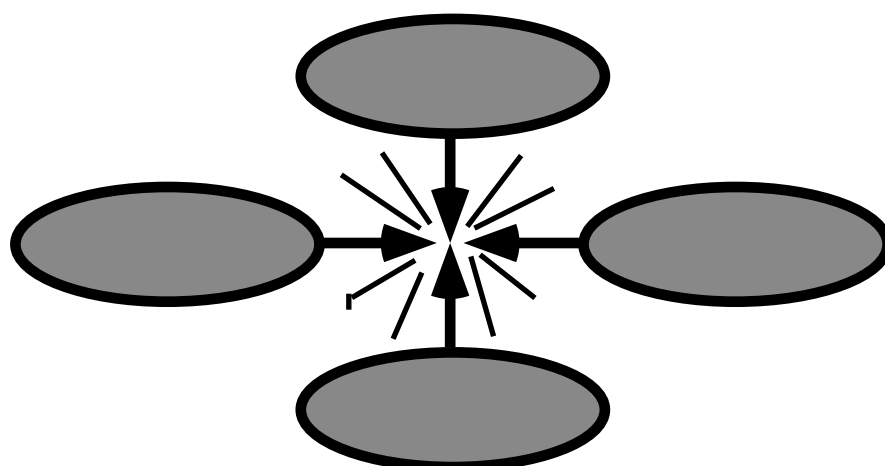


Per Flensburg

Personlig databehandling

introduktion, konsekvenser,
møjligheter



Förord

Det arbete jag här presenterar är resultatet av åtskilliga års arbete. De första spåren är mer än tio år gamla, men huvuddelen av arbetet har utförts under åren 1983-1986. I min ursprungliga uppläggning tänkte jag ta med läsaren på en färd genom de tio årens upplevelser och först på sista sidan tala om vad det hela egentligen handlar om. Nu blev det inte så, men förmodligen har materialet fortfarande en del av denna karaktär. Jag beskriver en forskningsprocess och inte endast ett forskningsresultat. Vidare försöker jag leva som jag lär och undviker i görligaste mån förutsättandet av disponibiliteten av en alltför alltför avancerad icke inhemsk nomenklatur! Forskning och forskningsresultat bör enligt min mening beskrivas på ett så pass enkelt sätt att en intresserad lekman kan förstå vad det rör sig om. Förmodligen har jag inte lyckats hundra procentigt i detta opus; det rör sig dock om en akademisk avhandling.

Det är många personer jag är tack skyldig när jag nu framlägger mitt verk. Min handledare, Markku Nurminen, har med kloka och underfundiga kommentarer hjälpt mig komma framåt. Kollegerna på institutionen har stött och hjälpt mig, främst då LUIS-gruppen. Ett speciellt tack vill jag framföra till Gunhild Sandström vars stöd och uppmuntran genom åren helt enkelt inte går att värdera, Siv Friis och Jan Berglind har också alltid ställt upp med råd och dåd då det kärvat till sig. Med Hans-Erik Nissen har jag under årens lopp haft många givande duster och utan hans helt otroliga energi och arbetsförmåga hade förmodligen aldrig detta verk kommit till.

Genom att granska manuskriptet har Agneta Olerup hjälpt mig undvika en hel del grodor. Niels Bjørn-Andersen, Kristo Ivanov, Margareta Bertilsson och Pertti Järvinen har även kommit med värdefulla synpunkter. Perttis hjälp med referenslistan uppskattar jag speciellt. Till alla dessa framför jag härmed mitt djupt kända tack!

Lund mars 1986

Per Flensburg

1. INLEDNING

En avhandling är redovisning av forskning, som bedrivits under en längre tidsperiod. Vad vore då bättre inledning till en redovisning än att beskriva hur det hela började? I mitt fall började det med ett bord!

Antag, att en forskare ska förklara för en av Nya Guineas mörkhyade urinnevånare vad ett "bord" är för något. Kanske går dialogen ungefär så här:

– Ett bord är en fyrkantig skiva med fyra ben på.

Vår mörkhyade gentleman, som är mycket klipsk, pekar på en stol och säger:

– Jag förstår. Det där är ett bord.

– Nej, det är inget bord, det är en stol. Bord har inget ryggstöd.

Nu pekar urinnevånaren på en byrå och säger:

– Jaha, men det där är ett bord för det har skiva, fyra ben och inget ryggstöd.

– Nej, det är en byrå. Bord har inga lådor under sig.

– Jag förstår. Ett bord är en fyrkantig skiva med fyra ben, utan ryggstöd och utan lådor under. Då kan det där inte vara något bord!

Säger Nya Guineanen och pekar på forskarens runda, trebenta matsalsbord.

– Jo, det är ett bord. Skivan behöver inte vara fyrkantig och antalet ben behöver inte vara precis fyra.

– Hur många ben måste det minst vara?

– Ja, det måste vara minst tre ben, för annars kan bordet inte stå. Men det finns ingen övre gräns. Det beror faktiskt på hur stort bordet är. I praktiken tror jag inte att det finns bord med mer än tio ben.

Vår vän från södra halvklotet pekar då på forskarens trädgårdsbord, som är en rund kvarnsten, stående på ett tjockt cementrör i mitten.

– Det där är alltså inget bord, för det har bara ett ben.

– Jo, det är ett bord! Jag hade fel när jag sa att tre ben var minsta antalet ben. Men mindre än ett ben kan man inte ha!

Vår öinnevånare får nu syn på sonens bord, som är ett väggfast fällbord.

– Det där kan uppenbarligen inte vara något bord, för det har inget ben alls.

– Faktiskt är det också ett bord. Tydligt behöver ett bord inte ha några ben. En skiva är tillräcklig.

Forskaren drar i detta ögonblick fram skärbrädan, som sitter ovanför kökslådorna för att skära sockerkaka till kaffet. Forskarens långväga gäst ser detta och undrar:

– Är det där också ett bord?

Nu börjar Forskaren sväva på målet och tar hastigt sin främmande vän under armen för att visa honom sängkammaren. Forskaren vet att där finns det i varje fall inga konstiga bord, som hänger på väggen eller inte har några ben. Men han hade glömt sitt eget skrivbord, som han ärvt av sin far och som var en bastant möbel med tjock skiva och lådhurts på ena sidan och skåphurts på den andra. Den mörkhyade urinnevånaren tittar på skrivbordet och småler:

– Det där är säkert inte ett bord, för det har lådor under sig!

– Hm, det där är ett bord. Det kallas skrivbord och används när man sitter och arbetar. Det skiljer sig från en byrå genom ett ledigt utrymme i mitten.

– Om jag då fattat saken rätt är ett bord en skiva, utan ryggstöd och med ett fritt utrymme under.

– Precis, jag kan inte uttrycka det bättre själv!

Forskaren drar en lättnadens suck och skyndar sig att lotsa ut sin vän från sängkammaren innan denne får syn på sängarna, som faktiskt har ett ledigt utrymme under sig. Man går in i biblioteket och där får vilden syn på Carl Malmstens runda bord med två små skivor.

– Det där är inget bord, för det har två skivor. Och om det är ett bord, måste alla dessa här andra föremålen också vara bord!

Säger han och pekar på alla bokhyllorna. Forskaren, som hittills visat prov på stort och änglalikt tålmod, tappar nu allt detta och hans yttrande är definitivt inte lämpligt att återge i en akademisk avhandling!

Av denna lilla dialog lär vi oss att fastän vi alla vet vad ett bord är, kan vi inte ge en tydlig och uttömmande definition därav. Samtliga mina läsare kan säkert skilja mellan bord, stolar, byråer, skärbrädor, bokhyllor, sängar och andra företeelser som kan anses vara "bordlika". Men jag är tämligen övertygad om att ingen kan ge en beskrivning av "bord" på, säg 20 ord, som omfattar samtliga "bord" och utesluter allt som inte är "bord". Det är egentligen märkligt. Resonemanget kan upprepas för i stort sett varje vardagligt föremål i vår omgivning. Föremål kan här tagas i vid bemärkelse och innefatta även sådant som samtal, smak på kaffet, äktenskaplig lycka och mängden kvarvarande socker i sockerskålen. Inget av dessa företeelserna låter sig definieras på ett kort och uttömmande sätt. Men när man utvecklar datasystem är ju ett av de viktigaste momenten definition av termer som används i systemet. När systemet används, ska dessa definitioner gälla. Användarna måste i systemdokumentationen få

sådan hjälp att de klart kan avgöra om en viss företeelse är term A eller term B. Hur ska en sådan dokumentation överhuvudtaget kunna göras, då man inte ens kan definiera ett så enkelt begrepp som "bord"?

Om vi studerar tillvägagångssättet vid bordresonemangen ovan, finner vi att "bord" hela tiden relateras till andra företeelser i omgivningen. Det är egentligen inte möjligt att beskriva "bord" utan att samtidigt tala om "stol", "byrå", "skärbräda" etc. "Bord" hänger intimt samman med hela vår västerländska kultur. En komplett definition av "bord" skulle i själv verket innefatta en beskrivning av hela kulturen. Det är intressant att jämföra med ett hologram, där man kan ta ut en liten bit av bilden och från denna återskapa hela den ursprungliga bilden. Något liknande förekommer också i DNA-molekylerna i våra celler. Det påstås att man från en enda cell teoretiskt sett skulle kunna återskapa hela den varelse från vilken cellen är hämtad. All information finns i DNA-molekylerna. På något liknande sätt förefaller det vara med "bord". Ja, det är inte bara "bord" förresten, utan det är hela vårt språk, som är beskaffat på detta vis. Checkland (1981) redovisar liknande förhållanden som typiska för systemsynsättet. Det finns, enligt honom, olika komplexitetsnivåer med egna regler som bara är relevanta på denna nivå. De är meningslösa på andra nivåer, men finns där ändå som dolda eller rudimentära egenskaper. Följande lilla tankeexperiment belyser förhållandet ytterligare.

Antag, att vi inte har begreppet "bord". De olika företeelserna, som vi idag kallar "bord", skulle naturligtvis finnas kvar (skulle de verkligen det egentligen?) men då heta något annat. Det fyrkantiga köksbordet skulle kanske heta "grylax" och ungefär betyda "fristående arbetsplats i hushållet". Andra exempel på "grylax" skulle då vara "strykbräda" eller "tvättmaskin". Det kräver en del fantasi att föreställa sig en sådan tillvaro, men nog skulle det vara egendomligt att "köksbord" och "strykbräda" skulle vara samma typ av föremål.

Det finns verkliga och riktiga exempel på sådana här saker. Eskimåerna har t ex inte ordet "snö" i sitt språk. En eskimå kan helt enkelt inte prata om "snö" på eskimåiska. Han (eller hon) har istället 8 andra, olika begrepp. Om jag som skåning ser ut över ett snöfält är det en tämligen homogen massa av "snö". Om eskimån ser på samma snöfält är det en mosaik av olika slags "snö". Där finns "skarsnö", "kramsnö", "kornsnö", "snö som nästan smält men sedan frusit" osv. Hans uppfattning av snöfältet kommer tack vare detta att vara väsentligt annorlunda än min.

Men eftersom eskimåerna lever i en snörik tillvaro har de behov av olika begrepp för olika sorters snö. Detta underlättar deras liv. Samma sak gäller även för andra. Det finns indianstammar i Sydamerika, som inte har något begrepp för färgen "grön". Istället har de ett antal olika begrepp för olika sorters "grön". Samma sak gäller även den sociala samvaron. Det finns andra indianstammar, som inte har egennamn på personerna utan personen identifieras med hjälp av sina gärningar. På detta språk skulle kanske undertecknad heta "han som skriver så konstig avhandling"! Dessa exempel visar att:

Vårt språk formar vår verklighet

Vår verklighet formar vårt språk

Dessa två påståenden är en av mina två stora utgångspunkter för denna avhandling. Det finns flera andra forskare och filosofer som har liknande utgångspunkter. Gadamer har en liknande dialektisk inställning (Engdahl et al, 1977, s 111-112):

"Att ha språk innebär, säger Gadamer, att ha värld, ty språket uttrycker inte den talandes själsliv utan den sak om vilken det talas, ordet tjänar som en spegel i vilken saken ensam kommer till synes ... varje ord innehåller en antydning om språkets helhet och ett förhållande till hela det betraktande av världen som förverkligats i detta språk...att vara och att uppenbaras språkligt kan betraktas som två sidor av samma sak"

Min andra utgångspunkt har med helhet att göra. Som vi såg av bordresonemanget ovan kan "bord" egentligen inte definieras utan att jag definierar allt annat som också har med "bord" att göra, dvs hela den västerländska kulturen.

Den helhetsmässiga utgångspunkten har i detta skede den största betydelsen, för den bestämmer själva metoden, själva uppläggningsen. Jag kan inte gå direkt på ett väldefinierat problem och lösa det utan att först undersöka ett större område, få någon form av överblick, hitta något viktigt problemkomplex, göra olika försök till avgränsningar, undersöka vad dessa innebär för min vidare undersökning och så småningom och kanske först som slutgiltigt resultat komma fram till den precisa problemformuleringen. Men i och med att problemet är formulerat är ofta även lösningen given. Processen har likheter med den hermeneutiska cirkeln eller spiralen. Langefors (1977) citerar Habermas (1967):

"The interpreter tests his hermeneutical pre-understanding against the text and modifies it until both horizons melt together, thus until the interpretation has succeeded within a framework of a language that is common to himself and him who is being interpreted... One might, perhaps, view this framework as the result of frequently repeated clinical experiences (of psychoanalysis) which were, themselves, accumulated through the more elastic procedure of circularly reaffirmed hermeneutical presumptions (Vorgriffe)."

Habermas visar på den grundläggande hermeneutiska förståelseprocessen, nämligen just att *förstå* en text i vid bemärkelse. Men när han antydningssvis nämner psykoanalys visar han också på något annat, nämligen att förståelsen sker i ett visst bestämt syfte, i detta fall att bota patienten. Botandet av patienten är för psykoanalytiker ett problem och förutsättningen för att han ska kunna lösa detta problem är att han förstår det. Ofta är det så att i och med att problemet är formulerat är också lösningen klar. På liknande tankar bygger mycket av strukturell design och strukturerad programmering inom informationsbehandling. Det gäller att åstadkomma en såpass

"bra" problemformulering att lösningen av problemet blir i stort sett en mekanisk process. Alla studenter som slitit med en programmeringsuppgift kan säkert vitsorda att deras uppfattning av uppgiften ändrats efterhand som lösningen framskridit. Att skriva ett program innebär ju egentligen att beskriva ett problem. Först när programmet körs löses problemet. Hela denna process kallar jag *hermeneutisk problemförståelse* (fig 1.1). Spiralen representerar här själva problemlösningsprocessen, medan förståelsen ökar ju längre ner vi kommer.

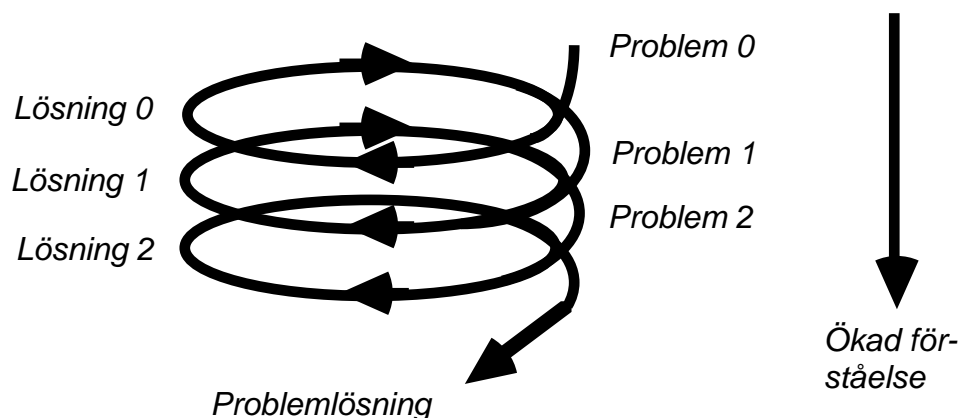


Fig 1.1 Hermeneutisk problemförståelse

Om vi nu återgår till "snö" och eskimåernas åtta "snöbegrepp" finns liknande förhållanden (fast kanske inte fullt så utpräglade) på många fler områden i vårt dagliga liv. Det exempel som är aktuellt att studera här är dataexpertens och dataanvändarens uppfattning av samma verklighet, nämligen det arbete som utförs i den givna organisationen. Det finns välgrundade skäl (och jag skall redovisa dem närmare sedan) att tro att deras uppfattningar på väsentliga punkter skiljer sig från varandra och att detta kan förklara en stor del av de brister som dagens datasystem trots allt uppvisar.

Den övergripande strukturen i denna avhandling är

- Problem
- Metod
- Empiriskt material
- Resultat

Jag börjar med att i kapitel 2, som heter "Bakgrund och problem", ange en ram eller begreppsapparat inom vilken jag håller mig. Jag diskuterar och bestämmer begrepp jag finner väsentliga för det fortsatta resonemanget. I dessa bestämningar föregriper jag kommande problemställningar. Jag börjar med grundläggande begrepp, nämligen *data*, *information* och *kunskap*. Data ser jag som symboler utan mening, medan information är dessa symboler insatta i ett syntaktiskt sammanhang, en utsaga. Kun-

skap är däremot ett betydligt besvärligare begrepp. Resonemanget mynnar emellertid ut i att det finns kunskap som inte går att explicit uttrycka s k *tyst kunskap*. Därefter inför jag en grundläggande distinktion mellan datasystem och informationssystem, nämligen att informationssystem inkluderar användning av datasystem. Detta leder till en diskussion av systemutvecklingsbegreppet och till en formulering av grundläggande föreställningar, som ligger bakom en utredningsmodell.

Dessa appliceras därefter på SIS/RAS-modellen. SIS/RAS står för Svensk Industri-Standard/ Riktlinjer för Administrativ Systemutveckling. Den viktigaste kritiken är att verkligheten ses som neutral och möjlig att beskriva på ett enhetligt sätt i expertspråk. Människan ses som en mekanism för att utföra specialiserade arbetsuppgifter. Som konsekvens av kritiken kommer frågan om maximalt användarinflytande, det som jag senare kommer att kalla *personlig databehandling*, men på detta stadium kallar *interaktiv systemutveckling*. Det leder till en diskussion om förhållandet mellan arbete och datasystem. Jag hävdar att det finns skäl att tro att interaktiv systemutveckling kommer att få positiva effekter arbete och arbetstillfredsställelse. Sedan ger jag en övergripande problemformulering. Bl a konstaterar jag att två olika ansatser är att föredra, en där jag observerar uppkomna effekter och en där jag aktivt deltar i och försöker introducera personlig databehandling i olika organisationer. Båda undersökningarna utgår från användaren och är på individnivå.

I kapitel tre diskuterar jag lämplig metodik. I denna redovisning baserar jag mig i stor utsträckning på Törnebohm (1983) och Heidegger (1981). Eftersom mitt område, "Personlig databehandling" är så pass nytt finns ingen tidigare forskning att direkt bygga på. Därför väljer jag en explorativ ansats och strävar efter bredd och översikt snarare än djup och avgränsning. Jag har tagit upp en mängd problem, men inte i djup och detalj penetrerat något av dem. Därmed inte sagt att djup och detalj i ett senare skede inte är nödvändigt. Semistrukturerade intervjuer föreslås för observation av effekterna och en undervisningsbaserad ansats föreslås för det aktiva deltagandet. Kapitlet avslutas med att jag i tesform sammanfattar mitt problem och dess bakgrund.

I kapitel fyra redovisar jag resultatet från mina fallstudier. Nitton intervjuer från tretton olika företag redovisas. Detta är en relativt detaljerad redogörelse och den baserar sig på arbetsrapporter jag tidigare skrivit. I den sammanfattas en del av det empiriska material som ligger till grund för större delen av de hypoteser jag formulerar i sjätte kapitlet.

I det femte kapitlet gör jag på samma sätt en redovisning av mitt deltagande i och initiering av personlig databehandling. Sammanlagt omtalas åtta fall, från lika många företag (skilda från de tidigare) men fyra av dem är summariskt redovisade medan ett av dem är mycket detaljerat redovisat. Jag försöker här beskriva den process som ledde till de resultat jag presenterar i nästa kapitel.

I sjätte kapitlet redovisar jag resultat på tre olika nivåer, som jag hämtat från Törnebohm. Den första nivån är hypoteser. Eftersom mina undersökningar är explorativa

är det obekräftade hypoteser jag presenterar, men det finns i mitt material saker som tyder på att det skulle kunna förhålla sig så som hypoteserna säger. Den andra nivån är specifika teorifragment. Främst kommer här ett typfall för hur personlig databehandling uppkommer i en organisation och vilka konsekvenser den får. Vidare redovisas en del krav man kan ställa på applikationsgeneratorer plus olika användarroller i samband med personlig databehandling. Den sista nivån är fundamentala teorifragment och där utgör själva förekomsten av personlig information och personlig databehandling det väsentligaste. Det får vissa intressanta teoretiska konsekvenser t ex att en speciell typ av informationssystem, *stödsystem*, introduceras. För dessa verkar den strikta skillnaden mellan utveckling och användning upphävas och således också den strikta rolluppdeleningen mellan användare och systemerare.

Avhandlingen avslutas till sist med en engelsk sammanfattning.

2 BAKGRUND OCH PROBLEM

Systemutveckling innebär "utveckling och konstruktion av datorbaserade informationsbehandlingssystem". Det indikerar att det rör sig om en konstruktionsprocess och att något framställs. Man kan då jämföra med övrigt tekniskt konstruktionsarbete. Checkland (1981, kap 5) pekar på en viktig egenskap i samband med "engineering" (här översatt "tekniskt konstruktionsarbete") nämligen att det ska bli en *ändamålsenlig* och *funktionsduglig* produkt. Han skriver (s 138):

"...there is a desired state, S1, and a present state, S0, and alternative ways of getting from S0 to S1. 'Problemsolving', according to this view, consists of defining S1 and S0 and selecting the best mean of reducing the difference between them. ... *The belief that real-world problems can be formulated in this way is the distinguishing characteristic of all 'hard' systems thinking.*"

Skillnaden mellan de två tillstånden (S1 S0) kan då ses som en beskrivning av "behovet" eller "målet" för konstruktionsprocessen. Detta mål tas av den professionelle ingenjören som givet vid konstruktionsprocessens början. Detta tillvägagångssätt tycks vara möjligt att tillämpa vid systemutveckling eftersom man faktiskt konstruerar en sak, nämligen ett *informationssystem*. Men föremålet som konstrueras är inte ett fysiskt ting utan ett immateriellt och är beroende av mänsklig tolkning för att fungera. Checkland talar om "*human activity systems*" och definierar problem inom ett sådant (ibid, s 155):

"A problem relating to real-world manifestations of human activity systems is a condition characterised by a sense of mismatch, which eludes precise definition, between what is perceived to be actuality and what is perceived might become actuality."

Grundläggande för den fortsatta diskussionen är begreppet "informationssystem", en immateriell produkt, vilken inte kan ses eller beskrivas på samma sätt som ett konkret föremål. För att belysa denna skillnad skall jag inleda problemdiskussionen med att diskutera "information" och "informationssystem".

2.1 Data, Information, Kunskap

Langefors (1966) definierar information såsom "kunskap eller tillskott till kunskap". Lägga märke till det lilla ordet "tillskott". Det förutsätter nämligen att det sker något i samband med informationen. På så sätt inför Langefors en processaspekt direkt i samband med definitionen av ett av de mest grundläggande begreppen. Än tydligare framgår det i sambandet mellan information och data, där Langefors (ibid s 197) anger information som en funktion av *data*, en *struktur* gentemot vilken data tolkas samt *tiden* som står till förfogande för tolkning. Men som i alla analytiska definitioner blir bara problemet skjutet framåt ytterligare ett steg. Langefors ger nämligen inte svar på vad "kunskap" är. Detta begrepp tas för givet. Att här uttömmande behandla kunskapsbegreppet för långt utanför området för denna avhandling. Det behandlas inom epistemologin eller kunskapsteorin i filosofin. Speciellt franska och tyska filosofer har ägnat sig åt kunskapsteoretiska frågor. Habermas engelska översättare ger en ledtråd till varför det förhåller sig så (Habermas, 1978, sid 319). I det tyska språket finns nämligen två olika ord för kunskap, "Erkenntnis" och "Wissen", likaså i franskan: "connaissance" och "savoir". Det gör att diskussionen kan föras mer nyanserat. Men även i Sverige och då speciellt under senare år har det förts en bitvis ganska intensiv kunskapsteoretisk debatt (T ex Israel 1979, 1980 och 1982).

I detta sammanhang måste jag också ta upp något om traditionell formell logik och dess sätt att se på "data" och "kunskap". Enligt denna finns ett antal sanna utsagor om sakförhållanden i tillvaron. Det är påståenden av typen "Det här bordet är rött" Sanningshalten i dessa utsagor kan endast avgöras genom empiriska observationer. Dessa utsagor kallas ibland *syntetiska utsagor*. Men om vi istället ser på utsagan "Rött är en färg" så är dess sanningshalt inte beroende av sakförhållanden utan avgörs endast genom sammanhanget med andra utsagor. Utsagan är en slags definition. Sådana utsagor kallas ibland *analytiska utsagor*. Kant (1924) definierar dem sålunda (sid 59 o 61):

"In allen Urteilen, worinnen das Verhältnis eines Subjekts zum Prädikat gedacht wird, (wenn ich nur die bejahenden erwäge: denn auf die verneinenden ist nachher die Anwendung leicht) ist dieses Verhältnis auf zweierlei Art möglich. Entweder das Prädikat B gehört zum Subjekt A als etwas, was in diesem Begriffe A (versteckterweise) enthalten ist; oder B liegt ganz ausser dem begriff A, ob es zwar mit demselben in Verknüpfung steht. Im ersten Fall nenne ich das Urteil analytisch, in dem anderen syntetisch."

Om dessa begrepp överförs till informationsbehandling (Flensburg 1984f) är utsagor om delar i datasystemet (informationsbehandlingssystemet), såsom t ex filbeskrivningar snarlika analytiska utsagor ty de hänför sig helt och hållet till datasystemet. De utsagor som *produceras av* datasystemet, t ex om mängden tretumsspik i lager, är då snarlika syntetiska utsagor, ty de ligger utanför datasystemet. Sanningshalten kan

bara avgöras genom att fysiskt gå och titta i lagret. Detta verkar utgöra en viktig skillnad mellan olika typer av information i samband med datasystem.

För att kunna tillämpa traditionell formell logik vid konstruktion av datasystem är motsägelsefrihet hos de uppställda förutsättningarna en nödvändig, men ej tillräcklig förutsättning för att man ska kunna lita på de utsagor datasystemet producerar. Förutsättningarna måste också vara i syntetisk mening sanna. Att avgöra detta är ett icke-trivialt problem, som t ex Churchman (1971) tar upp.

En annan intressant aspekt på kunskapsbegreppet tar Polanyi (1966) upp med begreppet "Tacit Knowledge". Han menar att det finns typer av kunskap, som vi inte är medvetna om eller åtminstone inte kan explicit redogöra för. Vi kan t ex inte beskriva hur ett visst ansikte ser ut, men däremot känner vi igen det, när vi ser det. Samma gäller för många av våra vardagliga företeelser, t ex bord som vi såg i inledningen. Likande tankar är Ryle (1949) inne på när han skiljer mellan *knowing that* och *knowing how*. Inom informationsbehandlingen har i Sverige främst Göranson, Nordenstam och Josefsson tagit upp begreppen "förtrogenhetskunskap" och "tyst kunskap" (Göranson 1983a och 1983b, Josefsson 1985). Internationellt var nog Weizenbaum (1976, 1985) den som först medvetandegjorde dessa aspekter inom informationsbehandling och då främst inom området "artificiell intelligens".

Avslutningsvis visar de tre begreppen *data*, *information* och *kunskap* på olika aspekter av erfarenheter i tillvaron. Här ska jag i ett litet exempel visa hur dessa kan kombineras.

"Data" är tecken utan någon innebörd i sig. Exempel på data är: "7", "Focularin", "31" "11 1985", "Härlanda". Om vi sätter ihop en samling data till en meningsfull utsaga blir det "information". Om vi i exemplet ovan säger: "Det satt 31.11 1985, 7 stycken focaliner internerade på Härlanda fängelse" är detta exempel på information. Om "information" vore detsamma som "kunskap", skulle alltså läsaren få kunskap av meddelandet ovan. Vilken kunskap då, kan man fråga sig? Jag förmodar att läsaren inte vet vad en focalarin är och därför inte är riktigt lika säker på meddelandets egentliga innebörd som om det hade stått t ex "knarklangare" istället för "foculariner". Antag, att meddelandet ovan finns i Svenska Dagbladet under rubriken "Officiella meddelanden". Då kan läsaren dra åtminstone två slutsatser. Det faktum att det den 31.11 1985 fanns 7 focaliner i Härlanda fängelse. Detta kallar Heidegger tillvarons *fakticitet* (Heidegger 1981, sid 81):

"Tillvaron förstår sitt innerligast egna vara såsom en viss "faktisk förhandenvaro" (im Sinne eines gewissen "tatsächlichen Vorhandenseins"). Ändå är den egna tillvarons faktums "faktiskhet" ontologiskt sett i grunden skiljaktlig från exempelvis någon bergarts faktiska förekomst. Den faktiskhet som utmärker tillvaron såsom factum (Die Tatsächlichkeit des Faktums Dasein) och i vilken egenskap varje tillvaro är till, kallar vi dess fakticitet (Faktizität)."

Men förutom detta i någon mening "uppenbara" faktum, finns ett annat faktum, nämligen att foculariner är något som kan sitta i fängelse. Detta senare är en förutsättning för det första faktumet. Dessa båda fakta kan tillsammans möjligen förmå läsaren att ta reda på vad "focularin" egentligen är. Först då kan man, enligt min mening, tala om erhållande av kunskap.

Men ponera nu för ett ögonblick att vi hade läst meddelandet ovan i en annan publikation, nämligen i tidskriften MAD. Visst hade vi då uppfattat det på ett helt annat sätt! Förmodligen hade vi ryckt på axlarna och trott att det var ett internt redaktions-skämt, obegripligt för icke invigda.

Nu *finns* det emellertid något som heter foculariner. En focularin är en medlem i focularer, som närmast kan karakteriseras som en katolsk lekmannarörelse. Det finns gissningsvis ungefär 10 st totalt i hela Sverige. Vilken kunskap får läsaren nu då? Såvitt jag ser finns utrymme för tre olika tolkningar:

1. *Focularerörelsen har blivit en militant lagbrytare, vilket är osannolikt.*
2. *Vi har inte någon religionsfrihet här i landet, vilket också är osannolikt.*
3. *Meddelandet är fel*

Den sista tolkningen är den mest sannolika och den uppmärksamme läsaren har redan noterat ytterligare en ledtråd till detta, nämligen att november bara har 30 dagar! Den kunskap vi får av denna information är således paradoxalt nog att informationen är felaktig! Detta exempel tycker jag visar att det finns en skillnad mellan kunskap och information och att samma information kan ge olika kunskap i skilda sammanhang. Det sammanhang i vilket informationen framstår är alltså mycket väsentligt.

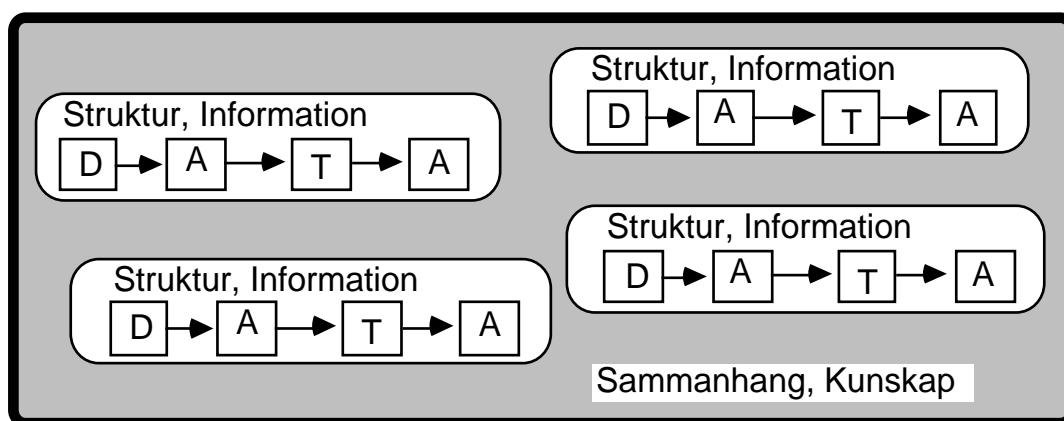


Fig 2.1 Data, information och kunskap

Sammanfattningsvis betraktar jag data som element i en struktur. Denna struktur blir information (fig 2.1). Strukturerna finns i ett sammanhang och först i detta sammanhang kan de tolkas och bli kunskap.

Till information och kunskap vill jag också koppla någon form av handlingsbegrepp. Häruti följer jag den sene Wittgenstein (1976), Churchman (1971) och tangerar språkaktsteorin (Austin 1962, Searle 1969, Goldkuhl & Lyytinen 1982, Goldkuhl 1984). Ett exempel på detta hände vid en av mina intervjuer med en inköpschef. Han berättade om en förhandling där han hade plottat index mot en viss leverantörs prisutveckling (fig 2.2). När leverantören fick se denna kurva gick han omedelbart med på att sänka priset. På detta tjänade inköpschefen mer än väl in kostnaden för hela datasystemet. Till grund för bilden ligger en uppsättning data, hämtade från något eller några register. Dessa data presenteras i en struktur, i det här fallet ett koordinatsystem. Bilden kan betraktas som information och den kunskap som förmedlas är att leverantörens priser är för höga. Det resulterar i att leverantören sänker dem.

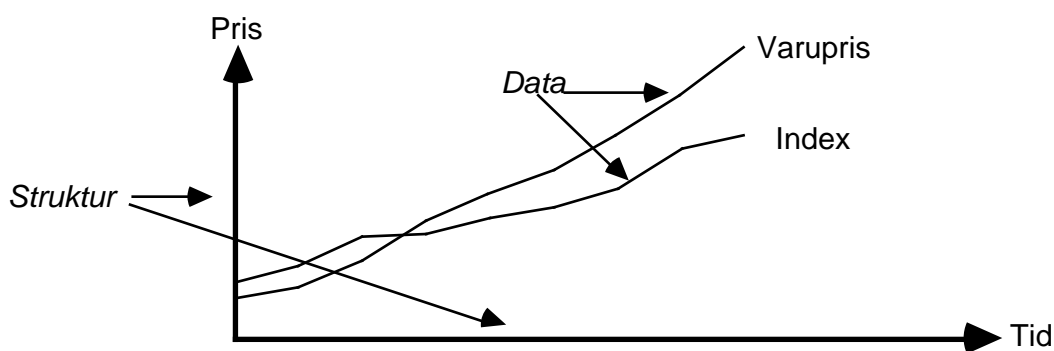


Fig 2.2 Inköpschefens kurva

Det finns sagt mycket mer om "data", "information" och "kunskap". Ivanov (1972) påpekar att varje syntetisk utsaga har ett mätfel, som man måste ta hänsyn till, Ciborra (1982) påpekar att administrativa utsagor kan vara avsiktligt färgade (opportunistiska). Här har jag endast refererat till en bråkdel och vill endast förmedla en insikt om att inget av begreppen är så självklart som vi ibland tar dem för.

Det väsentliga budskapet i detta avsnitt är informationens sammanhang och vad som görs med den.. Vid systemutveckling framställs informationssystem och jag ska nu närmare diskutera detta begrepp.

2.2 Datasystem, informationssystem

Enligt svenska akademins ordlista är "information" som "upplysning", "underrättelse". "System" betyder enligt samma källa detsamma "ordnat helt", "planmässighet", "reda och ordning". "Planmässigt ordnade upplysningar" förefaller vara en ganska ordagrann översättning. Detta ger ingen indikation på någon svårighet att använda strikt formella synsätt. Men jag ska här nedan gå genom några vanliga definitioner av detta begrepp, vilka ger indikationer i denna riktning.

Ett vanligt missbruk är att man använder termen "informationssystem" för att beteckna "datoriserat informationsbehandlingssystem". Detta har bl a Nissen (1980) kritiserat. Han föreslår termen "partiellt informationssystem" som alternativ. Denna förefaller något tungrodd varför jag vill föreslå ordet "datasystem" för att beteckna själva datorn, programmen, data, datastrukturen och bruksanvisningen. Alla dessa komponenter är väsentliga i den datoriserade delen av ett informationssystem. Termen "informationssystem" reserveras då för datasystem plus användare av datasystem (fig 2.3).

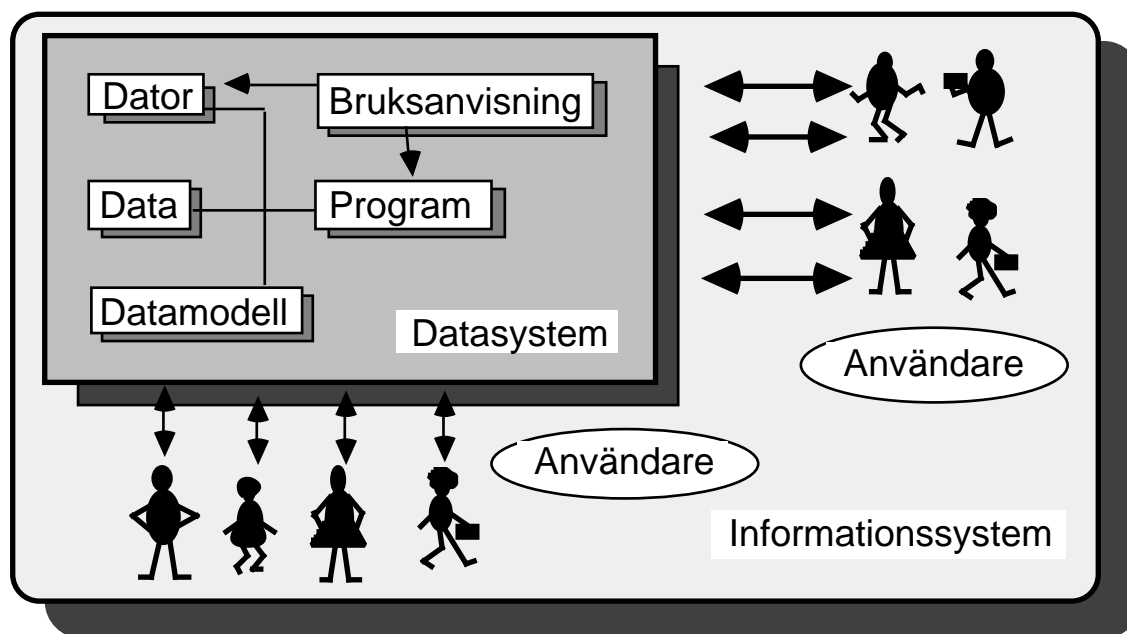


Fig 2.3 Samband mellan informationsoch datasystem

Informationssystem blir här ganska snävt definierat i och med att endast kommunikation med datasystemet inräknas däri. Kommunikation mellan människor räknas i denna definition inte till informationssystemet. Jag uppfattar att internationell praxis är så här. Givetvis innebär denna definition inte att jag förnekar mellanmänsklig kommunikation eller dess betydelse för att kunna använda informationssystem.

Tvärtom gör definitionen det möjligt att synliggöra den mellanmännsliga kommunikationen.

En annan intressant ansats har Nissen (1985b) tagit när han, baserad på Toulmin (1958), föreslår att informationssystem ska ses som ett specialfall av argumentation. Detta synliggör både den mellanmännsliga kommunikationen och ett av syftena med densamma, nämligen att övertyga någon annan om det berättigade i ett visst handlingsätt.

Det kan vara problematiskt att betrakta människor som del i ett system. Det kan medföra en objektiverad människosyn. För att undvika missförstånd vill jag påpeka att detta är inte min avsikt. Skillnaden mellan de båda typerna av system är att när jag talar om informationssystem, så menar jag även människorna som använder det, men när jag talar om datasystem, talar jag enbart om artefakter. Orsaken till detta är att jag anser att det krävs väsentligt olika vetenskapliga metoder för undersökningar av de olika sorternas system. För att konstruera ett datasystem, kan man använda formell metodik, eftersom det rör sig om helt formella artefakter. Men det går inte att bortse från datasystemets användande vid dess konstruktion. Som jag hävdade i avsnittet om information är det inte givet att det går att tillämpa enbart formella metoder vid konstruktionen. En väsentlig fråga är att avgöra när sådana metoder är möjliga att tillämpa och när de inte är möjliga att tillämpa. Jag kommer då in på vad systemutveckling innebär.

2.3 Systemutveckling och -användning

Systemering och systemutveckling använder jag synonymt. Systemering definierar jag sålunda (Flensburg 1981b):

Systemering är en process i vilken några personer kommer överens om hur en dator ska användas i en organisations administrativa verksamhet.

Möjligen kan man i denna definition reagera mot ordet "dator" och hävda att det behöver inte nödvändigtvis vara datorer inblandade. Principiellt är det rätt och "en dator" kan utbytas mot "strikt formella metoder", men vare sig praktiskt eller principiellt har detta tillägg någon betydelse. Systemering bedrivs enligt någon *systemutvecklingsmodell*, vilket enkelt uttryckt kan sägas vara råd, anvisningar och redskap för hur systemering bör bedrivas. Det finns olika typer av sådana modeller, men de vanligaste tillhör "Life cycle" typen (Olerup, 1982). I en sådan beskrivs systemering i ett antal sekventiella faser (t ex Semprevivo 1982 eller SIS 1979). Fasernas antal vari-

erar, likaså deras beteckningar. Här har jag valt SIS/RAS (SIS 145, 1979) som typexempel:

Problemstudie Målstudie	Analys
Informationsstudie Behandlingsstudie Systemstudie	Design
Detaljstudie Detaljutförande Införande	Realisering
Efterstudie	

Fig 2.4 Faserna i SIS/RAS

Gemensamt för samtliga varianter är en tredelning i "analys", "design" och "realisering" (Lundeberg m fl, 1978). Samma författare har en liknelse med ett husbygge. Analys innebär att fastställa av de krav man ställer på huset (Antal våningar, storlek, standard etc). Design innebär upprättande av ritningarna och realisering är själva byggandet.

Detta är inte platsen att fördjupa sig i en analys av olika systemutvecklingsmodeller. Detta har gjorts av andra (t ex Brandt & Johansson 1980, Brandt et al 1978, Olerup 1982, 1985, Sandström & Wormell 1980, Swanson 1976). Jag tar den svenska SIS/RAS som exempel, men menar att mina slutsatser gäller för andra modeller av samma typ, men däremot inte för andra typer av modeller.

Ibland har man en förstudie som är en ytlig genomgång av de fem första faserna, i syfte att undersöka om det är lönsamt att fullfölja systemeringen. Om det bedöms lönsamt genomförs en huvudstudie, som är en komplettering av förstudien i de fyra första faserna och ett genomförande av de fyra sista. Någon gång under någon av faserna framställs ett mycket viktigt dokument, *kravspecifikationen*. Detta dokument beskriver systemets yttre egenskaper, dvs vilka utdata som ska produceras, vilken form och vilken aktualitet de ska ha samt vilka som är mottagare. Där beskrivs också mål, både för utvecklingen och för dess produkt, kostnaden för utvecklingen och ibland även beräknade intäkter. Med kravspecifikationen skall systemutvecklaren *i princip* kunna sätta sig vid sitt skrivbord och göra systemet.

Detta kräver ett antal förutsättningar. Verksamheten måste vara så pass stabil att det är möjligt att formulera mål och informationsbehov för åtminstone några år framåt.

Detta gäller såväl produktion som administrativ verksamhet. Eftersom det är ett mycket kostsamt sätt att göra datasystem på, måste också den beräknade intäkten vara stor. Men företagen är idag ofta hårt rationaliserade och därför blir inte heller förväntade intäkter överväldigande. Avskrivningstiden för nya system är dessutom kort, 3-5 år enligt Bosrup & Holmberg (1984). Risken för konservering och stelbenthet är under sådana förhållanden stor. Detta visar sig i stora underhållskostnader. Mellan 50 och 80% av dataavdelningens kapacitet krävs för underhåll (Martin 1982).

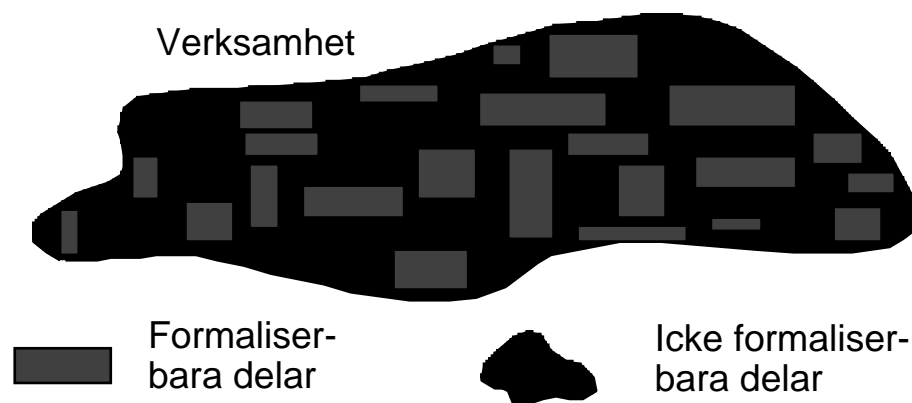


Fig 2.5 Formalisering och icke-formalisering av verksamhet

Förutom att verksamheten måste vara stabil, krävs också att den är formaliserbar för att kunna utföras i ett datasystem. Men det är den i regel inte. Tvärtom: de flesta verksamheter inte går att fullständigt formalisera. Lösningen anses då vara att skapa ett slags symbios mellan dator och människa, där datorn gör det som den är bra på och människan det som hon är bra på. Verksamheten blir således uppdelad i formaliserbara delar och icke formaliserbara delar. Dessa måste emellertid tillsammans forma en ändamålsenlig helhet (Fig 2.5).

