man åstadkomma flera olika typer av grafer såsom

datamodeller, verksamhetsgrafer, systemmodeller, programstrukturer och dokumentationsgrafer. Ett stort antal symboler finns att välja på. I Excelerators dictionary lagras samtliga bildelement och de relationer man etablerat mellan dem, samt eventuella elementbeskrivningar. Man arbetar hela tiden med hjälp av en mus med vilken man väljer funktioner i funktionslistan i bildens vänsterkant. Med musen visar man också var på skärmen, som man önskar placera en utvald symbol. Skärmbilden är endast en del av den totala grafbilden och man kan "scrolla" åt sidorna respektive "zooma" i graferna. Varje objekt på bilden kan "explodera" i en underliggande bild på en mer detaljerad nivå (tio underliggande nivåer kan skapas).

Data dictionary

Excelerator har ett aktivt data dictionary som uppdateras omedelbart då en förändring görs i grafer eller texter. I data dictionary lagras information om:

- * Data
- * Processer
- * Grafer
- * Input-data

Data dictionary utgör kärnan i Excelerator och medför att en beskrivning av ett objekt som skapats i en av funktionerna, exempelvis grafikfunktionen, blir tillgänglig för övriga funktioner. Det är möjligt att direktuppdatera data dictionaryet utan att gå via någon grafisk bild.

Skärmbildsgenerator

Med Excelerators skärmbildsgenerator kan man "måla upp" inmatningsskärbilder. Varje inmatningsfält kan kopplas till en term i data dictionary och åsättas egna hjälptexter, validerings- och editeringsregler.

Avsikten med skärmbildsgeneratoren är att användaren snabbt skall kunna se hur en skärmbildsdialog utformas och komma med ändringsförslag tills layouten är tillfredsställande. Skärmbilderna kan länkas i sekvens. Med Excelerators prototyping-funktion kan skärmen testas direkt mot data i en testfil som byggs upp och lagrats i data dictionary.

Rapportgenerator

Med rapportgeneratoren kan man "måla upp" en rapport och därefter köra den mot en testfil, på samma sätt som med skärmbilderna. Rapporter kan också göras mot data dictionary och med olika urvalskriterier. Fasta analysrapporter om de system som byggs i Excelerator finns.

Dokumentationsfunktion

Dokumentation genereras automatiskt enligt en grafisk dokumentationsstruktur som man själv definierar på ungefär samma sätt som man ritat grafer. I dokumentationen kan exempelvis finnas grafer, tabeller ur data dictionary eller fristående textfiler. Då utskrift beordras "plockar" Excelerator samman de senaste versionerna av de i dokumentationsgraferna specificerade dokumenten och skriver ut dem.

Servicefunktioner

Med hjälp av dessa funktioner installerar man Excelerator och utför "service" åt övriga funktioner. Exempel på funktioner:

- * Definition av kringutrustning (printer, skärm osv)
- * Backup- och inladdningsfunktion
- * Export- och importfunktion
- * Definition av olika användares behörighet
- * Inställning av datum och tid

Med export- och importfaciliteten kan olika datastrukturer föras över till större databasystem och applikationsgeneratorer. Data exporteras och importeras i MSDOS-format.

Övriga funktioner av vikt

Om en datamodell beskrivits kan Excelerator generera deklaraionsdelen i ett program i programmeringsspråken C, PL/1, Cobol, Info, Basic eller Pascal.

I nuvarande version finns inte svenska tecken och alla menyer bär engelska namn, men översättning pågår.

Då man tar ut grafer på printer eller plotter kan

man välja en hel graf eller en mindre del av grafen i valfri förstoring.

Excelerator innehåller ingen egen ordbehandlare utan man får själv "länka" sin egen ordbehandlare till verktyget.

Olika storlekar och fonter kan väljas för texterna i en graf.

Det är inte möjligt att själv definiera egna symboler.

Kommundata har använt Excelerator i olika projekt, bland annat:

- * Datamodeller hos Brandkårens Larmtjänst
- * Fastighetsdata (förvaltning)
- * Kurs- och konferenssystem

Verktyget används av Kommundata i grupper om ett par användare under ledning av en erfaren Excelerator-handledare. Några av de viktigaste fördelarna med användning av Excelerator som man där säger sig ha upplevt är:

- * Excelerator stödjer en mycket stor del av systemutvecklingsarbetet.
- * Dokumentation, analyser och resultat kan hållas konsistent och sammanhållet.
- * Excelerator är ganska generell och läser därmed inte upp användare till en speciell metod, utan kan stödja de flesta metoder som använder grafisk beskrivningsteknik.
- * Stödet är öppet, så till vida att informationen i data dictionary kan föras över till externa system och vidare analyseras eller bearbetas.
- * Prototyping-verktygen (rapport- resp skärmbildsgeneratoren) är lätthanterliga och lättföränderliga. Användarna "känner" att det är en prototyp de arbetar med och törs komma med förändringsförslag.
- * Dokumentationsfunktionen är lätthanvänd och det är mycket praktiskt att kunna begära utskrift av ett större antal dokument samtidigt.

Excelerator finns som en- och fleranvändarsystem. En installation kostar 8400 dollar. Programmet är kopieringsskyddat på så sätt att man måste ha en särskild nyckeldiskett i "discdrive'n" för att kunna använda det. Om man vill prova Excelerator under en månad så kostar det 750 dollar.

Den hårdvara som krävs är: IBM PC/XT, IBM PC/AT eller IBM 3270 PC. För XT och AT krävs ett expansionskort för att få tillräckligt god upplösning på skärmen. För XT och 3270 krävs dessutom ett expansionskort med parallell och asynkron port, klocka och extra minne.

Mus krävs för pekning i grafer och funktionsval.

Ett stort antal standardprintrar och plottrar kan användas.

Underlag för beskrivningen

Halvdagsdemonstration av Excelerator samt intervju med Leif Ortman, Kommundata.

Svenskt och amerikanskt broschyrmaterial om Excelerator.

2.6 GRAPHDOC

GRAPHDOC är ett grafiskt datorstöd, som stöder verksamhetsanalys enligt ISAC-metoden /LUN-78/. I ISAC-metoden, som har utvecklats vid Universitetet i Stockholm, beskrivs verksamheter, informationsmängder och flöden i en hierarkisk struktur av grafer. Genom sk förstoring av delverksamheter i en graf, skapas nya underliggande grafer, som kan förstoras i sin tur. Till varje graf hör en textsida, som skapas automatiskt och som kan redigeras genom att man lägger till ytterligare information i textform. Grafer och textsidor utgör tillsammans dokumentation över den beskrivna verksamheten.

En god egenskap hos GRAPHDOC är, att verktyget alltid håller alla grafer och tillhörande textsidor konsistenta. Om exempelvis en informationsmängd delas i flera mängder, så kontrolleras om informationsmängden redan är delad i någon annan graf. I så fall kopieras den tidigare delningen in i den aktuella grafen. Om en mängd förändras, så förändras den i hela grafstrukturen. Samtidigt uppdateras alla berörda textsidor. Ytterligare ett exempel på hur datorstödet upprätthåller konsistensen är att om man tar bort en förstordad verksamhet, så försvinner automatiskt samtliga underliggande grafer.

GRAPHDOC styrs med hjälp av en meny nerst på bildskärmen. Den grafiska markören flyttas antingen med hjälp av sifvertangentbordet eller med en tillkopplad mus. Att använda en mus måste vara att föredra, eftersom ovana personer annars hela tiden måste flytta blicken mellan bildskärm och tangentbord. Tyvärr verkar musens möjligheter ännu inte vara fullt utnyttjade i datorstödet.

GRAPHDOC kan köras på en IBM PC/XT, om den förses med lämpliga grafik- och styrkort. Datorstödet kan också köras på Victor PC och Sperry PC.

Underlag för beskrivningen

/GRA-85/.

2.7 JSP-TOOL

JSP-TOOL är ett grafiskt och analytiskt verktyg för systemutveckling, som är utvecklat av Jackson Software Sweden. Utvecklingen av datorstödet påbörjades för ca 5 år sedan. Ungefär 100 installationer har gjorts. JSP-TOOL säljs i Norden, Holland, Tyskland och Schweiz.

JSP-TOOL skall kunna stödja modellering och design enligt JSD-metoden /JAC-83/. Tyngdpunkten ligger dock på system- och programkonstruktion enligt JSP-metoden /JAC-75/, /ING-78/. Avsedda användare är programmerare och systemare som arbetar med JSP och JSD.

Utveckling pågår av ett komplement till JSP-TOOL, Jackson Speedbuilder, vilket bl a skall innehålla ett data dictionary som stödjer JSD.

Viktiga funktioner är:

- * Analysfunktioner
- * Automatiskt genererad dokumentation
- * Kodgenerering i olika programmeringsspråk såsom Cobol, Pascal, C, JSP-Cobol. PL/1-generatorn blir klar under november -85 och Fortran- och Basic-generatorerna under -86
- * Avlastning av mini- och stordatorer

JSP-grafer ritas och editeras direkt på skärmen med hjälp av mus och funktionstangen-

ter. Det är inte möjligt att göra kopplingar mellan symboler på något sätt som strider mot JSP-syntaxen. Förstoringar och förminskningar går att göra i flera nivåer.

Datorstödetts funktioner finns i menyer kring skärmbildens höger- och underkant. Funktionerna väljs genom att man pekar på funktionerna med musen och "klickar".

Den skärmbild man ser är bara en del av den totala bilden och man kan "scrolla" sig åt sidorna, respektive upp och ner, för att se bildens övriga delar.

Utsökningsfunktioner saknas för närvarande, men kommer att finnas med i Jackson Speedbuilder.

Exempel på funktioner i JSP-TOOL:

- * Uppläggning av sekvens-, selektions- och iterations-komponenter samt "backtracking"-komponenterna "posit" och "admit".
- * Radering, kopiering, tillägg, lagring och hämtning av strukturer.
- * Infogning av s k stommar.
- * Infogning av komponenter i en sekvens dvs förflyttning i horisontalled.
- * Namnsättning av komponenter.
- * Förstoring och förminskning av strukturer i flera steg.
- * Listning och allokering av operationer, villkor och s k "quits".
- * Hantering av både "gynnsamma" och "icke acceptabla" sideeffekter vid backtracking.
- * Hantering av flera strukturer samtidigt.
- * Texteditering med fullskärmseditor.
- * Generering av både generell strukturtext (pseudokod) och strukturtext av JSP COBOL-snitt.
- * Framställning och dokumentation med operationer, villkor och quits samt ritade strukturer i olika format.
- * Felmeddelanden och menytexter i valfritt språk.
- * Generering av kod i olika språk och dessutom s k nästfri kod.

Om man önskar köpa eller prova JSP-TOOL så vänder man sig till Jackson Konsult AB. JSP-TOOL kostar 7000 kronor. Varje kodgenerator kostar 3000 kronor. Om en köpare köper fler än två stycken JSP-TOOL blir priset lägre.

Skolor och universitet får extra rabatt. Programvaran är inte kopieringsskyddad utan varje program är försett med licens. Datorstödet levereras på diskett tillsammans med en engelsk manual. Då nya versioner av JSP-TOOL kommer får användarna information om detta, samt möjlighet att uppgradera sin version för ett lågt pris.

JSP-TOOL går att köra på IBM PC och kompatibler. En vanlig monokrom skärm kan användas. Den mus som kan användas är av fabriket Microsoft.

Underlag för beskrivningen

Intervju med, och demonstration av, Conny Leinstedt, Jackson Konsult AB.

Svensk broschyr om JSP-TOOL.

2.8 LID (Living In a Database system)

LID (Living In a Database) tillhör den kategori av datorstöd, som på engelska kallas "database browsers". Dessa datorhjälpmedel gör det möjligt att bläddra igenom och titta på informationsinnehållet i olika delar av en databas.

I LIDs fall använder man sig av en "bit-mappad" grafisk terminal med tillhörande mus för att skapa en användarvänlig kontaktyta mot en "Entity-Relationship"-databas. När man tar upp en instans av något informationsslag (post) på skärmen, så får man inte bara se den postens innehåll, utan även med en grafisk bild dess relationer till andra poster i databasen. Genom att följa dessa relationer kan man sedan bläddra fram och tillbaka mellan olika poster i databasen.

Idén bakom LID med grafisk layout av poster och deras omgivning är mycket tilltalande, men i artikeln, som refereras nedan, tar Dennis Fogg samtidigt upp två problem, som återstår att lösa för att göra LID mera praktiskt användbart.

Det ena problemet är, att LID måste kunna hantera inte bara instanser, utan även mängder av instanser på ett enkelt sätt.

Det andra problemet är, att LID som många andra datorstöd, som försöker att automatiskt generera bilder, lider av oförmågan att kunna skapa tilltalande grafer från ett stort antal noder och relationer.

Underlag för beskrivningen

/FOG-84/.

2.9 Sembase

Sembase är ett databashanteringssystem som baserar sig på en semantisk, eller objekt-orienterad, modell. Målet med Sembase är att avgöra om semantisk modellering kan transformeras från att vara ett abstrakt konstruktions-(design-) verktyg till ett användbart datahanteringsverktyg.

Sembase är framtaget vid University of Colorado, Boulder, CO, USA.

Den semantiska modell som finns bakom Sembase representerar en applikationsmiljö som en samling objekt. Alla objekt är instanser av typer och subtyper och arrangeras i hierarkier efter typ-/subtyprelationer. Det finns två typer av objekt. "Descriptor"-objekt har atomiska värden (teckensträng, heltal, etc) och används normalt som symboliska namn i databasen. Abstrakta objekt är icke-atomiska entiteter och definieras i termer av sina förhållanden till andra objekt genom attribut (en relation mellan två objekt). En subtyp formas enligt ett predikat som beror av värden hos attribut hos föräldertypen. En subtyp kan också definieras att vara "godtycklig", vilket betyder att dess medlemskap bestäms av en dataoperation (se nedan) i stället för av ett statiskt predikat.

Databasen manipuleras genom databasoperationer. Dessa innefattar operationer

- som initierar och avslutar användningen av en databas;
- på attribut;
- på objekt.

Abstrakta objekt kan vara ordnade efter det atomiska värdet hos ett attribut för den givna typen. Operationer finns som kan finna den första eller sista instansen av en typ, subtyp, eller multivärd attribut, och som kan stega till

föregående eller efterföljande instanser.

Implementering av Sembase har skett med hjälp av ett verktyg kallat Sedaco (the SEMantic DATabase CONstructor). Sedaco tillhandahåller primitiva operationer för att implementera databaser skapade efter en semantisk modell. Två strukturer skapas av Sedaco: den ena innehåller schemainformation, den andra innehåller data. Förutom de ovan nämnda databasoperationerna så innehåller Sedaco ett antal operationer för att konstruera semantiska databasscheman, samt för att "läsa" ett schema och översätta namn på schemakomponenter till interna referenser.

Nästa del av Sembase är Semdal (the SEMantic DATa Language), som är ett datadefinitions- och datamanipuleringspråk. Semdal innehåller Sedacos data- och schemaoperatorer och ger också tillgång till styrstrukturer och en passande syntax. En programmerare använder Semdal som ett snitt mot Sembase.

Ett grafiskt baserat komplement till Semdal är under konstruktion (hösten 1984). Detta kallas Ski (the Semantics-Knowledgeable Interface). Ski kombinerar uttrycksfullheten hos semantisk modellering med styrkan hos grafisk presentation, och ger en användare möjlighet att använda en databas på ett uppfattbart sätt. En användare har, med hjälp av de operatorer Ski innehåller, möjlighet att få tillgång till och undersöka en databas samtidigt som han kan dra full nytta av de semantiska förhållanden som finns i schemat.

Användaren konstruerar en "session view" grundad på semantiken hos data och styr både val av schemakomponenter och placeringen på skärmen. En "session view" består av en delmängd av schemat, och möjligen nya typer, attribut, restriktioner och subtyper. Man kan permanent notera en eller flera av de nya schemakomponenterna och man kan när som helst även använda sig av språket Semdal.

Förutom delsystemet "Schema View" hos Ski, vilket används vid konstruktion och genomförande av en "session view" finns delsystemet "Data View". Data View används till att undersöka attributvärden och subtypsmedlemskap i en "session view" och ger även tillgång till Semdal. Ett rapportgenereringsdel-system finns för att formatera data för presentation.

Underlag för beskrivningen

/KIN-84/.

2.10 Tools for Supporting Structured Analysis

Datorstödet har utvecklats vid Tektronix Inc, Beaverton, Oregon, USA, och stödjer dedicerat "Structured Analysis" /DeM-78/, /GAN-79/, /WEI-78/.

Strukturerad analys (SA) har använts vid Tektronix under flera år. Erfarenheterna är goda, säger man, och främst upplevs att förbättrade kommunikationer har blivit ett resultat. SA använder grafer och har en notation som, anser man, är koncis och lätt att använda. Tidiga idéer kan identifieras, skissas ner och kommuniceras till de olika gruppmedlemmarna. God kommunikation och tidig "feedback" ger färre och mindre allvarliga specificeringsfel.

Det finns dock problem med manuell hantering: ändringar i grafer är tidskrävande och felintroducerande, kontroll av konsistens och fullständighet av stora modeller blir svårare och svårare ju större modellerna är. Produktivitet och kvalitet påverkas.

SA använder tre typer av beskrivningar: dataflödesdiagram (DFD's), mini-spec'ar och Data Dictionary (DD). Ett DFD beskriver processer (transformationer) och dessas input- och output-dataflöden. En mini-spec är en detaljbekrivning av en process och utförs i begränsat naturligt språk ("Structured English"). I Data Dictionaryet samlas definitioner av de termer, som förekommer i DFD's och minispec'ar. Beskrivningstekniken är hierarkisk till sin natur — processer detaljeras i nya dataflödesdiagram.

Inför utvecklingen genomfördes en grov kostnads/intäktanalys och inriktningen var att inte automatisera mera än som var berättigat enligt denna, även om det skulle vara tekniskt intressant. Så t ex förkastades tanken på att bygga en halvautomatisk översättare som skulle assistera vid övergång från SA till "Structured Design" /STE-74/, /YOU-75/.

De resulterande krav som angavs inför utvecklingen av detta datorstöd var grovt att det skulle kunna modifiera och presentera SA-

beskrivningar, upptäcka ett antal specificerade fel i SA-beskrivningar, och att hjälpa till med att upprätthålla konsistens mellan SA-beskrivningar. Vidare skulle verket vara flexibelt i den meningen att det skulle stödja såväl oerfarna som erfarna användare samt beakta det förhållandet att olika användargrupper hade utvecklat olika varianter av metoden/beskrivningstekniken med hänsyn till notation och i hur rigoröst man tillämpade olika metodregler och riktlinjer.

