

SISU

analys

Nr. 85/2

NÅGRA ASPEKTER PÅ KON- TORSINFORMATIONSSYSTEM

SISU – Svenska Institutet för Systemutveckling

Box 515
182 15 Danderyd

Vendevägen 90
Danderyd

08 – 755 29 30

INNEHÅLLSFÖRETECKNING

| | |
|---|----|
| FÖRORD | 0 |
| SAMMANFATTNING | 1 |
| 1. KONTORSINFORMATIONSSYSTEM – KONCEPT, TEKNIKER OCH TRENDER | 3 |
| 2. IRM PÅ SE-BANKEN – MÅL OCH UPPFYLLELSE | 47 |
| 3. KOMMUNDATAS ANVÄNDARANPASSNING AV UTVECKLINGSPROCESSEN | 56 |
| 4. SEKRETERARROLLEN VID ORD- OCH TEXTBEHANDLING | 60 |

FÖRORD

SISU-analys är en tidskrift som utges fyra gånger per år. Varje nummer behandlar ett tema - ett aktuellt problemområde och söker då ge en bild av området ur olika aspekter.

Detta, vårt andra nummer av SISU-analys, har ett tema som utgör en viktig pusselbit i programmet för SISUs område 5, Interaktiva System - Kontorsinformationssystem.

Vi har valt att kalla temanumret för "NÅGRA ASPEKTER PÅ KONTORSINFORMATIONSSYSTEM". Vi gör detta trots att vi är medvetna om att begreppet "kontorsinformationssystem" är illa definierat och leder tankarna åt vitt skilda håll för olika människor. I ingressen tar vi upp just frågan om namngivning på området. Att vi använder prefixet "några aspekter på" är be-tingat av områdets snabba dynamik och kraftiga expansion - det går inte att inom ramen för detta temanummer av SISU-analys ta upp mer än ett fåtal av de många aspekter man kan lägga på utvecklingsområdet Kontorsinformationssystem.

Författare till detta nummer har varit:

- Kapitel 1: Matts Ahlsen, Anders Björnerstedt, Stefan Britts och Lars Söderlund, alla från SISU.
- Kapitel 2: Peter Söderström, SE-Banken.
- Kapitel 3: Leif Ortman, Kommundata.
- Kapitel 4: Gunilla Bradley, Cecilia Bergström och Lilleba Sundberg, alla från Sociologiska Institutionen vid Stockholms Universitet.

SISU tackar alla medverkande för deras bidrag.

SAMMANFATTNING

När ett "område" är i sin linda är det kanske oundvikligt att ett flertal, motstridiga bilder finns av vad området omfattar och hur det utvecklas. I vilken mening är "Kontorsinformationssystem" ett "område"? Möjligen råder enighet om att det är ett affärsområde - växande men inte välavgränsat. Ett stort antal produkter marknadsförs idag med hänvisning till kontorsinformationssystem som område och kontorsautomation som koncept. Med nyvaket intresse börjar intressenterna på denna marknad också observera behovet av att sälja tjänster som komplement till rena produkter och system.

Är Kontorsinformationssystem ett forsknings/utvecklingsområde? Är det ett metod/teknik-område? Det finns för närvarande goda anledningar att betrakta det så. Det pågår mycket stora satsningar internationellt, dels på grundläggande forskning om koncept och metoder, dels på tillämpad forskning syftande till att ta fram hela system med ny design. Det bör noteras att de koncept, metoder och design vi då talar om, ibland inte alls är helt nya. Ofta är det fråga om att vidareutveckla existerande koncept, metoder och design för att möta nya krav.

Då kommer vi till en intressant slutsats - att ett nytt forsknings/utvecklings-område mycket väl kan födas genom att ett antal nya möjligheter och problem sammanflätas i tiden. Till exempel har en hel serie nya MÖJLIGHETER till lagring, överföring och behandling av information av olika form (såsom text, data, ljud, grafik, bild) vuxit fram som ett resultat av innovationer i den grundläggande teknologin (utveckling av mikrodata, fiberoptik för digital överföring av ljud, bild, text och data, grafiska arbetsstationer, lokala nät mm). Behovet av att i kontorsmiljön ha datorstöd också för behandlingen av text, ljud, grafik och bild har naturligtvis funnits länge. Men täckning av detta behov kommer inte gratis i och med att den grundläggande teknologin finns. Ett antal PROBLEM återstår att lösa, till att börja med behöver en mängd programvara utvecklas för att göra en integrerad behandling av dessa informationsformer möjlig. Utnyttjande av de möjligheter som nämnts ovan parat med ett systematiskt tacklande av de nya problem som successivt uppstod, kom tillsammans att utgöra ett embryo till ett forsknings/utvecklings-område.

Så småningom kom man också till insikt om att de "traditionella" informations- och datamodelleringsstekniker vi har (och som utvecklats i samband med databasteknologins framväxt) kanske inte räcker till, ex vis när det gäller modellering av komplexa objektstrukturer innehållande alla ovan nämnda informationsformer. Nya problem, som alltså i en eller annan mening är specifika för "området", tillkommer ständigt och forsknings/utvecklings-området kan börja växa.

Även om Kontorsinformationssystem inte nödvändigtvis kommer att utgöra ett eget område på lång sikt, finns det alltså fog för att se det som ett forsknings/utvecklings-område just nu. Kapitel 1, som utgör lejonparten av detta nummer av SISU-analys, syftar till att berätta en del om de problem som den internationella forskningen och utvecklingen på Kontorsinformationssystem-området tacklar. Kaptitlet ger en översikt över koncept, tekniker och trender, vilka är av stor betydelse för detta områdes framtida utveckling.

Naturligtvis görs inga anspråk på att valet av de koncept, tekniker och trender skall vara heltäckande. Framförallt har vi valt att utsluta de mest elementära och redan kända. Kunskap om dominerande trender, vitala koncept och konkurrerande tekniker är - som alltid - en nödvändig bas för den, som har ansvar för teknikvärdering inom sin organisation. Vi hoppas att detta kapitel skall kunna ge något litet bidrag åt var och en på den punkten.

Som avslutning tar kapitel 1 i sina två sista avsnitt upp några få specifika forskningsproblem samt ger några lästips för dem, som vill fördjupa sig inom området.

Utöver kunskap om områdets trender, tekniker och baskoncept kommer en väl genomtänkt strategi för utnyttjande av detta teknikområde att vara av avgörande betydelse för organisationens överlevnad i informationsteknologi-eran. IRM-konceptet (Information Resource Management) har börjat etablera sig allt fastare som ett begrepp i diskussionerna om strategier för förvaltning av organisationsgemensam information. Kapitel 2 beskriver hur S-E-Banken sedan flera år etablerat och tillämpat en policy för delar av en IRM-strategi.

För att få en aktiv utveckling av informationsbehandlingen inom en organisation krävs att man kontinuerligt bevakar och värderar utvecklingen av verktyg, metoder och modeller. I kapitel 3 ges ett exempel på sådan bevakning. Där berättas om hur Kommundata ändrat strategi och gått mot en användaranpassning av utvecklingsprocessen.

Kontorsinformationssystem är också socio-tekniska system. I snabb takt får de problem, som ryms inom detta fält, ökad uppmärksamhet. Forskningen inom fältet har kommit igång sent men har under de senaste åren vitaliserats betydligt av områdets tekniska utveckling. Ord- och textbehandling blev det först etablerade datorstödet för kontorsarbete och sekreterarna var detta verktygs "försökskaniner". Därigenom kom sekreterarrollen tidigt i fokus för intresset av organisatoriska effekter av de nya hjälpmedlen. Det avslutande kapitlet redogör för en studie just om sekreterarrollen vid ord- och textbehandling.

1. KONTORSINFORMATIONSSYSTEM – KONCEPT, TEKNIKER OCH TRENDER

1.1 Introduktion

Begrepp som "kontorsautomation" och "kontorsinformationssystem" har en längre tid använts för att representera en mängd olika saker, allt ifrån användning av (numera) konventionella hjälpmedel såsom ordbehandling, elektronisk post etc, till nya typer av interaktiva system, vilka är objektorienterade och starkt decentraliserade.

Vi skall förvisso inte ägna oss åt att debattera lämpliga etiketter, men många frågar sig om en etikett som "Kontorsinformationssystem" har något berättigande över huvudtaget. Är det kanske bara fråga om ett nytt tillämpningsområde? Svaret beror delvis på hur högt man riktar sin blick, om man ser "det lilla eller det stora" i det som sker.

Å ena sidan är det ju så, att vare sig begreppet representerar det första eller det sista av synsätten i första stycket ovan, kommer den långsiktiga utvecklingen att sammansvetsa dessa hjälpmedel (alt. system) med "traditionella" ADB-tillämpningar på ett sätt som rycker undan grunden för att hålla kvar den speciella etiketten.

Å andra sidan kan man se - om man tittar noga - att den nya tekniken, som växer fram bakom denna etikett, delvis representerar nya ideer om hur datorstöd skall byggas och användas och den introducerar ett flertal nya koncept. Vi kan skönja nya krav på våra konventionella metoder för datamodellering, systemutveckling och systemförvaltning.

Vi skall ägna denna uppsats om "ämnet" Kontorsinformationssystem åt en bedömning av den tekniska utvecklingen i sig och framför allt åt en beskrivning av - som vi ser det - några viktiga koncept i och egenskaper hos de system, som idag ligger på ritborden i de mest offensiva laboratorierna världen runt. (För enkelhets skull kommer vi ofta i denna skrift att använda förkortningen KIS för Kontorsinformationssystem)

En stark trend just nu styr utvecklingen mot "integrerade system av datorstöd för kontorsarbete". Sådana system inkluderar en mängd olika dator- och kommunikationsstöd, såväl för den enskilde administratörens arbete som för gruppgemensamma och organisationsgemensamma arbetsuppgifter.

Det finns två starka motiv för utvecklingen av "integrerade dator- och kommunikationsstöd för kontorsarbete". Det första är en ren produktivitetsdrivkraft. För att en organisation skall kunna hantera den starkt ökande informationsvolymen finns ett stort behov att effektivisera alla former av kontorsarbete - såväl för handläggare och administratörer som för sekreterare.

Hittills har så gott som all ansträngning inriktats på att effektivisera sekreteraruppgifter med hjälp av ord- och text-behandlingssystem och liknande stöd. Emellertid har på senare tid intresset inriktats allt starkare på utveckling av datorstöd för det mer "ostrukturerade" kontorsarbetet, ofta kallade beslutstödsystem m.fl. andra namn.

Det andra motivet har att göra med att vi ser en ökande komplexitet i organisationernas beslutsprocesser och informationsförsörjning. Traditionella former för kommunikation såsom telefon, post och personliga möten är ej tillräckliga. Nya inslag av asynkron kommunikation kommer att utgöra ett verkligt starkt stöd för administrativt arbete. Likaså behöver de existerande formerna för informationssökning i organisationsgemensamma databaser kompletteras med effektivt datorstöd för lagring, återvinning och arkivering av information i personliga och grupp-gemensamma databaser. Informationssökning i organisationsexterna databaser behöver också förbättras.

Vi vill påpeka att denna uppsats betraktar utvecklingen ur ett rent tekniskt perspektiv. Vi skall börja med en allmän överblick över vad som är karakteristiskt för KIS, för att sedan fördjupa oss något lite i en introduktion av några av de starka forskningstrendernas viktigaste koncept - decentraliserad systemarkitektur, integrering, flexibilitet, dialogmodellering och objektorientering.

1.2 Kontorsinformationssystem:

Bakgrund och utveckling - en kritisk analys

1.2.1 Översikt

I detta avsnitt ger vi en bakgrund till utvecklingen av KIS-området samt en översikt av detta. Vi kommer bl a att beröra olika sätt att se på KIS-området och dess utveckling, datorstöd som erbjuds för kontorsverksamhet samt utmärkande drag för området.

Terminologi

Som antytts i inledningen förekommer en rad olika benämningar på området vid sidan av kontorsinformationssystem (KIS). Den vanligaste är "kontorsautomation" (KA), vilken vi kort vill kommentera. Denna benämning är en aning missledande. Introduktionen av datorstöd på kontoren innebär i själva verket mycket lite av automation. Ta t ex ordbehandling och elektronisk post. Vid användningen av dessa hjälpmedel sker inmatning av text ungefär som tidigare med skrivmaskin. Det som sker automatiskt är textformatering och förmedling av brev etc., vilket enbart utgör en liten, om än tidskrävande, del av allt arbete. Somliga undersökningar talar om att mindre än 10-20% av arbetsuppgifterna över huvud taget kan automatiseras, dvs helt tas över av ett datasystem. Anledningen är att arbetsuppgifterna på ett kontor är av en sådan art att de i hög grad fordrar mänskligt omdöme och kompetens för att kunna utföras.

Andra vanliga benämningar på området är framtidens kontor, det papperslösa kontoret, det elektroniska kontoret och det integrerade kontoret. Även dessa benämningar är missvisande. Framtidens kontor anknyter till tidsdimensionen och medför därför att allt vi talar om ligger i framtiden, vilket inte alls är fallet. Det papperslösa kontoret är mest en utopi och kommer troligen att så förbli. Möjligen bidrar de nya hjälpmedlen till att minska pappersmängden. Det elektroniska kontoret är en något för snäv benämning som enbart fokuserar de elektroniska hjälpmedlen, dvs i första hand datorerna, men glömmer bort andra komponenter i kontorsverksamheten, främst människorna. Det integrerade kontoret slutligen är också något missvisande då inte allt kommer att integreras inom ramen för ett och samma system.

Av dessa skäl har vi valt att benämna området KIS. Namnet säger precis vad det är fråga om, nämligen datorstödda informationssystem på kontor. Internationellt är också "Office Information Systems" (OIS) en av de mest gångbara benämningarna.

1.2.2 Datorstöd för kontorsverksamhet

Vad kan man då tänka sig för olika typer av datorstöd på ett kontor? (En kortare och mer koncis benämning för datorstöd allmänt är verktyg, varför vi i fortsättningen kommer att använda denna.) I klargörande syfte kan verktygen klassificeras i olika grupper beroende på användningsområde. En tänkbar indelning är följande:

- textbehandling,
- dokumenthantering,
- databashantering,
- analytiska verktyg,
- resursadministration,
- affärsgrafik,
- kommunikation,
- särskilda verktyg.

Ord- och textbehandling (OTB) torde vara ett av de mest spridda verktygen inom denna kategori. Verktuget omfattar allt från grundläggande redigeringsfunktioner till formatering och stavningskontroll. Även typsättning kan ingå här. Dessutom kan man tänka sig funktioner för kontroll av den språkliga behandlingen avseende grammatik, upprepningar, användning av ovanliga ord etc.

En utökad form av texthantering är dokumenthantering. Dokument innehåller vanligen också text men har dessutom en bestämd struktur som inom vissa ramar tillåts variera. Kopplade till varje dokument kan dessutom finnas viss systeminformation såsom skapelsedatum, skapare, senaste redigeringsdatum mm. Ofta brukar dokumentkonceptet generaliseras till att förutom text även omfatta t ex bilder, tabeller och grafer. Idag pågår arbete på flera håll för att etablera en standard för dokument, t ex inom GILT-projektet, IBMs Document Contents Architecture (DCA), och ISOs Office Document Architecture.

Verktyg för databashantering hör också till de mer spridda. Vanligast är informationssökning baserad på ett enkelt frågespråk, typ SQL eller Query-by-Example. Databasen användarna arbetar mot kan vara både privat, gemensam för en grupp (t ex en avdelning) eller för hela företaget/organisationen. Även publika databaser kan naturligtvis förekomma. Ofta kompletteras de enkla frågespråken med rapportgeneratorer och 4:e generationsspråk. Dessa tillåter mer komplicerade sammanställningar av data och erbjuder dessutom ofta möjligheter till enklare programmering.

Till de analytiska verktygen räknas sådana för analys och beslutsstöd. Matriskalkylprogrammen (spreadsheets) har på bara några år vuxit kraftigt i både antal och styrka trots (eller tack vare?) sin enkla uppbyggnad. I funktionaliteten påminner

de om rapportgeneratorer men med den stora skillnaden att data kan matas in i dialog med datorn. Ett stort användningsområde har därför blivit interaktivt modellbyggande och "what if...?" analyser. Beslutstödssystem (BSS) kan sägas representera en generalisering av olika former av analytiska verktyg där komplexiteten hos de beslutsstödande verktygen varierar högst avsevärt.

Verktyg för resursadministration är enkla men användbara. Resurser som kan administreras är både personer samt gemensamma lokaler och utrustning. Kalendersystem är den vanligaste benämningen på verktyg för administration av persontid. Ett kalendersystem fungerar i princip som en almanacka, fast på elektronisk väg. Man kan här boka in tider man är upptagen vissa veckor, dagar eller klockslag. I mer avancerade system tillåts också dubbelbokning med prioriteringar mellan olika inbokningar. Ofta kan också en påminnelsefunktion kopplas till kalendersystemet. Ett av de största problemen med personliga kalendersystem är att de vanligen måste fyllas i och uppdateras parallellt med en vanlig almanacka därför att kalendersystemet är svårt att nå om man rör sig inom eller utanför kontoret.

Affärsgrafik, eller presentationsgrafik som det också kallas, är ett verktyg för att omvandla rådata till olika former av grafiska framställningar. Rådata utgörs oftast av tabeller med siffror. När de matas in till affärsgrafikpaketet omvandlas de till cirkeldiagram, stapeldiagram eller någon annan grafisk representation.

Kommunikation är, vid sidan av OTB, det kanske mest omtalade kontorsverktyget. System för elektronisk post (t ex Memo) och konferenssystem (t ex QZ:as KOM-system) har blivit mycket populära inom många företag och organisationer. Dessa används regelbundet för kommunikation mellan både enskilda individer och inom grupper av individer.

För kommunikation mellan organisationer har både publika kommunikationstjänster, såsom Teletex, tagits fram liksom utvidgade meddelandehanteringssystem. Den senare typen av system bygger ofta på det ovan nämnda generaliserade dokumentkonceptet, exempelvis IBMs Document Interchange Architecture (DIA) och GILTs CBMS-modell (Computer Based Message System).

De verktyg vi har behandlat ovan är i huvudsak av generell natur och är därför inte direkt knutna till en viss tillämpning. Givetvis finns det även i kontorsmiljö behov av datorstöd för de traditionella administrativa rutiner vi nämnde inledningsvis, liksom av andra skraddarsydda applikationer. Dessa applikationer faller in under den gemensamma benämningen särskilda verktyg.

Klassificeringen av verktygen ovan avsåg att bibringa KIS-området viss struktur. Indelningen täcker dock ej in alla typer av verktyg. Här har främst behandlats verktyg som vänder sig till slutanvändare.

1.2.3 Perspektiv på KIS

Idag är bilden av KIS-området mycket splittrad. Anledningen torde vara dels att en mängd produkter under senare år lanserats kommersiellt inom området, dels att terminologin inom området ännu ej stabiliserats.

KIS kommersiellt

Kommersiellt är problemet inte så mycket mängden av produkter som sådan, utan vidden av deras användningsområde. Inom området kan man finna allt från enkla ordbehandlingssystem till integrerade programprodukter omfattande många funktioner. Där emellan kan man hitta bl a informationssökning (-återvinning), elektronisk post och persondatorer. Samtliga produkter gör dessutom anspråk på att representera "kontorsautomation". Skälet till att allt möjligt plötsligt lanseras på detta vis är troligen att KIS-området blivit på modet.

Ytterligare ett skäl till oredan är områdets bakgrund. Historiskt har tre typer av leverantörer av utrustning kommit att direkt involveras:

- datorleverantörer,
- leverantörer av kontorsutrustning och
- leverantörer av kommunikationsutrustning.

Datorleverantörerna representerar hela spektrat från stor-dator- till mikrodatorsidan. Kontorsutrustning står för t ex skrivmaskiner och ordbehandlingssystem. Kommunikationsutrustning slutligen omfattar lokala nätverk och abonnentväxlar. Alla typer av leverantörer gör anspråk på att tillhandahålla KIS, även om man bara täcker ett av områdena.

Resultatet av produktfloran för konsumenterna är i regel förvirring. Att bara studera marknaden är således ingen god väg att nå kunskap om vad KIS är.

"Akademiska definitioner", å andra sidan, ger heller ingen träffande bild av KIS och vi skall därför ej rekapitulera några sådana här.

En bättre metod att få grepp om KIS torde vara att i stället se till vad som utmärker området i allmänhet och kontorsverksamhet i synnerhet. M a o, vi måste finna karakteristika för området som särskiljer det från andra närliggande områden.

KIS-karakteristika:

Från vad som sagts ovan kan man sluta sig till att KIS har starka beröringspunkter både med traditionell databehandling på stordator och personlig databehandling på mikrodator. Trots att dessa tre områden överlappar varandra är de dock ej identiska. Allmänt kan sägas om personlig databehandling att systemen i huvudsak är avsedda för enskilt bruk. Ett väsentligt inslag i kontorsverksamhet är emellertid att flera personer kan vara inblandade i utförandet av en arbetsuppgift. Fleranvändarsystem är då nödvändigt, helst med gruppstöd. PC-systemens starka sida däremot är användargränssnittet (interaktivitet) vilket ofta är betydligt mer utvecklat än på större datorer.

