



I det här numret presenteras vår intressant Programator och några kända profiler därifrån.

T.v. ovan Per Tidén och Håkan Friberg, som är kontaktpersoner till SISU. T.v, Mats-Åke Hugoson. Ovan, Kalle Wigander.

Innehåll

- | | | | |
|---|---|--|----|
| • Inledaren | 1 | • Programator profilerar per Reflex; | 14 |
| • Bruksanvisning för detta nr | 1 | <i>Lars Bergman, SISU</i> | |
| • SISU kalenderium | 1 | • Dags att förnya systemtänkandet, | 16 |
| | | <i>Kalle Wigander, Progrmator;</i> | |
| | | <i>Lars Bergman, SISU</i> | |
| • Metoder och datorstöd i samverkan "CASE löser de problem du inte vet att du har"; | 2 | • Det tar lika lång tid att ändra info-system som att utkämpa andra världskriget - | 19 |
| <i>Lars Bergman, SISU</i> | | <i>Mats-Åke Hugoson;</i> | |
| • Datorstöd för systemutveckling; | 5 | <i>Lars Bergman, SISU</i> | |
| <i>Håkan Lövgren, Volvo Data</i> | | • Reserapport från USA o Canada; | 21 |
| • Framgångsfaktorer och förutsättningar vid datorisering inom AU; | 9 | <i>Erik Knudsen, SISU</i> | |
| <i>Anders Persson, Volvo Data</i> | | • Kompetensnät för Objektorienterad Systemutveckling, en inbjudan; | 23 |
| | | <i>Stefan Britts, SISU</i> | |

SISU Informa utges av Svenska Institutet för Systemutveckling.

Ansvarig utgivare: Janis Bubenko jr, tel 752 16 00.

Redaktionen: Lars Bergman & Marianne Sindler

Adress: Box 1250, 164 28 Kista. Besöksadr: Electrum, Kista. Tel. 08- 752 16 00. Fax: 08- 752 68 00

Inledaren

Att datorstöd för systemutveckling är en högaktuell fråga för många ADB-verksamheter kunde man nog känna i luften, men att det var så hett kom som en glad överraskning för oss, SSI Göteborg, Volvo Data Skolan och SISU, som arrangerade konferensen "Metoder och datorstöd i samverkan". På SISU går vi nu vidare med CASE-LAB och med CASE89.

CASE-LAB kommer att bli en verksamhet som omfattar ett antal delprojekt som syftar till att höja kunskapsnivån och att sprida kunskaperna snabbt inom området metoder och datorstöd för systemutveckling. Än är vi i diskussioner om verksamhetens utformning och ser fram mot en presentation, från Mattias Hällström, SISU, när inriktning och former klarnat. Vi har redan en seminarieserie kring detta i gång samt har börjat en ytkartläggning av metod- och verktygsanvändning i praktikfältet.

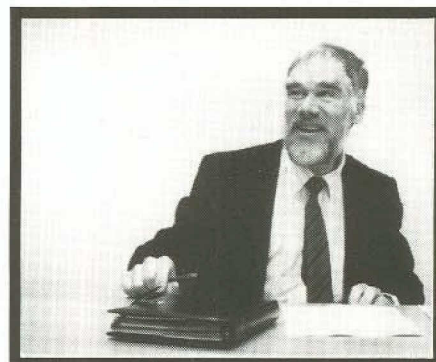
CASE89-konferensen som går 9-11 maj 1989 kommer självklart att omfatta samma område, bl a. Vi vill påminna om att ännu är det inte för sent att komma med bidrag. - Vi ser det som viktigt att du som representerar kunskap från praktikfältet bidrar med dina kunskaper och erfarenheter.

Ett annat område som kommer att aktiveras är Objektorienterad Systemutveckling, där Stefan Britts inbjuder till gemensam kunskapsutveckling i det kompetensnät han tänker bygga kring det temat. Närmare information om detta finner du längst bak i detta nr.

Dataadministrationsprojektet börjar nu nalkas resultat och vi räknar med att komma med en första information kring detta i nästa nummer.

Lars Bergman

Vår nye VVD, Björn Nilsson



Bruksanvisning för detta nummer:

• "CASE-verktygen löser de problem du inte vet att du har", refererar kortfattat föredragen vid konferensen "Metoder och datorstöd i samverkan".

• "Datorstöd för systemutveckling" ger en systematisk klassificering av olika verktygstyper och en bedömning av användning och trender.

• "Framgångsfaktorer och förutsättningar vid datorisering inom AU" ger en handlingsmodell för vad man skall uppnå och varför när man går in i skedet datorisering av AU.

• "Programator profilerar per Reflex" presenterar vår intressent Programator med tonvikt på metodsidan. Detta kompletteras med två "kändisporträtt" av Kalle Wigander och Mats-Åke Hugoson, som båda verkar inom Programator.

• "Dags att förnya systemtänkandet" presenterar Kalle Wigander och tankegångar som han har aktuella.

• "Det tar lika lång tid att ändra infosystem som att utkämpa andra världskriget" presenterar Mats-Åke Hugoson med tonvikt på den ansats för systemstrukturering han pläderar för.

• "Reserapport från besök i USA och Canada" redovisar intryck som Erik Knudsen, SISU:s KBS, samlat.

• Inbjudan till deltagande i kompetensnät för Objektorienterad Systemutveckling, presenterar närmare varför, vad och hur kring detta.

SISU kalendarium

Oktober

27

CASE i praktiken
Olga de Troyer från INFO-LAB, Holland, presenterar en ansats för databasdesign, som även inkluderar CASE-verktyget RIDL*.

November

25

CASE i praktiken
TECSIEL-seminarium.

Maj 1989

CASE89

Anmäl dig och din kunskap till konferensen!
- Vi vill gärna ha fler praktikerföredrag.
Skriv eller ring till Björn Nilsson.

"CASE"-verktygen löser de problem du inte vet att du har!

Vi står genom "CASE"-verktygen, (Computer Aided Software Engineering eller datorstöd för systemutveckling) inför stora förändringar inom systemutvecklingsområdet. Behovet av och möjligheterna för drastiska förändringar av sättet att driva systemutveckling är mycket stora. I år har antalet installerade "CASE"-verktyg ökat väsentligt i Sverige om man får tro leverantörerna. 1991 har antalet installationer ökat med 50% anser en expert. Om det 1987 fanns 20.000 installationer så skall det ses mot en potentiell marknad av säg 20.000.000 installationer.

Yrkesrollen kommer att förändras kraftigt och kraven på kompetenshöjning t ex när det gäller analysmetoder poängteras av flera experter inom området liksom av de praktiker som delgav erfarenheter.

Den kraftigt övertäckade konferensen, arrangerad av SSI Göteborg, Volvo Data Skolan och SISU, markerade i sig att området nu är "hett" här. 240 deltagare, 20-

talet föredragshållare och 10-talet utställare fyllde

av Lars Bergman, SISU

Volvohallen till en aktivitet som kommer att stanna i minnet som den mest välarrangerade i mannaminne. Detta mycket tack vare Ann Nilsson, Ann-Kristin Larsen, Eva Lövgren och Aleksander Ratz från Volvo Data samt Olof Björland från AV-teknikenheten i Volvohallen, vilka svarade för de praktiska arrangemangen på plats.

Göran Kling, tillträdande VD för Volvo Data inledde med att deklarerade: Det är kompetenta människor, tid och pengar som är begränsande faktorer. Trots stora insatser under en lång tid minskar inte behovet av applikationsutveckling. När det gäller produktiviteten klarar vi traditionellt att öka den med 5-10% årligen. Detta räcker inte utan vi måste nå en årstakt av 15-25%.

Datorisering av AU/ADB är nästa "våg". Vågen efter datorisering är troligen ett helt nytt arbetssätt med helt ny metodik. Det brukar dock dröja 10 - 20 år innan en ny våg slår igenom, hävdade Anders Persson, metodchef, Volvo Data.

Att ta till sig ett avancerat datorstöd inom AU/ADB utan att först till-

lämpa och förstå arbetsmodeller, modellering etc, kommer att sluta med HAVERI. Det finns många exempel på misslyckanden. Här krävs mognad och mognad kräver resurser.

Ett huvudproblem som vi har är bristen på kunniga personer inom analys och modellering (funktions- och datamodellering) Vi söker här generalister som både är analytiskt, pedagogiskt och tekniskt kunniga. De datorstöd som finns och kommer att finnas under en femårsperiod, kommer att kräva expertkunskap som är både analytiskt och tekniskt betingad. Vi kan här tala om kunskapsingenjörer.

Håkan Lövgren, seniorkonsult, Volvo Data gav en överblick över området i en inspirerad och inspirerande dialog med Ander Persson: Om man ser sig omkring på marknaden för "datorstöd för systemutveckling" finner man en nästan oändlig mängd av produkter. Innehållsdeklarationen är ofta dålig och likheten mellan två produkter som säljs under samma benämning kan vara närmast obefintligt.

Medan 4GL och applikationsgeneratorer inriktar sig på att effektivisera programmeringen så inriktar sig CASE, Computer Aided Software Engineering, på att höja kvalitet och effektivitet på analysarbetet. Software Engineering säger oss att det skall handla om ingenjörsmässigt utvecklande av programvara. I ordet "ingenjörsmässigt" ligger att man skall arbeta med ritningar, beräkningar etc.

Konferensen: Metoder och datorstöd i samverkan

Dessutom skall dessa ritningar och beräkningar utföras på ett sådant sätt att andra än upphovsmannen eller -kvinnan kan ta del av dem och verifiera att de är riktiga.

Detta leder fram till att ett CASE-verktyg skall vara inriktat mot specifikationsarbete och vara baserat på en förståelig metod eller metodkedja. Ser man på denna typ av verktyg har de så gott som alltid följande karaktär; körs på arbetsstation (kraftig PC), har grafiskt interface, nyttjar mus, har ett integrerat data dictionary och utför kontroller på det som matas in. Vanligen stöder verktyget metoder för datamodellering (normalt EAR, Entity-Attribute-Relationship) samt funktions-processmodellering (flödesmodell och/eller hierarkisk nedbrytning).

CASE kan ge mycket stora positiva effekter i analysarbetet. Framst genom förbättrade analyser och genom kontroller av analysarbetet.

Arbetsprocessen vid SU och dess uppbyggnad av metoder och aktuella datorstöd för dessa inom SAS, beskrevs av **Arne Nordgren**, under temat datorstöd för effektiv administrativ utveckling.

Vi tror att effektiviteten hos en utvecklingsmiljö handlar mer om en väl fungerande helhet än om något superverktyg som rationaliserar endast en del av utvecklingsprocessen, deklarerade **Ann-Charlotte Ohlsson**, ADB-kontoret i Göteborg. Vi kommer därför även i framtiden att välja verktyg som smidigt kan anpassas till och förbättra den metodkeja och den utvecklingsmiljö vi tror på.

ADB-kontoret i Göteborg har under de senaste 20 åren konsekvent satsat på en genomtänkt metod för utveckling av system. Stor vikt läggs vid införande av hjälpmedel och mycket tid läggs på utbildning och introduktion när nya verktyg införs.

Vår strategi ger oss en stabil grund att stå på när det gäller utvärdering och val av nya metoder och verktyg. Vi ser resultat av våra satsningar i kortare utvecklingstider, mer personoberoende system och i att det är lätt att växla mellan systemutveckling i IBM- och Digitalmiljö.

Att ha ett heltäckande och fungerande data dictionary är nog många

man tvingar fram frågor tidigt. Persontiden har minskat medan maskintiden är ungefär lika.

För förvaltningspersonalen är konsekvenserna bl a att man får "bättre system" och alltså mindre underhåll. Man får överblick och därigenom en bättre ändringsberedskap. Det är enklare att byta systemansvarig. Även här har man minskat



Michael Brodie från GTE Labs, USA, redovisade material från en undersökning av CASE, där han funnit 110 verktyg och i studien har med ca 60. Michael prognosticerar en marknad på 20 miljoner verktygsinstallationer. (Foto: Conny Nylén)

ADB-enheters dröm. - Volvo Lastvagnar har realiserat den drömmen. **Tore Altenstedt** vid Volvo Lastvagnars systemplanering presenterade erfarenheterna av detta som för systemutvecklingspersonalen är bl a att man nu har överblick över befintliga system, databaser och bakomliggande datamodeller. Dessutom har man kontroll över nya systems komponenter. Sena ändringar underlättas och man får viss genereringshjälp och viss, enhetlig, systemdokumentation fås automatiskt. Man har ett stöd för ett "rätt" arbetssätt bl a genom att

erforderlig persontid medan maskintiden är lika.