Datorstödet har realiserats i ett antal delverktyg, som kan användas separat eller i kombination. Ansatsen har inspirerats av UNIX-begreppen "filter" och "script". "Filter"-program utför förändringar i data som passerar genom dem, "script"-program länkar ihop "filter"-program till större konfigurationer. Användare kan själva definiera nya "scripts" dvs kombinera ihop fördefinierade "filter" till för dem lämplig konfiguration, men de kan också definiera nya "filter". Abstrakta datatyper för ett DFD, en mini-spec, etc, har fördefinierats för att underlätta detta.

Fyra huvudfunktioner finns:

1. Redigering: interaktiv modifiering av SA-beskrivningar och härledning av behov av ändringar som sedan utförs av den s k "Clean-up"-faciliteten.
2. Kontroll: upptäckt av fel i SA-beskrivningar.
3. Formatering: konvertering av modellerna från intern representationsform till lämplig presentationsform.
4. "Clean-up": validering och utförande av ändringskrav härledda av redigeringsverktygen.

Redigeringsverktygen är två: "Edit DFD" och "Edit SA". "Edit DFD" är en interaktiv grafisk editor med vars hjälp man skapar och modifierar dataflödesdiagram. Mus eller tumhjul används för positionering av grafelementen. "Edit SA" används för grafövergripande modifieringar. Enkla syntaktiska regler upprätthålls: ett dataflöde kan t ex inte introduceras om inte både dess ursprung och destination anges. Mini-spec'ar och Data Dictionary skapas och modifieras med hjälp av vanlig tillgänglig text-editor.

Kontrollverktygen analyserar SA-beskrivningarna. De är fyra till antalet, ett för varje beskrivningstyp och ett övergripande. Sökning sker efter två möjliga typer av fel: "hårda" fel (t ex odefinierade datanamn) som indikerar att SA-syntaxen brutits; "mjuka" fel -- åsidosättande av tumregler som finns i metodiken t ex beträffande antalet processer i ett dataflödesdiagram. Upptäckta fel rapporteras till användaren som avgör hur de skall avhjälpas.

Formateringsverktygen skönskriver minispec'ar och innehåll i Data Dictionary samt ritnar dataflödesdiagram. Positioneringen av symbolerna styrs av användaren via "Edit DFD". Man har avstått från automatisk bildgenerering, eftersom man anser att positioneringen är mycket viktig för läsbarheten och man saknar god heuristik för detta.

"Clean-up"-verktygen verifierar och genomför de ändringar i mängden av SA-beskrivningar som blir konsekvensen av att man ändrar i en beskrivning. Observera att det här är fråga om sådana ändringar, vilka ej kräver något beslut från användaren.

Datorstödet är implementerat i Modula-2 och körs under UNIX V7. En Tektronix 4014 terminal används för interaktiv grafik.

Underlag för beskrivningen

/DEL-82/ och /KER-81/.

2.11. USE (User Software Engineering)

User Software Engineering är en metod för utveckling av interaktiva informationssystem. Med interaktiva informationssystem avses system som ger konversativ access till data.

USE och stödjande verktyg har utvecklats vid University of California och Vrije Universiteit, Amsterdam. Kommersiellt marknadsförd produkt av Interactive Development Environments, Inc. Ca 100 exemplar finns utplacerade, varav hälften hos industriella användare och hälften i akademisk miljö.

USE följer en fasindelad systemutvecklingsmodell från analys till operationellt system, men med starka inslag av prototyping.

Utvecklingsgången är mycket kortfattat denna:

1. Kravanalys.

Datamodellering, aktivitetsmodellering och analys av användare- och användningskrav och -egenskaper. För data- och aktivitetsmodellering finns inga egenutvecklade eller föreskrivna tekniker. För datamodellering har flera konceptuella datamodelleringstekniker använts t ex Chen's "Entity-Relationship"-modell /CHE-76/. För aktivitetsanalys har använts ISAC's A-grafer, /LUN-78/, och dataflödesdiagram enl "Structured Analysis" /DeM-78/, /GAN-79/. För ritning av dataflödesdiagram finns ett interaktivt grafiskt verktyg utvecklat kallat DFD ("Dataflow Diagram Editor").

2. Extern utformning.

Istället för att nu, vilket är brukligt i många "top-down"-ansatser, fortsätta med att detaljera systemets funktioner, definieras systemets externa interface ("outside-in"-ansats). För modellering av dialogen används speciella sk transitionsdiagram (modifierade och utvidgade "state-transition"-diagram). För att stödja skapandet av sådana diagram finns ett interaktivt grafiskt verktyg utvecklat, kallat TDE ("Transition Diagram Editor"). Detta kan också generera input till RAPID/USE, som är ett applikationsutvecklingssystem. Med detta kan en körbar prototyp av dialogen, som definierats i transitionsdiagrammen, åstadkommas.

3. Från dialogprototyp till färdigt system.

Flera alternativa vägar finns nu att fortsätta på: antingen att utvidga prototypen eller att överge den och fortsätta på mer konventionellt sätt med systemutformning och implementering i traditionellt programmeringsspråk. Ett tredje interaktivt grafiskt verktyg, som finns utvecklat och som är användbart i design-fasen, är SCE ("Structure Chart Editor"). Med detta kan man skapa och manipulera strukturgrafer enl "Structured Design" /STE-74/, /YOU-75/.

Flera datorbaserade verktyg, varav några nämnts ovan, finns utvecklade för att stödja utveckling av system enligt USE-metoden:

Troll/USE: Ett relationsdatabashanteringssystem med ett relationsalgebra-liknande interface. Detta innehåller operationer på element-, tuple- och relationsnivå. Möjlighet finns att definiera, namnge, och lagra procedurer av datamanipuleringsatser samt förse dem med formella parameterar. Procedurerna kan sedan under körning anropas med aktuella parametrar.

Troll/USE arbetar på så kallad "message-passing"-basis och används som bakgrundssystem till bl a det skärmorienterade TBE-verktyget (se nedan) och data dictionary-delsystemet i DFE-verktyget (se nedan).

RAPID/USE: Ett applikations- eller prototyputvecklingssystem innehållande två huvudkomponenter:

- En "Transition Diagram Interpreter (TDI)" för exekvering av beskrivningar av användar/programdialoger upprättade med transitionsdiagramtekniken.
- En så kallad "Action Linker", som länkar TDI till bearbetningsrutiner skrivna i något högnivåspråk (C, FORTRAN 77, Pascal etc).

RAPID/FORM: Ett "what-you-see-is-what-you-get"-orienterat formulärbaserat program för uppdatering av och utsökning i databaser. Är under utveckling.

TBE (Troll/USE Browser and Editor): Är ett skärmorienterat redigerings- och "browser"-program för Troll/USE-databaser. Här kan man, på ett enklare sätt än att arbeta direkt med Troll/USE, skapa och ta fram existerande relationer, skumma igenom relationer en tuple i taget, selektera ut delmängder av tupler i relationer, lägga till, ändra i eller ta bort tupler, etc.

DFE (Dataflow diagram Editor): Ett interaktivt grafiskt verktyg för att rita och redigera dataflödesdiagram enligt den notation som följs av Yordon Inc. Fördefinierade symboler placeras ut och förbinds med pilar via musteknik. Vidhängande pilar följer med när symboler eller grupper av symboler flyttas. Processymboler numreras automatiskt. Symbolerna kan namnges, kommenteras med intilliggande text, som kan döljas eller visas, och ges längre textmässiga beskrivningar med hjälp av texteditor. Processer kan detaljeras i grafer på lägre nivå och man kan direkt hoppa mellan nivåer. Verktyget utför dock ingen syntaxkontroll,

vare sig med avseende på hur man kopplar symboler med varandra eller med avseende på inkonsistenser i hierarkin av grafer. Ytterligare exempel på faciliteter är att grafer kan "scrollas", förminsas och centreras.

TDE (Transition Diagram Editor): Med denna editor skapas och redigeras USE speciella transitionsdiagram för beskrivning av dialoger. Faciliteterna för grafredigering och hantering är i stort desamma som de i DFE. I transitionsdiagrammen representeras tillstånd av noder och transitioner mellan tillstånd av grenar. Så kallade "action"-noder kan placeras på transitionsgrenarna för att indikera olika bearbetningar (t ex repr av procedurer av Troll/USE datamanipuleringsatser). Med TDI (se ovan) kan körbara program skapas utifrån diagrammen. Genom experiment med dessa kan dialogen förfinas och fullständig. Exekveringen av prototypprogrammen kan animeras. Alltefter som exekveringen fortskrider blinkar uppnådda noder i aktuellt transitionsdiagram och till noden knutna meddelandetexter presenteras.

SCE (Structure Chart Editor): Innehåller likartade faciliteter som DFE för skapande och redigering av strukturdiagram enligt "Structured Design", där utformningar av program beskrivs hierarkiskt. Input- och output-angivelser kan göras för de olika komponentrelationerna, som numreras automatiskt.

Samtliga USE-verktyg är implementerade under UNIX operativsystem och körbara på Sun och VAX-datorer. De tre grafiska stöden är körbara på Sun arbetsstationer och använder Sun's fönsterhanteringssystem.

Underlag för beskrivningen

/USE-85a/, /USE-85b/, /USE-85c/ och /WAS-85/.

Kapitel 3

BESKRIVNING AV RAMATIC

RAMATIC kan ses som ett datorbaserat, grafiskt orienterat modelleringsstödssystem.

De första stegen i utvecklingen av RAMATIC togs 1980 inom SYSLABs regi (SYSLAB = Systemutvecklingslaboratoriet) i Göteborg. I och med uppstarten av SISU, lyftes merparten av utvecklingsarbetet över dit, och utgör nu ett viktigt inslag i SISU Göteborgs och därmed programområde 3:s verksamhet.

Fortsatt, mer teoretiskt och forskningsmässigt inriktat arbete kring RAMATIC, pågår dock fortsatt inom SYSLAB-G.

I den beskrivning av RAMATIC som följer, ges först en ytterligare bakgrund till utvecklingen och några viktiga intentioner med stödet anges. Därefter presenteras stödets yttre egenskaper, dess funktioner. En komprimerad sammanfattning av några viktiga egenskaper följer och avslutningsvis indikeras de planer, på kort och lång sikt, som finns för den vidare utvecklingen.

Den prototypversion av RAMATIC, som nu föreligger, beskrivs ytterligare i manual- och "tutorial"-form i /RAM-85a/, /RAM-85b/ och /RAM-86/. Andra referenser som kan ges är /DAH-84/, /JOH-83/ och /JOH-84/ liksom /ERI-83a/, /ERI-83b/ och /TOR-84/.

3.1. Bakgrund till utvecklingen av RAMATIC

Ursprungligen behövde vi ett grafiskt baserat datorstöd till beskrivningsteknikerna i den dåvarande beskrivningsteknikutvecklingen inom SYSLAB.

Där aktualiserades att flera beskrivningstekniker skulle kunna samverka med varandra. Vi ville undersöka vilka potentialer i form av modelleringseffekter detta skulle kunna innebära. Därifrån växte fram att motsvarande hanteringsmöjligheter skulle finnas i datorstödet.

Senare utvidgade vi hela datorstödsutvecklingen så att vi ville skapa ett datorstöd som skulle vara generellt gentemot olika typer av be-

skrivningstekniker. Detta medförde en mycket stor höjning av ambitionsnivå för utvecklingen genom den design som kom att krävas. Det var samtidigt en intressant forskningsuppgift.

En annan viktig intention var att göra verktyget så oberoende som möjligt både av operativsystem och av grafisk utrustning (tillverkare). Detta ledde oss in på de standardiseringsaktiviteter som var, och är på gång inom grafikområdet.

Efterhand uppkom insikten om, att för att vi skall kunna stödja överskådliga modeller och modeller med fler uttrycksmöjligheter i modellerandet, så behövs dels grafik men också möjligheter för datorstödet att ta hand om uttryck som inte alltid är lämpade att representera grafiskt. På det sättet måste datorstödet både ha goda grafiska egenskaper för inmatning, manipulation, presentation, etc., men det måste också vara 'intelligent' i olika avseenden, så att det kan ge analysstöd och kan göra olika former av kontroller av modellstrukturerna. Delarna som rör analysstöd och kontroller är minst lika svåra delar som de grafiska. Det är i utsöknings-, analys- och kontrolldelarna som mycket av framtidens modelleringsstöd ligger, bedömde vi.

En viktig bas för att kunna utveckla mer djupgående former av utsöknings-, analys- och kontrollmöjligheter samt för att kunna vidareutveckla verktyget i olika andra riktningar, är den uppdelning i en strukturell databas (Conceptual Data Base, CDB) och en grafisk databas (Spatial Data Base, SDB), som gjorts i RAMATIC. Detta är också en bas för generaliteten i verktyget.

Efter ett tag utvecklade vi idén och egenskapen att automatiskt, från en modellstruktur (oavsett bild), kunna generera grafer att visas på skärmen. Denna egenskap har visat sig mycket användbar, både när det gäller utsökningar och att överhuvudtaget grafiskt illustrera en struktur. Strukturen kan alltså vara inmatad på annat sätt än i grafisk form. Detta är vi fortfarande ganska ensamma om, när det gäller denna typ av verktyg.

3.2. Några intentioner för RAMATIC

Innan vi går in på de olika existerande och planerade egenskaperna för verktyget, så vill vi ta upp några viktiga principer och grundintentioner som använts vid utformningen av det.

Grafisk inmatning och presentation

Ett grafiskt baserat, pekorienterat "interface" för att aktivera funktioner i verktyget är centralt. Detta ger en enklare och vänligare dialog än kommandointeraktion för ovana användare. Det är dock troligt att vi med tiden också skapar ett mer specialistorienterat språk, som mer erfarna modellerare kan använda sig av för att snabba upp interaktionen.

Grafiska presentationsmöjligheter är av stor betydelse för att öka möjligheten att uppfatta, förstå och kommunicera underlag och resultat av modelleringsarbete. Det är också av stor betydelse för personer som inte är vana vid att abstrahera i modeller.

Ibland vill man dock ha modellinnehållet presenterat i text- eller tabellform och dessutom är vissa resultat inte speciellt lämpade att presentera i grafisk form. Därför inrymmer datorstödet också vissa faciliteter för presentation i text- och tabellform.

Generalitet avseende beskrivningstekniker

Datorstödet är konstruerat så att olika beskrivningstekniker kan definieras för det. Datorstödet, som på detta sätt är beskrivningsteknikoberoende, lär genom definitionen känna de beskrivningstekniker man vill använda. Det innebär att man dels definierar hur de aktuella symbolerna skall se ut och vilken betydelse de skall ha, dels definierar de syntaxregler som skall gälla för att koppla de olika symboltyperna till varandra.

Faciliteter håller nu på att implementeras, som skall möjliggöra regeldefinition så att man t ex i en graf skall kunna hålla reda på att en viss symboltyp har minst en ingående och en utgående "relation" eller "flöde". Detta kallas för grafsyntax på bildnivå. Dessutom håller vi nu på att implementera faciliteter som innebär, att man skall kunna definiera de olika kopplingar mellan beskrivningstekniker som man vill

att datorstödet på olika sätt skall kunna stödja.

Grupparbetsstation

Grupparbetsstations-konceptet är viktigt för att man skall göra det möjligt för en hel grupp att arbeta med verktyget samtidigt. I princip skall vem som helst i gruppen kunna göra förändringar i någon graf, samtidigt som de övriga i gruppen kan följa med. En förutsättning för detta är att skärmarna måste ha en viss storlek och upplösning.

Aktivt arbetsinstrument

Ett annat viktigt mål är att verktyget skall kunna följa en verklig grupps kommunikation i olika analys- och beskrivningsfaser. Det innebär att skisser och beskrivningar "noteras" direkt i verktyget så att man slipper dokumentera separat efter arbetspasset. Detta ställer höga krav på prestanda.

Flera bilder samtidigt

Möjligheten att se flera bilder genom att använda flera skärmar eller flera fönster, är RAMATIC fortfarande ganska ensamt om när det gäller stöd för modellering.

3.3 Yttre egenskaper

3.3.1. Definition av modelltyp

Att definiera en modelltyp eller en beskrivningsteknik för RAMATIC innefattar: att definiera de symboltyper som skall användas, att definiera de syntaxregler som skall gälla för hur man får koppla symboler av olika typer till varandra samt att definiera vilka av de operationer som RAMATIC tillhandahåller som skall användas. Vidare skall de enskilda menyernas exakta utseende anges liksom deras anropsstruktur.

Symbolerna kan skapas på ett fritt sätt genom användande av det speciella symboldefinitionspråket (se vidare avsnitt 3.4.2).

Symboler delas i RAMATIC in i "fria" och "kopplande" symboler. Etablering av en kopp-

lande symbol (gren- el pilsymbol) förutsätter att de fria symboler (nodsymboler), som skall utgöra dess ändpunkter, redan har etablerats. För olika beskrivningstekniker gäller i regel vissa restriktioner för hur man får koppla fria symboler till varandra med olika kopplande symboler. Syntaxregler av denna typ kan definieras med hjälp av ett speciellt syntaxdefinitionspråk (se vidare avsnitt 3.4.2).

För närvarande utvecklas funktioner också för kontroll av hela grafer ("grafsyntax på bildnivå"). Så t ex kan det i en viss beskrivningsteknik krävas, för att ett visst modelleringsobjekt skall anses vara fullständigt definierat, att det är associerat med ett visst minimiantal av andra objekt.

I den vidare utvecklingen av RAMATIC planeras ett sammanhängande och något mer lättanvänt språk för alla dessa typer av definitioner.

3.3.2. Definition av menystruktur

I RAMATIC kan man skapa sin meny-layout och sin menystruktur på ett mycket fritt sätt. De tillgängliga operationerna och definierade symboltyperna kopplas lätt till olika menyfönster, som sätts samman till menystrukturer passande till aktuell beskrivningsteknik och användargrupp. (Se vidare avsnitt 3.4.2.)

3.3.3. Skapande och manipulering av bilder

Funktionerna för inmatning och manipulation av grafer är, i huvudsak, indelade i två typer: bildfunktioner och symbolfunktioner. Bildfunktionerna påverkar en hel bild eller delar av en bild, medan symbolfunktionerna berör en, eller en grupp av utvalda symboler.

Nedan följer en kort beskrivning av vad som är möjligt att utföra i föreliggande version av RAMATIC (Version 1.0) avseende denna funktionsgrupp.