Systemutveckling bedrivs alltid i projektform. Det kan finnas olika sorters projektgrupper beroende på projektets storlek. Om projektet är stort kan man t ex ha olika sorters arbetsgrupper för olika delprojekt, man kan ha en eller flera referensgrupper och slutligen kan man ha en beslutsgrupp, som ska fatta de formella besluten. Om projektet är litet finns kanske alla dessa funktionerna i en och samma grupp. Det är också viktigt att det finns representanter för de som senare ska använda systemet i alla grupperna. Viktigast är det naturligtvis i arbetsgruppen, där det mesta av arbetet görs. Men det är inte alltid det fungerar med dessa representanter (Neergaard 1977).

Denna metodik har i huvudsak varit oförändrad sedan de stora satsvisa systemen på 1960-talet. Den baserar sig på en föreställning om att datasystemet görs ordentligt och grundligt en gång för alla. Men det finns även andra och mera grundläggande föreställningar bakom den traditionella systemutvecklingsmodellen. Som exempel ska jag här nedan göra en analys av de grundläggande föreställningar som ligger bakom SIS/RAS modellen. För att kunna göra detta måste jag emellertid först diskutera

"grundläggande föreställningar". Följande avsnitt baserar sig på ett kompendium (Flensburg 1980a) jag skrivit tidigare. Det är här väsentligt förkortat.

2.3.1 Grundläggande föreställningar

Forskaren studerar ett forskningsområde. Föreställningar om forskningens grundelement, som forskaren baserar sig på utgörs av en specifik *vetenskapsteori*. Mellan dessa råder vissa beroendeförhållanden (Fig 2.6).

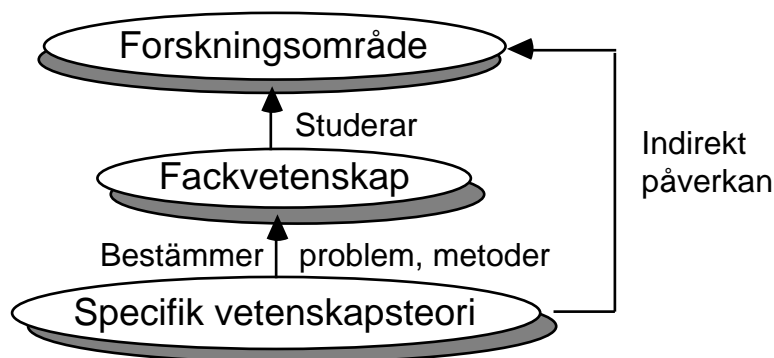


Fig 2.6 Vetenskapens grundelement på grov nivå

Varje fackvetenskap bestämmer ett forskningsområde. Detta kan vara olika för olika vetenskaper. En forskare i t ex mikrobiologi studerar helt andra fenomen än en forskare i nationalekonomi, men de kan också vara delvis överlappande. T ex kan bruk av datasystem i organisationer studeras både inom informationsbehandling-ADB, företagsekonomi, sociologi och psykologi. Varje fackvetenskap har givetvis sitt eget sätt att bedriva dessa studier och ställer sina egna specifika frågor. Fackvetenskapen baserar sig på en specifik vetenskapsteori, som bestämmer hur denna vetenskap ska bedrivas, vilka problem som ska angripas och vilka metoder man därvid ska använda. En och samma specifika vetenskapsteori kan också influera flera olika fackvetenskaper (fig 2.7). Det jag vill framhålla i fortsättningen är att det finns någon sorts bestämning, från vetenskapsteori till fackvetenskap och från fackvetenskap till forskningsområde. För att förenkla figurerna gör jag dem därför i fortsättningen i pyramidform.

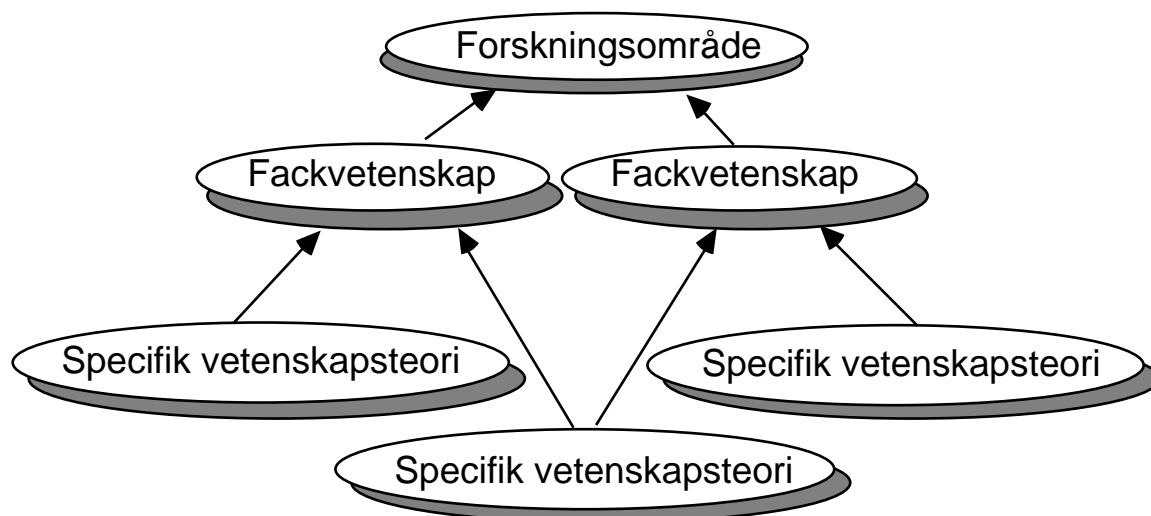


Fig 2.7 Förhållanden mellan territorium/vetenskap samt vetenskap/vetenskapsteori

Analoga förhållanden råder inom systemutveckling (fig 2.8). I systemutvecklingsprocessen bestäms hur det färdiga systemet ska användas. Denna process i sin tur bestäms av systemutvecklingsmodellen. På så sätt får vi tre olika nivåer, där användningen är den egentliga, pragmatiska nivån, systemutvecklingsprocessen är en metanivå och systemutvecklingsmodell utgör en metametanivå i förhållande till datasystemets användning.

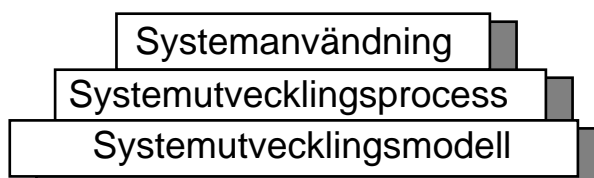


Fig 2.8 Systemutvecklingens grundelement

Systemutvecklingsmodellen bestämmer hur det färdiga systemet skall användas. Om en hierarkisk metod, som t ex HIPO (Hierarchy plus Input-Process-Output), används resulterar det i strikt hierarkisk användning. HIPO utvecklades ju av IBM som stöd åt sitt IMS databassystem, vilket var hierarkiskt uppbyggt. På något vis genomsyrar systemutvecklingsmodellens ideal samtliga nivåer ovanför. Därför förefaller det intressant att undersöka vilka dessa ideal är för någon välkänd modell. Jag har därvid valt SIS/RAS.

Systemutvecklingsmodellen är egentligen inte basnivå, för den i sin tur bestäms av en organisations och samhällsmodell. Historiskt sett har informationsbehandling sitt ursprung i företagsekonomi och då speciellt en koppling mellan organisationsteori och beslutsteori/operationsanalys (Ivanov 1984).

Vi kan nu tillämpa samma resonemang på forskning om systemutveckling. Forskningsområdet är användning och utveckling av informationssystem (fig 2.9). Det innefattar tre olika nivåer, nämligen *användning* av system, *utveckling* av system och *modeller* för system. "System" kan här betyda både datasystem och informationssystem. Enligt detta synsätt är användningen av systemen inkluderad i forskningsområdet. Det är alltså helt legitimt att inom ämnet studera hur informations- och datasystem används. Motivet är att datasystemens syfte måste rimligen vara att bli använda och följaktligen är användningen viktig att studera i forskningssammanhang.

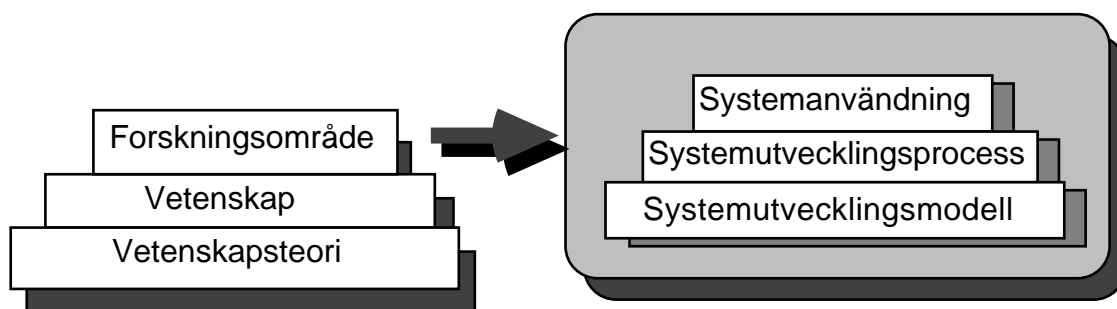


Fig 2.9 Området för forskning kring systemutveckling

Forskningsområdet för forskning kring systemutveckling omfattar samma eller liknande nivåindelning som vid vanlig vetenskap enligt fig 2.6. Det finns en strukturell likhet och därför kan vetenskapsteoretiska resonemang även vara tillämpbara innanför systemutveckling.

Jag ska nu försöka formulera grundläggande föreställningar i vetenskapliga sammanhang. Detta har gjorts av bl a Lindholm (1979, 1985). Han har koncentrerat sig på vad som ligger till grund för vetenskapsteorin. Han menar att *kunskapsfilosofi* (epistemologi), *naturfilosofi* (ontologi) och *moralfilosofi* ligger till grund för vetenskapsteoretiska överväganden. Kunskapsfilosofin omfattar föreställningar om vad kunskap är, hur kunskap genereras m m. Det finns i huvudsak två skilda kunskapsfilosofier, nämligen *holism* där man anser kunskap som hel och odelbar, samt *atomism* där man anser att kunskap är möjlig att dela upp i delar vilka inte kan delas upp mera. Ett sorts mellanting skulle kunna vara den generella systemteorin där kunskapens anses vara möjlig att dela upp, men inte i odelbara delar. Naturfilosofin omfattar föreställningar om verkligheten, hur den är beskaffad och vad som bör studeras inom densamma. Moralfilosofin slutligen handlar om etiska förhållanden, om hur vi betraktar våra medmänniskor, om vad som är "gott" och "ont", om forskarens sätt att relatera sig till sitt undersökningsområde och till samhället. Törnebohm (1975) påpekar också vikten av etiska överväganden och han skiljer dessutom på en intern etik, som berör förhållanden inom forskarsamhället och en extern etik, som berör förhållandet till forskningsområdet. Vidare tar Törnebohm upp estetiska överväganden som en del i forskningsprocessen.

Andra vetenskapsteoretiker (t ex Israel 1980) föreslår metodologi istället för moral-filosofi. Diskussioner om dessa grundläggande synsätt kallar Törnebohm *tematiska diskussioner*. Kuhn (1981) har observerat att sådana tematiska diskussioner tenderar att återkomma med jämna mellanrum (fig 2.10). Däremellan finns perioder då man knappast för några tematiska diskussioner utan enbart ägnar sig åt problem i det aktuella forskningsområdet. Denna typ av vetenskap kallar Kuhn *normalvetenskap*. Men Kuhn observerade att perioder med intensiva tematiska diskussionerna ofta ledde till skiftningar i de grundläggande synsättet på vetenskapen, s k *paradigmskiften*. Detta skede kallar han revolutionära skedet. Under ett sådant skede strider ofta flera vetenskapliga skolor mot varandra.

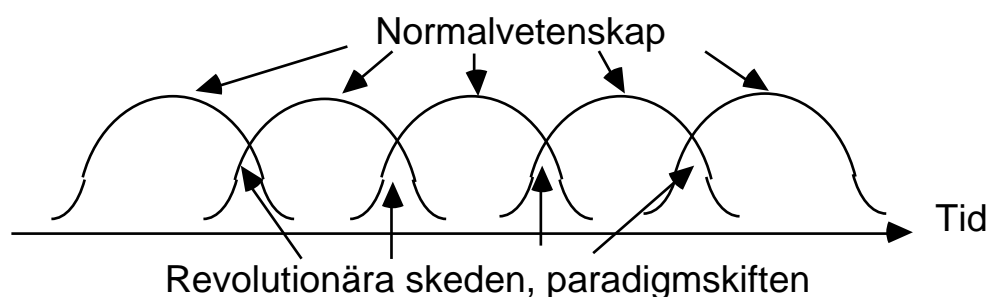


Fig 2.10 Vetenskapens historiska förlopp enligt Kuhn

Begreppet *paradigm* används av Kuhn i betydelsen "metod" och "forskningsproblem". Paradigmet anger dels vilka problem som kan angripas vetenskapligt, dels med vilka metoder det lämpligen görs. En slags institutionella regler med andra ord. Detta diskuteras närmare i kapitel 3. Här ska jag närmare diskutera vad som ingår i ett paradigm inom systemutvecklingsforskning. Forskningsområdet, består enligt tidigare, av tre nivåer. Till denna lägger jag, av historiska skäl, en fjärde, organisations- och samhällsteori, om vilken jag tidigare talade (Ivanov 1984). Om vi använder Lindholms begrepp gäller det alltså att finna vilka föreställningar om kunskap, verklighet och moral som ligger bakom paradigmerna för forskning om systemutveckling. Men föreställningarna om verklighet, kunskap och moral måste hänga ihop och vara inbördes konsistenta. Det går t ex inte att samtidigt föreställa sig att kunskap är odelbar och att verkligheten är möjlig att dela upp i separata av varandra oberoende delar. Det är också svårt att samtidigt föreställa sig verklighetens objektiva uppdelbarhet och att människan (inkl forskaren och de människor som finns i forskningsområdet) är en fri och självständig varelse, som skapar och påverkar sin egen sociala omgivning. Kunskaps- och verklighetsuppfattning kan således inte ses som separerade från värderingar och moraliskt/ etiska överväganden. Dessa beroenden mellan de olika delarna bestäms av en världsåskådning (fig 2.11).

Figuren är naturligtvis en förenkling. Beroendeförhållandena är betydligt mera intrikata än vad min figur visar. Det är sålunda endast delar av organisations och samhällsteorin som är relevanta i detta sammanhang.

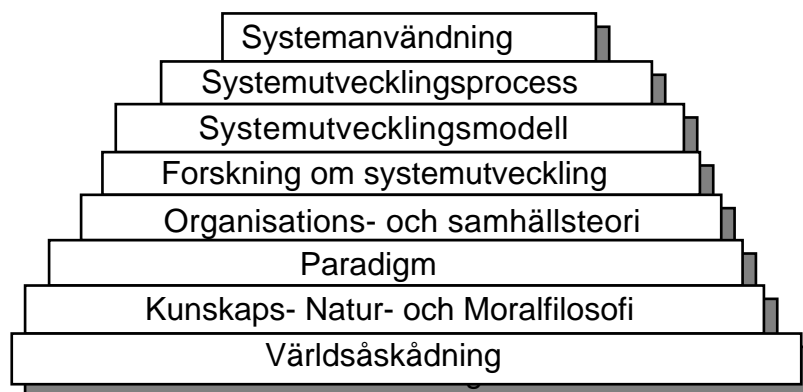


Fig 2.11 Den vetenskapliga pyramiden för forskning om systemutveckling

Vi kan nu operationalisera denna föreställningsram i ett antal specifika grundföreställningar och grundsyrer. Grundföreställningarna är:

- Föreställning om verkligheten (ontologin)
- Föreställning om kunskapen (epistemologin)
- Etiska föreställningar (moralfilosofi)

Grundföreställningarna ingår i den näst nedersta nivån i pyramiden i fig 2.11. Det är därför jag anser dem vara mer grundläggande än grundsyrerna, som ingår i högre nivåer. Detta innebär inte att grundsyrerna skulle vara mindre viktiga, tvärtom är det nog så att de är lättare att föreställa sig och få grepp om. Därigenom blir de viktigare i och med att de blir synligare och mera begripligare för oss. Grundsyrerna är följande:

- Människosynen
- Samhällssynen
- Organisationssynen
- Historiesynen

Människo- och samhällssynen har starka inslag av moralfilosofi, medan organisations- och historiesynen enligt min mening har mindre. Människosynen omfattar bl a föreställningar om människans "natur", den behandlar frågan om hur vi ser på förhållandet mellan forskaren och de människor som ingår i forskningsområdet men också om hur systemutvecklare ser på användare och vice versa. Forskaren har i sitt forskningsområde både syn på systemutvecklare, syn på användare och syn på systemutvecklarens syn på användaren och vice versa. I forskarens forskningsområde ingår således även systemutvecklarens människosyn (Hedberg & Mumford 1975, Gingras & McLean 1984). Därför bör forskarutbildningen även omfatta ett visst mått av psykologisk utbildning.

Samhällssynen är på samma sätt föreställningar om samhället och om hur grupper av människor agerar med varandra. Hit kan också forskningens roll i samhället föras. I aktionsforskningen är detta en central fråga. Inom informationsbehandling-ADB har sådana områden som "personlig integritet", "datalagen" och "informationssamhället" (t ex Masuda, 1984) starka samhällseliga aspekter.

Inom fackföreningsrörelsen har det framförts två olika perspektiv på samhället och på organisationer nämligen *konfliktperspektivet* och *harmoniperspektivet* (Sandberg et al 1978). Enligt konfliktperspektivet är samhället och organisationerna ett fält för konflikter. Kampen står mellan arbetsgivare och arbetstagare. I denna kamp anser man att den traditionelle systemutvecklaren tar arbetsgivarens parti och hjälper denne att styra och kontrollera arbetstagarna. En utredare är enligt konfliktperspektivet inte neutral utan måste alltid ta någondera partens parti. Enligt harmoniperspektivet finns det i grunden inga motsättningar utan dessa är skenmotsättningar beroende på bristande information. I harmoniperspektivet uppfattas organisationen som ett fält för gemensamma intressen, en strävan mot gemensamma mål. Utredaren är, enligt harmoniperspektivet en objektiv och neutral person som fullgör ett uppdrag enligt en given specifikation. Detta kan ske utan att utredaren behöver ta ställning för någon speciell grupp.

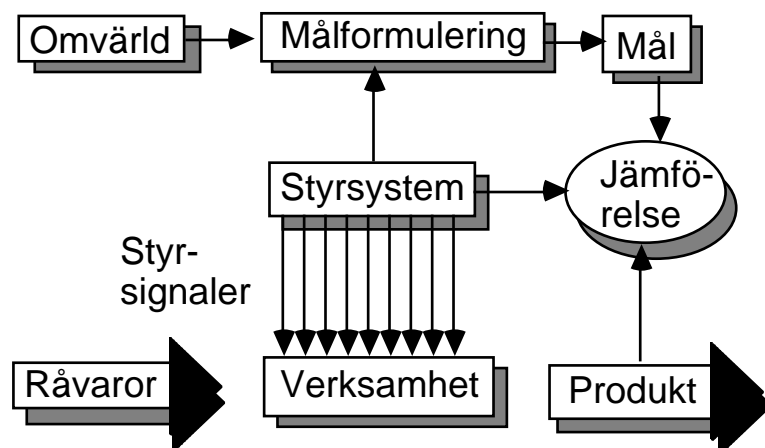


Fig 2.12 Organisationen som ett styrt system

I samband med systemutveckling är det vanligt att se organisationen som ett styrt system (Rehman 1964, Langefors 1968). Detta bygger på ett harmoniperspektiv. Enligt styrsystemsytset (fig 2.12) producerar en verksamhet vissa produkter eller tjänster. Det finns en målsättning för detta i form av en viss kvantitet av och en viss kvalitet på den aktuella produkten/tjänsten. Måluppfyllnaden mäts och rapporteras till ett styrsystem. Beroende på avvikelserna reagerar styrsystemet genom att sända vissa styrsignaler till verksamheten.

En avvikelse från målet kan bero på att det är någon störning i produktionen, men den kan också bero på att målet är felaktigt. Främsta indikationen på det senare är

impulser från omvärlden. Insikten om att målet är felaktigt genererar ny målformulering. Denna insikt uppkommer ofta tack vare en oformaliserbar och institutionell organisationskultur (Berger & Luckmann, 1968).

I styrsystemsinsättet kan man urskilja en del ontologiska och epistemologiska ställningstaganden. Att utfallet av produktionen går att mäta innebär att det finns något att mäta. Mätning innebär jämförelse med någon standardenhet. Det implicerar en atomistisk föreställning om i varje fall just denna del av verkligheten. Det implicerar också att mätningen går att göra på samma sätt oberoende av vem som gör den. Att detta mått går att jämföra med ett annat, tidigare uppställt mått implicerar att även kunskapen anses vara delbar och objektiv.

2.3.2 Granskning av SIS/RAS-modellen

Följande granskning baserar sig på SIS handbok nr 145 (1979), "Riktlinjer för Administrativ Systemutveckling". Systemutvecklingsmodellen (SIS/RAS) som beskrivs är i huvudsak en projektstyrningsmodell och inte en analysmodell.

I förordet redogörs för hur modellen tillkommit. Det framgår bl a att den aktuella handboken är en omarbetning av en tidigare handbok (SIS 113 från 1970). Men inga principiella ändringar har gjorts. SIS 145 skall vara "en bok som ger råd till hjälp vid planering av utvecklingsprojekt, men boken får inte tolkas som en standardiserad utvecklingsmetod". På sidorna 5 och 6 står följande:

"All verksamhet styrs av beslut och ju bättre underlagen är och ju lättare de kan förstås av beslutsfattarna desto större är sannolikheten att konsekvenserna kan förutses. Den enhetlighet i strukturering och dokumentering av systemutveckling som beskrivs i dessa riktlinjer bör verksamt underlätta förståelsen av de beslutsunderlag som krävs under arbetets gång och således vara till hjälp för de personer som skall ha medinflytande så som kommit till uttryck i de senaste årens lagar, avtal och förordningar."

Detta stycke ger uttryck för följande värderingar:

- *Verksamheten styrs av beslut*
- *Beslut fattas rationellt av beslutsfattare*
- *Konsekvenser av beslut kan förutsägas, om än med möda*
- *Om en sak beskrivs på ett enhetligt och strukturerat sätt kan alla begripa det.*

Vidare kan man dra slutsatsen att SIS/RAS ser organisationen som ett styrt system och att man anlägger ett harmoniperspektiv. Man verkar också se verklighet och kunskap som uppdelbara och oberoende av iakttagaren.

Arbetet med SIS/RAS började 1969 och enligt förordet har i princip inget förändrats ifråga om systemutvecklingsmodellen. Vid slutet av 60-talet och början av 70-talet gällde emellertid andra målsättningar och förutsättningar för datorisering:

- *Centralisering och storskaleekonomi var önskvärda*
- *Datasystem var stora, centrala och i satsvis form*
- *Datasystem gjordes av experter och bedömdes av experter*
- *Datorn var dyr och måste utnyttjas effektivt*
- *Problemen var i huvudsak tekniska, t ex litet primärminne.*
- *Medbestämmande var okänt. Däremot fanns "motstånd mot förändring"*
- *Idén om det totalstyrda företaget stod på sin höjdpunkt*

Det är alltså på dessa i stort sett enbart tekniska förutsättningar som SIS/RAS baserar sig. Men under tiden mellan omarbetningen har det skett en hel del förändringar i samhället och med datatekniken. Detta tas upp i inledningen:

"De senaste årens förändrade syn på verksamheten inom såväl de privata, statliga som kommunala sektorerna har starkt påverkat kraven på deras administration.

Bakgrunden är bl a den enskildes krav på medverkan för att utforma den egna arbetssituationen och en ny syn på hur denna påverkas av databehandling. Den nya synen, baserad på utbildning, erfarenhet och allmän informationsspridning har tillsammans med ökad tillgång till avancerade tekniska hjälpmedel skapat ett ökat engagemang i datafrågor...

Dagens teknik, som möjliggör snabb tillgång till stora datamängder, har också successivt effektiviserat administrationen. Utnyttjandet av dessa möjligheter beror dock på vår förmåga att överblicka och rätt förstå de administrativa samband som råder inom företag, myndigheter, organisationer, institutioner och verk...

Enhetlighet i arbetsmetodik och dokumentering är betydelsefull för att beskriva och förstå de administrativa sambanden. Dessutom underlättar denna enhetlighet såväl utvecklingsarbete, allmänna diskussioner som samordning av utbildning både i skolor och företag"

Det finns alltså en ny syn på databehandling, påverkad av medbestämmandelagen och samhällets övriga utveckling. Harmoniperspektivet framträder klart. Genom ökad och mera lättförståelig information överbryggas eventuella motsättningar. Utbildning, erfarenhet och allmän informationsspridning är hörnstenar i SIS/RAS' medbestämmande. Tekniken ger möjlighet till effektiv administration, men den kräver enhetlighet i arbetsmetodik och ordentlig dokumentering. Överhuvudtaget betonar man enhetlighet väldigt mycket. SIS/RAS framstår som en för alla gemensam terminologi i vilken man ska diskutera systemutveckling och administration. Att denna enhetlighet får en styrande effekt och att den leder till att endast en viss typ av datasystem, styrsystem, utvecklas sägs det inget om.

Värderingsfrihet, objektivitet och entydighet grundad på enhetlig dokumentation framträder på flera ställen i handboken. T ex på sidan 11:

"Dokumenteringen möjliggör också anpassning av systemen till den miljö i vilken de skall utnyttjas och vidareutvecklas. Vidare underlättas bedömningen av alternativa ändringsförslags konsekvenser inom och mellan systemen"

På samma sida står det också:

"Ett företag kan beskrivas som uppbyggt av ett antal system. Mellan dessa förekommer skiftande former av samband"

Detta indikerar att verkligheten består av olika delar. Däremot menar man inte att det är möjligt att studera delarna oberoende av varandra.

Människosynen skymtar i kapitel 2, där man tar upp arbetsformerna för utvecklingsverksamhet. Där står t ex (sid 18):

"... Oavsett organisationsform är det viktigt att klargöra arbetsuppgifter och ansvar för de i utvecklingsarbetet medverkande.

Först sedan man klargjort vilka aktiviteter som skall genomföras, hur dessa skall struktureras samt deras omfattning, sker koppling till organisationsform och därmed bestäms personalbehov och personalutnyttjande.

Utvecklingsarbete bedrivs ofta i form av projekt... Ett projekt måste organiseras, planeras, ledas och följas upp"

Detta visar på en klar tayloristisk syn på arbetet, nämligen specialisering och att planering skiljs från utförande. Det innebär att det finns en grupp av människor som planerar och en som utför. Utförandet är mer rutinbetonat och kan utföras av lägre avlönad arbetskraft. Enhetlig dokumentation och arbetsmetodik underlättar denna rutinisering. En syn på människan, som måste styras och övervakas och inte kan betros med ansvarsfulla arbetsuppgifter är förhärskande. Från en metodchef hörde

jag en gång följande mycket belysande yttrande: "Varför ge en människa chans att göra fel, när en dator kan göra rätt?" Greenbaum (1976) har gjort en historisk analys av arbetsdelningen inom datayrket.

Sammanfattningsvis kan man säga att SIS/RAS anser verkligheten vara objektiv och neutral, möjlig att dela upp i av varandra beroende delar samt möjlig att beskriva på ett enhetligt och objektivt sätt i expertspråk. Detta förutsätter att kunskapen även den är objektiv och neutral. Man anlägger ett harmoniperspektiv, där motsättningar bara beror på bristande information. Organisationen ses som ett styrt system och människan ses som en mekanism för att utföra specialiserade arbetsuppgifter. Detta är med andra ord ett klassiskt, tayloristiskt perspektiv.

Till saken hör att systemutvecklingsmodeller i praktiken sällan används på avsett vis. Motiveringen är att den inte går att anpassa till det just då aktuella projektet (Flensburg & Sandström, 1980). Men ändå anses de viktiga och det forskas mycket på dem (Revay 1977, Brandt et al 1978, Brandt & Johansson 1980). Liknande analyser av andra systemutvecklingsmodeller och med ungefär samma resultat har även gjorts av t ex Olerup (1982, 1985), Swanson (1976) och Wood-Harper & Fitzgerald (1982).

2.3.3 Övriga ansatser

Det har under senare år lanserats ett antal alternativa sätt att utveckla system. *Prototyping* och *evolutionär systemutveckling* är sådana alternativ. Idén med prototyper är att göra en i vissa avseenden förenklad eller ofullständig konstruktion, avsedd att testas av de blivande användarna. På så sätt får användarna direkt se systemerarens tolkning av deras önskemål. För systemeraren blir prototypen en fungerande kravspecifikation, vilket undanröjer många missförstånd. Enligt Jenkins (1985) bygger prototyping på ett enda enkelt antagande, nämligen att det är lätt för en användare att tala om vad som är dåligt i ett existerande dataeller informationssystem, men mycket svårt att säga hur det ska vara istället. Kraushaar & Shirland (1985) anger dock behov av ökad systemutvecklingskapacitet som ett av huvudskälen till att införa prototyping.

Det finns likheter, men också väsentliga skillnader, jämfört med prototyper i industriell design. Jenkins (1985, sid 2) pekar på en del skillnader (tab 2.1) vilka enligt min mening inte är principiella. Grundidén är densamma: nämligen att göra utvärdering och mätningar innan en massa pengar investeras i en färdig produkt eller produktionsenhet. Men i genomförandet påminner dock Jenkins om traditionell Systems Life Cycle. Användaren kallas dock user/designer och dataexperten kallas systems/builder. Denna terminologi visar på en mera realistisk rollfördelning, men har tillämpats i traditionell systemutveckling sedan lång tid tillbaka. Det finns även andra metoder föreslagna (Kraushaar & Shirland 1985, Berrisford & Wetherbe 1983).

Manufactured product prototype

Development of the prototype occurs over a long period of time

The prototype costs more than the traditional "production" version

Multiple production versions are manufactured

Information system prototype

Development of the initial prototype occurs in a short period of time

The prototype costs less to develop than final product versions.

One production version is prepared (except in commercial environment)

Tab 2.1 Skillnader mellan prototyper i industrin och informationsbehandling (ur Jenkins 1985, sid 2)

Friis (1984, 1985) har experimenterat med en alternativ modell i Floyds (1984) och Rzewskis (1984) anda, där en av de stora begränsningarna med prototyping upphävs, nämligen att prototypen bara används för att få fram en god kravspecifikation. Hon menar att den kan mycket väl ligga till grund för en *evolutionär systemutveckling*, där man gör gradvisa förändringar och förbättringar i den ursprungliga prototypen, bara det finns lämpliga redskap tillgängliga. Floyd och Rzewski menar att dels behöver inte nödvändigtvis systemutvecklaren göra prototypen, användaren kan också göra den, dels kan det tänkas att man inte behöver göra den efterföljande traditionella systemkonstruktionen utan kan ta prototypen i direkt produktion. Eftersom de behövliga redskapen nu börjar bli ganska allmänt förekommande finns de praktiska förutsättningarna för denna typ av systemutveckling.

En tredje typ av systemutveckling är *interaktiv systemutveckling* där användarna själva gör sina system med hjälp av applikationsgeneratorer o dyl. Här är inte någon dataexpert inblandad och användaren sköter själv både utveckling och förändring av datasystemet. En skala kan konstrueras (Fig 2.13):

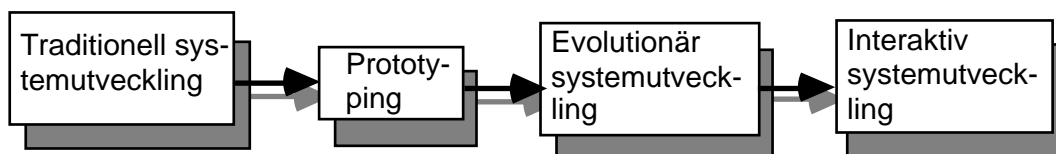


Fig 2.13 Mot mera användarinflytande

Ju längre åt höger man kommer desto större blir användarens inflytande och medverkan i systemutvecklingsprocessen. Dessutom blir skillnaden mellan utveckling och

användning allt mindre. Vid interaktiv systemutveckling är det i stort sett meningslöst att skilja på användning och utveckling.

Det finns inte här utrymme för en lika noggrann genomgång av grundföreställningar som jag gjorde för SIS/RAS. Ju längre åt höger vi kommer på skalan ovan, desto mer självständig och ansvarsställande anses människan och desto mer betonas individen. I evolutionär och interaktiv systemutveckling är styrsystemsinställningen borta och ersatt med ett inställning där individen är självständig och ansvarsställande. Eftersom prototyping bygger på visning och demonstration tyder det på att kunskap även kan vara oformaliserbar och icke uppdelbar eftersom den mänskliga tolkningen tillmäts stor betydelse. Verklighet och kunskap är inte längre objektiva och oberoende av iakttagaren. Däremot är det möjligt, åtminstone för en tid, bli överens om gemensamma företeelser, sk *intersubjektivitet*. Man har alltså helt andra grundsyner och grundföreställningar än vid traditionell systemutveckling vilket naturligtvis får konsekvenser för den färdiga produkten och dess användningsätt.

Sammanfattningsvis kan vi slå fast att systemutveckling är en process. I denna process framställs något, nämligen ett informationssystem. Processen är i huvudsak formaliserad. Resultatet av processen är dock inte alltid möjligt att fullständigt beskriva i formaliserad form. Det förefaller viktigt att bestämma under vilka villkor processen är möjlig att genomföra på angivet sätt samt föreslå alternativa processer i de fall den inte går att genomföra med hjälp av formella metoder. Prototyping och evolutionär systemutveckling är alternativ och i samband med deras bruk ökar användarinflytandet mer och mer. Interaktiv systemutveckling är ett annat alternativ där användarna tagit över processen helt och hållet. Därigenom främjas deras värden och de bestämmer själva hur arbetet ska förändras.

Jag ska nu diskutera några aspekter på informationssystem, som förefaller vara betydelsefulla vid val av systemutvecklingsmetoder. Ovan antydde jag, i anslutning till fig 2.13, att skillnaden mellan utveckling och användning minskar ju längre åt höger jag kom på skalan. Detta leder mig till att betrakta ändringsbarhet hos datasystemet.

2.4 Behovet av ändringsbarhet

Att utveckla ett datasystem innebär, om man anlägger ett arbetsperspektiv, att i förväg och för en relativt lång tidsrymd, i detalj planera hur ett arbete ska utföras. Det innebär att planering av arbete skiljs från utförande av arbete. Detta anser både Leavitt (1977) och Borum (1977) tyda på taylorism. För användarna av datasystem har arbetet övertagits av datorn och deras roll inskränker sig till att förse systemet med indata, ta hand om utdata eller övervaka det. Många gånger har det skett en dekvallifiering av yrkeskunskap.

Detta perspektiv innebär att datasystemet betraktas som relativt statiskt. Vissa typer av förändringar, såsom t ex lagersaldoförändringar, inleveranser, justering av beslutspunkter, ändringar i kontoplanen m m är förutsedda och kan tas om hand av det ordinarie systemet. Sådana typer av förändringar kallar jag förändringar på utförandenivån.

Men det kan inträffa andra och mer genomgripande förändringar, vilka man inte har förutsett. T ex kan helt nya verksamhetsinriktningar dyka upp, på grund av högre drivmedelspriser kan mera detaljerad planering av transporter krävas m m. Sådana förändringar kallar jag förändringar på specifikationsnivån. De är inte förutsedda i existerande system och kräver mer eller mindre avancerade insatser från dataexperter. De kan leda till både underhåll och nykonstruktion.

När nu världen förändras allt mer och allt djupgående (Toffler 1971) leder detta till ett ökat behov av flexibla system, system där man lätt kan ändra specifikationerna. Även dataavdelningarnas stora börda med underhåll av befintliga system leder till att eventuella nykonstruktioner måste bli flexibla. Orsakerna till fel på specifikationsnivån kan i princip vara av följande slag (Flensburg & Sandström 1980, Flensburg 1981c):

1. Kravspecifikationen är felaktig
2. Lämnade uppgifter har feltolkats av systemkonstruktören
3. Systemkonstruktören har begått misstag vid konstruktionen
4. Utdata tolkas felaktigt av användarna

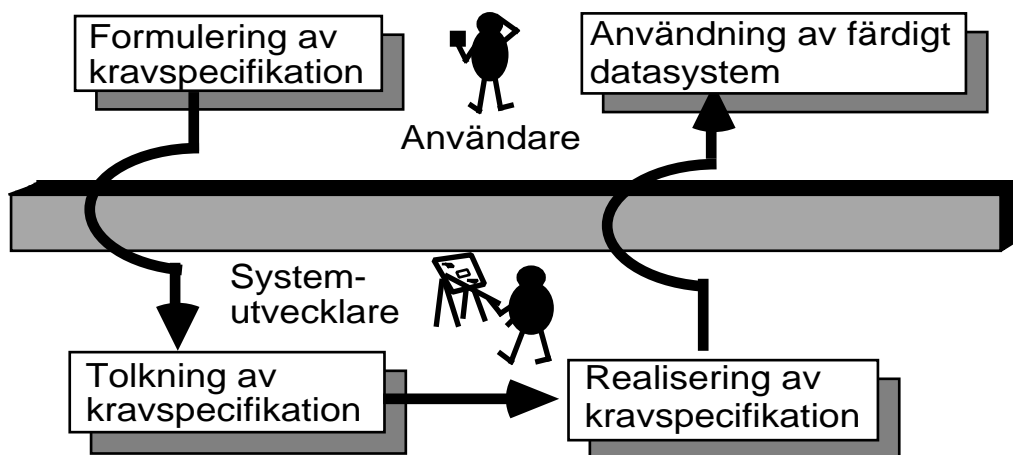


Fig 2.14 Kommunikationsmuren mellan dataexperter och användare

I figur 2.14 har jag beskrivit systemutveckling som en transformationsprocess (enl Kraushaar & Shirland 1985), där systemet under sin livstid kan inta fyra möjliga tillstånd:

- Kravspecifikation enligt användare
- Kravspecifikation enligt systemutvecklare
- Realiserat system
- System i användning

Misstag kan uppkomma i transformeringen mellan dessa olika tillstånd förutom att det kan vara fel från början. Speciellt stor är risken för misstag då man går över kommunikationsmuren mellan dataexpert och användare. Kommunikationen mellan dataexpert och dataanvändare sker som regel i dataexpertens yrkesspråk. Det innebär att endast fenomen som är relevanta i den miljön kan beskrivas och således medvetandegöras. Beskrivningen av det befintliga arbetet blir ofullständig. Dessutom behärskar användaren dåligt detta språk, varför missförstånd lätt uppstår. Ytterligare en källa till felaktigheter är att dataexpertens yrkesspråk i sig inte är generellt väldefinierat utan leverantörs- ja rent av datormodellbundet. Detta utnyttjas ofta som ett medel att hårdare knyta köpare av datorer till leverantörerna.

Misstag av typ 1 uppkommer automatiskt efter ett tag. Organisationen förändras, världen förändras och så även användningen av datasystem. Det måste därför finnas möjlighet att göra relativt täta justeringar av kravspecifikationen för att säkerställa ett korrekt resultat från datasystemet.

Felaktigheter av typ 2 är mycket vanliga och beror enligt min mening till stor del på användarens tysta yrkeskunskap, vilken inte går att förmedla explicit. Precis som man i språket ofta blir medveten om en regel först då man bryter mot den (Israel 1979), men ändå inte kan formulera den explicit, vet man vad man vill ha först då man fått det. Om traditionell systemutveckling tillämpas finns råd till bara ett försök. Med hjälp av prototyper och evolutionär systemutveckling kan man i viss utsträckning ändra på dessa förhållanden och förmodligen ännu mer vid interaktiv systemutveckling.

Misstag av typen 3, systemutvecklarens misstag, kan man minska på flera sätt. Ett är bättre utbildning av systemkonstruktören, ett annat är bättre systemutvecklingsmodeller. Det har gjorts lovande försök att specificera systemen genom att sätta upp ett antal "informations requirements" med tillhörande "constraints" i något beskrivningspråk och därefter automatiskt generera det färdiga systemet. I praktiken innebär det att delar av systemeringsarbetet blir automatiserat. Eftersom beskrivningspråket måste nära ansluta till användarens yrkesspråk medför det att användaren själv utan alltför stort besvär eller skolning själv skulle kunna göra åtminstone förändringar i systemet.

Den fjärde typen av fel, att utdata tolkas felaktigt av användarna kan bero på att utdata är felaktigt och då gäller något av de andra felen, eller att användaren inte kan

sitt yrke tillräckligt bra. Endast i detta fall ska det ske en anpassning till datasystemet, i alla andra fall ska datasystemet anpassas till användaren. Detta leder sammantaget till stora krav på flexibilitet och kontinuerliga förändringar på specifikationsnivå. Därmed upphävs den skarpa skiljelinjen mellan utveckling och användning.

Om nu interaktiv systemutveckling tillämpas, hur blir då systemen jämförda med dagens system? Vi nalkas då samma problem fast från ett annat håll.

2.5 Förändringar i verksamheten

All verksamhet förändras. Förändringar kan vara av minst två olika slag: Förutsedda och oförutsedda. De förutsedda är t ex förändringar i lagernivå, beställning av nya varor, tillbyggnad av ny fabrik. De oförutsedda är t ex oväntade devalveringar, krig, snabba marknadssvingningar, personalskifte osv. För att organisationen ska överleva krävs viss förmåga att anpassa sig till både förutsedda och oförutsedda förändringar.

För att kunna anpassa sig till en förändring krävs emellertid kunskap om att en förändring inträffat. Det finns ofta inbyggda mekanismer i datasystem för att upptäcka förutsedda förändringar, t ex lagerbrist, utleveranser. Det finns även långsiktiga planer om t ex nybyggnationer. Dessa typer av förändringar utgör i regel inga problem.

Det gör däremot de oförutsedda eftersom det inte finns färdiga mekanismer att upptäcka dem. Deras effekter visar sig genom oväntade förändringar i olika tillståndsvariabler (t ex exceptionellt låg orderingång). Datasystemet måste tillåta analyser av orsakerna till de oväntade förändringarna. Detta kan inte göras genom på förhand definierade listor utan det krävs sk ad hoc frågor. För att kunna ställa sådana krävs ofta stora förändringsmöjligheter hos datasystemen.

De oförutsedda förändringarna inträffar allt oftare och blir allt större. Flexibla datasystem behövs därför alltmer. Men då de redan befintliga datasystemen måste förändras i allt snabbare takt får systemavdelningarna än mindre tid för nyutveckling av dessa nya och flexibla system. (Martin 1982). Det yttrar sig som stor brist på programmerare och systemutvecklare. Det finns olika sätt att häva denna brist. Ett är att utbilda fler systemerare eller programmerare. Ett annat är att öka produktiviteten hos dagens datafolk. Detta kan ske genom effektivisering av traditionella metoder. Martin (ibid) räknar med 10-20 % ökning på så sätt. Ett annat sätt är att låta dataexperterna använda sig av de sk 4:e generationens programspråk. Därigenom kan produktivetsökningar på uppemot 50-80% uppnås i kodningen enligt Martin. Ett fjärde sätt är att förse varje användare med enkla och kraftfulla redskap som gör det möjligt för vederbörande att göra sina egna datasystem. Därvid ändras de traditionella rollerna för systemutvecklarna.

Den mest fördelaktiga kombinationen är den fjärde ty då blir det både fler och produktivare "dataexperter". Visserligen blir kanske en "vanlig" användare inte lika produktiv som en professionell programmerare, men de blir desto fler till antalet. En annan fördel med att låta användare bli "programmerare" är att verksamhetskunskap på ett enkelt sätt kan introduceras i datasystemen. Detta sker på flera sätt. Ett är rent språkmässigt, beteckningar och variabler får namn som är bekanta för den som arbetar i verksamheten. Ett annat är att man undviker anpassningar till standardlösningar och istället kan ägna sig åt genuin problemlösning. Den tredje och som jag nu ser det, största fördelen är att datasystemet blir lätt att ändra när behoven förändras. Användaren har både makten och ansvaret för att genomföra nödiga förändringar. Förhoppningsvis kan vederbörande också skaffa sig nödvändig kunskap.

Sammanfattningsvis kan vi alltså beträffande förändringar konstatera att kraven ökar hela tiden medan traditionell utvecklingsmetodik endast tar hänsyn till vissa typer av förändringar. Det borde på sikt leda till brister i datasystemen.

2.6 Brister hos datasystem

Datasystem görs för att bli använda. De måste därför anpassas till aktuell verksamhet. För den skull måste yrkeskunskap tillföras systemutvecklarna från användarna. Detta är inte alltid problemfritt. Ett antal potentiella och reella felkällor finns. Lyytinen (1983) har angivit 6 olika typer av brister på specifikationsnivån hos datasystem.

1. Objektivistisk brist

Innebär en tro att verksamheten kan beskrivas objektivt och oberoende av de personer som utför arbetet.

2. Deskriptiv brist

Innebär en tro att det är möjligt att göra en beskrivning av väsentliga förhållanden i verksamheten uttryckt i påståendesatser om hur det förhåller sig.

3. Konceptuell brist

Innebär en tro att det finns en uppsättning begrepp som beskriver en verksamhet och som är oberoende av de yrkesspråk som används i verksamheten.

5. Strukturell brist

Innebär en tro att studiet av data och deras mening kan reduceras till formellt studium av någon gemensam struktur.

5. Formalistisk brist

Innebär en tro att det är möjligt och tillräckligt att göra en formell beskrivning av en verksamhet för att förstå dess mening.

6. Konsensusbrist

Innebär en tro att det är möjligt att uppnå en beskrivning av en verksamhet som alla berörda personer instämmer i.

Lyytinen diskuterar utförligt dessa brister och analyserar även de inbördes förhållandet mellan de olika bristerna. Han kommer fram till att den första, den objektivistiska bristen, är en grundförutsättning och direkt ledande till i stort sett samtliga andra brister. Men tron på en objektiv tillvaro är en av grundstenarna i traditionell systemutveckling. Man kan alltså förmoda att Lyytinen observerade brister helt enkelt beror på ett okritiskt användande av formella metoder vid systemutvecklingen. Vad som möjligen kan invändas mot Lyytinen är att han utgår från ett särskilt perspektiv (språkaktsteori) och kritiserar ett annat (informationsmodellering). I praktiken tillkommer två mycket stora grupp av fel, nämligen rena programfel och inmatningsfel. Här har förutsatts att själva överföringen från specifikation till datorprogram kan göras felfritt. Detta är en mycket stark förutsättning, men inte ens med denna starka förutsättning kan man anta att systemutvecklingen ger det önskade resultatet.