Valet av strategi beror av ett antal situationsfaktorer, påpekade **Janis Bubenko Jr**, SISU (Svenska Institutet för Systemutveckling). En grupp sådana karaktäriserar organisationens systemutveckling, "mognad", policy, attityder och personalens kompetens och kompetensutveckling. En annan karaktäriserar organisationens metodutvecklingspolicy, mognad och strategi. En tredje utgör relationer till

Konferensen: Metoder och datorstöd i samverkan

och tillgång till kunniga CASE-leverantörer eller konsulter.

Intressant är utvecklingen från metodbundna CASE-verktyg till s.k. "CASE-skal" vilka möjliggör anpassning av verktyg till önskad metod. Därigenom förs frågan om val över mer till metodval än till verktygsval.

AI, artificiell intelligens, har som gemensam nämnare aktiviteter som förknippas med mänskligt beteende. AI kommer att påverka AU/ADB-verksamheten på flera områden. Kunskapen i systemet kommer att fokuseras vilket påverkar utvecklingsprocessen, databasdesign och användargränssnitt. Framtid krav på datoriserade hjälpmedel för integrering av databas- och kunskapssystem kommer att resas. I-CASE-system kan i sig utformas som kunskapssystem. Dethär inlägget skall ses som en väckarklocka för systemutvecklare, poängterade Jean Peter Fendrich, Volvo Data, alla kommer att påverkas.

Forskningen pekar på att man i framtiden kommer att ha ett antal olika verktyg från AI-området till sin hjälp i systemspecificering. För närvarande kan man se tre typer av AI-tekniker, naturligt språkbaserad,

formulärbaserade samt regelbaserade. Genom AI-stödet kan vinnas ökad kvalitet, ökad uttrycksfullhet och minskad utvecklingstid påpekade Rolf Wohed, SISU, som arbetar med en prototyp för kvalitetsdiagnos av konceptuella modeller gjorda i RAMATIC.

"CASE solves the problems you didn't know you had".

Den kritiska teknologin för CASE-verktyg är data dictionary. Michael Brodie från GTE Labs, USA, redovisade material från en undersökning av CASE där han funnit 110 verktyg och i studien har med ca 60. Dessa har en spridning på 20.000 installationer. Han förväntar sig en tillväxt på 50% till 1991. Och prognostiserar en marknad på 20.000.000 verktygsinstallationer. Vi är bara i början. Om fem år kommer verktygen att se mycket annorlunda ut liksom leverantörsstrukturen.

Kunskapsutveckling är en central komponent i stegen mot användning av CASE-verktyg anser både Ulla Frithiof, Ericsson och Stig Johansson, Volvo Personvagnar. De deltar i var sitt projekt i samverkan med SISU där CASE-skalsprototypen RAMATIC används och vidareutvecklas. Lars-Åke Johansson,

SISU, instämde villigt i den synpunkten.

IBM:s satsningar inom datorstöd, presenterades av Sten Kallin, IBM och DIGITAL:s syn på integrerade hjälpmedel för utveckling och förvaltning av Björn Lannerö, DIGITAL. IBM siktar på den långsiktiga utvecklingen via arkitektur och Digital koncentrerar sig på miljö för implementering och CASE-verktyg i samarbete med software-leverantörer.

Från analys till konstruktion och körbart system med IEF demonstrerades av Anders Romell, Volvo Data och Bill Gibson, JMA. Presentationen gav en bra överblick över hur ett CASE-verktyg kan användas och hur utvecklingsprocessen genomförs fram till körbart system.

Managerprodukterna presenterades av Clark Sjöwall och Garry Hall, MSP. Datamanager är kanske den mest spridda data dictionary-produkten bland IBM-s datororkunder. Nu presenterades påbyggnad med CASE-verktyg som arbetar mot data dictionary.

Datorstöd för systemutveckling

Om man ser sig omkring på marknaden för "Datorstöd för systemutveckling" finner man en nästa oändlig mängd av produkter.

av Håkan Lövgren,
Volvo Data

(artikeln är baserad på föredrag från konferensen: "Metoder och datorstöd i samverkan")

Innehållsdeklarationen är ofta dålig och likheten mellan två produkter som säljs under samma benämning kan vara närmast obefintlig.

Några av de vanligast förekommande typbeteckningarna är:

Applikationsgenerator
4GL
CASE
I-CASE

Syftet med artikeln är att ge en bild av hur de olika produkttyperna är relaterade till varann, samt vilka krav man bör ställa på respektive typ. Dessutom beskrivs de trender som för närvarande finns på datorstödsmarknaden.

På vissa ställen nämns enskilda produkter vid namn. Detta görs endast för att exemplifiera och skall ej tolkas som om dessa produkter är bättre än övriga av samma typ.

4GL

4GL är en marknadsföringsterm som används på nästan vad som helst inom datorstödsområdet i syfte att göra kunderna intresserade. Man kan dock särskilja två huvudtyper av 4GL:

Ersättare till COBOL (3GL)

Inom detta område finner vi ett antal produkter avsedda att nyttjas av systemutvecklare. Produkterna har ibland en integrerad databashanterare. Exempel: Ideal (ADR).

Information Center-produkter

Denna typ av produkter riktar sig till slutanvändare med viss SUI-kunskap. Tanken är att användarna själva skall framställa mindre applikationer, samt göra informationsuttag ur existerande register. Produkterna i denna kategori är ofta enklare att använda än "COBOL-ersättarna" men har å andra sidan ofta mindre flexibilitet och är ibland dyra att nyttja (hög resursförbrukning). Exempel: SAS (SAS Institute Inc.).

Applikationsgeneratorer

Det finns en annan typ av produkter inom samma område, de s k applikationsgeneratorerna (AG). Dessa är egentligen direkta konk-

urrentprodukter till 4GL, men saluförs ofta under just beteckningen 4GL. Skillnaden mellan AG och 4GL är att AG genererar kod (vanligen i 3GL, exempelvis COBOL). 4GL genererar ingen kod, de är språk i sig.

Karaktäristik för AG och 4GL.

Ur användarens (systemutvecklarens) synpunkt brukar 4GL av typ 1 (COBOL-ersättare) och AG te sig mycket lika och innehålla ungefär samma funktionalitet.

De bör åtminstone innehålla verktyg för att hantera:

- . Databaser
- . Process/program
- . Dialog-skärm (inklusive hjälpbilder)
- . Rapporter

Alla delar skall vara tillgängliga via en gemensam editor och ett gemensamt "språk".

Dessutom skall ett integrerat "Data Dictionary" finnas som håller ordning på alla komponenter och stödjer programmeraren så att han/hon inte behöver ge samma uppgifter/information flera gånger.

För och emot 4GL/AG?

Produkterna angriper två huvudproblem. Det första är låg kvalitet på utvecklade system och det andra är låg utvecklingstakt.

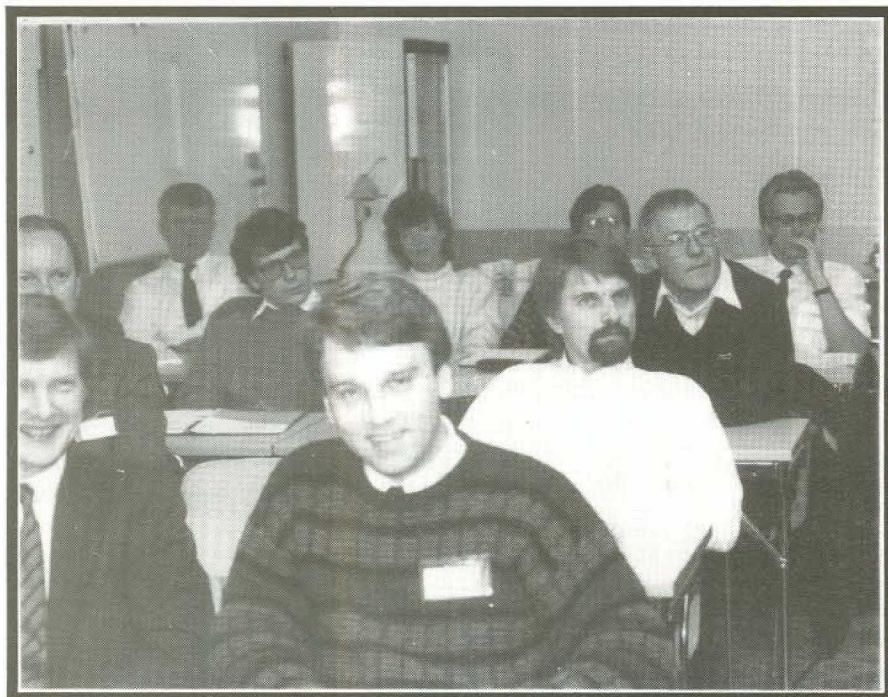
Det första problemet angrips i huvudsak kgenom nyttjande av prototyping, vilket ökar användarmedverkan och därmed ökar chansen att producera rätt system.

Datorstöd för systemutveckling

Det andra problemet angrips genom programmering på högre nivå än exempelvis COBOL. Detta görs oftast genom en kombination av att en så stor del som möjligt av systemen skrivs i ett resultatorienterat språk och övriga delar skrivs med hjälp av proceduriellt språk. (Olika leverantörer hävdar att programmeringsfasen effektiviseras med mellan 250% och 1000%, siffror man

(ofta omedvetet) finna fel på ett nytt verktyg.

Ett annat problem är performance. Denna varierar kraftigt mellan olika verktyg och man kan därför inte döma alla verktyg över en kam. AG har ofta bättre performance än rena 4GL. Den sistnämnda kategorin av verktyg kan ibland öka CPU-uttaget med upp till 100%. Det ökade



Håkan Lövgren (här i förgrunden vid kontaktpersonträff med SISU) är seniorkonsult vid Volvo Data och arbetar bl a med CASE-verktyg som IEF och tillämpning i projekt.

bör ta med en "nypa salt" eftersom de sällan anger vad jämförelse görs mot.) 4GL ger också (i varje fall enligt leverantörerna) minskat underhåll. Detta beroende på att rätt applikationer tas fram (genom prototyping) samt på att det är enklare att underhålla än exempelvis COBOL.

Som alltid har medaljen en baksida. Det finns en del problem med att föra in 4GL/AG.

Ett vanligt problem är att användarna (SU-personalen) inte accepterar produkten. Många har lagt ner en stor del av sitt liv på att förstå och på bästa sätt kunna nyttja COBOL-världen. Det är då lätt att

CPU-uttaget skall ställas i relation till vad man vinner i programmeringseffektivitet.

Vad gäller programmeringseffektiviteten bör man nog tänka efter två gånger innan man köper 4GL/AG.

För det första är COBOL långt ifrån en död produkt. Utveckling pågår ständigt och moderna editorer (ex. ISPF) tillsammans med integrerade testhjälpmedel gör att COBOL blir mer attraktivt (och 4GL/AG mindre attraktivt). En annan faktor som talar för COBOL är de nya COBOL arbetsstationerna (PC). Som exempel på dessa kan Micro Focus och Realia-COBOL nämnas.

CASE

Den allvarligaste invändningen mot 4GL/AG är kanske ändå den som kommer från CASE-anhängarna och som kallas "4GL's Moment-22".

Den går i princip ut på att 4GL/AG attackerar fel grundproblem. Om man enögt ger sig på programmeringsfasen kommer detta att leda till att antalet dåligt analyserade applikationer/system ökar explosionsartat, vilket i sin tur ger en explosionsartad ökning av underhållet, vilket i sin tur leder till att alla kommer att syssla med underhåll.

Lösningen är istället enligt CASE-leverantörerna att koncentrera sig på analysfasen.

Innan vi går vidare är det nog på sin plats att placera in CASE i förhållande till 4GL/AG.

4GL/AG

Inriktar sig på att effektivisera programmeringen och underhållsarbetet genom att höja programmeringsspråkets nivå.

CASE

Inriktar sig på att höja kvalitet och effektivitet på analysarbetet genom att nyttja datorstöd. CASE attackerar ej programmeringen i sig utan anser denna vara en avslutande "bi-syssla", dvs en följd av analysarbetet. Om 3GL eller 4GL nyttjas spelar ingen roll. Effektiviteten ligger i att bygga rätt system.

Karaktäristik för CASE

Att definiera CASE genom att se på de produkter på marknaden som kallar sig CASE är besvärligt. Man kan finna nästan vilken typ av produkter som helst.

Ett bättre sätt är att se vad orden Computer Aided Software Engineering bör innebära.

Datorstöd för systemutveckling

"Computer Aided" säger oss att det skall vara ett datorbaserat stöd/hjälp-verktyg.