Vad gäller beröringspunkterna mellan KIS och traditionell databehandling är de minst lika stora som mellan KIS och personlig databehandling. Applikationer av traditionell typ ingår som en del av verktygen i ett KIS (särskilda verktyg). Ofta måste dessa verktyg obehindrat kunna kommunicera med andra KIS-verktyg, t ex affärsgrafik eller rapportgenerator. En av de saker som är nya för KIS jämfört med traditionell databehandling är emellertid att mängden data som skall hanteras i ett KIS normalt sett är mindre och att rutinerna är ej så strukturerade och formaliserade som i en traditionell applikation. KIS-tillämpningar innehåller t ex ofta oplanerade moment där utsökningar i databasen, analyser och sammanställningar blandas på ett sätt som ej går att bestämma på förhand.

Vidare hanterar traditionella databehandlingssystem i huvudsak poster som ingår i hårt fixerade, fördefinierade datastrukturer. I KIS tillkommer nya datatyper såsom dokument, bilder, grafer, röstmeddelanden etc., vilka kan kombineras till godtyckligt komplexa strukturer. Exempelvis kan ett dokument bestå av en rapporttext strukturerad i avsnitt. Dokumentet innehåller vidare bilder och tabeller. Tabellerna kan genereras baserat på evalueringen av frågor mot databasen, t ex varje gång dokumentet tas fram, vid bestämda tidpunkter, eller dynamiskt vid uppdatering av grunddata.

Uppenbarligen finns det en hel del saker som särskiljer KIS-området från närliggande områden.

Det är därför idag motiverat att tala om KIS som ett eget område. Inte minst därför att detta ger oss möjlighet att till fullo analysera och möta de speciella behov en kontorsmiljö ställer. På sikt kommer emellertid gränsen mellan traditionell databehandling och KIS att suddas ut allt mer, för att slutligen försvinna helt. En ny generation av datorstödda, interaktiva informationssystem, betydligt kraftfullare än dagens, kommer därvid att skapas.

1.2.4 Analys

Låt oss analysera KIS-området utgående från de karakteristika vi ovan givit det. I samband med analysen görs också en rekapitulering av tidigare listade karakteristika plus vissa kompletteringar och förklaringar.

Karakteristika: KIS idag

Interaktivitet är ett typiskt kännetecken för KIS-verktyg. Idag finns de främsta representanterna för goda interaktiva verktyg på mikrodatorsidan. Exempel på sådana är XEROX-Star, Apples Lisa, VisiOn och Lotus 1-2-3. På större datorer har man oftast inriktat sig på att tillhandahålla uniforma gränssnitt, t ex formulär eller menyer, vilka dock är tämligen primitiva jämfört med mikrodatorsidan.

Under interaktivitet kommer egenskaper hos användargränssnittet in såsom användarvänlighet och terminaloberoende. Vi kommer i ett senare avsnitt att diskutera detta mer i detalj.

På kontoret pågår en mängd aktiviteter parallellt. Somliga av dessa hänger samman, andra är helt fristående från varandra. Den senare punkten är av störst intresse, särskilt då aktiviteterna fordrar någon form av synkronisering och/eller är distribuerade i tid och rum. Ett typexempel på detta är ärendeadministration. Tag ex.vis. en ansökan om byggnadslov. Efter det att ansökan diarieförts hos kommunen inhämtas synpunkter från berörda instanser varefter beslut kan fattas. Synpunkterna kan framställas och lämnas parallellt av olika personer på olika arbetsplatser (distribution). Beslut kan däremot inte fattas förrän alla nödvändiga handlingar inkommit (synkronisering). Idag saknas i det närmaste helt stöd för ärendehantering.

Integration kan behandlas på olika nivåer t ex vad avser verktyg, utrustning, media etc. Här berör vi främst den första aspekten, verktyg. Utrustning och verktyg av olika slag måste kunna kopplas samman till integrerade system. På PC-sidan har vi redan sk integrerade programpaket (Lotus 1-2-3, Knowledge-man m fl). Dessa är ett steg i rätt riktning när det gäller att koppla samman olika verktyg inom ett och samma system. Man måste emellertid gå längre och bygga system som dels inkluderar fler funktioner, dels i mycket högre utsträckning än idag integrerar funktionerna i respektive verktyg i en fleranvändarmiljö. Dessutom måste nya verktyg kunna utvecklas och integreras med de redan existerande.

En av de viktigaste aktiviteterna på kontoret är kommunikation. Därför är kommunikationssystem av stor betydelse. Man skall inte bara kunna utbyta meddelanden utan också arbetsresultat såsom dokument, analysresultat, bearbetad information,

etc. Eftersom arbetet kan ske distribuerat måste kommunikation vara möjlig inte bara inom företagets/organisationens eget KIS utan också med externa system.

Idag är kommunikationssystemen alltför begränsade genom att de enbart stöder en form av kommunikation, t ex meddelandeförmedling eller filöverföring mellan datorer. Olika typer av kommunikation måste därför integreras till ett kommunikationssystem som används av KIS:et. Inom en inte alltför avlägsen framtid bör dessutom kommunikation av ljud och bild också kunna ske integrerat med andra informationsformer inom ramen för ett KIS (t ex telefon och videokonferenser).

Kontorsarbete sker distribuerat i tid och rum vilket gör att en schemaförändring inte alltid kan tillåtas slå igenom på hela databasen. Tvärtom måste olika versioner av program, dokument etc tillåtas existera parallellt, åtminstone under en övergångsperiod. Här finns idag inga godtagbara former av datorstöd, annat än på forskningslaboratoriernas ritningsbord.

Förutom dessa karakteristika kan man också anlägga andra aspekter på ett KIS. I analysen ovan har vi varit tämligen verktygscentrerade. KIS är emellertid mer än en samling datorstöd. Betraktat i ett vidare perspektiv omfattar KIS många andra komponenter såsom människor, utrustning, information, rutiner och mål för verksamheten. Utrymmet medger emellertid inte en så bred behandling av området.

Låt oss dock slutligen ta upp en ytterligare aspekt på KIS, den ekonomiska. Finns motiv för att investera i KIS?

Ofta brukar ekonomiska incitament föras fram som motiv att satsa på ett KIS. Några av dessa argument lyder som följer:

- Produktiviteten på kontoren ökade 4% under 1970-talet samtidigt som produktiviteten i industrin ökade 80%.
- Investeringarna per anställd i industrin är ca 25.000 dollar. På kontoren är motsvarande siffra högst 1/10 av industrins.
- Personalkostnaden utgör ca 90% av kostnaderna för att driva ett kontor.

Dessa argument skall tas för vad de är - isolerade beskrivningar (ibland baserade på tveksamma mått) av den historiska utvecklingen, snarare än avgörande argument för beslut om framtida investeringar. Om man godtar argumenten kan det påpekas att de siffror som de hänvisar till har varit giltiga i flera decennier, utan att vi fått någon väsentlig ökning i satsningen på teknik på kontor. Förklaringen ligger delvis i att teknisk utveckling ofta sker språngvis. Just nu tar vi ett mycket stort språng genom att språng i utvecklingen av den

"underliggande teknologin" sammanfallit på flera områden såsom digital lagring och överföring av ljud, fiberoptik, datoriserade telefonväxlar, mikrodatorer, flerfunktionsterminaler med grafik och massminnesteknologin. Detta har lett till att först nu har en snabb utveckling av slutanvändarprodukter/system för KIS-området blivit tekniskt möjlig.

I praktiken har det visat sig vara mycket svårt att peka på direkta ekonomiska vinster av införande av KIS. Vad beträffar lönsamheten hos ett av de vanligaste verktygen, ord- och textbehandling, har t ex inga vinster kunnat påvisas. Det enda undantaget tycks vara kommunikationssystem av typen elektronisk post eller konferenssystem.

Att det är svårt att direkt påvisa ekonomiska vinster är dock ej detsamma som att sådana inte finns att hämta. Ofta brukar KIS kvalitetshöjande effekter på arbetet betonas. De anställda får t ex mer tid över för annat än rutinärenden. Som en indirekt effekt av detta kan lönsamheten direkt komma att påverkas i positiv riktning.

Låt oss efter denna allmänt hållna överblick över KIS-området belysa några av de viktigaste aspekterna på framtida KIS: Systemarkitektur, Flexibilitet, Integrering, Objektorientering och Användargränssnittet.

1.3. Systemarkitektur.

Två viktiga trender i användning och uppbyggnad av dagens datasystem är:

- 1- En utökning av domänen av de administrativa "rutiner" eller procedurer som datoriseras eller ges datorstöd. Gränsen mellan manuella aktiviteter och aktiviteter som kan (helt eller delvis) automatiseras har blivit flytande. Gränsen har även flyttats ut från en kärna av storskaliga, rutinbetonade och formaliserbara aktiviteter, till att täcka mer småskaliga, interaktiva och oformaliserade rutiner. En allt större mängd människor arbetar direkt eller indirekt med datasystem. Inom forskning och utveckling återspeglas denna trend i uppkomsten av områden som Kontors Informations System (KIS) och "Computer Aided Design" (CAD).

- 2- En förskjutning mot system som är administrativt decentraliserade och ofta fysiskt distribuerade. Med anledning av den minskande kostnaden för små, men kompletta, datorer blir det vanligare för enskilda avdelningar eller individer i organisationer att införskaffa egna maskiner. Istället för att vara beroende av tjänster från en speciell del av organisationen (dataavdelningen), eller andra organisationer (dataserviceföretag), så har man kontroll över de egna datorresurserna. Sådana system är givetvis avsedda för ett smalare spektrum av applikationer än dagens stordatorer.

Den första av dessa trender handlar om systemens funktionalitet. Denna har både generaliserats och gjorts anpassningsbar till mer lokala informationsbehov. Det är både utveckling av hårdvara (maskincykler kan ödslas på fler uppgifter) och mjukvara (en mängd av interaktiva applikationer) som ligger bakom detta.

En konsekvens av denna utökning i funktionalitet är ökade krav på lokal tillgänglighet och kontroll över informationssystemet. När en större del av verksamhetens uppgifter skall lösas mha ett datasystem blir frågor kring systemets:

- säkerhet,
- tillgänglighet,
- robusthet och
- flexibilitet

desto viktigare. Detta leder oss in på den andra trenden, som handlar om datasystemens arkitektur.

Vi kan tala om ett datasystems arkitektur på åtminstone tre nivåer:

- 1- Fysisk arkitektur, dvs hårdvaruarkitektur.
- 2- Arkitektur av systemprogramvara. Vi menar då mjukvara som i någon mening är generell och identisk över flera installationer.
- 3- Applikationsarkitekturen. Vi menar då den del av mjukvaran som är speciellt anpassad till en viss installation.

Gränsen mellan dessa nivåer av ett systems arkitektur är givetvis flytande. Ett annat problem i detta sammanhang är att det kan vara svårt att exakt peka ut vad som utgör "systemet", dvs var systemets gräns går. Dels måste man ha klart för sig vilken nivå man betraktar systemet på, dels finns ett behov av att göra systemen mer "öppna" och därmed gränsen mellan dem mer oklar.

De senaste åren har förekomsten av persondatorer ökat starkt i företag och organisationer. En schablonmässig modell är en fysisk arkitektur bestående av ett antal persondatorer som eventuellt delar på resurser som radskrivare, hårddiskar, bandstation etc, som är sammanbundna i ett lokalt nätverk. Dock har utvecklingen i realiteten oftare varit ganska okontrollerad, med persondatorer utan möjlighet till resursdelning eller inbördes kommunikation. I detta fall är persondatorn knappast en del av ett större datasystem. Behovet av lokal autonomitet har fått styra denna utveckling. Behovet av global samverkan har ofta lämnats obeaktat. På alla tre nivåerna av arkitektur kan en mindre avdelning fatta beslut om förändringar lokalt efter behov.

Men hittills har införandet av mindre system baserade exempelvis på persondatorer oftast medfört konsekvensen att dessa blivit relativt isolerade. Bristen på tillförlitliga kommunikationsmöjligheter har gjort graden av resursdelning låg, vilket i sin tur tvingar mjukvaran att vara ekonomiserande och förenklad. Det har snarare handlat om billiga och dåliga datasystem än säkra och flexibla system.

När det gäller den fysiska arkitekturen så finns idag möjligheter till relativt billig dator-till-dator kommunikation via lokala nät. Dessa kan bygga på mycket avancerad teknik såsom optiska fibrer, eller på det redan existerande telefonnätet. Det förekommer t o m lösningar som utnyttjar det befintliga lokala elkraftnätet som medium.

På systemprogramvarunivå, i det här fallet programvaran som sköter kommunikationen, är dock situationen fortfarande problematisk. Det fordras oftast relativt mycket kunskap för att få utrustning från olika leverantörer att kommunicera med varandra, trots att de är kompatibla på hårdvarunivå. Standardisering pågår på området om än långsamt.

Andra kommunikationstekniker än lokala nät existerar givetvis, t ex utbyte av disketter eller magnetband. I vissa fall räcker detta, dock tillåter inte sådan kommunikation någon högre grad av resursdelning.

För att det idag realistiskt skall gå att införa ett distribuerat datorsystem, som inte är "desintegrerat", fordras fortfarande en hög grad av diciplin. Standardiseringen av hårdvara och systemprogramvara kommer dock att vidga ramen för möjliga lokala beslut, som är acceptabla vad gäller service och kommunikationsmöjligheter.

1.3.1. Behovet av decentralisering.

Behovet av att decentralisera organisationers datasystem är gammalt. Att decentralisera kontrollen över arbetsuppgifter i en organisation ger fördelar såsom flexibilitet och mer meningsfulla arbetsuppgifter (och därmed ansvar) för anställda. Det är då en fördel om datasystemets arkitektur så långt som är möjligt passar in i organisationens struktur. Att decentralisera kontrollen över ett datasystem, ger även nackdelar, främst en mer komplex styrning.

Möjligheten till mer decentraliserade datasystem har ökat med utvecklingen av smådatorer, billiga massminnen och lokala nätverk. Utvecklingen ligger dock före på den fysiska nivån, medan systemprogramvara som tillåter decentralisering först nu börjar bli tillgänglig. Svårigheterna har framförallt gällt kommunikation inom och mellan system. Mer specifikt så är det svårt att upprätthålla konsistens i systemprogramvaran och åstadkomma koordinering av applikationer. Detta skall åstadkommas över både administrativa gränser i organisationen och hårdvaru/systemprogramvarugränser i systemet.

1.3.2. Decentralisering och Distribution.

I detta avsnitt tar vi åter upp frågan om vad som utgör "systemet" och dess gräns mot omgivningen.

Vi antar att vi i organisationen kan urskilja enheter som i någon mån är självständiga. Vad som utgör en sådan enhet går inte att svara på generellt, det är en modelleringsfråga som det ankommer på modellens konstruktör att definiera. För varje sådan enhet som har behov, eller kan komma att få behov, av datorstöd i någon form, definierar vi en komponent i vårt datasystem som vi kallar en nod. En nod utgör en centraliserad miljö, dvs den administreras och styrs som en enhet. En nod har tillgång till resurser för lagring, bearbetning och kommunikation. Kommunikation sker dels med användare utav noden och dels med andra noder. Det finns alltid en eller flera identifierbara personer som har ansvar för noden. Vi kan föreställa oss - i det enkla fallet - en nod som en dator med tillhörande utrustning

Organisationens struktur i form av "självständiga enheter" återspeglas i datasystemet av en struktur av noder. Varje nod skall erbjuda de informationsfunktioner som den organisatoriska enheten behöver. Användarna som arbetar i enheten skall normalt inte behöva bekymra sig för hur deras nod är implementerad. Vi kan då tala om organisationens datasystem som bestående av ett antal noder som kommunicerar med varandra och om en enhets datasystem så som en nod. En nod har givetvis en intern struktur som vi dock inte går in på här.

Organisationens datasystem är decentraliserat i den meningen att det inte finns någon auktoritet som direkt kan styra hela systemet i detalj. Arkitekturen av organisationens datasystem på fysisk nivå kan se ut i princip hur som helst, förutsatt att den uppfyller de funktionella krav som ställs. Exempelvis kan systemet vara implementerat på endast en dator, t ex en stordator. Alternativt kan systemet vara implementerat på flera datorer som kommunicerar mellan varandra. I det senare fallet säger vi att organisationens datasystem är distribuerat. (Även en enskild nod kan implementeras på en eller flera fysiska datorer.) Noden kan således vara distribuerad. Däremot antar vi att en nod alltid är logiskt centraliserad, dvs den upplevs av användare som endast ett system.

1.4. Flexibilitet och Integrering.

Det kommer allt starkare krav på att kunna samla och ha tillgång till verktyg för informationshantering med olika funktionalitet, inom ramen för ett och samma system. Det har under senare år kommit fram en mängd s.k. "integrerade" programvaruprodukter, eller "allt-i-ett" system, vilkas främsta syfte skall vara att ge användaren en samlad verktygsuppsättning för informationshantering samt i många fall även tjäna som systemutvecklingshjälpmedel.

Förutom att vara kraftfulla skall dessa system även vara lättanvända och tillåta löpande modifieringar av applikationerna. "Flexibilitet" och "integrering" blir honnörsord, men vad menar man egentligen med dessa begrepp och vilka av ett systems egenskaper förverkligas i begreppen?

Med ett flexibelt system menar vi ofta ett system som är "öppet", där öppenheten kan ha två betydelser, nämligen:

- 1) de möjligheter som systemet ger till förändring
- 2) de egenskaper som rör gränssnitt mot omgivningen

En alternativ benämning på dessa två aspekter skulle kunna vara den "inre" resp. "yttre" flexibiliteten hos ett system. Låt oss se närmare på dessa två betydelser av begreppet flexibilitet och karakterisera dem.

1.4.1. Förändring.

All ADB-verksamhet förändras över tiden, orsakerna varierar från yttre omständigheter som direkt påverkar verksamheten (t ex lagändringar), till mer systemspecifika förändringar såsom krav på ökad effektivitet och funktionalitet. I många fall kan införandet av framtida förändringar underlättas genom att applikationer förbereds för detta redan i utformningskedet.

Det går naturligtvis inte att ta hänsyn till alla upptänkliga förändringar, då detta kan resultera i att applikationerna görs alltför generella och därmed mister sin avsedda funktionalitet. Men det krävs stöd i själva utvecklingsmiljön som kan underlätta införande av såväl förutsedda som oförutsedda modifieringar.

Med förändring skall förstås utökning av funktionalitet i form av tillägg av nya funktioner (applikationer) eller modifiering av befintliga. Detta inbegriper naturligtvis också att funktioner ersätts eller helt tas bort.

Vid alla systemförändringar, och speciellt vid vidareutveckling, är det ett krav att kunna utnyttja den redan befintliga programvaran så långt som möjligt. Detta har främst två skäl: ett är att vi vill undvika att "återuppfinna hjulet" så fort

vi har behov av det och därmed hålla graden av redundans nere i systemet. Detta kan ses som en typ av resursdelning där olika program skall kunna byggas på/utnyttja gemensamma komponenter, t ex i form av programmoduler som representerar specifika funktioner.

Ett annat skäl är att vi vill ha möjlighet att stegvis bygga på och/eller förfina existerande applikationer, allt efter det att nya behov/krav uppkommer. Denna form av stegvis utbyggnad (eng. "Incremental development") blir ett kraftfullt hjälpmedel för både ny- och vidareutveckling, under förutsättning att det finns systemstöd för den. Den stegvisa utbyggnaden kan både tjäna som verktyg för s.k. snabb prototyputveckling (jmf. experimentell systemutveckling) och användas för vidareutbyggnad av ett produktionssystem.

1.4.2. Gränssnitt.

Den andra aspekten på flexibilitet, eller "öppenhet", rör ett systems gränssytor mot den omgivning det opererar i. Här kan vi prata om flera olika gränssnitt beroende på från vilket håll vi betraktar systemet. En tänkbar indelning av olika gränssytor kan vara följande:

- användargränssnitt
- kommunikationssnitt mot andra system
- gränssnitt för flyttbarhet (portabilitet)
- enhetsoberoende gränssnitt

Närmast till hands ligger kanske användarsnittet. Detta behandlas detaljerat i ett senare avsnitt (1.6), här nöjer vi oss med att notera den ökade vikt som läggs vid bra användargränssnitt.

Vi har även gränssnitt på andra nivåer i systemarkitekturen, ex vis för kommunikation med andra system eller applikationer. Dessa gränssnitt kan inbegripa diverse kommunikationsprotokoll för dataöverföring mellan geografiskt distribuerade delar av ett system, ev. med utnyttjande av externa/publika tjänster. Här ingår frågor rörande nätarkitekturer och standardisering (jmf OSI-modellen).