Traditionellt utvecklade basdatasystem (för termen "basdatasystem" se Brandinger & Norrby, 1980) ger ofta inte det utbyte man kalkylerade med vid deras tillkomst (Andersen m fl 1980). Jag anser att detta beror på främst två saker:

- Kommunikationsproblem mellan systemutvecklare och systemanvändare gör att kravspecifikationen missförstås och fel system utvecklas.
- Svårigheten att ändra i befintliga system. Detta beror i sin tur på ett antal andra faktorer såsom brist på programmerare, höga kostnader för att ändra, bristfällig dokumentation, uppgivenhet hos användarna, som inte får genom sina krav i rimlig tid.

En del av de upplevda bristerna kan förklaras genom svårigheter att överföra yrkes- och verksamhetskunskap. Att råda bot på detta kräver ökat användarinflytande. Då borde en hel del (men inte alla) av de här redovisade problemen försvinna eller åtminstone reduceras. Det kan vara värt att undersöka om problemen reduceras mest vid satsning på den mest extrema formen av användarinflytande, nämligen interaktiv systemutveckling. Det förefaller motiverat att undersöka denna ansats mera detaljerat. Förmodligen tillkommer nya problem och detta gör det ännu mer motiverat att undersöka saken. Ett sådant tänkbart problem rör arbetsmiljö och arbetstillfredsställelse.

2.7 Arbetstillfredsställelse

Det finns många olika dimensioner av detta begrepp. Jag brukar använda mig av en begreppsapparat, föreslagen av Enid Mumford (Mumford 1983). Följande dimensioner ingår i arbetstillfredsställelse eller *job satisfaction*:

- *Yrkeskunskap*, där en hög grad av yrkeskunskap väntas ge positivt bidrag till arbetstillfredsställelsen.
- *Organiseringen av arbetsuppgifterna* vilket innebär varierande och intressanta arbetsuppgifter, feedback och självstyre
- *De psykologiska behoven* av uppskattning och självförverkligande väntas ge positivt bidrag till arbetstillfredsställelse.
- *Den ekonomiska ersättningen* och *arbetsledningen* innebär rimlig ersättning för rimligt arbete med en rimlig grad av övervakning och styrning
- *De etiska värderingarna* och deras överensstämmelse med den organisation inom vilken arbetet utförs spelar mycket stor roll för arbetstillfredsställelse. Kanske rent av den största rollen. (Flensburg 1980b)

Mumfords ansats baseras på den så kallade "human relation"-skolan. Det finns även en annan ansats som tillämpas främst inom fackföreningsrörelsen. Den baserar sig i större utsträckning på klasskampsideologi. Mellan dessa båda ansatser kan striden ibland stå ganska skarp. Vissa företrädare för fackföreningsrörelsen (t ex Ehn i Göranson 1978) anser att den sociotekniska ansatsen är ett ovanligt elakartat försök från arbetsgivarnas sida att manipulera och suga ut arbetskraften. Vilket perspektiv man väljer är en politisk-ideologisk fråga och jag skall i detta arbete försöka åstadkomma någon slags sammansmältning. Detta är, enligt min mening, möjligt att göra på individnivå, men inte säkert på organisationsnivå.

Jag har nu ringat in ett problemområde, nämligen interaktiv systemutveckling, där slutanvändarna både designar och realiserar sina datasystem. Främsta fördelen förefaller vara möjligheten att på ett enkelt sätt ändra i datasystemen för att därigenom uppnå en bättre anpassning mellan system och verksamhet. Men jag måste då först se vad som redan gjorts inom området.

2.8 Övrig forskning

I detta avsnitt går jag främst igenom forskning i USA. Jag inleder med ett referat av Martins (1982) bok, som blivit ett slags standardverk för de som praktiskt arbetar med applikationsgeneratorer. Sedan redovisar jag forskning som bedrivits kring bruk av applikationsgeneratorer. Det visar sig att denna inte är speciellt inriktad på hur slutanvändarna upplever bruket av dessa utvecklingshjälpmedel. Detta blir så småningom mitt övergripande problemområde.

2.8.1 Applikationsgenerator

En applikationsgenerator kan definieras som en programprodukt med vars hjälp en icke dataexpert kan utveckla datasystem i huvudsak för eget bruk. Datasystemet skall vidare vara ett hjälpmedel i denne persons arbete. Det ska underlätta vederbörandes datahantering. Datasystemet behandlar data, som tolkas av dess användare och därvid erhåller en viss innebörd. Denna typ av databehandling blir resultatet vid interaktiv systemutveckling. Eftersom skillnaden mellan utveckling och användning i det närmaste är upphävd finner jag det föga meningsfullt att tala om utveckling. Som alternativ term föreslår jag därför *personlig databehandling*. Redskap behövs för:

- lagring av data
- återvinning av lagrade data
- bearbetning och presentation av data.

Funktioner för detta måste finnas i applikationsgeneratoren. Ett minimum av egenskaper hos denna är således:

- registeruppläggning
- registerförändring
- rapportgenerator
- beräkningar

Alternativa namn på applikationsgenerator är informationshanteringssystem (Lie-Nielsen & Colliander, 1983) och fjärde generationens programspråk. Dataexperter kan också använda applikationsgeneratorer, men de ska i alla fall vara så pass enkla att använda att det efter 1-2 dagars utbildning är möjligt för även en icke-dataexpert att använda dem.

Det finns fyra olika typer av hjälpmedel, som är mer eller mindre (mest mindre) integrerade:

1. Frågespråk mot databaser (registerhanteringsprogram)
2. Modellspråk (spreadsheet, kalkylprogram)

3. Grafik

4. Ordbehandling

Andra typer av hjälpmedel man kan tänka sig är:

5. System för informationssökning i interna eller externa databanker.

6. Elektroniska brev, datakonferenser

De olika typerna av hjälpmedel är i viss utsträckning integrerade. Det finns i regel möjligheter att överföra utsökningsmaterial från ett hjälpmedel till ett annat.

Martin (1982) talar om olika generationer av programspråk (s 28):

"The first generation was machine languages. The second generation was languages the level of assembler language. The third generation was machine-independent languages of the level of COBOL, PL/1 and BASIC."

Den fjärde generationens språk är, enligt Martin, sådana där man gör kodningen på högst en tiondel av den tid det skulle ta med ett tredjegerationsspråk som t ex COBOL. Men kodningen är ju bara en liten del av det totala systemarbetet. Det finns vidare olika sorters applikationsgeneratorer. Martin skiljer på sju sorter (s 14 -16), nämligen:

1. *Enkla frågespråk*, som helt enkelt dumpar ett postinnehåll. Sådana har funnits sedan länge och är inte applikationsgeneratorer i egentlig bemärkelse.

2. *Komplexa frågespråk*, där man kan ställa frågor rörande flera poster (*join*).

3. *Rapportgeneratorer*, som innebär att en viss typ av data extraheras ur en fil och presenteras i form av en lista. Totaler och subtotaler ska kunna erhållas. Sådana har också funnits länge.

4. *Grafikspråk*, som innebär att data erhållna t ex i ett komplext frågespråk kan presenteras i grafisk form (histogram, kurvor, tårtdiagram).

5. *Applikationsgeneratorer*, som används i hela tillämpningen med registerskapande, uppdatering, delrutiner, formulär, rapportgenerator.

6. *Superhögnivåspråk*, vilka innebär att inte alla detaljer i databehandlingen behöver beskrivas. Man kan uträtta mycket med ett enda kommando och ofta arbetar ett superhögnivåspråk mot en särskild databas.

7. *Parametriserade standardssystem*, där bearbetningen i viss utsträckning kan styras med hjälp av parametrar

Man kan naturligtvis ha invändningar mot Martins indelning. T ex innehåller en applikationsgenerator i sig ofta ett högnivåspråk. Det framgår inte heller om frågespråk berör ett eller flera register. Själv skulle jag vilja föreslå följande funktioner i en applikationsgenerator utan att för den skull klassificera den som ena eller andra typen:

- Möjlighet att skapa register.
- Möjlighet att ställa frågor som berör flera poster i samma register.
- Möjlighet att enkelt kunna skapa formulär för in- och utdatapresentation.
- Rapportgenerator med möjlighet till totaler och subtotaler.
- Frågor och rapporter ska i avancerade versioner kunna beröra flera register.
- Möjlighet att visa registerstrukturen.

Martin hävdar att antalet tillgängliga programmerare/dator minskar så fort att de flesta datorer i framtiden, åtminstone delvis, måste arbeta utan hjälp av programmerare. Det beror på att datorerna blivit så billiga. Men Martin tycks bortse från standardpaket.

Innan första världskriget trodde man, enl Martin, i USA att det inte gick att förse varje människa med egen telefon eftersom det inte fanns tillräckligt med telefonister. Hela USA:s arbetskraft hade i så fall behövts för att koppla telefoner (Vem skulle i så fall ringa?). Lösningen låg förstas i automatisering.

Samma sak menar han gäller för programmerare idag. Antalet datorer ökar f n med mer än 25% per år. Produktiviteten när det gäller systemutveckling kommer sålunda i runda tal att behöva hundrafaldigas under den närmaste tioårsperioden. Bättre systemeringsmetoder ger i detta sammanhang blott marginella förbättringar (10-20%). Alternativet är, enligt Martin, detsamma som för telefonen, automatisera så den enskilde användaren blir sin egen systemutvecklare.

I USA är genomsnittskostnaden per korrigerad instruktion c:a 10 dollar. Vid komplexa tillämpningar ökar den till upp mot 75 dollar. Det är ungefär lika mycket som en enkel mikroprocessor eller rentav mikrodator kostar.

De flesta dataavdelningar har enligt Martin en synlig backlog (kölista på nya uppdrag) på mellan 3 och 4 år. Till detta kommer ett antal tillämpningar där användarna inte ansett det vara lönt att ens be dataavdelningen göra den. Denna osynliga backlog kan ofta vara större än den synliga.

Martin anser att man kan nyttja tre strategier för att råda bot på programmerarbristen:

1. Slut användarna kan få hjälpmedel så att de kan göra sina egna system.
2. Systemutvecklare som arbetar tillsammans med användarna kan använda sig av redskap med vars hjälp de skapar systemet istället för att skriva programspecifikationer.
3. Förprogrammerade tillämpningspaket kan köpas utifrån.

Ett mycket stort amerikanskt oljebolag har formulerat följande strategiska mål för systemutveckling och systemanvändning för perioden 1984-1989 :

"End users can access and manipulate information (numeric, text and graphics) required to solve their problem 50% of time without DP help.

30% of applications can be generated or supplied directly from business requirements (without programming).

70% of application can be supplied with no more than 30% procedural code."

Martin menar att mindre företag i bästa fall har råd med någon systemerare som gör bildskärmslayouter eller genererar program med hjälp av programgenerator. Det mellanstora företaget har det sämre, för där vill man göra allt själv och använda traditionella metoder, för att "pioneers often get killed". Följande stycke är citat från sidorna 7 och 8 i Martin:

"To achieve the degree of end-user participation in computing required by the strategy in Fig 1.2 (punkterna för det stora företaget ovan) is a major change. It requires "user-friendly, nonthreatening" software and application dialogues. It requires a massive number of users to be trained. It requires a change in the role of the systems analyst who will increasingly assume a consultant role, guiding and teaching the users, showing them the possibilities, acting as salespersons of new methods, solving the users' problems and generally handholding. It will require an understanding by senior management that they must motivate the users to participate, putting statements in their management-by-objectives forms, which instruct and encourage users at all levels to apply the new DP tools.

An opinion commonly hold by DP professionals and by computer manufacturers and development staff is that end users are not mental equipped to participate in the DP creation process... Many users feel that they are too busy or have something more important to do. Many are frightened of terminals and do not use them because they might appear foolish in front of their colleagues. Some have a slightly arrogant attitude and will not learn, in the same way that a manager will not learn to type. Users often need to be encouraged or instructed from above to get over the hurdle which prevent them making better use of today's computers. ...

There are now many examples of end users participating in the DP process with results much better than those that would have come from DP professionals, because the end users know the subtleties of what they want to accomplish. To make this happen on a large scale needs software and tools quite different from those which the traditional programmer uses. Those facilities now exist and will rapidly improve. ..."

Martin tar här upp viktiga förutsättningar, nämligen utbildning av användare och behovet av lämpliga redskap. Han hävdar också att användare generellt sett är bättre systemutvecklare än dataexperter eftersom de har bättre kunskap om verksamheten.

För att öka produktiviteten i systemutvecklingen föreslår Martin ett antal metoder (s 9). Här är en koncentration av dem:

- Använd moderna databaser
- Använd icke-procedurella högnivåspråk
- Använd självdokumenterande tekniker
- Undvik att skriva programspecifikationer. Generera istället systemet eller åtminstone en prototyp tillsammans med användaren.

Förekomsten av en väl uttänkt databas är nyckeln till att förbättra systemeringsprocessen. Systemeraren kan specificera händelser som påverkar databasen och ange de aktiviteter dessa förorsakar.

En av de mera intressanta programmeringsformerna är de många "Vad händer om..."frågorna som företagsledningen ofta ställer. Som hjälpmedel för att svara på frågorna används i regel något kalkylprogram.

Det finns en mängd programvara i dessa olika kategorier. Martin har gjort en grov indelning i sådana som är lämpliga för slutanvändare och sådana som är lämpliga för dataexperter. Han definierar längre fram i boken (sid 102) vad han menar med "lämplig för användare". Det innebär i stort sett att slutanvändaren efter en tvådagarskurs skall kunna använda applikationsgeneratoren som nyttigt och lätthanterligt hjälpmedel i sitt arbete. En intressant passus är att man skall kunna lämna produkten i en vecka och därefter fortfarande kunna använda den. Dessutom skall det finnas inbyggd HELP. Martin menar att programspråken för slutanvändarna ska ha följande egenskaper (sid 180):

1. They are interactive, designed for terminal use
2. It is very easy to write beginners' programs. More difficult features can be learned a step at a time.
3. They can use existing or specially created data bases (preferably relational data bases) and high-level data-base manipulation commands.
4. Default options exist so that a program takes intelligent actions when not instructed to do otherwise. For example, it creates reports in a reasonable format if the user does not specify any specific format.
5. High-level nonprocedural routines are available for most operations: that is, the user tells the system what to do rather than how to do it.
6. Easy-to-use graphics capabilities exist with which attractive diagrams can be created and manipulated (in color where possible).
7. On-line instruction HELP facilities are always available to users.
8. There are good on-line debugging aids, and debugging can proceed incrementally."

Skillnaden mellan ett traditionellt programspråk och en applikationsgenerator är bara att mycket kan utträttas med ett enda och lättfattligt kommando. Lättfattligheten är väsentlig. APL t ex är ett kraftfullt språk, men kanske inte alltid så lättfattligt för en icke-APL-expert. Jag har dock sett exempel på vanliga användare som lärt sig APL

och i sitt dagliga arbete nyttjat APL. Många applikationsgeneratorer har APL som värdspråk.

2.8.2 Användning av applikationsgeneratorer

Jag tror att interaktiv systemutveckling är ett bra och effektivt sätt att göra goda och användbara datasystem. Många andra tror samma sak. En brytningstid är inne just nu och det är angeläget att på ett tidigt stadium försöka kartlägga effekter och användning, dels för att råda potentiella användare av applikationsgeneratorer, dels för att upptäcka eventuella missförhållanden så tidigt som möjligt. En del undersökningar har redan gjorts. Andersson (1983) redovisar följande problem i samband med applikationsgeneratorer:

- 1) Små företag som tillverkar dem
- 2) Kräver mycket datakraft
- 3) Svårt att koppla till befintligt system
- 4) Språken blir alltmer komplexa
- 5) Man använder fel språk t ex ett kalkylspråk i st f ett redovisningsspråk.
- 6) Man kan språket dåligt
- 7) Gamla systemutvecklingsmetoder används
- 8) Risk för många olika språk i ett företag
- 9) Motstånd från dataavdelningen som ser sin maktposition hotad.

Martin hävdar att det viktigaste skälet för interaktiv systemutveckling är att reducera dataavdelningens backlog och öka dess produktivitet. En undersökning på tio kanadensiska företag, gjord av Rivard & Huff (1984) visar emellertid att dessa skäl spelar en underordnad roll. Det viktigaste skälet är istället en ökad kvalitet på informationen och på arbetet. Mina undersökningar stöder än så länge Rivard & Huffs resultat.

Leitheiser & Wetherbe (1984a) har bl a gjort en survey över 25 stycken Information Center (IC, en avdelning som stöder och initierar personlig databehandling) för att undersöka vilken service de lämnar, ev inriktning på time-sharing eller persondatorer, anledning att ha Information Center, typiska föroch nackdelar samt faktorer som är avgörande för framgång. Undersökningen berör amerikanska företag i mellanvästern.

Man har i denna undersökning funnit att de flesta IC ger service åt antingen time-sharing på stordatorer eller åt bruk av persondatorer, mycket sällan stöder man både-och. Servicen bestod mest av konsulthjälp, utbildning och felhjälp. De redskap som användes mest var applikationsgeneratorer, rapportgeneratorer och statistiska paket. Främsta anledningen att använda IC istället för traditionell databehandling var att minska tiden mellan behov av datahjälp och dess verksställande. Främsta anledningen för användarna att nyttja IC:s tjänster var att de behövde den experthjälp som där fanns tillgänglig. Största framgången för IC var att enskilda tillämpningar blev lyckade. Största motgången var att dess kapacitet inte räckte till. Främsta anled-

ningen till framgång var tillhandahållande av önskad service inom rimlig tid och kompetent personal. Först på tredje plats kom valet av rätt redskap.

Denna undersökning koncentrerar sig på IC-personal och tar inte alls upp de egentliga användarnas erfarenheter. Det är en brist författarna är medvetna om och påpekar i rapporten. Vidare gäller det amerikanska förhållanden och dessa är inte säkert direkt överförbara till svenska eller skandinaviska. Bli a är storleken på företagen väsentligt större än vi är vana vid här i Sverige.

En annan brist i undersökningen är att man inte beaktar de system som görs av användarna. Om det är "bra" system eller "dåliga" system och för vem systemen i så fall är bra eller dåliga tas inte alls upp. Det tar däremot Davis upp i en artikel (1984), där han ärganska kritisk till "end-user computing" (personlig databehandling) och menar att det är stora risker att informationen i systemen blir felaktig. Leitheiser och Wetherbe har i en annan artikel (1984b) gett några råd för hur "end-user computing" ska organiseras för att bli framgångsrik. Främsta medlet är inrättande av IC med väl fastlagda serviceoch ansvarsnivåer. Men både Davis och Leitheiser och Wetherbe har mer eller mindre medvetet antagit att det är samma typ av datasystem som utvecklas interaktivt och som utvecklas på traditionellt vis. Så behöver emellertid inte vara fallet, utan andra systemstrukturer och andra arbetssätt kan bli aktuella. Interaktiv systemutveckling utgår nämligen från ett helt annat synsätt än traditionell systemutveckling.

Davis (1984) ger en definition av "end-user computing" (sid 4):

"End-user computing is the organization of computing resources and design of information systems applications such that:

Application systems provide direct, immediate support for user activities

Information requirements are specified by the user and may be changed by the user as the system is used

The development and use of the system is under user control"

Såvitt jag kan finna är det endast den sista punkten som skiljer "end-user computing" från traditionell expertorienterad databehandling i varje fall ur skandinavisk synvinkel. Davis tar också upp förhållandet mellan dataavdelning och användare i form av ett antal tänkbara "policies" (sid 5):

1. "Information center with analysts who program to user specifications
2. Informations center with analysts who assist users to utilize the facilities and provide training and consultation.
3. Analysts assigned to user areas who assist users by doing the work or by aiding users with the facilities

4. A policy supporting end-user computing, but no direct support organization
5. No policy or support
6. Policy that there will be no end-user computing"

Att lägga märke till är att dataavdelningen i många fall "gör jobbet". Men Davis betonar att det inte är någon tudelning mellan interaktiv systemutveckling och expertutveckling utan snarare ett kontinuum, där mer eller mindre av ansvaret ligger på användaren. Davis anger tre skäl för interaktiv systemutveckling (s 9):

1. "Relative shortage of system development personel
2. Eliminate the problem of informations requirements determination by information system personnel
3. Transfer the information system implementation process to users"

Han ser emellertid ett antal faror också. T ex är det lätt att personlig databehandling kräver mer stöd från dataexperter, i form av hjälp, krav på datatillgänglighet och kanske nya tillämpningar. Den största faran Davis ser är dock att interaktivt utvecklade system inte uppnår tillräckligt hög kvalitet i form av datakontroller, säkerhet och stabilitet. Det finns också risk för personliga system som inte ansluter till etablerade standards. Han menar också att det måste finnas någon som kritiskt granskar användarnas krav.

Davis uppehåller sig också mycket vid de olika sätt användare har för att göra kravspecifikationer (sid 13):

- "1. Relay upon current understanding of needs.
2. Obtain requirements from some existing system the user has seen or can examine.
3. Obatin requirements from analysis of the characteristics of the decision system or other utilizing system.
4. Obtain requirements through trial and error incremental change in an evolving application"

Men vad Davis inte ifrågasätter är behovet av kravspecifikation. Den sista strategin är dock ett steg på vägen. Samma strategi kallar Sandström (1985) för "systems development in the small". Davis ger uttryck för en mängd åsikter om hur människor beter sig ifråga om kravspecifikationer. T ex utgår de alltid från information som redan är tillgänglig, de utgår från den tillgängliga formen, de utgår från det som hänt nyligen och är slutligen urusla på intuitiv statistik. Alla dessa förhållanden gör att människor i allmänhet och beslutsfattare i synnerhet är dåliga på att ange kravspecifikationer. Systemutvecklare är enligt Davis sätt att se mycket bättre på detta, främst för att de är mera vana vid att strukturera problemen.

Men det står alldeles klart att interaktiv systemutveckling "är på gång". I samband med NordData 80 i Åbo redan, gjorde jag en enkel enkät bland en del konferensdeltagare. Jag ställde mig vid en trafikerad passage och frågade sammanlagt 58 personer om de ansåg att användarna i framtiden kommer att överta större delen av ansvaret för systemutvecklingen. Svartalternativen var: Ja, I någon mån, Nej, Vet inte. Endast fem personer svarade "Nej" (de arbetade alla inom driftsavdelningen), 30 stycken svarade "Ja".

I MIS Quarterly från september 1984 redovisas en undersökning av Dickson, Leitheiser, Wetherbe och Nechis rörande "Key Information Systems Issues for the 1980's". Den viktigaste frågan är "Improved IS planning" och den har fått 9.1 poäng av 10 möjliga. Därefter kommer "Facilitation and management of end user computing" med ett värde av 7.4. Undersökningen lades upp som en Delphi-studie med auktoriteter både från universitet (4 personer) och näringsliv (c:a 25 personer). Deras uppgift var att ordna de tio viktigaste frågorna i samband med informationssystem. Delphistudien genomfördes per post i fyra ronder och antalet svar i varje rond varierade från 52 till 102.

I fig 2.15 gör jag ett försök att klargöra vad jag här avser med de centrala begreppen Information Center och Personlig databehandling. Dataavdelningen ser jag som ett sammanfattande begrepp för både drift, underhåll och nyutveckling. Ganska fristående från denna avdelning finns Information Center. Dataavdelningen tillhandahåller datakraft och redskap medan Information Center stöder och hjälper vid bruket av dessa redskap. Resultatet av bruket blir personlig databehandling.

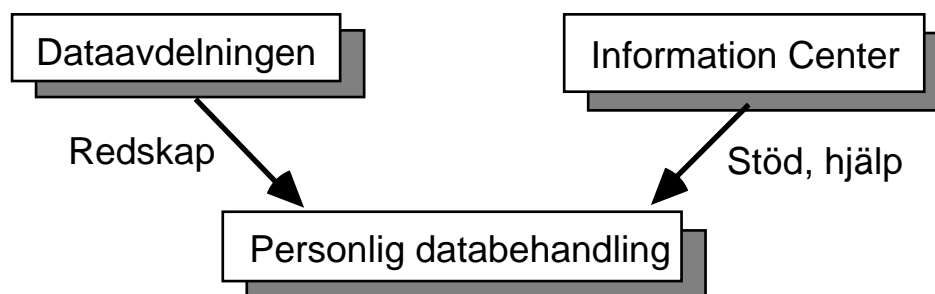


Fig 2.15 Begreppsrelationer

Avslutningsvis kan jag konstatera att det forskats mycket kring redskapen och ur dataexperternas synvinklar. Men det tycks fattas forskning ur användarperspektiv. Detta kan bli en uppgift för mig.

2.9 Avgränsningar

Man kan alltså grovt sett säga att mitt problemområde är interaktiv systemutveckling av datasystem ur användarnas egna perspektiv. Orsaken till att jag är intresserad av att studera detta är att det förefaller mig vara någon form av lösning till "det ingenjörsvetenskapliga dilemman", dvs att samtidigt använda både "mjuk" och "hård" metodologi enligt Checkland (1981). På detta vis hoppas jag att informationsteknologin ska kunna nyttjas på ett för samhället och människorna mera tillfredsställande sätt.

Det gäller att hålla nere missbruk och mindre lämpliga sätt att nyttja denna teknologi samtidigt som dess potential bör utnyttjas. Det förefaller rimligt att börja med att empiriskt undersöka effekterna av interaktiv systemutveckling. Då området är nytt och relativt outforskat, förefaller någon form av explorativ undersökning lämplig. Det förefaller rimligt att försöka åstadkomma någon beskrivning av vad som har hänt i ett antal fall av interaktiv systemutveckling, dvs kartläggning av effekter.

Det är viktigt att ordentligt förstå vad interaktiv systemutveckling innebär för de personer som arbetar med det, för att bättre kunna förstå de observerade effekterna samt ömsesidigt, för att bättre kunna observera och förstå effekter. Någon mer djupgående studie förefaller därför lämplig. Ett aktivt deltagande i interaktiv systemutveckling är ett exempel på en sådan studie. Det är naturligtvis inte alls oproblematiskt men jag tror att en djupare förståelse är nödvändig för att kunna ge underlag för upptäckt av viktiga faktorer för sådan utveckling. Det är knappast försvarbart att enbart sitta hemma vid skrivbordet och tänka ut vilka faktorer som är viktiga vid undersökning av effekter av interaktiv utveckling. Jag måste ut i verkligheten och med egna ögon, öron och andra sinnen varsebli vad som händer. Det kräver deltagande i interaktiva systemutvecklingsprojekt.

Min forskningsuppgift innebär alltså två undersökningar:

- **Insamling av erfarenheter av interaktiv systemutveckling.**
- **Aktivt deltagande i interaktiv systemutveckling.**

Eftersom interaktiv utveckling är ett så pass nytt område finns det inte mycket empirisk forskning inom det. Det finns inte heller några väl utbyggda teorier (ansatser finns för beslutsstödsystem). Det förefaller därför rimligt att göra en explorativ undersökning, med syfte att ge underlag för hypotesformulering, snarare än att formulera och bekräfta hypoteser. Interaktiv utveckling har dessutom praktiserats under såpass kort tid att några större erfarenheter inte hunnit sammanställas.

Jag vill här redan nu göra en avgränsning till skandinaviska förhållanden. Den är motiverad av ekonomiska skäl, naturligtvis, men även av kulturella. Jag tror det skulle stöta på allvarliga problem om man försökte överföra resultat från t ex en amerikansk undersökning direkt till svenska förhållanden. Sättet att arbeta i företagen, sättet att värdera sin insats, de värderingar som ligger bakom det amerikanska och

svenska arbetslivet är alltför olika för att man ska våga dra några alltför långt gående slutsatser av en jämförelse. Möjligen kan man göra mera rättvisande jämförelser t ex mellan de nordiska länderna. Men denna avgränsning innebär inte att jag alldeles utesluter värdefulla impulser från forskning från väsentligt annorlunda kulturer.

Jag gör också en avgränsning mot beslutstödssystem. Denna är visserligen inte skarp, men den innebär att jag inte studerar beslutstödssystem som beslutsstödsystem utan som interaktivt utvecklade personliga datasystem istället. Jag är t ex inte primärt intresserad av vilken effekt ett sådant utvecklat beslutstödssystem haft på organisationens beslutsfattande utan jag är istället intresserad av vilken effekt användarna anser att databehandlingen fått på deras arbete. I praktiken innebär det att jag inte speciellt utnyttjar befintlig forskning kring beslutsstödsystem. En metodologisk invändning är naturligtvis att de observerade effekterna är personberoende, men vem mer än användarna skulle kunna avgöra vilka effekter ett förändrat arbetssätt haft på deras arbetssituation?

Man kan inte studera eller ens observera ett fenomen utan att ha någon form av bakgrundsuppfattning. Den behöver visserligen inte alltid stämma ("Det hade jag ingen aning om", Aha-upplevelser), men den finns där icke desto mindre för att sätta in fenomenet i ett större sammanhang, att kunna sätta det i relation till vad jag redan sett, hört eller upplevt (De logiska positivisterna hävdar visserligen (von Wright, 1943) att det finns objektiva och subjektlösa "elementarupplevelser" som på något vis utgör den empiriska verkligheten). Den förväntade uppfattningen kallar hermeneutikerna *förförståelse* (Arbnor & Bjerke, 1977) och jag ska i fortsättningen använda detta begrepp. I dess egentliga innebörd är förförståelsen omöjlig att göra synlig på förhand. Den visar sig först i efterhand genom min tolkning av det upplevda fenomenet. Genom detta synliggörande blir förförståelsen transformerad till förståelse och kan medvetandegöras även för andra. Detta ser jag som en av forskningens viktigaste uppgifter. För att jag ska kunna synliggöra min och användarnas förförståelse måste jag delta i interaktiva systemutvecklingsprojekt.

Emellertid har jag inte blott en omedveten förförståelse när jag går in i denna undersökning utan även en medveten förhandsuppfattning om vad jag tror mig finna. I tidigare skrifter (Flensburg et al 1981) har jag och några andra forskare i Lund redovisat ett antal faktorer som vi tror påverkar utveckling och användning av datasystem. När det gäller interaktiv systemutveckling har jag senare redovisat tre huvudgrupperingar (Nissen m fl 1982), nämligen *organisationen*, *redskapet* och *tillämpningen*. Det förefaller ganska uppenbart att effekterna av sådan utveckling borde bli annorlunda i en stor, gammal och byråkratisk organisation än i en liten, ung och dynamisk sådan. Likaså borde effekterna vara beroende av det valda redskapet, hur pass lätt det var att lära sig, hur pass kraftfullt det var. Men detta måste sättas i relation till tillämpningen, det problem eller det arbete användaren vill utföra med hjälp av redskapet.

I stort sett samtliga redskap som går att bruka för interaktiv systemutveckling är avsedda för individuellt bruk i arbetet. Därigenom uppkommer en automatisk avgränsning till att studera effekter på den enskilde användarens arbete. Men givetvis ingår det studerade arbetet som en del i en organisations verksamhet och det kan således även vara motiverat att studera vilken effekt detta har på organisationens verksamhet som sådan. Men huvudintresset koncentreras till den enskildes arbete.

Vad gäller specifik tillämpning har det gjorts en hel del forskning inom beslutsstödsystem från i huvudsak organisationsteoretiska utgångspunkter. Däremot är det inte många som tagit utgångspunkt i den enskilde personens arbete och vederbörandes upplevelse av detta arbete vilket jag ser som ett intressant och forskningsbart problem.

Jag måste då redovisa min syn på organisationer. Jag ämnar studera någon form av användning av datasystem och betraktar datasystem som "formella arbetsbeskrivningar". Då förefaller det rimligt att betrakta även organisationen ur någon "arbetsaspekt". Arbete utförs av organisationen som helhet. Denna helhet består av ett antal samverkande enskilda arbetsuppgifter. I den klassiska organisationsteorin studeras hur och enligt vilka principer dessa uppgifter ska utformas för att organisationens arbete som helhet ska bli så bra och effektivt som möjligt. I moderna organisationsteorier läggs man alltmer vikt vid den enskilde arbetarens förmåga att ta egna initiativ, att utvecklas personligen etc. Detta tycker jag motiverar en avgränsning till att åtminstone i första skedet endast studera interaktiv systemutveckling på individnivå.

Vad gäller redskapen finns en väl utvecklad föreställning om hur användarvänliga program ska vara konstruerade, både från programspråk- och algoritmsynpunkt (Fitzsimmons & Love, 1978) och från användningssynpunkt (Hiltz & Turoff, 1978). Vad som för mig är av central betydelse är att möjliggöra tyst yrkeskunskap i tillämpningen. Detta bör leda till att det ställs stora krav på att de utvecklade systemen eller programmen skall vara lätta att ändra. Överhuvudtaget skall redskapet vara väl anpassat till gällande yrkesspråk.

Emellertid måste jag vara mycket försiktig när jag studerar redskapet eftersom jag själv är dataexpert och således hårt belastad med förutfattade meningar om datorers och datasystems djupgående inflytande på hela vår tillvaro. Dessutom kan jag förhållandevis mycket om tekniken och är inte skrämmd av den. Det kan t ex tänkas att redskapets faktiska utformning, utöver en viss miniminivå, inte har så speciellt stor betydelse. Vad som kanske talar för detta är undersökningarna om att det första programspråk en programmerare lär sig har stort och betydande inflytande på hans eller hennes uppfattning av andra programspråk (Shneidermann, 1980). Det sägs ju "en gång Fortran-programmerare, alltid Fortran-programmerare". Något liknande kan gälla för min uppfattning av redskapen. Om den som först demonstrerar ett redskap är positiv till det, är det risk att denna uppfattning färgar av sig på mig och att jag inte ser en del nackdelar.

3 METOD OCH DETALJERAD PROBLEMFORMULERING

Det problemområde jag har valt att studera är interaktiv systemutveckling/personlig databehandling, vilket innebär att den enskilde användaren själv med hjälp av någon form av applikationsgenerator utvecklar och använder sina system. I mina studier av detta område har jag valt två olika ansatser, nämligen att insamla erfarenheter av interaktiv systemutveckling och att själv delta i sådan. Båda typerna av ansatser kommer att vara explorativa, dvs ge underlag för hypotes- och ev teoriformulering snarare än att bekräfta hypoteser och teorier. Vidare har jag avgränsat mig till skandinaviska förhållanden och till de enskilda personer som nyttjar interaktiv systemutveckling i sitt arbete. I detta kapitel skall jag diskutera metodfrågor mera utförligt och så småningom komma fram till en mera detaljerad beskrivning av mitt valda problemområde.

3.1 Om att ställa frågor

Eftersom det är en vetenskaplig avhandling jag skriver, måste jag göra klart för både mig och läsaren vad jag avser med "vetenskap". Denna diskussion och precisering kommer att uppta första delen av detta kapitel.

Som en första ansats väljer jag att ställa en fråga som jag lärt mig av Stig Lindholm (1980). Den lyder helt enkelt: "Vad är vetenskap ett specialfall av?" Orsaken till att jag finner denna fråga fruktbar är att den sätter in vetenskapen i ett större sammanhang, den ger ett slags helhetsperspektiv på vetenskapen. Alternativet hade varit att ställa en fråga om vad vetenskapen består av och detta kommer jag att göra senare.

Såvitt jag kan finna är vetenskapen ett specialfall av att ställa och söka besvara frågor. Eftersom det är ett specialfall är det en speciell sorts frågor som ställs och en speciell sorts svar som sökes. Detta skall jag närmare diskutera senare, men låt mig här stanna ett tag vid proceduren "Att ställa en fråga". Heidegger har om frågande skrivit (1981, s 22):

"Varje frågande är ett sökande. Varje sökande har fått sin föregripande /föregående/ riktning från det söktas håll... Varje frågande har såsom frågande efter... sitt *efterfrågade* (sein Gefragtes). Allt frågande efter... är på ett eller annat sätt ett sig-förfrågande hos (Anfragen bei). Till frågandet hör således förutom det efterfrågade något *tillfrågat* (ein Befragtes)... I

det efterfrågade ligger då, såsom det egentligen intenderade, *det genom frågandet vunna resultatet* (das Erfragte), det i och med vilket frågandet når sitt mål."

För att tillämpa Heideggers sätt att förhålla sig måste jag i frågandet göra klart för mig

1) Vad är det *efterfrågade*? Det innebär att jag måste göra utgångspunkten för frågeställningen klar för att därigenom vara medveten om vilken föreställningsram jag presenterar att fylla i svaret. Jag måste också göra syftet med frågan klart för mig. Om jag t ex frågar hur många liter grön takfärg vi har inne, förväntar jag mig en viss typ av svar. Om jag frågar huruvida vi ska fortsätta sälja grön takfärg förväntar jag mig en helt annan typ av svar. I det första fallet är svaret kort och koncist, består av endast en sifferuppgift. I det andra fallet är det inte säkert att jag överhuvudtaget får något svar, frågan är kanske snarare en inledning till diskussion om sortimentet av takfärg.

2) Vem är *tillfrågad*? Genom att jag riktar frågan till en viss person eller viss vetenskaplig skola får jag en viss typ av svar, bl a beroende på den tillfrågades bakgrund och ställning. Om jag ställer frågan om hur många liter grön takfärg vi har inne till någon på lagret får jag ett svar som avser hur mycket som fysiskt finns där. Om jag ställer samma fråga till någon försäljare får jag snarare reda på hur mycket som är disponibelt för försäljning. Dessa båda siffror kan skilja sig åt beroende på t ex att en ny leverans väntas till i morgon eller att ett visst antal färgburkar redan är sålda.

3) Det genom frågandet vunna *resultatet* vilket kan vara något helt annat än svaret på frågan. Det kan vara en insikt om att frågan är fel ställd eller ställd till fel person eller har en uppsättning följdfrågor jag dittills inte tänkt på eller att vissa frågor kanske helt enkelt inte kan ställas. Ett resultat av frågan om mängden grön takfärg kan vara att vi snarast måste beställa hem mer sådan färg eller kanske tvärtom, att den har stått still så länge att det kanske är dags att rea det som finns kvar. Detta kan då leda till andra frågor, t ex den inledningsvis nämnda om sortimentet.

Jag har nu funnit tre nyttiga och kraftfulla typer av frågor i mitt frågande efter vetenskapen. Men Heidegger betonar att varje frågande är ett sökande. Detta innebär att frågorna måste ställas flera gånger och varje ny gång med hänsyn till det resultat jag uppnådde i förra frågeomgången. Jag får då en form av hermeneutisk spiral (fig 3.1). Resultatet av efterfrågandet kan ibland bli en ny spiral, vars resultat så småningom kan komma huvudspiralen till godo eller eventuellt bilda en helt fristående spiral. I detta kapitel skall jag beskriva min forskningsprocess i sådan form med ett antal olika varv i ett antal olika spiraler.

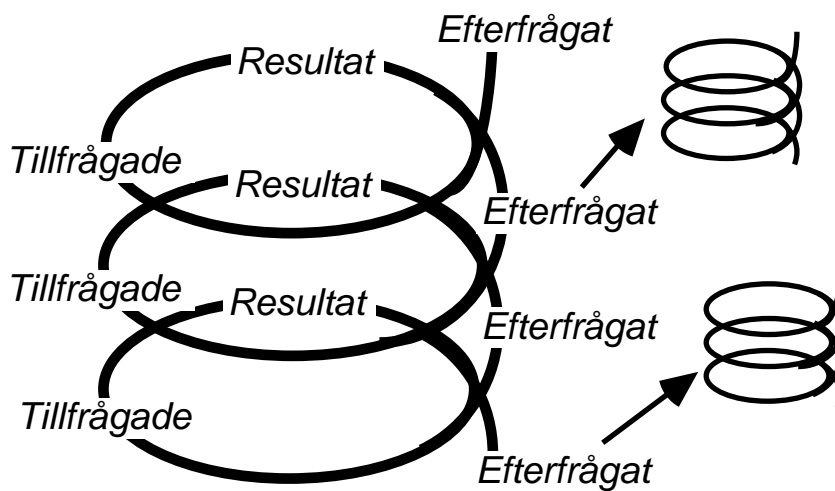


Fig 3.1. En variant på den hermeneutiska spiralen

I samband med frågans formulering måste jag också ta upp frågan om vilket språk den formuleras i. Om jag till exempel formulerar en fråga i matematiskt språk, får jag givetvis ett matematiskt svar. Men detta svar måste översättas för att kunna tillgodogöras. Törnebohm (1983) tar bl a upp frågan om språk i vetenskapliga sammanhang. Han formulerar följande centrala teser (sid 156-157):

- Det finns ett *formellt språk* och ett *naturligt språk* i varje teori.
- De språk som förekommer i en teori om ett verksamhetsområde X handlar om företeelser i X. De handlar inte om teorin.
- I en teoriutveckling finns det andra språk än de, som hör till en teori inom en sådan utveckling.

Det finns språk som används när resultat underkastas en kritisk granskning. Denna sortens språk *kontrollspråk* utmärks av att de innehåller kriteriebegrepp och att kritiska omdömen om teorin uttrycks i dem.

Vad beträffar språken i en teori hävdar Törnebohm vidare (s 14) att det finns tre olika sorters språk, nämligen en *schematism S*, ett *matematiskt språk M* och ett *naturligt språk N*. S och M ska dock uppfattas som schematismer och formella språk i vid bemärkelse och behöver inte ansluta till vad vi i strikt matematisk mening menar med schema och formalism. Schemata och formalismer är lämpliga att nyttja vid studier av föremål eller objekt. Men enbart studier av föremål är inte tillräckliga. Jag har också själv i andra sammanhang varit inne på hur lite jag egentligen kan uttrycka med strikt formalism (Flensburg, 1984f). Törnebohm hävdar att *det är möjligt att översätta det formella språket till det naturliga, men inte tvärtom* (sid 172173).

Att man måste använda ett annat språk när man talar om teorin än man använder inom teorin menar även Heidegger (1981, sid 24):

"... måste det genom frågandet vunna resultatet, i detta fall varats mening, kräva sin egen begreppslighet /begreppsapparat/, som även den väsentligt kontrasterar mot de begrepp inom vilka det varande uppnår sin betydelsemässiga bestämdhet."

Törnebohm hävdar, och även andra (Wittgenstein 1976, Israel 1979) att allt språk går att omvandla till vardagsspråk. Detta gäller även vetenskapliga språk. Den terminologi och den formalism man ofta använder inom vetenskapen är, enligt Törnebohm, bara ett instrument. De resultat jag uppnår måste formuleras på ett sådant sätt att de, i mitt fall, är relevanta för de personer som sysslar eller står i begrepp med att syssla med interaktiv utveckling av datasystem. Formalismen inom det vetenskapliga språket används för hantera begrepp inom det formella språket, men resultatet måste kunna översättas till vardagsspråket.

3.2 Att fråga om vetenskapen

3.2.1 Det efterfrågade

Med hjälp av den ovan angivna frågetekniken ska jag försöka komma en bit på väg i mitt sökande efter lämpligt angreppssätt. Jag börjar med att i detta inledande skede helt enkelt *efterfråga* vetenskapen, vad den är, vilket som är dess syfte och hur den bör bedrivas. Det jag vill *efterfråga* är än så länge vagt och diffust men jag föreställer mig att en första fråga skulle kunna vara vad en forskare gör när han eller hon bedriver vetenskap? Jag tror då att vetenskapen definieras genom sitt utövande vilket, åtminstone delvis, är beskrivbart.

3.2.2 Den tillfrågade

Den jag först *tillfrågar* är Swedner (1982) och han säger (s 9) att vetenskapsmannen vill beskriva den värld vi lever i, han vill förklara varför den ser ut på ett visst bestämt sätt, han vill förutsäga vad som kommer att hända och slutligen vill vetenskapsmannen föreslå åtgärder som leder till ett förverkligande av utstakade mål. *Resultatet* av denna fråga är att svaret inte verkar vara tillfyllest eftersom ju alla aktiviteterna ovan kan göras utan någon särskild vetenskaplig skolning. Swedners punkter är i själva verket goda övergripande mål för vilken utredning som helst. Swedner påpekar också längre fram, att vad som skiljer vetenskapsmannens försök till beskrivning, förklaring, förutsägelse och förslag från lekmannens är att han följer utstakade och noggrant prövade regler i sina försök att komma till hållbara resultat. En annan viktig punkt i Swedners resonemang är forskarens svårighet att vara värdeneutral.

Om detta svar vore riktigt skulle nyskapande inom vetenskaplig metodik vara omöjlig och vetenskapliga undersökningar vara likartade under seklernas lopp. Detta är dock

inte fallet. Kuhn (1981) var den som först införde termen *paradigm* (förebild, mönster) för att i detta sammanhang beskriva genomgripande vetenskapliga förändringar. Swedner är medveten om detta och menar längre fram (s 13) att "samhällsvetenskaperna... ständigt ställs inför uppgiften att studera en 'ny verklighet' och att producera kunskap som är relevant för människor som lever i denna 'nya verklighet'".

Lindholm (1979) definierar: "*paradigm är en uppsättning normer eller föreställningar som i ett visst forskarsamfund styr vad man ska forska på och hur det ska göras*" (s 55 f). Det är viktigt att här lägga märke till bindningen till forskarsamhället. Både Kuhn och Lindholm betonar den starkt. Det påminner om det fenomen som Berger & Luckmann (1968) diskuterar när de behandlar institutionalisering av organisationer.

3.2.3 Resultatet

Jag har kommit fram till att frågan om vetenskapen också måste innefatta villkoren för vetenskapens bedrivande, dvs även paradigmatiska frågeställningar. Jag har också kommit fram till att forskaren sysslar med att beskriva, förklara, förutsäga och ge förslag till förändringar. Detta görs enligt regler och procedurer som ingår i ett paradigm. Detta paradigm är inte värdeneutralt. Min ursprungliga fråga om vetenskapen har jag besvarat genom att ange en uppsättning olika delaktiviteter, vilka sammantagna bildar den totala aktiviteten. Delaktiviteterna kan naturligtvis i sin tur delas upp i ytterligare delaktiviteter, som i sin tur delas upp etc. På så sätt får vi en atomistisk beskrivning av vetenskapen. Men det är inte alltid en sådan beskrivning är lämplig. Ett fall då den inte är det, är när det *efterfrågade* utgör en helhet.