"Software Engineering" säger oss att det skall handla om ingenjörsmässigt utvecklande av programvara. I ordet "ingenjörsmässigt" ligger att man skall arbeta med ritningar, beräkningar etc. Dessutom skall dessa ritningar och beräkningar utföras på ett sådant sätt att andra än upphovsmannen eller upphovskvinnan kan ta del av dem och verifiera att de är riktiga.

Detta leder fram till att ett CASE-verktyg skall:

- Vara inriktat mot specifikationsarbete
- Vara baserat på en förståelig metod eller metodkedja

Om vi ser oss runt efter denna typ av verktyg finner vi att de så gott som alltid har följande karaktäristik:

- Körs på arbetsstation (PC)
- har grafiskt interface
- Nyttjar "mus"/"sork"
- Har ett integrerat data dictionary
- Utför kontroller av det som matas in

Man kan lite förenklat säga att det de flesta CASE-leverantörer gjort är att åstadkomma ett datoriserat ritbord för en metod eller en metodkedja som är arbetssam att hantera manuellt.

Vanligen understödjer CASE-verktygen:

- Datamodellering (normalt EAR)
- Funktions-process-modellering (flödesmodell och/eller hierarkisk nedbrytning)

Effekter av CASE

CASE kan ge mycket stora positiva effekter i analysarbetet. Det är, hur dumt det än må låta, många analyser som inte görs helt klara/kompletta helt enkelt därför att det blir

för arbetssamt att hålla på och uppdatera analysresultatet. Här har CASE en av sina stora styrkor, det är enkelt att ändra i analysresultatet.

Ett annat problem med manuell analysarbete är att man ibland inte kan överblicka hela analysresultatet. Detta leder ofta till att det finns motstridigheter i analysen, eller till att den inte blir komplett. Resultatet blir en inkonsistent kravspecifikation. Även här har CASE-verktygen en stor styrka. Verktygen innehåller oftast konsistenskontroller och funktioner som "larmar" om någon del av analysen ej är komplett.

Även för CASE finns det dock problem. I samband med 4GL nämndes att användarna (systemutvecklarna) ofta omedvetet gjorde motstånd mot verktygen. Detta gäller även för CASE. Speciellt gäller det om verktygets metodkedja inte överensstämmer med den som tidigare nyttjats.

En annan fara med CASE-verktygen är att man kan luras att tro att de utför er än de egentligen gör. Det finns ingen möjlighet att lyckas med CASE-verktyg om inte användarna har god utbildning i metoderna och erfarenhet av att nyttja dem. (En enkel jämförelse: Vem som helst kan ju inte konstruera en VOLVO bara för att CAD-verktyg införts.)

Det sista problemet med CASE som tas upp här är att verktygen ofta upplevs som allt för isolerade. Det finns sällan möjlighet att koppla ihop dem med den existerande SUMILJÖN på företagen.

CASE-skal

Som nämnts ovan är de flesta CASE-verktyg inriktade mot en specifik metod eller metodkedja. Det finns ibland viss möjlighet att anpassa verktyget så att valfria symboler

kan användas men dt är en fix modell i botten.

En annan typ av CASE-verktyg är de sk CASE-skalen. Beteckningen "skal" har de för att de i princip är tomma. Det är upp till användaren/installationen att lägga in sin metodkedja, sin grafteknik och sina begrepp.

Det kan givetvis i många sammanhang vara klokt att välja ett CASE-skal eftersom man då kan fortsätta att arbeta med de metoder och tekniker man är van vid. Man bör dock betänka att det inte är speciellt många metoder-tekniker som är så stringenta att de kan läggas in i ett datorstöd utan relativt stora insatser.

Det finns inte många produkter på marknaden idag som utger sig för att vara CASE-skal. De flesta är fortfarande på forsknings/utvecklingsstadiet.

För att en produkt skall kunna kalla sig CASE-skal bör den ge användaren möjlighet att:

- Skräddarsy vilka begrepp som skall lagras i dictionary (Metamodell)
- Skräddarsy beskrivningsteknikerna (diagram-grafer)
- Lägga in regler

I-CASE

Den senaste typen av CASE-verktyg på marknaden är de sk I-CASE-verktygen (Integrated CASE).

I-CASE är enkelt uttryckt summan av ett CASE-verktyg och ett 4GL/en AG.

I I-CASE-fallet genereras exekverbar kod (vanligen i 3GL) direkt från analysresultatet.

För närvarande finns det några få I-CASE-produkter på marknaden.

Exempel: IEF (Texas Instruments/ James Martin Ass.)

För att ett verktyg skall kunna kallas I-CASE måste det innehålla integrerade komponenter som täcker hela kedjan från analys till implementerat system. Det skall dessutom vara så att systemförändringar skall utföras genom att man går tillbaka till rätt ställe i metodkedjan, dvs oftast till analysen.

Data-dictionaryt skall hålla information om hela kedjan. Detta leder givetvis till att dictionaryt blir mycket omfattande. För att skilja denna typ av dictionary från mer enkla typer hör man ibland beteckningarna "encyklopedi" och "repository".

Karaktäristik för I-CASE

Det normala för ett I-CASE-verktyg är att:

- Utvecklingsarbetet görs på arbetsstationer
- Encyklopedin ligger centralt
- Generering utförs i central miljö
- De färdiga systemen är körbara i flera miljöer

Effekter av I-CASE

Det är ännu så länge mycket få företag som har erfarenhet av vilka effekter I-CASE ger. Man kan dock på goda grunder anta att effekterna kommer att bli stora. De huvudsakliga skälen till detta är:

- Analysarbete blir nödvändigt och helt avgörande
- Det rena programmeringsarbetet elimineras
- Förändringsarbete innebär ny analys
- Användarna kan eventuellt utveckla sina egna system

Beträffande den sista punkten bör nog sägas att det inte verkar möjligt i dagsläget men att successiva förändringar av dels verktygen och dels företagets organisation kan möjliggöra detta.

Om vi tänker oss att respektive enhet inom företagen får sk "informationsingenjörer", vilka dels har god verksamhetskunskap och dels god metod/verktygskunskap, så kan det ske snarare än de flesta av oss anar. I-CASE är troligen ett steg mot upplösande av centrala SU-funktioner.

Utveckling mot I-CASE

Även om det i dagsläget endast finns ett fåtal I-CASE-produkter på marknaden finns det mycket som talar för att detta är framtidens melodi.

För det första finns det redan idag exempel på direkta samarbetsavtal mellan 4GL/AG- och CASE-leverantörer.

För det andra finns IRDS.

IRDS uttyds Information Resource Dictionary System och är en standard som beskriver/borde beskriva hur ett Data Dictionary (encyklopedi) skall vara strukturerad. Problemet med IRDS-standarderna är att den i dagsläget är ganska svag. Troligen kommer IBM att sätta standarden när deras "Repository" släpps på marknaden. (Det kan vara intressant att notera att IBM har visat mycket stort intresse för IRDS och även visat stort intresse för att åstadkomma en CASE-standard.) Många 4GL/AG-leverantörer sätter sitt hopp till att kunna öppna sina produkter mot CASE-världen genom att följa IRDS. Eftersom standarden idag är svag råder dock ett försiktigt avvaktande.

Det återstår dock att se vad som händer med IRDS. Om standarden blir stark kommer vi troligen att få ett stort utbud av produkter, vilka kan samverka, på marknaden och därmed stor valfrihet. Om standarden ej blir stark kommer troligen det fåtal leverantörer som har resurser att bygga hela I-CASE-produkter eller att sluta samarbetsavtal att styra.

Andra tekniker

Förutom CASE och I-CASE finns det ett antal intressanta tekniker som kan ge oss ökad produktivitet och minskat förändringsarbete. Tillsammans med CASE och I-CASE kan de ge mycket stora effekter.

Bland de mest intressanta kan nämnas:

RE-Engineering: Denna teknik syftar till att ta "gamla ostrukturerade" program och strukturera dem (maskinellt).

Reverse Engineering: Handlar om att utifrån de system vi har idag, baklänges skapa modeller av verksamheten. Man har kommit längst på datasidan, dvs tekniken med att utgående från databasbeskrivningar, vilbeskrivningar etc skapa konceptuella datamodeller. Detta är en intressant teknik eftersom den ger oss en möjlighet att få beskrivningar av existerande system till CASE och I-CASE-verktyg. Man behöver inte fundera länge för att inse vilket starkt planeringsunderlag det skulle vara att ha en ideal bild av verksamhetens system (framtagen med hjälp av ny modellering) bredvid en bild av vad som i dagsläget finns i våra system.

Reusable kod (återanvändbar kod): Detta är egentligen ingen ny teknik. Det handlar om att skriva generella program som går att använda i flera system. Vad som däremot är intressant är att japanerna har börjat att hårt systematisera tekniken och till och med ge speciell ekonomisk ersättning till dem som skriver program som går att använda i flera system. Det kan bli en modell vi kan importera!

Framgångsfaktorer och förutsättningar vid datorisering inom AU

av Anders Persson,
Volvo Data AB

(Artikeln baseras på föredrag vid konferensen "Metoder och datorstöd i samverkan")

Att införa och bruka datoriserade hjälpmedel för AU/ADB-verksamheten är långt ifrån ett trivialt problem. Det finns många exempel på misslyckanden som kan härledas till att man fascinerats av en teknisk produkt, ett teoretiskt resonemang som är praktiskt ogenomförbart eller ett för snävt synsätt.

Hur skall vi se på datorstöd som stödjer AU/ADB-verksamheten? Var kan vi finna potentialer och effektivitetshöjningar? Vilka är de grundläggande förutsättningarna för att lyckas uppnå effekter? Vilka problem står vi inför idag som väntar på att bli lösta?

Dokumentet är noteringar för ett föredrag som belyser ovanstående områden och frågeställningar. Meningen var egentligen inte att dokumentet skulle läsas fristående.

Vad är problemet?

När vi talar om att effektivisera inom informationsbehandlingsområdet, måste det innebära att det finns ambitioner och krav som vi inte kan nå eller leva upp till idag. Att vi inte når dessa måste ju bero på ett antal problemställningar som har bakomliggande orsaker.

Några problem som upplevs är:

- ADB-systemen kan inte svara upp mot de krav som ställs. Här kan det vara funktioner som saknas, en dålig och ineffektiv utformad dialog mellan människa och dator, eller en tekniskt undermålig konstruktion.

- Tröghet och sårbarhet vid förändring. En förändring som är önskad från verksamheten kan innebära stora tekniska konsekvenser, höga kostnader och driftsstörningar.

- Stor "förändringsportfölj". Eftersom det är kostsamt, trögt, farligt och svårt att införa korrekta ändringar

uppstår en naturlig väntekö som är lång även om ändringarna är av trivial natur för verksamheten. Detta har också fört med sig att verksamheten själv inför data-tekniska verktyg (typ persondatorer) och utvecklar själva.

- Klyfta mellan verksamhets- och ADB-teknisk personal. Det förekommer uppenbara svårigheter att förstå varandra samt att översätta verksamhetens krav och behov till tekniska lösningar.

- Höga kostnader vid utveckling och förvaltning. Ett till synes "lätt system" kan ta lång tid att utveckla och kräver en kraftig förvaltning främst på grund av den komplicerade tekniska miljön, som är nödvändig.

- Förståelse och integration av information är bristfällig. Många gånger är det svårt att förstå vad data/information egentligen står för, med vilken kvalitet som den lagras och vem som ansvarar för den. Dessutom är det så att det förekommer data/information som både är synonymer (en sak kallas för två saker) och homonymer (en sak betyder två helt olika saker beroende på vem som tolkar).

Den tekniska miljön eller ett gammalt etablerat programmeringsspråk är inte den bakomliggande orsaken till problemen. Visserligen kan tekniken ha bidragit genom att



Anders Persson är projektledare för SISU-projektet DATAADMINISTRATION som kommer med sin rapport under hösten. Dessutom kontaktperson till SISU och chef för Metod och Utbildning inom Volvo Data.

den är mycket fascinerande, vilket gör att man lätt glömmer verksamheten.

Huvudorsakerna har att göra med vårt sätt att strukturera verksamheten och ADB-lösningar och vårt historiska arv. Företeelsen som allmänt kallas "Spaghettisyndromet" är en naturlig utveckling som alltid uppträder på vägen från en "pionjäranda" till en hög volym och en stor användning.

En mängd, ofta åtskilda, ADB-system har tagits fram för att tillhandahålla en given information. Detta innebär att vi har kostsamma och svårkontrollerbara samband mellan dessa. Antalet nya system ökar kraftigt i takt med nya krav på information i olika former och sammanställningar. Att det ser ut så här beror till stor del av det historiska arvet.