De två sista snitten har att göra med oberoende, i det första fallet av ex vis viss systemprogramvara, i det andra, av olika periferienheter. Att göra system fullständigt oberoende och därmed portabla mellan olika operativsystem är naturligtvis omöjligt. Det handlar här snarare om att göra graden av beroende så liten som möjligt. (Om gränssnittet mot ett underliggande OS är väldefinierat och överblickbart, ökar portabiliteten även om en större mängd anpassningar ändå måste göras.

Vi kan också se ett behov av väldefinierade gränssnitt mellan komponenterna inom ett system. Applikationer som t ex byggs ovanpå ett visst DBHS bör ha ett klart gränssnitt mot det senare, för att underlätta eventuell övergång till ett annat DBHS. Här handlar det kanske mer om "utbytbarhet" genom modularisering (se nedan) än om portabilitet.

Beträffande enhetsoberoende (eng. "Device Independence") så bör ett system innehålla alternativa drivrutiner för olika typer av I/U-enheter, och automatiskt se till att rätt rutin används vid rätt tillfälle. Detta för att tex möjliggöra att bitmap-orienterade bildskärmar dynamiskt kan ersättas av teckenorienterade skärmar. Enhetsoberoendet skall då resultera i att ex vis ett dokument kan presenteras på ett sådant sätt att struktur och innehåll framgår trots att presentationsmediets egenskaper kan variera mellan olika sessioner.

1.4.3. Integrering.

Integrering innebär generellt sett att komponenterna i ett system (eller delar därav) kan sammanföras till ett slags helhet. Ett "system" utgör naturligtvis i sig ett slags helhet bestående av sina delfunktioner och informationstillgångar, så för att kallas "integrerat" måste något mer specifika karakteristika tillföras. Om vi utgår från en mängd funktioner såsom OTB, Matriskalkyl, Meddelandehantering, Dokumenthantering, DBHS etc, så kan vi med integrering syfta på de sätt som vi kan utnyttja funktionerna tillsammans, t ex att kunna använda OTB-funktioner för hantering av elektronisk post, att använda DBHS frågespråk i formlerna för en matriskalkyl etc.

Allmänt speglar termen "integrering" på olika sätt hur komponenterna är relaterade till varandra och mot användare av systemet. Detta är emellertid en alldeles för vag beskrivning. Programvara kan integreras utifrån olika principer och vi kan prata om integrering på olika nivåer i en systemarkitektur. Dessutom kan målen för integreringen vara olika, i vissa fall är det enbart frågan om att skapa ett lättanvänt och sammanhållet slutanvändargränssnitt (typ konventionella menysystem), i andra fall kan förutom detta målet vara att även öka funktionaliteten.

1.4.3.1. Olika slags integrering.

Som nämnts ovan kan integrering betyda att olika funktioner kan användas på ett inbördes ömsesidigt sätt, funktion A kan utnyttja funktion B och omvänt, ett slags "funktionsintegrering".

Ett annat sätt att se integrering är att utgå ifrån den information som funktionerna hanterar, närmare bestämt hur denna modelleras och lagras. Integrering genom att använda en enhetlig struktur och gemensamma dataformat var ju en av grundideerna bakom databastekniken. Konventionella DBHS är ju dock begränsade vad gäller struktur/format, och det vi idag saknar är möjligheter att på ett enhetligt sätt lagra och bearbeta olika informationsformer, såsom text, data, ljud och bild. Lika viktigt är att kunna relatera informationsobjekt från olika funktioner, tex att låta en matriskalkyl ingå som en del i ett textdokument, och även att kunna hantera olika informationsformer inom ett objekt, såsom text och ljud i ett elektroniskt brev. En enhetlig hantering av olika informationsobjekt och informationsformer är en förutsättning för integrering på funktionsnivå.

Det är svårt att identifiera olika ansatser till integrering. De flesta bygger dock på funktions- och dataintegrering i varierande grad. Vi skall nedan kort nämna tre olika integreringsstilar och vad de innebär. Indelningen, som bygger på:

- oberoende integrering
- beroende integrering (integrering via värdfunktion)
- synergistisk integrering

behandlas närmare i avsnitt 1.6.

1.4.3.2. Integrering på olika nivåer.

I likhet med ett systems "öppenhet" i termer av olika gränssnitt, kan vi tala om integrering på olika nivåer i en logisk systemarkitektur. Man kan tänka sig följande indelning:

- användargränssnitt
- funktion
- modul

Användargränssnittet kan ur vissa hänseenden betraktas som ett medel för att åstadkomma integrering, utan att man behöver ha tillämpat någon form av integrering inom de underliggande nivåerna i systemarkitekturen. Ett gränssnitt skulle tex kunna sägas "utgöra ett ramverk för en samling funktioner" och samtidigt kanske också realisera "enhetlighet i syntax" mellan olika operationer som kan förekomma i ett flertal funktioner ("sök", "lagra", "sänd" etc.).

Det förefaller rimligt att operationen för att ex vis lagra en grafisk bild ej skiljer sig från dito för att lagra ett textdokument. Den syntaktiska likheten till trots kan emellertid operationerna ha olika semantisk betydelse. Att "lagra" ett textdokument respektive en bild kan betyda olika saker "under ytan". I det första fallet kan "lagra" också innebära att fritextsökning senare kan utnyttjas för att få tag i dokumentet, emedan detta kanske ej är möjligt för eventuell text som ingår i den grafiska bilden. Detta kan bero på att olika principer de facto tillämpas för att lagra dessa informationsobjekt, trots att detta ej framgår i användargränssnittet. Man skulle kunna säga att "integrering endast finns på ytan", i form av kommandon med enhetlig syntax man endast delvis enhetlig semantik.

Med integrering på funktionsnivå/verktygsnivå menar vi det sätt på vilket olika urskiljbara funktioner samverkar. Detta kan ske genom funktions och/eller data integrering enligt någon av de stilar som nämnts ovan.

Då vi pratar om integrering på modulnivå i systemarkitekturen utgörs komponenterna av programmoduler vilka fungerar som byggstenar för funktioner och verktyg. En modul kan tänkas implementera en viss funktion eller ett visst verktyg, men kan också tänkas utnyttjas av flera funktioner.

Denna integrering är på implementeringsnivå. Den kan åstadkommas genom en modulariseringsteknik på lämplig nivå, där man, istället för att implementera ett verktyg i en eller i ett fåtal större programmoduler, bryter ned det i delfunktioner som var och en blir en mindre modul. Tanken är att undvika redundans genom att göra moduler tillräckligt generella för att kunna ingå i flera verktyg, men även att skapa överblickbarhet och säkerhet i implementeringen. Modulariseringen ger oss också möjlighet att implementera väldefinierade gränssnitt mellan komponenterna i ett system, vilket vi nämnt betydelsen av ovan. Modulariseringen blir därmed en viktig grund för, och underlättar, integrering på högre nivåer.

1.4.4. Flexibilitet vs. Integrering.

Låt oss slutligen kort betrakta förhållandet mellan "flexibilitet" och "integrering". Rent intuitivt förfaller det finnas en motsättning mellan dessa två begrepp. En stark integrering eller sammankoppling av funktionerna i ett system skapar ju också inbördes beroenden vilka kan tänkas motarbeta kravet på flexibilitet i termer av föränderlighet.

Omvänt kan tyckas att hög flexibilitet skulle tvinga oss att låta funktionerna vara så inbördes oberoende som möjligt, för att därigenom minska effekten av lokala förändringar. Givetvis rör det sig om en avvägning som måste göras redan i ett systems utformningsskede. Klart är dock att de designbegrepp och den modell som används vid utformning av programvaran, bör inbegripa ett modulärt synsätt som skall vara tillämpligt inom de olika nivåerna i den logiska systemarkitekturen. En modulärt orienterad modell för konstruktion av programvara som också inbegriper en enhetlig syn på program och data, är ett kraftfullt hjälpmedel därvidlag.

1.5. Objektorienterade System.

Termen 'objekt' förekommer i många olika betydelser inom informationsbehandling. I KM-sammanhang (KM (Konceptuell Modellering), se SISU analys 85/1) och inom ramen för olika systemutvecklingsmodeller kan man prata om informations- och/eller processobjekt, lexikala och icke-lexikala objekt etc. Objekten utgör här abstraktioner av begrepp och företeelser i någon verksamhet, och blir ett första steg mot en formalisering av information (och dess ev. härledningsregler) med syftet att kunna representera och bearbeta den i ett datasystem.

De "objekt" vi här skall beskriva, inom ramen för det som kallas Objektorienterade system, kan sägas utgöra ett slags mellanting av modelleringskoncept och implementeringsverktyg. Vissa KM-ansatser rubriceras ibland som "objektorienterade", dessa skall dock inte förväxlas med den "objektorientering" som karakteriserar den klass av datasystem som är aktuell här. (Även om det finns beröringspunkter mellan KM-ansatser och det vi här kallar objektorientering, så är de objekt vi talar om här mer "datornära", och specifikt skall de vara exekverbara, dvs direkt implementerbara i ett existerande formellt datasystem.)

1.5.1. Begrepp och Historik.

Låt oss börja med att göra några informella definitioner av begrepp. Det är här lämpligt att varna för att de definitioner vi ger inte alltid är allmänt etablerade. Det mesta av den terminologi vi använder här tillhör dock etablerad terminologi inom forskningen på området.

Ett Objekt är en slags "behållare" för både program och datastrukturer, som har en oberoende existens i ett system som vi kallar för ett objekthanteringssystem. Med "oberoende existens" menar vi att objekt är den minsta enskilt identifierbara enhet som kan förekomma i systemet. Objekt är representerade av datastrukturer kallade attributvärden. Objekt kan identifieras (utpekas) av oss som användare av systemet, eller av andra objekt. Attributvärden kan dock inte direkt identifieras, utan är endast tillgängliga inom ramen för det objekt de tillhör. Attributvärden är beroende av det objekt de tillhör och tillhör alltid ett och endast ett objekt.

Alla objekt är förekomster av någon objekttyp och objekthanteringssystemet har faciliteter för att definiera nya objekttyper och förändra gamla. Objekttyper liknar det som kallas för abstrakta datatyper. Med detta menas att de är datatyper som definierats av någon användare (i allmänhet någon programmerare), dvs de är inte fördefinierade datatyper i det programspråk som används. Vidare kan manipulering av förekomster av en objekttyp endast göras enligt vissa regler som specificerats i objekttypen. Dessa regler representeras av operationer på ett objekts attributvärden.

Med Objektorienterad Programmering förstås användandet av programspråk vars primära struktureringsenhet utgörs av dylika objekt och objekttyper. Därtill skall också finnas primitiver (grundläggande operationer) för kommunikation mellan, och synkronisering av, objekt. Ett Objektorienterat System är ett kombinerat utvecklings- och målsystem för applikationer byggda på strukturer av objekt. Systemet innehåller databashanteringsfunktioner för lagring/sökning i objektdatabaser. Slutligen kan ett Objekthanteringssystem sägas utgöra kärnan i ett objektorienterat system, med funktioner för koordinering/synkronisering, transaktionshantering, åtkomstkontroll etc. Alltså i stort sett ett slags operativsystem.

Det har påståtts att objektorienterad programmering under 80-talet skulle bli vad strukturerad programmering var under 70-talet - alla skulle vara för det, tillverkare påstå sig kunna erbjuda det, programmerare använda det (på olika sätt) emedan ingen egentligen skulle veta vad det innebar. Riktigt på detta sätt har det väl inte blivit (än). Mer formella "definitioner" än som givits ovan är svårt att ge, däremot kan man peka på en mängd egenskaper som får anses mer eller mindre accepterade och som kan sägas karakterisera objektorientering och objektorienterade system. Vi kan börja med det "historiska" ursprunget.

Många av ideerna bakom objektorienterade system kommer från utveckling av operativsystem såväl som från olika programspråk. På operativsystemsiden kom redan i början på 60-talet olika arkitekturer fram där man såg systemet som bestående av en mängd likvärdiga resursobjekt. Några av de tidiga operativsystemens principer för adressering och skydd av minnessegment, generaliserades till att omfatta även andra "systemobjekt" och inte enbart segment i minnet. Ett objekt var från operativsystemets synvinkel helt enkelt en resurs som måste administreras, som t ex en fil, en skrivare eller en arbetsarea. Grundtanken var här att få fram ett enhetligt sätt för både adressering och skydd av dessa resurser. Detta gjordes med s.k. "capabilities", ett slags nyckel bestående av en adress/id till ett objekt samt ett antal åtkomsträttigheter för objektet ifråga. Ett visst program (objekt) förses med ett antal dylika nycklar, som definierar ett domän av objekt och möjligheter till manipulation ("capabilities") av dessa. Ett antal modernare operativsystem använder denna princip, dessa kallas ofta "Capability-Based Systems".

Flera grundläggande drag inom objektorientering kommer från ansatser till modularisering och dataabstraktion i programspråk. Olika modulkonstruktioner finns i en mängd språk, som tex Simula, Clu, Modula, Ada. Dessa är bla avsedda att hålla samman logiskt relaterad information, samtidigt som irrelevanta implementeringsdetaljer kan döljas från användaren/programmeraren. Simula är, med sitt Klassbegrepp, en av föregångarna inom området vad gäller språkkonstruktioner för Abstrakta Datatyper (ADT). Mycket arbete har också lagts ned på formell specifikation och verifiering av ADT. I objektorienterade pro-

gramspråk utgör principerna för ADT en väsentlig del. Det idag mest välkända objektorienterade språket är SmallTalk80.

Traditionellt betraktas data och procedurer (eller programkod) som två separata begrepp. Ett av målen med Objektorienterad programmering kan sägas vara att minska denna distinktion mellan data och program, mellan bearbetad och bearbetande information. I bägge fallen rör det sig ju om information, den vi vill manipulera (data) och den som vi manipulerar denna med (program, procedurer, funktioner, operationer). I vissa fall vill man kanske också betrakta den senare sortens information som just "data" för att kunna göra förändringar. Detta kan underlättas om vi betraktar data och program på ett enhetligt sätt. En programmerare producerar "data" i form av programkod som i ett senare skede tolkas som operationer. I ett objekthanteringssystem är därför objekttyper också objekt i systemet. Objekttyper är då förekomster av en viss objekttyp som vi kan kalla en meta-objekttyp. Det är detta som utgör grunden till att ett objekthanteringssystem blir både utvecklings- och applikationsmiljö.

1.5.2. Objektbegreppet.

I ett objektorienterat system finns endast ett begrepp för att representera både information och dess bearbetningsregler, nämligen Objekt. Dessa har, oavsett om de representerar "data", "program" eller både och, vissa gemensamma karakteristika som gör att de förtjänar sitt gemensamma epitet. Dessa egenskaper rör objektens inre och yttre utseende, hur de kommunicerar sinsemellan, deras tillstånd (eller status) och hur objekt refereras och relateras inbördes. Generellt för alla objekt gäller att de har:

- lagrings- och bearbetningskapacitet
- ett stabilt internt tillstånd
- adresserbarhet
- en specifikation
- en implementering

Nedan utvecklas dessa viktiga egenskaper ytterligare något.

Lagrings- och bearbetningskapacitet.

Som nämnts innehåller ett objekt dels attribut (datastrukturer) för lagring av information, samt operationer för att bearbeta denna. En viktig egenskap hos ett objektorienterat system är den säkerhet som ges i implementeringen genom att åtkomst till den inre strukturen hos ett objekt kan kontrolleras. Attribut och vissa operationer kan "gömmas", och all åtkomst till ett objekt sker via ett utåt synligt gränssnitt bestående av ett antal operationer. Detta brukar kallas informationsskydd eller kapsling (eng. "information hiding/encapsulation") och är en viktig egenskap hos abstrakta datatyper.

Operationerna i ett objekt kan betraktas som ett slags procedurer (i likhet med motsvarigheten i procedurorienterade programspråk). En operation aktiveras genom ett anrop, för att få något utfört. En operation i ett objekt kan vara realiserad mha en eller flera lokala procedurer i objektet, och även använda sig av operationer på andra objekt.

Aktivering av operationer kan i princip ske på två olika sätt; genom ett vanligt hierarkiskt proceduranrop eller genom att använda s.k. mönsterstyrd aktivering (även kallad Meddelandeöverföring, eng. "Message Passing"). Skillnaden mellan dessa är dels implementeringsbetingad, men påvisar också en (om dock subtil) distinktion mellan två kommunikationssätt. I det första fallet aktiverar anroparen en operation via dess namn följt av ev. parameterlistor. Det är det anropande objektet som har initiativet eller kontrollen, och det anropade objektet används som ett slags procedurbibliotek. I det andra fallet sker aktiveringen genom att ett meddelande, bestående av ett operationsnamn och ev. parametrar, "skickas" från ett objekt till ett annat. Det senare matchar meddelandets innehåll mot sina operationer och avgör därigenom vilka operationer som skall aktiveras.

De operationer som kan appliceras på ett objekt kan liknas vid subrutiner i den meningen att det krävs ett anrop från ett annat objekt för att operationen skall utföras. Det anropande objektet suspenderar (avbryter) i allmänhet sin exekvering medan den anropade operationen utförs. När denna är klar så returneras eventuella parametrar till det anropande objektet och exekvering återupptas där.

För att möjliggöra parallell exekvering av flera "program" i systemet kan vi tillåta att en objekttyp definierar ett speciellt slags operation kallad "Exekvera". Denna representeras av en sekvens exekverbara satser kallad objektets satskropp. Att aktivera ett objekts satskropp kallas följaktligen för att exekvera objektet. När ett objekt exekverar ett annat objekt så suspenderas inte exekveringen i det anropande objektet. Både det anropande såväl som det anropade objektet kommer därför att exekvera parallellt.

Internt tillstånd och adresserbarhet.

Ett objekt har vid en viss tidpunkt ett avgörbart inre tillstånd eller status. Detta representeras av de värden objektet har i samtliga sina attribut. Flera objekt kan emellertid ha identiska värden i sina attribut trots att de är olika objekt. För att objekthanteringssystemet skall kunna skilja på objekt förses de med unika identifierare. Dessa används exempelvis i referenser som ett visst objekt kan ha till andra objekt. Dessa referenser kan t ex användas för att skapa aggregeringar av objekt.

En viktig egenskap hos ett objektorienterat system är att objektens tillstånd är stabilt i den bemärkelsen att deras attribut behåller sina värden oavsett om de befinner sig i databasen eller under exekvering (1). Endast anrop av någon operation kan påverka attributen. Ett objekts tillstånd inbegriper även dess exekveringstillstånd, dvs om och, i så fall, hur långt exekveringen av dess satskropp har hunnit. Detta är nödvändigt för att objekthanteringssystemet skall kunna återuppta exekveringen av objekt som på ett eller annat sätt blivit avbrutna.

Specifikation och implementering.

Den sista punkten av karakteristika för objekt har att göra med utseendet och strukturen hos dem, och hur de beskrivs. Den rent syntaktiska framtoningen av objekt blir naturligtvis beroende av det (eller de språk) som används i det objektorienterade systemet. I allmänhet separeras dock de yttre egenskaperna hos ett objekt från de inre genom en Specifikations- resp. Implementeringsdel. Specifikationen innehåller operationshuvuden med ev. parameterlistor och resultat för de från andra objekt tillgängliga (publika) operationerna. Andra saker såsom objekttypens namn, informell beskrivning etc. kan också förekomma.

I implementeringsdelen finns attributdefinitioner samt koden för de publika operationerna såväl som helt lokala (privata) operationer. Andra komponenter som tillhör ett objekt utgörs av diverse datastrukturer vilka används av det underliggande objekthanteringssystemet, och som ej är direkt synliga ens för objekten själva.

```

OBJEKTYP <namn>
SPECIFIKATION
  <publika operationer>
IMPLEMENTERING
  <attribut>
  <kod för publika op>
  <privata operationer>

```

Fig.1.5-1 : Objekttypsbeskrivning.

(1) Ett undantag från detta utgör vissa datastrukturer som enbart är förknippade med anropet av en operation, tex temporära variabler i implementeringen av en operation.

1.5.3. Instansiering: Beskrivning och Förekomst.

När ett objekt skapas, görs detta alltid enligt en beskrivning i någon objekttyp. Beskrivningen definierar objektets beteende och innehåll. Objekten som är skapade enligt beskrivningen utgör en klass av objekt med inbördes liknande egenskaper. Beteendet representeras av operationer och en eventuell satskropp, innehållet representeras av värden organiserade i attribut. En sådan klass av objekt sägs utgöra en bestämd Objekttyp (eller typ), medlemmarna i klassen kallas Objektinstanser (eller objekt) av typen. Det är till dessa instanser som vi kan härleda ett visst tillstånd, och alltid kan identifiera en viss instans såsom unik mha dess unika identifierare.