3.3 Om vetenskapen som en helhet

3.3.1 Det efterfrågade

I detta senare fall (då det *efterfrågade* utgör en helhet) måste själva frågeställningen bli en annan. Ett sätt är att istället för att direkt fråga "Vad är vetenskap?" mera indirekt fråga "Vilka typer av vetenskap finns det?". Genom att ange vilka olika typer som finns fokuserar jag intresset på olika aspekter som tillsammans bygger upp en mer komplett bild av forskning.

3.3.2 De tillfrågade

I en tidigare upplaga säger Swedner (1970, sid 29):

"Traditionellt brukar man skilja mellan teoretiskt och empiriskt forskningsarbete. Med teoretiskt menas då sådant arbete som en forskare kan utföra vid sitt skrivbord med hjälp av papper och penna och med tillgång till böcker och forskningsrapporter. Med empiriskt arbete avses arbete på fältet och manuell eller maskinell bearbetning av det insamlade materialet."

Denna uppdelning tar han delvis tillbaka i nästa mening där han säger "... att i faktiskt forskningsarbete är empiriskt och teoretiskt arbete sammantvinnat". Eftersom utgångspunkten (det *efterfrågade*) var en helhet låter det lovande. Vad man kan invända mot är att empirisk forskning skulle bestå enbart av datainsamling. Detta har emellertid Swedner (1982) utvecklat vidare och inför två olika paradigmer med forskaren som *åskådare* och forskaren som *aktör*. Forskaren som är åskådare anser sig vara neutral och värderingsfri medan aktörforskaren tar ställning för någon part i forskningsområdet.

Törnebohm (1983) tar också upp frågan om olika typer av empiriska undersökningar. Han baserar sig därvid på den av honom föreslagna modellen för vetenskapliga undersökningar, det Löpande Forsknings Paradigmet. Enligt detta kan empiriska undersökningar delas upp i *kunskapsutvecklingar* (sid 8):

- *Kunskapsutveckling av ordning (0)* , som producerar data t ex i form av mätresultat
- *Kunskapsutveckling av ordning (1)*, som producerar enstaka kunskaper dvs bekräftade hypoteser
- *Kunskapsutveckling av ordning (2)*, som producerar förklaringsmönster
- *Kunskapsutveckling av ordning (3)*, som producerar speciella teorier
- *Kunskapsutveckling av ordning (4)*, som producerar fundamentala teorier

Det här förelagda forskningsproblemet handlar om tre olika sorters kunskapsutvecklingar. För det första beskriver jag vad som har hänt vid interaktiv systemutveckling. Detta är kunskapsutveckling av ordning noll. Dessa primära resultat kan andra forskare arbeta vidare med. För det andra försöker jag ge någon form av förklaringsmönster, dvs kunskapsutveckling av ordning två. Det kan t ex innebära att jag försöker formulera ett typfall eller en rekommendation för att uppnå, i någon angiven mening, bra resultat av interaktiv systemutveckling. För det tredje kan det tänkas att jag kan producera speciella teorifragment som rör användarutveckling. Det skulle t ex kunna vara någon teori om hur en lämplig applikationsgenerator ser ut eller vilka typer av informationssystem som är lämpliga att användarutveckla. För det fjärde slutligen skulle det kunna tänkas att jag kan belysa eller åstadkomma nya infallsvinklar till fundamentala teoribegrepp inom vårt område.

Hur ska då en sådan undersökning läggas upp? Vilka komponenter finns däri? Törnebohm menar också att alla undersökningar, således även de av högre kunskapsordning måste innehålla följande komponenter (1983, sid 8):

- En samling *hypoteser* om sakförhållanden i undersökningsområdet
- *Problem* som forskaren vill undersöka
- En samling *instrument*, fackspråk, begrepp, tekniker etc med vars hjälp forskningen bedrivs.
- *Kritisk granskning* av resultaten utifrån en samling kriterier
- *Taktiska planer*, vilka är en övergripande forskningsplanering

Jag ska nu koncentrera mig på den tredje punkten, dvs instrument, fackspråk o dyl för att bedriva forskning. I föregående avsnitt var jag inne på ett instrument, nämligen språket. Ett annat instrument är *metod*. Aubert (1969) definierar metod sålunda:

"En metode er en framgangsmåte, et middel til å løse problemer og komme fram til ny kunnskap. Et hvilket som helst middel som tjener dette formålet, hører med i arsenalet av metoder.

Detta är en tämligen vid definition av begreppet "metod". Hellevik (1980) koncentrerar sig på vad hon kallar "kvantitativa metoder", vilka hon definierar sålunda (sid 8):

"Med dette menes framgangsmåter der forskaren først systematisk skaffer seg sammenlikbare opplysninger om flere undersøkelseobjekter av ett visst slag, så uttrykker disse opplysningene i form av tall, og til slutt foretar en analyse av mønsteret i dette tallmateriel

Hellevik anger också fem olika normer, som hon anser bör vara vägledande för god forskning (ibid, sid 9-11):

1. *Överensstämmelse med verkligheten* är det högsta sanningskriteriet.
2. Så *objektivt* bruk av data som möjligt.
3. Presentation av resultaten som tillåter *kontroll, efterprövning och kritik*.
4. *Systematiskt urval* av data.
5. Strävan till *kumulativ* forskning.

Metodläran är enligt Hellevik ett stöd för att uppnå dessa mål. Hon vill dock på intet vis monopolisera dessa regler såsom giltiga för alla typer av forskning (sid 12):

"Ingen bør imidlertid sitte igjen med det intrykk at metodelaeren består av absolutte og uforanderlige sannheter. Til ulike tider og på ulike steder har en hatt forskjellige oppfattninger av hva forskning er og hvordan den bør drives."

Hellevik diskuterar dock inte alls sambandet mellan problem och metod. Hon tycks mena att om ett problem inte kan angripas med de ovan angivna metoderna är det inget vetenskapligt problem.

Jag ska nu undersöka i hur stor utsträckning Helleviks kriterier är tillämpbara på mina föreslagna undersökningar. Det första kriteriet indikerar att det finns en objektiv, mätbar verklighet. Som jag antydde redan i inledningen tror jag inte det generellt sett är så. Såvitt jag kan finna är Swedner (1982) av samma åsikt. Däremot kan man säkert i en praktisk situation komma överens om att så och så är nu gällande (*intersubjektivitet*, jfr Berger & Luckmann, 1968). Under dessa premisser är det möjligt att tillämpa det första kriteriet. Så torde också vara det vanligaste sättet att idag bedriva samhällsvetenskap.

Det andra kriteriet indikerar att forskaren är objektiv och neutral. Men jag menar att detta kan inte förutsättas utan vidare (Bergström 1976, Swedner 1982). Människan är dock i stor utsträckning en stängd bok, även för sig själv. Det finns värderingar som jag inte är medveten om och därigenom påverkar min bedömning och mitt bruk av data. Därför vill jag i mina undersökningar ersätta objektivitetskriteriet med ett ärlighetskriterium. Detta innebär att jag, så gott det nu är möjligt, redovisar mina utgångspunkter och min bakgrund innan jag samlar in och använder data.

Det tredje kriteriet kan jag inte heller alltid tillämpa. En intervenering i en organisation t ex kan aldrig göras om. Det går inte heller att kontrollera denna intervenering på samma sätt som man kan kontrollera ett laboratorieexperiment. Däremot tycker jag att resultatet ska redovisas så öppet och ärligt att en meningsfull kritik är möjlig, både av själva resultatet men också av de utgångspunkter och tillvägagångssätt som ligger till grund för det.

Det fjärde kriteriet förutsätter att det finns något system att systematisera efter. Detta system utgörs av teorin. Om nu problemet innefattar kunskapsutveckling av högre ordningar än ett finns inte säkert denna teori given från början och då blir det svårt att leva upp till detta kriterium. Men samtidigt förefaller det ganska meningslöst att helt utan någon som helst utgångspunkt samla in data. Jag skulle vilja återföra detta till mitt tidigare föreslagna ärlighetskriterium och koppla systematiken i datainsamlingen till de utgångspunkter jag där redovisar. Men det innebär att systematiken inte får bli ett självändamål, att man ska ha rätt att sätta sig över den, om förhållandena så kräver. Det åligger en ärlig och uppriktig forskare att avgöra när så är fallet.

Det femte kriteriet, innebär att man ska sträva efter att bygga på forskning som redan finns. Detta gäller för normalforskning enligt Kuhn. I ett revolutionärt skede gäller snarare omvända regeln (Feyerabend 1972). Problemet för den enskilde forskaren är förstas att avgöra i vilket skede han eller hon befinner sig.

3.3.3 Resultat

Om de ovan redovisade kriterierna tillämpas på mitt förelagda problem leder det till att jag i förväg måste redovisa mina utgångspunkter mycket noga och samtidigt hela tiden vara beredd att revidera. Enligt Törnebohm ska jag nu även hitta lämpliga in-

strument för detta. Det leder mig till att närmare studera vad som avses med empirisk forskning.

3.4 Empirisk forskning

Empirisk forskning kan enligt Hellevik underindelas i *hypotesprövande* undersökningar, *experimentell* forskning och *aktionsforskning* (fig 3.2). Hypotesprövande undersökningar bedrivs i regel med någon form av statistisk verifiering och kommer, enligt min mening, in på det stadium i forskningen, då en teori redan har blivit så formaliserad att det är möjligt att inom dess ram formulera mätbara konsekvenser och sätt att utföra mätningar.

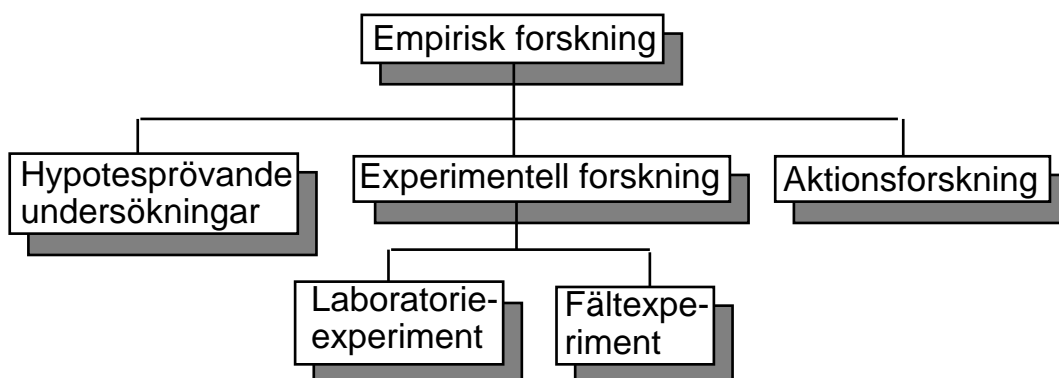


Fig 3.2 Olika slags empirisk forskning

Inom informationsbehandlingen kan hypotesprövande forskning användas t ex vid undersökningar av olika datasystems effekter på väl avgränsade arbetsresultat. Det kan handla om att studera datasystem i användning och försöka konstatera vissa kritiska faktors betydelse för det färdiga resultatet.

Experimentell forskning kan bedrivas i laboratoriemiljö eller fältmässigt. Vid laborativ forskning kan man kontrollera alla variabler, som anses relevanta, medan detta vid fältmässiga experiment inte är möjligt. Resultaten från ett fältmässigt experiment kan verka mindre pålitligt, men man ska tänka på att det är en konstlad omgivning som används i ett laboratorium. Kvaliteten i mätningen blir därför i detta fall måhända t o m sämre. Fältmässiga experiment används inom informationsbehandling då man t ex vill undersöka effekterna av ett visst tillvägagångssätt vid systemutveckling. Skillnaden gentemot de hypotesprövande undersökningarna är att man här går genom ett eller ett fåtal fall och söker visa på det typiska för dessa fall, snarare än det gemensamma. Det är alltså skillnader inte bara i tillvägagångssätt utan även i den typ av kunskap man söker.

3.4.1 Aktionsforskning

I Sandberg m fl (1978) redogör Abrahamsson och Swedner för begreppet *aktionsforskning*. Det går inte att definierar på ett enkelt och entydigt sätt utan står för lite olika saker bland skandinaviska och anglosachsiska forskare. Abrahamsson & Swedner menar att aktionsforskning är:

"ett målinriktat och planerat förändringsarbete i syfte att förbättra förhållandena för resursfattiga och underprivilegierade grupper och individer baserat på fältexperiment och noggrann utvärdering av utfallet av dessa fältexperiment. Inom aktionsforskningsorienterat förändringsarbete har man under senare år i växande grad därutöver kommit att lägga stor vikt

- 1) vid återföring av resultat från fältexperiment och utvärdering till de grupper som drabbas av de missförhållanden som man vill eliminera.
- 2) vid att dess grupper för möjlighet att utöva inflytande på besluten om mål och arbetsmetoder.
- 3) vid medvetandegörande och mobilisering av dessa grupper och
- 4) vid att arbete med att främja resurssvaga gruppers intressen drives med kraft, även om detta leder till konflikter med dem som gynnas av systemet."

Abrahamsson och Swedner redogör för olika traditioner inom aktionsforskningen. Den första är den *human-teknologiska traditionen* som började med fältexperimenten på Western Electric i slutet av 1920-talet. Typiska drag är stor vikt vid fältexperiment och noggrann utvärdering av dessa. Man utgår från företagsledningen, men betonar att den måste ha god kunskap om vad de anställda tänker, känner och gör för att få ett bra förändringsarbete.

Kurt Lewin representerar, enligt Abrahamsson och Swedner, en annan tradition inom aktionsforskning, nämligen *återföringstraditionen*. Typiskt för denna är att forskarna solidariserar sig med de resurssvaga grupper som drabbas av förändringarna i samhället. Men fortfarande bygger idén på fältexperiment och utvärdering av dessa. Det nya är återföringen av resultaten.

Nästa tradition, som Abrahamsson och Swedner tar upp är *samhällsarbetartraditionen*. Den har sin grund i Storbritanniens program för att påskynda social, ekonomisk och demokratisk utveckling i sina kolonier. Det man starkast betonar är att resurssvaga grupper ska få delta i beslutsprocessen kring förändringsarbetet och att man inte skyggar för de ofrånkomliga konflikterna. Abrahamsson och Swedner uppehåller sig mest vid detta perspektiv och tar upp fyra teman:

- Partecipations-resonemanget
- Mobiliseringsresonemanget
- Medvetandegöringsresonemanget
- Konflikten som instrument i förändringsarbetet

Participation (medbestämmande) innebär att de som berörs av förändringar ska vara med i beslutet om och utformningen av förändringen. Mobiliseringsresonemanget innebär att outnyttjade resurser hos de missgynnade grupperna tas i anspråk. "Hjälp till självhjälp" är ett vanligt talesätt. För att detta krävs en *förändringsagent* (change agent), som hjälper i mobiliseringsprocessen. Innan mobiliseringen kan ske, behövs ofta en medvetandegöring om vad som håller på att ske. Denna medvetandegöring sker genom utbildning, antingen i frigörande syfte (Freire 1972) eller genom "avlärning" av gamla fördomar och/eller traditioner (Illich 1973). I denna process uppkommer, enligt Abrahamsson och Swedner, alltid konflikter mellan underprivilegierade och privilegierade.

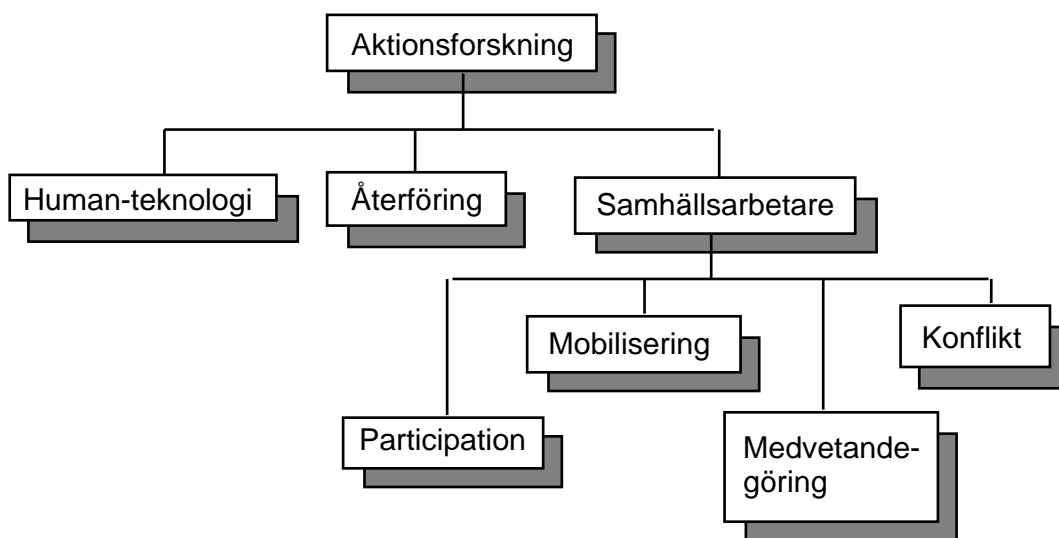


Fig 3.3 Olika typer av aktionsforskning enligt Abrahamsson & Swedner

Susman & Evered (1978) tar upp aktionsforskning som ett alternativ till traditionell forskning. Man citerar Lewin (1946), som beskriver aktionsforskningsprocessen sålunda: "a spiral of steps, each of which is composed of a circle of planning, action and factfinding about the result of the action" (Fig 3.4). De hänvisar till Rapoport (1970, s 499) angående definition av aktionsforskning:

"Action research aims to contribute both to the practical concerns of people in an immediate problematic situation and to the goals of social science by joint collaboration within a mutually acceptable ethical framework"

Vad som vidare kännetecknar aktionsforskning, enligt Susman & Evered är

- Den är framtidsorienterad
- Den innebär samarbete med klientorganisationen
- Den innebär utveckling av klientorganisationen
- Den genererar teorier, som grundas på praktiska handlingar

- Den reviderar fortlöpande sina förutsättningar
- Den är situationsanpassad

Aktionsforskning har inom informationsbehandling-ADB bedrivits enligt några olika linjer. Den human-teknologiska traditionen är ganska spridd genom forskning som baseras på Tavistock-institutets arbeten i början på 50-talet och i USA genom sådana organisationer som Center for Quality of Working Life vid University of California, Los Angeles. Det brukar kallas socioteknisk forskning. Återföringstraditionen har inte blivit speciellt artikulering, såvitt jag vet, medan däremot en hel del forskning bedrivs enligt samhällsarbetartraditionen. Mumford (1983a och b, Mumford & Henshall 1979) står för en participationsinriktad gren, medan fackföreningsorienterad forskning, t ex Nygaard (1974), Ehn & Sandberg (1979), Göranson (1978, 1983a, b) mer bedrivs enligt ett konfliktperspektiv. Den forskningsgrupp jag själv tillhör, LUIS-gruppen (Lunds Universitets forskning kring Information System) bedriver också aktionsforskning med tonvikt på mobilisering och medvetandegöring (Sandström m fl 1983, LUIS-gruppen 1985).

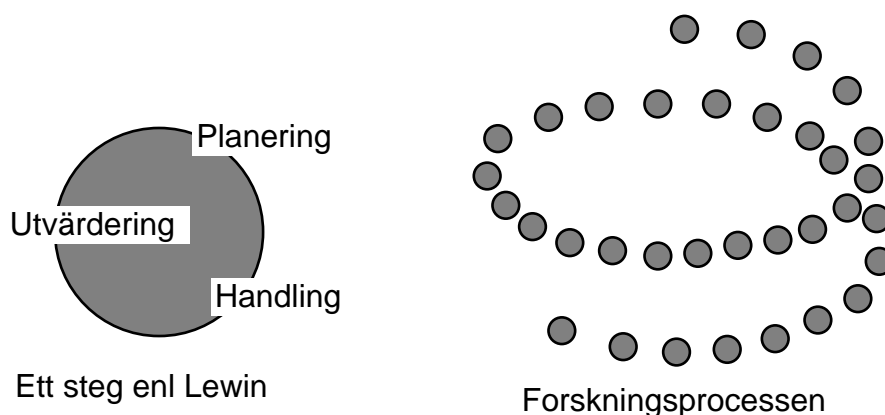


Fig 3.4 Aktionsforskningsprocessen enligt Lewin

Detta gör att jag finner aktionsforskning vara lämplig i den andra delen av mina undersökningar. Men hur ska den genomföras? Lewin har ovan (fig 3.4) givit en del anvisningar. Susman & Evered föreslår en liknande ansats (fig 3.5).

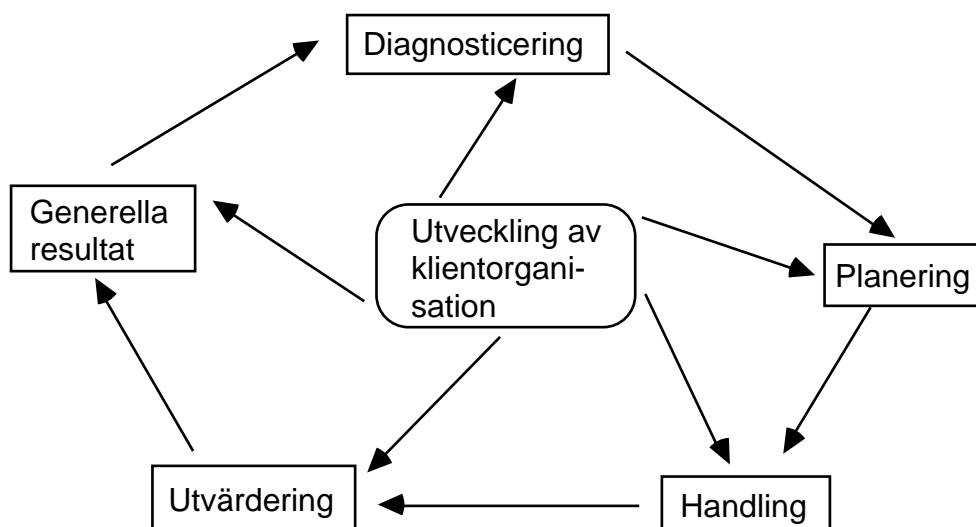


Fig 3.5 Aktionsforskningsprocess enligt Susman & Evered

Båda modellerna visar på den cykliska strukturen. Mitt syfte är i detta sammanhang att skaffa mig kunskap om interaktiva systemutvecklingsprocesser. För den sakens skull blir det en starkare betoning på "generella resultat" än vad som kanske är vanligt inom aktionsforskning. Under de olika rubrikerna kan man tänka sig följande delaktiviteter:

- Diagnostisering
 - *Kontaktsökande*
- Planering
 - *Övergripande samtal*
 - *Grundläggande utbildning, medvetandegöring*
- Handling
 - *Initiering av interaktivt systemutvecklingsprojekt*
- Utvärdering
 - *Intervjuer*
- Generell kunskap
 - *Sammanställning av erfarenheter*

3.4.2 Övriga typer av forskning

Innan jag fortsätter ska jag bara kort konstatera att alla dessa olika former av forskning mycket väl kan kombineras i ett och samma projekt. De olika undersökningsformerna är komplementära, inte uteslutande. Vidare är det sällan möjligt att renodla dem på det sätt som jag försökt här göra. Min genomgång ovan är ett idealtypiskt resonemang, där jag försöker renodla de enskilda dragen i varje inriktning. I praktiken är det sällan möjligt att göra så.

Man kan också göra en indelning med avseende på syftet med forskningen. Detta har Törnebohm varit inne på (t ex ifråga om kunskapsutveckling). Med enkla ord kan man sammanfatta syftet i följande tre huvudtyper:

- Explorativa undersökningar
- Hypotesprövande undersökningar
- Kritiska undersökningar

En explorativ undersökning görs då man inte vet speciellt mycket om forskningsområdet och inte har någon direkt teori att bygga på. Avsikten med en sådan undersökning är att ge underlag för formulering av hypoteser, vilka kan testas i hypotesprövande undersökningar. Resultatet av detta blir någon form av teori. Denna teori och de resultat den leder till granskas vid de kritiska undersökningarna. Om vi sätter in dessa typer av undersökningar i den paradigmatiske tidsföljden enligt Kuhn (fig 3.6), borde det under ett paradigms inledande skeden vara en viss övervikt på explorativa undersökningar. Under normalforskningsperioden borde det vara övervikt för hypotesprövande och under slutskedet borde det vara övervikt för kritiska. I mitt resonemang har jag hittills följt detta schema. Jag har kritiserat det traditionella systemutvecklingsparadigmet och hävdar att det innehåller viss fundamentala brister. Här föreslår jag en explorativ undersökning, som kanske kan leda till något alternativt systemutvecklingsparadigm. Det jag i så fall måste företa mig är en mera detaljerad problemformulering.

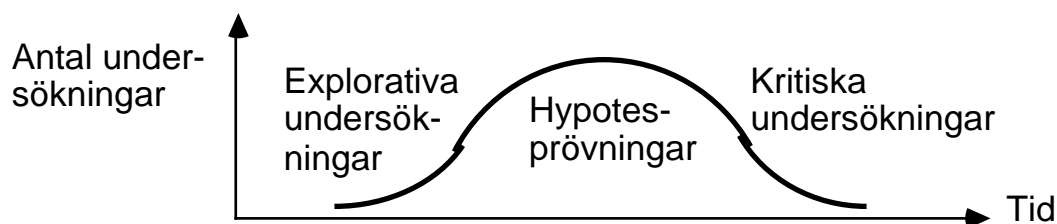


Fig 3.6 Olika typer av undersökningar under ett paradigms livscykel

3.5 Detaljerad problemformulering

Som utgångspunkt för den explorativa undersökningen väljer jag en teorigrund som baseras på min förståelse av forskningsområdet. Jag anser alltså att för den teoretiska grunden ska gälla:

- Den ska innehålla delar som handlar om effekter av förändringar i en verksamhet.
- Dessa förändringar skall ske med så stor användarmedverkan som möjligt.
- Förändringarna skall ha med införande av datasystem att göra.

Tidigare har jag gjort en del avgränsningar och även motiverat varför de är gjorda. Här ska jag ytterligare precisera själva problemet och även komma in på användbara teorier och föreställningar från angränsande områden.

Alla datasystem finns i någon verksamhet (fig 3.7). Denna verksamhet bedrivs med ett visst bestämt syfte, vilket kan kallas organisationens mål, affärsidé eller existensberättigande. I verksamheten finns människor. För att de skall kunna göra sitt jobb krävs kunskap om verksamheten, både ren yrkeskunskap och kunskap om dess övergripande syfte. Den sistnämnda behövs för att fatta omdömesgilla beslut. Allt detta kallar jag *verksamhetskunskap*.

Denna kunskap är delvis omedveten. Och även om den är medveten är den ofta omöjlig att beskriva eller verbalt förmedla. Det rör sig ofta om oskrivna regler av samma slag som de regler vilka styr vårt språk. Vi vet inte att vi följer en regel förrän vi bryter mot den (Israel, 1979). Ett exempel jag kan anföra rör sig om rabattsättning av kunder och följande lilla berättelse, hämtad ur en av mina tidigare forskningsrapporter (Flensburg, 1979), illustrerar detta. Bakgrunden är att konsulten vid systemutvecklingen fick reda på att varje kund hade en bestämd rabatt. Någon annan rabattsats eller typ av rabatt fanns inte, enligt vad konsulten fick reda på. Vid ett besök på firman strax efter systemstarten berättar konsulten följande:

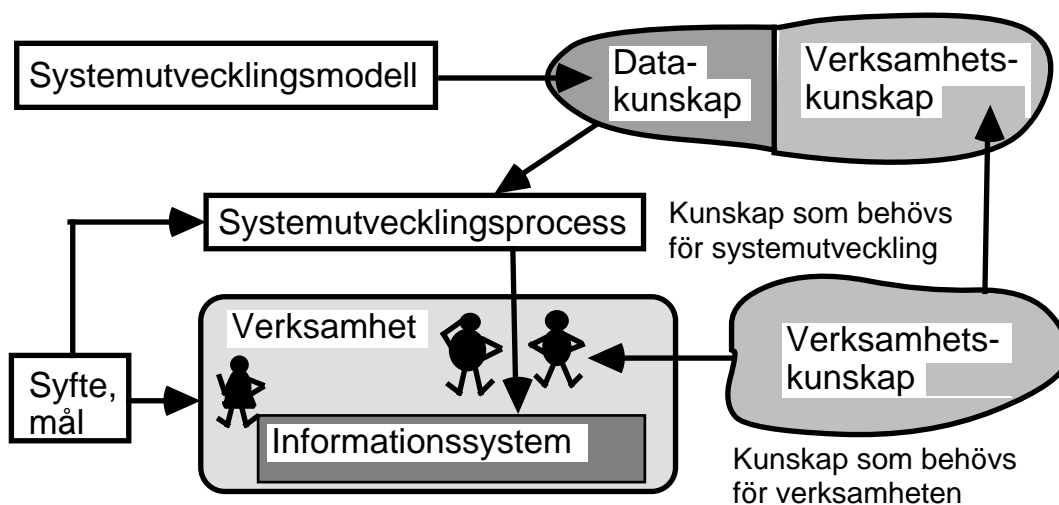


Fig 3.7 Forskningsområde för administrativ databehandling

"Det visade sig att någon kund på slätten skulle ha 15% extra rabatt för att han tog hem en hel bunt. Detta hade han inte fått. Cecilia (VD) var alldeles uppskärnad och beordrade en genomgång av samtliga fakturor för att se om samma fel fanns på flera andra. Efter diskussion enades jag och Bertil (försäljningschefen) om att det bara var kunderna på slätten och egentligen bara vissa av dem som var aktuella för just detta problem. Cecilia ville

ha en trippelfnatt på dessa kunder i kundregistret, men varken jag eller Adam (datachefen) var vidare pigga på det. En konstruktiv lösning, som de flesta tänkte på var att istället skicka två buntar fraktfritt och då skulle två buntar motsvara minst två ton. Firman skulle tjäna på det (trodde Cecilia) och det skulle bli enkelt att lägga in ..."

Berättelsen visar tydligt att det åtminstone i den här firman fanns kunskap om yrkesområdet (specialrabatter) som visserligen är fullt beskrivbar, men där villkoren för deras givande inte kunde klart anges. På grund av detta fick verksamheten anpassa sig till datasystemet!

Syftet med traditionell systemutveckling är att påverka verksamheten så att dess syfte eller målsättning uppnås i större utsträckning och/eller med mindre resursförbrukning. Detta kräver att de personer som sysslar med systemutveckling måste ha goda kunskaper om den verksamhet datasystemet ska ta över när det är klart. Det krävs att systemutvecklarna är lika goda yrkesmänniskor som människorna i den aktuella verksamheten. Följaktligen måste de även behärska den omedvetna, obeskrivliga och tysta verksamhetskunskapen. Detta är den viktigaste förutsättningen för att kunna göra bra datasystem, men är i praktiken omöjlig att uppnå.

Det finns då principiellt två olika vägar att gå. Den ena är att begränsa den verksamhet datasystemet ska ta över till att enbart omfatta normaltransaktioner. Undantagen, vilka i större utsträckning kräver tyst yrkeskunskap, behandlas manuellt. En viss procent normala transaktioner behöver möjligen behandlas manuellt för att handläggarna ska bibehålla känslan för vad som är normalt. Detta tillvägagångssätt har föreslagits av Nissen (1985b). Det andra sättet innebär att användaren istället ska bli systemutvecklare och ta över de delar av systemutvecklingen som kan klaras av med hjälp av en applikationsgenerator. Det är detta sätt jag studerar i dessa undersökningarna. Men fortfarande behövs systemutvecklaren för besvärliga och komplicerade projekt.

Man skulle kunna tro att kunskap om den verksamhet i vilken datasystemet ska användas anses vara önskvärd för systemutvecklare och andra dataexperter. Så verkar emellertid inte alltid vara fallet. Jag slog upp en sida (38) på måfa i ett nummer av tidskriften Nordisk Datanytt (nr 2, 1984) och fann där följande belysande formuleringar i platsannonser:

"Du skall ha hand om installation och underhåll av MVS/SP och JES 2. Vi vill att du besitter goda kunskaper i SMP och ASSEMBLER.

...

Din uppgift blir att införa NCCF/NPDA hos oss. Du skall ha goda kunskaper i VTAM/NCP"

I en annan annons i samma tidning ska en AU/ADB-konsult kunna:

- AR-planering, organisation och utveckling av ADB-verksamhet
- systemutvecklingsmetodik, projektledning och projektadministration
- effektivitets- och lönsamhetsfrågor beträffande ADB-verksamhet "

Det finns dock datafolk, som är medvetna om att det av systemutvecklarna även krävs yrkeskunskap inom användarnas verksamhetsområde. En av dem, Kurt Gladh, datachef på Atlas Copco, säger i Computer Sweden, nr 6 1984:

"En bra systemman ska kunna lika mycket om t ex lager, kundreskontra som de direkt ansvariga i ett företag. Det är affärer vi sysslar med det måste stå i förrummet."

Att det är dålig kommunikation mellan systemutvecklare och systemanvändare tror jag beror på dels att det rör sig om kommunikation mellan två yrken med varsitt yrkesspråk, där kommunikationen sker i yrkesspråken, dels på de ojämlika maktförhållandena.

I fig 3.8 är det traditionella forskningsområdet inom systemutveckling markerat. Det omfattar systemutvecklingsmodeller (tagna i vid bemärkelse), studium av systemutvecklingsprocesser (även dessa i vid bemärkelse) samt studium av vilken datakunskap som är behövlig för att bedriva systemutveckling.

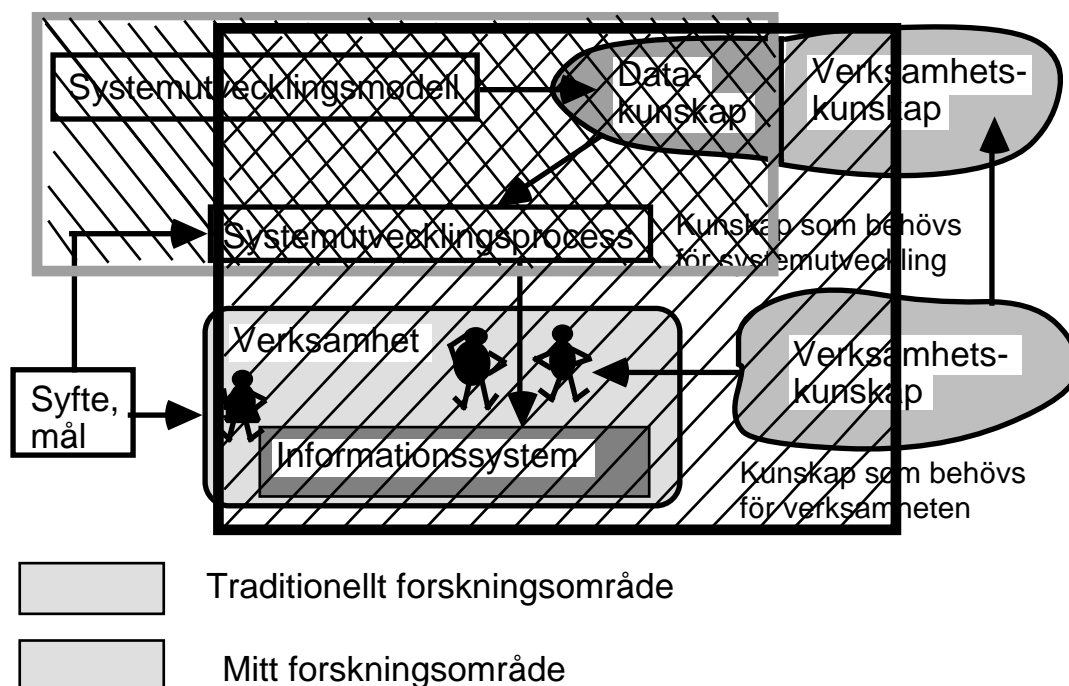


Fig 3.8 Problemområden

Det här aktuella området är också markerat. Det omfattar delar av det traditionella men även användning av informationssystem i verksamheter och delar av behövlig verksamhetskunskap. Allt sett från användarnas synpunkt.

3.5.1 Föreställningar om sakförhållanden inom forskningsområdet

Mina föreställningar om forskningsområdet ska jag försöka formulera i ett antal teser. En tes tar jag som utgångspunkt, medan hypotes är ett av teserna härlett påstående, som är formulerat i empiriskt prövbar form. Om hypotesen kan verifieras bidrar detta till den empiriska verifieringen av tesen (von Wright, 1943 sid 123 ff).

Den mest betydande tesen är att en önskad bättre anpassning mellan datasystem och verksamhet kan åstadkommas genom att användarna får större inflytande över systemutvecklingsprocessen. Därigenom har verksamheten bättre förutsättningar att uppfylla sina mål. Maximalt inflytande uppnås då användarna själva gör sina system. Detta är idag möjligt (eller påstås vara möjligt) genom användandet av applikationsgeneratorer. En något stringentare formulering av detta ser ut så här:

- (1)** *Det finns kunskap, som är av betydelse för verksamheten, men som inte går att formulera så att den är möjlig att överföra till datasystem (tyst kunskap)*
- (2)** *Traditionell systemutveckling tar inte speciell hänsyn till tyst kunskap.*
- (3)** *Tyst kunskap visar sig bl a genom kommunikationsproblem mellan systemutvecklare och systemanvändare.*
- (4)** *Existensen av det förhållande som nämns i tes (2) visar sig genom dålig anpassning av datasystemet till befintlig verksamhet.*
- (5)** *Ökad grad av användarmedverkan gör att problemen som följer av tes (1) reduceras.*
- (6)** *Maximal användarmedverkan innebär att användarna själva gör sina system.*
- (7)** *Tillgång till applikationsgeneratorer är en nödvändig men icke tillräcklig förutsättning för att användarna själva ska kunna göra sina datasystem.*
- (8)** *Tillgång till applikationsgeneratorer i en organisation gör att datasystem av användarna kan användas som hjälpmedel i helt nya tillämpningar.*
- (9)** *Om en person i sin verksamhet använder lämpliga applikationsgeneratorer får detta positiva effekter på vederbörandes arbete.*

Dessa teser bygger i stor utsträckning på varandra. Sammanfattningsvis kan sambandet mellan teserna illustreras i fig 3.9. Jag har här gjort ytterligare en uppdelning i så måtto som att teserna (1) - (4) behandlar traditionell systemutveckling, medan

teserna (5) - (9) behandlar interaktiv systemutveckling. Plus anger att sambandet är förstärkande, minus att det är motverkande.

En ökad användarmedverkan reducerar problem som uppkommer genom att man inte tar hänsyn till tyst kunskap i systemutvecklingsprocessen. Maximal användarmedverkan innebär interaktiv systemutveckling. Applikationsgeneratorer är en av flera förutsättningar för interaktiv systemutveckling. Dessa redskap ger även positiva arbetseffekter och möjlighet till nya tillämpningar.

Ingen av de ovan redovisade teserna kan anses vara verifierad i traditionell mening. Inte heller är de formulerade på ett sådant sätt att detta är möjligt. Man kan nu starta ett verifieringsarbete och börja verifiera dem i tur och ordning. Det kräver emellertid att verifieringskriterierna är beskrivna i en teori, som är mer omfattande än den som teserna är formulerade i. Att någon sådan skulle finnas och vara tillräckligt stringent för att ge underlag för verifieringskriterier förefaller osannolikt.

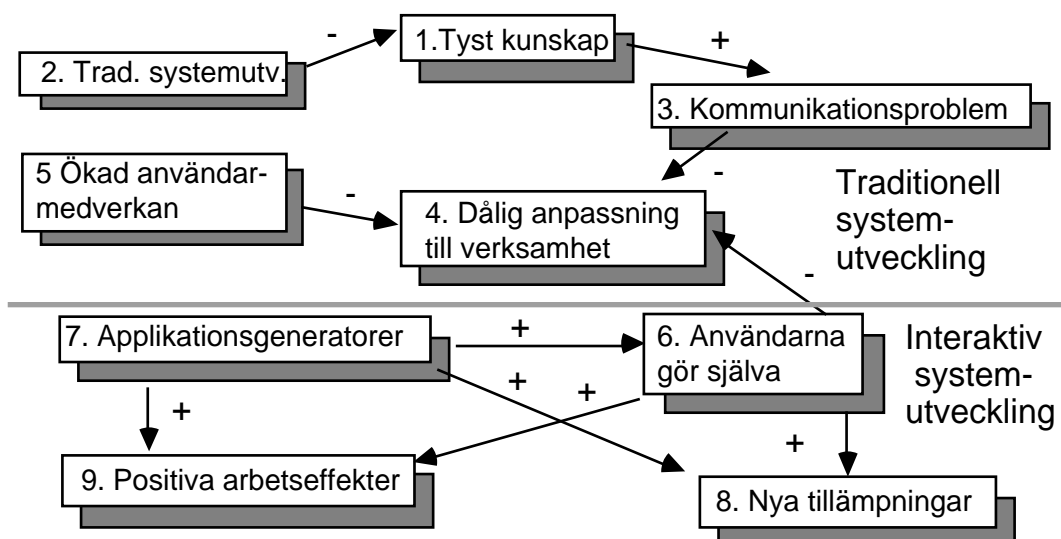


Fig 3.9. Samband mellan teserna

3.5.2 Verifieringskriterier

Jag måste således söka verifieringskriterier för den blivande teorin. Törnebohm (1983, sid 362) anger vad man ska göra i ett sådant fall som detta: "Om det saknas härledningskriterier för att lösa aktuella förklaringsuppgifter innehåller P (problemområdet) uppgifter att anskaffa, utveckla eller nybilda sådana forskningsinstrument." Hela detta kapitel kan ses som ett försök att göra just detta. Törnebohm ställer upp fyra kriterier för bedömning av teorier:

1. *Konfirmationskriteriet*, som innebär att alla hypoteser i teorin bör vara bekräftade. Det innebär speciellt att de måste själva vara empiriskt prövbara eller kunna relateras till empiriskt prövbara hypoteser.
2. *Koherensskriteriet*, som innebär att teser och utsagor i teorin bör stämma överens.
3. *Förklaringskriterium*, som innebär att teorin skall ha stor förklaringsförmåga, ju större desto bättre. Förklaringsförmågan är antalet förklaringsmönster som är möjliga att bilda inom teorin.
4. *Tillhörighetskriteriet* slutligen, innebär att hypoteser bör ingå i förklaringsmönster, att förklaringsmönster bör ingå i teorier och att teorier bör tillhöra familjer av teorier. Det är en form av kumulativitet som Törnebohm tar upp här.

Konfirmationskriteriet innebär att mina 9 teser bör vara empiriskt bekräftade på något vis. Den mest centrala är tes (1) och den är i viss mening bekräftad genom forskning utförd av Göranson (1978, 1983a, b) vid Arbetslivscentrum. Även Weizenbaum (1976), Cooley (1980) och Rosenbruck (1984) har bekräftat att så är förhållandet. Polanyi (1966) kallar motsvarande kunskap "Tacit knowledge". Halldén (1985) och Sandström (1985) pekar också på betydelsen av sådan kunskap. Att traditionell systemutveckling inte tar hänsyn till denna typ av kunskap (tes (2)) tycker jag forskarna ovan, speciellt Göranson har visat.

Teserna (3) och (4) är försiktigt formulerade. De säger endast att existensen av tyst kunskap kan påverka systemutvecklingsprocessen och resultatet av systemutvecklingen. De säger inte att detta är den enda orsaken utan bara en bidragande. Forskarna ovan har också visat dessa teser. Att de ovan angivna problemen reduceras genom en ökad användarmedverkan får nästan räknas till "de eviga sanningarna" vid det här laget. Låt mig här endast nämna Neergaard (1977), Bjørn-Andersen (1976), Borum (1976) och Høyer (1974), som alla kan anses ha visat detta.

Teserna (5) och (6) introducerar ett slags linjärt tänkande, nämligen att ju mer användarmedverkan desto bättre. I tes (7) sätts det upp en viktig förutsättning för interaktiv systemutveckling, nämligen förekomsten av applikationsgeneratorer. Teserna (8) och (9) slutligen, är andra föreslagna konsekvenser av teserna (7) och (6). Koherensskriteriet tycker jag är väl intäkt av mina teser ovan eftersom de i stor utsträckning bygger på varandra och utgör en gradvis avgränsning. Målet för min undersökning blir att undersöka det rimliga i att teserna (6), (8) och (9) följer av tes (7).

Om formuleringen av teserna (3) och (4) kan skärpas så att en stor del av de negativa effekterna av dålig kommunikation och dålig anpassning till arbetet kan anses bero på förhållanden angivna i (1) och (2) blir förklaringsförmågan mycket stor. Att visa detta förefaller dock svårt på grund av svårigheter att identifiera och renodla faktorer som påverkar kommunikation och verksamhetsanpassning. Om däremot (6), (8) och (9) kan prövas, uppnår jag en motsvarande ökning av förklaringsförmågan hos teorin och öppnar dessutom dörrarna till ett nytt fält. Det verkar alltså vara en vettig strate-

gi att satsa på en prövning av teserna (6),(8) och (9) istället för en ytterligare skärpning av teserna (3) och (4).

Tillhörighetskriteriet förefaller också vara väl tillgodosett eftersom prövningen av teserna (1) (5) baseras på tidigare forskning. Jag anknyter därmed till både fackföreningsinriktad forskning och till socioteknisk forskning.

I den taktiska planen ingår också planer för hur det uppnådda resultatet skall kunna omvandlas till nya forskningsproblem. Det resultat jag hoppas uppnå i denna första, empiriska undersökningen är beskrivningar över vad som hänt i ett antal fall då användarna själva gjort sina system. Denna kunskap kan brukas på olika sätt och det är väsentligt att jag gör klart för mig vilket syftet med kunskapen är.

3.6 Utgångspunkter

Jag skall nu redogöra för de utgångspunkter jag har inför mina båda undersökningar. Jag börjar med den där jag insamlar erfarenheter.

3.6.1 Insamling av erfarenheter

En teori jag valt som utgångspunkt är Enid Mumfords teori om arbetstillfredsställelse och som i sin tur grundar sig på Talcott Parsons forskningar i början av 50-talet (Mumford 1974,1983a, b). Mumford menar att för god arbetstillfredsställelse krävs god anpassning mellan ett antal faktorer hos arbetet och hos den som utför arbetet. Jag har redan räknat upp dem på sid 38. Ett arbete med hög arbetstillfredsställelse har hög grad av anpassning av alla dessa variabler.