Några av dagens arbetssätt bygger på att utvecklingsprojekt syftar till att lösa ett relativt avgränsat problem i verksamheten oftast avgränsat till den formella organisationens informationsproblem.

Genom att vi arbetar sekvensiellt

med ett avgränsat informationsproblem i taget har det vuxit upp en ganska komplex värld av system för verksamheten. ADB-systemen har stora och starka kopplingar sinsemellan och data som lagras ligger "fastlåst" och kan inte hanteras såsom vi skulle vilja.

För varje nytt informationsbehov kommer system att tas fram eller förändras. Behovet av ny information är stort vilket kräver mycket resurser av ADB-specialistkaraktär. Eftersom varje system begränsats till att fokusera "sin" utdata istället för verksamhetens data/information, finns samma data lagrat i många register och dessutom har de olika aktualitet och förfiningsgrad.

Elakartade begreppsförvirringar råder över samma data därför att olika ADB-system benämner dem med olika namn. Begreppsförvirringar skapar oreda i systemen och ineffektivitet i verksamheten. Detta förstärks ytterligare av ett ofta oklart ansvar för data och information i verksamheten.

Många charmeras av tekniker typ Fjärde Generationens Språk, kon-

torsautomation, CIM-koncept, persondatorer etc. Visserligen kan dessa tekniker lösa enstaka problem och där ge en stor effekt men de löser inte vårt spaghettisyndrom på ett kontrollerat sätt. Det kan tyvärr vara så att dessa tekniker och koncept förvärrar situationen och "bygger på" spaghettisyndromet i ett accelererande tempo. Vi kan inte fortsätta så här, det leder bara till en "Dead End". Det måste finnas andra sätt som skapar en bättre och mer flexibel portfölj som stödjer verksamheten.

Vi måste utgå dels från en stabil grund och ge den dynamik som är önskvärd. Detta måste kunna åstadkommas på ett betydligt effektivare sätt än idag.

Ett helhetsperspektiv

Vad är stabilt och vad är föränderligt i en verksamhet?

Grundläggande begrepp och definitioner av data är mycket stabila. För en tillverkande industri är order, artikel, kund, faktura basbegrepp och måste ha en definition som alla är överens om. Kring dessa basbegrepp byggs hela informationsstrukturen med de register och lokala begreppsvärldar som behövs.

Hanteringen av data kring dessa begrepp, dvs logistiken är relativt stabil men kan förändras då man önskar arbeta enligt helt nya koncept. Exempel på nya koncept inom industriområdet är "just-in-time", "kundorderstyrd produktion".

Arbetsrutiner som utför den logistik som behövs är relativt föränderlig.

Personal som genomför arbetsrutiner är mycket föränderlig och lika så den formella organisation som finns.

Att bygga en "portfölj" av system

Framgångsfaktorer

som utgår från enskilda individers eller organisationers behov är därför tveksamt. Visserligen är det ibland kortsiktigt effektivt, men mer långsiktigt innebär detta en "Dead End".

När vi utvecklar våra system måste vi därför bygga på en stabil kärna av begrepp med data som speglar och förstås av verksamheten.

"Lindat" kring kärnan bygger vi upp den logistik (de funktioner) som behövs. Detta måste vi göra på ett klart funktionellt och avgränsat sätt.

Ytterst lägger vi sedan på det gränssnitt (den dialog), det arbetssätt, den personal och formella organisation som behövs.

Hela tiden är det verksamhetens krav och behov som vi utgår från. Baserat på dessa kan vi sedan nyttja den teknik som behövs.

För att 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.

Många kompetenser såsom verksamhetsmänniskor och olika specialister måste samverka i ett utvecklingsarbete. Vi skulle kunna se detta så här:

- Vad är det för uppgifter som skall genomföras i verksamheten (mål), om vilka data och funktioner behövs?
- Hur skall vi lösa uppgifterna? Skall vi automatisera allt eller delar? Vilken teknik skall brukas?
- Vem, viken kompetens och vilka befattningar behövs?
- Var skall personal och uppgifter finnas? Vilken formell organisation skall finnas, vilka ansvar ges och

var finns detta placerat geografiskt (centralt, lokalt, etc)?

Att underlåta någon del i denna helhet ger konsekvenser.

Detta är vad vi kallar för helhets-syn. Det handlar inte bara om systemutveckling, utan om administrativ utveckling. Alla delar måste samverka.

Strukturering av Informationssystem

Vi skall sträva efter att bygga våra informationssystem både stabila och dynamiska. Hur uppnår vi detta?

Första steget är att göra en väl genomtänkt avgränsning av verksamheten så att vi kan styra den på ett entydigt sätt. Gör vi inte detta får vi en okontrollerad och svår situation som förvärras av de informationssystem som byggs in.

Notera att här handlar det inte om ADB-system utan istället om hur den "nya verksamheten" skall se ut. Ett ledord att sträva efter är autonomitet, dvs logiskt oberoende delar som hänger samman på ett kontrollbart sätt. Här krävs AU-planering.

Vi skall sträva efter att strukturera våra system så här:

Datahållande och logistikbaserade delar

- Den stabila delen bestående av begrepp och definitioner av data i verksamheten är fundamentet. Här skall vi ta fram en datamodell som speglar vilka behov som finns och med vilken kvalitet som denna data måste ha.
- De basfunktioner som hanterar data, dvs den logistik (funktioner) som fastställs måste integreras med datamodellen.

Informationshanterande och informationsproducerande delar

Utsökningar och sammanställningar av information skall svara mot flexibilitet och föränderlighet. Detta bör helst avskiljas från ovanstående och dessutom konstrueras med teknik som ger möjlighet till ändringar.

- De datahållande och logistikbaserade delarna kan mycket väl använda sig av traditionell teknik, typ "gamla COBOL"-språket. Här är vi vana vid att arbeta på traditionellt sätt.

- Informationshanterande och informationsproducerande delar lämpar sig mycket väl för fjärde generationens tekniker, persondatorer etc. Här kan vi tala om IC-verktyg som helst bör styras av verksamhetspersonal. Endast vid komplicerade dialoger behövs specialistkonstruktion.

Genom att göra så uppnår vi följande:

- Om vi kan formalisera data och logistik entydigt kan vi minska kunskapsglappet mellan verksamheten och specialisten. Vi kan också automatisera de datahållande delarna. Det är här som tekniker och koncept typ IE (Information Engineering), CASE (Computer Aided Software Engineering) främst hör hemma.
- Specialistkompetens typ programmerare får en klart underordnad roll och kan istället utvecklas till "Informationsingenjörer".
- IC/PC-verktyg som är tänkta att nyttjas får en klart större genomslagskraft. Detta kräver dock att verksamheten får kunskap om och tillgång till den data som finns.
- Större möjligheter att svara upp mot verksamhetens behov.

Framgångsfaktorer

- Minskad förvaltning och en mer flexibel miljö.
- Klart ökad produktivitet i AU-processen i både tid och kostnad.

Arbetsätt idag

Idag förekommer diverse ansatser som ofta bygger på en arbetsmodell och verktyg enligt följande:

Den stora "resurspuckeln" är konstruktionsfasen och här ingår oftast mycket manuellt arbete.

Signifikant för dagens arbetsätt är:

- Hantverk
- Tradition och konservatism
- Teknikfixering
- Glapp i kunskapsöverföring
- Produktion av dokumentation som sedan är passiv
- "Kaos" vid införande och användare som är "tagna på sängen".

AU/ADB-verksamheten saknar väsentliga informationssystem (datorstöd). Ordspråket "skomakarens barn går i de sämsta skorna" ger full rättvisa av hur det ser ut. Vi befinner oss på pionjärstadiet.

Genom att datorisera AU/ADB-verksamheten kan vi bättre utnyttja våra resurser och använda dem på rätt ställe. Vi skall prioritera "konstruktion av verksamheten" före konstruktion av ADB-systemet. Stora delar av ADB-systemet kan vi generera istället.

Modellerna konvergerar och återanvänder data

Studerar vi de mest förekommande sk "nya" modellerna som finns på marknaden märks en klar tendens till att de börjar likna varandra. Skillnaderna ligger i att man tar större eller mindre hänsyn till helheten (dvs AU-synen) och om man har byggt in projekt- och kvalitetsstyrning etc.

Signifikant är att de flesta bygger på avgränsning, funktions- och datamodellering. Dessutom vill man nyttja det som skapats i ett tidigare steg till ett senare, dvs bygga upp metodkedjor.

Återanvändningsprincipen gör att det är möjligt att datorisera större delar av AU/ADB-processen. Det krävs dock en klar eftertanke, före, om vad man önskar lagra och återanvända eftersom varje lagring kräver förvaltning.

Karaktern av datorstöd för AU/ADB-verksamheten

Vi skall bygga på vår stabila grund av funktioner och data. Vi skall bygga på återanvändning och metodkedjor. Det finns en mängd datorstöd som fyller sin funktion inom AU/ADB-verksamheten.

Signifikant är att hela portföljen av datorstöd strävar efter att successivt beskriva lösningen i detalj för att möjliggöra en automatisering av de tekniska delarna. Beskrivningarna lagras i en eller flera sammankopplade register som ofta kallas för Data Dictionary, Datakataloger eller Encyklopedier.

De delar av register som beskriver den tekniska delen bör i görligaste mån vara omöjlig att förändra utan att först gå via delen som beskriver verksamheten. - Först då vi får en aktiv, tvingande och integrerad miljö av register som stöder traditionell systemutveckling, IC-verksamhet och verksamhetsbeskrivning kan vi uppnå stora effekter.

Notera att det inte finns endast ett register som täcker allt utan flera i en bestämd arkitektur. Att endast tro på ett register för oss tillbaka 15 år då det sk "MIS"-konceptet härskade. Dessutom måste vi ha kopplingar till våra "gamla system" vilka kräver sina datakataloger och överföringssätt.

Effekter på förvaltning av informationssystem

Många problem inom förvaltning kan direkt härledas till en dåligt strukturerad lösning samt en bristande dokumentation. Problemet är dock inte att det saknas dokumentation utan att den inte hänger samman på ett bra sätt eller att den har för dålig aktualitet. Vi kan notera följande:

- Ett utvecklingsprojekt kan, pga sitt omfång och komplexitet tvingas till tidspress vilket gör att dokumentationen blir lidande. Speciellt är detta märkbart just innan installation av systemet.

- I utvecklingsarbetet har vi inte kalkylerat dokumentationsarbetet. Beställaren av ett system är ibland ovillig att betala för något som han inte direkt kan få nytta av.

- De som utvecklar saknar ett direkt incitament för att dokumentera. Det finns ingen anledning för dem eftersom det inte för projektet framåt.

- De som skall förvalta kommer in för sent och kan inte påverka lösningen eller den dokumentation som finns. De ställs ofta inför ett fullbordat faktum.

- Akuta problem tvingar fram ändringar som genomförs utan dokumentation. Det finns inte tid att dokumentera när det "brinner under fotsulorna" men ambitionen finns att göra det när allt har lugnat ned sig. Det lugnar aldrig ned sig.

- Förvaltningsarbete upplevs tråkigt och föga karriärsbefrämjande. De flesta strävar istället mot utvecklingsarbete. Detta innebär en hög omsättning av personal vilket gör att kunskapen om systemet försvinner.

Här ger ett datoriserat arbetsätt en klar effektivisering både inom verk-

Framgångsfaktorer

samheten och inom ADB-verksamheten.

Målet inför framtiden skall vara att:

- Skapa system från dokumentation och bygg på klara metodkedjor som är aktiva och tvingande istället för som idag:

- Skapa dokumentation från systemet.

En ny kompetens behövs, ett nytt skrå är på väg

Att ta till sig ett avancerat datorstöd inom AU/ADB utan att först tillämpa och förstå arbetsmodeller, modellering etc, kommer att sluta med haveri. Det finns många exempel på misslyckanden. Sluta med tilltro kring en produkt om det inte finns en mögnad för den. Mögnad kräver resurser.

Ett huvudproblem som vi har är bristen på kunniga personer inom analys och modellering (funktions- och datamodellering). Vi söker här generalister som både är analytiskt, pedagogiskt och tekniskt kunniga. Det kanske är så att våra karriärvägar idag (chef eller specialist) hämmar denna utveckling?

De datorstöd som finns, och kommer att finnas under en femårsperiod, kommer att kräva expertkunskap som är både analytiskt och tekniskt betingad. Vi kan här tala om "informationsingenjörer".