Manipulering av bilder

Hela bilder

För manipulering av hela bilder finns grundläggande funktioner som att initiera och namnge

en ny bild, att hämta fram en redan skapad och lagrad bild, att kopiera en bild och att radera en bild från databasen. Vid radering av en bild fås först en verifieringsfråga, så att man inte av misstag skall kunna ta bort bilder. Vid flitig flyttning och radering etc av symboler i en bild, kan ibland delar av bilden tillfälligt bli förstörda. Bilden kan då ritas om med hjälp av ett enkelt "refresh"-kommando.

En bild kan förstöras eller förminskas och man kan "scrolla" hela bilden i den riktning och omfattning man själv önskar. Detta är användbart om man vill göra bilden större än vad som rymms på själva bildskärmen.

När man placerar ut symboler i en bild, kan det vara svårt att lägga dem snyggt i jämna rader, vågrätt och lodrätt. För att underlätta detta, finns det en funktion som justerar hela bilden efter ett rutnät, som man själv kan definiera och bestämma storleken på.

Funktionen **placera** ("place") ritar upp en automatgenererad bild utifrån en användarritad bild. Den genererade bilden kan man sedan arbeta vidare med och finjustera på de punkter man inte är helt nöjd med resultatet. Utgångsbilden förstörs inte utan finns kvar som tidigare. Utplaceringsalgoritmen försöker lösa problemet med korsande pilar och lyckas i de flesta fall på ett godtagbart sätt. Om den bild vi utgår från är mycket komplicerad, kan dock den genererade bilden bli aningen rörig. Eftersom den automatiska bildgenereringsfunktionen arbetar direkt på den strukturella databasen (i fallet ovan alltså egentligen på den användarritade bildens strukturella representation i CDB), finns här potentialen att bildmässigt presentera strukturer, som inte behöver vara grafiskt inmatade.

Till varje bild man skapar, finns det också möjlighet att göra en verbal beskrivning. Denna beskrivning är i fritt format och det finns för närvarande inga funktioner som antar någon struktur hos textmassan. När funktionen väljs, startas den text-editor upp, som man angett för körningen.

Delar av bilder

Förutom de funktioner som beskrivits ovan, finns det ytterligare ett antal funktioner på bildnivå. Dessa arbetar dock inte på hela

bilder, utan syftar till att, från en eller flera symboler i en eller flera redan existerande bilder, skapa olika utgångspunkter för nya bilder. Några av dessa funktioner utgörs av olika utsökningsfunktioner och beskrivs därför i avsnitt 3.3.4.

I en del beskrivningstekniker upprättas successivt alltmer detaljerade beskrivningar, genom att symboler på grov beskrivningsnivå förstoras i detaljeringsgrafer. För detta finns en funktion **förstora** ("enlarge"), som skapar utgångspunkten för en detaljeringsbild för den symbol som valts. Den symbol, som skall förstoras, representeras på detaljeringsbilden av en stor centrerad rektangel och till symbolen in- och utgående andra symboler läggs ut kring denna. Det är möjligt att i syntaxdefinitionen för beskrivningstekniken ange om de skall läggas ut horisontellt dvs till vänster och till höger om rektangeln eller vertikalt dvs över och under mittrektangeln. Med detta som utgångspunkt kan nu detaljeringsarbetet fortsätta. På detta sätt kan man bryta sig ner i flera nivåer och erhålla en koppling mellan bilderna.

Funktionen **delbild** ("select") gör det möjligt att välja ut ett antal symboler i en bild och erhålla en ny bild med enbart dessa symboler på. De utpekade symbolerna placeras i samma position som de har i utgångsbilden. Kopplingar som finns mellan utpekade fria symboler fås med endast om man selekterar dem. Denna funktion lämpar sig att använda om man är intresserad av att fokusera vissa symboler lite mer i detalj, men ändå ha kvar dem i samma position som i utgångsgrafan. Genom att välja ut endast ett fåtal symboler blir inte bilden så komplex och svårförståelig.

Funktionen **utvidga** ("extend") är snarlik funktionen **delbild**. Här selekteras också ett antal symboler ut från en eller flera bilder och en ny bild skapas med hjälp av dessa. De kopplingar som finns mellan utpekade fria symboler tas med. Skillnaden är här att den automatiska bildgenereringsfunktionen används för utplacering av symbolerna. Symbolerna kommer alltså inte att placeras på samma plats som de fanns på i de gamla bilderna. Bilden går sedan att arbeta vidare med.

Denna funktion är användbar t ex om vi arbetar med en bild som beskriver objekt och deras relationer och vill visa i en annan bild de attribut som tillhör ett visst objekt. Då kan vi göra

en utvidgning på endast en symbol och fortsätta att lägga till de attribut som är aktuella för objektet.

Manipulering av symboler

Enstaka symboler

Att lägga till nya symboler görs enkelt genom att välja någon av de fördefinierade symboler som finns i menyn och peka på den position i vilken symbolen skall placeras. För kopplande symboler (pilar) anges start- och slutsymbol. Uppsättningen av fördefinierade symboler är olika beroende på vilken beskrivningsteknik som används.

Symboler som placerats i bilden går att flytta eller ta bort. De pilar som är kopplade till symbolerna följer med vid flyttning eller tas automatiskt bort om någon av ändsymbolerna tas bort. För varje symbol kan man ange ett namn och/eller en identifierande beteckning (ett Id). Dessa bör vara unika och om ett visst namn eller Id tidigare satts på en symbol kommer en varningstext som talar om att namnet eller Id:et redan finns i en annan symbol. Dessutom får vi reda på i vilka bilder denna symbol finns. Varje symbol har ett antal attribut som definierar dess storlek, färg, linjetjocklek, etc. Dessa attribut kan man ändra på om man vill. Vidare går färgen och storleken på den text, som man har lagt in i symbolen, att ändra. En symbol kan roteras efter valt gradtal, eventuell text som finns i symbolen roteras också med samma gradtal. Om man vill ändra storleken på en symbol går det bra. De flesta symboler har relativ storlek 1 i bilden, men är förminskade till halv storlek i menyarean för att inte ta för stor plats.

Man kan definiera ett rutnät i bildarean som gör det möjligt att lägga symboler i snygga rader, vågrätt och lodrätt. Denna funktion finns även som bildfunktion men justerar där en hel bild, till skillnad från denna som endast placerar in en vald symbol.

Precis som man vill kunna beskriva en bild, vill man även ha möjlighet att beskriva en enskild symbol verbalt. Detta utförs på samma sätt som för bilder.

Grupper av symboler

Förutom att behandla enskilda symboler, kan man ibland ha behov av att arbeta med grupper av symboler. Om vi t ex vill kopiera ett antal symboler till flera bilder, är detta mycket användbart.

En grupp definieras genom att de symboler som skall ingå i gruppen pekas ut. En grupp kan flyttas genom att man pekar på en symbol som ingår i gruppen och därefter anger dennas nya position. För att kopiera en grupp måste den först ges en identifierare. Grupper som man inte längre har användning för kan upplösas.

Specialritningsfunktioner

De symboler som finns tillgängliga när man skapar grafer i en viss modelltyp är fördefinierade och går inte att ändra under körning. Ibland vill man dock i sin bild lägga in andra symboler, figurer och texter. Man kanske t ex i förtydligande syfte vill ringa in en del av bilden med en streckad linje, lägga in en rubriktext, etc. För detta ändamål finns en sk "Draw"-facilitet med vars hjälp man kan rita olika typer av linjer och streck, öppna eller slutna kurvor, polylinjer eller polygoner, och lägga in texter var som helst i bilden, texter som inte är kopplade till någon speciell symbol. Specialritningsfunktionen finns alltid tillgänglig oavsett vilken beskrivningsteknik som man använder sig av. De specialritade symbolerna ges endast en representation i den grafiska databasen (SDB), eftersom de inte har någon för verktyget definierad innebörd.

3.3.4. Utsökning och presentation

Goda utsöknings- och presentationsmöjligheter kan vara till stor hjälp i analysarbetet. Speciellt accentueras detta behov när antalet grafer blir stort. Till stor del är det fråga om att skapa överblick. Man kan skapa tabulära sammanställningar av vad som finns i graferna, men man kan också skapa grafiska sammanställningar genom att generera en bild som projicerar det man "frågat efter". För det senare används den automatiska bildgenereringsfaciliteten. För ett exempel, se "närhetsfunktionen" ("adjacent") nedan. Här skapas en gra-

fisk sammanställning över bildgränser. Man går då inte igenom vad som finns i graferna i sig (den grafiska databasen) utan vad som finns i den bakomliggande strukturbeskrivningen (CDB).

Idag har vi ett utsökningspråk som använder en UNIX-liknande syntax. Ett nytt "fill-in"-språk mer anpassat till datorovana personer planeras.

Nedan beskrivs de utsökningsfunktioner som finns i Version 1.0 av RAMATIC. Inför kommande versioner kommer mer tid att läggas på utveckling av utsökningsfunktioner. Det gäller också utsökningsfunktioner som avser hur olika beskrivningstekniker hänger ihop och samverkar.

Utsökningsfunktioner

Lista ("list") listar de bilder som finns lagrade i databasen. Om endast **lista** anges utan parametrar får man alla de bilder som skapats i den beskrivningsteknik som man för tillfället arbetar i. Man kan även specificera speciella sökuttryck för att få de bilder vars namn matchar dessa uttryck. Man kanske använder speciella prefix eller suffix i bildnamnen och kan använda dessa vid utsökningen. Det är även möjligt att få listat namnen på bilder gjorda i en annan beskrivningsteknik än den man för tillfället arbetar inom. Även för namn på beskrivningstekniker kan ett sökuttryck anges.

Om man vill veta vilka namn, Id:en eller symboltexter som använts använder man **finn** ("find"). Här anges dels typ, dvs om det är namn, Id:en eller symboltexter man söker efter, dels ett sökuttryck. Som resultat fås alla namn som matchar sökuttrycket och som är av den typ som vi angett. Dessutom får vi reda på i vilka bilder som dessa namn, Id:en eller symboltexter finns. Funktionen letar reda på allt som matchar, oavsett vilken nivå som namnet finns på. Detta är mycket användbart när vi arbetar med att framställa grafer och behöver veta vilka namn eller Id:en som använts i redan ritade symboler. Genom **finn** kan vi få tips på namnsättning utifrån redan satta namn och vi kan även se vilka namn och Id:en vi använt i vissa speciella grafer.

Närhet ("adjacent") söker ut och sammanställer i en graf en utvald symbols grannar i det antal led som vi anger. Grafen genereras med den

automatiska bildgenereringsfaciliteten. Den utvalda symbolen placeras i centrum av bildarean och dess samlade omgivning "blommar ut" från denna så långt man önskar. Observera att funktionen arbetar över bildgränser och sammanställer olika delomgivningar, som ursprungligen specificerats i olika skapande bilder.

Som en vidare utvidgning av denna funktion kan man tänka sig att man dessutom skulle kunna ange vilken eller vilka typer av samband och typer av grannar som skall ingå i en önskad omgivning.

3.3.5. Kontrollfunktion

De viktigaste typerna av kontroller av skapade modeller är: syntaxkontroll dvs hur symboler får kopplas till varandra, och syntaxkontroll på bildnivå, som t ex att en viss symbol måste ha minst en "inkommande" och minst en "utgående" symbol i en graf för att den skall vara korrekt.

Den senare typen av kontroll kan ju bara vara aktuell att utföra efter det att man anser att grafen har blivit färdig. Innan dess måste den betraktas som en "skiss" som ännu är ofullständig.

Andra typer av kontroller har mer karaktären av analysstödsfunktioner. Med det menas att datorstödet inte med säkerhet kan avgöra vad som är rätt och fel, utan datorstödet kan ge indikationer och göra påpekanden, som kan ge anledning för "modelleraren" att vidta justeringar eller kompletteringar av olika slag. Exempel är här kontroller av namn, delar av namn etc., för att dels hitta direkta likheter i namnsättningen, dels indikera att ett modellerat fenomen kanske är det samma som eller bör distingeras från något annat fenomen för att inte kunna förväxlas.

Kontrollfunktioner som inriktar sig på kopplingar mellan beskrivningstekniker kommer också att tillföras framgent.

3.3.6. Hjälpfunktion

För samtliga funktioner och kommandon i RAMATIC kan man, var man än befinner sig i interaktionen, be om hjälp för att få information om vilka möjligheter man har.

Vi planerar också att lägga in hjälp- och instruktionsinformation, som anger hur man använder en viss beskrivningsteknik på ett bättre sätt dvs som ger "modelleraren" råd och anvisningar om hur det egentligen är meningen att man skall använda beskrivningstekniken samt som förklarar vad de olika modelleringsbegreppen och symbolerna betyder. Detsamma gäller hjälp och instruktion på metodnivå.

3.3.7 Hantering av multipla skärmar

Möjligheter att arbeta på flera skärmar eller i flera separata grafiska fönster finns redan idag och kommer i Version 2.0 av RAMATIC. Skall man ha flera fönster på samma skärm krävs en något större upplösning.

Man arbetar i ett av fönstren och använder de övriga för att samtidigt titta på andra grafer eller sammanställningar. Hopp mellan fönstren sker med speciellt kommando.

Uppenbara fördelar med detta arbetssätt är t ex att man kan se översiktsgrafer och detaljgrafer samtidigt och att man parallellt kan ha grafer från flera olika beskrivningstekniker framme, där graferna var och en belyser olika aspekter av den "verklighet" man modellerar.

3.3.8. Hardcopyfunktion

Det är viktigt att kunna ta ut grafiska modeller och annat modellinnehåll på papper och på overhead-film, inför presentationer, rapporteringstillfällen, etc. Ibland vill man kanske ta med sig vissa resultat hem för att fundera vidare.

En sk driver har utvecklats för plotter och ett särskilt program som administrerar plotterkön har skrivits. Arbete med en driver för matris-skrivare pågår.

3.3.9. Dokumentgenereringsfaciliteter

Med hjälp av en dokumentationsfacilitet, något som idag inte finns implementerat i RAMATIC, skulle man, utifrån en dokumentationsstruktur som användaren själv definierar, kunna låta datorstödet "plocka ihop" en önskad dokumentation över uppnådda modelleringsresultat. I denna skulle kunna ingå grafer,

formulärstyrda dokument, tabeller, fri text-beskrivningar, etc. Vidare skulle dokumentationen automatiskt kunna sidnumreras, förses med innehållsförteckning och korrekta sidhänvisningar m m.

3.3.10. Backup- och recovery-funktion

I samband med uppstart av RAMATIC tas alltid en backup-kopia av aktuell databas. Under körning loggas samtliga transaktioner i en speciell logg-fil. Efter ett onormalt avbrott startar man upp RAMATIC igen och som första åtgärd återskapar RAMATIC automatiskt databasen från backup-kopian och logg-filen.

3.3.11. Fri text-hantering

När man skriver fria texter, t ex vid verbal beskrivning av en bild eller en symbol, så startas automatiskt den UNIX text-editor upp som man har angett för sin installation. Om denna inte skulle vara ens "favorit-editor" finns det möjlighet att byta till denna innan körningen startas upp.

3.3.12. Auktorisationsfunktion

I nuläget finns ingen auktorisationsfunktion implementerad i RAMATIC, men behov av en sådan finns och planeras implementeras framgent.

Med hjälp av en sådan facilitet skulle man t ex kunna definiera och övervaka tillgång till olika bilder och modelleringsresultat och vilken typ av operationer som får utföras, t ex att ett delprojekt inte får manipulera ett annat delprojekts modeller, men man får titta på dem. I andra fall får andra än upphovsmännen över huvudtaget inte se modellerna.

3.3.13. Helt fri ritning av bild

När man har tillgång till ett datorstöd med så pass kraftiga grafiska faciliteter som RAMATIC, så är det ofta intressant att kunna använda det för att göra grafiska beskrivningar utan att arbeta i någon speciell beskrivningsteknik. Man vill t ex ha tillgång till alla de olika symboler som finns definierade utan att för den skull behöva arbeta efter en viss syntax.

Det finns därför ett särskilt beskrivningsalternativ kallat "FREE", där man kan rita och använda tillgängliga symboler helt fritt.

3.3.14. Ångerfunktion

Ibland kan genomförandet av en operation medföra att mycket arbete går förlorat i modelleringstödet, som t ex att "sudda" ett objekt som har många relationer. Alla kopplande symboler (pilar) som går till och från objektet kommer ju då också att suddas ut.

En mycket användarvänlig funktion att tillgripa i ett sådant läge är en så kallad ångerfunktion, som innebär att operationen "görs ogjord" och tidigare status återställs. En sådan ångerfunktion finns implementerad i RAMATIC.

3.3.15. Val av nationellt språk

Undersökningar har visat att man i Sverige trots allt vill ha datorstöd som "talar svenska". Motsvarande gäller rimligen också för andra nationer.

De olika hjälp- och meddelandetexterna ligger därför i separata filer som kan bytas ut, beroende på vad användarna helst vill använda för språk.

3.3.16. Formulärhantering

Ibland vill man kunna mata in modellinnehåll via formulär. Presentation och sammanställning av modellinnehåll kan också vara lämpliga i tabulär- eller formulärform.

Faciliter för att hantera formulär (definition, ifyllnad, "läsning" av fält etc.) kommer successivt att tillföras.

Om man har de funktioner som RAMATIC erbjuder kan man sedan presentera formulärinmatade strukturer i grafisk form, om man så föredrar.

3.3.17. Mail- och konferensfunktion

Med tiden planerar vi att göra det möjligt att skicka modellinnehåll och resultat via de ordinarie "electronic mail"-system som ofta finns

åtminstone i de större organisationerna. Man vill t ex snabbt kunna få synpunkter från någon referensperson på ett utkast som man gjort. Medverkansmöjligheter och intresse borde kunna ökas på detta sätt.

3.4. Inre egenskaper

I detta avsnitt beskrivs översiktligt några egenskaper hos RAMATIC ur en mera intern synvinkel. De lösningar, som har valts för implementering av verktyget, är naturligtvis relaterade till de krav, som har ställts på RAMATIC. Några sådana grundläggande krav är:

- generalitet
- flexibilitet vid påbyggnad och förändring av verktyget
- enkel användning även för datorovana personer
- möjlighet till analys och kontroller av skapade modeller
- programportabilitet.

Generalitet innebär i det här sammanhanget, att datorstödet i så hög grad som möjligt skall vara oberoende av systemutvecklingsmodell. Dessutom skall det för ett visst företag, som använder verktyget, vara enkelt att tillföra nya och förändra existerande modelltyper.