En nackdel med Mumfords undersökningar är att hon tar utformningen av datasystemet som mer eller mindre given och sysslar mest med organisationen av det manuella arbetet kring det. Men lika väl som de enskilda arbetarna själva kan organisera sitt arbete, lika väl kan de med de moderna applikationsgeneratorerna utveckla även datasystemen.

Till en början har jag satt upp ett antal områden jag tror kan vara av betydelse för att förklara det jag kommer att upptäcka. Dessa områden avspeglar någon form av föreställning eller teori jag mer eller mindre medvetet har. Listan är ostrukturerad och upptar här endast korta stickord. I kapitel 4 kommer jag att mera utförligt motivera och beskriva de olika stickorden. Vid min undersökning studerade jag förhållanden kring följande områden:

- Applikationsgeneratorn
- Företaget som använder applikationsgeneratorn
- Historia
- Nuvarande användning av applikationsgeneratorn

- Påverkan på yrkeskunskap
- Påverkan på arbetsorganisation
- Psykologisk påverkan
- Påverkan på arbetsledning
- Social påverkan
- Tillgodoseende av etiska krav
- Ändringsbehov
- Förhållande till dataavdelningen (motsv)

Det är viktigt att även nya områden kan komma in. Den valda undersökningsmetodiken måste alltså vara så öppen att det är möjligt att upptäcka nya områden. Men samtidigt måste den vara så klar att det är möjligt att utesluta sådana som inte har haft någon effekt.

Det valda forskningsområdet: "Erfarenheter av interaktiv systemutveckling med hjälp av applikationsgeneratorer" är ett nytt område där det inte finns gjort särskilt mycket ur forskningssynpunkt. Jag har därför valt en explorativ undersökning, som kan ligga till grund för senare formulering av hypoteser. Men explorativa data kan behandlas enligt två perspektiv. Enligt det första perspektivet betraktar man personerna i huvudsak som objekt och ställer på förhand givna och strukturerade frågor till dem i syfte att få data för kvantitativ analys (åskådarparadigmet enligt Swedner 1982). På detta sätt kan man emellertid inte få någon kvalitativ kunskap utöver den man redan har sagt sig vilja ha. Kunskap av en annan typ, som kan medföra kvalitativa revisioner av sättet att se på forskningsområdet, erhålls i huvudsak genom interaktiva metoder (aktörsparadigmet enligt Swedner 1982). Det finns däremot inget som hindrar att jag kombinerar de båda perspektiven, dvs jag ställer ett antal på förhand givna frågor till personerna inom forskningsområdet, men är samtidigt öppen för nya infallsvinklar.

Rent praktiskt innebär det att datainsamling lämpligen sker med hjälp av intervjuer, ty först då kan det bli omdelbar återkoppling mellan den som intervjuar och den som blir intervjuad. Sidospår, som för den intervjuade ter sig skäligen betydelselösa, kan genast följas upp och ge värdefulla kunskapsstillskott. Härigenom får jag en öppen, men ändå förhållandevis stringent undersökningsmetod. Den kallar jag för *semi-strukturerad intervju*.

Det är viktigt att jag på förhand gör klart för mig vad jag tror och tänker om olika saker inom mitt undersökningsområde för att i görligaste mån bli medveten om vilket mitt huvudspår är. Som ett första led tänker jag mig följande :

- Utarbetande av frågor till de ovan angivna undersökningsområdena. Försök att klargöra vad jag har för avsikt med var och en av dem.
- Klargörande av hur jag tänker mig interaktiv systemutveckling gå till.

Utgångspunkterna för frågorna finns redovisade i kapitel 4. Interaktiv systemutveckling föreställde jag mig tillgå ungefär så här: En dataavdelning, har hand om större delen av företagets databehandling. På denna avdelning har man upptäckt vilken stor

nytta man kan ha av applikationsgeneratorer och det har bildats en grupp (IC) som håller i denna verksamheten. Det är handläggarna på avdelningarna som använder applikationsgeneratoren och som således är huvudpersoner i dessa undersökningar. De har haft hjälp av någon från IC-avdelningen med att starta användningen av applikationsgeneratoren. Kanske rent av denne har lagt upp det första systemet, som användarna nu ändrar eller kompletterar när de känner behov av det. Aktuell att intervjua är naturligtvis användaren, men även användarens chef och någon från dataavdelningen kan vara aktuell.

När jag gjorde frågorna utgick jag från de tidigare redovisade huvudområdena och formulerade ett antal frågor inom varje område. Totalt blev det ca 80 frågor, vilka även kom att utgöra en god dokumentation av min förförståelse. I redovisningen av resultatet från fallstudierna (kap 4) finns en detaljerad redovisning av varje fråga, både av vad jag trodde innan undersökningen började och hur den utföll.

De uppställda frågorna var inte avsedda att ställas ordagrant. Inte heller var det meningen att alla frågor alltid skulle ställas. En del kan vara irrelevanta beroende på företagets organisation. T ex är det inte säkert att företaget har någon dataavdelning och då är det inte meningsfullt att ställa frågor om förhållandet till denna obefintliga avdelning. Följaktligen måste den enskilde intervjuaren själv avgöra vilka frågor som är relevanta med hänsyn till det aktuella företagets organisation.

Jag insåg också ganska snabbt att jag själv inte skulle hinna med tillräckligt antal intervjuer. Därför har några forskare och studerande hjälpt mig med datainsamlingen. De har sedan själva använt det insamlade materialet i andra sammanhang, t ex som examensarbete (Persson 1985, Berg & Mattsson 1986). För att alla intervjuare skulle ha ungefär samma utgångspunkter gjorde jag anvisningar, där jag fråga för fråga redogjorde för varför just denna fråga ställdes och vad jag förväntade mig att få för svar (Flensburg 1984a). Detta ligger till grund för avsnitten "*Planerat*" i kap 4. Men trots att anvisningarna är utförliga finns det säkert massor av situationer och tillfällen jag inte tänkt på. Därför är det viktigare att intervjupersonen får sagt vad han eller hon vill säga än att alla frågorna besvaras.

Ordningen mellan frågorna är tämligen godtycklig i dessa anvisningar. Det hindrar inte att den i vissa fall kan vara mycket betydelsefull. Det finns något som kallas "halo-effekt" och som innebär att stämningen från en eller flera tidigare intervjufrågor inverkar på de följande frågorna. Detta kan ibland utnyttjas för att få svar på mera känsliga frågor.

Rent tekniskt gjordes intervjuerna med hjälp av bandspelare. Vid mina egna intervjuer gjorde jag inga anteckningar, men hade med mig ett frågeschema för att efterhand kontrollera att jag fick svar på alla de frågor som var relevanta. Intervjuerna skulle så långt som möjligt ske i samtalsform, eller kanske mera monologform, där intervjupersonen berättade om sina erfarenheter. Intervjuerna tog mellan en och två timmar att genomföra. Därefter skrevs intervjun ut efter bandet så fort som möjligt. Intervjupersonen fick sedan sig tillsänt en kopia av den utskrivna intervjun för kommentarer.

Denna utskrivna intervju blev mitt primärmaterial (Flensburg 1984b). Från det härledde jag svar på mina ursprungliga frågor och dokumenterade detta i nio intervjuer (Flensburg 1984c).

3.6.2 Deltagande i interaktiv systemutveckling

Föregående del av mina undersökningar ger mera översiktlig kunskap. Men det finns genomgripande kunskap som inte kan erhållas vid en förhållandevis ytlig intervju (jfr med tyst kunskap). Sådan kunskap erhålles bäst genom att aktivt delta i användarutvecklingsprojekt. Den forskning som på så sätt bedrivs har större anknytning till aktionsforskning. Man kan skilja mellan olika grader av medverkan, nämligen *observation*, *deltagande observation* och *aktivt påverkande*. Graden av medverkan är beroende av det uppsatta forskningsmålet. I intervjuundersökningen söker jag efter mönster och ett av målet med deltagande i interaktiva systemutvecklingsprojekt är att få uppslag till nya mönster eller belägg för gamla. Man kan mycket väl tänka sig att även bryta gamla mönster.

Att finna mönster

Om forskaren ska finna mönster måste det egna inflytandet minimeras eller åtminstone hållas under kontroll. I detta fall är "deltagande observation" den lämpligaste undersökningsformen. Att enbart observera skulle visserligen reducera inflytandet ännu mer, men är ur etisk synpunkt en mindre tilltalande metod. Hur skulle jag med gott samvete kunna undanhålla mina medarbetare på företaget den kunskap jag själv erhållit?

Det huvudsakliga forskningsarbetet består i att dokumentera och efterhand analysera det som sker. Det är viktigt att dokumentationen sker i så nära anslutning som möjligt till den aktuella händelsen. Annars glömmar man saker eller lägger omedvetet saker och ting tillräta. Detta primärmaterial bör vara så att även andra forskare kan få ta del av det för egen analys.

Man kan diskutera vad som ska dokumenteras. I princip skulle varenda samtal som förs om det aktuella projektet dokumenteras. Detta är ogörligt eftersom det då skulle kräva att någon forskare ständigt följde med varenda person på hela företaget för att eventuellt uppsnappa något som skulle kunna ha relevans för det aktuella projektet.

Ett annat sätt är att låta någon eller några på företaget sköta om denna dokumentation. Det kan emellertid vara svårt att motivera dem till detta. Det tar dock ganska mycket tid, som förmodligen inte finns. Däremot skulle kanske en forskare med jämna mellanrum kunna intervju dem om vad som hänt sedan sist. Nackdelen är att informationen i så fall inte blir primär utan lätt blir tillrättalagd eftersom den samlas in i efterhand. Detta skulle i viss utsträckning kunna balanseras genom att flera personer får berätta om samma sak. Emellertid kan denna metod få den nackdelen att

forskaren uppfattas som en slags "skvallerbytta" och att man därför undanhåller information. Det är lätt att det som sägs i ett sällskap kan misstolkas i andra. Kravet på forskningens öppenhet och tillgänglighet gör att risken för sådana misstolkningar icke är försumbar.

Jag har därför valt deltagande i form av projektmöten där forskaren och de inblandade användarna och eventuellt andra aktuella personer träffas och gemensamt arbetar fram lösningar. Dessa möten kan vara både informella och reguljära ungefär som vid vanlig systemutveckling. Här är risken att forskaren blir betraktad som en konsult vilken ska förse användarna med färdiga lösningar. Traditionen inom databehandlingsområdet är ju sådan och det är väldigt lätt att man omedvetet tar sig denna rollen. För min egen del har jag en lång erfarenhet som handledare på kurser i systemutveckling vid Lunds universitet och detta gör det något lättare för mig att inte komma med färdiga lösningar utan istället uppmuntra deltagarna till egna funderingar i kanske andra banor än tidigare. Men här saknar jag den viktiga verksamhetskunskapen och kan därför sällan riktigt bidra till lösningsprocessen. Men enbart det faktum att möten kommer till stånd har säkert en viss effekt.

Det som händer vid projektmötena dokumenteras på platsen, antingen i form av bandinspelningar eller minnesanteckningar. Dessa redigeras ut i maskinläsbart skick i regel samma kväll eller dagen efter med angivande av datum, både för möte och utskrift. Minnesanteckningarna distribueras sedan till samtliga personer som deltagit i projektmötet och till de andra forskarna i samma forskargrupp (i det här fallet LUIS-gruppen). Med minnesanteckningarna som grund utarbetas lägesrapporter en gång om året.

Den använda dokumentationstekniken har alltså varit en blandning av bandinspelningar och mötesanteckningar. Stor vikt lades vid att inte göra primärmaterialet för omfattande eftersom erfarenheter från andra liknande projekt (Röstlinger & Selldén, 1983) visar att man lätt får en alltför ohanterlig mängd primärmaterial.

Att ändra på mönster

Syftet är nu att få användarna att på ett konstruktivt och mänskligt sätt nyttja det tillgängliga redskapet i sitt arbete. Forskarens uppgift är att initiera detta nya arbetssätt och att ge användarna kurage och kunskaper att själv våga och kunna ta itu med användarutvecklingen.

För min personliga del har jag funnit att det absolut bästa sättet att starta en sådan process är genom en kurs. Redan 1981 höll jag en första kurs i "att ställa krav på datasystem" för en del bibliotekarier i Malmöhus län. Syftet var att frigöra dem från en mängd oriktiga föreställningar om datasystem och vad sådana kan användas till. Det finns rapporterat i Flensburg (1982) och ledde så småningom fram till att användarna själva tog fram ett system, "Limes" vid Malmö stadsbibliotek (Mundt-Almquist et al 1985).

Liknande kurser har hittills hållits för ytterligare fyra organisationer. Tanken med kursen är att den gradvis och omärkligt ska övergå i ett interaktivt systemutvecklingsprojekt. Därför brukar jag med kursdeltagarnas hjälp utforma en del av kursen som ett exempel på en systemutredning där de själva tillhandahåller problem. Härunder antar jag alltmer rollen av sekreterare och dokumenterare och ingriper bara när jag tycker mig se alldeles uppenbara möjligheter. Jag är alltid mycket noga med att ordentligt redovisa hur jag resonerar och varför jag tror att de förändringar jag föreslår skulle vara bra. Den på tavlan eller papper utarbetade lösningen har sedan implementerats med hjälp av någon applikationsgenerator. Det kräver att det finns tillgång till både dator och applikationsgenerator, något som tyvärr inte alltid varit fallet.

3.7 Samband mellan undersökningarna

De huvudfrågor jag studerar är hur interaktiv systemutveckling och personlig databehandling påverkar användarnas arbete. Jag tror att det i huvudsak blir en positiv påverkan, men ämnar med två olika strategier undersöka detta. Den första strategin är att genom personliga intervjuer få reda på vad användarna i ett antal fall själva anser om sin personliga databehandling. Den andra strategin är att själv delta i och initiera interaktiva systemutvecklingsprojekt.

De båda undersökningarna påverkar både varandra och forskningsområdet. I fig 3.10 (som är en modifiering av en liknande figur i Checkland, 1981) försöker jag klargöra dessa samband.

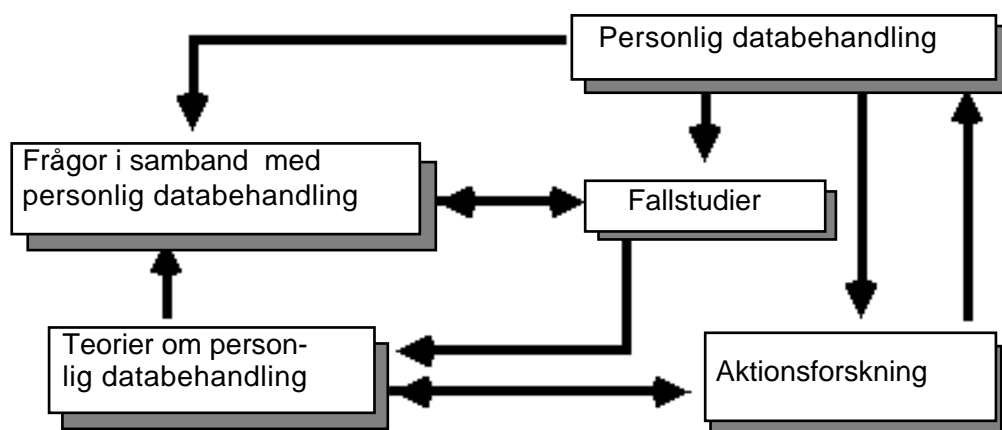


Fig 3.10 Samband mellan de båda undersökningarna

Det jag studerar är "personlig databehandling". I samband med dessa studier formulerar jag vissa utgångspunkter "Frågor i samband med personlig databehandling". Till grund för dessa frågor ligger vissa teorier om det studerade området. Med hjälp av intervjuundersökningar, "fallstudier" undersöker jag de uppställda frågorna och kan därigenom få nya bidrag till teorier om personlig databehandling. Med hjälp av den deltagande ansatsen, "aktionsforskning", får jag annan, mera direkt kunskap om det studerade området och kan även direkt påverka den personliga databehandlingen. Mitt beteende under denna studie styrs av de teorier jag hyser om forskningsområdet, men givetvis påverkas dessa också av mina undersökningsresultat.

4 EMPIRISKT MATERIAL FRÅN FALLSTUDIERNA

Detta kapitel baserar sig främst på fyra egna rapporter, nämligen "Erfarenheter av användarutveckling planering av en undersökning", (Flensburg 1984a), "Erfarenheter av användarutveckling intervjuer med användare" (Flensburg 1984b), "Erfarenheter av användarutveckling en sammanställning av ett antal intervjuer med användare" (Flensburg 1984c) och "Erfarenheter av användarutveckling tolkning av några intervjuer, synpunkter på en undersökningsmetodik" (Flensburg 1984d). Metodologisk och teoretisk bakgrund ges i Flensburg 1984a medan det empiriska materialet redovisas i Flensburg 1984b och 1984c. I Flensburg 1984d ges ett försök att urskilja några typiska mönster vid interaktiv systemutveckling, men också granska den använda metoden och frågeschemat. Dessa rapporter baserar sig på nio intervjuer.

Karakteristik av företag	Användare	IC-pers.	Dator
1. Medelstor, gammal, process	1	1	Servicebyrå
2. Liten, ung, verkstad	1		Servicebyrå
3. Stor gammal service	1		Mini
4. Stor, gammal, process	3	1	Stor
5. Mkt stor, gammal, tillverkning	2	1	Stor o person
6. Liten, ung, service	1		Person
7. Liten, ung, verkstad	1		Person
8. Liten, ung, service	1		Person
9. Liten, ung, tillverkning	1		Person
10. Stor, gammal, process		1	Stor o person
11. Stor servicebyrå		1	Stor
12. Stor servicebyrå		1	Stor
13. Stor, gammal, process		1	Stor
Summa	12	7	

Tab 4.1 Översikt av intervjuade företag och personer

Förutom dessa egna rapporter baserar jag mig på fyra andra, nämligen Hansson & Pettersson 1982, Persson 1985, Romanovsky 1985 samt Berg & Mattsson 1986. De har inte riktigt samma disposition som mina egna rapporter och täcker inte alla frågorna. Detta kapitel är en genomgång av det insamlade materialet fråga för fråga i likhet med Flensburg 1984a och 1984c. Kapitlet är endast en sammanfattning av de ovan angivna rapporterna. De omfattar sammanlagt c:a 450 sidor. De läsare som önskar ett fylligare empiriskt material hänvisas till dessa rapporter. För överskådlig-
hetens skull har jag gjort en sammanställning av de olika företagen i tabellform (tab 4.1).

4.1 De viktigaste resultaten

Först sammanfattas de viktigaste iakttagelserna i denna undersökning. Den första är uppkomsten av en *mellanhand*, dvs en person som hjälper övriga användare i deras arbete med de interaktivt utvecklade datasystemen. Denne person har teknisk handhavandekunskap om de använda redskapen, men också kunskap om vilka register som finns, vad de innehåller och hur en i vanligt språk ställd informationsbegäran kan formuleras i det aktuella frågespråket. Denna typ av kunskap kallar jag *modellkunskap*.

Den enskilde tjänstemannens beslut att satsa på interaktiv systemutveckling är *slumpmässigt*. Vederbörande har ett dataåtkomstproblem eller ett beräkningsproblem, träffar av en slump någon från Information Center och berättar av en slump om sitt problem. Han blir då förevisad något lämpligt redskap, får en kort demonstration, blir kanske hjälpt genom det första problemet. Personen ifråga skaffar sig behörighet att använda redskapet, utbildar sig genom *trial and error* och upptäcker vilken fantastisk nytta han har av både detta och andra tillgängliga redskap.

Användaren upplever att arbetet *blir färdigt fortare och blir bättre*. Vederbörande tycker det är roligt att jobba med datorn och känner sig inte hindrad av begränsningar, utom då de önskade uppgifterna inte är tillgängliga.

Om tjänstemannen arbetar med någon form av kalkylredskap, sparas i regel kalkylen för senare eventuell användning med andra siffror. Om det är rapportuttag från register, sparas inte själva rapportprogrammet för det är mycket lätt att skriva nytt. Det finns här en nivå mellan programmering och användning av färdiga program.

Möjligheten att *integrera olika användningar*, t ex kalkylering, ordbehandling och grafik uppskattas mycket. Speciellt grafikens förmåga att överskådligt presentera ett komplext siffermaterial är mycket användbar. Grafiska möjligheter används flitigt där de finns tillgängliga.

Man använde stordatorer i åtta av de tretton företagen. Det har fördelen att det finns viss *kommunikationsmöjlighet mellan de olika användarna*. T ex kan man skicka ut resultatet av en bearbetning på någon annans skrivare. Men samtidigt försöker man i

stordatorn skapa en miljö vilken så mycket som möjligt påminner om personatorn. Det man främst uppskattar är stordatorns större hastighet och möjlighet till kommunikation med andra tillämpningar, t ex den gemensamma databasen.

4.2 Redovisning fråga för fråga

I detta avsnitt går jag genom alla frågor jag hade på mitt frågeschema. För varje fråga finns det två olika avsnitt, *planerat*, som beskriver min förutfattade mening och *resultat*, som är en redovisning och i viss mån tolkning av givna svar.

Jag har valt att på ett och samma ställe samla all denna information för att därigenom göra den mera lättillgänglig och insatt i sitt naturliga sammanhang. I en del fall finns inte alla avsnitten med. Det beror på att frågan antingen tillkommit efter det att jag skrivit min planeringsrapport (Flensburg 1984a) då det inte finns något under avsnittet "planerat" eller att frågan inte blivit ställd. I det senare fallet finns det då inget under avsnitten "resultat".

Jag har försökt redovisa kvantitativa uppgifter där detta varit möjligt. Men det är inte fråga om någon statistisk undersökning. Många gånger stöter jag på stora mätproblem. T ex sade jag ovan att man i 8 av 13 fall använder stordatorer. Ett annat mått skulle kunna vara att 14 personer av 19 intervjuade använde stordator. Men ett av dessa fallen är ett Information Center där man stöder användning av både person och stordatorer. Hur ska detta räknas? Jag har hänfört det till stordatorfallet. Dessutom finns det en användare av minidator, som varken hör till stordator eller personator. I frågor under avsnitt A är totala populationen 19 personer, i de övriga är den 12 personer. Jag har då räknat bort personer som enbart jobbar på Information Center. Vidare har inte alla personer svarat på alla frågor och dessutom har i en del fall frågan varit irrelevant.

4.2.1 A. Frågor om redskapet

Allmänna kommentarer om detta avsnitt

En avsikt med detta avsnitt är att ge en beskrivning av applikationsgeneratorns faktiska egenskaper *såsom de upplevs av dess användare* (Flensburg 1984a). Om man vill göra en bred och jämförande undersökning av olika applikationsgeneratorer vore en enkät hos leverantörerna rätt metod. Emellertid studerar jag här applikationsgeneratorer i bruk och då måste jag hela tiden utgå från de som brukar den. Både användare och personal vid Information Center ska besvara dessa frågor.

I de undersökningar jag här genomfört har jag varit noga med att inte känna till något om applikationsgeneratorn innan jag intervjuade dess användare. Detta har inte varit helt lyckat, för jag har många gånger inte riktigt begripit vad som sagts. Därför

rekommenderar jag att först ställa frågorna till Information Center personalen innan man frågar användarna.

Innan jag gjorde undersökningen trodde jag att applikationsgeneratorns egenskaper var mycket väsentliga för de som använder den. Men det visade sig inte vara så och därför har frågorna under detta avsnitt förlorat en hel del av den vikt och betydelse jag tillmätte dem från början.

Frågorna var ämnade att ställas i första hand till dataavdelningen (speciellt Information Center), men även användaren ska svara på dem. En del av frågorna kan enklast besvaras genom ett exempel, andra genom studium av manualer, broschyrer o dyl.

A1. Vilket företag tillverkar eller marknadsför applikationsgeneratoren? Årsomsättning? Antal anställda?

Planerat

Syftet med denna fråga var tvåfaldigt. Dels att få reda på den kulturella miljö i vilken verktyget konstruerats, dels att ge inköpsunderlag för läsare som funderar på att skaffa sig något sådant redskap. Kulturell miljö kan vara viktig för att förklara vissa egenheter hos redskapet. T ex har man i USA en helt annan inställning till medinflytande frågor än vi har i Sverige. Vad beträffar firmans storlek föreställde jag mig att en stor firma kan ta bättre ansvar för produkten, att den på något vis var mera pålitlig än en liten firma. Emellertid kan jag ha fel i detta, den här speciella typen av redskap torde gå lika bra att utveckla i en liten firma som i en stor, kanske rent av ännu bättre. Om applikationsgeneratoren är en produkt av den egna dataavdelningen bör detta också anges här.

Resultat

Alla visste vad redskapen hette. Sex stycken, varav fyra från IC visste även namnet på tillverkaren. Ingen kände till antal anställda eller årsomsättning på tillverkningsföretaget. Sammanlagt talas om 23 olika redskap. Tre av dem är tillverkade av det egna företagets dataavdelning, två andra av mjukvarufirmor i Sverige. Resten är från USA.

A2. Vilken dator kan applikationsgeneratoren användas under? Vad krävs i övrigt? (dvs primärminne, operativsystem, m m) Är den specialgjord för en viss konfiguration? Hur mycket kostar den?

Planerat

Denna fråga ger en del antydningar om "kraftfullheten" hos den undersökta applikationsgeneratoren. Den talar också om hur stor investeringskostnaden är. Många applikationsgeneratorer är specialgjorda för en viss maskinkonfiguration och då ska det anges här. Omvändningen finns också, att datorn är specialgjord för applikationsgeneratoren (T ex Mapper 5)! Frågan kan också tjänstgöra som indelningsgrund för olika typer av applikationsgeneratorer. Priset kan vara svårt att jämföra mellan olika appli-

kationsgeneratorer. Ofta säljs de modulvis och antalet moduler är beroende av tillämpningen. Kanske vill det undersökta företaget inte uppge hur mycket de betalt för redskapet utan någon sorts typpris i form av intervall kan då vara rimligt.

Resultat

Alla stordatoranvändarna visste bara att "...det är nån sorts IBM-maskin vi har, men fråga mig inte om vilken eller hur stor den är!". De fem som använde persondatorer visste vilka dessa var. Personerna från IC hade god reda på sig i denna fråga. Beträffande priserna förekom två varianter. Dels det stora flertalet, där man i stort sett struntade i vad det kostade och dels två stycken som tyckte det var alldeles för dyrt att använda applikationsgeneratorn. Men ingen jag träffade bekymrade sig över ett högt inköpspris. De fem personer som använde persondatorer var också helt obekymrade om kostnaderna.

Svaren har blivit en redovisning av vilken applikationsgenerator som används och vem som tillverkat eller marknadsför den. Jag lägger märke till:

- USA:s dominans (18 av 23 redskap från USA. Sex av dessa från IBM)
- Att man använder en uppsättning redskap istället för ett enda. (15 fall av 19)
- APL:s dominans som värdspråk (APL är enda angivna värdspråk, i åtta fall)

Det är helt meningslöst att fråga efter tillverkarens exakta årsomsättning eller exakt antal anställda ty de har ingen betydelse för denna undersökning. Det kan betyda något om det är ett stort eller litet företag som marknadsför applikationsgeneratorn eller om det är den egna dataavdelningen. Två av redskapen har senare fått kritik av användarna och dessa var båda gjorda i Sverige, av fristående mjukvaruhus och av egna dataavdelningen. De kulturella skillnaderna förefaller fortfarande vara relevanta, fast denna undersökning inte varit så djupgående som från början avsågs.

De flesta datorer var stordatorer. Det är svårt att ange exakt siffra här, eftersom det i fem företag förekom satsningar på både stora och små datorer. Fem av de här redovisade tillämpningarna kördes enbart på persondatorer. Sju av dem var rena stordator-tillämpningar. Det är förvånande att persondatorer inte används mer. Dock kan en viss skevhet föreligga i urvalet eftersom det var personal från dataavdelningen som introducerade intervjupersonerna. Men användningssättet liknade persondatorns. På ett av företagen talade man dessutom om "persondatorn i stordatorn". Man kan förmoda att efterhand som mikrodatorerna blir allt kraftfullare och snabbare, blir de mer använda. Där mikrodatorer användes var det främst för kalkyler. Av någon anledning tycks det bli billigare att ge 200 användare varsin PC med 500 KB primärminne än att ge dem motsvarande utrymme i en centraldator. Åtminstone var det så en dataexpert såg på saken!

Kostnaderna för redskapen har inte varit möjliga att få fram. Men endast i två fall har intervjupersonen direkt klagat på kostnaden. Valet av redskap görs vid inrättandet av Information Center och är därefter inte möjligt att påverka. Intressantare är att se vad man vill ändra på. Men den frågan kommer senare.

A3. Vilka funktioner finns i redskapet (t ex registeruppläggning, registerförändring, rapportgenerering, formulärhantering, kalkyler, grafik, frågespråk, möjlighet att skriva procedurer, kommunikation med andra datorer, möjlighet att kommunicera med övriga användare, speciell datamodell, felsökningsmöjlighet, värdspråk m m)

Planerat

Detta var den stora och viktiga frågan i detta avsnitt. Om man ställer den till en dataexpert riskerar man en längre, lyrisk lovsjungning av redskapets förmåga och kapacitet. Ställer man den till användaren, riskerar man en detaljerad beskrivning av aktuell tillämpning. Bägge är intressanta och innehåller säkert svar på många av de andra frågorna. Man bör således inte avbryta utan istället notera uppföljningsfrågor. Ett exempel är också ett bra sätt att besvara denna fråga på. Om man får låna en manual, kan utdrag ur den användas som svar. Viktigt är att få med så mycket som möjligt, både av det man använder och det man inte använder i aktuell tillämpning. Frågan rör ju själva redskapet i sig, inte tillämpningen. Därför är det kanske bättre att ställa denna fråga till dataexperten.

Resultat

Det visade sig snabbt att frågan inte alls var så viktig som jag hade trott. Ställde jag den till användaren begrep inte *han* vad jag var ute efter. Om jag bad om en demonstration fattade inte *jag* något. En idé jag fått från professor Staffan Persson gick ut på att man bara använder 10-15 olika kommandon. När jag frågade intervjupersonerna instämde man i detta. Det är för övrigt ganska enkla funktioner som behövs, men vad som verkar viktigt är integration av ordbehandling, register, kalkyl och grafik. Materialet ger inget underlag för att säga något detaljerat om egenskaperna hos applikationsgeneratorerna, såsom t ex kommandosekvenser o dyl.

Man kan urskilja två typer av svar. Sådana som är tämligen kortfattade och bara säger vilken typ av redskap det är och andra som är långa beskrivningar med körnings-exempel och allting. Den förra typen finns redovisad i nästa fråga, den senare i bakgrundsmaterialet, främst Flensburg 1984b. Jag anser att datapersonalen ska demonstrera redskapen, för med dem har jag möjlighet att diskutera i dataspråk och kan bitvis gå ganska djupt i de tekniska delarna.

A4. Är applikationsgeneratorn avsedd för någon speciell tillämpning? Vilken ?

Planerat

Detta var en relativt enkel och rak fråga och var också tänkbar indelningsgrund. Jag tänkte att uppgiften lämpligen hämtades från tillverkaren.

Resultat

Denna fråga hänger hop med den förra. I praktiken har det blivit en klassificering av applikationsgeneratorer i de fyra klasserna:

- Registerhanteringsredskap (7 st)
- Kalkylredskap (5 st)
- Grafikrutiner (5 st)
- Integrerade redskap (6 st)

Även här är det en del problem att räkna de olika fallen. Om en person på ett Information Center säger att man har 200 användare anslutna och att man tillhandhåller en viss applikationsgenerator, ska man då räkna dessa tvåhundra som användare? Jag har inte gjort det utan räknat IC som en användare, eftersom det var med IC-personal jag pratade. En del personer använde huvudsakligen databaserna, andra använde huvudsakligen kalkylprogrammet, men bara i ett fall använde man båda. Om det fanns grafik användes den flitigt.

Jag tror inte det är meningsfullt att i denna undersökning försöka göra en detaljerad redovisning av egenskaperna hos någon applikationsgenerator. Bästa sättet att göra detta torde vara genom ett väl valt exempel. Men det handlar då snarare om att jämföra applikationsgeneratorer än att beskriva erfarenheterna av dem. Sådana jämförelser har gjorts av t ex Kuvaja & Similä (1985) och Erikö (1985).

A5. Vem är applikationsgeneratören främst tänkt för? Dataexpert eller icke?

Planerat

Enligt reklamen kan de flesta applikationsgeneratorer användas utan någon föregående programmerings eller datorkunskap. Här bör man ta reda på vad reklamen säger om det aktuella redskapet och sedan jämföra detta med dataexpertens eller användarens erfarenheter. Helst bör intervjuaren försöka bilda sig en egen uppfattning genom att studera manualen eller prova själv. Att ställa de olika personernas uttalande mot varandra och sedan jämföra med reklamen för den aktuella tillämpningen kan kanske ge underlag för intressanta frågor.

Resultat

17 av redskapen var tänkta att användas av slutanvändaren och följaktligen var 6 tänkta att användas av Information Center. Men det kräver rätt mycket träning för att effektivt kunna utnyttja en väl utbyggd applikationsgenerator. Som ett kuriosum kan noteras att en person som använde ren APL, inte var någon dataexpert utan en helt vanlig användare som lärt sig APL enbart genom att studera exempel och genom att experimentera!

Det är intressant att samtliga intervjupersoner inte tyckte det var svårt att lära sig applikationsgeneratören. Däremot trodde de att någon annan att med kort varsel skulle kunna sätta sig in i deras arbete. Möjligen kan det bero på missuppfattning mellan "arbete" och "bruk av applikationsgenerator". Det visar emellertid också på en viktig komponent av kunskapen, nämligen kunskap om den information som tas ut från datasystemet och hur den är organiserad.

A6. Har applikationsgeneratoren något värdspråk? I så fall vilket?

Resultat

Sex stycken har APL som värdspråk, övriga har inget. Men det beror på att IBM:s Information Center filosofi från början baserades på APL. Det är ett synnerligen resurskrävande språk.

A6. Hur länge har applikationsgeneratoren funnits på marknaden?

Planerat

Denna fråga ger också underlag för eventuell kategorisering. De äldre redskapen skiljer sig kanske i någon mening från de nyare. Den ger också tips om möjligheten att erhålla även beskrivningar av långsiktiga effekter.

Resultat

Denna uppgift var svår att få tag i och kändes inte heller speciellt väsentlig. Det finns emellertid två "generationer" av applikationsgeneratorer, vilket allmänt erkänns av de personer jag talat med. 9 st använde redskap från den första generationen, men var ändå inte speciellt intresserade av att få dem utbytta. Det finns ett undantag och det var en person som börjat använda ett andragenerationsredskap och sedan blivit tvungen att "byta ner sig". Han var inte nöjd.

A7. Hur många installationer finns den i? I Sverige? I världen?

Planerat

Detta är en indikation på hur pass populärt redskapet blivit och därmed i viss mån på hur "bra" det är. Frågan bör jämföras med den om maskinutrustning och operativsystem som redskapet kan köras under. Ett redskap skrivet för ett populärt operativsystem säljs rimligtvis mer än ett annat skrivet för ett sällsynt operativsystem, även om det senare i sig är mycket bättre. Uppgiften hämtas säkrast från tillverkaren eller möjligen ur översikter och jämförelser i fackpressen. För att få jämförbara siffror skulle egentligen alla sålda redskap divideras med antalet tänkbara sålda sådana, dvs antalet datorer under vilka redskapet kan användas. Detta medför t ex att redskap avsedda för mikrodatorer blir snedrepresenterade eftersom sådana i regel är enanvändaresystem. Ett mått man kanske kunde använda istället är antalet användare totalt sett. Men det är säkert mycket svårt att fram siffror därvidlag.

Resultat

Min diskussion visar hur svårt det är att mäta sådana här uppgifter. Dessutom tillkommer problemet att avgöra vem som är användare eller inte. Är den person, som kommer till mellanhanden och ber denne ta fram några uppgifter en användare? Mina mätproblem löstes eftersom att ingen kände till antalet.

A8. Vilka brister har applikationsgeneratorn?

Planerat

Denna fråga ställes lämpligen till både dataexperten och användaren. Den ger underlag för att ställa relevanta krav på hur applikationsgeneratorer skall se ut och är därför en av de allra viktigaste i detta avsnitt. Förmodligen måste de önskade utvidgningarna relateras till den aktuella tillämpningen.

Resultat

Alla applikationsgeneratorer har sina brister. Frågan borde egentligen inte ställas här utan komma senare eftersom de brister som anges här är sådana som intervjupersonerna upplevt som brister. Redovisade brister (för i huvudsak första generationens redskap) är:

- Endast ett register kan behandlas åt gången
- Besvärlig inloggning
- Bristande integrering
- För få funktioner
- Låg driftsäkerhet
- För långa svarstider
- Ej färdig produkt

I den andra generationens applikationsgeneratorer är de flesta av dessa brister borttagna. Man kan undra om detta sammanträffande är en ren slump eller om man ser bristerna först när man ser något annat som inte har dessa brister.

Jag lägger också märke till att intervjupersonerna prioriterar olika brister. Möjligen är låg driftsäkerhet något prioriterat (3 personer har angivit detta). Det kan tyda på att det inte finns några mer generella brister eller att bristerna är mer hänförliga till det egna arbetssättet. Intressant är också att användaren i regel är nöjdare än dataexperten och uppfattar färre brister.

4.2.2 B. Frågor om användarföretaget.

Allmänna kommentarer

Orsaken till att jag ville ställa dessa frågor var att jag trodde typen av företag kunde ha betydelse för vilka effekter användarutveckling fick. Man skulle t ex kunna förmoda att decentraliseringstendenser i en större koncern understöds. Det kunde tänkas att företagets storlek hade något med effekterna att göra. I den klassiska enkätundersökningen samlar man in en massa uppgifter om företaget. Dessa uppgifterna används sedan som bakgrundsvariabler. I denna undersökningen visste jag på förhand att något sådant inte gick att genomföra. Därför har jag försökt hålla nere antalet frågor i detta avsnitt.

Senare har jag mer eller mindre avgränsat mig från dessa organisatoriska effekter, inte för att de inte är intressanta eller viktiga, utan för att jag ville börja med de individuella effekterna. På individnivån uppfattar jag att det gjorts mycket mindre, medan det på organisationsnivån finns rätt mycket forskning, som åtminstone har beröringspunkter med mitt problem.

B1. Hur många anställda är det? Hur stor är årsomsättningen?

Planerat

Båda siffrorna är mätt på storleken. Antal anställda säger en del, men inte allt. Jag vet ett företag som har 4 st anställda men en omsättning på 250 miljoner! Är detta ett litet företag eller ett stort? Båda svaren är riktiga, det beror på vilken definition man använder. Därför vill jag ha båda. Dessutom borde jag ange vilket år det gäller.

Resultat

Mätproblemen har inte varit särskilt stora. De två måtten har följts åt och jag anger företaget i olika klasser: mycket stort, stort, medelstort och litet.

Frågan har snarare blivit en karaktäristik av företaget som använder applikationsgeneratoren än ett bokstavligt svar. De faktorer jag klassificerar efter är ålder på företaget, typ av företag och storlek på företag. Jag anger också mycket grova mått. Resultatet finns i tab 4.1

B2. Vilken tillämpning använder man applikationsgeneratoren för?

Planerat

Denna fråga var avsedd att ligga till grund för eventuell klassificering av tillämpningarna. Den kunde också generera en hypotes om vilka tillämpningar användarna själva kan utveckla. Det kunde också tänkas att man använde applikationsgeneratoren för en helt annan tillämpning än vad tillverkaren avsett. Erfarenheterna av sådant är intressanta att dokumentera. Bakom denna fråga låg vidare en föreställning att det användes en och endast en applikationsgenerator i varje enskilt fall och att den används för en och endast en tillämpning. Detta visade sig dock vara alldeles fel!

Resultat

Följande användningsområden redovisas:

- Budget och budgetuppföljning: 4 st (Registerhantering)
- Ekonomiska simuleringar: 2 st (Kalkyler)
- Ekonomisk planering: 3 st (Kalkyler)
- Statistik: 1 st (Registerhantering)
- Lagerhållning: 1 st (Registerhantering)
- Grafik, typ rita diagram: 3 st
- Underhåll av maskinpark: 1 st
- Information Center verksamhet: 4 st

Det tycks inte vara så att en viss typ av tillämpning dominerar utan det är snarare möjligheten att göra kalkyler, möjligheten att upprätta register som är det viktiga. Om det är ekonomiska kalkyler eller anbuds-kalkyler eller produktkalkyler spelar ingen större roll. Det är själva databehandlingen som är det viktiga. Flera olika redskap kan användas i en och samma tillämpning.

Jag tänkte från början att denna fråga skulle ställas främst till dataexperterna. Men de som vet mest om tillämpningarna är utan tvivel användarna. Dataexperterna på Information Center vet inte så mycket om vilka tillämpningar man använder redskapen till, de vet mera om själva redskapen.

B3. Hur gammalt är företaget? Hur länge har man haft databehandling?

Planerat

Bakom denna fråga låg en mer eller mindre vanföreställning om att äldre företag är mera stela och rigida och har svårare att anamma nya idéer än vad nyare företag har. Huruvida det är så eller ej, visste jag inte, men genom att ställa denna fråga kan jag kanske få någon indikation på detta. Likaså borde användarna ha en annorlunda inställning om ett företag har haft databehandling länge, än om sådan nyligen införts.

Resultat

De stora och medelstora företagen (12 intervjuer i 6 företag plus två servicebyråer) har haft databehandling länge. Vidare har man någon form av gemensam databas. De mindre företagen (5 st) har tidigare haft någon form av databehandling, men den har inte varit vidare framgångsrik.

B4. Har man egen dataavdelning? Om så är fallet: Hur stor är den? Vilken/ vilka dator(er) har man? m m

Planerat

Om man har egen dataavdelning eller kör mot en servicebyrå borde få olika effekter på användarutvecklingen och den allmänna inställningen och uppfattningen av databehandling. Användarens chef eller dataexperten är rätt person att svara på denna fråga.

Resultat

Alla stora företag hade egen dataavdelning och alla de små hade ingen. Det medelstora använde servicebyrå. De stora företagen hade också IC (motsv).

4.2.3 C. Historisk kartläggning

Allmänna kommentarer

Avsikten med denna del var att skaffa underlag för att förstå de fenomen forskaren träffar på vid kartläggningen. Här stötte jag på delvis annorlunda problem än tidigare. Det historiska förloppet är inte objektivt givet, utan beskrivningen är alltid sedd

genom människors ögon, tolkade av människor och kanske förmedlade i flera led. Frågorna ställdes därför till alla personer som intervjuades för att få så många versioner som möjligt. Det bästa sättet var att helt enkelt fråga: "Hur började det här med interaktiv systemutveckling egentligen?" Intervjupersonen fick sedan fritt berätta. Väldigt ofta var det som att trycka på en knapp och i stort sett alla relevanta svar kom genast.

Ett speciellt problem man kan stöta på är att den intervjuade inte vet något alls om förhistorien eftersom vederbörande inte kom till företaget förrän efter det att applikationsgeneratorn tagits i bruk. Man bör emellertid ställa frågorna i alla fall, men acceptera något mindre utförliga svar. Istället kan man då fråga efter vem man ska ställa sådana frågor till.

C1. Varför valde man interaktiv systemutveckling istället för vanlig systemutveckling?

Planerat

Detta är alltså huvudfrågan. Jag frågar inte efter hur man kom på just den aktuella tillämpningen utan istället varför man valde interaktiv systemutveckling och inte traditionell systemutveckling. Frågan skulle kunna utvidgas till att gälla hur man inom företaget under tidernas lopp sett på användarmedverkan.

Resultat

För de stora företagen är det riktiga svaret: "Tja, det bara blev så...!" Valet att använda applikationsgenerator verkar vara initierat av personer på Information Center där sådant finns. Egentligen ganska naturligt eftersom det är där kunskapen om redskapen och deras möjligheter finns. Två följdfrågor uppkommer: Varför började man använda applikationsgenerator för en viss tillämpning och hur kom dataavdelningen på att det fanns sådana produkter?

På fråga ett är jag benägen att svara att det beror till stor del på slumpen. Av en slump träffade man någon från Information Center och av en ren slump började man tala om något av aktuellt problem. Personen från Information Center föreslog att man skulle använda något av deras hjälpmedel och visade vad som fanns. Användaren tog det till sig, skaffade behörighet och därmed var den interaktiva systemutvecklingen ett faktum.

Om vi sedan ser varför dataavdelningen gick in för interaktiv systemutveckling ger materialet inte underlag för några alltför vittgående slutsatser. Från en av dataavdelningarna saknas det direkt material, men det framgår av de övriga intervjupersonerna att man lanserat ett begrepp: *personal computing*. En annan dataavdelning talade om "persondatorn i stordatorn". I båda fallen förefaller mikrodatorer och programvaran till dessa ha tagits som utgångspunkt. Man riktar sig direkt mot den enskilde personens individuella behov av databehandling. På ett av företagen gjorde utvecklingen av databaser det inte längre möjligt att använda Easytrieve. En satsning på Informa-

tion Centermiljön mer eller mindre tvingades fram. En intressant fråga att följa upp kan vara om detta är något vanligt förekommande eller om det är unikt för ifrågavarande företag. Man kan också fråga i vilken utsträckning IBM var medveten om vilka problem övergången till mera sofistikerade datastrukturer skulle medföra. Begreppet "Information Center" lanserades ju, enligt min sagesman, vid ungefär samma tid av IBM. Det finns mycket noggranna anvisningar från IBM (IBM 1982) om hur Information Center ska inrättas och vilka dess uppgifter är. Berg & Mattsson (1986) menar att huvudsakliga syftet med Information Center är att låta dataavdelningen utöva viss kontroll över den personliga databehandlingen.

Servicebyråerna lanserade Information Center verksamhet som ett konkurrensmedel. På samma sätt som dataavdelningarna på stora företag bildade man Information Center. En skillnad kan dock vara att servicebyråerna är mera kräsna i val av verktyg. Därför ligger man efter jämfört med storföretagen. Det kan möjligen på sikt medföra att servicebyråer kan erbjuda bättre redskap och tillhandahålla bättre service än dataavdelningarna på de större företagen. Men troligen kan inte användare på de större företagen nyttja servicebyråer, eftersom de inte har tillgång till företagets databas. De företag i min undersökning som använde sig av servicebyrå tyckte båda att det var mycket dyrt. Om priset blir lägre kan servicebyråerna ha en ny marknad här. Men hela detta frågekomplex bör bli föremål för grundliga undersökningar.