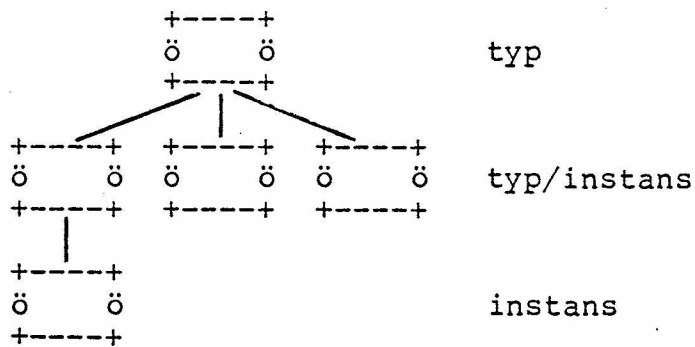


Fig.2: Objekttyper och objektinstanser.

Eftersom objekttyper också betraktas som objekt i systemet så måste även dessa vara skapade enligt någon beskrivning som definierar deras beteende och innehåll, dvs enligt någon objekttyp. Detta innebär att vi kan ha flera "nivåer" av instansiering i systemet (2), där ett typobjekt på en nivå har sina instansobjekt på en lägre nivå, vilka i sin tur kan vara typobjekt för instanser på nästa lägre nivå osv.

Om vi då har två objekt, varav det ena får vara typobjekt och det andra en instans av denna typ, ett instansobjekt, så innebär det att vi kan använda operationer i typen för att påverka värden i instansen lagrade enligt attributdefinitioner i typen. Av detta följer en viktig egenskap i objektmodellen, nämligen att alla objektoperationer alltid opererar på, eller inom ramen för, någon instans av den typ för vilken operationerna är definierade.

(2) Med instansiering menas att skapa en instans, dvs en förekomst.

1.5.4. Specialisering genom Egenskapsarv.

Ett sätt att skapa nya objekttyper är att definiera dem i termer av redan befintliga typer. En ny typ blir en utvidgning, eller specialisering, av en eller flera andra typer. Specialiseringen sker genom att nya egenskaper, i termer av operationer och attribut, tillförs. Ett sätt att åstadkomma detta är att låta en ny objekttyp "ärva" egenskaper från en eller flera typer, samtidigt som den själv introducerar nya. Detta brukar kallas för Egenskapsarv och kan användas som modelleringskoncept för att göra specialiseringar samtidigt som det är ett bekvämt sätt att strukturera beskrivningar.

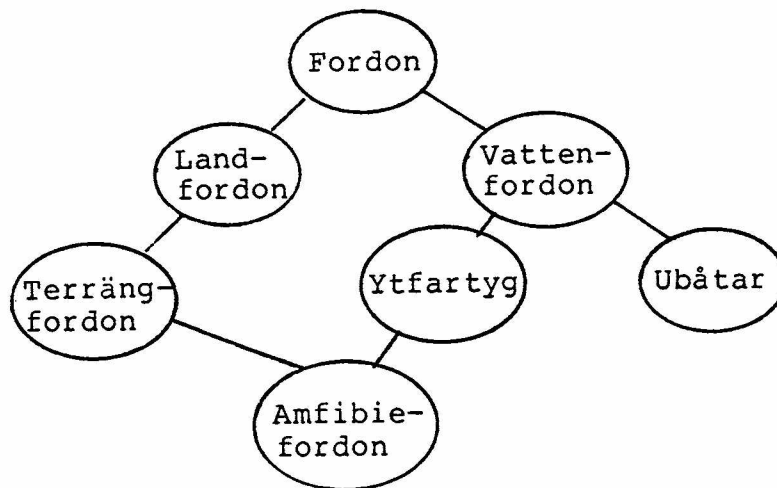


Fig.1.5-2: Specialisering av objekttyper.

Om vi ex vis har objekttyper som representerar olika typer av fordon (Fig.3), så kan den mest generella typen "Fordon" innehålla sådana egenskaper som gäller alla fordon oavsett om de är avsedda att framföras på torra land eller på Hårsfjärdens botten. Via successiva specialiseringar av "Fordon" kan nya typer tillföras. Vissa typer kan samtidigt ärva flera andra objekttyperns egenskaper, som "Amfibie"-typen i Fig.3.

Två objekttyper O1 och O2 har relationen Supertyp - Subtyp, om O2 ärver egenskaperna hos O1. Klassen av instanser av en subtyp utgör en delmängd av klassen av instanser av dess supertyp. Detta betyder att operationerna som är tillgängliga för instanser av t ex typen Landfordon (Fig.3) också är applicerbara på instanser av typen Terrängfordon, men ej omvänt. Generellt bildar arvsstrukturerna en riktad graf utan cykler, en objekttyp skall alltså inte kunna "ärva sig själv".

I ett objektorienterat system har man vanligen ett antal "systemobjekttyper" placerade överst i arvshierarkin och som då "automatiskt" används som supertyper då nya objekttyper introduceras. Dessa systemobjekttyper definierar egenskaper som man vill att varje objekt skall ha. Exakt vilka dessa är beror på hur systemet är konstruerat. Gränsen mellan vad som är "systemobjekttyper" och "applikationsobjekttyper" i ett objektorienterat system är dock mycket vag. Det är mycket sällan som en programmerare skapar "helt nya" objekttyper. Istället åter-

används tidigare definierade objekttyper i så stor utsträckning som möjligt. Systemtyperna kan i sig också vara specialiseringar av den mest generella typen i hela systemet (denna "urtyp" får betraktas som "inlödd" i systemet).

Olika principer kan tillämpas för själva arvsmechanismen. Teoretiskt kan man t ex tillåta att egenskaper kan ärvas selektivt från en typ till en annan, så att en subtyp enbart ärver en delmängd av supertypens egenskaper. Men även om denna flexibilitet i vissa fall kan anses befogad, så kan den också leda till missförstånd bland användare och programmerare. Det kan t ex resultera i att klassen instanser av en subtyp ej längre utgör en delmängd av klassen instanser av en supertyp. Det vanligaste torde därför vara att samtliga egenskaper ärvs från en supertyp till en subtyp. Här bör också nämnas att arvsmechanismen måste ta hänsyn till ev. namnkonflikter som kan uppstå då en subtyp introducerar namn på egenskaper som kommer i konflikt med namn på ärvda egenskaper.

1.5.5. View-objekt.

Analogt med View-definitioner i databaser, så kan också en objekttyp ha en eller flera View-objekt definierade mot sig. En View definieras mot operationerna som är tillgängliga i någon objekttyp. Vi kan kalla dessa för "Domänobjekt" till det aktuella View-objektet. (En View kan också definieras mot operationer som är tillgängliga i en annan View. Dvs, en objekttyp kan ha "en kedja" av Views definierade mot sig.)

En View innehåller en delmängd av operationerna från sitt domänobjekt, och kan på så sätt fungera som ett slags filter. Operationerna i View-objektet kan specificera transformationer av domänobjektets operationer. En View definieras alltså enbart i termer av operationer och kan inte introducera några ytterligare permanenta attribut. Ett View-objekt kan också specificera en instansbegränsning, så att endast en delmängd av domänobjektets klass av instanser kan nås via View-objektets operationer.

View-objekt är instanser av en View-objekttyp, men är ej själva några typobjekt, även om de i många avseenden liknar objekttyper. Flera, "alternativa" View-objekt kan definieras på ett visst domänobjekt, på basis av olika krav på åtkomst till operationer och mängden av instanser. (Notera skillnaden mellan detta och en "kedja av Views" enligt ovan.) View-objekt är i detta avseende mycket användbara för att realisera specifikation och kontroll av behörighet.

1.5.6. Versioner av Objekt.

Genom specialisering av objekttyper och definition av View-objekt kan objekttyperna i ett system stegvis byggas ut och anpassas i takt med att krav uppkommer. I vissa fall kan det också vara önskvärt att kunna förändra existerande objekttyper, istället för att införa en ny typ eller View. För detta ändamål kan vi införa versioner av objekt.

Då en objekttyp ändras skapar objekthanteringssystemet en ny version av objekttypen. I många fall innebär en förändring på typnivå att tidigare skapade objektinstanser blir inkompatibla med den ändrade typen. Den tidigare versionen kan därför hållas kvar i systemet så länge det existerar gamla instanser, så att dessa kan bearbetas även om en ny version av typen skapats. I de fall där gamla instanser efter hand blir inaktuella och tas bort, kan den tidigare typversionen till sist också tas bort. Detta innebär att nya instanser endast kan skapas enligt den senaste versionen av en objekttyp.

Man kan också tänka sig att systemet tillåter att instanser skapade enligt en gammal typversion, kan bearbetas via en ny/senare typversion, och omvänt. Detta kräver att transformationer kan specificeras mellan olika objekttypversioner (i vissa fall kan dock skillnaden mellan typversioner vara av sådan art att detta ej möjligt). I de fall där tillräckliga transformationsregler verkligen kan specificeras, kan introduktionen av en ny typversion göras transparent för det omgivande systemet. Tillsammans med Specialisering och View-definitio-ner blir versionshantering på typnivå en hjälp vid stegvis utbyggnad av ett system.

Versionshanteringen kan generaliseras till att omfatta alla objektinstanser i ett objektorienterat system, oavsett om de representerar objekttyper, instanser av objekttyper, View-objekt etc. Alla förändringar av ett objekt kan då tänkas resultera i att en ny version skapas. Ett system som aldrig "uppdaterar" sina objektinstanser (utan skapar en ny version av instansen vid varje förändring) kan sägas bevara alla historiska data. Av praktiska skäl måste givetvis den ständiga ökningen av antalet objektinstanser i systemet begränsas. I vissa fall (för vissa operationer) kanske det ej heller är önskvärt att upprätthålla mer än en version (den senaste).

Versionshantering kan också användas internt av objekthanteringssystemet, t ex för att möjliggöra återgång till äldre versioner av objekt efter en systemkrasch som lämnat vissa nya versioner i ett inkonsistent tillstånd.

1.6. Kontorsinformationssystemens krav på nya användargränssnitt

En hög grad av interaktivitet är ett av kännetecknen för KIS. Därmed kommer gränssnittet mot användaren att inta en central roll i systemet. Eftersom en stor mängd verktyg förekommer inom ett KIS ökar risken för att dessa sinsemellan inte är enhetliga. Stor omsorg måste därför läggas ned på att verkligen åstadkomma användarvänliga system. I detta avsnitt kommer vi att närmare studera utvecklingen inom området och olika sätt att utforma användargränssnittet. Här ingår också att beakta användningen av olika informationsformer och -media.

Här bör i klargörande syfte nämnas att man skiljer mellan informationens form (typ), dess representation samt informationsmedium. Ett informationsmedium är en informationsbärare såsom en terminal eller ett magnetband. Informationens representation avser hur denna representeras i form av data på ett eller flera informationsmedia. Informationens form, eller typ, kan antingen vara text, ljud, bild eller strukturerade data.

1.6.1. Nya interaktionsformer

Interaktionsformer och dialogstilar

Användardialogen hör till det viktigaste som finns i systemet. Utan ett bra gränssnitt gentemot användarna upplevs systemet som mindre bra, oavsett hur kraftfullt det än är under ytan. För att erhålla ett bra gränssnitt måste de system vi bygger anpassas till vårt sätt att tänka och arbeta. Detta perspektiv måste genomsyra alla delar av gränssnittet, från dialogstil och inmatningstekniker till perceptionsmodell, dvs den bild systemet presenterar av sig själv för användarna (mer om detta senare).

Under senare år har en rad nya former för interaktion mellan människa och dator växt fram. Anledningen är bl a att nya typer av enheter för in- och utmatning utvecklats. Tidigare dominerade det vanliga tangentbordet, eventuellt kompletterat med funktionstangenter, och en bildskärm för textpresentation. Idag finns dessutom en rad enheter såsom pekskärm, styrdosa, röstinmatning och -utmatning samt grafiska enheter, bara för att ta några exempel.

Oavsett vilka enheter som används för in-/utmatning finns ett antal grundläggande stilar för dialogen med systemet. Dessa brukar klassificeras enligt följande:

- formulär
- kommando
- meny
- fråga-svar

Samtliga stilar torde vara bekanta för läsaren och dessa kommer därför inte att behandlas närmare här. Vad som däremot är intressant är hur dessa dialogstilar förhåller sig till de nya inmatningsteknikerna.

| Inmatningsteknik: | Tangent- bord | Styrdosa/ pekskärm | Röst- inmatning |
|-------------------|------------------|-----------------------|--------------------|
| Dialogstil: | | | |
| - meny | X | X | X |
| - kommando | X | 1 | X |
| - formulär | X | 2 | 3 |
| - fråga-svar | X | 4 | 3 |

X-markerade kombinationer är direkt tillämpbara, under förutsättning att programvara finns tillgänglig förstås. Med siffror markerade kombinationer indikerar dock vissa restriktioner eller förutsättningar som måste vara uppfyllda.

1 - Kommandoteknik i kombination med styrdosa eller pekskärm är endast tillämpligt under förutsättning att möjliga kommandon visas på bildskärmen då val sker genom utpekning. Kommandoteknik närmar sig här menyteknik.

2 - Formulärteknik i kombination med styrdosa eller pekskärm fungerar i regel inte då antalet möjliga inmatningsalternativ har ett bredare spektrum än vad som kan presenteras på bildskärmen.

3 - Röstinmatning vid formulär- eller fråga-svar-teknik har endast begränsad användbarhet så länge som teknik för avancerad röstanalys och -tolkning inte finns tillgänglig.

4 - Fråga-svar-tekniker i kombination med styrdosa eller pekskärm fungerar heller inte såvida inte mängden svarsalternativ är begränsad och kan presenteras på bildskärmen för utpekning. Även här närmar vi oss menyteknik.

Inmatningsteknikerna och dialogstilarna är inte ömsesidigt uteslutande utan kan kombineras på olika sätt inom en och samma applikation.

Perceptionsmodell

Många har funderat över vilken modell eller bild av systemet som skall visas upp utåt mot slutanvändarna. Denna kallas perceptionsmodell med en term hämtad från psykologin. Vanligt förekommande synonymer till perceptionsmodell är metafor, paradigm eller konceptuell modell (den senare icke att förväxla med konceptuella modeller i konceptuell modellering).

Dagens system bygger oftast på en traditionell modell av ett datasystem med filer och poster relaterade i olika strukturer, kommandospråk samt program. Användaren måste här lära sig databehandlingssystemets modell och tillämpningarna anpassas till denna. Ur slutanvändarnas perspektiv är detta sällan acceptabelt. Nya och mer naturliga perceptionsmodeller har därför utvecklats.

Dokument- eller formulärbegreppet är basen för en av dessa modeller. Genom att bygga KIS:et kring ett begrepp som redan existerar har systemen kunnat förenklas avsevärt. I stället för att tala om poster talar man nu om formulär eller dokument. Ofta kompletteras modellen också av andra bekanta begrepp såsom arkivskåp, lådor och mappar i stället för filer och filkataloger. Användarna känner sig därför snabbt hemma i det nya systemet och mindre tid går åt för upplärning.

Det är emellertid inte självklart att man skall använda det manuella kontoret eller vårt nuvarande sätt att tänka som förebild vid utformningen av ett datasystem. Datatekniken erbjuder en rad nya möjligheter och delvis helt nya sätt att lösa gamla problem på. Genom att anpassa ett nytt system till en existerande modell är risken uppenbar att man inte utnyttjar de möjligheter som datatekniken erbjuder. Vad som är bra och dåligt i detta avseende torde endast erfarenheten kunna utvisa.

Genom att kombinera formulärkonceptet med grafiska bildskärmar har skrivbordsmetaforen skapats. Här har man helt enkelt låtit avbilda olika koncept såsom dokument, skåp, skrivare, papperskorgar etc. grafiskt på bildskärmen i form av små s k ikoner. Ikonerna kan bli flyttas, staplas på varandra för att ge illusionen av ett skrivbord. Det har visat sig att gränssnitt som dessa är mycket lätta att lära sig men att frekvent användning kan leda till att man snart tröttnar.

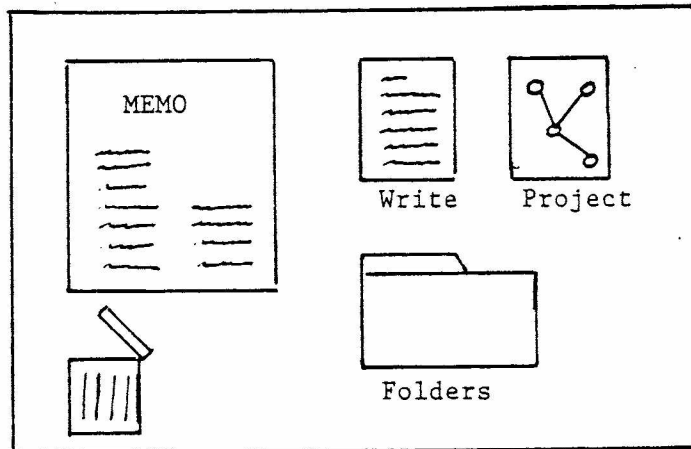


Fig.1.6-1. Lisa/Mac

En visuell representation av t ex pappershögar på skrivbordet har visat sig ha fördelen att den låter användarna tänka och minnas i termer av bilder. Detta gör det möjligt att komma ihåg ett dokument genom dess plats på "skrivbordet" eller i en hög. I traditionella system kan bara återsökning ske på namn eller innehåll.

Modellerna ovan utgör försök att göra systemen mer enhetliga på en övergripande nivå. Internt mellan olika applikationer förekommer dock fortfarande stora olikheter. Försök att tackla det senare problemet har gjorts i flera integrerade programpaket, typ Lotus 1-2-3. Här har uppsättningen kommandon i olika applikationer anpassats till en bestämd modell. För Lotus gäller att en modell av en matriskalkyl utgör basen. Detta innebär att alla applikationer på ytan beter sig som en matriskalkyl. Exempelvis behandlas poster och termer från databasen som rader och kolumner i en matriskalkyl. Andra liknande produkter kan exempelvis vara uppbyggda kring en texteditor.

Detta sätt att bygga upp en mängd verktyg kring en gemensam modell kallas beroende integrering därför att alla verktyg är avhängiga ett enda verktyg. Om enhetligheten är den största fördelen med denna integreringsstil så är den främsta nackdelen att många verktyg kan vara onaturliga att sammanfoga på detta sätt.

Även andra integreringsstilar förekommer såsom oberoende och synergistisk integrering.

Vid oberoende integrering passas inte verktygen in i en gemensam modell utan är helt fristående från varandra. Enhetligheten i användargränssnittet är helt avhängigt konstruktörens sinne för systemergonomi. På detta sätt blir användargränssnittet mer naturligt sett ur varje enskild applikations perspektiv. Risker är emellertid att onödiga skillnader mellan applikationer kan förekomma.

Synergistisk integrering slutligen, kan sägas kombinera fördelarna med de två tidigare nämnda integreringsstilarna utan att drabbas av deras nackdelar. Synergistisk integrering innebär en utökning av funktionaliteten genom en i stort sätt fri interaktion mellan olika delfunktioner. Användaren behöver kunskap endast om de funktioner som är relevanta för honom. Funktionerna kan användas fristående eller sammankopplat. Den så kallade "synergieffekten" skulle då bestå i att den totala funktionaliteten blir högre än summan av de individuella funktionernas. Vi kan alltså "göra mer" genom att utnyttja flera funktioner sammankopplat, än om vi skulle använda dem var och en för sig.

Försök har gjorts att karakterisera verktyg som av sina användare upplevs som lättlärd och trevliga att använda. Gemensamma faktorer för sådana system tycks vara att det användaren bearbetar är synligt på bildskärmen och att det kan påverkas med snabba, reversibla (undo) och inkrementella kommandon. Kommandona ges dock inte i form av klartext utan genom att föremål och kommandon pekats ut på ett eller annat sätt. Resultaten av utförda operationer skall vidare vara direkt synliga på bildskärmen. Det traditionella indirekta sättet att arbeta, dvs användarens kommando avtolkas och utförs i ett (i ett senare steg) av ett verktyg, verkar därför mindre attraktivt än ett mer direkt arbetssätt. Därur har termen "direkt bearbetning" uppstått. Andra benämningar på samma företeelse är WYSIWYG (What-You-See-Is-What-You-Get), visuell programmering och visuellt minne.

Exempel på verktyg som faller in under direkt bearbetning, förutom skrivbordsmetaforen, är bildskärmseditorer och matris-kalkylprogram. Genom att bearbetningen görs direkt på synliga föremål tenderar själva verktyget att försvinna. Verktyg som dessa ökar vanligen effektiviteten vid användningen och minskar antalet fel som hänger samman med kommandosyntaxen. En kritisk faktor är dock vad som presenteras på bildskärmen. Mycket information, oväsentlig eller rörigt presenterad sådan leder bara till förvirring.

1.6.2. Dialogmodellering

Dialoghantering är en gemensam funktion för alla interaktiva applikationer. Ett naturligt steg är därför att generalisera denna funktion och bryta ut den från varje enskilt program. Vi kan jämföra detta med utvecklingen på databasområdet. Där har generella databashanteringssystem utvecklats. Dessa baserar sig i sin tur på olika datamodeller, t ex relationsmodellen. En liknande utveckling är också önskvärd för dialoghantering. (Dialoghantering används här i vid bemärkelse för att täcka alla faciliteter som rör användargränssnittet.) En förutsättning för att detta skall lyckas är emellertid att bra dialogmodeller utvecklas för att beskriva och analysera dialoger.