Som framgår av beskrivningen nedan, så har de uppräknade kraven i hög grad kunnat tillgodoses.

3.4.1. Uppdelning i SDB och CDB

RAMATIC är ett grafiskt datorstöd, där man kan rita grafer med symboler givna för en viss modelltyp. Dessa grafer kommer att ha en hel del grafiska data som exempelvis symbolernas utseende, placering och färg. Utöver dessa grafiska data lagras i RAMATIC information om grafernas strukturer och andra egenskaper, som inte är av grafisk karaktär.

Eftersom det finns starka skäl att hålla dessa två typer av information åtskilda, så har den lagrade informationen i databasen delats upp i två delar, nämligen dels i en grafisk del (SDB = spatial data base), dels i en strukturell del (CDB = conceptual data base). I CDB beskrivs varje system (modell) med de två begreppen noder och kopplingar mellan noder. En sådan

modell kan innehålla ett stort antal noder och kopplingar för att beskriva någon modellerad "verklighet". Antalet noder kan då vara för stort för att enkelt åskådliggöras på en grafisk skärm eller plottas på ett vanligt papper. Man kan därför betrakta de grafer, som skapas med hjälp av verktyget och lagras i SDB, som grafiska projektioner av modellen i CDB. En nod i CDB kan återfinnas i 0, 1 eller flera grafer i SDB.

Med denna datastruktur är det enkelt att skapa andra typer av projektioner från en given CDB-struktur, exempelvis beskrivningar i tabell- eller formulärform.

Uppbyggnaden av dels graferna (projektionerna) i SDB och dels den globala strukturen i CDB sker normalt sett samtidigt. Däremot kan det göras på många olika sätt. Det vanligaste sättet är naturligtvis, att man ritar separata grafer på skärmen, varvid både SDB och CDB uppdateras. I efterhand kan man "identifiera" vissa symboler i olika grafer som samma nod i CDB. Med "selektering" kan man skapa en delgraf av en redan existerande graf. Med "utvidgning" väljer man ut en eller flera symboler från en eller ett fåtal grafer och låter verktyget självt skapa en ny graf utifrån dessa symboler. Denna nya graf ändras och byggs sedan vanligen på med nya symboler. Om man tar en "kopia" av en graf, så dubblas följaktligen SDB-strukturen med avseende på grafen, medan CDB-strukturen förblir oförändrad. Däremot påverkas kopplingen mellan CDB och SDB. Med "förstoring" kan man ange, att en symbol finns i en förstörd skepnad i en annan graf, exempelvis vid beskrivning av en funktion med sina delfunktioner.

3.4.2. Definition av modelltyper och interaktion

Som tidigare nämnts så har RAMATIC ingen inbyggd kunskap om en viss modelltyp eller beskrivningsteknik, utan står till tjänst med ett antal allmänna operationer för att manipulera grafiska beskrivningar. Däremot interagerar RAMATIC med en användare på ett fastlagt sätt genom menyval av operationer samt en hierarki av menyer med likartade operationer på en viss nivå. I övrigt lämnas öppet för en användare att utnyttja detta och skraddarsy beteendet hos RAMATIC enligt egna önskemål och för detta krävs möjlighet att definiera:

- symboler
- menyer
- samt gärna även syntaxen för en modelltyp.

Vi ska inte gå in i detalj på hur detta sker utan endast lämna en översiktlig beskrivning.

Symbolbeskrivning

Symboler är godtyckligt formade grafiska objekt. En symbol kan "plockas" från menyn och placeras i den aktuella grafen. En symbol definieras i det enklaste fallet genom att beskriva dess utseende i form av ett antal linjer. Start- och slutpunkter för dessa linjer anges relativt en referenspunkt ("tyngdpunkt") i symbolen. I övrigt kan anges grafiska attribut till symbolen såsom typ av linjer, färg på linjer, om symbolen skall fyllas med färg eller ej, etc.

Menybeskrivning

En meny kan ses som ett antal rader av text eller symboler. Menybeskrivningen består i att för varje sådan rad ange

- den text eller den symbol som skall visas i menyn
- den operation som skall utföras när menyraden väljs
- om operationen innebär att en ny meny skall visas måste namnet på denna meny anges.

På detta sätt kan en godtycklig menyhierarki beskrivas. RAMATIC håller reda på den väg man tagit i menyträdet så att man kan återvända till föregående nivå när man är klar på en lägre.

Syntaxbeskrivning

För varje beskrivningsteknik finns oftast angiven en syntax, dvs ett godkänt sätt att koppla samman symboler. Genom att beskriva syntaxen för RAMATIC kan en omedelbar kontroll utföras så att felaktiga sammankopplingar uppmärksammas.

En syntaxbeskrivning består av ett antal delar. Varje definierad symbol har ett (modelltypsberoende) namn. Detta symbolnamn ges ett modelltypspecifikt namn som senare används i syntaxbeskrivningen (och kan använ-

das i framtida mera utförliga kontroller och ut-sökningar på modellinnehåll). Sedan anges vilka symboler som är s k **fria** symboler, dvs som kan existera utan kopplingar till andra fria symboler, samt vilka som är s k **kopplande** symboler, dvs som måste vara kopplade till två fria symboler för att kunna existera. Slutligen anges mellan vilka fria symboler det är tillåtet att koppla en kopplande symbol.

Meddelanden på olika språk

Till sist kan nämnas att meddelanden som RAMATIC använder finns lagrade separat från programkoden. Det är alltså möjligt att skapa en meddelandebeskrivning i ett godtyckligt språk. När man startar RAMATIC får man meddelanden i ett installationsberoende standardspråk eller i ett språk valt för just den körningen. För närvarande kan meddelanden fås på svenska eller engelska.

Notera att menytexter ej finns bland meddelandertexterna. På motsvarande sätt som för texter måste flera menybeskrivningar (och eventuellt symbolbeskrivningar) göras, en för varje språk.

3.4.3. Ritstödsfaciliteter

Som framgick av syntaxbeskrivningen ovan, så indelas symbolerna i fria symboler och kopplande symboler. De fria symbolerna, som kan placeras ensamma i en graf, består normalt av en sluten figur som en rektangel eller en cirkel, medan de kopplande symbolerna är pilar av olika slag och måste förbindas till/från fria symboler för att kunna ritas ut.

Fria symboler och kopplande symbolers utseende definieras i den ovan nämnda symbolbeskrivningen. För pilar kan olika typer av pilspetsar och pilslut anges. Pilar kan också definieras med dubbla streck med eller utan färgutfyllnad.

De fria symbolerna kan placeras fritt i en graf eller justeras i förhållande till ett valt rutnät. Vid uppritning av en pil behöver man endast peka på från- och till- symbolen, varvid en pil ritas ut automatiskt mellan de båda symbolernas kanter. Detta sker genom att det först skapas en pil från/till de två symbolernas referenspunkter och därefter klipps pilen vid symboler-

nas kanter. Det är också möjligt att ange ett antal mellanliggande punkter genom vilka pilen skall passera antingen med raka streck eller med en anpassad kurva.

Om man flyttar en symbol, en grupp av symboler eller anpassar hela grafen till ett rutnät, varvid de fria symbolerna flyttas, så ritas alla pilar om automatiskt enligt samma klippringsförfarande.

Med RAMATIC är det möjligt att generera nya grafer utifrån en lista av CDB-noder. Dessa noder skall vara av typen fria symboler. Pilar ritas ut automatiskt från de kopplingar, som existerar i CDB-strukturen mellan de angivna noderna. Den algoritm, som används för utplaceringen av symbolerna finns beskriven i /DAH-84/, och skall inte återges här. Det bör också påpekas, att denna rapport är litet föråldrad, när det gäller de flesta andra implementeringsaspekter.

3.4.4. Programportabilitet

Vid val av programvara och programspråk för utveckling av RAMATIC har kravet på programportabilitet varit vägledande. All egen programvara är skriven i programspråket C.

Vid val av grafikprogramvara har vi försökt att hålla oss till standards. Vid tiden för den första implementeringen av RAMATIC var CORE en stark standardkandidat, varför ett CORE grafikpaket valdes /COM-79/. Fortfarande används CORE i form av SunCore för Sun-maskiner. Så fort GKS har blivit tillgängligt från Sun, så kommer RAMATIC att konverteras till GKS, den grafikprogramvara som ju har blivit standard /COM-84/.

Som databashanterare används databashanteraren till det associativa relationsdatabas-systemet CS5 /JAN-84/. Valet av CS5 har möjliggjort en enkel och flexibel lagring av data, vilket är betydelsefullt på grund av den starka förändring som RAMATIC hela tiden genomgår. Det pris man får betala är att CS5 är något långsamt. Däremot är gränssnittet mot databashanteraren i CS5 definierat i en speciell modul. Detta gör det enkelt att vid behov ersätta databashanteraren mot någon effektivare genom att helt enkelt byta ut den existerande modulen mot en ny.

3.5. Framtida egenskaper

3.5.1. Egenskaper för de närmaste åren

Koppling mellan beskrivningstekniker

Som tidigare påpekats så kommer funktioner att tillföras som gör det möjligt att skapa referenser mellan "fenomen" i olika modelltyper. Dessa referenser kan tilldelas olika betydelser.

Bakgrunden till detta är, att allt fler metoder observerar att olika modelleringstekniker kan användas i kombination och ge intressanta ytterligare analysmöjligheter. Det kan gälla såväl att modelleringstekniker används parallellt som att modelleringstekniker i en senare utvecklingsfas tar över resultat från andra modelleringstekniker som använts i en tidigare fas.

Ett exempel på detta är att en informationsmängd (ett informationsflöde) i en viss beskrivningsteknik kan ha referenser till en "objektmodell" genom att termerna i mängden kan referera till motsvarande attribut i objektmodellen.

Kontroller mellan beskrivningstekniker

Genom att ovanstående kopplingar görs, så kan också verktyget fullgöra olika typer av kontroller enligt dessa kopplingar, t ex att alla termer finns definierade i objektmodellen (eller i någon av objektmodellerna).

Historik

En annan funktion, som bl a en del personer i intressentföretagen tryckt på, är att kunna komma tillbaka till hur en viss graf såg ut vid en tidigare tidpunkt. Med andra ord att kunna hålla reda på historik.

Inom ramen för en sådan funktion skulle man också kunna hantera de olika alternativ, som kan växa ut från ett visst läge på en graf. Detta kan bl a vara aktuellt när en beslutsgrupp vill ha ett antal olika möjliga lösningar att välja bland.

Expansion av symbol

Expansion av en symbol kan illustreras genom följande exempel. Antag att ett objekt har en mycket stor mängd attribut kopplade till sig. Grafen kommer att belastas hårt av dessa attribut och kan bli oöverskådlig. Attributen kan då göras osynliga när man normalt kallar upp grafen och uppenbarar sig först när man begär att få se dem.

Export-/importfunktion för modellinnehåll

Ett format för att kunna föra över modellinnehåll till t ex ett dictionary är en viktig komponent för att man skall kunna skapa ett samspel mellan olika typer av verktyg för modellering.

Det kan gälla verktyg för modellering av typen RAMATIC, verktyg av dictionary-karaktär och enklare skisseringsverktyg som kan köras på små maskiner.

Överföring från dictionary till modelleringsstöd kan t ex vara aktuellt när en utvecklingsinsats startar, och andra vägen kan vara aktuell när man vill koppla ihop de skapade modellerna med dem som redan finns i dictionaryet. (Ett dictionary används ju ofta för drift och förvaltning av informationssystem.)

Tolkning av vissa former av restriktioner

Eftersom vi strävar efter att utveckla RAMATIC till ett modelleringsstöd, som kan stödja också mer "formella" uttrycksformer, vill vi också skapa möjligheter att lagra olika typer av restriktioner för det som beskrivits i modellerna. Vi vill t ex kunna lagra härledningsregler, restriktioner för värdemängder för attribut, etc.

Stöd för att integrera grafer

Om man vill skapa ett verktyg som skall bidra med analysstöd, så är en annan tänkbar funktion stöd för integration av flera modeller till en ("view integration").

Här kommer problematik in som t ex bevakning av huruvida två objekt möjligen kan vara samma, eller om det kan röra sig om ett generiskt förhållande mellan objekten, etc. Kanske

kan jämförelse av objektens attributstrukturer ge vissa indikationer.

Mer utvidgad form av namngranskning

En annan form av analysstöd för att avgöra likheter bland objekt, relationer, funktioner, informationsflöden, etc., skulle kunna fås genom olika typer av granskning av namn.

Man kan också tänka sig mer avancerade former av textmatchning, så att datorstödet kan titta på olika delar av namn och på så sätt komma fram till indikationer på, och uppslag till, tänkbara likheter mellan olika modellerade fenomen. De olika delarna i namnen kan ges olika prioritet (ändelser ges t ex lägst prioritet), etc.

RAMATIC som fleranvändarsystem

I ett första steg är RAMATIC främst avsett för att användas som ett gruppstödsverktyg eller för en användare i taget.

Vi avser emellertid att göra det användbart för flera användare samtidigt. Det kan vara intressant t ex om flera projektgruppsmedlemmar, var och en för sig och från sina ordinarie arbetsplatser, vill ha möjlighet att komma åt modellerna i datorstödet.

3.5.2. Egenskaper på längre sikt

Modellering av skeenden och händelser

Idag finns det i RAMATIC möjligheter primärt att stödja beskrivning av statiska förhållanden i en verksamhet eller avseende ett informationssystem, förhållanden om t ex relationer mellan objekt eller flöden mellan verksamhetsfunktioner.

Då man har tillgång till mer välutvecklade modelleringsstöd, så är steget inte så långt till att också skapa möjligheter att stödja modellering som inkluderar beskrivning av dynamiska aspekter och beskrivning av skeenden. Som exempel kan nämnas beskrivning av hur ett ärende hanteras och förs vidare i en funktionsstruktur. Det ger upphov till att olika aktiviteter successivt utförs, beroende på hur aktuella villkor evalueras.

Generella kopplingar och överföringsmöjligheter till applikationsgeneratorer

Efter det att man byggt olika typer av modeller av verksamhet och informationssystem, vill man ofta också använda de prototypmöjligheter som finns i systemutvecklingsverktygen. Det innebär att man så snabbt som möjligt vill kunna få en körbar modell av den informationsmodell som man har skapat i modelleringsverktyget.

På samma sätt som vi bygger överföringsmöjligheter till t ex dictionary-system, kan vi också skapa möjligheter till att föra över modeller till s k applikationsgeneratorer och experimentsystemverktyg.

Koppling till elektronisk post- och konferenssystem

Som tidigare nämnts i avsnittet om olika funktioner i datorstödet, så planerar vi att kunna skicka modellinnehåll via en organisations ordinarie elektronisk post-system för att avge och motta synpunkter på modeller och delar av modeller. Det innebär att man även skall sända iväg bilder i dessa kanaler.

Kapitel 4

NÅGRA ERFARENHETER AV DATORSTÖD

I detta kapitel, som består av 2 delavsnitt, redovisas i det första några erfarenheter från användning av datorstödet EXCELERATOR inom Kommundata. Avsnittet är skrivet av Lars Axelsson, Kommundata. Det andra avsnittet behandlar användning av ett datorstöd för MBI. Erfarenheterna är sammanställda av Christer Fall, Programator.

4.1. EXCELERATOR - en mångsidig verktygslåda

Lars Axelsson, Kommundata

När vi 1984 letade efter datorstöd för att snabbt producera och förändra projektdokument, var vi illa ute. Problemet var, att våra modell- och metodidéer inom Kommundata inte hade upphöjts till vare sig internationell eller skandinavisk standard.

För oss är grafer centrala. Men våra grafer är en mix av olika beskrivningsätt. Till slut hittade vi ett generellt och kraftfullt verktyg, EXCELERATOR från InTechTM.

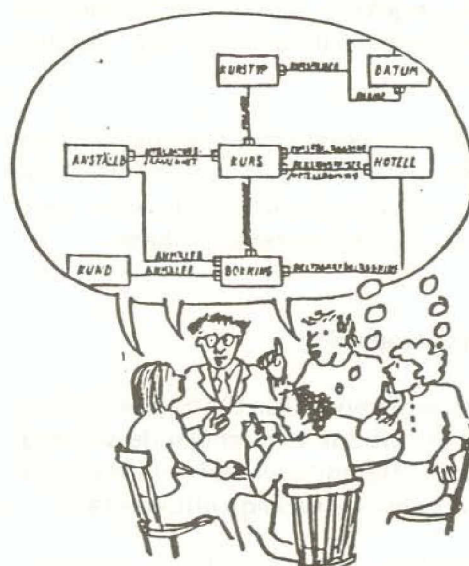
Bakgrund

En utgångspunkt för vår mix av metoder - som nu formulerats i utvecklingshandboken SVEA /ORT-85/ - är att vi vill använda metoder som stöttar något vi kan kalla de två T:nas princip:

- * Tänk klart!
- * Tillsammans!

Innebörden är att vi för viktiga metodsteg diskuterar med användargrupper i arbetsseminarier. Dynamiken i korsande erfarenheter och infallsvinklar utnyttjas.

För att diskussionerna inte skall flyta ut i vaga verbaliteter, fogar vi in ordbeskrivningarna i olika typer av grafer. De enkla men relativt stringenta graferna blir ett gemensamt språk.



De två T:nas princip betyder att graferna vi använder skall var lätta att lära och de skall hjälpa till att åskådligt beskriva komplexa flöden och samband.

I utvecklingsarbetet är det naturligt med en successiv förfining. Grafer och andra dokument förändras vid avstämningar. Kravet på förändringsbarhet bidrog till att vi sökte efter ett CAD-verktyg för systemutvecklingsarbetet.

Mer kraft tidigt

Fokus för EXCELERATOR är de tidiga stegen i utvecklingsarbetet: Kartläggning, analys och design. Men i och med att systembeskrivningarna byggs upp och lagras i PC/CAD-miljö, bör verktyget få en viktig roll även för det sista steget i livscykeln: Underhållet.

Mycket talar för att vi behöver satsa mer kraft på just de tidiga utvecklingsstegen. Vi vet av egen erfarenhet och genom olika undersökningar att:

- * Flertalet missar/fel **uppstår** tidigt. Bara en liten del är rena kodningsfel.
- * Missarna **upptäcks** ofta sent, när de är dyra att rätta till. Den relativa kostnaden för att rätta missar är 10 - 100 gånger större när vi befinner oss i kodnings-/införandestegen jämfört med kartläggnings-/analysstegen.