Alla de mindre företagen menar att man valde användarutveckling för att undvika kommunikationsproblem med datafolk. I samtliga fall "gick man på hej vilt" och prövade sig fram. Detta gällde också användarna på de stora företagen.

C2. Vem (befattning) tog initiativet?

Planerat

Detta är en uppföljning av den förra frågan. Jag är intresserad av att veta om det kanske finns något mönster bland initiativtagarna. Möjligen kunde IBM:s lansering av begreppet "informationscenter" leda till bruket av applikationsgeneratorer. Det viktigaste är nog att utröna om initiativet kom från dataavdelningen eller från något annat håll. Motstridiga uppfattningar kan dessutom förekomma.

Resultat

På stora företag är det meningslöst att säga att någon tog initiativet, eftersom användarutveckling uppkom av en slump. Men det bör betonas att Information Center, verkligen ställde upp och drev saken i början. Det behövs också någon eldsjäl på avdelningen för att det ska slå genom ordentligt. Även på dataavdelningen bör finnas några eldsjälar som drar igång Information Centerverksamheten. Att Information Center hänger hop med IBM:s lansering säger alla. På de mindre företagen togs initiativet i samtliga fall av ägaren/VD.

C3. Vilka personer eller grupper berördes av initiativet? Hur reagerade man? Vilka skäl angav de för sin reaktion? Hur hanterade man ett eventuellt motstånd? Vad tyckte man efteråt?

Planerat

Detta är ett känsligt frågekomplex. Orsaken till att jag trots det frågar är att jag föreställer mig att det kan bli motstånd mot något alldeles nytt och okänt. Orsakerna till detta motstånd och sättet att handskas med det kan säkert ge goda lärdomar åt andra företag, som eventuellt befinner sig i liknande situationer. Den som intervjuar bör nog ange dessa skälen innan frågan ställs så att den som blir intervjuad kan inse varför vi vill veta detta.

Resultat

Det visade sig att myten om motståndet mot bruket av applikationsgeneratorer just var en myt. Ingen har nämligen angett att det förekommit något motstånd!

Denna fråga skulle möjligen vara alltför ledande i och med att dataavdelningen anges som ett möjligt svar. När frågan ställdes gjorde jag det i två etapper, först frågade om det mötte något motstånd, sedan speciellt om dataavdelningen. Bakgrunden var en artikel jag läst i Datavärlden (Ahrnell, 1983) där Sverker Söderström, chefskonsult på statskontoret studerat ett tjugotal företag som använde applikationsgenerator. Han påstod att ADB-specialisterna var motståndare och försökte hindra bruket av användarnära utvecklingsredskap. Något sådant har jag dock inte kunnat upptäcka i detta material. En orsak kan vara att intervjupersonerna samtliga var mycket entusiastiska och intresserade av att använda applikationsgeneratorerna. Då märker man kanske inte så mycket ett måhända tyst och passivt motstånd från omgivningen. En annan orsak kan vara att jag misstolkar sådana uttryck som "det har ju gått lite trögt i början...". De kan möjligen betyda mer än de står för, men vid intervjutillfället hade jag ingen anledning förmoda något annat.

För att få ett bättre svar på frågan om den egentliga dataavdelningens eventuella motstånd, måste även denna tillfrågas. De fyra undersökningar som direkt berör IC-verksamhet har emellertid inte indikerat någon sådan motsättning.

C4. Vilka skäl fanns att satsa på applikationsgenerator?

Planerat

Här ville jag få reda på om man hade övervägt möjligheten att använda någon annan form av metod än interaktiv systemutveckling för den aktuella tillämpningen samt om man gjort någon utvärdering innan man började.

Resultat

Applikationsgeneratorer är idag enda möjligheten vid personlig databehandling, som det ju är fråga om här. En rationell ekonomisk utvärdering gjordes egentligen bara i ett fall (medelstort företag) då man körde mot servicebyrå.

C5. Vilka förväntningar hade man? Infriades de? Finns målsättningen dokumenterad? Gjordes någon utvärdering efteråt? Om så var fallet, hur gjordes den och vad kom man fram till?

Planerat

Denna fråga kan också formuleras som målsättningen med att satsa på interaktiv systemutveckling. Förmodligen har några svar redan givits under fråga C1 ovan (huvudfrågan). Om den inte blivit besvarad där, bör den bli det nu. Det är viktigt att intervjuaren först frågar efter målsättningen och sedan efter den skrivna målsättningen. Det kan nämligen tänkas att den intervjuade personen uppfattar målsättningen som en annan än den nerskrivna. Om det är så, kan man gärna fråga efter orsaken. Givetvis frågar man i detta sammanhang om målsättningen blev infriad.

Resultat

På de stora företagen var ursprungliga anledningen att en viss rapport skulle kunna framställas snabbare. Efterhand upptäckte intervjupersonen att applikationsgeneratorn även gick att använda till andra rapporter och att det fanns fler applikationsgeneratorer med andra funktioner. På så sätt startades ett stundtals mycket intensivt bruk av datorbaserade hjälpmedel för det egna arbetet. Man kan alltså lugnt påstå att förväntningarna mer än infriades.

Två av de mindre företagen sade sig inte ha haft några förväntningar, ett av de andra menade att förväntningarna blev mycket väl infriade medan det sista inte alls hade fått sina förväntningar infriade.

I och med att användarutveckling tillkom mer eller mindre av en slump finns det inte klart utsagt några förväntningar eller målsättningar i förväg från användarnas sida. Om man istället frågar om användaren är nöjd blir svaren i alla fallen utom ett "Ja".

Beträffande utvärdering har intervjupersonerna svarat på frågan om man gjorde någon utvärdering innan man började medan min fråga gällde om man hade gjort någon uppföljning. Det är viktigt att använda rätt ord!

På Information Center har man gjort någon form av utvärdering innan man valde verktyg, men det har varit ganska enkla utvärderingar.

C6. Använde man någon speciell metod eller modell vid själva systemeringen? Vilken i så fall? Skilde denna sig från den traditionella systemutvecklingen?

Planerat

Denna fråga bör man nog ställa först till datafolk och sedan till användare. Den har kanske redan blivit besvarad under huvudfrågan ovan. Jag ställer frågan därför att jag tror inte man kan använda samma systemutvecklingsmodell vid interaktiv systemutveckling som vid traditionell systemutveckling. Egentligen borde man på något vis knyta hop resultatet av utvecklingen med den tillämpade metoden (eller modellen) men detta bedömer jag föra alldeles för långt i detta sammanhang. Viktigt när

man ställer denna fråga är att försöka få reda på hur pass medvetet valet var vid den aktuella tidpunkten. Att få tillförlitliga svar på detta kan nog bli svårt.

Resultat

Sju användare av tolv använde sig av "trial and error". Av de övriga har två övertagit ett i stort sett färdigt system, men har lärt sig detta delvis genom "trial and error". De övriga tre har haft en ganska god kunskap om redskapen och systemets utseende redan från början, utan säger sig helt enkelt "ha gjort systemet". Det förefaller som en vanlig metod är "fritt experimenterande" i en till tre månader, därefter blir applikationsgeneratoren mer ett redskap i stil med räknedosa och lika naturligt att använda. Uppenbart är att man inte använt traditionella systemeringsmetoder. Snarare har man inte använt någon metod alls. Detta borde ha fått negativa effekter i samband med uppbyggnaden av en individuell databas, men några sådana har jag inte upptäckt. Ett av systemen är ganska konventionellt utvecklat av personal från servicebyrån. Det används också på ett väldigt konventionellt sätt (satsvisa körningar). Anmärkningsvärt är att man efter tre års användning fortfarande inte litar på systemet utan räknar för hand bredvid!

I denna fråga ligger ett av undersökningens huvudresultat. Eftersom det är en helt annan typ av informationssystem vid användarutveckling än vid traditionell systemutveckling måste man följaktligen använda sig av annan systemutvecklingsmodell. Detta har IC-personalen insett och påpekat. I samband med interaktiv systemutveckling är skillnaden mellan utveckling och användning mycket mindre än vid traditionell utveckling.

C7. Gjordes motsvarande uppgifter tidigare med datorstöd? Hur såg detta ut?

Resultat

Det typiska svaret på denna fråga är: "En del uppgifter gjordes tidigare för hand. För att dessa skulle kunna göras snabbare började man använda applikationsgeneratorer. Sedan blev det fler och fler uppgifter." Svaren här stämmer väl överens med svaren på fråga C1.

C8. Hade man problem vid införandet? Vilka i så fall?

Resultat

I det fall då man använde systemet satsvis hade man likartade problem som vid traditionella satsvisa körningar. I ett av de mindre företagen hade man stora problem att få programmen att fungera. I övriga fall har man inte haft några direkta problem.

C9. Hur länge har man haft applikationsgeneratoren?

Resultat

Svaren på denna fråga kan relateras till konsekvenser av första, andra eller tredje ordningen enl Göranson (1983a). Det blir då följande fördelning:

- Första ordningen (tekniskt handhavande, 0-0.5 år): 5 st
- Andra ordningen (psykosociala och arbetsmiljömässiga, 0.5-1.5 år): 8 st
- Tredje ordningen (yrkeskunskap, 4-5 år): 6 st

För Information Center-personal är inte denna indelning meningsfull. I min undersökning kan jag inte se någon större skillnad i konsekvenser mellan de olika kategorierna. Detta är intressant och kan innebära att det här rör sig om en annans sorts informationssystem eller andra sorters användning.

4.2.4 D. Nuvarande användning av applikationsgeneratortorn

Allmänna kommentarer

Detta är huvudfrågorna till användaren. Precis som i det historiska avsnittet bör man låta den intervjuade få prata så fritt som möjligt. Frågorna nedan är bara avsedda som en slags stödpunkter.

D1. Vilken tillämpning använder man applikationsgeneratortorn till?

Planerat

Detta är bara ett klassifikationsunderlag. Jag tror att t ex om ett företag har en bokföringstillämpning där man funderar på att nyttja interaktiv systemutveckling är man intresserad av att titta på andra sådana tillämpningar och inte av t ex produktionsplaneringstillämpningar. Jag vill rekommendera att intervjuaren försöker få den aktuella tillämpningen beskriven så noggrant som möjligt. Indelningsgrunden kan komma att ändras och det är bra att i grundmaterialet kunna göra omstruktureringar. Intervjuer är en dyrbar datainsamlingsteknik och det är en fördel om materialet kan utnyttjas på flera sätt.

Resultat

Tillämpningsområdena är redovisade i fråga B2. Här ska jag ta upp hur man använder applikationsgeneratortorn.

1. Som satsvis system: 1 st
2. Genom mellanhänder: 6 st
3. Slut användaren direkt: 11 st
4. Som utvecklingshjälpmedel för standardsystem: 1 st

Fallet 1 är ett extremfall. Fallet 4 är möjligen också ett extremfall, beroende på hur pass modifierbart det färdiga systemet är. Fallet 2 innebär att det finns personer som är intresserade av att få viss information. De går till mellanhanden, som då med hjälp av applikationsgeneratortorn tar fram den önskade informationen. Mellanhanden blir en lokal dataexpert. En av personerna i fall 3 var också något av en mellanhand eller kanske snarare utbildare och hjälptillhandahållare. Detta var dock övergående. En av

mellanhänderna antydde att det finns även slutanvändare som kan ta ut enkla rapporter med hjälp av lätthanterliga faciliteter hos applikationsgeneratoren.

Med "tillämpning" avsåg jag här den traditionella betydelsen, dvs typ ordermottagning, fakturering, reskontra etc. Men på stora företag utvecklas sådana tillämpningar inte av användarna, även om jag tror att det mycket väl skulle kunna låta sig göras, utan interaktiv systemutveckling är en kontinuerlig process snarare än enskilda utvecklingsansatser. För tre av de mindre företagen är dock tillämpningen direkt relaterad till den operativa verksamheten.

D2. Hur många timmar per dag används applikationsgeneratoren?

Planerat

Det finns en ekonomisk tes som säger att investeringar ska utnyttjas så mycket som möjligt. Interaktiv systemutveckling och applikationsgeneratorer är en stor investering och det kan kanske leda till att man tycker sig vara tvungen att sitta vid datorn mer än man kanske behöver bara för att visa att man använder den. Motsatsen kan också tänkas, speciellt om man kör på servicebyrå, att det är så dyrt att man inte har råd att använda den i någon större utsträckning. Det kan också tänkas att uppfattningen om hur applikationsgeneratoren påverkar det egna arbetet på något vis hänger samman med antalet timmar man använder den. Många gånger kan förmodligen inte användaren säga hur lång tiden är, ibland kan det hända att tiden varierar mycket, men man kanske kunde få någon form av maxel minvärde.

Resultat

Tiderna varierar från "en halvtimme", över "ett par timmar" till "i stort sett hela dagen". Det hänger hop med olika användarkategorier. En renodlad mellanhand använder applikationsgeneratoren mycket mer än inköpschefen som använder den för sina inköpskalkyler.

Samtliga intervjupersoner utom två använder datorn dagligen. En av dem var helt beroende av datasystemet. De som använt applikationsgeneratorer längst angav relativt korta nyttjandetider. En av de flitigaste användarna hade inte ens terminalen i sitt rum.

De tider på en à två timmar som anges får man ta med en nypa salt. Det varierar mycket från dag till dag. I början används applikationsgeneratoren betydligt mer i syfte att experimentera med dess möjligheter. Det är tveksamt om detta experimenterande ska räknas som användning. Det borde snarare hänföras till utbildning. Det förefaller således som användaren av applikationsgeneratoren efter två till tre månader blivit "fullärd" och därefter minskar på användningen. Jämför även med svaren under avsnitt E.

Avsikten med denna fråga var inte att få en exakt och precis upplysning utan snarare ett grovt mått på hur intervjupersonen själv upplever sin användning av applikationsgeneratorerna. Detta mått skulle naturligtvis kunna kontrolleras gentemot eventuell nyttjandestatistik som gjorts upp. Det skulle dock kräva alldeles för mycket resurser för denna enkla undersökning.

D3. Hur ofta behöver man ändra i befintliga program (motsv.) Vem gör det?

Planerat

Jag kan tänka mig två olika sätt att använda applikationsgeneratorer. Det ena är att alla program, alla rapporter, tabeller eller listor lagras "ifall man vill ha ut det någon mer gång". Det andra sättet är då det motsatta, att inget sparas utan allt görs från början var gång. Tänkbara förklaringar kan vara kostnader att använda applikationsgeneratorn, svårighet resp lätthet i att specificera en rapport, tabell eller listuttag. Användningssättet kan naturligtvis skilja från fall till fall och man bör följa upp med någon fråga om hur ofta man skriver nya program och hur ofta man tar bort program. Om applikationsgeneratorn blir demonstrerad kan man be att få se på programförteckningen.

Resultat

Det finns ingen klar trend här. En del redskap, som t ex ADI, inbjuder till att börja från början varje gång, medan andra mera inbjuder till att man lagrar sina rapporter etc. Många gånger är det nog i hög grad personberoende.

I sju fall av elva gör man inga ändringar, därför att det är lika lätt att skriva nytt från början. I ett fall, görs ändringar av mellanhanden på uppdrag av slutanvändarna. Det verkar dock ske något motvilligt. I ett annat fall, som berör APL-programmering, har programmen en inkörsperiod på uppemot ett halvår, då ändringar och tillägg görs, men sedan används programmen utan ändringar. I ytterligare fyra fall görs ändringar av dataavdelningen.

Med program har jag här avsett program skrivna i applikationsgeneratorns "språk" eller lagrade rapportbeskrivningar o dyl. Det har inte varit något problem att få intervjupersonerna att förstå detta, däremot har utomstående dataexperter och forskare hängt upp sig på ordet "programmering". Dock har jag inte lyckats hitta något bättre begrepp. "Procedurer" möjligen.

D4. Hur ofta skriver man nya program?

Resultat

Frågan har tolkats som om det frågades efter nya rapporter, kalkyler eller dylikt istället för program. Svaren stämmer väl överens med vad som angetts i frågan innan. De studerade applikationsgeneratorerna förefaller vara så pass enkla att handskas med att det är lättare att skriva en ny rapportbegäran än att leta upp den gamla. I två fall förekom flitigt experimenterande. Efter ett tag verkar man ha fått vad man behöver och då blir det inte så mycket "programmering".

D5. Hur ofta tar man bort program?

Resultat

Här avgavs bara två svar. Båda låter dem ligga kvar.

D4. Vem eller vilka använder applikationsgeneratorn? (Position i organisationen) Varför just dessa? Vem har bestämt det?

Planerat

Jag är ute efter organisatorisk position. Men jag är också ute efter vem som står för beslutet att just denna person ska använda applikationsgeneratorn. Kanske har användaren blivit pådyvlad ett redskap han eller hon inte själv varit speciellt intresserad av. Eller användaren har kanske själv varit en drivande kraft. Detta har betydelse för tolkningen av övriga svar.

Resultat

Man får intrycket att det bara är intervjupersonerna som använder applikationsgeneratorerna. Detta kan bero på att de är mer eller mindre handplockade som "flitiga användare". Men ett mönster går igen. 6 st hjälper andra användare att få tag i uppgifter de behöver Dessa är typiska mellanhänder. Av övriga är det elva som använder datasystemet för det egna arbetet och en som gör datasystem för andra med hjälp av applikationsgeneratorn. Precis som i fallet med skogsmätningar (Göranzon 1983a, b) har det uppkommit en mellanhandsfunktion, men till skillnad från det fall Göranzon beskriver är den här planerad och har en relativt hög position i organisationen. Mina undersökningar tycks indikera att användaren i allmänhet befinner sig i gränsskiktet strax under avdelningschef, alltså typiskt "middle management". Det har hävdats att denna kategori skulle vara rädda för att bli utslagna av sina chefers arbetsstationer (Ahrnell 1983, Asphjell 1984). Detta gäller endast mindre företag. Övriga chefer har inte tid att sitta och lära sig en besvärlig applikationsgenerator utan det får "middle management" personal klara av. Istället för utslagning har det blivit stöd.

D5. Hur många använder applikationsgeneratorn? Hur många är "erfarna användare" (gör t ex egna program)?

Planerat

De applikationsgeneratorer jag hade hört talas om innan undersökningen var typiska enarbetsplatser. Det fanns en användare som tjänstgjorde som förmedlare till andra personer som använde listorna, tabellerna eller rapporterna man fått ut. Denna fråga testar om det gäller även i det aktuella fallet.

Resultat

Bilden av hur många som använder applikationsgeneratorn är något splittrad. På alla stora företagen utom ett säger man 3-4 stycken. På detta säger användarna att en applikationsgenerator (ADI) är mycket spridd. Samtidigt säger Information Center personal att användningen inte är särskild spridd. Tolkningen av denna spridning eller kanske brist på spridning är för närvarande omöjlig att göra. På mindre företag är det i alla fyra fallen bara en användare.

Vad gäller "erfarna användare" tycks min definition hålla streck. Fem av intervjupersonerna menar att det finns användare, som i stort sett bara kan svara på frågor och den vägen få fram en rapport. Dessa är då icke-erfarna. Intervjupersonerna betraktar

sig själva som erfarna och det innebär att man kan "trixa" lite mer, ta ut rapporter andra vägar, lägga upp egna register. Just i att kunna lägga upp egna register, tycks det finnas en skiljelinje. Någon av de mera erfarna hade faktiskt ibland problem med detta.

D6. Har det tillkommit några nya och icke förutsedda användningar sedan applikationsgeneratoren togs i drift?

Planerat

Jag trodde att man anskaffat en applikationsgenerator för ett speciellt ändamål. Det är emellertid ett både kraftfullt och generellt redskap, som förmodligen skulle gå att använda till betydligt mer. Huruvida detta gjordes ville jag veta med denna fråga.

Resultat

Frågan är förmodligen inte alldeles korrekt ställd. Den baseras på föreställningen att applikationsgeneratoren skaffas för ett visst specifikt ändamål, men att man sedan upptäcker att den kan användas även till annat. Snarare tycks det vara så att man i och för sig skaffar applikationsgeneratoren för en viss tillämpning, t ex en viss arbetskrävande rapport, men att man vid anskaffningstillfället är medveten om att den kan och kommer att utnyttjas till många fler tillämpningar. Detta har också hänt i samtliga fall.

D7. Använder man uppgifter från det gamla systemet? (t ex register)

Planerat

Denna fråga hade också kunnat ställas under rubriken "egenskaper hos applikationsgeneratoren". En av de saker jag är ute efter här, är nämligen att få reda på om det är möjligt att utnyttja befintliga register. Den andra saken är naturligtvis om det görs. Jag föreställer mig att man i många fall drar sig för att lägga upp ett nytt stort register om man kan använda det gamla. Jag föreställer mig också att konstruktörerna av applikationsgeneratorer är medvetna om detta. Men jag kan tänka mig motsatsen också. Jag kan också tänka mig att företaget inte känner till att det är möjligt.

Resultat

I en tredjedel av fallen var man tvungen att använda separata register. Man hade där första generationens redskap. I regel var det möjligt att överföra data till dessa register från produktionsregistren, vilket i så fall skedde i satsvis form. Men i den andra tredjedelen använde man produktionsregistren med andra generationens redskap. Möjligheten att göra detta var ett starkt motiv för att skaffa det speciella verktyget framför något annat. I den sista tredjedelen använde man sig inte så mycket av register utan det var mera rena kalkyltillämpningar.

Bruket av applikationsgeneratorer har inte initierat någon större verksamhet kring informations- och datamodellering. I de studerade fallen var man snarare intresserad av att kunna göra saker med den information som finns än att göra informations- och

dataanalys och andra aktiviteter som behövs för att skapa en logisk och konsistent databas.

D8. Önskar du använda applikationsgeneratorn mer om så vore möjligt?

Planerat

Detta är en kontrollfråga för att få reda på om användaren är nöjd. Frågan om användaren är nöjd bör inte ställas i direkt anslutning till denna fråga.

Resultat

Samtliga användare ville använda applikationsgeneratorn mera. Detta kan möjligen tolkas så att användaren håller på att bli dataexpert! Att man är nöjd kan nog fastslås utan problem. En av intervjupersonerna varnade för risken att syssla med icke relevanta petimeterdetaljer.

D9. Hur mycket kostar det att använda applikationsgeneratorn? Händer det att man inte använder den på grund av höga kostnader?

Planerat

Den här frågan tror jag kan vara känslig. Det kan mycket väl hända att företaget inte vill erkänna att man inte använder applikationsgeneratorn av ekonomiska skäl. Lite lirkanden och antydningar krävs nog. Att det är dyrt att använda en applikationsgenerator bör också komma till läsarens kännedom. Ett av syftena med fallstudierna är ju att tjäna som rådgivning åt potentiella köpare av applikationsgeneratorer.

Resultat

Ekonomin tycktes inte spela någon större roll. Endast tre av nitton anförde ekonomin som hinder för användning av applikationsgeneratorerna. Det var inte heller svårt att få folk att svara på denna frågan. Beroende på kostnaden blev tillämpningen i ett fall väldigt egendomlig (satsvisa körningar). Av de övriga var det ingen som särskilt exakt visste hur mycket det kostade eller hur avgifterna debiterades.

D10. Har man några problem vid nuvarande användning?

Planerat

Här ville jag ha reda på alla typer av problem, driftsproblem, personalproblem, problem med datacentralen, med leverantören etc. De samlade upplysningarna tänkte jag presentera för leverantörerna och hoppas på förbättringar. Detta kunde vara en motivering för användaren att redogöra för sina problem.

Resultat

Ännu har jag inte träffad någon helt problemfri installation. Ett genomgående tema är manualerna (9 av 12 användare missnöjda), men annars finns inget speciellt. Här har istället presenterats en lång lista med olika problem. De är i huvudsak driftstekniska problem som att man trillar ur systemet och att väntetiderna är långa. Ett pro-

blem är mer organisatoriskt, nämligen att se till så dataavdelningen uppdaterar de gemensamma registren.

Ett annat problem angavs av en person på Information Center. Han tyckte det var ett problem att man inom företaget satt så fast i APL. Med hjälp av APL kunde man göra det mesta, men det finns få personer som är duktiga på APL. Alltså snarare ett personalproblem.

Det framförs också bitvis ganska skarp kritik mot tre av de omnämnda redskapen. Kritiken gällde främst driftsproblem. I ett fall ansåg man till och med att produkten inte var färdig att säljas. Det kan också noteras att samtliga tre redskap är svenska.

D12. Vilken nytta/intäkt har man av applikationsgeneratoren?

Planerat

Denna fråga hänger delvis samman med motiveringen ovan under C. Den användas som kontrollfråga för att se om tänkt och faktisk användning är desamma men den har även ett värde i sig själv. Jag föreställer mig att potentiella användare av applikationsgenerators och även tillverkarna av dem är mycket intresserade av att få svar på denna fråga.

Resultat

Detta är en fråga som borde ha ställts, men aldrig ställdes. Den fanns med i mina kommentarer, men i det frågeschema jag hade med mig hade den fallit bort. Ett allmänt intryck har jag dock. Samtliga användare, utom möjligen en, uttryckte någon gång under samtalet att de hade haft mycket stor nytta av applikationsgeneratoren. Någon siffra nämndes dock inte och jag tror det hade varit svårt att få fram någon. För en av användarna var frågan irrelevant, eftersom applikationsgeneratoren för henne var ett mycket klart uttalat arbetsredskap, vilket hon alltid haft och förutan vilket hon inte skulle kunna fullfölja sitt jobb. Nyttan anges i övriga fall indirekt genom bättre kvalitet på arbetet, både resultatmässigt och typografiskt.

4.2.5 E. Effekter på yrkeskunskap

Allmänna kommentarer

Detta är ett av de intressantaste områdena när det gäller interaktiv systemutveckling. Det mest uppenbara skälet att använda datasystem är ju att de på något vis ska "förbättra" företagets verksamhet. Interaktivt utvecklade system syftar direkt till att förbättra den enskilde personens arbete. Att undersöka hur yrkeskunskaperna påverkas av denna process och av att använda ett sådant system förefaller vara av centralt intresse. Att ta reda på effekterna på yrkeskunskap är emellertid mycket svårt, för de märks inte. Användaren hinner vänja sig och följaktligen ofta är omedveten om effekterna. Ett sätt är att istället fråga andra personer om hur de uppfattar användarens

"skicklighet" efter det att applikationsgeneratoren tagits i bruk. Om det är flera användare kan man be dem bedöma de andra. Det är dock mycket svårt att få tillförlitliga svar inom detta område och det är väldigt lätt att intervjuaren gör feltolkningar.

Det har varit svårt att få fram uppgifter inom detta område eftersom många endast använt applikationsgeneratoren en kort tid eller börjat sitt arbete efter det att den tagits i bruk.

E1. Hur har kvaliteten på ditt jobb förändrats sedan du började använda applikationsgeneratoren?

Planerat

Detta är den raka frågan. Har man tur räcker det att ställa den, så kommer allt som ett rinnande vatten. Men då ska man nog ha tur. Begreppet "kvalitet" kan vara något svårbegripligt. Man kan möjligen istället fråga om användaren gör ett "bättre" jobb och sedan be vederbörande definiera vad han eller hon menar med "bättre". Fråga kan möjligen på något vis vara "onaturlig" för användaren, eftersom vederbörande nog inte tänker på sitt jobb i sådana termer. Den som intervjuar måste se till att så mycket som möjligt fånga den exakta innebörden.

Resultat

Mina farhågor var alldeles ogrundade! Allt kom som ett rinnande vatten! Men det kan bero på att alla användare var akademiskt utbildade och var vana att tänka i abstrakta termer. 10 av 12 användare säger att det går fortare och att man hinner med mer än tidigare. Samtliga säger också att kvaliteten i något avseende (snabbare, bättre, nya möjligheter) har förbättrats. En intervjuperson anser inte att han gör ett bättre jobb, utan istället snabbare. Han betraktar datasystemet helt enkelt som en bättre räknemaskin. Vederbörande är en av de som använt applikationsgeneratorer längst tid.

E2. Hur har själva jobbet förändrats?

Planerat

Denna fråga är bredare och tar upp även annat än yrkeskunskap. Eventuellt kan man ställa denna fråga före frågan om kvaliteten på jobbet. Det borde finnas möjligheter att ställa motfrågor och be om detaljer och få en tämligen saklig beskrivning av hur arbetet förändrats.

Resultat

Svaren på denna fråga bekräftar svaren från föregående fråga. Men i ett av fallen gör man fortfarande efter 3 år manuella parallellberäkningar. Intressant är också en av användarna, som på en lista direkt kan se att siffrorna är fel. Jämför med Per Svenssons beskrivning över felsökning i datarapporter (Göranzon 1983a, s 29).

E3. Vem bestämmer beteckningar och text på bildskärmen?

Resultat

I samtliga fall gör användaren det. Frågan verkar vara rätt så meningslös.

E4. Hur tycker du att dina kamraters arbete har blivit påverkat av deras bruk av applikationsgeneratoren?

Planerat

Denna fråga var baserat på samma idé som projektiva test inom psykologin, dvs genom att fråga om någon annans inställning får man även en indikator om den egna inställningen. Om man vill betona detta kan man kanske ställa en följdfråga som ungefär lyder: Hur tycker dina arbetskamrater att deras arbeten blivit påverkade av arbetet med applikationsgeneratoren? Det finns även en etisk aspekt, som den enskilde forskaren måste ta ställning till.

Resultat

Svaret på denna fråga var något oväntat. Det finns inga andra arbetskamrater som använder applikationsgeneratoren! Därför är det svårt att tycka något om hur deras arbeten blivit påverkade. Samtidigt antyder fyra av intervjupersonerna att det finns andra personer som använder applikationsgeneratorerna, men att de använder dem mycket litet. Två av dem säger också att andra blivit "frälsta" och på Information Center ser man sig ofta som "missionärer".

E5. Applikationsgeneratoren erbjuder dig vissa möjligheter (kanske t ex understrykning av vissa ord, sortering på olika begrepp osv). Hur många av dessa utnyttjar du? Har dessa möjligheterna någon direkt relevans för din yrkeskunskap eller resultatet av ditt arbete? Är de kanske bara "gimmicks" som applikationsgeneratoren tillhandahåller?

Planerat

Bakgrunden till denna fråga är ett i mitt tycke närmast perverst utnyttjande av datorns möjligheter till olika videoattribut som t ex omvänd video med hel eller halv intensitet, olika intensitet på texten, understrykningar, skyddade fält, blinkningar och pipningar m m. Jag har också iakttagit mitt eget arbete med textbehandling där jag har blivit väldigt intresserad av att kunna använda t ex Diablo-skrivarens möjligheter, utan att det egentligen är nödvändigt. Det tycks som om bara det faktum att skrivaren kan skriva med t ex fetstil gör att jag använder det, även om texten i sig inte kräver det. Ännu värre har det blivit sedan jag började använda Macintosh med möjlighet till olika typsnitt, bilder m m. Vad jag ytterst är ute efter är att yrkeskunskap kompletteras med teknisk handhavandekunskap, som kanske ibland tar överhanden och dominerar. Det kan t ex bli så att layout-mässiga synpunkter på rapporterna tar överhanden över de innehållsmässiga.

Resultat

Svaren på detta frågekomplex är magra. Det kan bero på att applikationsgeneratorerna inte erbjuder så många "gimmicks" utan bara nyttiga funktioner. En hypotes att jobba vidare från kan vara att användaren nyttjar de kommandon han eller hon kan. I de fyra fall jag explicit ställde frågan var det bara ett tiotal kommandon som användes.

Tre personer sa att snygga rapporter var viktigt. Det stora intresset att utnyttja overheadframställningsmöjligheterna tyder också på att min föreställning inte är alltför verklighetsfrämmande. Två andra användare sa sig utnyttja nästan alla de funktioner som redskapen erbjuder.

Frågan är också svår att ställa eftersom det är svårt att i en kort mening formulera precis vad man är ute efter. Kanske skulle det vara bättre att fråga efter om intervjupersonen vill ha fler funktioner eller möjligen om det finns onödiga funktioner. Man kan ev undersöka hur många funktioner (kommandon) som finns totalt, hur många som intervjupersonen känner till och hur många vederbörande använder med någon viss minimifrekvens. Om man hade gjort upp en lista över vanliga typer av kommandon i förväg hade detta underlättat. Men det blir en mer omfattande undersökning än jag hade tänkt mig.

E6. Hur tycker du att organisationens arbete som helhet blivit påverkat av applikationsgeneratorm? Finns det några påtagliga belägg för det?

Planerat

Detta är egentligen samma fråga som första frågan om förändringar i yrkeskunskap. Det är bara ett annat sätt att närma sig samma problem. Här kommer vi in på möjlighet att ge service åt övriga organisationen, att snabbt och tillförlitligt få fram beslutsunderlag som inte gick att få fram tidigare. Men man kunde också komma in på hur pass tillförlitliga uppgifterna från datasystemet är och hur pass beroende man är av det. Denna fråga kan också anknyta till målsättningen med applikationsgeneratorm och uppfyllandet av denna målsättning.

Resultat

Alla intervjupersonerna säger att deras arbete blivit bättre. Detta har indirekta effekter för andra personer som är beroende av detta arbete. En sådan är ökad kontroll av tredje part. En annan effekt är möjligheten att i organisationen sprida hjälpmedel att bedriva verksamheten bättre. I ett sådant fall har idé, inköp och utveckling helt och hållet gjorts av en användare.

E7. Tyckte du det var svårt att lära dig jobba med applikationsgeneratorm? Hur lång tid tog det innan du kände dig säker?

Planerat

Tanken är att om man har lagt ner en massa tid och arbete på att lära sig ett redskap borde man vara mer benägen att se det positiva i det och mindre benägen att se det

negativa. Men hursomhelst är det viktigt att dokumentera om användarna tycker en speciell applikationsgenerator är svår eller lätt att lära sig.

Resultat

Ingen tyckte det var särskilt svårt. Alla kom ingång efter bara några timmars jobb. Den gemensamma utbildningsmetoden är "trial and error", dvs man sätter sig ner vid maskinen och prövar. Efterhand lär man sig alltmer avancerade saker. Man tycks vara fullärd efter ungefär 3-6 månader. Då finns det inget nytt att lära sig.

E8. Vilken utbildning har du fått i användning av applikationsgeneratorn? Vad tyckte du om den?

Planerat

Kunskapen att hantera applikationsgeneratorn blir till en del av yrkeskunskapen. Ett ideal är att det ska vara lika enkelt och självklart att använda en applikationsgenerator som en skrivmaskin. Precis som en driven maskinskrivare skriver snabbare och säkrare än en otränad så kommer naturligtvis en tränad applikationsgeneratoranvändare att nyttja denna på ett annat och bättre sätt än en otränad, men jag hävdar att en otränad eller okunnig ska ändå inte vara helt utlämnad utan ha en rimlig chans att få gjort vad han eller hon vill ha gjort. Om en omfattande utbildning är nödvändig för att kunna använda en applikationsgenerator tyder detta på att den inte uppfyller ovan angivna ideal. Det är därför viktigt att intervjuaren följer upp frågan med vad användaren tyckte om utbildningen. Fråga gärna om den var nödvändig överhuvudtaget. Om redskapet demonstreras kan man själv bilda sig en uppfattning om hur svårt eller lätt det är att använda sig av redskapet.

Resultat

Den gemensamma utbildningsmetoden är "visning" och "trial and error". Efterhand lär man sig alltmer avancerade saker. På två av företagen fanns det kurser, men bara en av intervjupersonerna hade gått dem. Han ansåg kursen vara meningslös. Men samtliga intervjupersonerna tyckte att "trial and error" var alldeles rätt utbildningsform. Mest imponerad blev jag av en intervjuperson som själv lärt APL utan hjälp av varken läroböcker eller manual utan endast genom att studera exempel och genom eget experimenterande.

Samtidigt kan man tänka på Festingers teori om kognitiv dissonans (1957). Den skulle tillämpad i detta fall resultera i att intervjupersonen tyckte att det sätt han lärt sig redskapet på är det bästa och ha svårt att se några andra alternativ. Emellertid tillämpade Festinger sin teori på monotona rutinarbeten. Några sådana är det inte fråga om här, varför man inte säkert direkt kan överföra resonemanget.

E9. Vad tycker du om manualer, bruksanvisningar och liknande?

Resultat

Alla applikationsgeneratoranvändarna utom en tyckte att manualerna inte var särskilt bra. I något fall t o m direkt hopplösa! Däremot var man mycket förtjust i de

hjälp texter man kunde få ut på skärmen. Felmeddelandena från de APL-baserade redskapen var inte heller bra, för det var tydligen APL:s egna. Enda nyttan man hade av dem var att man fick reda på att något var fel.

E10. Tycker du att det språk du använder i ditt yrke har förändrats sedan du började jobba med applikationsgeneratorer? Om så är fallet, har det påverkat ditt yrke?

Planerat

Detta är en svår fråga. Den bör nog på något vis konkretiseras eller exemplifieras för att bli begriplig. Bakgrunden är att förtrogenhetskunskap rimligtvis inte kan beskrivas i ett datasystem. Om användaren själv gjort det och själv använder det föreställer jag mig att den ändå på något vis kommer in i behandlingen och ger riktiga resultat. Om användaren tycker att språket har blivit påverkat så har även yrkeskunskapen blivit påverkad. Vilka olika typer av påverkan kan man då tänka sig? Man kan t ex tänka sig att man mer tänker i termer av vad man *kan* göra med applikationsgeneratorn än vad man *vill* göra. Man kan tänka sig att sådana ord som t ex "filer", "sökprofiler", "sökvillkor", "procedurer", "kommandon", "listlayout", "printar ut", "skriver in" etc används av användaren i sitt arbete som ersättningar för andra begrepp eller som kompletteringar till andra. Detta märks vid intervjun när användaren beskriver sitt arbete.

Resultat

Det finns uppgifter från 9 personer. Endast två av dem antydde att sättet att tänka vid problemlösning förändrats. En tyckte att han lärt sig en del engelska datatermer. Övriga ansåg inte att deras yrkesspråk eller tankesätt förändrats. Men, som en intervjuperson påpekade: "Sånt märker man inte."

Svaret på denna fråga måste nog tolkas med försiktighet. Det är risk att intervjupersonerna inte riktigt uppfattat vad jag egentligen avsåg med den. En av dem, anger att de enskilda kommandona är på svenska. Detta tror jag kan vara väsentligt, eftersom man i så fall bättre begriper innebörden i ett kommando. För en svensk, som normalt sett inte pratar engelska måste det vara begripligare att skriva "spara" istället för "save".

Jag försökte även under samtalen lyssna på det språk intervjupersonen använde för att se om fanns många "dataord" däri. Men de var ytterst sparsamma. En person använde "printa ut" en gång och en annan undrade om det fanns något "memo" till honom.

E11. Stämmer det att din yrkeskunskap har ersatts med eller åtminstone blivit kompletterad med teknisk handhavandekunskap av applikationsgeneratorn?

Planerat

Detta är också en fråga som egentligen redan är ställd. Avsikten är att få reda på hur pass väsentlig del av sitt arbete användaren anser denna tekniska handhavandekunskap är. Jag kastar här fram en hypotes, baserad bl a på Bo Göranzons forskning

(1983a, b), om att så är fallet. Frågan är förmodligen överraskande för användaren och svaret riskerar bli tämligen ogenomtänkt, om det blir något överhuvudtaget. Frågan följs delvis upp senare i det sociala avsnittet.

Resultat

Denna fråga är nu formulerad som en ledande fråga. Samtliga instämde också. En av intervjupersonerna påpekar vikten av att kunna systemet och inte bara redskapet, s k systemkunskap. Man kan tänka sig att denna är en form av förtrogenhetskunskap medan redskapskunskap är handhavandekunskap.

4.2.6 M. Effekter på arbetsuppgifterna

Allmänna kommentarer

Dessa frågor har blivit tillagda senare, men hör logiskt sett hemma här. Därför har jag flyttat hit frågorna och kommentarerna. I Flensburg 1984b och 1984a finns de sist i rapporten.

M1. Upplever du arbetet med applikationsgeneratorn mer monotont eller mekaniskt än tidigare?

Resultat

I två av fallen förekommer inslag av rutinarbete och monoton. I ett av dem finns det visserligen inte angivet, men rimligtvis måste det vara tråkigt att mata in en samling bokstäver och siffror från ett körschema varje månad, utan att veta vad eller varför man gör så eller så. I alla övriga fall är det tvärtom, arbetet har blivit bättre.

M2. Har det uppkommit någon form av mellanhandsfunktion vid användningen av applikationsgeneratorn?

Resultat

Sju av tolv var mellanhänder. Det som gjorde att de var mellanhänder var

- Fördjupad kunskap om handhavandet av applikationsgeneratorn
- Kunskap om vilka register som finns och deras struktur
- Kunskap om hur ett visst problem kan lösas med hjälp av applikationsgeneratorn.

De övriga personerna som själva byggt upp system eller rutiner för sitt arbete har alla dessa slagen av kunskap. Däremot finns inget mönster i mitt material för vem som blir mellanhand. Det tycks vara en kombination av personlighet och slumpen.

M3. Vilka övriga effekter har applikationsgeneratoren haft på uppläggningsen av ditt arbete?

Resultat

Det är egentligen bara en användare som fått andra effekter på arbetsorganisationen. Han har i stor utsträckning blivit sin egen sekreterare. Det är främst ordbehandlingen som haft denna effekt. I tidskrifter (t ex Data Effektiv nr 1, 1984) har jag sett intervjuer med personer som "med ordbehandlaren i gång fyller ut-korgen på nolldtid". Där emot tycks motsvarande effekter inte vara lika uppenbara vid användningen av register- eller kalkylredskap.

M4. Har applikationsgeneratoren medfört några andra organisatoriska förändringar?

Resultat

Den enda förändring som har skett är inrättandet av en Information Center avdelning eller motsvarande. I ett av fallen hade man anställt en särskild dataexpert, som skulle ha hand om utveckling av nya tillämpningar. I praktiken kom han att köra satsvisa körningar.

4.2.7 F. Psykologiska effekter

Allmänna kommentarer

I detta avsnitt har jag lagt in frågor om självförverkligande, yrkesstolthet, uppskattning och liknande. Även detta är ett i dubbel bemärkelse "känsligt" avsnitt. Det påminner samtidigt ganska mycket om föregående avsnitt om yrkeskunskap, så det förefaller rimligt att låta avsnitten komma efter varandra. En lämplig frågeteknik i detta avsnitt kan nog vara att ställa ja/nej frågor, som följs upp med ett "varför?" eller "hur då?".

F1. Känner du dig mer uppskattad av din chef eller dina arbetskamrater eller dig själv sedan du började använda applikationsgeneratoren?

Planerat

Detta är en rak fråga om uppskattning från andra. Preciseringsen till olika kategorier kan vara ett sätt att komma igång. Det kan naturligtvis tänkas att även andra kategorier kan uppskatta och intervjuaren bör vara öppen för det. Orsakerna till den eventuellt ökade uppskattningen är viktiga att försöka få fram. Det återverkar ju direkt på målsättningen. Det kan ju också tänkas att organisationen är en "Jante-organisation", dvs en organisation där man aldrig visar någon som helst uppskattning av vad någon annan har gjort och alla personer som kommer med något nytt initiativ bemöts med attityden "Du ska inte tro att du är något!". Detta är det första budet i Jantelagen (Sandemose, 1933). Förhoppningsvis märker den intervjuade ganska snart om det är

en Jante-organisation eller inte. En trevare om t ex "Vad skulle behövas för att din chef ska uppskatta dig mer?" kan möjligen kunna ge någon ytterligare indikation på hur pass vanligt det är att visa uppskattning eller hur mycket som behövs för att bli uppskattad. Om det är möjligt kan man ställa frågan till någon av arbetskamraterna.

Resultat

Samtliga användare av applikationsgeneratorer känner sig mer uppskattade. Det som förvånar mig är att man så pass noga skiljer på att själv vara uppskattad och att ens arbete blir uppskattat. Men samtidigt verkar det inte vara speciellt viktigt för intervjupersonerna om de är mer eller mindre uppskattade. Möjligen kan man spåra en inställning hos några stycken att det egentligen bara är de själva som riktigt förstår vad de gör.

F2. Har någon fått mer inflytande sedan applikationsgeneratorn togs i bruk? Kan-ske rent av du själv?

Planerat

Frågan bör ställas med finesse och försiktighet. Det bästa är nog att försöka fråga någon som inte är direkt berörd, t ex chefen eller någon annan arbetskamrat. Dagens applikationsgeneratorer är fortfarande så pass krångliga att det behövs någon typ av expert för att handskas med dem och denna expert kommer då att bli en informationsförmedlare till de andra. Svaret på denna fråga bör på något vis jämföras med hur många som använder applikationsgeneratorn. Man kan också anknyta till den historiska beskrivningen.

Resultat

De som har mellanhandsfunktioner har fått mer makt. Men man vill inte direkt erkänna det. Man var kanske inte medveten om det heller. Två användare ville båda sprida nyttjandet av applikationsgeneratorerna mera. En av dem hade inte behov av större makt, eftersom han redan var chef och den andre tyckte det var direkt fel att han skulle ha så mycket makt. För de fyra små företagen var användaren redan chef och hade all makt som behövdes.

F3. Har du blivit stoltare över ditt yrke sedan du började använda applikationsgeneratorn?

Planerat

Detta är en variant på föregående fråga. Det gäller den egna uppskattningen och denna formulering kan vara mer tilltalande än att fråga efter om man uppskattar sig själv. Naturligtvis skall frågan följas upp med på vilket vis användaren känner sig stoltare än förut.

Resultat

Fem personer har svarat på denna fråga. Alla utom en har blivit stoltare. Men de vill inte gärna säga det rent ut. De framgår också från de andra intervjuerna att användarna har blivit stoltare.

F4. Har du någon gång upplevt att applikationsgeneratoren "inte har räckt till" för det du har velat göra? Hur kändes det? Orsak? Händer det ofta?