Det är tveksamt att i nuläget kraftigt decentralisera informationsingenjörerna. Vi behöver erfarenhet, kunskap och en kritisk massa för att lyckas. En av de avgörande faktorerna för att lyckas är spridning och intim koppling mellan metodutvecklare och pågående projekt. Endast då man provar verktygen i praktiken kan vi uppnå resultat.

Att tro att verksamhetsmänniskor eller enbart linjesystemmän kommer ta över hela utvecklingsarbetet

är fel. Verksamhetspersonal kommer istället in i styr- och arbetsgrupper och förmedlar sin kompetens på ett aktivt sätt. Först vid nästa "våg" kan det kanske vara möjligt med en direkt integration i verksamheten.

Den AU/ADB-verksamhet som vill hålla sig fast vid den gamla "COBOL-lästen" kommer att få svårt att hävda sig. Det som beskrivs händer inte under en natt, men räkna med språngvisa effekter.

Det kommer säkert att finnas utrymme för COBOL-programmerare under en period på fem till 10 år. Dock kommer intresset och insatserna att minska allt i den takt som gamla system ersätts.

Datorisering av AU/ADB är nästa "våg"

Dagens arbetssätt, modeller, metoder har funnits länge. Dessa brukas mer eller mindre, helt beroende på styrning, tycke och smak.

Att vidareutveckla området ger en liten effekt. Effekten ligger istället i att sprida ett enhetligt sätt att arbeta inom AU/ADB. Här handlar det om användning och en överenskommen referensram för de personer som ingår i utvecklingsarbetet. **Sluta att utveckla, anpassa, paketera.** - Inför istället på ett målmedvetet sätt. Det stora "lyftet" blir datorisering av det som redan görs manuellt. Manuellt arbete kan då reduceras och göras mer stringent. Vi vill dock kraftigt varna för tron att ett datorstöd utesluter användning av en arbetsmodell.

Att bli förälskad är lätt, speciellt i sofistikerade och "glättiga" datorstödsprodukter. Ingen kan köpa sig fri genom en produkt och tro att effekterna uppkommer utan en massiv och planerad insats för införande. Tänk istället att "allt mellan

ögonblicken av inspiration är hårt arbete".

Vågen efter datoriseringen är troligen ett helt nytt arbetssätt med helt ny metodik. Det brukar dock dröja 10 till 20 år innan en ny våg slår igenom.

Programator profilerar per Reflex

Programatorprofilen växer fram tydligare

Vi har nog varit mest kända för att vara duktiga på att bolagisera dataavdelningar och sätta sprätt på dem.

Under senare år har vi mer och mer som framgår av vår internutbildning betonat affärsperspektivet och nu sammanfattar vi vår affärsidé: "För kunden lönsam informationsbehandling".

av Lars Bergman, SISU

Informationsstrategi är ett område, där vi har en väldigt stark position. Där är vi marknadsledande. Det är fler och fler verkställande direktörer som vill ha en bättre styrning av sin informationsbehandling. Vi har utvecklat ett genomarbetat arbetsätt för sådana uppdrag, som naturligtvis ändå måste genomföras situationsanpassat och "individuellt". Uppdrag inom strategiområdet är väsentligt för oss också på det sättet att de ger oss en god plattform för att infria vårt marknadslofte i affärsidén och ofta genererar följdjobb.

Ett annat område som vuxit fram är "metodupphandlingar", där vi vill sälja oss som samarbetspartner inom metodområdet. Där gör Reflex oss mycket starkare i konkurrensen.

När det gäller metod- och verktygsidan hade vi ju DASAK som lades ner för ca 3 år sedan. Det blev helt



Programators kontaktpersoner till SISU är Per Tidén och Håkan Friberg. Inom moderbolaget finns stockholmsverksamheten med bl a metod och utbildning, som Per Tidén ansvarar för. Håkan Friberg ansvarar för utbildningsverksamheten inom Programatorkoncernen.

enkelt för dyrt att utveckla i paritet med marknadsutbudet t ex när det gällde användargränssnittet.

Nu har vi i stället flera samarbeten med leverantörer av CASE-verktyg. Vi har följt utvecklingen på datorstödsmarknaden för att kunna ge rekommendationer till kunder men också för att åstadkomma anpassningar till våra metoder. Det är t ex fallet med vårt samarbete kring Excelerator och Deft. Vi ser

det som intressantkomplement till våra metoder. Vi går och väntar på ett genombrott på datorstödsidan för det kommer att ge draghjälp åt våra metoder.

REFLEX-samling och profilering

Bakgrunden till att vi satsade på Reflex, var att vi i metodportföljen hade ett antal metoder som utvecklades med en succession av "features" och det blev svårare och svårare

re att hålla överblick och hålla ihop.

Ett viktigt syfte var därför att göra det hela (metodportföljen) begripligt för våra egna konsulter och för våra kunder. Målet var att paketera för att få metodportföljen begriplig och intressant, på ett smakfullt sätt.

Arbetet startade för ungefär 1 år sedan. Där har vi dels arbetat för att göra portföljen till en helhet. Det har också inneburit att vi arbetat med att täcka en del luckor framförallt i gränssnitten mellan metoder.

Ett viktigt led i vår interna utveckling

Under några år har vi i internutbildningen, med LOTS-konceptet, satsat hårt på att affärsorientera våra konsulter. Detta har siktat på att ge dem ett synsätt som skall fungera både för deras egen del som anställd inom företaget men också på att ge dem ett affärsperspektiv i kontakten och arbetet med och hos kunden. Det affärsmässiga perspektivet är viktigt för oss i vårt säljarbete. Vi säljer till företagsledningar och linjechefer mer än vi säljer till datachefer. Jag vill påstå att vi har en mycket god förankring i företagsledningar i stora företag och myndigheter. Det är därför viktigt att vi kan tala med dem om vår kompetens utifrån deras perspektiv och inte kommer med ett databehandlarperspektiv till dem.

När vi arbetat fram Reflex har vi sett det som ett stort vidareutbildnings- och utvecklingsprojekt. Vi har tagit fram en produkt som ger oss möjlighet att visa en enhetlig och begriplig profil för Programators konsultverksamhet. Det gällde då att också få den inarbetad inom Programator. Därför har vi satsat stenhårt på att ge Reflex en bra paketering och få den spridd inom Programator.

Vi har tagit fram ett presentationsmaterial för användning av konsul-

terna ute i deras verksamhet. Vi har kört Reflex-utbildning för alla (100) chefer inför årets "kick-off"-resor. Cheferna har fått en 40 timmar utbildning för att i sin tur kunna genomföra en 20-timmars utbildning för sin egen personal. I paketet ingår också material för en 2-timmars presentation av Reflex för systemutveckling "från ax till limpa".

Utbildning

Håkan Friberg ansvarar för utbildningsverksamheten inom Programator-koncernen som omsätter 75 MSEK. I den verksamhetsdelen ingår Lärdata och CBI som tillsammans omsätter 20 MSEK. Dessutom ansvarar Håkan för ledarför-sörjning inom Programator. AB Programator som i sig är ett rörelsedrivande konsultföretag, har 350 anställda men fungerar samtidigt som moderbolag för ett 50-tal bolag. VD är Anders Skarin. Organisationen av dotterbolagen är divisionaliserad.

Inom moderbolaget finns stockholmsverksamheten med bl a metod och utbildning, som Per Tidén ansvarar för.

FoU-verksamheten styrs av ett FoU-råd, vars inriktning är att prioritera för koncernen gemensamt intressanta satsningar. Den arbetar med en budget på 8 MSEK varav ungefär tre fjärdedelar på metoder. FoU-arbetet byggs på en grov plan som upprättas årligen. Den utarbetas i förslagsform av Per och beslutas i FoU-rådet. Arbetet bedrivs sedan i projektform med bemaning från konsultsidan. FoU-folket sitter alltså inte i stab. Normalt bedrivs dessutom arbetet så att det går att förena med kunduppdrag. Vi driver sällan renodlade inter-noprojekt för metodutveckling.

SISU

En grundinställning vi har är att vi vill ha högsta möjliga utväxling på

våra FoU-pengar. Därför är det självklart intressant om vi kan hitta partners för att snabbare komma fram inom intressanta områden, men vi måste också se till att arbeta med en praktisk och resultatorienterad syn. För att samarbetet med SISU skall bli verkligt intressant måste utvecklingen inom SISU-ramen ligga inom för oss prioriterade områden. Vi ser positivt till att vi gemensamt skall söka projekt som ligger rätt i vår prioritering och där vi kan engagera oss i insatser som då är klart resultatnriktade.

Metodutveckling

Hela Programator medverkar i metodutvecklingen. Vi arbetar i projektform med lämpliga personer och blandar teoretiker och praktiker. På så vis åstadkoms en naturlig förankring, vilken är mycket viktig för oss.

Vi arbetar från en prioritering utav vad som "bör åtgärdas under året" med projektledare antingen från konsultsidan eller metodsidan. Projektledaren utarbetar en projektbeskrivning. Vi från metodsidan är med i referens- eller arbetsgrupp. Hela Reflex-paketeringen är ett exempel på det arbets sättet. Vi försöker dessutom alltid att utveckla tillsammans med kunder. Exempel på det är tex utvecklingen av modellen för verksamhetsutveckling, en parallell till systemutveckling. Där arbetade vi tillsammans med EDS, (Ericsson Data Services) och SINOVA ett managementkonsultföretag. Mats Lejdeby hos oss var projektledare. Modellen ingår nu som en del i Reflex-paketet.

En annan typ av utvecklingsinsatser i samverkan bedriver vi med verktygsleverantörer. Just nu har vi genomfört eller är på gång med anpassningar till Reflex med MIMER, DEFT-Ingres, Oracle, IDEAL och Excelerator. Vi bygger samtidigt upp egen verktygskompetens

även t ex för IEF och IEW. Vår målsättning är att vi skall kunna konsultera inom alla de här områdena.

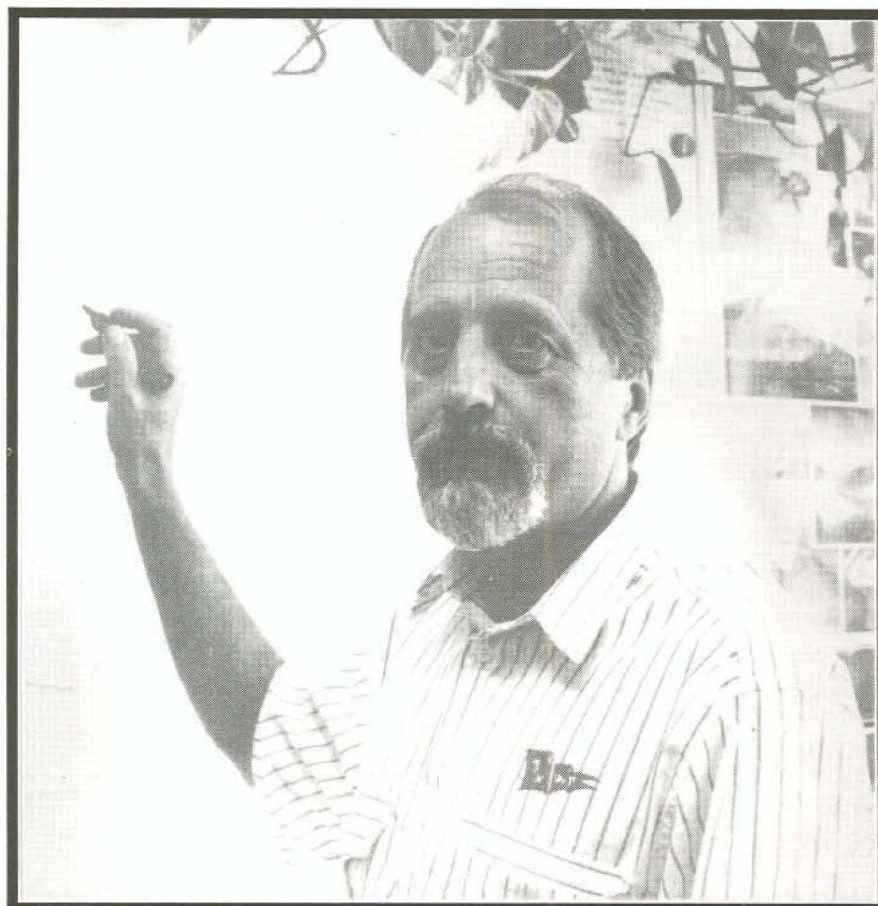
Reflex och IE (Information Engineering) har mycket gemensamt. Vi har jobbat med kopplingarna mellan MBI och SAK, så att de handböcker vi nu har verkligen hänger ihop. I stort kan man säga att MBI nu utgör stommen i Reflex - Analys och SAK i Reflex Systemutveckling.