Krav på användargränssnittet

I och med utvecklingen både på maskin- och programvarusidan tillkommer olika typer av perifera enheter, applikationer och användare. Dessa måste kunna samordnas inom ramen för ett och samma KIS. Krav måste därför ställas på användargränssnittet avseende bl a:

- enhetlighet,
- anpassningsbarhet (customization),
- enhetsoberoende ("device independence")
- representationsberoende.

Enhetlighet berör både kommandostruktur och perceptionsmodell. De kommandon som har ungefär samma semantiska betydelse i olika applikationer bör så långt det är möjligt också ha samma syntax. Detta betyder att exempelvis dokument skall skapas och utplånas på samma sätt som matriskalkyler, brev etc. Samma sak gäller perceptionsmodellen. Denna bör naturligtvis också vara gemensam för olika applikationer. Används skrivbordsmetaforen i en applikation bör även övriga applikationer, inklusive nyutvecklade sådana, begagna sig av denna. Kravet på enhetlighet gäller naturligtvis också dialogstil, inmatningsteknik, hjälp och felhantering etc.

Användargränssnittet skall inte bara vara enhetligt utan måste också kunna anpassas till användarnas önskemål. Användarnas datorvana kan vara större eller mindre varför systemet, beroende på användare, t ex måste ge mer eller mindre hjälp. Inom vissa ramar skall det också vara möjligt att välja interaktionsteknik, dialogstil, perceptionsmodell etc.

Även utåt mot terminaler och andra perifera enheter måste gränssnittet gå att anpassa. Detta kallar vi enhetsberoende (device independence). Systemets möjligheter att interagera och presentera data på olika sätt styrs i hög grad av de perifera enheternas egenskaper. Det är i regel inte acceptabelt att applikationer görs enhetsberoende på ett sätt som begränsar användningen av dessa till exempelvis en viss terminaltyp. Både in- och utmatning måste m a o, inom rimliga gränser, kunna anpassas till tillgänglig utrustning. Ett exempel på anpassning är grafiska arbetsstationer och vanliga texttermina-

ler. Använder man den senare typen av utrustning måste alla applikationer presentera data i form av tecken. I det förra fallet kan även grafiska presentationer komma i fråga.

Representationsoberoende är också önskvärt. Detta innebär att olika former av information - text, strukturerade data, ljud och bild - skall behandlas på ett enhetligt sätt av systemet. Skillnaderna mellan t ex en editor för text och en editor för ljud skall inte vara andra än sådana som föranleds av informationens form.

Dialogmodeller

Kraven på användargränssnittet ovan påverkar direkt dialogmodellens uppbyggnad. Allmänt gäller att modeller hjälper till att strukturera ett problemområde och att reducera komplexiteten hos problemen i fråga. Syftet med en dialogmodell, mer precist, är att fungera som en referensmodell på basis av vilken dialoghanteringssystem kan implementeras. Dialogmodellen ligger på en hög abstraktionsnivå och identifierar och organiserar funktionerna i användargränssnittet på ett sådant sätt att uppfyllandet av kraven ovan underlättas.

Funktioner i användargränssnittet och egenskaper hos dialogen som kan avbildas i dialogmodellen är t ex dialogstruktur, navigering i dialogen, behörighet, etc. Dialogstrukturen beskriver interaktioner mellan människa-dator och hur interaktionerna är kopplade till varandra. Navigering i dialogstrukturen handlar om hur man förflyttar sig mellan olika dialogavsnitt. Behörigheten avgör vem som får göra vad i dialogen. För att beskriva dessa funktioner behövs ett språk för dialogmodellering liknande databasvärldens datadefinitionsspråk (DDL) och datamanipuleringspråk (DML). Vi kan kalla språket för specificering av dialoger "dialogspecificeringspråk" (DSL - dialoger "dialogkontrollspråk" (DCL - Dialog Control Language)).

Med hjälp av DSL beskrivs strukturen hos dialogen i en applikation på en övergripande nivå. Språket kan användas som komplement till traditionella systemutvecklingsmodeller. Enskilda interaktioner eller hela dialogavsnitt och deras inbördes relationer kan här specificeras. Dessutom kan man ange behörighet, villkor som måste vara uppfyllda för att ett dialogavsnitt skall kunna påbörjas eller avslutas ("pre-conditions" resp. "post-conditions"), när vissa kontroller skall utföras etc.

DCL behövs för att navigera i dialogstrukturen som specificerats med hjälp av DSL. DCL kan uttrycka hur en vandring kan ske mellan olika dialogavsnitt, med de restriktioner som specificerade villkor anger. Initiativet för val av väg i strukturen kan ligga hos användaren eller dialoghanteraren. Hela tiden måste också operationer på objekt kunna initieras (Hantering av "objekt" behandlas i senare avsnitt). Kontrollstrukturen i en applikation kommer därför helt eller delvis att brytas ut ur programmen och läggas i dialogen i stället.

Logiskt sett har användaren en viss uppsättning funktioner till sitt förfogande, t ex GÅ TILL <dialognod> eller TA BORT <objekt>. Beroende på tillgänglig utrustning, vald dialogstil etc. måste initieringen av funktioner och presentation av data fysiskt kunna utföras på olika sätt. Exempelvis kan operationen TA BORT <objekt> ha åtminstone följande utseende: Under förutsättning att man använder en vanlig textterminal kan man skriva TA BORT omedelbart följt av namnet på objektet. Använder man en grafisk arbetsstation med styrdosa under skrivbordsmetaforen kan samma operation initieras genom att man pekar på ikonerna som representerar objektet som skall tas bort och därefter pekar på papperskorgen. Semantiken hos de båda handlingarna är dock densamma.

För att enkelt kunna hantera anpassningen till olika perifera enheter och användarkrav måste DCL delas upp i flera nivåer. På den lägsta nivån finns enhetsberoende operationer. På den översta nivån finns rent logiska operationer av den typ vi beskrivit ovan. Hur många nivåer som eventuellt behövs däremellan är föremål för aktuell forskning.

Genom att strukturera dialogen i ett lämpligt antal nivåer, från implementeringsberoende till implementeringsoberoende, erhålles en mer flexibel och kraftfull dialoghantering. Detta kommer att avsevärt underlätta utveckling av nya system. En enhetlig beskrivningsmodell som kan accepteras allmänt har dessutom fördelen att närmast identiska dialoghanteringssystem kan utvecklas för olika datorsystem. (jfr olika implementeringar av CODASYL-förslaget från DBTG).

1.6.3. Mot generella dialoghanteringssystem

Tidigare utformades och implementerades de interaktiva delarna av ett program separat för varje enskild applikation. Eftersom antalet rader kod för dialoghantering i en applikation mycket väl kan uppgå till över hälften av det totala antalet rader kod ledde detta till mycket dubbelarbete och slöseri med resurser. Genom att bygga upp procedurbibliotek eller använda sig av speciella I/O-paket, t ex formulärhanterare, kunde stora delar av dialogen nu återanvändas. Dessa förbättringar har dock än så länge endast i begränsad omfattning försökt tackla de problem vi beskrivit rörande bl a enhetsberoende och användaranpassning.

Som vi sett ovan påverkas dialogens utformning av främst tre faktorer:

- användarens preferenser,
- krav som applikationen ställer, och
- egenskaper hos de fysiska enheter som används.

För att lättare kunna jämföra samma krav som rör dessa faktorer vid konstruktion och implementering av dialoger bör användargränssnittet separeras helt och hållet från applikationerna. Därigenom blir det också möjligt att generalisera dialoghante-

ringen och göra den gemensam för alla program. Generaliseringen måste bygga på en nivåindeldad modell som tillåter att de krav vi listat uppfylls. Resultatet av detta är ett generellt dialoghanteringssystem (jfr databashanteringssystem). En framtida systemarkitektur kan komma att se ut som på figuren nedan.

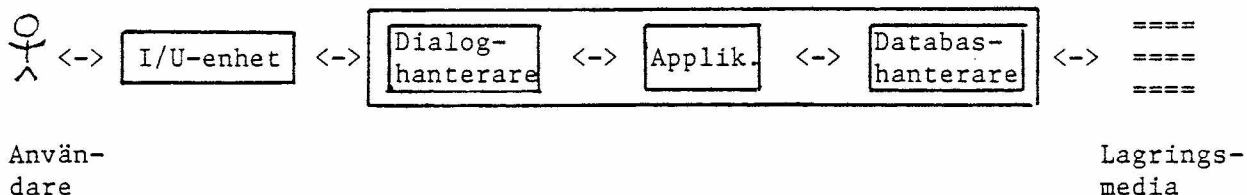


Fig.1.6-2: Framtida systemarkitektur

Dialoghanterarens uppgift är både att presentera information för användaren och ta emot, avtolka och utföra eller vidarebefordra användarens kommandon. Dessutom måste dialoghanteraren kontrollera att användaren följer de regler och ev. begränsningar som angivits i dialogspecifikationen.

Vad beträffar presentation av information måste dialoghanteraren avbilda utdata från en applikation till ett format och en layout som lämpar sig för presentation på användarens I/U-enhet, och som är anpassat efter användarens önskemål samt krav som applikationen ställer.

Vid inmatning konverteras kommandon och indata till ett internt format som kan avtolkas respektive lagras. Kommandon kan t ex avse både presentation av data (t ex byt sida, flytta fönster, använd formulär i stället för meny) och applikationen som sådan (exekvera, avbryt, hämta nästa dataelement).

Embryon till system som dessa existerar ännu bara på skisstadiet. Vid universitetet i Toronto, Canada, har ett forskningssystem kallat OFS (Office Form System) utvecklats. Formulär är ett centralt begrepp i systemet. Dessa är dock endast rena datastrukturer. Formulären kan betraktas på ett I/O-medium genom en mall (template). Mallen förser datastrukturen med en layout. Försök har gjorts att vidareutveckla systemet genom att generalisera det till att även klara av att hantera olika former av information. Så tillåts exempelvis att röstmeddelanden och bilder presenteras tillsammans med data. Begränsningar hos datamodellen som systemet bygger på samt representationsformen hos informationen gör dock att hanteringen av andra former av information än data och text blir mycket primitiv. Betydligt intressantare är då försöken att bygga nya modeller av användargränssnittet. Mer om detta i kapitlet om aktuell forskning.

1.7. Några forskningsproblem inom KIS-området

Vi skall nu vända blickarna mot några specifika forskningsproblem som under senare år tilldragit sig speciellt intresse inom KIS-området.

1.7.1. Olika former av information

Dagens informationssystem är huvudsakligen orienterade mot hantering av strukturerade data och text. Framtida system kommer emellertid att integrera ytterligare två former av information - ljud och bild. Forskning kring sådana system pågår med olika inriktning på många forskningslaboratorier världen runt. Trots att forskningen till stor del behandlar hantering av olika former av information och olika representationer kallas dessa system ofta felaktigt för multimediasystem.

Vid universitetet i Toronto, Canada, forskar en grupp kring dessa frågor. Målet är att utveckla system som kan hantera olika former av information på ett enhetligt sätt. Betoningen ligger på frågor rörande representation, presentation och kommunikation av information samt hur manipulering och utsökning av information skall ske.

Vid Bell Labs har ett väl avgränsat problem rörande editering av information tacklats. Resultatet kallas för en "virtuell editor". En virtuell editor har förmågan att hantera text och ljud på ett enhetligt sätt. Detta har åstadkommit genom att systemet håller reda både på hur informationen är representerad och dess struktur (text stycken och kapitel). Logiskt sett består editorn av ett antal olika editorer, en för varje form av information. Ur användarens perspektiv finns dock bara en gemensam uppsättning operationer tillgängliga, plus vissa formberoende operationer, vilka sedan översätts till, och utförs av, den verkliga editorn.

Hantering av olika former av information är inte bara aktuell inom ett datorsystem. Informationen måste också kunna kommuniceras. Experiment kring detta utförs bl a av televerket inför uppbyggnaden av ett digitalt kommunikationsnät, ISDN.

De ansatser som nämnts ovan befinner sig fortfarande på prototyp- eller idestadiet. Tillämpbarheten är också mycket begränsad jämfört med vad man önskar sig i ett kommersiellt användbart system.

1.7.2. Avancerade systemfunktioner

Ett KIS bör tillhandahålla funktioner för att hantera fel eller oförutsedda situationer, funktioner som automatiskt kan utlösa andra aktiviteter samt funktioner som underlättar stegvis utveckling av systemet. Alla dessa funktioner behöver inte vara direkt synliga för slutanvändarna. Många av dagens experimentsystem, men också flera existerande KIS, innehåller sådana funktioner.

KIS kännetecknas bl a av att system är höggradigt interaktiva och att transaktioner kan ta mycket lång tid. Transaktionen svarar här inte direkt mot begreppet databas - transaktion, men en likhet är att transaktion representerar en sekvens av aktiviteter, vilka alla (eller ingen) måste utföras för att informationen i databasen skall bevaras konsistent. Typiskt för KIS är att sådana aktiviteter kan sträcka sig över många dialogsessioner och ta dagar i anspråk snarare än minuter. Fel som inträffar under utförandet av en transaktion eller under dialog med användaren får inte orsaka att hela systemet hänger sig eller kraschar. Systemet måste därför tillhandahålla en funktion för sk hantering av undantag ("exception handling"). Denna funktion skall upptäcka och analysera fel och helst omärkligt för användaren ställa allt till rätta igen. Om detta ej går skall så mycket som möjligt sparas undan av det aktuella tillståndet för att underlätta återstart och spara arbete.

I kontorsmiljön upprepas många aktiviteter regelbundet eller utlöses av vissa villkor. Många ekonomiska rutiner utförs t ex på daglig basis, andra rutiner utlöses om t ex en stor utbetalning försöker göras. Mekanismen som hanterar sådana funktioner i ett KIS brukar kallas för "trigger" (avtryckare eller utlösningmekanism). En trigger består av ett vilkor som talar om när trigger-mekanismen skall aktiveras. Villkoret kan antingen vara en tidpunkt eller ett godtyckligt villkor. Kopplat till triggern finns dessutom åtgärder specificerade vilka skall vidtagas antingen villkorslöst eller under förutsättning att ytterligare angivna villkor är uppfyllda. Triggers används i första hand i situationer som inte så enkelt låter sig hanteras i traditionella programspråk eller databashanteringssystem.

Förändringar av databasens innehåll sker mycket frekvent. Ibland är det av intresse att inte bara kunna tillhandahålla den mest aktuella informationen utan också något äldre uppgifter. Enkla system för att hålla fler än den senaste versionen av databasens innehåll tillgängliga finns redan. Även applikationer, dokumenttyper etc. förändras dock över tiden. För detta saknas det idag kraftfullt stöd. De system som finns klarar bara av att hålla bibliotek av versioner av applikationer tillgängliga, t ex SCCS under UNIX. Vad som behövs är emellertid ett aktivt stöd från systemets sida som dels hjälper en systemkonstruktör att utveckla nya versioner av program och dokument, dels gör förändringarna transparenta gentemot slutanvändarna, om möjligt och där så är möjligt. Detta betyder att i framtida KIS kan flera typ- och instansversioner komma att finnas tillgängliga samtidigt.

1.7.3. Användargränssnittet

Forskning kring användargränssnittet pågår parallellt på många olika fronter. Dels finns mer psykologiskt orienterad forskning som försöker besvara frågan: Hur skall ett bra användargränssnitt se ut? Annan forskning är mer teoretiskt eller tekniskt orienterad och fokuserar generella modeller av användargränssnitt eller implementering av avancerade arbetsstationer.

På den psykologiska sidan har XEROX tidigare lagt ned en hel del arbete vid utvecklingen av Star-systemet. Problem som tacklats är bl a hur folk arbetar i en kontorsmiljö och hur ett datasystem kan efterlikna detta arbetssätt för att åstadkomma lättlärd och lättanvända system. En mängd annan forskning av mer aktuellt datum pågår med aktivt deltagande av psykologer. Denna typ av forskning ligger emellertid utanför ämnesområdet för denna rapport. Här fokuseras i stället embryon till tekniska lösningar på aktuella problem

Vid INRIA, ett franskt forskningsinstitut, har under ett antal år ett projekt kallat KAYAK pågått. Inriktningen har allmänt kretsat kring KIS. Inom projektet har också forskning kring avancerade arbetsstationer drivits. Särskilt intresse har här ägnats åt hur olika applikationer skall kunna samsas om ett och samma gränssnitt, trots olikheter under ytan. Ett grafiskt gränssnitt som tillhandahåller generiska funktioner gemensamma för de flesta applikationer såsom ta bort, skapa, sök, kopiera prövas som en lösning på problemet.

För att kunna presentera information på olika typer av enheter måste programvaran som hanterar I/O generaliseras. Enklare exempel på detta är bildskärms hanterare som i stället för att arbeta med specifikationer i form av absoluta adresser på bildskärmen för det som skall presenteras, t ex rad 4 kolumn 12, arbetar med relativa angivelser av typen: överst och centrerat.

Förutom försöken att göra programvaran enhetsoberoende görs också ansträngningar att separera applikationer och data helt från aspekter rörande hur dessa skall presenteras. "Template"-begreppet är ett sådant försök. En template är en mall som talar om hur något skall presenteras, inklusive layout, format och ledtexter. Enklast ses en template som ett ej ifyllt formulär med ledtexter och tomma fält. Användarna eller olika program kan senare fylla i tomma fält med värden.

Av den just nu pågående forskningen hör konstruktionen av formella modeller av komponenterna i användargränssnittet till den mest intressanta. Ett exempel på denna typ av forskning är en referensmodell för kommandospråk till operativsystem (IFIP). Hela användargränssnittet modelleras i form av objekt som vart och ett har sin bestämda funktion. Målet är att förbättra operativsystemens kommandospråk och att göra dem mer enhetliga. Samma princip är emellertid också tillämpbar på användargränssnittet i stort.

1.8. Litteraturtips

En hel del litteratur, från böcker till enskilda artiklar, har publicerats inom området under senare år. Nedan recenseras och kommenteras ett urval av dessa.

1.8.1. Böcker

Bjurling, Lööf - Datoriserad kontorsproduktion, Liber Förlag, Malmö, 1982.

Översikt: Författarna anlägger ett evolutionärt perspektiv där KIS ses som en naturlig påbyggnad på den databehandling som redan existerar i de flesta organisationer. En mängd aspekter på datorisering av kontor presenteras plus grundläggande data-teknik, datakommunikation och televerkets tjänster. Dessutom behandlas synpunkter från leverantörer, politiska partier och andra organisationer.

Omdöme: Boken är ganska allmänt hållen och tillför inte så mycket nytt för den allmänbildade.

Uhlig, Fahrber, Bair - The Office of the Future, North-Holland, 1982.

Översikt: Boken ger en mycket bred översikt av området kontorsinformationssystem. Författarna är ansvariga för var sin del i vilken de ger sin syn på området ur olika infallsvinklar. Tre olika områden behandlas - (1) användningen av datorer på kontor, (2) teknologiska imperativ och (3) av datorstöd på kontor till olika typer av distribuerade system. Del 3 slutligen tar upp systemens effekter ur organisationens, gruppens och individens perspektiv. Dessutom diskuteras något om införande av kontorsinformationssystem och lönsamhetsaspekter.

Omdöme: Ytliga betraktelser varvas med detaljerade redogörelser för enskilda problem. Boken utgör ett inte helt lyckat försök att täcka in allt inom området. Detta leder i sin tur till att djupet blir lidande, bortsett från några enstaka riktiga djupdykningar ned till bl a VLSI och bubbelminnen. Ändå innehåller denna bok en av de få översikterna av området. Om boken läses med urskiljning utgör den en god investering.

D.N. Chorafas - Office Automation: The Productivity Challenge Prentice-Hall, 1982

Översikt: Författaren presenterar mot bakgrund av den låga produktivitetsutvecklingen på kontoren sin syn på vad kontorsinformationssystemen kommer att betyda. Presentationen inleds med en genomgång av olika verktyg som stöder kontorsarbete, från ord- och textbehandling till elektronisk post. Därefter sätts verktygen in i sitt sammanhang i ett KIS. Vidare görs en

detaljerad genomgång av olika hårdvarukomponenter såsom skrivare och lagringsmedia. Slutligen behandlas hur man förbereder sig för de nya systemen kompletterat med ett exempel från bankmiljö.

Omdöme: Boken är lättillgänglig men känns ibland lite för elementär och okritisk. Den innehåller dock ett av de få försöken att motivera behovet av KIS och olika verktyg.