Planerat

Det kan diskuteras om denna fråga ska stå under denna rubrik eller under någon annan. Som den är formulerad kan den uppfattas väldigt "sakligt" dvs man får en redogörelse för sakförhållandena kring en eventuell otillräcklighet. Detta är naturligtvis också viktigt och tjänar till grundval för att föreslå förbättringar av applikationsgeneratorerna. Men det jag annars är ute är hur det känns att vilja göra en sak, men inte kunna därför att en dum dator lägger hinder i vägen. Orsaken kan bero på verktyget i sig, men det kan också bero på okunskap om hur man ska handskas med det. Om bristen verkar osannolik, kan intervjuaren be att få titta på det och eventuellt om han eller hon har tur, komma på ett sätt att rätta till det eller hitta motsvarande i manualen.

Resultat

Två användare menar att applikationsgeneratoren räcker till, men däremot inte de registeruppgifter som finns bakom. I fyra av fallen finns det ingen uppgift. Övriga som svarat har klagomål på applikationsgenerator och maskinutrustning. I tre fall vill man ha fler funktioner, i två fall är minneskapaciteten för liten och i ett fall tycker man att redskapet behöver utvecklas ytterligare.

Frågan har uppfattats väldigt sakligt, precis som jag befarade. Den person som bäst beskrivit vad jag egentligen var ute efter är APL-användaren, som ger en god och målade beskrivning av hur det är när man gett sig själve hm hm på att lösa ett problem och inte får igenom det. Det är nog svårt att komma åt andemeningen i denna fråga, eftersom de flesta användare, trots att de är mellanhänder, inte använder datorn speciellt mycket. Dessutom är nog frågan irrelevant för de flesta, eftersom man efter 3-6 månader har utforskat applikationsgeneratorns möjligheter och sedan helt enkelt inte råkar ut för att den inte räcker till.

F5. Finns det några arbetsuppgifter som är speciellt lätta att göra med applikationsgeneratoren? Vad har detta för inverkan på övriga uppgifter? Kan det tänkas att du någon gång gör en "applikationsgeneratoruppgift" när du egentligen borde ha gjort någon annan?

Resultat

Jag har funnit det ganska meningslöst att ställa denna fråga eftersom den besvaras i avsnitt A.

F6. Händer det någon gång att du skyller på datorn när något blivit fel i ditt arbete?

Resultat

Uppgiften är ganska meningslös eftersom resultatansvar gäller för samtliga. Endast en av mellanhänderna, som var mycket beroende av datorn, kan skylla på den när den är trasig eller dataavdelningen inte uppdaterat registren.

F7. Har du någon gång låtit bli att göra något, därför att det inte gått eller du inte känt till, hur man skulle göra det med applikationsgeneratoren? Hur ofta då?

Resultat

Den här frågan är alldeles för invecklad! "Vad gör du när du kört fast?" borde den lyda istället. Svaren på denna finns i materialet. Hälften ber Information Center om hjälp, den andra hälften försöker lösa det själv.

F8. Har du någon gång känt dig osäker eller otrygg i ditt arbete med applikationsgeneratoren? Varför gjorde det? Vad gjorde du då?

Planerat

Det förefaller rimligt att känna sig otrygg eller osäker i början och det är kanske inte mycket att säga om det då. Men om det fortsätter blir det ett allvarligt problem och påverkar användarens psykiska arbetsmiljö i stor utsträckning. Orsakerna till detta är viktiga, det kan röra sig om lokala förhållanden, kanske rent av något så enkelt som försummad utbildning. I detta fall kunde kanske forskaren lite diskret säga det till någon ansvarig. Det kan också vara betydligt mer komplicerade förhållanden men fortfarande lokala. Även i detta fall kunde forskaren göra en insats. Men en kvarstående osäkerhetskänsla kan också bero på verktyget i sig, att det har en utformning som gör att användarna känner sig osäkra. I så fall är det ett viktigt forskningsresultat.

Resultat

Endast fyra personer har svarat på denna fråga. En av dem undrar ibland över innehållet i det hon får fram (jfr felsökningen i Göranson 1983b). En annan var aldrig osäker på gamla utomordentliga manualer och anvisningar. Han hade ändå ingen direkt hjälp från Information Center, såvitt jag kan förstå. En persondatoranvändare hade ibland svårt att hitta fel i programmen han själv skrivit. Då brukade han tillfråga en extern konsult. En annan persondatoranvändare tyckte ibland att han hade för små kunskaper.

F6. Händer det att du sitter och pysslar med applikationsgeneratoren bara för att du tycker det är spännande? Hur ofta då?

Planerat

Här är jag ute efter "hacker"-symptom. Frågan är bra, för folk svarar nog tämligen ärligt och rakt fram på den. Men mycket "lekande" tyder på en klar förskjutning från yrkesintresse till maskin eller verktygsintresse. Eller också på en sällsynt förmåga att skilja på arbete och nöje, även under arbetstid. Vid positivt svar kan man eventuellt följa upp genom att fråga om användaren skulle vilja syssla mer med applikationsgeneratoren, kanske på heltid. Man kan spekulera över orsaker till detta "lekande", om

det beror på människan eller miljön eller om vissa verktyg kanske inbjuder till extra mycket "lek", medan andra inte gör det. Om så är fallet är egenskaperna hos de olika typerna av applikationsgeneratorer en viktig och värdefull upptäckt.

Resultat

Sex personer säger att de gör det. Det ingår som ett led i "trial and error"-utbildningen, men det blir inte särskilt mycket av det efter det att man upptäckt alla möjligheterna. Två personer säger att de har inte tid.

4.2.8 G. Effekter på arbetsledning och arbetsorganisation

Allmänna kommentarer

Frågorna under detta avsnitt behandlar möjligheten för användaren att själv lägga upp sitt arbete, göra det i den ordning han eller hon tycker är bäst utan att vara styrd av något datasystem. När man inför olika typer av datasystem riskerar man att de formaliserbara uppgifterna blir överrepresenterade och att de oformaliserbara blir undertryckta. Ett sätt detta visar sig på är att användaren känner sig styrd av datasystemet, att han eller hon tycker arbetet är mekaniskt och monotont, att han eller hon inte har möjlighet att organisera arbetet på samma sätt som tidigare. Ibland är denna känsla inte medveten, speciellt om applikationsgenerators är så ny att det fortfarande är spännande att arbeta med den.

Ett helt annat sätt att resonera är att säga att de formaliserbara och rutinmässiga uppgifterna nu går så pass fort att användaren har tid att ägna sig åt andra och i längden viktigare saker. På så sätt tar datorn över de tråkiga uppgifterna och människan får hand om de viktiga. Frågorna i detta avsnitt ska ge indikationer om vilken föreställning som användaren uppfattar som den riktiga. Felkällor kan vara att användaren är mycket positivt eller mycket negativt inställd till datorer och att denna inställning färgar av sig på hans eller hennes upplevelse av arbetet. Å andra sidan är det ju användarens uppfattning av det egna arbetet jag vill åt, så det är ur den synvinkeln inte fel.

G1. Kan du själv bestämma hur du vill utföra ditt arbete? Har detta ökat eller minskat sedan applikationsgenerators togs i bruk?

Planerat

Det enda rimliga svaret borde vara "ja, ökat", men jag tänkte mig motsatsen också. Med hjälp av applikationsgenerators kan man uppnå mycket hög grad av självbestämmande, mycket mera än med traditionella datasystem. En förutsättning för detta är att användaren själv lägger upp sitt system, eller åtminstone har mycket stort in-

flytande över denna uppläggning. Jag misstänkte att det ofta var datafolk som gjorde första systemet, bara för att användaren skulle komma igång. Detta system var förmodligen en färdig lösning till något av dataexperten uppfattat behov. Det var emellertid inte säkert, och dataexperten håller nog också med, att det var rätt lösning eller ens rätt problem. Men användaren kom in i ett tänkesätt och fick kanske svårt att frigöra sig från detta. Det var kanske inte heller så enkelt att ändra på alltihop utan att förlora t ex alla register.

Resultat

Åtta personer har svarat på frågan och alla hävdar att de själva bestämmer över sitt arbete. Alla utom en har gjort det hela tiden och någon ändring har inte inträffat. I ett av fallen finns det arbetsuppgifter som är detaljreglerade och intervjupersonen försöker lägga över dessa på andra personer. Av intervjuerna med de övriga tre användarna framgår också att de själva helt bestämmer över sitt arbete med datorredskapen.

G2. Har arbetsmängden förändrats sedan du fick applikationsgeneratoren? Har lönen förändrats motsvarande?

Planerat

Jag spekulerade ovan i att applikationsgeneratoren skulle avlasta användaren från en mängd tråkigt rutinarbete. Denna avlastning trodde jag används för att lasta på mer arbete. Kanske också mer rutinarbete, som då är tänkt att göras med hjälp av applikationsgeneratoren. Med denna fråga kollade jag detta. Frågan om lönen var inte speciellt viktig, men jag föreställde mig den kan ha intresse ändå.

Resultat

Åtta stycken säger att arbetsmängden ökat, men att man trots det inte arbetar längre tid. Ett svar är något osäkert, eftersom man gjort en massa omorganiseringar under tiden. I ett annat fall ansåg man arbetsmängden vara oförändrad. En tyckte sig ha fått sämre löneutveckling än sina tidigare arbetskamrater och mycket mera övertid.

G7. Har jäkt och stress förändrats sedan applikationsgeneratoren togs i bruk? Om så är fallet varför det?

Planerat

Det hävdas att deadlines skapar stress, att alltför långa väntetider skapar stress, att alltför korta väntetider skapar stress, att krånglande datorer skapar stress, först när man inte kan göra något och sedan när man ska ta igen det förlorade arbetet. Allt detta är tämligen belagt för traditionella datasystem. Nu tror jag att så inte är fallet vid interaktivt utvecklade system och det är därför jag ställer denna fråga.

Resultat

Det som eventuellt kan vara stressande är att sitta och vänta vid långa väntetider. Endast en person anger man kan bli nervös för att inte hinna med innan man ska

hem. Den som programmerade i APL kände sig också ibland frustrerad under en intensiv programmeringsfas. En användare av Lotus 1-2-3 blev frustrerad av väntetider på en à två sekunder! Han tyckte inte att han hade något att göra under den tiden! En annan hade övertider på 40-50 timmar i månaden mot några enstaka tidigare. I övrigt var det inte någon större jäkt eller stress.

G4. Upplever du dig mer eller mindre styrd och övervakad sedan applikationsgeneratorn togs i bruk?

Resultat

Tre stycken har svarat med ett klart "nej". Av övriga svar framgår att detta gäller för de andra också.

G5. Har kontakten med dina chefer ökat eller minskat sedan du började använda applikationsgeneratorn? Om så är fallet vilken typ av kontakt och hur har den förändrats och vad tror du är orsak till förändringen?

Resultat

Sju svar är avgivna och alla säger att så inte är fallet. Möjligen kan den ökade produktionen leda till fler kontakter beroende på ökad mängd rapporter etc.

4.2.9 H. Sociala effekter

Allmänna kommentarer

Mycket av verksamheten i ett modernt företag bygger på lagarbete och avsikten med detta avsnitt är att undersöka vilka effekter interaktiv systemutveckling med hjälp av applikationsgeneratorer har för kontakten och samarbetet mellan användarna. Det kan röra sig om både kontakter mellan användare av applikationsgeneratorn och kontakter mellan användare och icke-användare. Det kan också röra sig om både kvantitativa och kvalitativa skillnader i kontakter.

H1. Känner du dig mer eller mindre isolerad sedan du började använda applikationsgeneratorn? På vilket sätt?

Planerat

Som vanligt börjar vi med en rak och enkel fråga. I uppföljningsfrågan skall forskaren försöka få någon form av grepp om både kvantitativa och kvalitativa effekter. Ökat eller minskat antal kontakter är tämligen enkelt att svara på, däremot kan det vara värre att ange de kvalitativa skillnaderna. Om man bara pratar om datorn och applikationsgeneratorn är detta en kvalitativ förändring. Likaså om användaren "å tjäns-tens vägnar" utbyter ett antal rutinmeddelanden med andra personer och det inte längre finns någon anledning eller möjlighet att "byta några ord" i förbifarten är detta också en kvalitativ förändring. Som forskare ska jag i detta läge undvika att själv ta

ställning om det är en förändring till det bättre eller sämre, men gärna fråga användaren vad han eller hon tycker om den saken!

Resultat

I sju fall säger man mindre isolerad. Fyra av dessa säger att kontakterna ökat. En säger sig ha blivit mer isolerad. Om vi ser på mellanhandsfunktionen innebär den täta kontakter med personer som vill ha hjälp med sina rapporter. Detta arbetssätt borde alltså rent generellt leda till ökade kontakter. Resultatet motsäger inte denna hypotes.

H2. Kommer dina arbetskamrater som inte har tillgång till applikationsgeneratorn och ber dig om hjälp? Hur ofta i så fall?

Planerat

Jag ville ta reda på spontan hjälpsökning, dvs någon som satt med ett problem och inte riktigt visste hur det skulle lösas eller bara hade lite vaga idéer. Vederbörande tror då kanske att man med datorns hjälp skulle kunna få fram något av intresse och uppsöker därför den som har hand om datakörningarna, dvs mellanhanden. Denne tar i detta fall aktiv del i problemlösningen och vidarebefordrar inte enbart informationsbehov. Detta är en typisk kvalitativ förändring i de sociala kontakterna. Men det måste klart framgå när frågan ställs att det är fråga om denna mer problemlösningsorienterade kontakt jag är ute efter. Här kan det också dyka upp alldeles motsatt fenomen, dvs att användaren tycker att folk kommer i tid och otid och vill ha ut listor, som man inte har nytta av eller redan har fått, men inte gitter leta upp igen.

Resultat

I 8 fall förekommer det. I två av dessa vill intervjupersonerna att även andra ska kunna arbeta med applikationsgeneratorn. Två användare hjälper ibland dataavdelningen, en med att hålla grafikkurser, en annan med ren APL-handledning. I det första fallet har kontakterna mellan avdelningarna ökat kraftigt.

H3. Pratar ni mycket om applikationsgeneratorn i pauser och sånt? Om så är fallet vad har denna konversation ersatt?

Planerat

Detta är också fråga om kvalitativa skillnader i den sociala samvaron. Svaret ger indikation på om yrkeskunskapen skiftar från kunskap om verksamheten till kunskap om tekniskt handhavande av ett hjälpmedel i verksamheten. Det skall då gärna jämföras med svaret ovan under rubriken yrkeskunskap.

Resultat

Tre intervjupersoner anger mycket prat om databehandlingen. Möjligen har en av dem något missuppfattat frågan. I de övriga fallen finns inga uppgifter.

H4. Skulle du vilja lära dig mer om applikationsgeneratorn? Vad är det i så fall som hindrar? Skulle du vilja känna till någon annan sådan?

Resultat

Samtliga intervjupersoner verkar i och för sig fullärda, men fem stycken säger att de gärna vill lära sig mer. Men en av dem tycker det är för dyrt och två andra har inte tid.

H5. Kan användarna skicka meddelande till varandra via applikationsgeneratoren? Om så är fallet, vilken typ av meddelanden skickas vanligtvis? Hur ofta?

Resultat

I åtta av nitton intervjuer har man ett elektroniskt brevsystem. I fyra av fallen kan man också skicka utskrifterna till vilken skrivare som helst på företaget och dess filialavdelningar belägna upp till 40 mil från huvudkontoret. Detta uppskattas mycket av användarna.

H6. Har det skett några personalförändringar på grund av applikationsgeneratoren?

Planerat

Det sägs att datorerna tar bort jobb, men många gånger blir det faktiskt nästan tvärtom. Interaktiv systemutveckling tror jag inte har så särdeles stora effekter på antalet jobb. Denna fråga är avsedd att kontrollera saken. Man bör dock kanske inte nöja sig med ett enkelt "ja" eller "nej" utan försök föra ett mer nyanserat resonemang. T ex om man inte hade haft applikationsgeneratoren, hade det varit lika många anställda då? Ger kanske det nya arbetssättet större möjligheter för företaget att "gå bra" och därmed kanske fler arbetstillfällen i framtiden? Personalförändringar kan kanske ibland vara ändringar i arbetsuppgifter eller omplaceringar eller "naturlig avgång".

Resultat

Det har inte skett några avskedanden på grund av interaktiv systemutveckling. Frågan är helt irrelevant. Man kan lika gärna fråga om telefonen eller skrivmaskinen har tagit bort några arbeten. Applikationsgeneratorer är ett rent hjälpmedel i arbetet. I ett fall har det dessutom medfört att ytterligare en person anställts.

4.2.10 I. Etiska effekter

Allmänna kommentarer

Dessa frågor är mycket svåra att både formulera och att besvara. Men samtidigt är de viktiga och jag tror att många människor är intresserade av att diskutera dem. Detta gäller både datafolk och användare. Frågorna jag vill ställa i detta avsnitt påminner mycket om frågorna kring yrkeskunskap. De är också i mångt och mycket "samvetsfrågor", dvs de handlar om att leva upp till någon form av högre ideal. Jag tror att en forskare har mycket större chans att få ärliga svar än t ex en konsult. Det är därför viktigt att verkligen ha visat god forskareetik under den tidigare intervjun så att den

man intervjuar verkligen har fått förtroende för en. Ett kanske inte fullt så etiskt tillvägagångssätt, men möjligen desto effektivare är att helt enkelt inte ställa dessa frågor i första omgången utan först i samband med att när man efteråt kontrollerar riktigheten helt "osökt" komma in på den här typen av frågor. Detta sätt har jag dock inte tillämpat i denna undersökning.

De föreställningar jag går ut med, när jag formulerar dessa frågor är att interaktiv systemutveckling ger mycket stora möjligheter till enskilt omdöme och därmed också gott etiskt handlande. Om intervjupersonen värjer sig mot "etik" kan forskaren försöka använda "omdöme", "sunt förnuft" istället.

I1. Har du blivit mer regeltrogen sedan du började använda applikationsgeneratoren?

Planerat

Denna fråga ansåg jag vara mycket viktig. Traditionella datasystem begränsar individens arbetsfrihet. Det är ett medvetet syfte. Talesättet "varför ge en människa chansen att göra fel då en dator kan göra rätt" är ibland förekommande. Denna styrande och kontrollerande uppgift hos traditionella datasystem bör inte ograverad överföras till interaktivt utvecklade system. Icke desto mindre förmodade jag att så var fallet. Orsakerna till denna förmodan var följande:

- Det är ofta en dataexpert som gjort systemet från början. Detta påverkar användarens möjligheter att göra radikala ändringar samt även användningen.
- Utbildningen av användaren lär ut ett visst sätt att använda applikationsgeneratoren. Detta färgar av sig i vederbörandes egen systemutveckling.
- Själva verktyget stöder ett visst användningssätt. Detta påverkar användningsmöjligheterna.

Den eventuella påverkan dessa förhållandena utövar kan visa sig på många sätt. Ett av de mera uppenbara är att det blir svårare att "bryta" mot reglerna.

Resultat

Det visade sig att frågan var irrelevant eftersom hela arbetsuppläggningsen är fri och självständig. Grunden för att ställa den finns inte eftersom det av användarna utvecklade datasystemet inte alls har det angivna styrande syftet.

I2. Kan du ge kunderna (el motsv) den service du skulle vilja ge dem? Hur har applikationsgeneratoren påverkat din möjlighet att ge tillräcklig service? Orsak?

Planerat

Denna fråga är något dubbelbottnad. Den etiska komponenten är om användaren tycker han kan ge den service som vederbörande önskar. Om så verkligen är fallet eller inte är för just detta sammanhang ointressant. Det är givetvis inte fel att registrera förändringar i servicen mot kunderna, men detta är en förändring i arbetsinnehåll, inte en etisk fråga. Det etiska kommer i samband med den personliga inställningen till detta sakförhållande. Om stora förändringar inträffat och användaren är

likgiltig inför detta, kan det tyda på en förslöande och förslappande inställning som följd av bruket av datasystemet. Men detta kan förstås vara användarens allmänna inställning. Detta bör intervjuaren försöka ta reda på eller åtminstone bilda sig en egen uppfattning om.

Resultat

I åtta avgivna svar har man svarat "bättre". Ingen likgiltighet påvisades. Detta tolkar jag som att de datasystem man själv utvecklat inte haft de ovan antydda effekterna.

13. Om du tror att företaget skulle tjäna på att du i ett viss fall bröt mot reglerna, kan du då göra det? Hur har applikationsgenerators påverkat denna "förmåga"?

Resultat

En intervjuperson svarar "nej" på frågan, men hennes kunder skulle gärna vilja att hon kunde det. I tre fall svarar man att det är möjligt, men att frågan är irrelevant. För övrigt är inga svar avgivna. Frågan baserades på en syn på dator som styrande och kontrollerande vilken inte stämde.

13. Tycker du att någon missbrukar applikationsgenerators? På vilket sätt i så fall?

Planerat

Här tillämpar jag en teknik hämtad från projektiva test, där man frågar någon om dennes åsikt om andras inställning istället för att fråga om vederbörandes egen inställning. Detta anses ibland ge en bättre indikation på personens egen inställning. Det är viktigt att det aldrig någonsin får framskymta ens skuggan av en misstanke att användaren kanske skulle kunna missbruka applikationsgenerators. Men andra kan ju... Om det inte skulle finnas några andra användare kan man fråga om tänkbara möjligheter till missbruk. Orsaken till att jag ställer denna fråga är att det borde vara viktigt både för företagen som använder applikationsgenerators och de som tillverkar dem att veta vilka möjligheter till missbruk som kan tänkas förekomma.

Resultat

"Missbruk" har här tolkats som "otillåten åtkomst av data". Åtta svar är avgivna, varav ett är meningslöst och de övriga är "nej". De aktuella systemen är alla öppna och all information är tillgänglig för dess användare. Därför har man inga hemligheter i dessa system. En av användarna sysslade med anbudsvärderingar och dessa kan i princip missbrukas. Men han litar på lösenord och "signon-kod".

4.2.11 J. Ändringar

Allmänna kommentarer

Att införa en applikationsgenerator är naturligtvis redan det en stor förändring i en persons eller organisations verksamhet. Hur stora och hur pass omfattande dessa är försöker jag få grepp om i detta avsnitt. Jag skiljer mellan olika typer av förändringar:

- Organisatoriska förändringar
- Förändringar i användningen (nya program t ex)
- Förändringar i själva applikationsgeneratoren.

Orsakerna till att jag vill veta något om dessa förändringar är flera. T ex kan en organisationsförändring medföra en del effekter som man måste ta hänsyn till vid undersökningen av applikationsgeneratorernas effekter. Önskemål om förändringar av själva redskapet är viktiga för de som tillverkar dem. Likaså kan förändringarna i användningen av applikationsgeneratoren såsom t ex ändringar av befintliga program vara viktiga att kartlägga. Om man sällan ändrar i befintliga program kan det bero på att det är svårt att ändra, att man har som praxis att inte ändra utan skriva nytt istället eller att det inte finns något ändringsbehov. Om det finns något generellt mönster i dessa svaren är det av betydelse för tillverkarna av applikationsgeneratoren. Även om detta inte föreligger kan det vara av intresse för potentiella köpare av ett visst verktyg avsett att användas i en viss verksamhet att på förhand ha någon uppfattning om kommande ändringsbehov. Det kan t ex motivera en mer omfattande utbildningsinsats.

Jag har ofta märkt att efter det att ett datasystem införts kommer det upp en mängd önskemål om förändringar. Dessa är många gånger vettiga och förståndiga, men går tyvärr inte att genomföra på grund av tids- och/eller penningbrist. Det är intressant att undersöka om samma förhållande gäller för interaktivt utvecklade datasystem.

Jag är också intresserad av att få veta om användaren tycker att någonting behöver ändras i själva applikationsgeneratoren. Kanske det behövs någon utvidgning i kommandospråket, kanske kommandon är onödiga etc.

J1. Vad vill du ändra på i applikationsgeneratorns nuvarande utformning? Har du försökt göra det?

Planerat

Detta är den öppna frågan om ändringar i själva applikationsgeneratoren. Det kan röra sig om nya kommandon, ändrade kommandon eller särskilda rutiner för att göra en viss sak.

Resultat

Svaren är blandade. Följande typer av svar avgavs:

Bättre teknisk utrustning (högre kapacitet): 3 st

Bättre grunddata: 2 st

Helt nya applikationsgeneratorer: 3 st, varav 2 st datafolk och en med erfarenhet av annan applikationsgenerator.

Utökning av befintlig applikationsgenerator: 2 st

Fleranvändarsystem: 2 st

Ingen ändring: 3 st

J2. Har du övriga förslag eller tankar om vad som skulle ändras i samband med bruk av applikationsgenerator?

Planerat

Detta är förslagslådan och gäller förslag till ändrad användning. I en följande fråga tar jag upp förslag till ändringar i själva applikationsgeneratoren. Jag föreställer mig att det är ganska svårt för användaren att hålla isär dessa båda nivåer och det borde få effekter i båda frågorna. Svårigheterna kan bero på applikationsgeneratoren, som i sig inte tvingar fram någon skarp skillnad eller kanske rent av bortser från den. Vissa skraddarsydda applikationsgeneratorer kan jag tänka mig har t ex kommandon som är hänförliga direkt till verksamheten.

Resultat

Tre svar är avgivna. Ett betonar bara vikten av kunskaper om data- och informationsmodell och de övriga två har inga förslag.

J3. Har någon omorganisation inträffat sedan applikationsgeneratoren togs i bruk? Om så är fallet hur påverkade detta datasystemet?

Planerat

I denna fråga är jag ute efter omorganisationer som inträffat av andra orsaker än interaktiv systemutveckling. Det jag vill ha reda på är hur denna omorganisation påverkade datasystemet, om det gick att anpassa det, om det inte behövdes eller om det inte gick att anpassa alls. Frågan utgör någon form av mätning på applikationsgenerators "robustness", dvs hur den tål att användas till andra saker än man ursprungligen tänkt sig.

Resultat

Sju svar är avgivna. Två säger att förändringar inträffat, men att de inte påverkat datasystemen. Detta tolkar jag så att de utvecklade datasystemen var huvudsakligen personberoende och inte organisationsberoende.

J4. Hur många gånger (per mån, vecka eller år) har du behövt be om hjälp i samband med ändringar av programmen? Vem har du i så fall bett?

Planerat

Detta handlar egentligen om lättheten att ändra i redan gjorda program eller eventuellt att lösa besvärliga problem i samband med att göra nya program. Jag föreställer mig att om man måste be om hjälp ofta drar man sig för att göra ändringar. Man kan också följa upp med att fråga hur många ändringar som görs utan hjälp.

Resultat

Nio svar är avgivna. Två stycken hjälper sig själv. Tre stycken har inga ändringsbehov. Övriga fyra säger sällan eller aldrig. De som hjälpt har varit Information Center eller motsvarande. I ett fall var det leverantören av redskapet och i ett annat fall var det tvärtom intervjupersonen som fick hjälpa dataavdelningen (med APL!).

J5. Vem sköter vidareutveckling och underhåll?

Planerat

Detta är en vidare formulering av föregående fråga. Som jag ser det kan antingen användaren själv stå för underhållet, dvs den löpande förändringen och utvecklingen eller också gör någon särskild utsedd person det, troligen någon dataexpert. Frågan försöker något penetrera ansvarsförhållandena och synen på användaren hos den aktuella organisationen.

Resultat

I fem fall Information Center-personal och i två fall användaren själv.

J6. Har det skett några ändringar i organisationen på grund av applikationsgeneratorn?

Planerat

I denna fråga är jag ute efter omorganisationer på grund av applikationsgeneratorn. Jag föreställer mig att det kan vara av intresse både för presumtiva användare och tillverkare av applikationsgeneratorer att få veta vilka typer av och hur pass omfattande omorganisationer man kan räkna med. Exempel på vanliga omorganisationer är bildandet Information Center eller inrättande av en särskild tjänst med uppgift att använda applikationsgeneratorn. Det behöver inte vara någon heltidssyssla, men bruket av redskapet skall vara organisatoriskt knutet till en antal personer.

Resultat

Enda ändringen är bildandet av Information Center eller motsv.

4.2.12 K. Förhållande till dataavdelningen (motsv)

Allmänna kommentarer

Söderström (Ahrnell 1983) hävdar att dataavdelningens motstånd förhindrar användandet av applikationsgeneratorer. Detta är en intressant hypotes, som i och för sig inte verkar vara alldeles otrolig. Det är ju fråga om att dela med sig av sin makt över företagets informationsbehandling. Ävenså kommer ett utspritt bruk av användarnära hjälpmedel att påverka dataavdelningens roll högst väsentligt (Se t ex Flensburg 1980b och Lie-Nielsen & Colliander 1983). Det kan vara intressant att se hur dataav-

delningarna reagerar när de ställs inför dessa mer eller mindre uppenbara "hot". Det är också viktigt att förmedla denna kunskap till andra dataavdelningar, som snart kan komma i en liknande situation. På så sätt kan man underlätta en smärtfri övergång från expertsystemering till användarsystemering. Frågorna nedan bör ställas till både användare och dataavdelning, givetvis med en del omformuleringar. Nedan är de formulerade i användartermer. En jämförelse med samma fråga ställd till dataavdelningen kan vara avslöjande.

K1. Uppmuntrar dataavdelningen bruket av applikationsgeneratorer? På vilket sätt?

Planerat

Här tror jag att dataavdelningen säger sig göra det, ty det är idag icke legitimt att låta bli ta tillvara användarnas intressen. Däremot tror jag användarna har en något anorlunda uppfattning, men säkert är det inte.

Resultat

Om man med dataavdelning menar Information Center är svaret obetingat "ja". Om man menar den ordinarie dataavdelningen är de mera passiva om det finns någon Information Center, men annars uppmuntrar de. Någon skillnad mellan användarnas svar och dataavdelningens svar har jag inte upptäckt.

K2. Hur ofta är folk från dataavdelningen på besök hos dig? Anledning?

Planerat

Täta besök kan tyda på att en eventuell uppmuntran inte bara är en läpparnas bekän- nelse utan även ett faktum, medan ringa antal besök i viss mån kan tyda på motsat- sen. Orsaken till täta besök måste naturligtvis klargöras, det kan ju röra sig om krånglande verktyg eller någon annan "legitim tjänsteangelägenhet".

Resultat

I samtliga fall kommer folk från dataavdelningen väldigt sällan på besök. Orsaken härtill kan vara att man inte har tid. Flera personer på Information Center klagade också på att de hade ont om tid. Undersökningar i USA (Leitheiser & Wheterbee 1984a) tyder på att användarna gärna vill att IC skall helt och hållet göra deras sy- stem. Något sådant har jag dock inte upptäckt här.

K3. Kan du själv bestämma vilken applikationsgenerator du vill använda? Även reellt?

Planerat

Vad jag är ute efter är hur stort inflytande användarna hade på valet av applikations- generator. Inte bara formellt, utan även reellt. Min tro är att de har ingen alls utan valet och utvärderingen står dataavdelningen helt för. En uppföljande fråga här kan

vara om det framfördes några skäl till valet och vilka effekter användaren tror det fått på företagets verksamhet.

Resultat

Alla användare utom fyra säger sig i princip kunna fritt välja. Två av dem kan bara använda en viss applikationsgenerator, medan två andra på grund av ett policybeslut fattat på högre nivå, också bara kan använda en. En av dem ville gärna ha en annan, men det gick inte. I praktiken väljer man dock bland de som finns på Information Center.

K4. Finns det några tecken på att dataavdelningen undanhåller viktig information angående applikationsgeneratorn?

Planerat

Den här frågan skall nog inte ställas utan den bör observeras. Om forskaren märker att användaren nyttjar redskapet på ett "konstigt" eller "klumpigt" sätt utan att använda funktioner som "uppenbarligen" borde finnas kan det beror på okunskap hos användaren eller ett dåligt verktyg. Om det är okunskap kan det vara intressant att veta om den är medveten eller inte. Det kan vara motiverat att inte lära ut ett visst kommando om dess bruk på något vis t ex äventyrar databasens säkerhet och konsistens. En följdfråga till dataavdelningen efteråt kan klargöra förhållandet. Problemet kan lätt bli ett forskningsetiskt problem, nämligen om forskaren upptäcker det önskvärda kommandot och lär ut det till användaren och det inte var meningen att han eller hon skulle kunna det.

Resultat

Fyra svar är avgivna och samtliga är "nej". Observationer verkar bekräfta.

K5. Hur stark är bindningen till dataavdelningen?

Resultat

Sex svar har avgivits och samtliga är "Inte särskilt stark".

Frågan är något oklart ställd eftersom "bindning" tolkas på många sätt. Istället kunde jag t ex ha frågat: "Kan du köpa en applikationsgenerator utan att gå via dataavdelningen? Om ja, skulle du kunna tänka dig göra det?"

K6. Hur har applikationsgeneratorn påverkat dataavdelningen?

Planerat

Jag trodde dataavdelningen kände sin traditionella position som enväldshärskare över företagets informationsbehandling hotad och att man därför försökte lite grand "hålla" på bruket av applikationsgeneratorn. I regel finns massor av arbete och kunskap nerlagda i den befintliga databehandlingen och jag trodde inte detta gick att av-

veckla över en natt. På sikt kan dataavdelningen möjligen i stort sett bli obehövlig. Många systemerare och programmerare är oroliga för detta.

Resultat

I två fall har folk på dataavdelningen själva börjat använda redskapen så smått. Man har gillat dem och uppmuntrar bruket. En uppenbar sak är att Information Center har inrättats. Någon större oro för att bli avvecklade tycks man inte hysa. Snarare för att få ännu mer arbete med att hjälpa användarna hålla reda på sin personliga databehandling.

4.2.13 L. Övrigt

Allmänna kommentarer

Detta är en samling frågor jag tycker verkar viktiga, men inte fått in någon annanstans.

L1. Vilka framtidsplaner finns?

Planerat

Om man hittar något generellt mönster i planeringen inför framtiden kan det vara ett kraftigt bidrag till en god teori för interaktiv systemutveckling.

Resultat

- Byte till bättre teknisk apparatur: 3 st
- Utveckling av samma utrustning: 2 st
- Helt nya applikationsgeneratorer: 3 st (varav 2 Information Center)
- Utvidgning av befintlig utrustning: 2 st (persondatorer)

Två IC befinner sig i ett generationsskifte av applikationsgeneratorer. De "gamla" ADI, ADRS etc understöds inte längre av leverantören utan nyare redskap har tagits fram. Ett av dem har senare valt SAS. Detta redskap har vidare valts av två andra IC.

L2. Finns det andra applikationsgeneratorer i organisationen?

Planerat

Man vill kanske gärna experimentera och provar flera olika applikationsgeneratorer. Orsaken till att man gör detta är intressant (om det finns någon överhuvudtaget). Det kan t ex tänkas att det är "dataavdelningens lekstuga" det handlar om. Eventuella jämförelser mellan olika applikationsgeneratorer kan vara intressanta att ta del av, likaså subjektiva tyckanden.

Resultat

På Information Center finns tillgång till en uppsättning verktyg för olika ändamål. Exempel på sådana är: SAS, ADRS, ADI, FOCUS, MAXICALC, Formel 1, Grafik, Ord-

behandling, APL, Basic, Easytrieve m fl. Men där man har en central dataavdelning finns det ofta på denna en mängd andra redskap "för test" som man brukar säga. Exempel på sådana är: Mercur, MIMS och RAMIS. De som testas är i regel andra generationens redskap, medan man ute i organisationen fortfarande använder första generationens. Vi torde stå inför ett snart generationsskifte. Berg & Matsson (1986) indikerar att detta generationsskifte äger rum omkring 1985-86.

L3. Har man funderat på att skaffa flera? Av samma sort eller någon annan?

Planerat

Detta är en uppföljning eller alternativ till föregående fråga.

Resultat

Som framgår av svaret på fråga L1 pågår ett generationsskifte. Främst vill man ha redskap som är mer integrerade och som både dataexperter och användare kan nyttja. Även vill man kunna köra mot produktionsregistren.

Det kan fö vara intressant att följa upp denna fråga med en om hur systemeringsmetoderna har förändrats med hänsyn till de nya redskapen.

L4. Hur tror du ditt arbete kommer att se ut om 5 år? Om tio år?

Resultat

Endast en användare hade någon klar uppfattning om detta. Men det beror på att hon stod inför ett tämligen omedelbart förestående byte.

Frågan borde nog istället formuleras ungefär så här: "Om du fick önska dig precis vad du ville, hur skulle du då i framtiden vilja använda applikationsgeneratoren?"

L5. Vad tror du dina chefer tycker om din användning av applikationsgeneratoren?

Resultat

Tre har svarat och samtliga tre chefer är helt ointresserade av datasystemet. Detta tyder på att datafrågor kan drivas separat utanför direktionnivån (för det var där cheferna fanns). Två andra användare var själva chef.

L6. Antag att du blir sjuk och måste ha vikarie. Antag också att en sådan finns, men vederbörande kan bara ditt arbete utan applikationsgenerator. Skulle denna vikarie kunna ta över direkt med hjälp av applikationsgeneratoren?

Planerat

Denna fråga är ett försök att testa hur pass beroende arbetet är av den tekniska handhavandekunskapen. Jag tror nämligen att många gånger gör en applikationsgenerator arbetet mer komplicerat istället för att förenkla det. Man kan gärna fråga detta direkt i samband med denna fråga. Det kan också tyda på någon form av revirtänkande hos användaren, något försök att göra sig själv oumbärlig.

Resultat

Nio stycken svarade. Sju stycken sa först "nej". Sedan var det två som ändrade sig och sa att det kanske skulle kunna gå att göra de allra enklaste rutinerna. Fem sa att det inte gick. Två sa att det inte gick beroende på att man måste kunna datorn medan de övriga tre sa att det berodde på att ingen kunde deras jobb som sådant. Två andra personer sa att det skulle gå att klara rutinjobben, men inte någon nyutveckling. Det förefaller alltså som om intervjupersonerna betraktar sig som tämligen outhärliga. Vidare är nog inte applikationsgeneratorerna så enkla som reklamen vill göra gällande.

L7. Vem har hand om servicen?

Resultat

I fem fall har man svarat och det är dataavdelningen (motsv) i samtliga fallen.

L8. Hur är applikationsgeneratorn jämförd med vanliga programspråk?

Resultat

Två personer har svarat och det framgår även av samtalet med någon av de andra att det är en tämligen meningslös fråga. I de fall jag studerat har det rört sig om att svara på frågor eller fylla i rutor. Det har inget med programmering att göra.

L9. Hur långa är väntetiderna?

Planerat

Väntetiderna är ett gammalt kärt tvisteämne och jag tyckte att jag måste ha med någon fråga dem också. Dessutom är det faktiskt en viktig faktor, för om man får vänta länge blir det väldigt irriterande.

Resultat

De är överlag långa. I två fall anges mycket långa (10 15 min!). Men detta sades bero på teleledningarna. Ett skäl att använda stordator kan vara att svarstiderna kan hållas nere. Ett annat skäl är den gemensamma databasen. I de flesta fall finner sig användarna snällt i att sitta och vänta. Men i ett fall var det en person som blev väldigt irriterad över att vänta i 2 sekunder medan kalkylen i Lotus 1-2-3 räknades om!

5. EMPIRISKT MATERIAL FRÅN DELTAGANDE

5.1 Bakgrund, metodik

Till grund för detta material ligger 9 stycken påbörjade experiment med deltagande i och initiering av användarutveckling. De flesta av dem pågår fortfarande och endast ett av dem är helt avslutat. Det har nämligen visat sig ta mycket längre kalendertid än jag från början trodde. Jag har i regel kommit in på ett mycket tidigt stadium och ofta har det då av rent organisatoriska skäl inte kunnat gå att få fram ett beslut om interaktiv systemutveckling. Företagets ledning har helt enkelt inte vetat om vilka fördelar man kan uppnå med detta arbetssätt och sålunda dragit sig för en investering i behövlig maskin- och programvara. Många av de deltagande organisationerna har också varit statliga eller kommunala förvaltningar, där sådana beslut tas långt från de blivande användarna. Jag har försökt påverka de enskilda slutanvändarna till att själva ta initiativ till utveckling av nya datasystem. Jag har agerat som förändringsagent och igångsättare, men när processen väl kommit igång har jag inte tagit någon större del i den utan endast på avstånd följt vad som hänt. Avsikten har ju varit att användarna själva ska utveckla sina datasystem. Omfattningen av min insats har varierat mycket, från lärare under sex lektioner till i stort sett ett halvt års heltidsarbete. Två av fallen finns rapporterade i Flensburg 1982 och 1984e.

En fråga som kan diskuteras i detta sammanhang och som bl a Kristen Nygaard tagit upp i sitt samarbete med Norsk Jern- og Metallfackförening (Nygaard, 1974) är hur resultatet av forskning för fackföreningen mäts. Traditionellt sker resultatrapportering genom publicerade forskningsrapporter, men Nygaard hävdar att detta mått är av föga värde i ett sådant sammanhang. Han menar istället att (ibid sid 196):

"resultater er alle handlinger fra NJMF, sentralt og lokalt som med støtte i prosjektet sikter mot å gi forbundet og dess medlemmer større innflytelse innen databehandling og styrning i bedriftene"

Detta sätt att mäta resultat är förenligt med de övergripande föreställningar om användarmedverkan som beskrivs i Sandström m fl (1983). Jag skulle vilja kalla detta för *det primära resultatet*. Om mina samarbetspartner i organisationerna själva tar hand om systemutvecklingen är detta ett mycket gott resultat av mina ansträngningar. Men forskningens resultat skall på något vis spridas till en större församling. Det primära resultatet kan då spridas genom fallstudier, som är beskrivningar av hur man gjorde där och då. Därigenom kan andra personer känna igen sin situation och kanske bli hjälpta att se alternativa handlingssätt.

Vidare anser jag att man ska redovisa så fullständigt och uppriktigt som möjligt. Fullständighet innebär att så mycket som möjligt skall redovisas. Även om ett förhållande skenbart inte tycks ha någon betydelse kan det i en framtida tolkning visa sig vara väsentligt. Men det finns ekonomiska ramar som sätter gränser för hur pass fullständig denna rapportering kan vara. T ex är det sällan möjligt att videofilma projektmöten, intervjuer, samtal och arbete. Vissa glimtar kan möjligen ges, men knappast mer. Flera forskare kan följas åt och någon observerar och dokumenterar utan att alltför aktivt delta i processen. En fördel med detta sätt är att det uppkommer flera samtidigt tolkningar av händelseförloppet. Men det kostar också pengar och det är inte möjligt annat än vid vissa tillfällen. I regel är man utlämnad åt sig själv. Ett annat alternativ är att någon av deltagarna från företaget samtidigt för anteckningar och dokumenterar förloppet. Men även detta stället sig svårt, ty dessa personer har ju ett annat arbete vid sidan om och befinner sig ofta under hård tidspress. Användarutveckling uppkommer ju ofta på grund av något problem i verksamheten, vilket gör att den inte flyter så smidigt som den borde göra. Nygaard löste dilemmat genom att med grund i sina erfarenheter i NJMF-projektet och delvis tillsammans med fackliga förtroendemän som deltagit i detta, skriva en lärobok i avsikt att användas av fackliga studieförbund. De fackliga förtroendemännen skrev också en del egna rapporter. I detta projekt har jag inte kommit så långt, men jag har dokumenterat mina erfarenheter och sammanställer dem här. Eftersom jag har ett större antal fall än vad Nygaard hade, har jag viss möjlighet att göra jämförelser och hitta embryon till mönster. Detta kan dokumenteras i traditionella forskningsrapporter och blir då ett *sekundärt resultat*.

5.2 Sammanfattning av resultat

Jag har funnit att en kurs med en ungefärlig inriktning: "Att ställa krav på datasystem" är *ett* bra sätt att starta en användarutveckling. Kursen startar med en allmän och översiktlig genomgång av vad datorer är för typ av fenomen och övergår gradvis i ett interaktivt utvecklande av datasystem. I fyra av de nio fallen har jag inlett på detta vis. Gemensamt i nästan alla fallen har varit problem att skaffa utrustning. Därför har experimenten i regel fått avbrytas när vi konstruerat en prototypdatamodell. Denna modell har först gjorts på papper men sedan av mig implementerats på någon av de datorer jag haft tillgång till. Det hade varit en stor fördel om organisationen haft tillgång till maskiner som hade kunnat användas för kursen men även driftsmässigt i fortsättningen.

Bortsett från utrustningsproblem har det fungerat mycket bra. Det har tagit cirka 24 lektionstimmar att genomföra en sådan kurs. Därav ligger ungefär 12 timmar på en grundkurs i databehandling. Kan motsvarande kunskaper förutsättas borde tiden kunna minskas. Dock tror jag inte man ska minska för mycket på tiden, eftersom det behövs en viss tid för läraren och eleverna att "bli bekanta" med varandra och veta

var man står någonstans. Den inledande grundkursen är en lämplig programförklaring.

Någon speciell systemeringsteknik har jag inte använt. I något fall ritade vi upp en väggraf över organisationens nuvarande verksamhet, men för övrigt var det anteckningar på blädderblocksblad. Men jag har sökt uppnå en stomme till en databas i tredje normalformen.

5.3 Bibliotekssystemet

Detta projekt började 1982 med att både länsbiblioteket i Malmö och stadsbiblioteket i Lund ville ha hjälp av Lunds universitet med att utreda datorstöd åt lokalsamlingsystemen och skånebibliografin. Lokalsamlingar finns på de flesta stadsbibliotek och är en samling artiklar, tillfällighetstryck, böcker, notiser m m som berör den aktuella orten. Materialet insamlas utan de kvalitetskrav som ställs på övrigt biblioteksmaterial och utgör främst ett forskningsmaterial för de som vill veta mer om hembygden. Lokalsamlingar finns både i Lund och i Malmö. Dessutom ger länsbiblioteket varje år ut en sammanställning, skånebibliografin, av all skånelitteratur som utgivits under året.

Man undrade nu från bibliotekens sida om det var möjligt att använda datorer för att lättare hitta i lokalsamlingarna och för att underlätta sammanställningen av Skånebibliografin. Eftersom jag vid denna tid sökte efter ett "riktigt" praktikfall för en kurs på systemvetenskapliga linjens sista år, kom denna förfrågan för min del i alldeles rätt tid. Det bestämdes att mina 20 elever skulle göra två stycken prototyper till söksystem, ett för lundasamlingen och ett för skånebibliografin. Dessutom var man också från bibliotekets sida intresserad av att få någon enkel kurs för att därigenom kunna ställa bättre krav på sin egen databehandling.