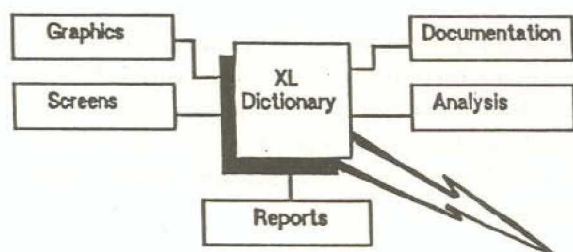
* "Lyckade" utvecklingsprojekt satsar väsentligt mer kraft på den tidiga analysen i samspel med användare. En definition på **lyckade projekt** är att de uppnått stor användartillfredsställelse och bra budgethållning.

Om vi skall kunna satsa mer kraft på de första stapplande stegen i utvecklingsarbetet, har vi i realiteten bara ett alternativ. Vi behöver verktyg som ökar produktiviteten i arbetet.

Verktygets utformning

Enligt våra beräkningar kan EXCELERATOR fungera som ett produktivitetshöjande verktyg. Vi har dock fortfarande för liten erfarenhet för att efterkalkylera de verkliga effekterna.

EXCELERATOR är utformat på följande sätt:



Grafik

EXCELERATOR representerar en ny typ av ritverktyg. Det är vare sig metodspecifikt (bundet till en viss metodkedja) eller ett enkelt "stand alone"-ritverktyg (jämför med Nordisk DATAytt 16/85, s 42-44).

Sex typer av grafer kan ritas, varav fem är mer forbundna. Med de forbundna graferna går det att beskriva datamodeller, logiska flöden, funktionella nedbrytningar m m. Med den fria presentationsgrafan kan en mängd olika grafer ritas, bl a rutinskisser ("vägg-grafer").

Det finns en funktion som gör det möjligt att peka på ett objekt eller en förbindelse i en graf och att därifrån "explodera" - dvs fördjupa beskrivningen - i upp till tio nivåer, i nya grafer eller andra beskrivningar.

En "plocka & peka"-teknik med styrdosa/mus gör ritverktyget lätt att använda.

Projekt-dictionary

I centrum finns hjärnan eller hjärtat, projekt-dictionaryet (XLD), som håller ihop de fem tentaklerna. Det finns en tät koppling mellan graferna och XLD. Uppgifter i XLD kan förändras och tas bort från graferna.

XLD är central förvaringsplats för **all** systeminformation. Uppgifter om data, processer, grafer, skärmar m m lagras här. 28 st s k entity types utgör grunden för XLD.

I projekt-dictionaryet finns även 80 st relationships, som möjliggör diverse korsreferensfrågor.

Kort uttryckt bidrar XLD till att redundansen minimeras, konsistensen säkras och analyser förenklas.

XLD gränssnitt

Genom en interface-funktion finns det möjlighet att flytta innehåll i ett projekt-dictionary till andra projekt, mellan olika systemerare och mellan PC-arbetsplatser och olika stordatormiljöer.

Detta innebär att stora projekt kan dela på standarder (t ex för termer) och att uppgifter om tabeller och skärmar kan förmedlas till och från vissa produktionsmiljöer.

Skärmar och rapporter

Med denna funktion utformas skärmar och rapporter med målningsteknik. Termernas egenskaper kan hämtas från XLD. Raka dialogkedjor kan byggas och skärmarna kan testas genom att "levande" data matas in.

Lagrade uppgifter kan sedan tas ut, selekteras och sorteras i rapporter.

Funktionerna möjliggör en enkel och tidigt arbetande modell av systemet. Användarna kan snabbt pröva de "produkter" som efterfrågas.

Analyser

Med analysdelen kan projektmedarbetare skraddarsy sina egna rapportuttag från XLD. Selektioner (prefix/suffix m m) och sorteringar av uttagen görs enkelt.

Här finns också vissa kontroller av logik och konsistens i dataflöden och i grafiska nedbrytningar.

Dokumentation

Med hjälp av en hierarkisk graf, går det här att bestämma indelning av och innehåll i dokumentuttag. Med dokumentfunktionen går det att kombinera ordbehandlingstext med grafer.

Funktionen gör att datasystemet/verksamhetsavbildningen enkelt kan dokumenteras till olika avstämningstillfällen. Våldokumenterade system utgör därtill en grund för morgondagens förändringsbehov.

Några erfarenheter

Av beskrivningen bör framgå att EXCELERATOR är en omfattande produkt. Det tar tid att fullt ut lära sig använda alla funktioner. Inom Kommundata är verktyget i ett införandeskede.

Utifrån vår ännu så länge begränsade användning, skall jag dock kort sammanfatta några positiva, negativa respektive intressanta erfarenheter. Eftersom vi är både användare och återförsäljare av produkten, skall jag lägga band på mig i lovorden, och strikt begränsa mig till tre erfarenheter av vardera sorten.

Positivt

- * De kraftfulla ritfunktionerna. Snabbheten med vilken objekt och symboler kan ritas, flyttas, förminskas/förstoras, förbindas etc. Lättheten att, med bibehållen överblick, växla mellan olika förstoringsnivåer och att kombinera andra grafer och textbeskrivningar med en graf.
- * Den sammanhållande "slaven" XLD, som med automatiserad tysk nitiskhet håller reda på alla uppgifter och dokument i ett pro-

jekt. Den administrerande uppgiften att hålla reda på alla termer, tabeller, skärmar etc och "entiteternas" inbördes samband, upplevs nog av de flesta som en mindre stimulerande uppgift.

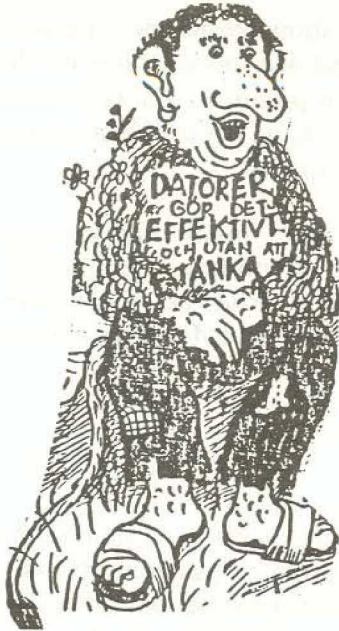
- * Den enkla mus-, meny- och meddelandetechniken. I en omfattande produkt som denna, är det viktigt att användaren leds fram i arbetet utan allför mycket manualsiddoblickar.

Negativt

- * Hänvisningen till ett antal givna ritsymboler. Våra beskrivningssätt är delvis annorlunda. Även om det går bra att använda en standardkostym, är det givetvis önskvärt att användaren kan få den skraddarsydd/måttbeställd.
- * Priset. Produkten är relativt dyr som PC-produkt (U S dollar 8 400 för ett exemplar är mycket med nuvarande dollarkurs). De flesta företag och organisationer lär liksom vi behöva pröva investeringen och göra en ordentlig test och effektkalkyl.
- * Uteslutningen av betydelsefulla hårdvarumiljöer här i Skandinavien. Produkten körs i DOS i IBM PC-miljöer. Som alla vet är kompatibilitet ett tänjbart begrepp. Vi vet t ex att det inte är problemfritt att köra EXCELERATOR och byta ut en IBM PC mot en Ericsson PC.

Intressant

- * Den vane användaren kan efter ett tag uppleva svarstiden för visning av stora grafer och för växlingar mellan menyer som litet betungande. Kraften i en PC med AT-kapacitet är nog mer lämplig för ett sådant här CAD-verktyg.
- * Graferna som tas ut på skrivare kan bli av högst skiftande kvalitet. Vid en investering i PC och mjukvara är det idé att satsa på en bra skrivare, som gör full rättvisa åt produkten.



- * Ett grafverktyg stödjer oss i arbetet med att tänka klart tillsammans. Men vi måste likväl tänka tanken först. Vår egen kompetens och de praktiska metodstegen för att frigöra verksamhetsexperternas kunnande i utvecklingsarbetet, är därför fortfarande **helt avgörande** för hur lyckat ett projekt blir.

Själv är jag ganska nöjd med att AI-trollen i våra moderna verktyg ännu så länge är små och beskedliga.

4.2 Några erfarenheter av ett datorstöd för MBI

Christer Fall, Programator

I samband med verksamhetsanalys enligt MBI-metoden har man inom kriminalvården prövat DAMBI (Datorstöd för MBI). DAMBI är en produkt, som utvecklats av Programator, vilken utnyttjar MacIntosh-programvaran FILEVISION.

Maskinkonfigurationen består av

- MacIntosh 512K
- Extern diskettenhet
- Imagewriter

Följande erfarenheter kan lämnas på den hitillsvarande användningen av DAMBI avseende skede 1 och 2 i MBI-metoden. Arbetsgången har i huvudsak bestått i att inledande dokumentation från blädderblock och dyl överförts till DAMBI. Inför efterföljande seminarier har OH-kopior på graferna i DAMBI

framställts. Eftersom den analyserade verksamheten får anses vara omfattande, ca 80 grafer har framställts, och endast en maskin varit tillgänglig, har DAMBI's dokumentbilder inte använts fullt ut. För kompletterande verbala beskrivningar har ordbehandlare använts.

Nackdelar

- Den nuvarande dokumentationen av DAMBI blir fullständig och förståelig först efter genomläsning av manualen för FILEVISION. Detta leder till en onödigt lång inläringstid.
- Utskrift på Imagewriter läser bildskärmsarbete.
- Vissa handgrepp i samband med konstruktion av bilder är omständliga, t ex att knyta samman text och figur till en symbol.

Fördelar

- Den största förtjänsten ligger i att dokumentationen bli väl sammanhållen och, när den väl lagrats på diskett, mycket snabbt och enkelt kan modifieras.
- Kostnaden för programvaran är blygsam, FILEVISION kostar ca 2 500:-.
- Länknings mellan bilder, via beskrivna symboler i graferna (t ex funktioner och informationsmängder) gör att konsekvensändringar i olika grafer kan genomföras smidigt.
- DAMBI går lätt att modifiera för anpassning till olika verksamhetsområden. Symboler och dokumentinnehåll kan ändras för att tillgodose olika krav.
- Snygga grafer och texter.
- Utrymmesbristen på skärmen "tvingar" användaren att skapa grafer som blir överskådliga.
- Många funktioner i FILEVISION är värdefulla vid konstruktion av grafer. Exempelvis kan "highlight" användas för att markera ärendeflöden och för att snabbt ta fram selektiva funktionsstrukturer ur en funktionsgraf.

Kapitel 5

TANKAR KRING DATORSTÖD

Anders Persson, Data Logic

INLEDNING

Om vi gör en tillbakablick och studerar systemutveckling kan vi konstatera att de datoriserade informationssystemen ofta motiverades av direkta kostnadsreduktioner, personalbesparingar samt av att få en bättre kontroll på materialflöden och penningströmmar.

Visserligen är samma motiv fortfarande relevanta, men det finns också andra. Informationshantering tillsammans med ADB kommer nu successivt att användas på taktiska och strategiska nivåer inom företaget/myndigheten och är då ofta intimt sammanvävd med affärsverksamheten och den strategiska utvecklingen. Datoriseringen griper nu också in i de mest centrala delarna och påverkar organisationsstrukturen och principerna för arbetsfördelning. Vi har också börjat se de effekter som morgondagens kontor skapar, inte minst ur kommunikationssynpunkt.

Utvecklingen av informationssystem är inte längre enbart ett tekniskt problem som löses av några datatekniker. För att lyckas med vårt utvecklingsarbete måste vi ha en helhetssyn som tar hänsyn till sambanden mellan arbetsuppgifter, arbetsmiljö, organisation, styrsystem, teknik etc. Detta innebär att många intressenter engageras i utvecklingsarbetet. Exempel är verksamhetsutvecklare, linjeansvariga, utredare (systemerare), organisationsutvecklare, personalutvecklare, dataadministratörer, systemmän/programmerare, systemförvaltare.

En administrativ förändring är ofta så mångfacetterad att en till synes liten förändring i sättet att arbeta får direkta konsekvenser för organisationen, för människorna i organisationen och på den teknik som brukas. För att vi skall kunna uppnå en samlad totaleffekt skall vi sträva efter lösningar som är en kombination av teknisk effektivisering, effektiv styrning, förbättrade förutsättningar för människans motivation, arbetsglädje och utvecklingsmöjligheter.

Hantering av information spelar i detta sammanhang en mycket central roll. Detta är en idag för många företag/myndigheter kritisk, dyrbar och viktig aktivitet. Vi måste lära oss att "anskaffa" information på ett rationellt sätt, "lagerhålla/arkivera" denna så optimalt och entydigt som möjligt, samt "bruka" den på ett korrekt sätt.

De tidigare etapperna av utvecklingsprocessen blir allt viktigare. Här är det nödvändigt för oss att använda kraftfulla, lättförståeliga och samverkande beskrivningstekniker för att alla intressentgrupper skall kunna delta utan alltför stora och dyrbara utbildningsinsatser.

Under det inledande analysarbetet arbetar vi med **modeller** vilka beskriver **vad/vilka** funktioner, arbetsuppgifter, begrepp, styrprinciper, organisation/ansvarsområden etc som skall finnas. Därefter utformar vi lösningar (**hur**) som ger oss de önskade effekterna. Under utvecklingsarbetet har vi behov av att detaljera oss successivt och belysa varje enskilt område/komponent.

Flertalet modelleringstekniker som vi arbetar med har den nackdelen att deras resultat är arbetssamma att framställa/förvalta och ändra med bibehållen konsistens. Hanteringen är ofta mycket manuell och primitiv. Vi önskar dessutom att modelleringsteknikernas resultat samverkar och vi vill kunna ta fram rapporter/beslutsunderlag ur en mängd aspekter.

Att få ett bra grepp om ovanstående med ett acceptabelt kvalitetsmål utan datorstöd är ofta en övermänsklig uppgift. Människan är ju primitiv när det gäller att lagra och selektera information, däremot är hon suverän på att associera och beskriva.

Vi skall därför ha bra beskrivningstekniker som passar människan. Vi skall ge datorn de uppgifter som vi är mindre bra på.

Genom datorstöd kan vi få en ökad möjlighet till att sammanställa resultat från utvecklingsarbetet i nya former. Information från flera

samverkande modelleringstekniker kommer att ge nya landvinningar vilket i sin tur ger en ökad effektivitet.

ALLMÄN REFLEKTION ÖVER OMRÅDET

Det mest påtagliga är avsaknaden av normer och regler för datorstöd samt avsaknaden av datorstöd för väsentliga delar inom AU-området. Vi kan, om vi generaliserar, säga att området befinner sig på ett mycket primitivt stadium.

Under utvecklingsarbetet vill vi ibland kunna se verksamheten lite annorlunda än vad vi ursprungligen har dokumenterat. Vi skulle vilja "vrida lite på vår vy" och se samma sak ur exempelvis organisationens synvinkel, styrsystemets synvinkel, teknikens synvinkel. Dvs, vi erhåller **samverkans effekter** från flera modelleringsteknikers resultat och vi vill dynamiskt kunna utnyttja detta.

Ovanstående låter naturligt men är idag i praktiken helt omöjligt. Det finns idag mycket primitiva och hantverksmässiga verktyg för denna hantering. Dessutom talar flertalet modelleringstekniker i "sitt eget språk" och man är inte "riktigt säker" på att det är samma sak man modellerar.

Vid systemutvecklingen tillverkar vi för verksamhetens räkning ofta dataregister (vilka helst utgår från konceptuella modeller). Dessa dataregister håller sedan data över de företeelser som verksamheten arbetar med (artiklar, kunder, order etc). Dataregister och tillhörande ADB-system är självklara hjälpmedel som ger verksamheten möjligheter till att öka sin effektivitet.

Om vi drar paralleller med vårt utvecklingsarbete kan vi konstatera att vi befinner oss på hantverksnivån. Vi har inte enats om vilka begrepp som vi vill hålla information om, dvs vi har inte beskrivit **vad** som skall ingå i **vårt** dataregister. Exempel på tänkbara begrepp i detta sammanhang är: funktion, process, information, term, händelse, objekt, relation, mål, nyckelområde.

Om vi inte enas och definierar upp området, är chanserna stora att vi kommer att misslyckas med modellerings- och datorstödssatsningar, därför att vi inte får önskade samordnings-

och förvaltningseffekter. Vi kommer istället att arbeta med isolerade produkter utan möjlighet till samverkan, kommunikation eller spridning (en s k "filorienterad miljö"). I längden kommer vi att bygga upp ett lappverk av produkter och vi får svårt att överblicka konsekvenser av ändringar (vi bygger upp vårt eget s k "spagetti-syndrom").

Trots den ringa utbredningen och användningen av datorstöd är intresset för området mycket stort. Vi ser också att en mängd produkter har börjat dyka upp på marknaden som var och en verkar lovande.

Det finns idag ett antal datorstödsprodukter som kan användas för att dokumentera och kontrollera den enskilda ADB-miljön. Det finns också datorstöd som är av typen modelleringsstöd vilka kan användas vid analysarbete i de tidiga etapperna av utvecklingsprocessen.

Fenomenet "Data Dictionary" blir allt vanligare och debatteras livligt. Det råder oklarheter över vad vi menar, speciellt när vi talar om centrala, lokala, projektorienterade Data Dictionary, **vad** vi skall lagra i dessa, samt vilka roller som de skall spela.

Vi känner därför en osäkerhet när det gäller satsningar inom området och vi saknar en teoretiskt underbyggd grund. Detta gäller för såväl synsätt/koncept som produkt.

Osäkerheten är naturlig eftersom området är en ganska ung företeelse. Vi kan direkt jämföra med införande av annan ny teknik typ fjärde generationens språk, artificiell intelligens, konceptuell modellering etc.

Av ovanstående anledning känns det angeläget att belysa datorstödsområdet ur ett helhetsperspektiv med målet att klarlägga området, fastslå inriktningar och anvisa hur satsningar på bästa sätt kan göras. Därefter kan vi belysa varje enskilt datorstöd så att detta på bästa sätt passar in i helheten.

ETT SYNSÄTT PÅ DATORSTÖDS-STRUKTUR

Nedanstående synsätt är en **idé-struktur** som försöker att sätta in respektive datorstödstyp i sitt sammanhang.

Data Dictionary

Data Dictionary ser vi som den **databas** vilken lagrar all information från vårt analys- och konstruktionsarbete.

Detta innebär **inte** att vi endast menar **en** databas som lagrar allting. I stället kan det finnas ett antal databaser som var och en betjänar sitt "applikationsområde". Det viktiga är att vi har full kontroll på vårt data i de olika databaserna, vi vet vad dess innehåll betyder och vi vet dess aktualitet.