D. Tschritzis (ed.) - Office Automation, Concepts and Tools Springer Verlag, 1985.

Översikt: Boken presenterar en rad forskningsproblem inom området. Huvudtemat är "tillämpning av koncept och tekniker från databas- och AI-området för implementering av verktyg för kontorsautomation (vår översättning). Problem som behandlas är utformning av användargränssnitt, hantering av olika typer av information (data, text, röst, bild), avancerade system för elektronisk post, specificering av ärenden (office procedures), konceptuell modellering och KIS mm. Boken är egentligen en sammanställning av artiklar rörande aktuell forskning vid universitetet i Toronto, Canada, för ett av de ledande forskningscentrumen inom området.

Omdöme: Artiklarna i boken ligger på olika nivå vad gäller tillgänglighet. De flesta bör dock kunna läsas med god behållning även av icke-forskare. För den som är intresserad av specifika forskningsproblem och inte bara den kommersiella bilden av KIS kan boken vara ett intressant tillskott.

N. Naffah (ed.) - Office Information Systems, North-Holland, 1982.

Översikt: En bred men något äldre sammanställning av forskningsartiklar återfinns i denna bok som är resultatet av en internationell konferens i Frankrike redan år 1981. Artiklarna behandlar allt från språk för implementering av KIS till metoder för studier av kontorsarbete.

Omdöme: Kvaliteten på artiklarna varierar kraftigt. Dessutom har mycket hänt sedan boken publicerades. Vissa artiklar är dock fortfarande aktuella.

S-K Chang (ed.) - Management and Office Information Systems, Plenum Press, New York, 1984.

Översikt: Boken är en aktuell sammanställning av artiklar från en konferens. Betoningen ligger på KIS ur ett "management"-perspektiv. Beslutstödssystem och databashantering i organisationen utgör centrala inslag. Även mer tekniska aspekter på KIS behandlas.

Omdöme: Artiklarna är generellt sett av hög kvalitet. Nivån vad beträffar förkunskaper som krävs av läsaren varierar dock kraftigt. Alla kan förmodligen hitta åtminstone någon intressant artikel.

Adriansson m fl - En väg mot möjligheternas kontor, Riksdataförbundet (RDF), 1982.

Översikt: Boken bygger delvis på en empirisk studie rörande tillämpningar av kontorsteknik som gjorts inom ramen för ett projekt i RDF:s regi. Målet är att förmedla en "helhetssyn" på den pågående förändringen av kontorsarbetet baserad på tidigare erfarenhet. Dessutom ges råd beträffande införande och användning av den nya tekniken.

Omdöme: Ger ett organisatoriskt snarare än ett tekniskt perspektiv på KIS. Innehållet känns ibland lite väl allmänt och delvis föråldrat. Detta hindrar dock inte att boken är värd en genomläsning för sin speciella infallsvinkel på KIS.

R A Hirschheim - Office Automation: Concepts, Technologies and Issues Addison-Wesley 1985

Översikt: Författaren säger att boken är avsedd för en akademisk publik, men att den har tillräcklig täckning av praktiska problemställningar för att vara av intresse för praktikfältet. För att rikta sig till en sådan publik får man säga att boken kräver ovanligt lite av grundläggande kunskaper i informationsteknologi och databehandling.

Boken behandlar tre huvudteman:

- Orsaker till intresset för kontorsautomation, återblick på den tekniska utvecklingen hittills, belysning av begreppet kontor.
- Den grundläggande teknologin (i vid mening) som speglar nuvarande "state-of-the-art". Detta tema behandlas på ett strukturerat sätt genom att separera beskrivningen av teknologier för inmatning, utmatning, behandling resp. distribution av information samt arkitekturer (teknisk struktur) för kontorsinformationssystem i relation till olika ansatser (här: ungefär "strategier").
- Organisatoriska aspekter på planering av kontorsautomation samt sociotekniska perspektiv på införandet av kontorsinformationssystem

Något närmare omdöme av denna helt färska bok som just kommit i våra händer vill vi inte göra eftersom vi inte haft tillfälle att studera den grundligt än.

1.8.2. Andra publikationer

Förutom böckerna ovan publiceras regelbundet både rapporter och tidskrifter inom området.

Statskontoret har t ex publicerat en rad rapporter på temat kontorsinformationssystem, bl a följande:

- Kontorsautomation i statsförvaltningen (1981:12)
- Datorer på kontor (1983:30)
- Kontorsarbete i förändring (1983:40)
- Inför ett integrerat kontor (1984:33)
- Bättre datorstöd i kontorsarbetet - väsentliga områden för vidareutveckling (1985: Preliminär titel, publiceras hösten 1985)

Innehållet behandlar ofta KIS ur den offentliga förvaltningens perspektiv. Många punkter är dock av ett mer allmänt intresse. De två senare rapporterna - "Inför ett integrerat kontor ?!" och "Bättre datorstöd i kontorsarbetet - väsentliga områden för vidareutveckling" - ställer krav på dagens KIS respektive behandlar KIS med sikte på framtiden.

Teldok är namnet på en rapportserie som ges ut av televerket. Rapporterna behandlar allt från videokonferenser och bredbandskommunikation till tillämpningar av och exempel på kontorsinformationssystem i olika länder. Nedan är några exempel på aktuella rapporttitlar listade:

- Datakommunikation via persondatorer för småföretag i USA
- Grannskap 90 - distansarbete i Nykvarn
- Trender inom kontorsautomation i USA
- Telebild 2 - fortsättning av videotexverksamheten
- Office Automation in Europe - A Follow-Up

Av naturliga skäl dominerar kommunikationsaspekter bland titlarna. Rapporterna är i regel mycket allmänt hållna och kan därför läsas av vem som helst, även av personer med mycket ringa kunskaper inom området. Tyvärr betyder detta också att de inte alltid tillför så mycket nytt till den som är verksam inom fältet. De är dock läsvärda för den som önskar en fortlöpande orientering och erhålles dessutom gratis.

Beträffande rena forskningsartiklar tillåter utrymmet här inte en recension av enskilda artiklar i den rika flora som finns. Däremot vill vi peka på de viktigaste av de regelbundet utgivna publikationerna - och rekommendera läsning av dessa till dem, som vill gå djupare in i ämnet.

Transactions on Office Information Systems (TOOIS) är en tidskrift som ges ut av ACM kvartalsvis. (ACM, Association for Computing Machinery, är en USA-baserad internationell sammanlutning av forskare.) Artiklarnas innehåll spänner över allt från aktuella forskningsproblem till erfarenheter av användningen av KIS.

ACM håller också en årlig konferens på temat kontorsinformationssystem. Konferensen går under beteckningen ACM-SIGOA (Special Interest Group on Office Automation). Papper som presenterats vid konferensen ställs samman i en konferensdokumentation ("proceeding"). Särskilt intressanta artiklar brukar publiceras i TOOIS (se ovan).

Eftersom KIS spänner över så många områden brukar artiklar då och då dyka upp även i andra ACM-tidskrifter och andra organisationers tidskrifter (ex vis IEEE Computer Society, som ger ut "IEEE Database Engineering Bulletin" och "Computer").

Dokumentation från internationella databaskonferenser innehåller numera ofta en hel del material om kontorsinformationssystem. Speciellt bör då nämnas "The international conference on Very Large Data Bases".

2. IRM PÅ SE-BANKEN - MÅL OCH UPPFYLLELSE

INNEHÅLL

IRM PÅ SKANDINAVISKA ENSKILDA BANKEN - MÅL OCH UPPFYLLELSE

Mål som satts upp för "IRM"
Tillvaratagande av befintliga operativa data
Graden av självständighet
Tillgängligt överallt
Val av arbetsstation

Nuläge
Användning av INFO-Fråga
Användning av INFO-Dialog

Hur vi uppnådde våra mål
Tidiga försök
Byggandet av INFO-Fråga
Utveckling av INFO-Dialog

Till slut

FÖRORD

S-E-Banken har sedan 1979 följt en strategi som går ut på att efterfrågade data skall vara lätt tillgängliga för dem som behöver dem, oberoende av var vederbörande sitter, vilken utrustning som användes och vilken kompetens som personen har. Begränsning av dataåtkomst skall ske genom behörighetssystemet och enligt uppgjorda prioriteringar för relationen åtkomsttid/kostnad.

Uppfyllandet av policyn har innefattat utveckling av verktygen INFO-Fråga för direkt penetration och analys av stora centrala register och INFO-Dialog för vidare analyser av extraherade data eller personliga register. Kopplingar har gjorts till alla datamiljöer, t ex PC och PC-baserade verktyg.

Ett tusental användare i Banken har utbildats i de nya miljöerna, bl a har alla lokalkontor tillgång till dem.

IRM PÅ SKANDINAVISKA ENSKILDA BANKEN - MÅL OCH UPPFYLLELSE

MÅL SOM SATTS UPP FÖR "IRM"

Nedan följer en beskrivning av de mål som man satte upp 1969.

TILLVARATAGANDE AV BEFINTLIGA OPERATIVA DATA

SEB har en jämförelsevis lång tradition på dataområdet och man har bl a byggt upp ett antal omfattande kundregister på olika områden. Ett nära till hands liggande mål har varit att ställa A-L-L-A dessa i grunden operativa data till förfogande för bankens handläggare på olika områden.

Alla handläggare, t ex en kontorschef, skulle direkt kunna sammanställa data om t ex en viss kundgrupp som underlag för ett givet affärsbeslut. Centrala instanser skulle kunna jobba på samma sätt för bl a den mera strategiska planering man sysslar med där.

GRADEN AV SJÄLVSTÄNDIGHET

För att handläggarna själva skall kunna fatta korrekta beslut i en viss affärssituation krävs också att handläggarna utan speciellt stöd från annan instans, t ex dataavdelningen, kan klara av att slå upp, genomleta och sammanställa de data han eller hon önskar.

De använda verktygen måste vara mycket enkla och helt befriade från datatekniska oväsentligheter. Det måste således vara möjligt att:

1. Se vilka data som finns tillgängliga

Detta gäller på alla nivåer från bandarkiv till betydelsen av enskilda värden i ett fält. Allt måste vara utförligt verbalt förklarat och aktuellt.

2. Veta hur man skall göra

Användaren kan inte förväntas vara van vid verktygen, utan minsta kommando måste dokumenteras noggrant åtkomligt både via terminalen och skriven dokumentation.

3. Enkelhet

Syntaxen måste vara naturlig med svenska kommandon utan förkortningar eller ovidkommande skiljetecken. En användare som bara vill utföra vissa begränsade funktioner i ett system måste kunna arbeta på ett enkelt sätt oförhindrad av alla de andra möjligheter som erbjudes.

Förkortningar skall vara möjliga ned till ett tecken, men ingen förkortning skall vara nödvändig.

TILLGÄNGLIGT ÖVERALLT

Användaren skall kunna efterfråga tjänster från sin vanliga arbetsstation och i princip när som helst, även utanför kontorstid. Det betyder att datorsystemet måste ha faciliteter för att prioritera bearbetningar efter viktighet och beräkningstyg.

VAL AV ARBETSSTATION

De bearbetningar som användaren önskar utföra skall kunna styras till lämpligaste plats med hänsyn till t ex beräkningskapacitet, tillgängliga verktyg, svarstider etc.

Vidare skall användaren kunna kombinera och/eller lagra data i de register som han eller hon önskar utnyttja.

NULÄGE

När man studerar nuläget kan man med viss tillfredsställelse konstatera att de uppställda målen till hög grad har uppfyllts. Visst kan mycket göras för att få ett ännu bättre användargränssnitt; idealet vore förstås att använda avancerad artificiell intelligens med naturligt språk som bas för all dialog med systemet. I alla fall har vi uppnått ett gränssnitt som accepterats av tusentalet banktjänstemän som inte har någon speciell databakgrund.

ANVÄNDNING AV INFO-FRÅGA

Produkten INFO-Fråga används för frågor mot stora och små register, på såväl magnetband som skivminne. INFO-Fråga användes inom de flesta av våra systemområden, men för varierande saker. Nedan följer några exempel:

* Marknadsföring

Kontoren ställer frågor mot flera kunddatabaser. En av dem omfattar mer än 2 miljarder tecken. På så sätt kan uppgifter snabbt tas fram om utveckling och status på olika kundgrupper och tjänster, med avseende på lönsamhet, spridning och omfattning.

För kontoren är det viktigt att kunna identifiera undantagen i denna enorma datamassa: Vilka produkter är våra stöttepelare, vilka är förluskällor. Vilka företag är på väg in i en risksituation etc. Det viktiga är att handläggare kan ställa avancerade och NYA frågor som är anpassade till det problem som gäller vid ett visst tillfälle.

* Resultatsammanställningar

Med INFO-Fråga göres sammanställningar utifrån de variabler som analytikern vill titta på. Man sammanställer ofta en ackumulerad rapport över totalmaterialet eller en mindre del.

* Diverse ad hoc rapportering

Inom utlandsrörelsen tar man fram rapporter för bl a länderbevakning, Loro och Nostro avstämning, internrapportering och värdepappershantering.

* Filöverföring till INFO-Dialog eller PC

INFO-Fråga användes inom de nämnda områdena och för alla slags analyser för att hämta data. Data specificeras på samma sätt som när man vill ställa en fråga, men genom att använda ett annat kommando formateras data som en fil istället. Data är normalt selekterade och eventuellt ackumulerade när de överförs. De kan nu användas till vidare analys i andra verktyg.

ANVÄNDNING AV INFO-DIALOG

Produkten INFO-Dialog användes för personliga register eller för data som hämtats från annat system t ex via INFO-Fråga. INFO-Dialog har avancerade funktioner för dataredigering, analys, beräkningar och presentation.

Den här typen av produkter användes mycket av slutanvändare som utvecklar egna system eftersom det är lätt att skapa egna register och begära fram t ex rapporter ur dem.

INFO-Dialog användes t ex inom:

* Personal och revision

För att hålla reda på grupper av individer, projekt, utbildning etc. För beläggningsplanering o dyl.

* På lokalkontor och centrala avdelningar

För reskontraföring av värdepapper, upplagda lån, säkerheter o dyl. I sammanhanget göres ofta bevakning av t ex lånestock, upplupen ränta, återstående utrymme mm. Man framställer många gånger rapporter sorterade i angelägenhetsordning. En del gör även avisering till berörda kontor, lånegivare och liknande.

* på utlandsavdelningarna

Här användes INFO-Dialog bl a för att överblicka de viktiga variabler som har så stor betydelse för utlandsrörelsen.

Således beräknar man valutaindex och presenterar gärna resultaten grafiskt. Grafisk presentation är givetvis den vanliga formen att redovisa trender, det må gälla råvarupriser, BNP, eller något annat.

Vidare använder man INFO-Dialog inom områden som länderrapportering, företagsbevakning och förmedlingstjänster etc.

* I övrigt

Användes INFO-Dialog inom så gott som alla områden på S-E-Banken. Det kan gälla beräkning av obligationsstockar, distribution av värdepapper, beräkning av kommunikationsnät, utvärdering av offerter, beräkning av avgifter, etc.

HUR VI UPPNÅDDE VÅRA MÅL

TIDIGA FÖRSÖK

Våra första försök på IRM området bestod i att kunniga tekniker och tillämpningsexperter satte samman databaser med väsentliga kunddata. Det visade sig bli många magnetband.

Eftersom de produkter som vi först stötte på var IBM:s APLDI och GIS så prövades dessa produkter. Begåvade och intresserade slutanvändare fick tillfälle att prova produkterna och att använda resultaten.

Vår första erfarenhet var: Bara siffror skapar inte tillräckligt intresse. Avnämaren måste ha förståelse för hur siffrorna tagits fram och framför allt ha förtroende för materialet. Användare som är vana att få all information via standardrapporter är inte heller alltid så kritiska som de borde vara; det finns en klar risk att man använder en siffra som i och för sig är korrekt, men i fel sammanhang.

Slutligen visade det sig att våra verktyg i stort sett var alltför komplicerade för beslutsfattare. De vill ha ganska komplexa frågor besvarade, men accepterar bara mycket enkla redskap för att samtala med datorn.

BYGGANDET AV INFO-FRÅGA

INFO-Fråga byggdes för att slutanvändarna lätt skulle kunna ställa sina frågor mot stora och komplicerade register utan att ha datautbildning.

Version 1 uppfyllde många av kraven. Enkla kommandon på vanlig svenska, frågemöjligheter på fältnamn, etc. Via INFO-Fråga startade användarna satsvisa jobb i datorn som innehöll utdata från en i INFO-Fråga inbyggd programgenerator.

Proven utföll bra, men det visade sig att användarna behövde mer stöd än vad som fanns i version 1 för att uppnå önskade resultat. I version 2 fanns bibliotek med "färdiga frågor" som direkt kan anropas av användaren. Vidare skapades bryggor varigenom dictionaryinformation direkt var tillgänglig för användaren. Vidare gjordes omfattande hjälpsystem och utförliga svarsmeddelanden på svenska. Systemet blev nästan självinstruerande. Slutligen skapades ett omfattande behörighetssystem som gjorde det möjligt för många användare att nå de data de har behörighet till men inte andra.

I version 3 fick användarna tillgång till alla bankens IMS-databaser och hade därmed tillgång till praktiskt taget alla lagrade data i banken. Vidare skapades bryggor till INFO-Dialog och PC-baserade programvaror (Lotus t ex).

Nu hade vi ett verktyg där användare HELT utan assistans från dataavdelningen och utan speciella förkunskaper med enkla, korta svenska kommandon kunde sammanställa en godtycklig delmängd av vilket register som helst i banken (som vederbörande hade behörighet till) samt ta ut resultatet på skärmen, på papper eller som en fil. Filen kan sedan antingen uppdateras med fullskärmsteknik och/eller joinas med andra flata filer som användaren lagrat själv eller hämtat från andra register.

Tack vare det avancerade gränssnittet till vårt data dictionary kan användaren noggrant penetrera innehållet i olika delar av register och databaser.

UTVECKLING AV INFO-DIALOG

INFO-Dialog skapades som ett skissverktyg som byggde på relationsdatabasteknik. Omfattande indatakontroller och inmatning/redigering i fullskärmsteknik var viktiga bitar.

Med INFO-Dialog var det möjligt redan 1981 att göra prototypsystem genom att extrahera data ur befintliga databaser, modifiera och joina dem samt skapa önskade utdata på mycket kort tid. En koppling bild-fil gjorde att alla filer hade en defaultbild som kunde modifieras med formulärhanteraren INFO-Bild.

Senare har systemet byggts ut med avancerad intelligent grafik, där bilden inte bara genereras utan ingående delar beräknas ungefär som i en rapportgenerator. Den som kan skriva en rapport kan därmed också göra en bild (i bägge fallen ett enradskommando i INFO-Dialog).

TILL SLUT

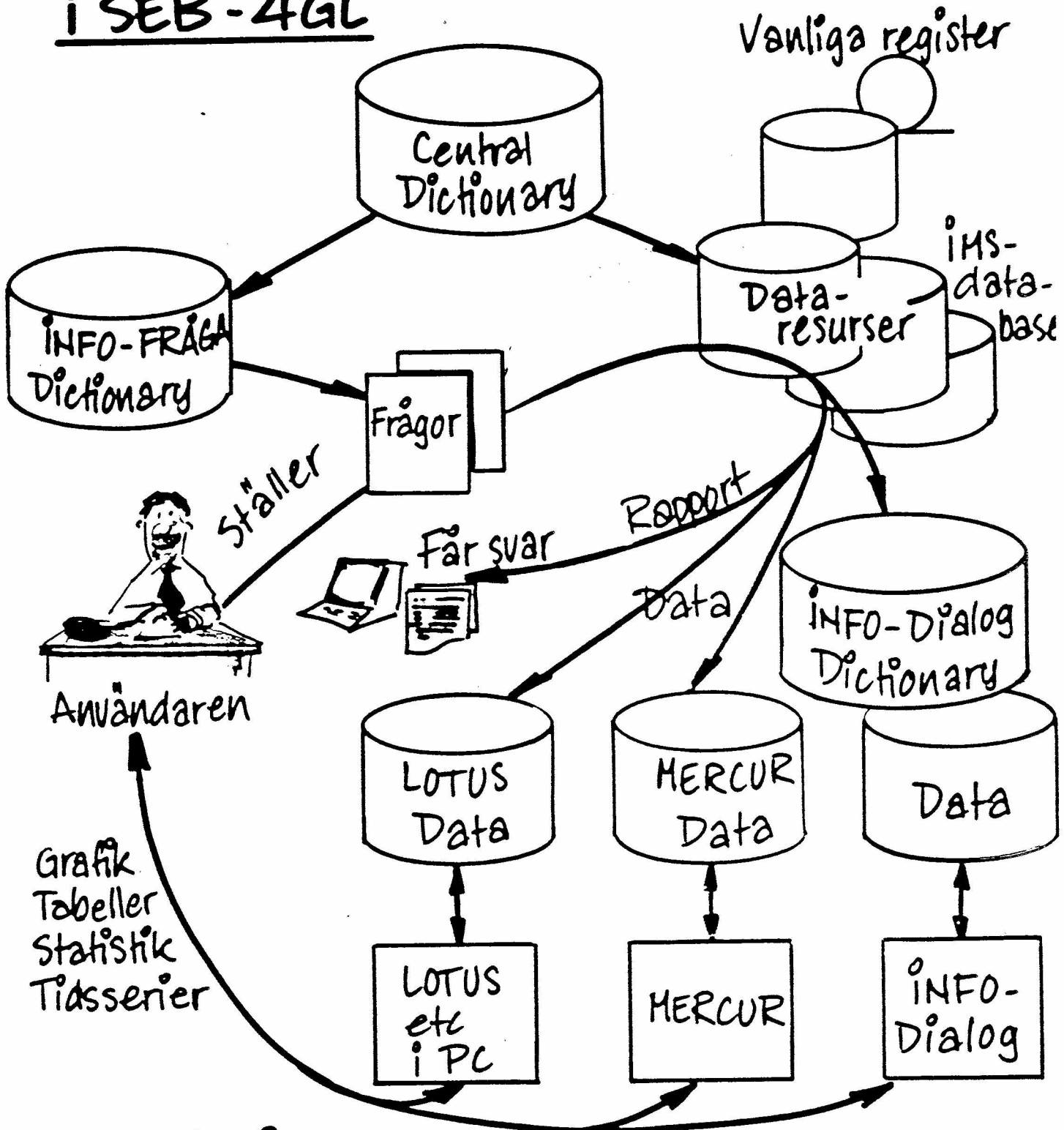
Vi har genomfört en långt gången standardisering på databasidan som har möjliggjort standardiserad åtkomst till data.