Vi satsar väldigt mycket på hur man skall använda metoden med CASE och med 4GV. I "Reflex med moderna verktyg" har vi byggt in PRO4GV. Det är framförallt Håkan Fägersten som varit aktiv i Riksdatabundets MOSE-projekt och Christina Lagerkvist, som ansvarat för den delen.

Vi har inte tidigare profilerat verksamhetsutveckling som del i kompetensportföljen men det kommer vi att göra nu. Vi kommer överhuvudtaget att vara vassare inom metod- och verktygskompetens och därmed i konkurrensen på det området.

Idag inser de flesta att metoder måste man satsa på. - Men vilken skall man satsa på? - Det finns metodvärderingsgrupper ute i de flesta företag idag. Intresset är alltså stort och tillflödet av nya verktyg förstärker det intresset. - Vi räknar med att det kommer en boom inom metodområdet inom något eller några år bl.a. tack vare den flora av verktyg som växer fram. Verktygen i sig förutsätter ju ett metodiskt arbetssätt och som alltid gäller: "först metoden - sedan verktyget".

Kalle Wigander, Programator:



Jag tror att vi kraftigt kommer att influeras av amerikanskt systemtänkande. Där ligger de kanske 5-10 år före oss anser jag. Det systemsynsätt som presenteras där tycker jag man kan ana hos en del företagsledningar ute i Sverige.

Så här beskriver Kalle Wigander sig själv och sin bakgrund. - Från början var jag sjöbefäl, men efter diverse seglatser världen runt kom jag av en slump in i ADB-branschen.

av Lars Bergman, SISU

Där började jag som programmerare och har nu varit i branschen i 28 år, och fascinerades av yrket och läste därför, parallellt med arbetet in en fil kand med ADB som grund på 60-talet. Sedan fortsatte jag att

arbeta och läsa på 70-talet och tog en tekn lic på KTH.

Samtidigt tog sjömansblodet ut sin rätt och jag byggde en åtta meter lång motorbåt med vars hjälp större delen av vår skärgård kunde beses och vännerna besökas i 35 knops fart, fastän det oftast gick långsammare. Jag tycker om mekanik och pular följaktligen ofta med mina bilar, snickrar och bygger en hel del hemma och byggde bl a om huset nyligen.

Dags att förnya systemtänkandet

Kalle har arbetat på Programator i 20 år och dåundrar man kanske hur man kan vara på ett och samma ställe så länge. Det går ypperligt, utbrister Kalle, eftersom arbetet som konsult är så omväxlande och ger utrymme för de mest olika arbetsinsatser. Just nu arbetar jag som projektledare i PI=90 på Postgirot, ett projekt som har som slutmål att " ... bidra till

- Ny registerstruktur med renodlat innehåll

- Förändrad systemanvändning

- Ökad säkerhet och kvalitet

För att få till stånd ett fortsatt affärs-mässigt, lönsamt och effektivt arbete inom GK på 90-talet.". D v s ett stort projekt av central betydelse för uppdragsgivaren, vilket i sig också kan ses som en utmaning både arbets- och kunskapsmässigt. Jag arbetar också åt Pripps där jag hjälper till med att bygga upp och hålla vid liv ADB-avdelningens handböcker.

Lite skämtsamt brukar jag säga att jag arbetar åt fyra av Sveriges största:

- Pripps, vårt största bryggeri och för övrigt ett av våra största åkerier, för den som inte känner till det.

- Postgirot, vår största penninghanterare, som omsätter 5.000 miljarder per år.

- Programator som de flesta i branschen känner som ett ledande konsultföretag.

- Stockholms Universitet och KTH som är vår största utbildare i "Data och systemvetenskap", där jag har

ett halvt lektorat och där jag hoppas att snart få min doktorsavhandling klar.

Att arbeta inom ett dynamiskt företag som har resurser och ställer dem till förfogande är den andra sidan av att arbeta inom Programator, som i kombination med att lära och utveckla sig själv gör det möjligt att nå den vinstgivande kombinationen " Vetenskap och beprövad erfarenhet".

Jag hör definitivt inte till den grupp som brukar kallas "branschens belackare" och som för fram sina synsätt genom att påpeka hur dåliga vi är när det gäller att utveckla system.

Vi arbetar helt enkelt olika med olika typer av system, som kräver olika ansatser och inte minst kommer vi i kontakt med olika personer som var och har sin erfarenhet och kunskap som bas för arbetet.

Med andra ord hör jag till dem som tycker att det allra mesta vi gör görs bra och helt klart bidrar till allas vårt väl och ve.

Därmed inte sagt att vi inte kan bli ännu bättre och det är nu SISU kommer in i bilden för min del.

För mig som läser en hel del och dessutom har förmånen att arbeta åt trevliga klienter med bra uppdrag är SISU en naturlig mötesplats där vetenskapen och den beprövade erfarenheten kan mötas.

SISU blir på så sätt ett forum som

skapar synergieffekt, dvs bidrar till att 1+1 blir mer än 2, vilket i sin tur leder till att vi alla blir ännu bättre antingen vi står på forskningssidan eller på den praktiska sidan eller står med en fot i vardera lägret och ibland känner oss lätt splittrade.

Ge exempel på några intressanta företeelser som är på väg upp över horisonten. - Kan du med din radar se dem nu?

Distribuerade databaser

Jag tror att vi är i början av en fundamental förändring när det gäller informationsanvändningen. Den har påbörjats i och med att man nu kopplar upp PC mot centrala databaser eller oftare tar kopior av utdrag av dessa. - Förut har man som beslutsfattare eller användare kört sina statistikuttag i batch. Man har haft dålig kontroll över underlagets kvalitet och struktur. Nu kan man sitta och bearbeta materialet på olika ledder, ställa sig frågor, ställa nya frågor mot materialet och få snabba svar. Det tror jag är en utveckling som nu bara befinner sig i ett begynnelsestadium. Vi kommer att få se en stor ökning av "lokalt processande" och därmed en helt annan nivå på informationsanvändning för beslutssituationer och analyser.

En annan företeelse som Kalle noterar är användningen av skisseringshjälpmedel. Här finns det nu stora risker för en "back-lash" genom felslagna förhoppningar hos användare och fack. Jag tror att det

är mycket väsentligt att ha ett bakomliggande fundament av utvecklingskunnande när man går in i systemskissering. Man skall ha en bra grund i väldefinierade avgränsningar och veta vilka rutiner man arbetar med och hur de knyter an till omgivningen. Annars är risken stor att man får lösryckta lösningar som på sikt skapar svåra problem såväl för användarsidan som för ADB-sidan. Man måste vara klar över var i utvecklingskedjan man skall använda skissering. Hur man skall göra det och när man skall göra det samt VAD det är som skall resultera. Vi har i vårt Reflex-koncept bl a satsat på att bygga in skisseringen som en del i en helhet, som gör att skisseringen hamnar på rätt plats. Metodiken som sådan presenteras i PRO4GV och kallas "utveckling med moderna verktyg" där Reflex ger ramar och sammanhang.

Du har arbetat länge och mycket med metodfrågor och representat "programatormetodiken". Vad är det egentligen som hänt under åren?

1974

"Gula faran" (Systematiserad systemering) jobbade Lars Irstad och jag fram som ett sätt att angripa hur man skulle arbeta enligt SIS RAS, som kom ut något år innan.

1979

SAK-boken. Den hade sin grund i uppdrag åt Länsförsäkringsbolagen men konkretiserades, utvecklades och prövades samt reviderades i anslutning till detta i ett projekt vid Televerket. Lennart Schoug hos oss och Christer Dahlgren, som då var metodchef på Televerket, var aktiva författare tillsammans med mig. Man skulle kunna säga att vi strävade att beskriva systemutvecklingen som en procedur.

1985

ROS-konceptet publicerades men började föras ut på allvar under 1986. Under början av 80-talet började intresset för modellering växa fram. Tidigare hade ett processinriktat beskrivningssätt dominerat systemutveckling. ROS är ett koncept som kopplar ihop objekt- och funktionsmodeller i utvecklingskedjan. Dessutom kan man säga att vi i konceptet ville lyfta fram resultat, milstolpar, i utvecklingsprocessen för att göra den mera styrbar.

1987

PRO4GV-konceptet släpptes ut. Det var framförallt Christina Lagerkvist hos oss som drog det jobbet. Där ges en metodik för systemutveckling i vilken man utnyttjar möjligheterna hos moderna hjälpmedel.

1988

REFLEX lanseras. Det representerar egentligen integration av våra olika tidigare synsätt och metoder. Det ger dels en helhetssyn som är knuten till verksamheten och som placerar in ett strategiskt tänkande, men dels också placerar in SAK, ROS, MBI och PRO4GV samt förvaltning i en helhet. Där har vi också siktat till att skapa fungerande kopplingar mellan delarna. Vi vill i detta bädda för inte bara produktivitetsförbättringar utan kanske mer för effektivitet som att skapa rätt system och acceptans för systemen. Mönstret i ROS kommer att stå stabilt men poängteringen kommer att förändras, t ex kommer databasskapande och programskapande att finnas men inte synas på samma sätt i systemutveckling framöver som idag. Verktyg kommer att bli mer integrerade i modell och metod framöver. Vi har t ex nu i samarbete fått fram en version av Excelerator som är anpassad till REFLEX. Det mönstret tror jag kommer att bli vanligare framöver.

Vilka idéer eller synsätt ser du under uppsegling

Tror du CASE-verktygen kommer att leda till att vi får en inströmning av "amerikanska metoder". - Nej jag tror att vi nu har så stort metodkunnande och metodintresse att vi väljer metoder utifrån en bättre och oftare djupare kunskap än man gör i ADB-branschen generellt i USA. Vi kommer nog att låta metoderna styra verktygen snarare än tvärtom och se till att verktyg och metoder hamnar rätt i en helhetssyn. Däremot kommer säkert användningen och utvecklingen inom CASE-verktyg att bli starkt ökande.

Däremot tror jag att vi kraftigt kommer att influeras av amerikanskt systemtänkande. Där ligger de kanske 5-10 år före oss anser jag. I en av de kurser jag håller på universitetet "Informatik med generella system och cybernetik" har vi med några mycket intressanta författare, Schoderbek och Miller. "Living Systems" heter den senares bok. Det systemsynsätt som presenteras där tycker jag man kan ana hos en del företagsledningar ute i Sverige. Oftast med annan bakgrund än ADB t ex civilekonomer och civilingenjörer. Miller har en modell med system i 7-nivåer och där han visar på en generalitet i att varje systemnivå innefattar och behöver 19 kritiska funktioner för sin existens. Jag tror att det här synsättet kommer att påverka oss och få stark spridning redan inom en femårsperiod och då inte främst på ADB-sidan utan ute i övriga delar av organisationen och i dess ledning. Detta generella och funktionella systemtänkande kommer naturligtvis att påverka synen på ADB-system och deras plats i verksamheten. Ökade krav kommer att ställas men också från en mer insiktsfull kravställare. - Man kan notera redan idag att ordet system har slutat att betyda ADB-system i allmänt språkbruk. Det är idag använt inom många olika områden som har föga med ADB att göra..

"Det tar lika lång tid att ändra infosystem som att utkämpa andra världskriget"

Mats-Åke Hugoson, Programator

Mats-Åke Hugoson, Programator, är känd för många av oss som upphovsman till MBI-metoden.