Några viktiga aktiviteter för utvecklingen av Data Dictionary-konceptet är:

Att klargöra vad vi menar med centrala, lokala och projektorienterade Data Dictionary och hur dessa samspelar med varandra.

Att anvisa hur vi organiserar och styr hanteringen kring Data Dictionary samt att klargöra vilka roller och yrkesprofiler som förekommer.

Att vi tar fram en "konceptuell modell" över vilka begrepp som skall lagerhållas i Data Dictionarykonceptet. Genom denna kan vi sedan erhålla samverkans effekter.

Att klarlägga vilka begrepp som vi skall förvalta. Skall vi förvalta det som härrör sig till ADB-systemet, informationssystemet eller verksamhetssystemet?

Hur ser vi på olika versioner av data i Data Dictionary? Skall vi även hantera historiska versioner av modeller som har tagits fram under utvecklingsarbetet? Hur gör vi vid förändringar, stora ändringstillfällen etc?

Hantering av data i Data Dictionary

När vi arbetar med datorstödsprodukter av typen Data Dictionary har vi ofta stora krav på att "föda" dessa med information från vårt analysarbete. Informationen som vi matar in vill vi sedan manipulera och sammanställa under enkla former.

Av denna anledning måste vi kunna klarlägga vilka systemfunktioner som tillhör själva Data Dictionary-konceptet och vilka som ligger utanför. Klarlägger vi denna snittyta "bäddar" vi för en kontrollerad integration med möjliga samverkans effekter som följd.

Genom att dessutom få tillgång till effektiva utsöknings- och sammanställningsmöjligheter kommer analysarbetet att erhålla nya landvinningar som annars inte är möjliga. Vi får tillgång till en ökad "analyskraft" som kan öka kvaliteten i modelleringsresultatet. Nya möjligheter kan skapas genom att vi kan selektera information om begrepp via ett datorstött synsätt. Exempel på effekter inom några områden är:

- Tidigare manuellt och "tråkigt" arbete utförs nu maskinellt med en betydligt högre säkerhet och kvalitet. Resurser frigörs som kan användas för kreativt analysarbete.
- Vi kan få nya önskvärda arbetsuppgifter som tidigare inte har varit möjliga. Vi kan bättre bedöma vår förändring i ett helhetsperspektiv under hela utvecklingsarbetet. Samtidigt bör det bli möjligt att eliminera tidigare tidsödande och monotona arbetsuppgifter.
- Ökad förståelse av data och information i användarleden som kan öka besluts kvaliteten etc.

Datorbaserade modelleringsstöd

Modelleringsstöd i olika former och i olika tillämpningsområden är, i analogi med traditionellt systemutvecklingstänkande, de "applikationssystem" som vi använder i vårt arbete för att bli effektiva.

Vi kommer att få behov utav att använda såväl grova **skisseringsstöd** som välutvecklade och sofistikerade **modelleringsstöd**.

När vi arbetar i utredningsform har vi behov av snabba och enkla hjälpmedel för skissering och rapportering. Möjligt är också att det finns klara behov av att dessa stöd skall kunna vara portabla. Exempel:

- Skissering av grova funktionsstrukturer.
- Grova begreppsmodeller.
- Nätverksdokumentation typ orsak/verkanstrukturer samt korsreferenser för dessa.
- Grafisk rithjälp i olika former.
- Stöd vid protokoll- och rapportframställning.
- Enkel ord- och textbehandling.

När vi arbetar i avgränsade aktiviteter i ett utvecklingsprojekt kan vi ha klara behov av kraftfulla modelleringsstöd. Ofta finns ett modelleringsstöd för att beskriva "en vy" av verksamheten. Exempel:

- Datamodellering, både på grov och detaljerad nivå.
- Funktionsmodellering, både på grov och detaljerad nivå.
- Kopplingar mellan organisation, funktioner och data.
- Kopplingar mellan organisation och arbetsuppgifter per individ.
- Simulering av logiska och fysiska databaser givet den konceptuella modellen.

Om vi sammanfattar, ser vi att det förekommer en mängd olika modelleringsstöd som riktar sig till olika målgrupper. Vissa stöd/målgrupper har behov av grafiskt baserade beskrivningstekniker, andra har behov av textbaserade beskrivningstekniker. Det finns också modelleringsstöd som kommer att generera olika former av "kod" och som till viss del är aktiva.

Ovanstående behöver inte självklart betyda att vi arbetar med flera datorstöd i olika miljöer (även om det är det mest sannolika). Däremot är det självklart att datorstöden har olika funktionella betydelser och riktar sig till olika målgrupper allt i den takt som utvecklingsarbetet genomförs.

Förutom att vi måste klargöra vilka modelleringsstöd som skall stödja vilken aktivitet i utvecklingsprocessen, finns ett klart behov av att kunna transportera det resultat som modelleringsstöden ger. Vi vill kunna göra transporter mellan modelleringsstöd och mellan modelleringsstöd och Data Dictionary. Annorlunda uttryckt, behöver vi någon form av "arkitektur" som möjliggör samverkans effekter och nya utsökningsmöjligheter.

SAMMANFATTNING

Det finns all anledning till att stödja framtagning av datorstöd för utvecklingsarbetet. Vi kommer att se, (inom vissa delar har det redan skett), en mängd produkter som kommer att kunna öka den sammanlagda effektiviteten i vårt arbete.

Genom att vi med hjälp av olika former av modellering höjer "kod-nivån" kommer vi troligen att kunna få en ökad rationalitet av förvaltningsarbetet.

Om vi tänker igenom datorstödsområdet på ett planmässigt sätt innan vi rusar åstad och utvecklar eller köper produkter, kommer vi sannolikt att erhålla önskade effekter.

Kapitel 6

KAN DATORSTÖD FÖRBÄTTRA VERKSAMHETSANALYSEN?

Mats-Åke Hugoson, Programator

Inledning

Detta avsnitt kommer att behandla möjligheter att utnyttja datorstöd för verksamhetsanalys. Avsikten är inte att presentera en viss teknik eller ett visst verktyg utan snarare att ställa krav och diskutera användningsmöjligheter för sådana hjälpmedel.

Datorstöd för verksamhetsanalys kan tänkas i flera olika former. Det som här kommer att behandlas är att avbilda, modellera verksamhetssystem och att använda dessa modeller för att i någon mening förbättra verksamhetsanalysen.

6.1. Två paradoxer

En verksamhet kan innehålla dels **formaliserade**, mekaniserade delsystem (t ex robotar, databehandlingssystem) dels **administrativa** delsystem, där människor ingår som väsentliga komponenter.

Att modellera verksamhetssystem med datorstöd är ur flera synvinklar paradoxalt:

1. Administrativa system låter sig inte formaliseras.

Kännetecknande för administrativa system är att människors förmåga, beslutsfattande och olika åtgärder i verksamheten leder till resultat som icke kan förutsägas.

Att med datorstöd avbilda administrativa system är paradoxalt.

2. I en verksamhet råder många olika uppfattningar både om hur verksamheten bedrivs och hur den skall bedrivas, hur den styrs och hur den skall styras, etc.

Både ur systemteoretisk synvinkel ("soft system theory") och ur organisationsteoretisk synvinkel är dessa förhållanden nödvändiga för att verksamheten skall fungera.

Att verksamhetsanalysen förbättras med en datorstödd modell är paradoxalt.

Till synes ännu mera motsatsfullt blir det om datorstödet hanteras av en specialist, just i avsikt att få fram **en enda** formaliserad beskrivning. De verkliga förhållandena och uppfattningarna i verksamheten kommer då ej att beaktas. Datorstödet får inte bli ett hot mot användarorienterad verksamhetsanalys.

Vi måste uppenbarligen närma oss den datorstödda verksamhetsanalysen försiktigt och inte tro att man enbart genom att skapa ett datoriserat verktyg som grafiskt eller på annat formaliserat sätt försöker avbilda en verksamhet kan ge ett verkligt stöd för verksamhetsanalysen.

6.2. Systemteoretiska utgångspunkter

Verksamhetsanalys innebär analys av den verksamhet som skall förses med informationssystem.

En viktig utgångspunkt är därvid att informationssystemet inte är systemet.

De delar av verksamheten som **använder** informationen ingår i systemet även om de inte kan formaliseras. Det eller de delsystem som **producerar** den formaliserade informationen (dvs informationssystemet) är enbart delsystem i ett större sammanhang. Informationssystemet kan inte representera verksamheten. Naturligtvis avbildas vissa förhållanden i verksamheten formaliserat i informationssystemet, men det räcker inte att i verksamhetsanalysen avbilda informationssystemet.

En annan viktig utgångspunkt är att man för att kunna analysera verksamheten måste skapa en i systemteoretisk mening god, dvs överblickbar, delsystemstruktur.

Ett angreppssätt som visat sig vara framgångsrikt är att använda funktionella delsystemstrukturer (funktionsstrukturer), där varje

delsystem svarar mot en avgränsad del av verksamheten med bestämda uppgifter. Mellan delsystem finns många olika samband, några är formaliserade informationssamband som kan och bör beskrivas. Andra relationer (t ex personliga relationer mellan personer inom olika funktioner) kan inte entydigt beskrivas. Ett verksamhetssystem kan alltså aldrig fullt ut beskrivas formaliserat.

I denna funktionsstruktur, som avbildar hela verksamhetssystemet, kan informationsdelssystem införas och den formaliserade informationsförsörjningen beskrivas (se fig 2 och 3).

6.3. Verksamhetsanalysens och datorstödet syften

Den allmänna utgångspunkten är att vi utvecklar informationssystem för att nå effekter i verksamheten, för att i något avseende underlätta eller effektivisera verksamheten.

Med denna truism (?) som underlag kan två huvudsyften för verksamhetsanalysen anges:

1. Att specificera lämpliga informationssystem.
2. Att medverka till rätt användning av informationssystem.

Det senare syftet är inte oväsentligt. Användes informationssystemet inte på rätt sätt, blir det inga effekter i verksamheten hur väl specificerat informationssystemet än är.

Datorstöd för verksamhetsanalys är potentiellt ett hjälpmedel för båda dessa syften. Därutöver kan datorstöd tjäna ytterligare ett syfte, nämligen:

3. Att effektivisera själva verksamhetsanalysen.

Detta syfte får dock icke råka i konflikt med övergripande syften för verksamhetsanalysen.

Eftersom användarmedverkan i verksamhetsanalysen har visat sig vara av avgörande betydelse för både syfte 1 och 2, kan följande restriktion formuleras:

Datorstödet får inte försvåra de verkliga användarnas möjligheter att direkt inverka på verksamhetsanalysen.

6.4. Vad kan förbättras med datorstödd verksamhetsanalys?

6.4.1. Styrstrukturen

I inledningen av verksamhetsanalysen måste verksamhetssystemet (verksamhetsområdet) definieras. Ofta sker detta implicit genom att de som bedriver analys gör antaganden. De som är verksamhetsansvariga kan ha gjort andra antaganden. Datorstöd kan bidra till att explicit visa verksamhetssystemets avgränsning.

Inom det avgränsade verksamhetsområdet skall därefter den administrativa styrstrukturen visas.

Den datorstödda modellen är i detta skede närmast en förklaringsmodell. Funktioner i verksamheten identifieras och deras uppgifter beskrivs. Sättet att genomföra uppgifterna kan däremot inte formaliserat beskrivas. Förändringar i styrstrukturen kan konkretiseras i modellen. Om datorstödet är flexibelt och enkelt kan visa ändrade funktionsstrukturer, kan analysarbetet avsevärt underlättas.

6.4.2. Konsistensanalys - samordning

I verksamhetens funktioner råder olika uppfattningar om hur verksamheten bedrivs och skall bedrivas. Peter Checkland har skapat begreppet "Root definitions" och visat hur olika Root Definitions gäller och måste gälla för en och samma del av verksamheten.

Hittills har hjälpmedel saknats att systematiskt analysera inkonsistens och samordningsbehov som en följd av dessa olika Root Definitions. För formaliserade informationssamband (men endast för dessa) kan ett datorstöd erbjuda sådana möjligheter. I det datorstöd som benämns CABA (Computer Assisted Business Analysis) och som håller på att utformas inom AB Programator, används termen **etablerat** informationssamband när inkonsistens råder dvs när en funktion anger ett informationsbehov som den andra parten beskriver på ett annat sätt (eller inte beaktar) och

accepterat informationssamband när funktionerna är överens. Datorstödet bygger på s k selektiv teknik som medger olika Root Definitions och det är väsentligt att både etableringen och samordningen skall genomföras av användaren inom respektive funktion. Specialistens uppgift är närmast att hålla fram verktyget.

Denna detaljerade analys och samordning kan förväntas ge bidrag både till att de kommande informationssystemen får rätta egenskaper och rätt användning. Den formaliserade konsistensanalysen är möjlig med datorstöd, men skulle med manuella insatser vara alltför omfattande.

6.4.3. Lokala informationsbehov - informationssystem

I första omgången kan frågan om lokal information ses som omvändningen på konsistensproblemet, nämligen att fastställa vilken information som endast berör en funktion inom verksamhetsområdet.

Detta är en viktig fråga i sig, speciellt i samband med decentraliseringssträvanden och krav på en ökad autonom styrning. Datorstöd kan förväntas ge väsentliga bidrag till denna analys under förutsättning att varje funktion kan hantera sin version av verksamhetsmodellen.

I förlängningen ligger emellertid en än viktigare fråga, nämligen avgränsning av funktionellt orienterade informationssystem.

Grundprincipen bör vara att den funktion som har ansvar för en viss verksamhet också förfogar över det, eller de, informationssystem som är hjälpmedel för denna verksamhet. Varje sådant delsystem måste samverka med andra informationssystem, så att man får en struktur av funktionellt orienterade men samverkande informationssystem.

Att med datorstöd sammanställa lokala informationsbehov och att efterhand fånga upp och visa alternativa förslag till avgränsningar av informationssystem, bör väsentligt öka

möjligheterna att avgränsa lämpliga informationssystem. Att selektera och visa vilka samband som skall gälla **mellan** informationssystemen kan sedan enkelt ske med datorstöd.

6.4.4. Övergång till informationsmodeller

De datorstödda verksamhetsmodellerna avser (enligt vår tidigare systemteoretiska utgångspunkt) hela verksamhetssystemet, även om förhållanden inom verksamhetens funktioner endast i mycket ringa omfattning kan avbildas.

Med informationsmodell avses här en modell av de verksamhetsdelar man beslutar att avbilda (kunna avbilda) i ett informationssystem.

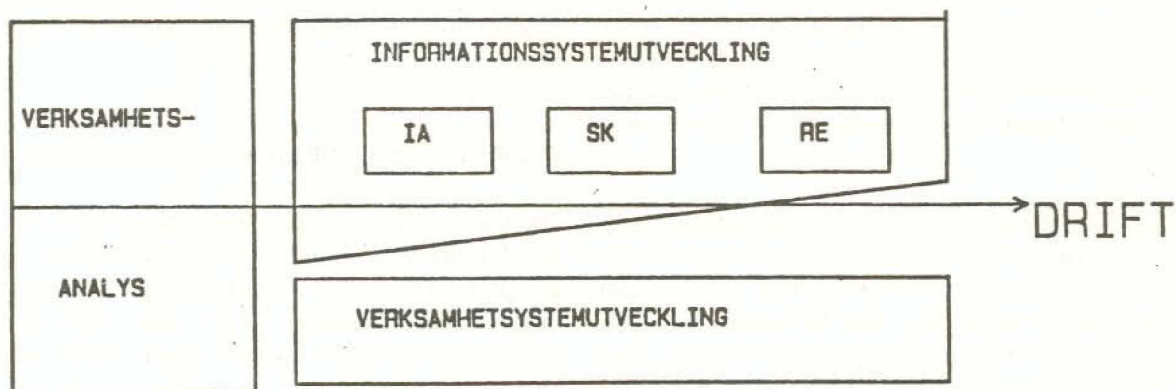
Det finns naturligtvis ett mycket stort antal objekt, händelser, entiteter (eller vilket begrepp man vill använda) i verksamheten som inte skall avbildas i informationssystemet av den enkla anledningen att det inte skulle ge några väsentliga effekter i verksamheten. Att på ett formaliserat och stringent sätt för varje informationssystem ur verksamhetsmodellerna extrahera de delar som skall avbildas i en informationsmodell, framstår som en fundamental uppgift för datorstödd verksamhetsanalys.

Det bör emellertid framhållas att vi inte har alldeles klart för oss hur detta urval skall ske. En fördjupad samverkan mellan "verksamhetsanalytiker" och "informationsanalytiker" är angelägen.

6.4.5. Verksamhetsmodeller i verksamhetsutvecklingen

De önskade effekterna i verksamheten uppnås inte enbart genom att man utvecklar nya informationssystem. Användarna i olika funktioner i verksamheten måste få möjligheter att lära sig arbeta enligt de förändrade formerna och att använda informationssystemen, dvs verksamhetens funktioner måste utvecklas parallellt med informationssystemen (se fig 1).

SPECIALISTORIENTERAD ANALYS



ANVÄNDARORIENTERAD ANALYS

Figur 1: Användar- resp specialistorienterad analys.

Detta utvecklingsarbete ligger visserligen efter verksamhetsanalysen, men datorstödet för verksamhetsanalysen kan ge modeller som bör vara användbara.

Samverkan mellan olika verksamhetsfunktioner kan spelas fram om datorstödet har byggts så att det inrymmer en tidsdimension. Det rör sig således om ett verksamhetsspel där informationssystemens användning kan simuleras, men eftersom den faktiska verksamheten avbildas, har dessa tillämpningar en helt annan relevans än de företagsspel som hittills prövats i utbildningssyfte.

6.5. Sammanfattning

Ett rittekniskt hjälpmedel som försöker avbilda verksamheten på ett normaliserat sätt är inte tillräckligt som datorstöd för verksamhetsanalys. Det kan till och med störa verksamhetsanalysen genom att användarmedverkan försvåras.