En handläggare (icke dataspecialist) kan i dag när som helst sammanställa de data han eller hon önskar ur godtyckligt register. Det kan ske både för små och mycket stora register, även på magnetband osv. Dessa data kan direkt blandas med data från andra register och sedan vidare analyseras, sammanställas och presenteras i det verktyg man önskar använda. En fullständig integration med vår APL-miljö gör att dessa data direkt också kan användas av ett nyskrivet APL-program eller av något av våra många befintliga APL-system.

Även systemavdelningen uppskattar lättheten att använda verktygen och att de lätt kan inlemmas i ett befintligt system. Verktygen är i dag inlemmade i många av våra traditionellt utvecklade system. Ofta användes den beskrivna miljön istället för uppsättningar av färdiga datalistor eller bilder. Speciellt som INFO-Fråga direkt stöder alla programmeringsspråk som används i banken (APL, COBOL och Assembler) och erbjuder en säker miljö för våra användare att utföra diverse jobb i datorn.

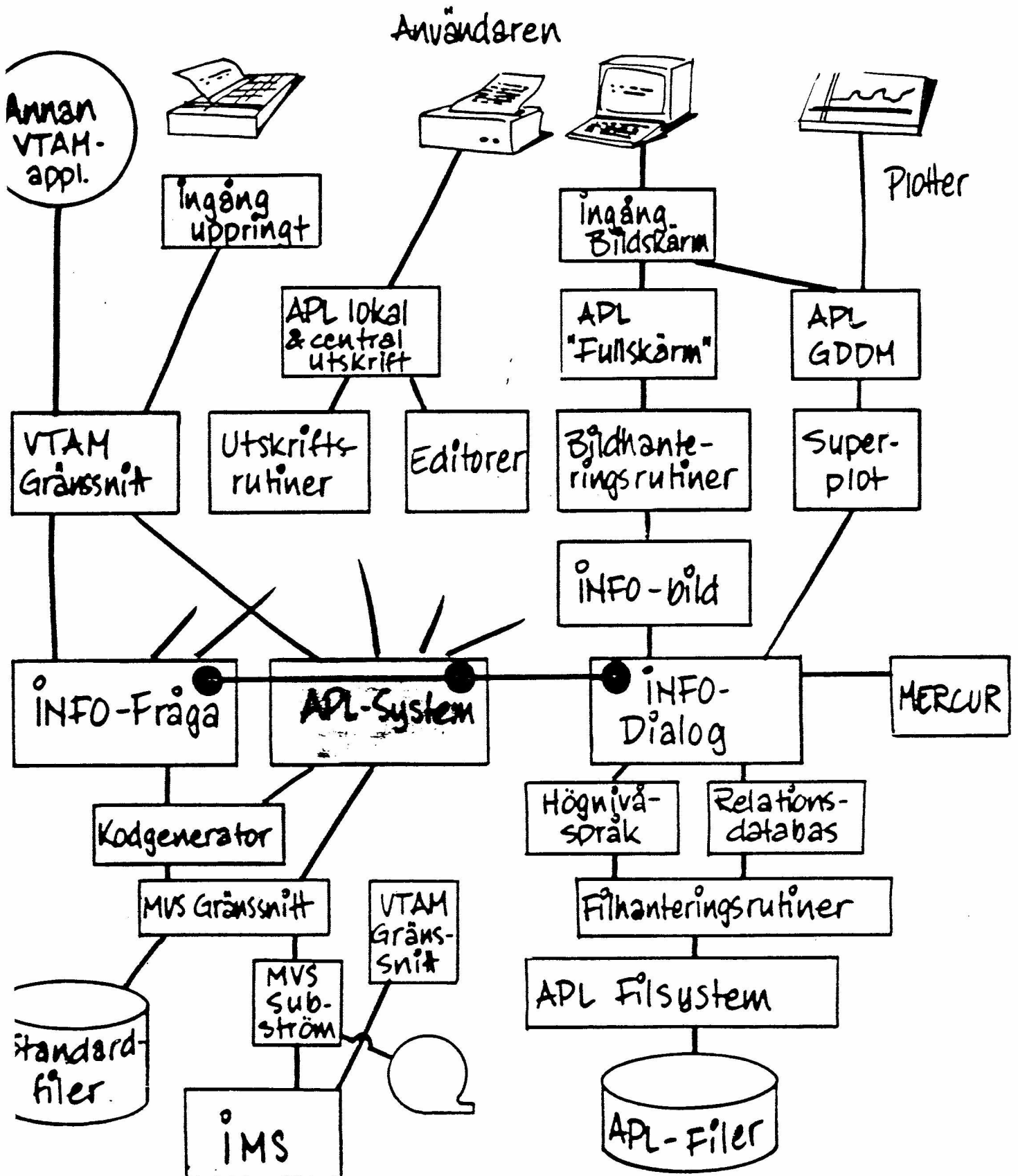
Beslutsstödsystem

i SEB-4GL



OBS! Ingen programmering på någon nivå

SEB-4GL APL-miljön Decentrala system



3. KOMMUNDATAS ANVÄNDARANPASSNING AV UTVECKLINGSPROCESSEN

Kommundata har utvecklat tusentals system sedan starten på 60- talet och investerar årligen åtskilliga miljoner kronor i utvecklingsresurser (1984 ca 100 milj). Vi har därför försökt att följa med utvecklingen av metoder, verktyg och modeller för att få en aktiv utvecklingsprocess.

Jag tror att vi har gjort samma fel som många ADB-avdelningar. Vi har koncentrerat oss på att finna metoder och verktyg för den egna personalen, specialisterna. Successivt har användarnas medverkan i utvecklingsarbetet ökat, men de har fått deltaga på specialisternas villkor. Använda metoder och verktyg har varit koncentrerade på att beskriva de tänkta systemens uppbyggnad. Vi utgick ofta från en problembeskrivning, hoppade över analysen av orsak och möjligheter att utveckla verksamheten, och letade direkt efter tekniska lösningar som kunde eliminera de symptom på problem vi hittat.

Datamodellering och prototyping - en förändring

Dessa nya metoder har inneburit en förändring. Användare har deltagit i utvecklingsprocessen på ett mer konstruktivt sätt. Vi har plötsligt arbetat med konkreta begrepp och rutiner i verksamheten och det har varit lättare att bedöma verklig nytta och effekt. Men datamodellering och prototyping räcker inte. Vi måste skapa en "komplett" modell med praktiska arbetssteg.

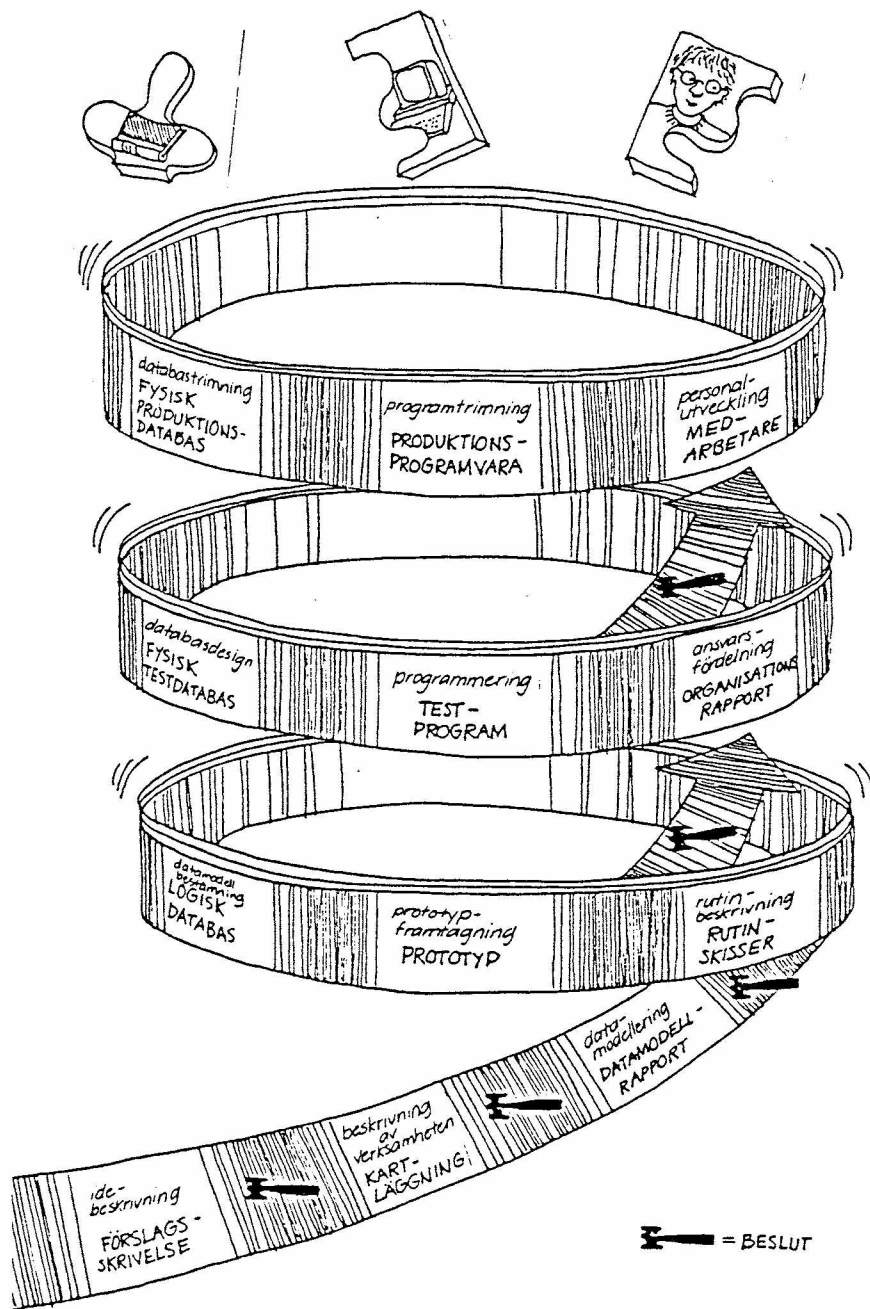
Det krävs hjälpmedel

Att arbeta i en iterativ process med många grupper och med grafiska modeller och beskrivningar, som ofta skall ändras och uppdateras, kräver ett datorstöd. Det finns ett antal verktyg som stödjer själva programkonstruktionen och databasuppbyggnaden. Men vi har haft svårt att finna verktyg som stödjer arbetet i de inledande faserna.

Krav på verktyget

Exempel på krav vi ställer på ett verktyg som skall användas i de inledande faserna i utvecklingsprocessen:

- Man skall på ett enkelt sätt kunna rita upp olika grafer och modeller som datamodeller, verksamhetsgrafer (MBI), systemskisser (SAK) och programstrukturer (JSP, JSD).
- Fri text, identifierare och beskrivningar skall kunna skrivas direkt i grafen och relateras till specifika objekt eller relationer.



SVEA - vårt förslag till modell

Kommundatas förslag till modell finns avbildat ovan. Formen av en spiral skall ange den process av iteration mellan olika aktiviteter och arbetsgrupper som krävs för att få en gemensam utveckling av informationssystemets tre pusselbitar individ/org, information/data och hårdvara/mjukvara. Modellen finns närmare beskriven i Utvecklingshandboken SVEA (Kommundata 85 82s).

- Objekt och relationer på en övergripande nivå skall kunna brytas ned i underliggande grafer eller tabeller.
- För varje detalj i en graf skall man på en speciell bild kunna beskriva detaljen mer ingående i fri text.
- Man skall kunna måla och testa skärmbilder och rapporter direkt utan att först ha varit tvungen att specificera en databas eller systemstruktur.
- Alla funktioner i verktyget skall vara kopplade till ett aktivt data dictionary.
- Grafer skall kunna ändras på ett enkelt sätt genom att peka på det objekt som skall flyttas och sedan på den plats i grafen som det skall flyttas till. Alla relationer och texter som är kopplade till aktuellt objekt skall automatiskt följa med.
- Man skall kunna hoppa mellan definitioner, objekt, relationer och nivåer i modellstrukturen genom att peka i grafen.
- Verktöget skall arbeta på en PC men resultaten skall kunna flyttas till andra datormiljöer.
- Verktöget skall innehålla ordbehandling och ha funktioner för att framställa projektdokumentation. Informationen i data dictionary skall kunna integreras med text skriven med ordbehandlaren.
- En person som kan hantera en terminal skall kunna använda verktöget efter 2-3 timmars instruktion.

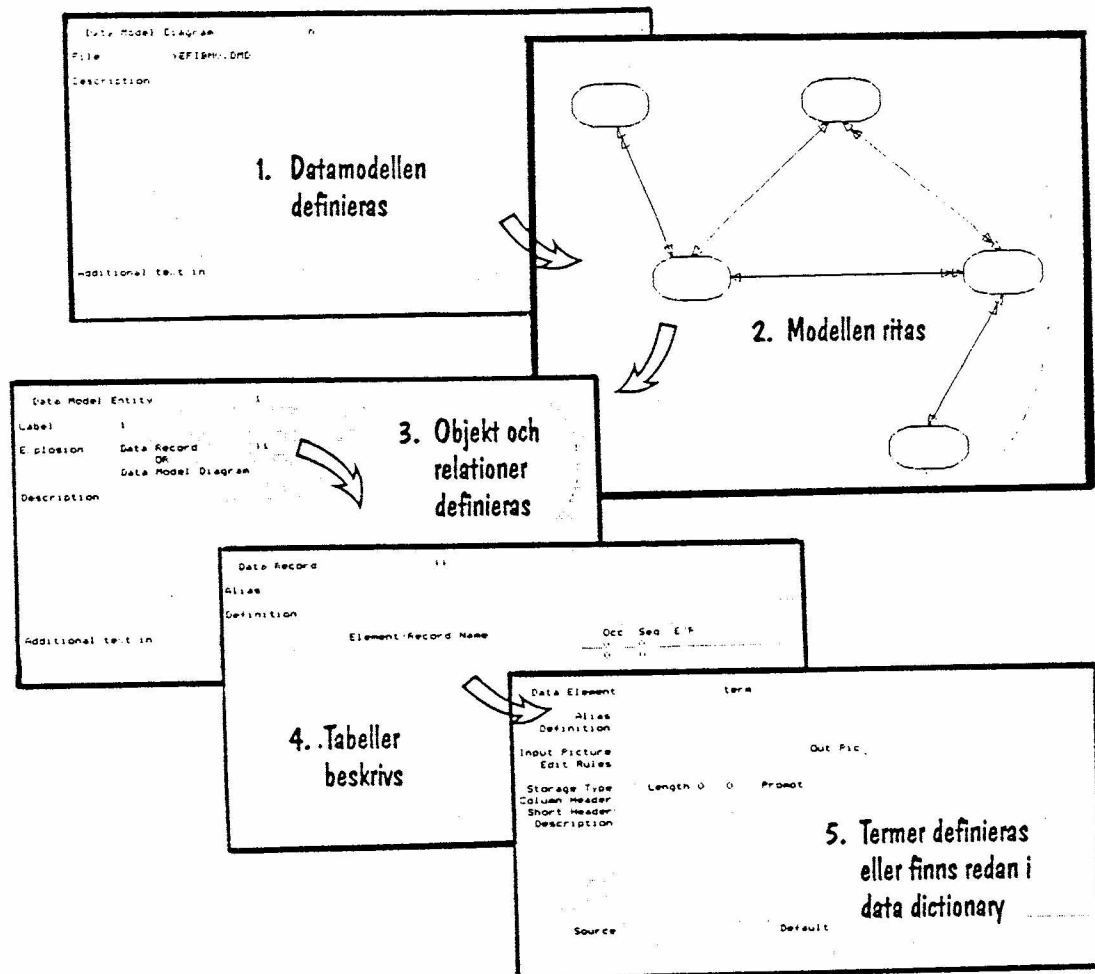
EXCELERATOR en lösning?

Vi har nu funnit ett verktyg som vi tror kan fylla de luckor vi har haft i datorstödet för de inledande utvecklingsfaserna. Det heter EXCELERATOR och är utvecklat av Index Technology Corp Boston. Det köres på en IBM-kompatibel PC 512k med primärminne, styrdosa och en fast disk med 10mb minne.

Vi har bedömt verktöget och de organisationer som arbetar kring det som så intressanta att Kommundata och ISI A/S gemensamt är återförsäljare i NORDEN. Detta är ett nytt grepp från företaget men vi tror att vi måste ta ett steg närmare verktygsproducenterna för att aktivt kunna påverka verktygens utveckling. Verktögen kommer mer och mer att bli grundstommen i utvecklade system.

I nästa figur visas ett exempel på användningssteg i EXCELERATOR.

Ett exempel på hur en datamodell kan byggas upp.



Effekten av nya metoder och verktyg?

Även om vi ännu ej har ett komplett koncept med metoder och verktyg som till 100% täcker SVEA-modellens alla delar kan vi redan se effekter av de nya arbetssätten.

Om användaren deltar aktivt på det sätt som vi beskriver i SVEA kommer vi närmare de "dagliga problemen" och har större möjlighet att uppnå önskad nytta och effekt i verksamheten. Den totala utvecklingstiden tills vi har ett system som verkligen fungerar i organisationen förkortas.

Jag tror att användarnas aktiva deltagande i datamodelleringar och andra analyser av verksamhetens informationsanvändning kommer att medföra både en explosion och revolution för dagens ADB-system. Användaren kommer inte enbart att deltaga i olika utvecklingsprojekt; hon kommer själv att aktivt driva "vardagsutvecklingen och rationaliseringen".

4. SEKRETERARROLLEN VID ORD- OCH TEXTBEHANDLING

Ett kontorsyrke i förändring

INLEDNING

Datatekniken och dess användning är en faktor som i hög grad styr utformningen av vårt arbetsliv och samhällsliv. Datatekniken griper in i allt fler områden av samhället. Både "hårdvara", den fysiska utrustningen och "mjukvara", de program som knyts till datorn, förändrar vår verklighet. Det gäller arbetsmarknaden i stort, det enskilda företagens verksamhet och de arbets- och livsvillkor som vi har.

Flera av oss har inte "skolats in" i ett datorsamhälle. Det möjliggör för oss att ställa frågor och peka på önskvärda förändringar.

RAM-projektet, Studier av rationaliseringens effekter på arbetsmiljö och arbetsförhållanden för några tjänstemannagrupper, har sedan mitten av 1970-talet bedrivits vid Stockholms Universitets sociologiska institution. Projektet har genomförts i två faser, RAM:1, och RAM:2.

Den här sammanfattningen avser en delstudie inom RAM:2-projektet och syftet har varit att beskriva och analysera effekterna av mikrodatoriseringen i samband med införande av ord- och textbehandling inom en grupp kvinnliga kontorsanställda. Undersökningsgruppen har utgjorts av kvinnliga sekreterare vid Philips Elektronikindustrier AB.

Delstudien är en fördjupning och fortsättning av den forskning som tidigare har utförts i det så kallade RAM:1-projektet (se Arbetarskyddsfondens sammanfattning nr 611).

Studien är indelad i tre problemfält eller nyckelområden. Med utgångspunkt från dessa studeras även arbetsuppgifternas karaktär och innehåll samt den arbetsorganisatoriska strukturen i företaget:

- * Datateknik och kommunikationsstruktur i arbetet
- * Datateknik och ledarskap/medarbetarskap respektive chefsroll/medarbetarroll
- * Datateknik - arbetsliv - livsvillkor utanför arbetet

De fullständiga resultaten från undersökningen redovisas i boken Sekreterarrollen och ord- och textbehandling - kontorsyrke i förändring (se slutet av sammanfattningen). Resultaten är möjliga att generalisera till en större grupp kontorister eftersom undersökningsgruppen till stor del har motsvarande arbetsuppgifter.