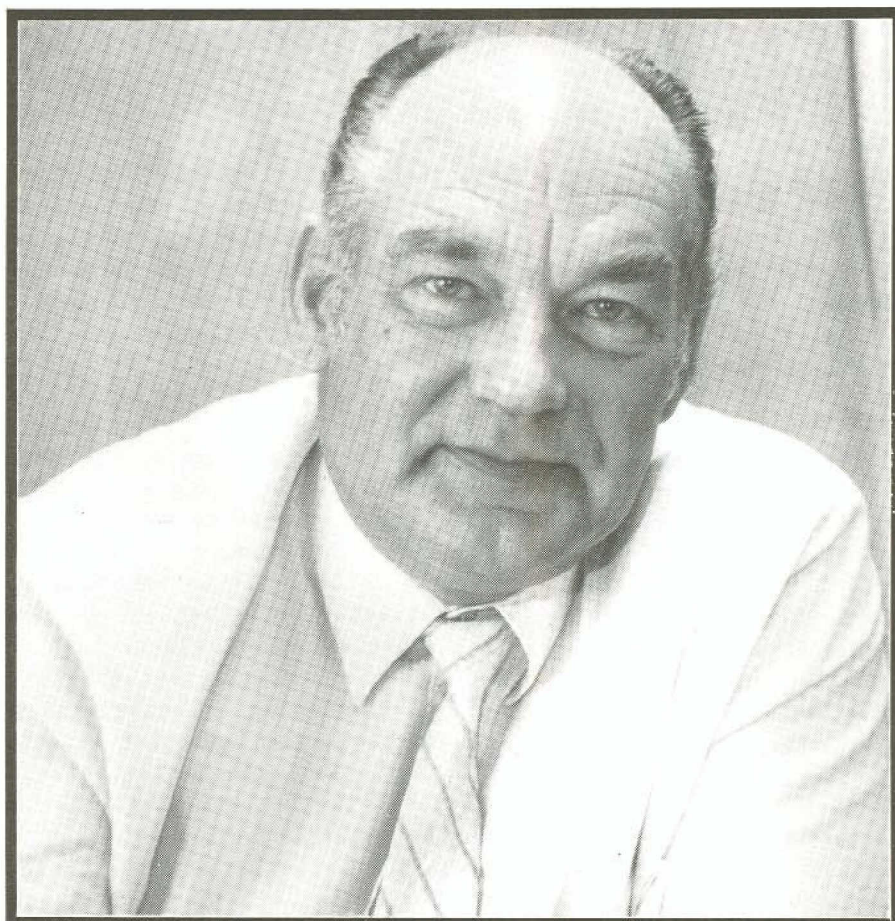
av Lars Bergman, SISU

MBI kan karaktäriseras som ett funktions/processorienterat sätt att analysera en verksamhet. Den innehåller styrsystemtänkande och överhuvudtaget en betoning på systemtänkande. De önskade effekterna i verksamheten är utgångspunkter vid specifikation av informationssystem. Mycket förenklat ser metoden ut som följer:

Skede 1 innebär en nedbrytning top-down, med ett ledningsperspektiv och med syfte att få fram en struktur med definierade uppgifter och ansvarsgränser.

Skede 2 behandlar de prioriterade delarna av verksamheten med snarast bottom-up angreppssätt. Bl a skall infångning av kopplingar mot ledningsmål göras. Viktigt är att det gäller att skapa en motivationorienterad målsyn. Synsättet innebär att man i detta skede skall utforma lösningar för verksamhetsuppgifterna och därifrån härleda informationsbehoven.

Skede 3 innebär att man utformar verksamhetsstöd i form av ett antal informationssystem som samverkar med varandra. Teknologin är se-



Mats-Åke Hugoson pläderar för ett struktureringstänkande som bygger på autonoma verksamhetsbaserade informationssystem. Där fokuseras sambandsinformationen mellan verksamheter.

kundär. Sambandsstrukturen är en spegling av styrstrukturen. Ansatsen medger en successiv systemutveckling med hög grad av flexibilitet.

Chalmers

På Chalmers har Mats-Åke varit universitetslektor från 69/70 till

ungefär 1985. Nu har han en personlig tjänst som göromålsprofessor på deltid. Den innebär att Mats-Åke skall få lov att forska kring saker som intresserar honom med inriktning på verksamhetsorienterade administrativa system. Dessutom handleda de doktorander som är intresserade av det området. Och slippa administration.

Strategiuppdrag

I sin konsultverksamhet arbetar Mats-Åke huvudsakligen och mycket med strategiuppdrag. Det är utifrån det perspektivet vi skall se de synpunkter Mats-Åke presenterar när det gäller syn på strategi och systemutveckling i de följande avsnitten och Mats-Åkes engagemang i SISU:s AU90-projekt. SAS, ICA, Billerud och landsting är uppdragsområden som han varit inne på nyligen. Sjukvårdsområdet är mycket hett för närvarande.

Strukturfrågorna börjar nu ses som starkt strategiska och företagsledningen ställer krav på systemlösningar till datasidan, berättar Mats-Åke. De stora systemen från 70- och 80-talen skapar stora problem vid verksamhetsförändringar. Företagsledningen diskuterar om strategi för AU och styrning. Det är strukturfrågor, inte teknik eller enskilda infosystem. Strukturfrågorna är alltså ledningsfrågor och inte enbart tekniska sådana.

Man ser informationsorienterade frågor men inte "informationsteknologi". Man är intresserad av resursen information och verksamhetens utnyttjande av och försörjning med information. Langefors tes om att "alla strukturfrågor måste ses uppifrån" håller bra även idag.

Struktur är ett mönster av komponenter och hur de skall samverka. Man gör mycket tidiga bedömningar av vilka informationssystem man vill ha, hur de skall samverka och vilka krav som skall ställas samt fördelning mellan ansvar/beställarfunktion kontra utförandefunktion. Det råder ju en allmän decentraliseringstrend i företagen, som jag för anser hälsosam och positiv. Man måste dock vara varken för konsekvenser av sitt handlande i fråga om organisation och informationssystem. Decentraliserade informationssystem (liksom decentralisering i företaget) kan

leda till risk för anarki. - Kan man inte klara samverkan så löper man stor risk för stora problem i informationsbehandlingen.

"Det tar lika lång tid att ändra infosystem som att utkämpa andra världskriget".

Dataverksamheten är inget självändamål även om man fortfarande ofta kan se att den uppfattningen i praktiken styr ADB-sidans tänkande. Dataverksamheten måste inordnas i ett verksamhetsperspektiv. Vi har inga infosystem - som existerar självständigt i företaget utan vi har behov av stöd ute på olika ställen i verksamheten. De olika finformationsdelsystemen skall nära integreras med verksamheten men de måste också samverka sinsemellan.

Vi måste nå fram till att systemansvar i meningen systemägande och förfogande är en realitet. Verksamhetsorientering är A och O för dataverksamheten. Kombinerar vi en god systemstruktur med systemägande och systemförfogande så uppnår vi enkelhet och därmed följande anpassbarhet och flexibilitet och snabbhet, när det gäller ADB-systemen. Ta organisationsändringar som jämförelse. Där kan man snabbt, kanske på ett kvartal till ett halvår genomföra drastiska förändringar även av stora organisationer. Samma sak om man ser produktionen i en verkstad. Där ligger ADB-verksamheten med dess nuvarande sätt att fungera, långt i lä. - Att radikalt ändra i stora centrala ADB-system kan ta 4-5 års tid, d.v.s. lika lång tid som att utkämpa andra världskriget.

ADB-folkets har några trossatser inarbetade i sin livsåskådning Information skall ha en enhetlig lagring med gemensam lagring av gemensamma data och med gemensam åtkomst. Detta skall åstadkommas via integration av register

och med undvikande av all dubbel-lagring. Låga driftskostnader skall samverka med gemensam lagring.

Många på ADB-sidan klagat över den stora underhållsördan. Men enligt Mats-Åke skördar man just där frukterna av sina grundläggande trossatser. Minskat underhåll måste angripas från de tre nivåer som underhållet kan kategoriseras i, nämligen: rättelser, tekniskt underhåll p g a förändringar i den tekniska miljön resp anpassning till förändringar i den stödda verksamheten. Lösningen ligger säger Mats-Åke med emfas i att man ändrar den grundläggande strukturen i ADB-systemen genom att anlägga ett annat struktureringsstänkande som utgångspunkt för att utveckla system. Detta andra struktureringsstänkande, **autonoma verksamhetsbaserade informationssystem**, som Mats-Åke pläderat för under ett par år, poängterar sammanknytning mellan en verksamhet och dess system och samverkan mellan system via sambandsinformation.

Att vi har den situation vi har idag anser jag beror mycket på att systemdesigners har ett totalangreppssätt i stället för att försöka strukturera, fortsätter Mats-Åke. Han anser att datafolkets grundprinciper för lagring och sökning måste ifrågasättas.

Den grundläggande princip eller det synsätt som Mats-Åke för fram fokuserar **sambandsinformationen mellan verksamheter** (och system om dessa utvecklats enligt hans princip), därför att den är:

- är enkel
- har låg volym
- har låga tidskrav (större än tio sekunder)
- är stabila över tiden

Svårt men viktigt att hålla isär är här begreppen och principerna:

decentralisering contra distribution samt verksamhetsfunktion i betydelsen "lokalt delsystem" i motsats till funktion som vertikal delning av en verksamhet. Mats-Åke vill se en verksamhetsdecentralisering som åtföljs av ansvar och befogenheter för lokala delsystem. Samverkan mellan system sker där med sambandsinformation i stället för registersamverkan.

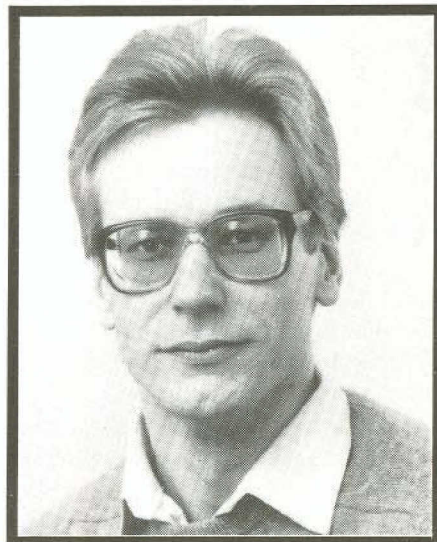
För att åstadkomma detta skisseras följande arbetssyn:

Först gäller det att ordna upp strukturen. Sen kan man mycket väl jobba med t ex objektorientering. Det gäller att ha gjort klar ansvar för information (meddelanden, transaktioner). Det är meningslöst att tala om termansvar i det sammanhanget.

Begrepp står för term enligt vår nuvarande terminologi. Vi arbetar nu med GBK, gemensam begreppskatalog som ett sätt att få god beskrivningsprecision. Alla meddelanden skall byggas på GBK. På så vis säkras sambandsinformationens meddelanden och transaktioner. ■

Erik Knudsen är ansvarig för området KBS, Kunskapsbaserade system inom SISU.

Reserapport från besök i USA och Canada



Under ett par hektiska veckor i början på juni besökte jag två universitet samt ett par av IBM:s forskninglaboratorier utanför New York för att avsluta det hela med att delta i en konferens i Edmonton, Canada.

av Erik Knudsen,
SISU, KBS - Kunskapsbaserade system

Huvudsyftet med resan var att presentera forskningsrapport; "Expressing Unrestricted Grammars by Extended DCG"*, på sagda konferens som hade blivit accepterat som bidrag tidigare under året. Då jag ändå var "over there" var det lämpligt att ta tillfället i akt och besöka olika platser som kunde vara av intresse för den verksamhet som bedrivs inom SISU och SYS-LAB.

CSCSI-88

Den fullständiga titeln på konferensen var "The 7th Biennial Conference of the Canadian Society for Computational Studies of Intelligence" och förkortas CSCSI. Konferensen är ett evenemang som går av stapeln varannat år och äger rum på olika platser i Canada vid varje tillfälle och nu var det således Edmonton som hade fått den äran. I år löpte konferensen parallellt med två andra konferenser som behandlade bland annat vision interface respektive graphics interface. Antalet deltagare var ca 300 vilka huvudsakligen var från Canada även om de utlänningar som deltog var spridda från ett tämligen stort antal länder. Enligt initierade personer på plats har man strävat efter att internationalisera konferensdeltagandet än mer utan att lyckats alltför väl. De rent praktiska arrangemangen fungerade klanderfritt och helhetsintrycket var gott. Rent innehållsmässigt kanske konferensen var något ojämn varierande från en del mycket bra presentationer till inte fullt så lysande dito. De olika ämnesområ-

*DCG = Definite Clause Grammars.

den som förekom var indelade i följande grupper:

- Natural Language
- Reasoning
- Knowledge representation
- Perception
- Knowledge Base Systems
- Applications

Av dessa ämnesområden var reasoning representerat med flest antal rapporter medan natural language var minst. Sex inbjudna gästtalare förekom. En mycket stor del av de rapporter som presenterades behandlade således reasoning (sv resonemang) och då speciellt icke monotont resonemang helt i enlighet med de strömningar som pågår idag inom forskningsvärlden. Det var därför i allra högsta grad en munter upplevelse att få höra en sådan auktoritet som Charles Morgan mer eller mindre sabla ner de forskare som ansågs sig syssla med olika obskyra typer av logik i samband med icke monotont resonemang. Framför allt sköt han in sig på de som ansåg sig syssla med icke monotont logik vilket ansågs vara nys emedan denna typ av logik egentligen inte var något utöver de "klassiska logikerna". Hans huvudinvändning var således: "You can do non monotonic reasoning but you can not do non monotonic logic beacuse there is no such thing as non monotonic logic". Denna presentation var helt obetalbar och något utöver det vanliga.

I samband med perception presenterades ett par rapporter som med hjälp av neural networks angrep olika problem inom taligenkänning och bildanalys.