Datorstödet måste anpassas till den aktuella metoden för verksamhetsanalys och till de si-

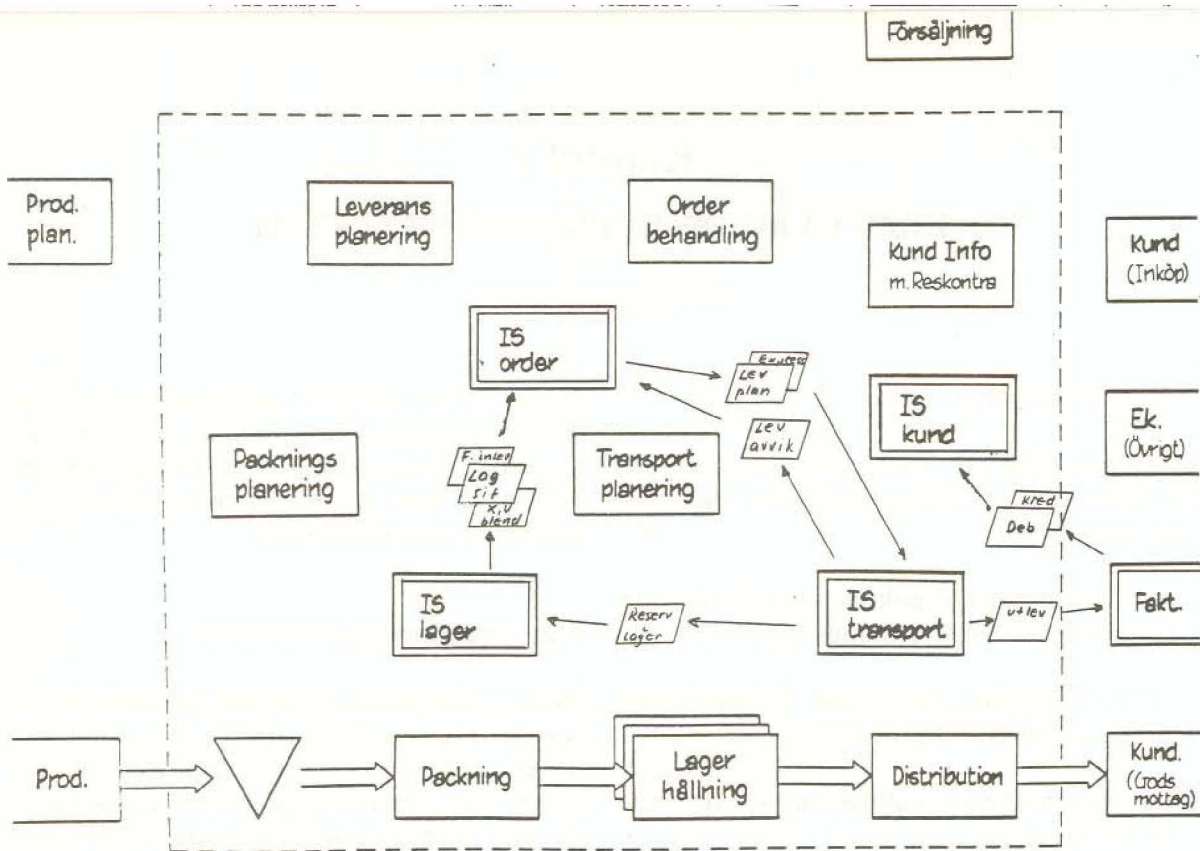
tuationer som uppstår ute i verksamheten.

Olika, och ofta inkonsistenta versioner av verksamheten måste kunna hanteras och analyseras av datorstödet.

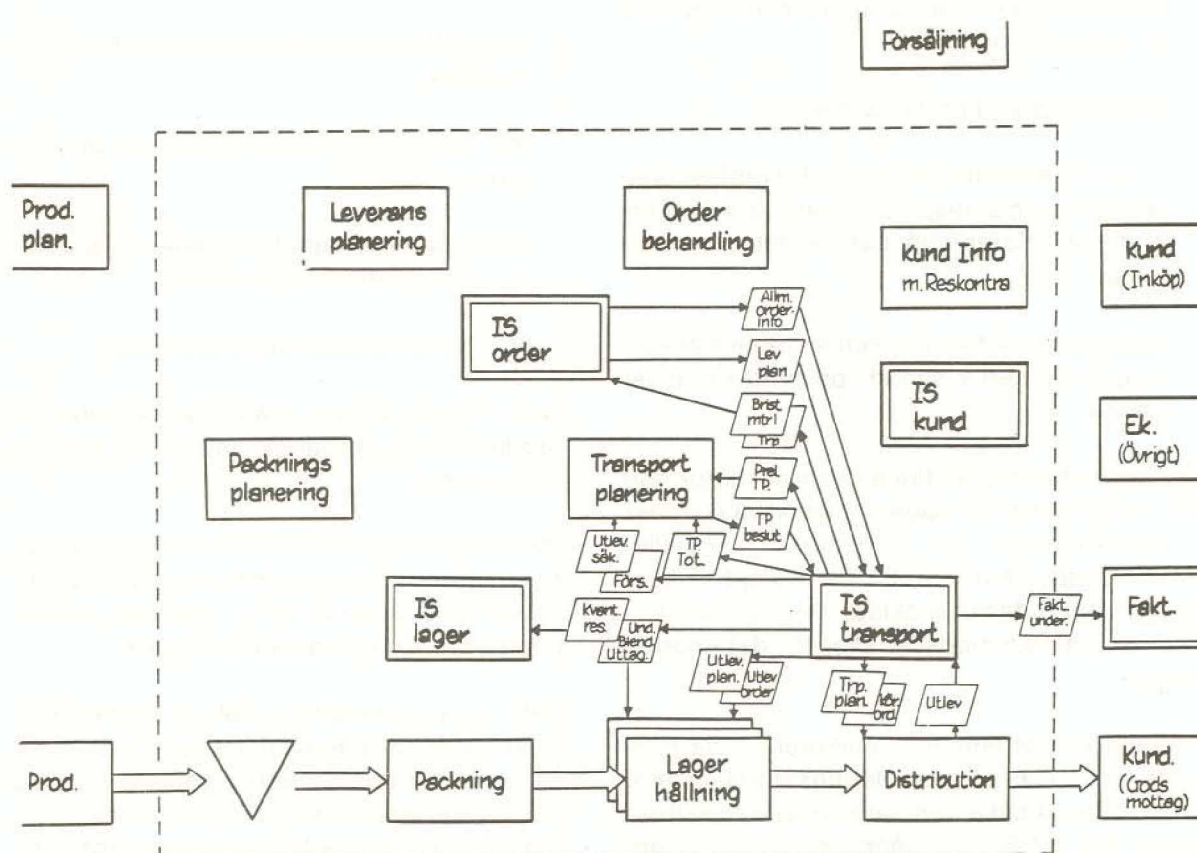
För att vi skall nå det egentliga syftet att effektivisera verksamheten, måste datorstödet vara ett direkt hjälpmedel för de verkliga användarna när deras verksamhet analyseras.

De informationssystem som avgränsas med datorstödet måste dels vara relevanta ur användarnas synpunkt och dels vara entydigt specificerade som underlag för den fortsatta systemutvecklingen.

Bedömningen är att datorstöd för verksamhetsanalys kanske inte radikalt kan förenkla verksamhetsanalysen, men kan medverka både till att säkrare avgränsa och specificera informationssystemen och till att öka förståelsen för hur informationssystemen skall användas.



Figur 2: Samband mellan informationsdelsystem.



Figur 3: Slutlig avgränsning av informationsdelsystem.

Kapitel 7

KRAV PÅ MODELLERINGSSTÖDSSYSTEM

Ulf Sundin, SYSLAB

7.1. Introduktion

Vid utveckling av informationssystem används en mängd olika modeller, t ex modeller som beskriver

- den nuvarande respektive den förväntade omgivningen till informationssystemet
- den information som skall representeras och bearbetas i informationssystemet
- hur informationen skall bearbetas i informationssystemet.

Dock är ofta skapandet av dessa modeller, modelleringsarbetet, både tidsödande och svårt. Svårigheten i modellering är beroende av flera faktorer, t ex:

- komplexiteten hos de språk som används för modellering
- komplexiteten i objektsystemet
- skapade modeller behöver ofta omformuleras och preciseras, vilket kan ge svåröverskådliga effekter på övriga delar av modellen
- svårigheten att avgöra om en modell är korrekt, både med avseende på syntax och semantik.

Detta innebär att det finns behov av datorstöd för modellering. Ett sådant datorstöd kommer i fortsättningen att benämnas "modelleringsstödssystem" (MSS). En mängd sådana system har redan utvecklats, t ex för olika former av grafiska modeller eller för datamodellering.

I denna artikel ämnar vi diskutera vilka krav som kan ställas på modelleringsstödssystem. Först belyses olika tänkbara användingssituationer för MSS, och några av de krav som dessa användningar ställer på ett MSS. Dessa krav relateras kortfattat till egenskaper hos existerande MSS. Därefter formuleras några

generella krav på MSS, och slutligen diskuteras hur dessa krav kan uppfyllas. I artikeln görs ingen begränsning till krav på grafiskt orienterade MSS, utan grafik betraktas enbart i termer av användarinterface.

7.2. Problemspecifika krav

Med ett problemspecifikt krav på MSS, så avses ett krav som är aktuellt i en viss modelleringssituation. Exempelvis, vid utvecklandet av ett visst informationssystem, användande en viss systemutvecklingsmetod och vissa modelleringsspråk, så beror kraven på ett MSS i hög grad på en mängd situationsspecifika faktorer, t ex:

- Till vilka uppgifter skall ett MSS användas? Några tänkbara uppgifter kan vara att:
 - underlätta manipulation och skapande av modeller
 - att understödja dokumentation av skapade modeller
 - att analysera skapade modeller med avseende på syntax och semantik
 - att utföra resonemang i modellen.

Naturligtvis måste ett MSS vara kapabelt att stödja, eller automatiskt utföra, de avsedda uppgifterna.

- Vem skall använda MSS? Är användaren inte en erfaren datoranvändare, så bör naturligtvis MSS ha ett lämpligt anpassat användarinterface, t ex baserat på grafik.
- Vilka typer av modeller skall användas? Förutom att MSS måste kunna stödja de aktuella modelleringsspråken, så måste vi också beakta karaktäristika hos modellerna, t ex om de är statiska eller dynamiska, formella eller informella, etc.

Om vi nu betraktar några existerande MSS,

t ex /ATZ-82/, /BOL-84/, /CHA-80/, /DEL-82/ och /REI-84/, så finner vi att de flesta endast är kapabla att stödja en begränsad mängd modelleringspråk, och därmed ofta är specialiserade för någon speciell uppgift, t ex databasmodellering och design. Vidare, så fokuseras ofta intresset på möjligheten att skapa och manipulera modeller, samt att understödja dokumentation. Vad det gäller analys av modeller, t ex med avseende på korrekthet, så begränsas den ofta, om den alls förekommer, till vissa rent syntaktiska kontroller.

Från detta kan vi dra den triviala slutsatsen att i vissa modelleringsituationer så uppfyller ett givet MSS våra krav, och i andra inte. Detta påvisar faktumet att problemspecifika krav på MSS inte utgör speciellt lämpade kriterier, och särskilt inte för design av MSS.

7.3. Generella krav på MSS

Ett sätt att kringgå problemet med specifika krav på MSS är att försöka identifiera ett antal generellt tillämpbara krav. Utan att hävda att de utgör en komplett uppsättning, så kommer följande krav att behandlas:

1. Ett MSS skall vara kapabelt att stödja en stor mängd varierande typer av modeller och modelleringspråk.
2. Ett MSS skall inte bara vara ett verktyg för konstruktion och manipulation av modeller, utan skall även ha kapacitet att analysera och utvärdera modeller, samt att utföra resonemang.
3. Ett MSS skall ha ett användarinterface som möjliggör att icke-expertter kan delta i modelleringsarbetet.

7.3.1. Om att stödja en stor mängd modelltyper

Motivet till det första kravet är att metoder för systemutveckling ofta inkluderar en mängd olika typer av modeller. Om ett MSS endast kan användas till vissa av dessa, så leder detta till att antingen måste ett antal olika MSS användas, eller så uppstår "luckor" för vilka inget MSS används. Båda dessa alternativ leder till problem:

- det gör det omöjligt att använda resultaten från en modelleringsaktivitet som input till en efterföljande aktivitet.
- de som utför modelleringsarbetet tvingas lära sig en mängd olika MSS och deras speciella interface.

Ett minimikrav är att ett MSS skall vara kapabelt att stödja åtminstone de olika modelleringsaktiviteter som ingår i en viss metod för systemutveckling.

7.3.2. Om att analysera korrekthet hos modeller

Vad gäller det andra kravet, så innebär det för det första att MSS skall möjliggöra och stödja olika former av korrekthetskontroller på konstruerade modeller.

Förutom rent syntaktiska kontroller, som att avgöra om en modell uppfyller modelleringspråkets syntax, har existerande MSS en mycket begränsad kapacitet för korrekthetskontroller, vilket leder till en del problem. Det är ju t ex uppenbart att egenskapen hos en modell att vara syntaktiskt riktig i förhållande till ett visst modelleringspråk inte säger någonting om huruvida modellen är en riktig, eller lämplig, representation av ett objektsystem.¹⁾ Naturligtvis är "semantiska" kontroller ett mycket svårt problem, vilket är lätt att inse om vi jämför med motsvarande resultat för bevis om programs korrekthet. Speciellt svårt är det naturligtvis vid användande av "informella" modeller, som saknar väldefinierad semantik.

7.3.3. Att utföra och stödja resonemang

För det andra så bör MSS, om det anses relevant, kunna användas för att utföra resonemang i modellen, och därmed också indirekt om objektsystemet. Just detta att möjliggöra resonemang och slutsatsdragande om objektsystemet är, oavsett vilka aspekter av objektsystemet som representeras i modellen, syftet med modellering. Om nu modellen av objektsystemet är komplex och svåröverblickbar,

¹⁾ Den som inte övertygas av detta kan jämföra med programmeringspråk: Ett syntaktiskt riktigt program behöver inte lösa det avsedda problemet, eller ens något problem.

och kanske också innehåller komplexa regler, så är det önskvärt att MSS antingen kan tillhandahålla lämpligt interaktivt stöd för resonemang och slutsatsdragande, eller automatiskt kan resonera i modellen.

En egenskap som ett MSS bör ha för att kunna stödja resonemang i komplexa modeller är att möjliggöra förenklingar av modellen, t ex följande:

- Ur en given struktur, presentera en relevant och överblickbar delstruktur. T ex när det gäller användningen av grafiska modeller, så ger dessa ofta en god överblick av modellen. Detta gäller dock endast under förutsättning att modellen inte är för omfattande. Det är i så fall nödvändigt, om modellen är representerad i ett MSS, att ha tillgång till ett kraftfullt frågespråk, med vilket det är möjligt att selektera en lämplig delmängd av modellen (jfr 'view' i samband med konceptuell modellering).
- Ur en given modell abstrahera bort vissa detaljer, för att erhålla en mer överblickbar modell, när så önskas. Vid exempelvis datamodellering är det ofta önskvärt att bortse från attribut, för att framhäva olika relationer mellan entitetstyper.
- Eliminering av redundant information från en modell. Detta kanske inte leder till så dramatiska förändringar av modellen som de ovanstående punkterna, men har å andra sidan den goda egenskapen att ingen information förloras.

Utan att i detalj gå in på automatiskt resonande, så kan följande funktioner hos ett MSS tänkas vara av intresse:

- Möjlighet att avgöra om ett visst faktum finns representerat i - eller är härledbart ur - modellen. I det enklaste fallet, när inga härledningsregler finns för modellen, se ovan. I det mer generella fallet krävs dock någon form av teorembevisare.
- Hypotetisk slutsatsdragning. Exempelvis kan vi vara intresserade av vilka slutsatser som kan dras om något ytterligare faktum adderas till modellen. Ett specialfall av detta är att avgöra om modellen fortfarande är korrekt om hypotesen inkluderas i modellen.

- Avgöra olika förhållanden som gäller mellan olika modeller. Till exempel vid "view integration" är det önskvärt att veta om en del av en modell är identisk eller isomorf med någon del av en annan modell. Observera dock att i detta fall, såväl som i fallet med korrekthet ovan, så är det inte frågan om resonemang i modellen om objektsystemet, utan föremålet för resonemanget är modellen själv.

7.4. Hur kan dessa generella krav uppfyllas?

Vad beträffar det första kravet, att ett MSS skall vara kapabelt att stödja en stor mängd olika modelltyper, kan två alternativa lösningar urskiljas. Antingen så "skräddarsys" MSS till en viss systemutvecklingsmetod, för att därmed uppfylla vårt minimikrav, eller så skapas ett generellt MSS med vilket en stor mängd olika modelltyper och design-steg kan stödjas.

Av dessa alternativ är det senare att föredra, huvudsakligen på grund av följande:

- Det är en stor brist på "kompleta" metoder för systemutveckling, och i praktiken är det därför vanligt att olika "partiella" metoder kombineras och blandas.
- Även om vi baserar MSS på en viss metod, så är det ofta förekommande att både själva metoden, såväl som de modeller som ingår i den, förändras och vidareutvecklas. Detta leder till att även modelleringsstödsystemet ständigt måste förändras och anpassas till metoden.

Dock, alternativet med generella MSS leder till att ytterligare krav måste ställas. Ett generellt MSS (GMSS) måste, om det skall ha godtagbara egenskaper, även uppfylla följande krav:

- Ett GMSS skall inkludera generell kunskap om modeller och modellering, /ERI-83a/. Detta är den grundläggande kunskap som behövs för att kunna definiera nya typer av modeller och deras användning.
- Ett GMSS skall inkludera ett interface för att definiera modelltyper, med avseende på det språk som används, härledningsregler som gäller, samt olika former av korrekthetskriterier. Denna facilitet motsvarar ett databashanteringssystem's datadefinitionsspråk (DDL).

- Ett GMSS skall baseras på ett representationsspråk som tillåter definition och representation av en mängd olika typer av modeller.

Vidare bör ett GMSS också inkludera ett generellt språk, dvs användbart för alla de modelltyper som definierats, för att göra utsökningar och manipulera modeller, jfr DML.

Huvudproblemen med det andra kravet är att:

- Ett MSS måste baseras på en formalism för att representera modeller som möjliggör ovan nämnda kontroller och resonemang. Detta leder tankarna till generella språk för kunskapsrepresentation, /BAR-82/. Detta förefaller också rimligt med avseende på ovan ställda krav på representationsspråk.
- De flesta typer av modeller som används vid systemutveckling har inte en formell syntax och/eller en väldefinierad semantik. Till följd av detta är det svårt att veta vad det är som kontrolleras, och vad en slutsats betyder, och därför är detta krav i mycket ett krav på de använda modellerna.