RESULTATBilden av sekreteraryrket

För att få en uppfattning om sekreterarrollens utveckling i samband med datoriseringsprocessen har en litteraturstudie gjorts. Den har omfattat några sekreterarhandböcker från decennierna närmast efter andra världskriget. Böckerna förmedlar en nyanserad bild av en yrkeskunnig och väl meriterad kvinna. Men den bilden har kommit att undanskymmas. I stället har sekreteraryrket setts som en förlängning av en traditionell kvinnoroll och arbetet har inte värderats särskilt högt. Det framgår av de debattböcker, utredningar, uppsatser och tidningsartiklar som också har ingått i litteraturstudien.

Jämförande analyser mellan sekreterare och andra kvinnliga anställda

För att få reda på om det finns några skillnader mellan sekreterare och övriga kvinnliga anställda när det gäller faktorer som

- * socialpsykologiska aspekter av arbetsmiljön
- * demokrati på arbetsplatsen
- * effekter av terminalsystemet och datorisering i övrigt samt
- * hälsobesvär

har en fördjupad analys gjorts från ett enkätmaterial i en tidigare studie inom RAM:l-projektet. Materialet gäller personal vid ett större försäkringsföretag i Stockholm. Gruppen övriga kvinnliga anställda bestod av handläggande personal.

Det visade sig att gruppen sekreterare var mer missnöjda (statistiskt säkerställt) med följande aspekter av arbete och arbetsvillkor:

- * Graden av intressant arbete
- * Graden av intressant arbete vad gäller tillvaratagande av mänskliga resurser
- * Självständigheten i arbetet
- * Kontaktmöjligheterna i arbetet
- * Sammanhållning med arbetskamraterna

Vidare var de mer missnöjda med:

- * Befordringsmöjligheterna
- * Demokrati och inflytande i arbetet
- * Inflytande på arbetstider och ledighet

Sekreterarna var också mer missnöjda med områden i arbetet som brukar föras fram som viktiga för arbetstillfredsställelsen. Detta tycks ytterligare bekräftas av det faktum att de hade sämre hälsotillstånd (statistiskt säkerställt) - de hade oftare magbesvär och nervösa besvär.

De hade emellertid en positivare inställning (statistiskt säkerställt) till den nya tekniken. Det gällde terminalsystemets effekter på arbetstakt, uppmärksamhet och ansvar samt effekter på fysiska och psykiska besvär i samband med ny teknologi - såväl terminalsystemet som datorsystemet i övrigt på företaget. De tycktes sålunda se tekniken som en möjlighet till befrielse från en yrkesroll som inte fungerade bra.

Analyserna bygger på en enkät som insamlats 1977. Företaget hade använt ord- och textbehandling några år och var ett av de första företagen som börjat använda denna teknik.

På många arbetsplatser befinner sig fortfarande denna yrkesroll i en utvecklingsfas där, i samband med ny teknik, möjligheter och risker synliggörs, vad gäller yrkesrollens framtida utformning.

Vilka förändringar kan iakttagas vad avser arbetsinnehåll bland sekreterarna?

Vilka arbetsuppgifter utför en sekreterare?

De arbetsuppgifter som sekreterarna vid Philips Elektronikindustrier AB anger att de till största delen utför under en arbetsdag är just de uppgifter som de anser ha ökat sedan ord- och textbehandlingstekniken började tas i bruk - skrivarbete, bokning av resor och hotell samt att ordna sammanträden och konferenser.

Vilka förändringar i det objektiva arbetsinnehållet har skett i samband med användningen av ord- och textbehandlingsteknik?

40 procent av de intervjuade anger att en utökning har skett av själva skrivarbetet. En tolkning som framförts är att mängden text som skrivs ut nu har blivit större. "Tyngre"

dokument läggs t ex inte ut på skrivbyråer som tidigare, 60 procent av de intervjuade anger att redigeringsarbetet ökat. Att boka resor/hotell, arrangera sammanträden och konferenser samt preparera tabell- och bildmaterial har i någon mån ökat. Att ha en sammanhållande funktion har både ökat och minskat. De flesta intervjupersonerna skulle inte vilja vara utan detta tekniska hjälpmedel.

Att skrivarbetet ökat kan man kommentera på följande sätt.

Datorn används som ett "effektivitetshöjande" hjälpmedel. Att människans resurser används på ett ändamålsenligt sätt är bra både för de anställda och organisationen. Tyvärr blir det ofta så att man inte får mer tid över för t ex arbetsbetingade personliga kontakter såsom kunder och till vidareutbildning som aktualiseras av den nya tekniken. Snarare läggs mer och mer arbete av samma slag på en individ t ex antal ärenden, sökningar per tidsenhet mångdubblas, utskriftsarbete. Våra egna ambitioner och krav på oss själva ökar samtidigt. Detta är förvisso inte teknikens fel i sig, utan beror snarare på dåligt genomtänkta principer för t ex administrativ rationalisering.

Ska det tekniska hjälpmedlet användas/införas bör arbetet spridas på olika yrkeskategorier och nivåer relativt samtidigt. Möjligheter finns därvid att omforma yrken som idag har alltför stor ensidighet, både "höga" administrativa befattningar (handläggare, chefer) och mer detaljstyrda befattningar. Hittills har man huvudsakligen inriktat datoriseringen på de lägre nivåerna i företagen.

Att redigeringsarbetet ökat antyder en förskjutning av arbetsinnehållet för sekreterarna med påföljande utbildningskonsekvenser. Det torde bli aktuellt med utbildning i t ex språkbehandling, redigering och layout.

De sekreterare som önskar reducera mängden skriv- och redigeringsarbete för fram förslag till lösningar som rör arbetsutvidgning. Man nämner även organisation av skrivcentraler. Man sammanför det rena skrivarbetet till en organisatorisk enhet, inom eller utom företaget. Sådana förslag till lösningar skulle leda till osynliggörande av behovet att se på arbetsinnehåll, arbetsuppgifter som en fördelningsfråga som bör tas upp på den egna arbetsplatsen.

Fördelningsfrågan gäller gränsdragningar mellan olika befattningar och yrkesroller. En tydlig nomenklaturförvirring finns. Det "korridor snackas" t ex om att användare av ord- och textbehandlingsutrustning, OTB-användare, "skrivs upp till terminalarbetare" eller "ned till kontorister". Sådana resonemang är även tecken på problem som rör den egna yrkesidentiteten och som har en historisk utveckling relaterad till användning av ny teknik inom kontorsarbete.

Förvirringen i nomenklaturplaceringen beror också på att man av vana ser organisationen enbart statisk och inte som en process, där ständig utveckling sker.

Fem aspekter på förändringar vad avser arbetsinnehållet i samband med datoriseringen aktualiseras här:

- * tillföra andra arbetsuppgifter i "produktionskedjan", vilka finns på samma nivå
- * tillföra planerings- och kontrolluppgifter som finns på annan nivå
- * analysera om vissa arbetsuppgifter bör göras av maskin
- * analysera vilka arbetsuppgifter som bör göras av maskin eller människa
- * analysera om datoriseringen är lämplig (etik, kvalitet etc)

Har förändringar skett i subjektiva aspekter av arbetsinnehållet?

Om vi först ser på ett antal frågor som gäller socialpsykologiska aspekter av arbetsinnehållet (socialpsykologisk och psykosocial används här synonymt) så har en stor andel angivit ett oförändrat tillstånd. Cirka hälften (52%) har markerat att graden av stimulans och spänning i arbetet ökat, 44% att omväxlingen i arbetet ökat, 40% att möjligheter att använda egna ideer och kunskaper ökat.

Vad gäller olika aspekter av inflytande i arbetet, så har likaledes en stor andel angivit ett oförändrat tillstånd. 60% har dock markerat att inflytande på det egna arbetstempot har ökat. Omkring en tredjedel har menat att inflytande på planeringen av det egna arbetet (32%) respektive inflytande på val av arbetsmetod, dvs sätt på vilket det egna arbetet skall utföras, har ökat (36%).

Resultaten är direkt motsatta till erfarenheter från främst RAM:1-projektets första företag, ett affärsdrivande statligt verk, där ett datorsystem av s k batch-typ användes (satsvis bearbetning av information). Det var ett exempel på användning av centraliserad datorkraft - stordatortiden.

Att inflytandet i det egna arbetet ökat vad gäller arbetstempo, planering och val av arbetsmetod, kan visa på en möjlighet som decentraliserad datorkraft, i form av ord- och textbehandlingssystem, erbjuder. Användning av mikrodatorbaserad utrustning ger andra förutsättningar för arbetsvillkorens utformning. Som exempel kan nämnas att man vid mikrodatoranvändning inte i varje moment är beroende av teleteknik och telenätet. Vidare finns andra möjligheter för strategier för införande.

Vad avser frågor om arbetstakt och beroende av maskinutrustning anger så många som 64% att kraven på uppmärksamhet och noggrannhet har ökat i samband med användning av ord- och textbehandling. Likaså säger 60% av intervjupersonerna att arbetstaktens beroende av maskin/utrustning har ökat.

Datorn används de facto som ett rationaliseringsinstrument. Intensifiering av arbetstakt och ökat maskinberoende blir ofta följderna. Yttre krav på prestation, som möjliggörs med ny teknik blir lätt inre krav som vi ställer upp för oss själva i arbete och privatliv. Detta fenomen har kommenterats på flera håll i boken "Arbetsmiljö och terminaler" (Bradley 1981, ASFs sammanfattning nr 611). Vårt förhållningssätt till tiden är och kommer att vara en av nyckelfrågorna för framtiden.

Hur kan sekreterarens framtidsperspektiv på arbetsinnehållet beskrivas?

Vad gäller möjligheter och farhågor inför framtiden i samband med OTB-användning finns å ena sidan en oro för den egna yrkesrollens utveckling, en risk att bli utbytbar. Å andra sidan finns också en framtidstro, tankar om positiva utvecklingsmöjligheter för den egna yrkeskåren. Man ger en rad exempel på hur man kan återerövra den allsidighet i yrket som ursprungligen funnits, vare sig man kallas operatör, assistent eller kontorist. Man markerar ett intresse för att arbeta med t ex budgetfrågor, planering och personalfrågor.

Har kommunikationen på arbetsplatsen förändrats i samband med användning av ord- och textbehandling?

Kommunikation och datorisering

Ordet kommunikation kan ges olika innebörd. Kommunikation ses här ur ett socialpsykologiskt perspektiv, huvudsakligen knutet till arbetslivet. Tyngdpunkten ligger på samspel mellan människor både kvantitativt och kvalitativt.

Förändringsprocesser i samband med datateknologin påverkar kommunikationen direkt analogt med syftet med rationaliseringsåtgärden eller syftet med datasystemet. Exempelvis medför användning av bildskärmsterminaler att man "samtalar" direkt med en dator, samtidigt som bakomliggande system i sig är utformade av människor.

Datateknologins inverkan sker även indirekt. Tekniken och det sätt på vilket den används påverkar nämligen en rad områden i arbetsmiljön (organisations- och arbetsutformning, maktstruktur på arbetsplatsen och arbetstider). Vissa fenomen i dessa arbetsmiljöområden inverkar sedan i sin tur på kommunikationen mellan människor.

Även om en rad viktiga strukturella förutsättningar finns för samhörighet och samarbete för vi dock till gruppen eller arbetsgemenskapen med oss en rad erfarenheter och upplevelser från familj och skola som kan försvåra och äventyra våra relationer på arbetet.

Det är viktigt - också ur arbetsmiljösynpunkt att arbeta med "sig själv", att utveckla den egna personen på olika sätt. När vi arbetar med förändringar t ex i samband med datorisering är det viktigt att vi "mår bra", att spelrum för neurotiskt förankrade försvarsmekanismer minimeras.

Ibland kan förändringsarbete behöva inledas med kurser i självförtroende - så att vi vågar framföra våra åsikter och ta fram våra erfarenheter.

Öppenhet, förtroende/tillit, respekt och målorientering i arbetet är faktorer som sekreterarna framförde när de skulle beskriva begreppet kvalitet i kommunikation mellan människor på arbetsplatsen.

Vad i arbete och arbetssituation bidrar till att kommunikationen fungerar bra?

De orsaker som nämns associerar man till största delen till fenomen/egenskaper hos individen själv. Intervjupersonerna tar i svaren upp både bidragande och hindrande fenomen. Man nämner "avundsjuka", men går inte in på tänkbara orsaker här till, t ex belöningsystem, organisatorisk placering. Man nämner "känsla av självständighet", men tar inte upp frågan huruvida de egna arbetsuppgifterna ger utrymme för självständigt agerande. Det är ett vanligt fenomen i alla organisationer att en individualisering, en psykologisering sker av samarbetsproblem. Det är ofta svårt att analysera orsaker knutna till objektiva, strukturella faktorer på olika analysnivåer.

De hinder som nämns är knutna både till individen själv t ex slutenhet, rädsla, oförmåga att både ge och ta och till faktorer i arbete/arbetssituation t ex tidsbrist och överbelastning, nomenklaturklarheter, rumsfrågor, chefsöverskott.

Personlig kontakt i arbetet önskar

- * 32 procent i alla sammanhang
- * 24 procent vid vissa arbetsuppgifter, exempelvis när information skall utbytas ömsesidigt, vid kundkontakter och vid kontakt mellan avdelningar inom företaget.
- * 20 procent när känslomässiga saker finns med i bilden

Så många som 16 procent av intervjupersonerna anger att de inte i något sammanhang vad avser arbetets karaktär önskar ha personlig kontakt.

Vad gäller hinder för personlig kontakt nämner 44 procent psykosociala hinder, främst:

- * statustänkande och språkbarriärer knutna till relationen mellan yrkesgrupper
- * arbetsbelastning och stress
- * personanknutna orsaker som indirekt kan härledas till arbete och arbetssituation

Arbetsbelastning och stress har ökat i samband med användning av ord- och textbehandling. Arbetsbelastning och stress påverkar och utgör hinder för personlig kontakt i arbetet enligt resultaten ovan. Ett av de fysiska hindren för personlig kommunikation som nämns är bundenheten till OTB-utrustningen.

Vilka förändringar i kommunikation och kontakt har skett i samband med användning av ord- och textbehandling?

Förändringar i kommunikation har studerats med avseende på tre aspekter:

Kvantitativa aspekter (om kontakt tillkommit, försvunnit, utökats, nedtonats o d). Kontakten med chef, närmaste medarbetare (tekniker, ingenjör) samt kunder har utökats i stor utsträckning. För övriga kontakter är "oförändrat" det vanligaste svaret. Men över lag förekommer fler kontakter per tidsenhet.

Kontaktsätt (personlig, telefon, brev, telexkontakt o d). De flesta intervjupersonerna har en OTB-utrustning som ej har kommunikationsmöjligheter och svaren återspeglar detta förhållande. Möjligheten att sammankoppla mikrodatorbaserad utrustning via teletekniken är en kritisk fråga för kommunikation, ur både tekniska och socialpsykologiska aspekter.

Innehållet i det som orsakar kontakter i arbetet. Det man pratar om med varandra har alltmer kommit att få ett tekniskt innehåll (jämför psykosociala hinder för personlig kontakt i det föregående). Den språkklyfta som finns mellan sekreterare och tekniker/chefer, delvis beroende på olika utbildningsbakgrund, tenderar att förändras.

Det har uppstått nya gemensamma arbetsintressen - "man glider över till att förstå dataspråket mer". Innebär detta att man distanserar sig från arbetets egentliga innehåll och resultatet av arbetsinsatserna? Har man mer kommit att koncentrera uppmärksamheten på själva tekniken som redskap? Detta är svårt att utläsa ur intervju svaren, då produkterna av arbetsinsatserna i denna bransch utgörs av teknik, såväl mjukvara som hårdvara.

Inom gruppen sekreterare har användningen av ord- och text-behandling blivit en direkt "kontaktorsak". - Tekniken har bidragit till större samhörighet inom denna yrkesgrupp.

På vilket sätt kan kommunikationen förbättras?

Intervjusvaren visade att frågan associerades med förbättringar i teknisk mening, t ex snabbare informationsutbyte, snarare än socialpsykologiska aspekter av kommunikationen.

Hur ser det formella respektive informella kontaktnätet ut på arbetsplatsen?

Sekreterarmötena är exempel på informella kontaktnät och gruppbildningar som växer fram i samband med mikrodatoriseringen. Olika former för kollektivt agerande för kvinnor och män förekommer. De har olika utformning dels i olika länder dels under olika faser av kvinnors och mäns emancipation.

Har yrkesrollen förändrats?

Flertalet sekreterare arbetar tillsammans med en grupp ingenjörer. Över hälften av sekreterarna (52%) tycker att den egna yrkesrollen har ändrats i samband med OTB och mer än hälften ser positivt på sina möjligheter till utveckling i företaget. Det egna intresset för vidareutbildning är uttalat positivt (72%).

Enligt sekreterarnas uppfattning har ingen förändring skett av chefens roll i samband med OTB (72%) och inte heller i samspel och samarbete mellan sekreterare och chef(chefer (76%).

Könsroll och yrkesroll är invävda i varandra. Yrkesrollen är enligt sekreterarens uppfattning enbart inriktad på de förväntningar, rollförväntningar från omgivningen, som finns på hennes yrkesutövande. Roll i den socialpsykologiska litteraturen refererar till det beteendemönster som förknippas med både skyldigheter och rättigheter i en viss social position t ex ett yrke. Individerna kan göra bruk av skyldigheter och rättigheter i en position, dvs ta en roll.

Med hjälp av en rollteoretisk utgångspunkt kan man dels föra fram och sammanfatta några centrala resultat i vår studie, dels mer generellt diskutera datateknik och förändring i arbetslivet.

Har användning av ord- och textbehandling indirekta eller direkta effekter på privatliv/familjeliv/fritid?

Både möjligheter till och intresse för (respektive procenttal inom parentes) flertalet av de vanligaste fritidssysselsättningarna har förändrats i samband med den nya tekniken. Det gäller kursverksamhet (40% resp 56%), hem och familjeliv (20% resp 36%), samvaro med vänner (20% resp 24%), läsning (28% resp 20%), TV-tittande (24% resp 20%), hobbyverksamhet (24% resp 20%). Dessa resultat kan jämföras med tidigare studerade företag (RAM:1), där sådana effekter av datoriseringen i arbetslivet på fritiden och dess innehåll inte har erhållits.

Undersökningen pekar på ett ökat kollektivt agerande, men detta tar sig här uttryck i form av de så kallade sekretärarmötena, som ej har facklig anknytning. Att kvinnliga anställda har ett lågt fackligt intresse har en rad orsaker.

Viktigt att notera är att intresse för kursverksamhet har ökat påtagligt - en viktig förutsättning för en positiv utveckling av yrkesrollen i samband med ny teknik.

På frågor om vad man skulle vilja få ut av den tid man inte förvärvsarbetar anger en större del (60%) av undersökningsgruppen aktiviteter som kategoriserats som självförverkligande (studier, resor och kulturella aktiviteter).

Även när det gäller utbyte av arbete väger självförverkligande (stimulans, frihet, självkänsla och delaktighet) tyngst. Sociala motiv (kontakt med andra människor) har också en relativt stor betydelse.

Den arbetsfria tiden (vardagskvällar och helger) sammanknyts av 32% renodlat med avkoppling och ledighet, av 52% med avkoppling/ledighet tillsammans med merarbete (hemarbete på kvällen: städning, tvätt etc). Renodlat merarbete sammanknyts med den arbetsfria tiden av 16%.

DISKUSSIONSFRÅGOR

Den här undersökningen har visat på flera intressanta resultat när det gäller effekterna för de anställda vid datoriseringen i arbetslivet. Det finns många frågor att diskutera. Här är exempel på några:

- * Har införande av ord- och textbehandling förändrat inställningen till datorer och till framtiden?
- * Hur ska man se på problemet sysselsättning, trygghet inför framtiden?
- * Vilka risker och möjligheter gäller för barnen?

- * Vilka åtgärder bör olika aktörer vidta vad avser arbetsmiljö och datoriseringen?
 - * Vilka krav bör man framföra på ett ord- och textbehandlingsystem?
 - * Vilken önskvärd utveckling bör gälla för den egna yrkesrollen?
-

BOKEN

Sekreterarrollen och ord- och textbehandling - ett kontorsyrke i förändring (221 sid) kan beställas kostnadsfritt från Arbetarskyddsfonden, Box 1122, 111 81 Stockholm, tel 08-14 32 00.

Studien har följts av en lokal, partssammansatt referensgrupp, utsedd av skyddskommitten inom företaget. Resultat och erfarenheter har successivt diskuterats med referensgruppen. Dessutom har resultaten återförts till sekreterargruppen, produktansvariga inom försäljningsorganisationen och nyckelpersoner inom företaget.

Under hösten 1985 utkommer Bradley med boken "Psykosocial arbetsmiljö och datorer". Akademilitteratur.