Under konferensen knöts en hel del kontakter med olika universitet och ett institut av SISU:s karaktär även om storleken skiljer sig åt avsevärt. Jag tänker då på Alberta Research Council med säte i Alberta som bedriver en omfattande verk-

samhet inom olika områden inom data och systemvetenskap. Ett bestående intryck från Canada är att de verkar vara både lyhörda för och uppmärksamma på ideer som kommer från andra delar av världen och det arbete som bedrivs där.

Besök

Washington

Vid George Mason University strax utanför Washington besökte jag institutionen Information Systems and Systems Engineering som leds av Larry Kerschberg. Dessa sysslar bland annat med kunskapsrepresentation och expert system i kombination med data baser. Ett förhållande som torde vara känt i och med institutionen var med om att anordna konferensen om Expert Data Bases som hölls under våren i år.

IBM

Några mil norr om New York finns ett par av IBM:s forskningscentra. I Yorktown Heights besökte jag den grupp som arbetat med TQA som är ett NLP (Natural Language Processing) system. En användare kan göra utsökningar ur en relationsdatabas i naturligt språk. Det finns ett antal intressanta egenskaper hos systemet bland annat parafrasering. Internt är systemet baserat på den teori om transformationsgrammatik som Chomsky utvecklade. I Watson Research Center träffade jag bland andra Michael McCord som är en av de mer framstående inom området Machine Translation (maskinell översättning). Vid detta forskningslaboratorium pågår ett mycket stort antal olika aktiviteter vilket resulterade i ett givande och stimulerande utbyte av ideer och erfarenheter.

Toronto

På Department of Computer Science vid University of Toronto träffade jag John Mylopoulos som leder arbetet där. Tyvärr hade de flesta av forskarna redan åkt till Edmonton för att besöka CSCSI varför jag just inte träffade mer än ett par personer. Man bedriver forskning inom en rad olika områden. Mitt intryck var dock att tonvikten ligger på området konceptuell modellering i olika former och i kombination med kunskapsteknik.' ■

Vi vet att många av våra intressenter antingen redan satsat eller står i begrepp att göra satsningar på objektorienterade ansatser. Om Du är en av de som skall arbeta med detta och eftersom detta ändå är något Du måste lära Dig - varför inte göra det tillsammans?

SISU inbjuder till bildandet av KOS

Kompetensnät

Objektorienterad Systemutveckling

Bakgrund

Objektorienterade metoder, språk och verktyg har på kort tid blivit mycket populära för att specificera, konstruera och implementera informationssystem. Än så länge är det dock betydligt fler som pratar om objektorienterade ansatser än som verkligen använder dem.

Sina största framgångar har de objektorienterade ansatserna rönt inom programmeringsområdet. Vi behöver bara nämna programspråk såsom Simula, Smalltalk, C++ och Objective-C. Här har också det mesta av begreppsapparaten utvecklats. Till programspråken har också kopplats konstruktionsmetoder. Också inom andra närliggande områden såsom databashantering och konceptuell modellering (datamodellering) återfinns många av dessa idéer.

Vi tror att det objektorienterade synsättet kan användas genomgående från specifikation till implementering av informationssystem. Frågor som inställer sig är bl a: Hur förhåller sig konceptuell modellering till objektorienterad programmering och objektorienterade databaser? Kan dagens specifikationstekniker direkt kopplas till objektorienterade ansatser för konstruktion och implementering? Krävs förändringar / anpassningar av dessa? Bör i stället helt nya ansatser utvecklas?

SISU inbjuder därför till bildandet av ett kompetensnät i form av ett samarbetsprojekt. Projektet kommer att behandla en eller flera av dessa punkter:

- 1 - Tillämpningar av det objektorienterade synsättet inom olika områden samtlikheter och skillnader mellan dessa.
- 2 - Hur objektorienterade verktyg eller tekniker kan fogas in i existerande metoder och miljöer.
- 3 - Skissera en sammanhängande metodkedja byggd på det objektorienterade synsättet.

Mål

Målet för projektet är att:

- Höja den allmänna kompetensnivån beträffande objektorienterade ansatser hos projektdeltagarna.
- Utreda om, och i så fall hur, objektorienterade verktyg kan integreras inom ramen för existerande metoder / modeller för systemutveckling.

forts nästa sida

Uppläggning

Projektet kommer att läggas upp i form av regelbundet återkommande möten (1-2 per månad). Förslagsvis görs de tre punkterna ovan till etapper i projektet. Deltagarna kommer att ha stora möjligheter att själva påverka vad som skall studeras inom ramen för projektet.

Etapp 1

Under varje möte specialstuderas ett område. Förslagsvis presenterar någon eller några projektdeltagare inläst material vilket sedan diskuteras. Samtliga projektdeltagare måste naturligtvis vara pålästa för att detta skall fungera. Vidare kommer ett antal praktikfall ("cases") att presenteras vilka skall lösas av projektdeltagarna.

Exempel på områden att studera är objektorienterad programmering (C++, Ada, CLU), objektorienterade databashanteringssystem (GemStone, SIM, Postgress), objekthanteringssystem (AVANCE, VBASE) samt olika designmetoder och konceptuella modeller. Syftet är dels att bygga upp en gemensam begreppsapparat dels att visa på konkreta likheter och skillnader i synen på objekt mellan olika ansatser. Praktikfallen är främst tänkta att konkretisera och därigenom öka förståelsen för det objektorienterade synsättet.

Resultatet av denna etapp är främst en kompetenshöjning hos deltagarna. Om intresse finnes kan även en kortare dokumentation produceras innehållande bl a en översikt av området, begreppsdefinitioner och förslag till lösningar på praktikfallen.

Etapp 2

Baserat på vunna kunskaper från etapp ett kommer försök att göras att foga in ett eller flera av de verktyg och tekniker som studerats ovan i en existerande metod. Företrädesvis väljs en väl beprövad och/eller dokumenterad metod, t ex AU-metoden (Volvo) eller PAM (Ericsson/SISU). Metoder som dessa består ofta av två grenar, en gren vilken modellerar funktioner vilka implementeras som program i ett konventionellt programspråk och en annan gren som modellerar objekt (data) vilka implementeras som relationer i en relationsdatabas.

Exempel på områden att studera: Hur påverkas metoden om det konventionella programspråket ersätts med ett objektorienterat sådant, eller om relationsdatabasen ersätts med en objektorienterad sådan, eller om både programspråk och databashanterare byts ut mot objektorienterade motsvarigheter? Ytterligare studieobjekt är hur funktionsmodellen och den konceptuella modellen förhåller sig till objektorienterade programspråk respektive databaser.

Resultatet av denna etapp bör vara en rapport som beskriver metoden, hur man gått tillväga för att infoga respektive objektorienterat verktyg i denna samt beskrivningar och analyser av problem som dykt upp under detta arbete.

Etapp 3

Baserat på vunna erfarenheter under etapp två kan en metodkedja börja skisseras som i högre grad än sina konventionella motsvarigheter drar nytta av det objektorienterade synsättet. Den framtagna metoden bör också testas på ett eller flera praktikfall.

Den skisserade metoden dokumenteras i en rapport.

Tidsplan

Projektet beräknas starta med förträff under november 1988 (etapp 1)

Intresseanmälan görs till Stefan Britts, SISU, senast 7 november.

SISU-matrikel

ARTHUR YOUNG AB

Åke Ekström
Box 3143, 103 62 Stockholm
Tel: 08/796 33 00

ABB DATA AB

Gunnar Nilsson
ABB DATA AB, 721 80 Västerås
Tel: 021/32 33 00

AU-GRUPPEN

Sven-Bertil Wallin
AU-Gruppen AB,
Kungsg. 53, 111 22 Stockholm,
Tel: 08/24 34 20

DATA LOGIC

Örjan Odelhög
Data Logic AB,
Fröfästeg, 125, 421 31 Västra Frölunda,
Tel: 031/45 03 40

DIGITAL

Staffan Westbeck
Digital Equipment AB,
Allen 6, 172 89 Sundbyberg
Tel: 08/733 80 00

ENEA

Bo Steinholtz
ENEA DATA Svenska AB,
Box 232, 183 23 Täby
Tel: 08/756 72 20

ERICSSON

Christer Dahlgren
HF/DT ERICSSON, 126 25 Stockholm
Tel: 08/719 07 53

FÖRSVARETS

RATIONALISERINGSINSTITUT
Mikael Franzén
FRI, Box 80008, 104 50 Stockholm
Tel: 08/788 75 00

FÖRSVARSTABEN

Torleif Olhede
Försvarstaben,
Box 80001, 104 50 Stockholm,
Tel: 08/788 78 67

IBM

Lars Arosenius
IBM Svenska AB, 163 92 Stockholm
Tel: 08/793 40 60

INFOLOGICS

Dick Eriksson
SU TVT Infologics AB,
Chalmers Teknikpark
Sven Hultins g. 9, 9A, 412 88 Göteborg
Tel: 031/72 42 60

IRM CONSULT

Eskil Swende,
IRM Consult AB
Box 100, 161 26 Bromma,
Tel: 08/26 93 10

KOMMUNDATA

Gunnar Sandberg
Kommundata AB, 125 86 Älvsjö
Tel: 08/749 80 00

MANDATOR AB

Hans Willars
117 81 Stockholm
Tel: 08/709 31 73

MIMER SOFTWARE AB

Lars-Erik Jansson
Box 1713, 751 47 Uppsala
Tel: 018/18 50 00

PEAB

Stellan Borg
Philips Kistaindustrier AB
Dream, Box 33, 164 93 Kista
Tel: 08/703 10 00

POSTEN

Gert Persson
Posten,
Koncernstab KP, 105 00 Stockholm
Tel: 08/781 10 00

PROGRAMATOR

Håkan Friberg o Per Tidén
AB Programator,
Box 20072, 161 20 Bromma,
Tel: 08/799 35 00

RIKSSKATTEVERKET

Lars Olsson,
Riksskatteverket, 171 94 Solna
Tel: 08/764 80 00

SAAB-SCANIA

Sven Yngvell
Saab, Flygdivisionen Dataservice
581 88 Linköping
Tel: 013/18 23 86

SAS DATA

Ove Lundvall
SAS Data, 161 87 Stockholm
Tel: 08/797 10 18

S-E-BANKEN

Peter Söderström
S-E-banken, SMD M4, 106 40 Stockholm
Tel: 08/763 50 00

SKANDIA

Anders Fungdal
Skandia-Data, 103 50 Stockholm
Tel: 08/788 17 26

SKF

Bo Lindahl
SKF Group Headquarters
415 50 Göteborg,
Tel: 031/372626

SPADAB

Göran Lustig
Box 341, 101 24 Stockholm
Tel: 08/13 41 54

STATSKONSULT

Per-Olof Hultman
Statskonsult Admin. Utveckl. AB
Box 4040, 171 04 Solna
Tel: 08/730 03 00

STATSKONTORET

Kerstin Norrby, Lars Hellberg
Statskontoret,
Box 34107, 100 26 Stockholm
Tel: 08/738 45 94, 08/738 47 77

SÖDRA SKOGSÄGARNA

Jerry Nilsson
Södra Data AB,
Box 832, 264 00 Klippan,
Tel: 0435/12090

TELEVERKET

Henry Samuelson
Televerket, ADB-Service
Q24:05, Box 164, 136 23 Haninge
Tel: 08/707 10 00

TELEVERKET

Avd f Grundteknik inkl dotterbolag
Birgitta Carlsson
123 86 Farsta
Tel: 08/ 713 38 51
Karl-Erik Carlsson
Telelogic AB
Box 883, 851 24 Sundsvall
Tel: 060/16 14 44

TELUB TEKNIK AB

Hans Holmberg
Telub Teknik AB,
Box 1381, 171 27 Solna
Tel: 08/730 50 00

UNISYS

Inge Dahlberg
Unisys AB, 171 91 Solna
Tel: 08/55 15 00

VATTENFALL

Georg Karlén
Statens Vattenfallsverk,
Vattenfall Data, 162 87 Vällingby
Tel: 08/739 50 00

VOLVO-DATA

Kenneth Pettersson o Anders Persson
AB Volvo-Data,
Avd 2800, 405 08 Göteborg,
Tel: 031/66 76 48, 66 56 48

VOLVO LASTVAGNAR

Magne Källström
Volvo Lastvagnar AB,
Avd 24180 BC4, 405 08 Göteborg,
Tel: 031/66 66 52 50

VOLVO PERSONVAGNAR

Uno Eriksson
Volvo PV AB,
Avd. 50820 AU, 405 08 Göteborg,
Tel: 031/592074