Kapitel 8

MER ATT LÄSA - REFERENSER

Det råder stor brist på litteratur av krav- och produktsammanställande karaktär inom datorstödsområdet. Inte minst gäller det inom det delområde som detta nummer av SISU-analys inriktas på dvs grafiskt baserade datorstöd för systembeskrivning. Vi begränsar därför vår "mer-att-läsa"-lista nedan till två in-slag och hänvisar i övrigt till referenslistan.

8.1. Mer att läsa

Schneider, H.J. och Wasserman, A. (Eds.): "Automated Tools for Information Systems Design", Proc. IFIP WG8.1 Working Conference on Automated Tools for Information Systems Design and Development, New Orleans, North-Holland, 1982.

IFIP's arbetsgrupp 8.1, med namnet "Design and Evaluation of Information Systems", anordnade 26-28 januari 1982 i New Orleans en arbetskonferens kring "Automated Tools for Information Systems Design". De olika bidragen har samlats i denna bok och utgör konferensens "proceedings".

Här presenteras ett 15-tal datorstöd av skiftande karaktär vilket framgår av den indelning i fem olika kategorier som gjorts: 1) Verktyg för analys, 2) Verktyg för systemdesign, 3) Verktyg för data-design, 4) Applikationsutvecklingsystem och 5) Integrerade utvecklingsmiljöer. Ett av analysverktygen, "Tools for Structured Analysis", Tektronix Inc., återfinns bland de datorstöd som tagits upp i kapitel 2.

Bidragen är av forskningsrapportkaraktär, ganska teoretiska till sitt innehåll och kräver en del av läsaren.

Lie-Nielsen, S: "CAD/CAM Teknikken vil revolutionere systemutviklingsarbeidet", artikel i Nordisk DATAnytt 16/85.

I denna korta, komprimerade och lättlästa artikel tas 15 grafiska datorstöd för systemutvecklingsarbete upp till översiktlig behandling. Datorstöden klassificeras i:

1. Generella ritverktyg.
2. Intelligent grafritningssystem.
3. Verktyg som stödjer existerande metoder.
4. Verktyg som stödjer en bestämd metod och som genererar färdig exekverbar kod.

I en produktöversikt markeras med ja/nej om datorstöden

- har grafik
- ger möjlighet att definiera egna symboler
- har integrerat data dictionary
- är användbart för utformning av skärmbilder och rapporter ("prototyping")
- stödjer strukturering och ajourföring av dokument (dokumenthantering)
- har kodgenereringsfacilitet

samt anges leverantör, vilken/vilka metoder de understödjer, vilken hårdvara som krävs och slutligen det pris de betingar.

8.2. Referenser

/ATZ-82/

Atzeni, P., et al.: "A Computer Aided Tool for Conceptual Data Base Design", i /SCH-82/.

/BAN-84/

Bansler, J., Bodker, K. och Schmeltz Pedersen, G.: "Systemudviklingssystemer. Rapport från SUSA-projektet", Dok. nr DDC 101/1984-03-23, Dansk Datamatik Center, Lyngby, 1984.

/BAR-82/

Barr, A. och Feigenbaum, E.: "The Handbook of Artificial Intelligence", Vol. 1, Pitman, 1982.

/BOL-84/

Bolstad, M.F. och Hamre, M.: "SIKSAK - System for Interaktiv Syntaksstyrt oppbygging av diAgrammer og Konsistenssjekking av diagrammer", DAISEE arbeidsrapport nr 25, The University of Trondheim and Norwegian Institute of Technology, 1984.

- /BUB-82/
Bubenko jr, J.: "On Model Types and their Conceptual and Spatial Representation in RAMATIC", SYSLAB, 1982 (ej publicerad).
- /CHA-80/
Chan, E.P.F. och Lochovsky, F.H.: "A Graphical Data Base Design Aid Using the Entity-Relationship Model", i Chen P.P. - S. (Ed.): "Entity-Relationship Approach to Systems Analysis and Design", North-Holland Publ. Co., 1980.
- /CHE-76/
Chen, P.P.-S.: "The Entity-Relationship Model - Toward a Unified View of Data", ACM Transactions on database systems, Vol. 1, No. 1, March 1976.
- /COM-79/
Computer Graphics, the ACM SIGGRAPH Quarterly, Vol. 13, No. 3, August 1979.
- /COM-84/
Computer Graphics, the ACM SIGGRAPH Quarterly, Special GKS issue, February 1984.
- /DAH-82/
Dahl, R. och Torbjär, H.: "A Description of the Internal Functions of RAMATIC", SYSLAB Working Paper 57, Chalmers Tekniska Högskola, Göteborg, 1982.
- /DAH-84/
Dahl, R.: "The Data Base Representation in the RAMATIC Tool and its Implication", SYSLAB Report No 27, Chalmers Tekniska Högskola, Göteborg, 1984.
- /DEL-82/
Delisle, N.M., Menicosy, D.E., och Kerth, N.L.: "Tools for Supporting Structured Analysis", i /SCH-82/.
- /DeM-78/
DeMarco, T.: "Structured Analysis and System Specification", Yourdon Inc., New York, 1978.
- /ERI-83a/
Eriksson, D. och Sundin, U.: "On correctness of graphical models", SYSLAB Report nr 28, Chalmers Tekniska Högskola, Göteborg, 1984.
- /ERI-83b/
Eriksson, D.: "The Modeling Concept within Ramatic", SYSLAB, 1983.
- /FOG-84/
Fogg, D.: "Lessons from a 'Living in a Database' Graphical Query Interface", i Yormark, B. (Ed.): "SIGMOD 84 Proceedings", ACM SIGMOD Record, Vol. 14, No. 2, 1984.
- /GAN-79/
Gane, C. och Sarson, T.: "Structured Systems Analysis", Prentice-Hall, Englewood Cliffs, NJ, 1979.
- /GRA-85/
Datorstödd Systemering. GRAPHDOC (Version 3.00), EPOC data ab, Stockholm, 1985.
- /HUG-83/
Hugoson, M.-Å., Hesselmark, O. och Grubbström, A.: "MBI-metoden. En metod för verksamhetsanalys", Studentlitteratur, Lund, 1983.
- /ING-78/
Ingevaldsson, L.: "JSP - en praktisk metod för programkonstruktion", Studentlitteratur, Lund, 1978.
- /JAC-75/
Jackson, M.: "Principles of Program Design", Academic Press Inc., London, 1975.
- /JAC-83/
Jackson, M.: "System Development", Prentice Hall Int., London, 1983.
- /JAN-84/
Janning, M., Berild, S. och Nachmens, S.: "CS5", Preedition, Databaskonsult DBK AB, 1984-12-05.
- /JOH-83/
Johansson, L.-Å.: "Graphics in Requirements Specification of Information Systems", Proc. CAMP'83, Berlin, 1983.
- /JOH-84/
Johansson, L.-Å.: "RAMATIC: A Computer-Aid for Modeling Activities (Use Situations)", Proc. SPOT-3, SYSLAB, Chalmers Tekniska Högskola och Stockholms Universitet, Göteborg, 1984.

- /KER-81/
Kerth, N.L., Delisle, N.M., och Menicosy, D.E.: "Tools for Supporting the Structured Analysis Methodology", Computer Research Laboratory, Tektronix Inc., Beaverton, Oregon, May 29, 1981.
- /KIN-84/
King, R.: "Sembase: A Semantic DBMS", i Kerschberg, L. (Ed.): "Proceedings of the First International Workshop on Expert Database Systems", Kiawah Island, S.C., USA, Oct. 24-27, 1984.
- /LIE-85/
Lie-Nielsen, S.: "CAD/CAM Teknikken vil revolusjonere systemutviklingsarbeidet", Nordisk DATAnytt nr 16 (November) 1985.
- /LUN-78/
Lundeberg, M., Goldkuhl, G. och Nilsson, A.: "Systemering", Studentlitteratur, Lund, 1978.
- /NEW-82/
Newman, P.S.: "Towards an Integrated Development Environment", IBM Systems Journal, Vol. 21, No. 1, 1982.
- /ORT-85/
Ortman, L. och Axelsson, L.: "En ny utvecklingsmodell -- Varför?", i "NordDATA 85 Konferencedokumentation", Del 1, København, 1985.
- /RAM-85a/
RAMATIC Användarmanual (Version 1.0), SISU, Göteborg, 1985.
- /RAM-85b/
RAMATIC Tutorial (Version 1.0), SISU, Göteborg, 1985.
- /RAM-86/
RAMATIC System Administrator's Guide (Version 1.0), SISU, Göteborg, 1986.
- /REI-84/
Reiner, D., et al.: "A Database Design and Evaluation Workbench: Preliminary Report", Proc. SPOT-3, SYSLAB, Chalmers Tekniska Högskola och Stockholms Universitet, Göteborg, 1984.
- /SAK-82/
Sakamoto, J.G. och Ball, F.W.: "Supporting Business Systems Planning studies with the DB/DC Data Dictionary", IBM Systems Journal, Vol. 21, No. 1, 1982.
- /SCH-82/
Schneider, H.J. och Wasserman, A. (Eds.): "Automated Tools for Information Systems Design", Proc. IFIP WG8.1 Working Conference on Automated Tools for Information Systems Design and Development, New Orleans, North-Holland, 1982.
- /STE-74/
Stevens, W.P., Meyers, G.J. och Constantine, L.L.: "Structured Design", IBM Systems Journal, Vol. 13, No. 2, May 1974.
- /TOR-84/
Torbjär, H.: "The Graphics Subroutine Package: Support for Programming and Interaction in Ramatic", SYSLAB, Chalmers Tekniska Högskola, Göteborg, 1984.
- /USE-85a/
Dataflow Diagram Editor (DFE) User Manual, Interactive Development Environments, San Francisco, CA, 1985.
- /USE-85b/
Structure Chart Editor (SCE) User Manual, Interactive Development Environments, San Francisco, CA, 1985.
- /USE-85c/
Transition Diagram Editor (TDE) User Manual, Interactive Development Environments, San Francisco, CA, 1985.
- /WAS-85/
Wasserman, A.I., et al.: "Developing Interactive Information Systems with the User Software Engineering Methodology", rapport presenterad vid IFIP WG8.1 Working Conference on Environments to Support Information System Design Methodologies, Bretton Woods, NH, USA, Sept. 4-6, 1985.
- /WEI-78/
Weinberg, V.: "Structured Analysis", Yourdon Press, New York, 1978.
- /YOU-75/
Yourdon, E. och Constantine, L.L.: "Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design", Yourdon Press, New York, 1975.

Appendix

UTKAST TILL EGENSKAPSRAM

I det följande skisseras ett antal egenskapsgrupper/egenskaper för karaktäristik av grafiskt baserade datorstöd för systembeskrivning. Observera att det är ett utkast. Vidare bearbetning kommer att ske.

1. ALLMÄNNA EGENSKAPER

1.1 Namn

Fullständigt namn på datorstödet med förklaring av ev förkortningar.

1.2 Utvecklare

Företag, universitet, institution etc, som har utvecklat datorstödet.

1.3 Utvecklingsstatus och historik

Vilket utvecklingsmässigt status har datorstödet: specificerat, under utformning och konstruktion, på prototypstadiet, marknadsförd produkt, etc?

Utvecklingsbakgrund: När påbörjades utvecklingen, hur har det vuxit fram, vidare utbyggnadsplaner, etc?

1.4 Spridning

Vilken spridning har verktyget fått: antal installationer, antal tillämpningar, etc?

Fördelning Sverige/Norden/utom Norden.

1.5 Anskaffning

Om jag är intresserad av att anskaffa verktyget eller att få veta mer om det, vart vänder jag mig? Om produkt - av vilka marknadsförs det?

Ungefärlig kostnad för anskaffning. Ev restriktioner för anskaffning/utnyttjande.

2. HUVUDTYP

För att snabbt få en första uppfattning om verktyget, kan man här, som en sammanfattning av nedanstående, inränga det bland några få centrala huvudtyper av datorstöd. Kategorierna kan vara 'korsande' dvs behöver inte ömsesidigt utesluta varandra. T ex:

- Grafiska vs icke-grafiska
- Statiska vs dynamiska
- Verktyg för enbart beskrivning vs verktyg för beskrivning och produktionskörning (prototyping)
- Generella vs dedikerade verktyg
- Skisseringsstöd/modelleringsstöd/design-stöd/förvaltningsstöd
- Enbart beskrivande vs mer 'intelligenta' verktyg.

3. SYSTEMUTVECKLINGSPROCESS

Avsikten är här att ge en bild av hur datorstödet kan relateras till en systemutvecklingsprocess. En enkel referensmodell kunde kanske ligga i botten:

- modellering (verksamhet, konceptuell, information, data)
- design/utformning
- prototyping
- konstruktion
- förvaltning.

3.1. Underliggande systemutvecklingsmodell

Förutsätter datorstödet någon speciell systemutvecklingsmodell? I så fall vilken och hur ser modellen grovt ut (etapper, faser, etc)?

3.2. Plats i systemutvecklingsprocessen

Var någonstans i den enkla generella resp. i den speciella systemutvecklingsmodellen är datorstödet avsett att användas? Finns interface till andra verktyg för andra moment i processen (beskrivs utförligare under "Interface till andra verktyg" nedan).

4. ANVÄNDNINGSSITUATION

Vilka är de tänkta användarkategorierna?

Vilka är de primära användningssituationerna för verktyget?

Hur kommer användarna att arbeta:

- grupparbetsstation
- en som sysslar med dokumentation
- kan flera användare köra mot verktyget samtidigt?

5. DEDIKERAT ELLER GENERELLT DATORSTÖD

Är datorstödet helt dedikerat till/skräddarsytt för en speciell beskrivningsteknik eller är det generellt och kan i princip stödja vilken beskrivningsteknik som helst?

5.1. Vilka beskrivningstekniker stöds?

Om skräddarsytt - vilken/vilka beskrivningstekniker?

I vilken omfattning stöds denna/dessa tekniker: grafer, tabeller, text, spec formulär, formella notationer, etc?

5.2. Vilka beskrivningsobjekt hanteras?

Vilka 'beskrivningsobjekt', relationer och attribut för dessa förutsätter datorstödet därmed?

5.3. Hur definieras egna beskrivningstekniker?

Hur går det till att definiera en ny teknik alt. omdefiniera/komplettera en tidigare definierad teknik?

Språk för detta?

Vem/vilka kan/bör/får göra detta?

5.4. Kopplingar mellan beskrivningstekniker

Om flera beskrivningstekniker stöds - kan de 'kopplas' till varandra? Hur ser dessa 'kopplingar' ut? Språk för detta?

6. INTERAKTION MED DATORSTÖDET

6.1. Språk för interaktion

6.1.1 Inmatning

Hur matas strukturerna in?

Finns alternativa sätt?

6.1.2. Utsökning

Vilka typer av utsökningsfunktioner finns?

Typ av språk för utsökningar?

6.2. Expertnivåindelad dialog

6.3. Ångermöjlighet

7. GRAFIK

Eftersom inriktningen är på **grafiskt baserade** datorstöd bör de grafiska faciliteterna (i den mån de finns) presenteras mer detaljerat.

7.1 Grafiska manipuleringsfaciliteter

7.1.1. Olika typer av kopplingar mellan bilder

Hierarkiska kopplingar?

Utvidgningsorienterade kopplingar (bilder i samma plan)?

7.1.2. Strukturer oavsett bild

Finns faciliteter för att definiera grupper av symboler?

Finns faciliteter för att ha flera typer av relationer mellan fria symboler?

7.1.3. Översikter över grafer

7.2. Flera bilder samtidigt

Flera skärmar.

Fönsterteknik.

Fönsterteknik: uppdatering i alla fönster.

7.3. Automatisk generering av bilder

Kan datorstödet generera bilder från en strukturbeskrivning?

Hur avancerad är denna bildgenereringsfunktion: finns begränsningar betr storlek av struktur, typ av struktur, etc?

Kort om använda utplaceringsalgoritmer. Utvecklingsmöjligheter, planer.

7.4. Grafisk utrustning

Används något grafiskt mjukvarupaket? Bygger detta i så fall på någon standard eller på något standardförslag?

Vilken grafisk utrustning förutsätts? Grad av beroende till grafisk utrustning.

7.5. Grafer på hardcopy-enheter

8. VALIDERING OCH VERIFIERING

8.1. Analyser och kontroller

Vilka typer av regler och kontrollmöjligheter över strukturerna ges?

Olika typer av syntaxkontroller? Kontroller av mer semantisk, 'intelligent' karaktär?

Hur definierar jag önskade kontroller?

In- och urkopplingsbarhet av kontroller?

Diagnostiska rapporter/korsreferenslistor.

Konsekvensändringar.

Hur uppträder verktyget när något i en struktur bryter mot en regel ('mjukhet'/ovillkorlighet)?

9. RAPPORT- OCH DOKUMENT-GENERERING

10. INTERFACE TILL ANDRA VERKTYG

Finns faciliteter för att skapa t ex en struktur eller ett schema för en databas av någon typ?

Faciliteter för att skapa en körbar modell av någon typ?

Faciliteter för att flytta strukturer till annat verktyg som sørjer för ovanstående?

11. DATA DICTIONARY-FUNKTIONER

Kan man lagra sådana typer av uppgifter om modelleringsfenomen som görs i data dictionary-system?

Finns möjligheter att exportera till eller importera från annat verktyg av dictionary-typ?

12. MASKIN- OCH OPERATIVSYSTEMMILJÖ

Körbar på vilka maskiner och operativsystem?

Om man kan köra på olika maskiner, är prestandaskillnaderna märkbara?

12.1. Maskinstorleks- och prestandakrav

12.2. Flyttbarhet mellan operativsystem

13. HJÄLP- OCH LÄRANDEFUNKTIONER

Handhavandestöd - modelleringsstöd - metodstöd.

14. AUKTORISATION

15. LOGGNINGS-, BACK UP-, OCH RECOVERY-FACILITETER

16. VERSIONSHANTERING

Historik, versioner, dubletter, kopior.

Olika 'färdighetstillstånd' hos grafer eller modellstrukturer.

17. ANIMERINGSMÖJLIGHETER - DYNAMIK

Har datorstödet animeringsmöjligheter dvs kan det åskådliggöra dynamiska skeenden eller förlopp? Hur? Typer av dynamik?

18. SPECIELLA EGENSKAPER

Här beskrivs olika speciella egenskaper som är utmärkande för verktyget, men som inte har kunnat inrangeras under någon av de övriga rubrikerna.

19. REFERENSER

19.1. Underlag för beskrivningen

Vilket material har använts för att göra denna beskrivning? Rapporter, broschyrer, manualer, artiklar, intervjuer, demonstrationer, etc.

19.2. Ytterligare referenser