Prototyperna och arbetet med att ta fram dem finns dokumenterade i Axelsson m fl (1982) och Bertilsson m fl (1982). Denna dokumentation gjordes av eleverna och är, förutom en beskrivning av själva systemet också en beskrivning av processen med att ta fram dem. Processen var långtifrån smärtfri, det förekom missförstånd (jag krävde något helt annat av eleverna än vad de trodde), tekniska problem (till sist blev det 20 elever på 1 (en!) mikrodator), hård tidspress (bl a schemaläggning 24 tim/dygn!) och mycket mera, men trots allt upplevde både jag, eleverna och bibliotekspersonalen äventyret som mycket lärorikt och lyckat.

Efter elevernas prototyper höll jag en grundkurs för bibliotekspersonal från hela Malmöhus län. Kursen kallade jag "Att ställa krav på datasystem" och den blev upptakten till den kurs jag numera använder som introduktion till interaktiv systemutveckling. Kursen tog upp följande moment:

- Datorn som fenomen
- Datasystems effekter på arbete, samhälle och organisationer

- Genomgång av hur en dator fungerar
- Introduktion till programmering

Jag inledde med att visa några bilder ur gestaltpsykologin:

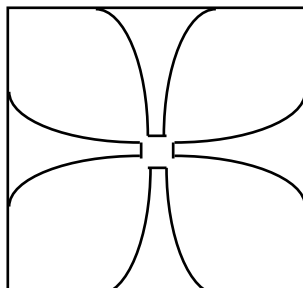


Fig 5.1 Introduktionsbild

Denna bild slog genast an rätt ton. Det förekom gissningar på blommor, fyrklövrar etc men "rätta" svaret är "Fyra elefanter som dricker vatten ur en fyrkantig vattenho" (Fig 5.2). Bilden har jag hämtat ur Asplund (1971).

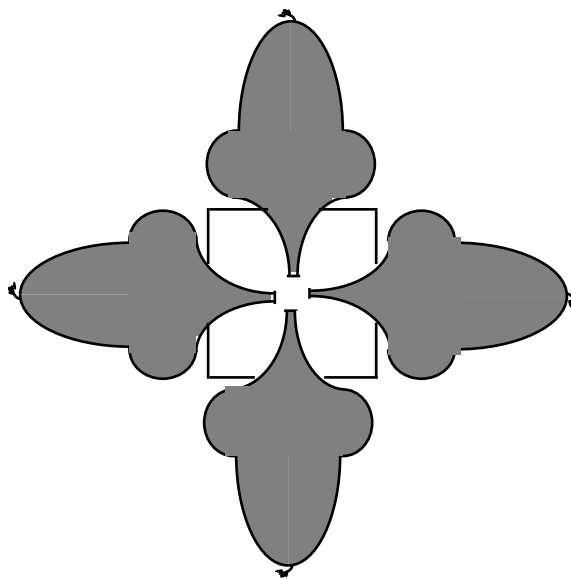


Fig 5.2 Den andra introduktionsbilden

Avsikten med dessa bilder var dels att få en lättsam introduktion till kursen, att så att säga slå an rätt ton, men också ge en metafor till vad datasystem innebär. Sambandet mellan elefantbilderna och datasystem är nämligen att datasystem bara beskriver en *begränsad* del av en verklighet. Den begränsning som valts ut är en *säregen* begränsning, vilket medför att de personer som sedan ska använda datasystemet inte känner igen den bild av verksamheten som detta förmedlar. Även om varje del i sig är riktig, är det ändå en "konstigt" utvald del. Den ovanliga begränsningen gör att helhetsuppfattningen blir främmande. Därefter fortsatte jag med bild 5.3.

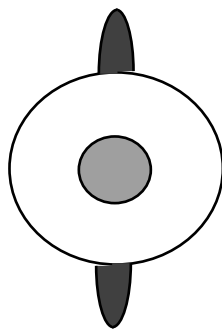


Fig 5.3 Den tredje introduktionsbilden

Bilden föreställer en mexikan, som cyklar. Med den vill jag visa att datasystem framställer en verksamhet från ett visst perspektiv. Perspektivet beror på syftet med beskrivningen. Om syftet är att beskriva hur stor markyta mexikanaren upptar är bilden ovan alldeles utomordentlig. Men om syftet istället är en undersökning av hur många cyklande mexikanare som har randiga skjortor är det inte en bra bild. Då kanske nedanstående bild är bättre:

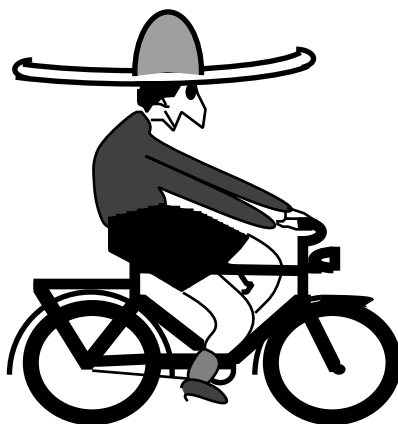


Fig 5.4 Alternativ mexikan

Men det är ingen principiell skillnad mellan dessa båda bilder vad avser perspektivet. Båda ger en beskrivning från ett visst perspektiv. Möjligen kan man säga att det andra perspektivet är något mera välbekant. Vi får här ytterligare en aspekt på hur datasystem kan förfrämliga vardagliga och välkända företeelser.

Som en sista inledning visade jag bilden med de två gubbarna:

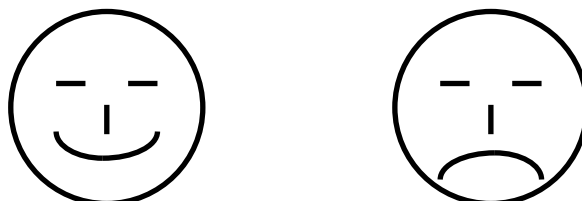


Fig 5.5 Den sista introduktionsbilden

Det som ska noteras är skillnaden i uttryck mellan de båda gubbarnas ögon. Ögonen är exakt lika (i själva verket har jag en overhead-bild där munnen är lös och jag bara vänder den upp och ner), men ändå ser de så olika ut. Det jag illustrerar med denna bild är sammanhangets betydelse. Om vi ändrar på ett ställe kan det få konsekvenser på helt andra ställen. Man hör ofta datafolk säga att "systemet har inte haft några som helst konsekvenser användarna gör samma saker som tidigare!". Detta visar bara på okunnighet om arbetsorganisationens betydelse. För övrigt är det meningen att det ska bli effekter när ett datasystem införs!

Bilderna av gubben är också en illustration till begreppet "tyst kunskap". Något förändras, men man kan inte explicit säga hur. Genom att kursdeltagarna på detta vis blir medvetandegjorda om den tysta kunskapen är det lättare för dem att finna exempel på denna även i sitt eget arbete.

Att inleda med dessa bilderna visade sig vara ett lyckat grepp och jag har sedan dess konsekvent använt dem. Att bilderna går hem så pass bra kan dock möjligen beror på att jag i huvudsak har visat dem för personer med akademisk bakgrund inom humanistiska ämnen.

Jag tyckte själv att det där med hur en dator fungerar var mindre viktigt och inte alls behövdes tas upp. Min motivering var att man kan mycket väl köra bil utan att veta hur en bil fungerar. På samma sätt, menade jag, skulle man kunna använda datorer utan att veta hur de fungerade. Men kursdeltagarna insisterade på att de ville få denna kunskap. Och när jag då till sist tog in en dator i klassrummet och skruvade sönder den, var lyckan fullkomlig. "Detta har vi väntat på hela kursen!" var den spontana kommentaren från en av deltagarna. Förmodligen är det så att datorn upplevs som ett främmande fenomen för många av oss och det är då tryggare att på ett ungefär veta hur den fungerar. De flesta som kör bil vet ju också ungefär hur bilen fungerar.

Det är också intressant att notera att de flesta av deltagarna var humanister. Man trodde från början att det skulle krävas massor av kunskap i matematik för att förstå hur datorer fungerade. Men jag förklarade mig vara enbart glad åt att de inte hade den belastningen också! För att illustrera hur datorer och då speciellt program fungerar hade vi en övning där deltagarna delades in i två grupper och var och en skulle skriva instruktioner till en robot som skulle flytta en på bordet placerad stol till motsatta hörnet av rummet. Det fanns bara en högst begränsad instruktionsrepertoar, men problemet var möjligt att lösa. Dessutom hade vi alla väldigt roligt när deltagarna stegade omkring och bar stolar i lokalen. Tipset till denna övning hade jag för övrigt fått från Olov Forsgren från informationsbehandling-ADB i Umeå.

Kursen blev mycket väl mottagen och jag och personalen på Malmö stadsbibliotek/Länsbiblioteket bestämde oss för att försöka fortsätta samarbetet genom att gemensamt utveckla ett datasystem för Skånesamlingen och Skånebibliografin på Länsbiblioteket i Malmö. Ansökan om utvecklingsbidrag ingavs, men beviljades inte.

Dock fortsatte bibliotekspersonalen själva att söka efter möjligheter och förverkligade så småningom det önskade systemet "LIMES" (Mundt-Almquist m fl, 1985). Detta är det första och hittills enda projekt som nått sin fullbordan. Det blev lyckat därför att efter det att ansökan avslagits, genomförde trots det användarna projektet helt utan min medverkan. Jag var med och startade upp processen, men de övriga deltagarna fullbordade! Detta är enligt min mening tecken på en mycket lyckad användarutveckling.

5.4 Kommunal förvaltning

Detta projekt började med att en elev till en av mina kolleger (Siv Friis) tog kontakt med henne angående en utredning man hade gjort inom en kommunal förvaltning om en eventuell datorisering av lönebehandlingen. Man hade på personalavdelningen gjort en förberedande utredning och ville nu ha hjälp från högskolan med att genomföra en inledande utbildning. Siv ville att jag också skulle delta i projektet eftersom jag hade större erfarenhet än hon av sådan utbildning.

Den idé jag och Siv förde fram och också fick gehör för var att rikta utbildningen i huvudsak mot systemutveckling, hur den bedrivs och vilka konsekvenser den kan få. För att visa hur det kunde gå till skulle vi gå genom ett exempel från kursdeltagarnas egen verksamhet. Möjligheten att använda en applikationsgenerator för prototyper och/eller färdiga system var man också intresserad av. Kursplanen ovan presenterades och antogs. All personal på löneavdelningen (c:a 25 personer) skulle utbildas och under arbetstiden. Därför fick utbildningen ske i två grupper, en på förmiddagen och en på eftermiddagen, eftersom det inte gick att "stänga" löneavdelningen helt och hållet tre timmar i veckan.

Kursen gavs våren 1984 och sköttes växelvis av mig och Siv Friis. Vi gav först en kort och allmän överblick över databehandling i allmänhet, ungefär som det beskrivits ovan. Sedan tog vi upp följande moment, i anslutning till av personalen angivna problem:

- Nulägeskartläggning
- Problemanalys
- Informationsbehov
- Datamodellering

När vi skulle exemplifiera med ett av deras problem fanns vi båda på plats och en av oss observerade och gjorde anteckningar. Avsikten var att snabbt och ytligt gå genom de inledande etapperna i SIS/RAS-modellen för att ge användarna någon liten aning om hur det färdiga systemet skulle kunna te sig.

Eftersom det var två grupper fanns det goda möjligheter att göra jämförelser mellan deras skilda problem- och behovsuppfattningar. Följande är en beskrivning av ett experiment jag gjorde i syfte att kunna göra sådana jämförelser.

5.4.1 Experiment med datamodeller

Syftet med detta experiment var att jämföra två datamodeller som gjordes av två skilda grupper. Experimentet gjordes som avslutning på kursen. All personal på avdelningen deltog i kursen, men var uppdelad på två grupper om c:a 13 personer i varje. Innan datamodellen gjordes hade några personer i grupperna gjort en slags kravspecifikation till ett eventuellt lönesystem.

Själva experimentet var en "tillämpad föreläsning" kring relationsdatamodeller och normalisering. Introduktion och övrig presentation från min sida var i stort lika i båda grupperna. På förmiddagen deltog Siv Friis som observatör. Gruppstorleken var något ojämn med c:a 14 personer på förmiddagen och endast 6 st på eftermiddagen.

Relationsdatamodellen introducerades såsom en uppsättning informationsobjekt och egenskaper hos dessa objekt. Kursdeltagarna ombads ange informationsobjekt de behövde i sin verksamhet. Följande objekt angavs:

<u>Förmiddag</u>	<u>Eftermiddag</u>
Avtal	Person
Person	Tjänstgöringsrapport
Arbetsgivare	Löner
Tjänst	Avtal
Arbetsplats	
Avdelningsenhet	

Den ordning i vilken objekten anges ovan var den ordning i vilken de föreslogs. Det förekom även viss diskussion om deras innebörd, både från min sida och framförallt och mest livligt från kursdeltagarnas sida. En viss styrning i syfte att uppnå ökad precision och förtydligande förekom från min sida.

Som dataexpert tror man ofta att de begrepp och kategorier som används i en verksamhet är alldeles kristallklara och gemensamma för de som arbetar i denna verksamhet. Men så är inte fallet, vilket detta exempel visar. Sandström (1985) exemplifierar med andra erfarenheter, främst från sjukvård.

Efter det att informationsobjekten var föreslagna fick grupperna för vart och ett av objekten ange vilka egenskaper hos detta objekt de ansåg sig behöva känna till i sin verksamhet. Diskussionen blev här än livligare mellan kursdeltagarna. Man kan nog lungt fastslå att väldigt många termer används på olika sätt av de olika enheterna. I båda grupperna var det mindre deltagargrupper som dominerade, men av naturliga skäl (6 deltagare) var sådana grupper mindre dominerande i eftermiddagsgruppen.

Här anges egenskaperna för varje informationsobjekt såsom de först föreslogs:

Förmiddag

Avtal

Här ansåg man sig inte kunna
ange några speciella egenskaper

Person

Namn
Adress
Telefon
Utbildning
Anställningsdatum
Lönegrad
Tjänst
Syssetsättningsgrad
Skatt
Avdelningsenhet
Sparad semester
Restskatt
Facktillhörighet
Gruppliv
Tjänstledighet
Årets semester
Ålder

Arbetsgivare

Förvaltningsnamn
Avdelningsnamn (flera inom varje förvaltning)
Avdelningsenhet (flera inom varje avdelning)

Tjänst

Omfattning
Avdelningsenhet
Lönegrad
Konstruktion

Eftermiddag

Avtal

Avtalsnamn
Giltighetsdatum

Person

Person-id
Adress
Arbetsställe
Bankkonto
Befattning
Löneklass
Tjänstetid
Skatteuppgift
Telefon
Utbildning
Sjukdygn
Semester
Anställningsdatum
Anställningsgrupp
Närvaro
Övrig tjänstledighet
Övrig frånvaro full lön
Fackuppdrag
(Tjänst)
Uppdragsarvode
Omkostnadsersättning
Avtalstillhörighet
Ålder

Tjänstgöringsrapport

Namn
Arbetsställe
Befattning
Tidpunkt
Frånvarotid
Frånvaroorosak
Närvarotid
Övertid
Mertid
Övertidsuttagstyp
Arbetsledare
Attest (namnteckning)

Innehavare
Avtal

Begreppet "arbetsställe" i eftermiddagsgruppen var detsamma som "avdelningsenhet" i förmiddagsgruppen. Detta gavs som svar på en av mig direkt ställd fråga. Lika så angav eftermiddagsgruppen på en direkt fråga från mig att "tjänstetid" var detsamma som "sysselsättningsgrad". Båda grupperna föreslog personnummer istället för namn, men jag envisades med att ha namn istället. Detta för att behovet av "ålder" skulle komma fram uttryckligen. Förmiddagsgruppen accepterade utan protester medan eftermiddagsgruppen var mer påstridig. Behovet av "ålder" visade sig inte förrän efter ett bra tag. Därför står det också sist i båda grupperna ty termerna är angivna i den ordning de föreslogs.

Efter denna uppställning gick jag genom första och tredje normalformen. Som exempel på normalisering till första normalformen anfördes "arbetsgivare" enligt förmiddagsgruppens beskrivning. Från början såg tabellen ut så här:

Arbetsgivare: Förvaltningsnamn, Avdelningsnamn, Avdelningsenhet

Det finns flera avdelningar inom varje förvaltning och flera enheter inom varje avdelning. Normaliseringen till första normalformen visades genom att tabellerna "avdelningsenhet" och "avdelning" konstruerades:

Avdelning: Avdelningsnamn, Förvaltningsnamn

Avdelningsenhet: Avdelningsenhetensnamn, Avdelningsnamn, Förvaltningsnamn, Arbetsledare

Detta upptäcktes naturligt i förmiddagsgruppen, men på grund av tidsbrist bara visade jag hur man skulle göra i eftermiddagsgruppen.

Sedan introducerade jag begreppen "primär nyckel" och "funktionellt beroende". Ett sådant föreligger mellan två eller flera egenskaper om en eller flera av dem alltid bestämmer vilka de andra är. Vi gick genom personalobjektet term för term och kursdeltagarna angav själva vilka beroenden som fanns. Självt förklarade jag att "jag vet inget, det är ert jobb, det är ni som är experter!" Man lyckades också under en halvtimme göra denna normalisering. Nedan anger jag vilka termer de olika grupperna ansåg höra hop. Personobjektet anges i tredje normalformen, med primärnyckeln understruken. De övriga tabellerna anges, men utan objektnamn. Motsvarande termer står nu mitt för varandra, för att klarare markera skillnader i terminologi. Om termen inte har någon motsvarighet är positionen i den andra gruppen tom.

<i>Person</i>	<i>Person</i>
<u>Namn</u>	<u>Namn</u>
Adress	Adress
Sysselsättningsgrad	Tjänstetid
Skatt	Skatteuppgift
Sparad semester	

Restskatt
Facktillhörighet
Gruppliv
Tjänstledighet

Årets semester
Ålder

Fackuppdrag

Övrig tjänstledighet
Övrig frånvaro full lön

Ålder
Utbildning
Sjukdygn
Närvaro
Uppdragsarvode

I förmiddagsgruppen utökades informationsobjektet "tjänst" med en del termer, i eftermiddagsgruppen skapades ett nytt informationsobjekt "befattning", som resultat av konstaterade beroenden.

Tjänst

Omfattning
Avdelningsenhet
Lönegrad
Avtal
Konstruktion

Anställningsdatum
Utbildning

Befattning

Benämning
Omfattning
Arbetsställe
Lönegrad
Avtal

Anställningsgrupp
Omkostnadsersättning

Observera att termen "utbildning" tolkas olika av de olika grupperna. På förmiddagen tolkades den som "För tjänsten behövlig utbildning" och på eftermiddagen tolkades den som "den aktuella personens utbildning". Eftermiddagsgruppen gjorde också en utökning i "Avtal". Denna tillkom genom att deltagarna konstaterade att ett antal termer hängde ihop. Först när allt var klart kom man på att dessa termer borde höra hemma under informationsobjektet "Avtal":

Avtal
Avtalsnamn
Giltighetsdatum
Löneklass
Ålder
Tjänsteår
Antal semesterdagar
Belopp

Tjänstear och ålder bestämmer här vilken löneklass den anställde ska ha. Att lägga märke till är att först här och först på initiativ från mig kommer det in begrepp som direkt har med pengar att göra, nämligen "belopp". Förmodligen var det alldeles uppenbart för deltagarna att "Avtal" även innefattar "belopp", men det blev inte sagt förrän jag påpekade det.

Tiden medgav inte fler övningar, men det är min förmodan att jag efter en sådan här kurs under 1 à 2 dagar tillsammans med kursdeltagarna kan åstadkomma en realiserbar relationsdatabas i tredje normalformen.

5.4.2 Vad som hände sedan

Detta kan sägas väldigt kort: Inget! Det skulle fattas ett politiskt beslut om utvecklingen, men ett sådant beslut dröjde. Annars fanns det förutsättningar både att göra prototyper och att engagera användarna. Man hade en egen dataavdelning på förvaltningen och datachefen var med på några lektioner. Han gav intryck till att vara positiv till en fortsättning. Men jag har inte forskat vidare i varför det inte blev någon fortsättning. Utbildningen och hela processen tjänar ju som hjälp till självhjälp och jag tycker därför inte att jag ska blanda mig i alltför mycket. Det verkar i varje fall inte ha varit kursen som sådan som har gett dessa effekterna utan det är snarare politiska skäl som satt käppar i hjulet.

5.5 AMI-projektet

Jag har sedan 1980 haft sporadiska kontakter med länsarbetsnämnden i Malmö och arbetsförmedlingen i Lund. Det har varit konferenser, föredrag och informationsdagar både med ledning och via fackförbunden. På hösten 1983 höll jag en informationsdag för AMI i Malmöhus län. Deltagarna där ville absolut att jag skulle följa upp informationsdagen med att hålla en kurs för dem. Deras problem var att skicka personer på olika datakurser utan att riktigt veta vad kurserna innehöll och vad som krävdes för att gå på dem eller vad de ledde till för yrken. Jag hjälpte dem att något reda ut begreppen, såsom jag uppfattade det och det gav tydligen mersmak. Vi bestämde oss sålunda för att på hösten 1984 hålla en kurs i "att ställa krav på datasystem" för personal vid AMI och AF i Lund. Kursen hölls en dag i veckan direkt efter arbetets slut.

Även här inledde jag med bilderna på elefanter och mexikaner. Förväntningarna var från början högt ställda och mina okonventionella bilder lockade ytterligare några. Men kanske blev en del skrämde av dem. Kursen i övrigt följde samma mönster som tidigare kurser, så när som på att vi lade ner lite mer möda på att göra en problemdefinition och skissa på hur det skulle se ut i framtiden. Vi följde vad jag kallar "Systemeringsmodell för användare":

- Verksamhetsanalys
 - Vad är verksamheten?
 - Vad är målen?
 - Hur ser det ut nu?
 - Hur vill vi det ska se ut?
- Informationsanalys
 - Informationsbehov?
 - Vilka informationsobjekt finns?
 - Vad behöver vi veta om varje informationsobjekt?
 - Vad betyder informationen?
 - Vad måste vi absolut veta?
- Datamodellering
 - Hur ser våra tabeller ut?
 - Eliminera "subtabeller"!
 - Primärnycklar?
 - Hitta beroenden! Eliminera!
- Konstruktion
 - "Programmera"
 - Testa! Gå tillbaka och ändra!

Tillvägagångssättet påminner, åtminstone i sina inledande skeden om SIS/RAS och andra traditionella systemutvecklingsmodeller. Det tror jag beror på att det bakom både dessa och den här ligger en allmän problemlösningsmetodik.

På kursen valde vi antagningen till AMU-kurser som övningsexempel. Flertalet av deltagarna upplevde det som ett av sina större problem i det dagliga arbetet. Extra besvärligt var det då AMU-kurserna ofta var översökta och man fick lämna återbud, men trots det gick kurserna halvfulla eller i värsta fall inte alls på grund av för få deltagare! Det blev en lång lista av önskemål. Men i början var man mycket försiktig i sitt önskan. Jag var tvungen att uppmuntra ordentligt för att få kursdeltagarna att våga komma med alla sina önskningar. Vissa saker man tyckte var enkla var inte alls särskilt lätta att lösa rent tekniskt medan andra saker, som man bara kunde drömma om i sin vildaste fantasi ibland kunde gå att realisera med relativt enkla medel.

Vi gick relativt snabbt över på informationsbehovet. Precis som i fallet med den kommunala förvaltningen blev det diskussioner kring vad de olika begreppen betydde. Speciellt besvärligt blev det då man talade om "kurser" och jag med förflutet som lärare ansåg mig ha en klar uppfattning om vad "kurser" var för något. Denna villfarelse blev jag snabbt tagen ur och ett tag kände jag mig väldigt förvillad och desori-

enterad eftersom jag inte kunde begripa hur man t ex kan ha kontinuerligt intag till en kurs. Men jag kom på mig med detta och sade till kursdeltagarna att jag ju stod och betedde mig som en vanlig dataexpert och försökte sälja färdiga lösningar istället för att kartlägga problem!

Själva datamodellen blev i och för sig ganska omfattande, men var i praktiken ett enda stort register. En del av kursdeltagarna var mycket aktiva medan andra var mera avvaktande. Sista gången vi träffades hade jag med mig en dator där jag i DBase II hade gjort en liten prototyp till det färdiga registret. Meningen var sedan att vi skulle söka medel hos AMS för att implementera och vidareutveckla skissen. Men detta fick vi överge av två skäl.

För det första måste registret omfatta hela landet och därför ligga på AMS' centrala dator. På denna fanns inte någon applikationsgenerator så det var rent tekniskt inte möjligt att genomföra den delen. För det andra visade det sig att någon annan höll på att göra ungefär samma system och det skulle spridas ut till kontoren om någon månad. Vi bedömde därför utsikterna att få genom idén som mycket små och gjorde inget mer åt det.

Men på AMI tyckte man att vi ändå skulle försöka genomföra något. Min kontaktperson där erinrade sig ett projekt där man skulle hålla reda på praktikplatser hos olika arbetsgivare. Detta fungerade nu med hjälp av trio-pärmar, men man bedömde att en datorisering skulle effektivisera utnyttjandet mycket. Ansökan om medel till en portabel mikrodator har ingavs till AMS men inte heller denna gång fick man något stöd. Projektet vilar för närvarande.

5.6 Övriga organisationerna

Eftersom det visat sig besvärligt med maskiner, har jag i de senare experimenten arbetat på att först få företagen att skaffa maskiner, därefter genomföra kursen. Det finns nu två privata företag som är intresserade av att ställa upp på detta. Ett av dem har ingen tidigare datorisering och där har man köpt en mindre mikrodator med standardsystem för bokföring. Meningen är att användarna sedan med hjälp av DBase II själva ska utveckla lagerhantering, fakturering och eventuellt viss produktionsplanering. Man var tvungen att komma igång väldigt snabbt med bokföringen eftersom den befintliga, mekaniska bokföringsmaskinen var alldeles uttjänt. Företaget var berett att mer eller mindre anse detta första system som läropengar och efter något år satsa på ett helt nytt. Man tecknade ett förmånligt avtal beträffande standardsystemet och kom igång ganska bra. Men man har nästan ingen tid att satsa på utveckling och utbildning.

Det andra företaget har redan en mikrodator, men man är inte nöjd med de system man där har. Det finns dock, enligt min bedömning, möjligheter att på ett betydligt

bättre sätt utnyttja den programvara som finns. T ex är de olika rutinerna inte kompatibla med varandra. Man är tvungen manuellt mata in siffrorna från den ena rutinen i den andra istället för att hämta dem från befintliga register. Men arbetet med att studera detta har precis börjat.

De nästa två företagen är båda föreningar. Ett av dem är en lokal krets av ett fackförbund med c:a 70 000 medlemmar. Den andra föreningen har c:a 2000 medlemmar men är primärt intresserad av att kunna hålla ordning på de konferenser man anordnar. På dessa kommer upp till 800 personer, med skiftande önskemål om inlogering, betalning, kringaktiviteter m fl.

Ifråga om fackförbundet är det två skilda nivåer av aktiviteter. Man tycker att kostnaderna för att hantera det centrala medlemsregistret har blivit alldeles för höga och söker efter sätt att minska dem. Samtidigt tycker man ute i kretsarna att man kan inte utnyttja registret speciellt mycket och vill gärna få ökat inflytande över förbundets databehandling. I denna fråga har jag vid några möten diskuterat vad som skulle kunna göras och har även skissat på några tänkbara förslag till lösning och vad de kan kosta. På den centrala kongressen har man, delvis tack vare aktiviteterna från denna krets, fattat ett principbeslut i den önskade riktningen.

Den andra nivån gäller stöd åt de rent lokala aktiviteterna på krets nivå. Vi har inte diskuterat så mycket vad detta skulle kunna vara, men det gäller främst att hålla reda på förtroendeuppdrag och vem som har dem.

Den andra föreningens huvudsakliga uppgifter är att anordna konferenser och utge en periodisk publikation. De huvudsakliga intäkterna kommer från ett annat företag, till vilket man står i ett invecklat ägare/beroendeförhållande. Man tänkte sig främst använda dator för att kunna hålla reda på konferensdeltagarna och deras önskemål. Man har lånat en Macintosh och funnit den vara synnerligen användbar. Man ansökte om medel hos sin huvudman om att få införskaffa en sådan dator och dessa medel beviljades. Jag har åtagit mig viss introducerande utbildning men sedan är det meningen att personalen själva ska hitta lämpliga användningsområden.

Nästa organisation är en avdelning på Lunds lasarett. Två av läkarna där var intresserade att använda datasystem i sitt arbete/forskning som hjälp att göra vissa utsökningar i centrala register. Man hade också behov av att kunna följa upp medicineringen för olika patienter. För att undvika personalkonflikter ville man gärna ha med sig den övriga avdelningen. Därför nappade man på ett erbjudande om att två forskare skulle hålla en introduktionskurs i databehandling. Man ställde också upp på en förutsättningslös inventering av eventuella problem på avdelningen. Inventeringen gjordes och kursen hölls, men det visade sig att några av personalen kände sig mer eller mindre kommenderade att gå på kursen, som hölls strax efter ordinarie arbetets slut. Men kursen var helt frivillig.

Kursen följde samma mall som tidigare. Vi valde att skissa på ett förrådssystem för avdelningen som exempel på systemutveckling. Detta beslut fattades ganska slump-

mässigt och vi hade inte alls diskuterat förutsättningar eller tänkbara konsekvenser i förväg. Det var från början helt klart att det skulle aldrig gå att införa något sådant system, utan det hela fick ses som ett pedagogiskt experiment. Konstruktionen av datamodellen gick bra och en skiss gjordes av personalen själv med hjälp av ett registerhanteringssystem på en Macintosh. Detta tog längre tid än jag hade trott, främst för att jag överskattade redskapets och datorns lätthanterlighet. Det är inte möjligt för en total novis att på två timmar lägga upp ett register, inte ens på en Macintosh!

5.7 SOKRATES på Företaget

Den sista studien är relativt omfattande och djupgående och har bedrivits under lång tid tillsammans med ett större, divisionaliserat industriföretag. Studien startade genom personliga kontakter med ekonomi- och datapersonal på Företaget. De var båda intresserade av samarbete med universitetet och av att bedriva användarnära systemutveckling i enlighet med de riktlinjer som fanns redovisade i Sandström m fl (1983). En lägesrapport finns publicerad (Flensburg 1984e) och det mesta av materialet i detta avsnitt kommer från denna.

5.7.1 Företaget

Företaget grundades i slutet av 1700-talet men har anor ända från 1500-talet. Det är stort (2 500 anställda), är ett tillverkningsföretag och världsledande inom sitt område. Det är uppbyggt som en koncern med centrala staber, fristående divisioner och dotterbolag, både i och utanför Sverige. En av divisionerna står för ungefär hälften av omsättningen och större delen av vinsten. En annan är ganska liten men man sysslar där med för Företaget nya områden. En är lokaliserad till annan ort än Företaget och består i det närmaste av ett dotterbolag. Det har tidigare varit ett problembarn, men börjar nu återigen ge vinst.

Företaget är dominerande på orten och har under tidernas lopp haft stort inflytande på dess utveckling. Under några år i slutet av 70-talet och början av 80-talet blev läget mera ogynnsamt för Företaget och kraftiga omorganisationer och nedskärningar ägde rum. Under 80-talet har antalet anställda minskat med ungefär 600 st. Detta gör att stämningen på Företaget vid projektets start var en smula "avvaktande". Organisationen var inte heller helt inkörd i de nya förhållandena och en del personer, främst tjänstemän, kände sig osäkra på sin fortsatta anställning. Under andra halvåret 1983 inträdde emellertid en markant förbättring för Företaget och en stor del av osäkerheten har försvunnit.

5.7.2 Företagets ekonomisystem

Det aktuella projektet berör Företagets ekonomisystem. Målet är att förbättra dess tillgänglighet. SOKRATES-projektet ska delta i denna process och dokumentera de erfarenheter som vunnits och med hjälp av dem utveckla mer generell kunskap kring användarutveckling t ex i form av någon "systemutvecklingsmodell för interaktiv systemutveckling".

För närvarande är Företagets ekonomisystem satsvis orienterat och framställer ett antal listor varje månad. Det finns emellertid önskemål om en flexiblere rapportering och möjlighet att kunna få specialrapporter med kort varsel.

Dessa önskemål kommer både från den centrala ekonomiavdelningen och från de olika divisionerna. Fig 5.6 ger en översiktlig beskrivning av det nuvarande ekonomisystemet och de system som levererar indata till det, de s k försystemen. De enskilda divisionerna börjar nu ställa större och egna krav på ekonomisystemet. Vissa försök att anpassa systemet till de olika divisionernas önskemål har gjorts. Tre av divisionerna har likartad typ av verksamhet varför deras rapporteringsbehov också är likartat, men en division har helt annorlunda behov.

Inrapporteringen till ekonomisystemet sker från ett antal omkringliggande system. De flesta körs på lokala mini- eller mikrodatorer. Till exempel finns order- och faktureringsystem på minidatorer på divisionerna. Rapporteringen till det centrala systemet sker genom disketter och band. Utdata består av listor.

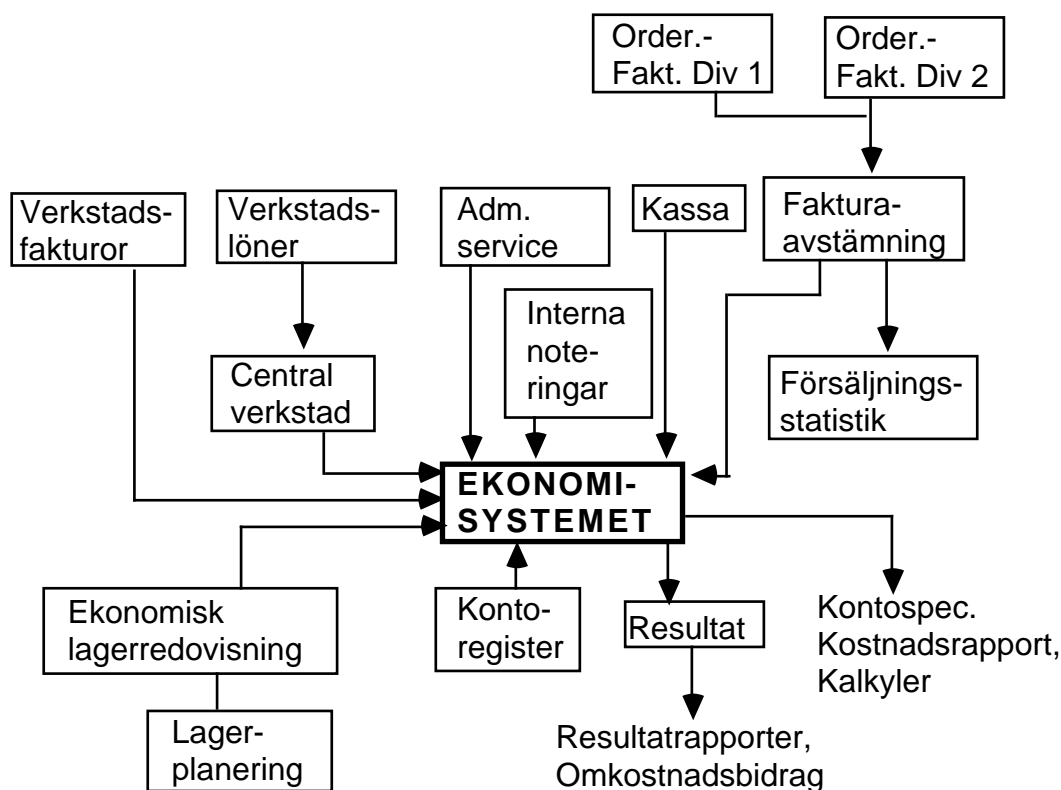


Fig 5.6 Ekonomisystemet och dess försystem

Från ekonomisystemet kommer kostnadsrapporter på olika detaljeringsnivåer. Den mest detaljerade innehåller redovisning per enskilt konto och underkonto, medan den mest aggregerade endast innehåller redovisning per kontoklass. De framställs en gång per månad och körningarna tar relativt lång tid. Ekonomiavdelningen får samtliga listor, medan divisionerna bara får det som berör dem. Listorna är tänkta som en slags uppslagsböcker. En av projektdeltagarna på centrala ekonomiavdelningen har undervisat i hur listorna ska användas. Vissa lär sig det medan andra går till dem som kan och frågar. Listorna är uppbyggda hierarkiskt med olika detaljeringsgrad. Man vill komplettera listorna med sökningar på skärm. Ingen av projektdeltagarna på Företaget tror dock att det är lättare t ex för en försäljare att använda en terminal än att använda listorna. Terminalen skulle främst behövas för frågor som inte är fördefinierade.

Delsystemen körs enligt en given tidsplan. Denna går inte att ändra eftersom de olika körningarna bygger på varandra (fig 5.7). Det finns olika typer av register: rena transaktionsregister, registersummeringar och styrande register.

Varje transaktion har en kod som anger vilket försystem den kommer från. Transaktionerna utsätts för vissa kontroller och summeringar innan de matas in i kontospecifikationsregistret, som är hela huvudboken, såväl affärs- som driftsbokföring. Utdata från ekonomisystemet är i princip aktuell ställning i kontospecifikationsre-

gistret, med vissa summeringar, utdrag och ev omstruktureringar. Denna bearbetning styrs av ett antal register vilka kallas styrande register. Genom att ändra i dessa register kan man således styra den information man vill ha ut från kontospecifikationsregistret.

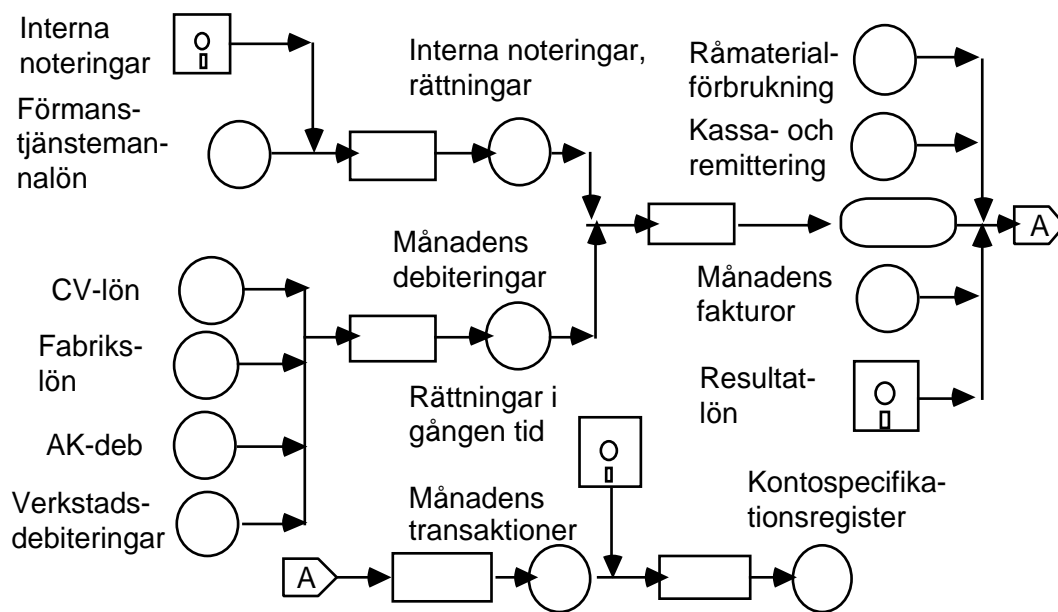


Fig 5.7 Bearbetning av uppgifter från försystemen

Kontospecifikationsregistret upptar c:a 35 MB, medan de transaktioner som berör ekonomisystemet, upptar c:a 350 MB. Det är således en kraftig aggregering av uppgifter som görs i samband med inmatningen till ekonomisystemet. Vid analys av uppgifter i kontospecifikationsregistret har dock ekonomiavdelningen ofta behov av att kunna se vad som döljer sig bakom en viss siffra. I samtliga försystem finns uppgifter om både kronor och kvantiteter, utom i kassasystemet, som enbart har kronalsuppgifter. Kvantitetsuppgifter kontrolleras i nuvarande system sämre än kronalsuppgifter. Krontalen avstämmas alltid vilket inte sker med kvantiteterna.

Kontoplanen

Kontoplanen bygger på en modifiering av Mekanförbundets normalkontoplan. Klass noll är tillgångar, skulder och eget kapital, klass ett är ett utgifts- och inkomstkonto. I klass två samlas alla råvaror och lager, i klass tre samlas alla tjänstemannakostnader, i klass fyra finns hjälpkostnadsställen, i klass fem finns alla fabriksomkostnader, i klass sex ligger nyanläggningarna, i klass sju ligger direkta kostnader t ex råmaterial, direkt lön och andra direkt till produkten hänförliga kostnader, i klass åtta finns intäkterna, i klass nio bokslutet. Klasserna noll, ett, åtta och nio utgör finansbokföringen, övriga är driftsbokföringen. Dessutom finns en serie underkonton för

både finans- och driftsbokföringen. Dessa anger speciella kostnadslag som t ex drivmedel, frakter m m.

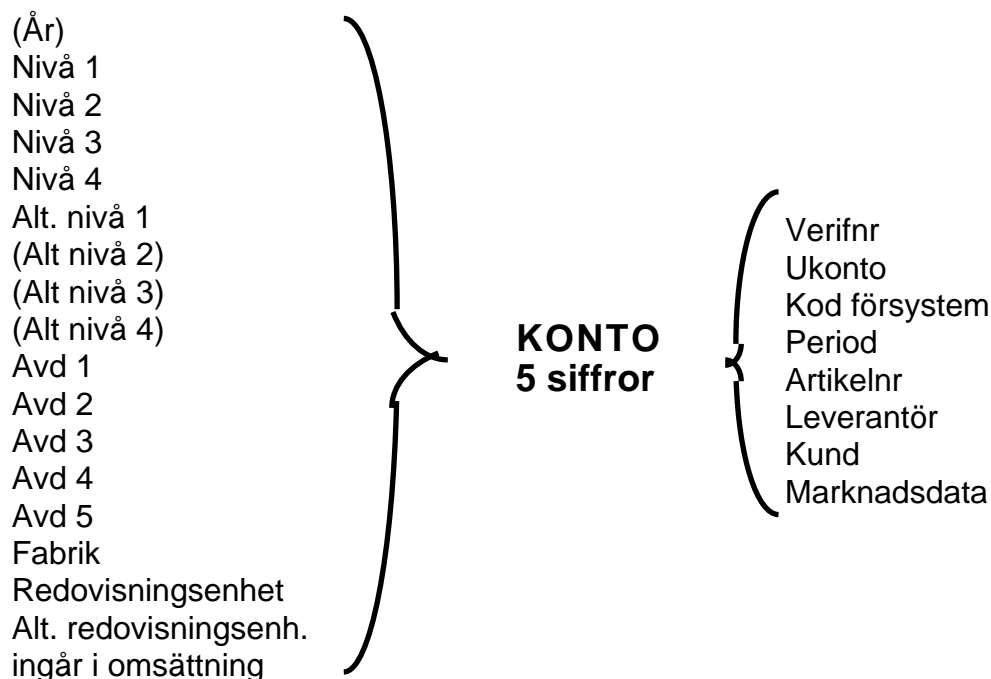


Fig 5.8 Kontobegreppet vid Företaget

Till varje konto finns en "vänstersida" och en "högersida" (fig 5.8). Med hjälp av dessa kan man koppla samman kostnadsställe och kostnadslag. Innehållet varierar mellan olika kontoklasser och figuren är bara ett exempel. Vänstersidan kan där grovt sett representera kostnadsstället, både produktorienterat (nivå) och organisationsorienterat (avd) medan högersidan, också den mycket grovt sett, representerar kostnadens ursprung. Kostnadsstället anges hierarkiskt och det finns möjlighet att ange alternativa hierarkier. Lönekostnaden för att centralverkstaden reparerar en maskin kan t ex hänföras både till den produkt som tillverkas på maskinen och till centralverkstaden.

Högersidan har inte en lika konsistent uppbyggnad. Den innehåller uppgifter bl a om kostnadslag (underkonto), kostnadsställe (försystem) samt även detaljerade ursprungsuppgifter (kund, artikel).

Det kan t ex vara så att nivå 1 är produkt A, nivå 2 är olika sorter av produkt A. Vänstersidan kan då ses som ett slags identifikationsbegrepp till det femsiffriga kontonumret. På så sätt kan man via hierarkier bygga upp strukturer. Genom att summera vissa konton kan man få reda på allt om t ex gröna produkter A. Det är också möjligt att identifiera t ex grönfärgning i olika fabriker.

Strukturen och innebörden av dessa båda "sidor" finns i kontoregistret, vilket är ett styrande register. När kontot används går man först ut till kontoregistret och anger

då t ex att man vill ha tag i allt om Div A, fabrik B. Kontoregistret ger då ett antal konton kombinerade med olika underkonton vilka innehåller den önskade informationen. För ett och samma konto kan det således finnas många olika beskrivningar. I dag jobbar man med två olika kontoregister:

- Ett tillverkningsorienterat kontoregister (beskrivet ovan).
- Ett marknadsorienterat kontoregister för uppföljande statistik (Förs.stat i fig 5.7).

Uppgifterna inom parentes i fig 5.8 finns inte idag, men man önskar sig dem. För övrigt anser man att det nuvarande systemet väl fyller sin uppgift. Det bör också påpekas att figuren ovan bara är ett exempel på hur det kan se ut. De olika kontoklasserna har olika vänstersidor. Ett och samma postfält kan alltså betyda helt olika saker beroende på vilken kontoklass som behandlas.

Kontoplanen är strikt produktorienterad. Tidigare hade man dock en marknadsavdelning som stod för all försäljning. För de utländska dotterbolagen är det fortfarande så, eftersom man inte kan ha fyra olika dotterbolag på varje ställe. Det kan då bli konflikter mellan en stark dotterbolagschef och en divisionschef.

Högersidan knyter via verifikationsnumret själva affärshändelsen till kontot. Det sätts oftast (utom vid interna noteringar) automatiskt av datasystemet. Underkontona får man däremot oftast sätta själv. I ett fåtal fall sätts de dock automatiskt. Kod anger oftast vilket försystem transaktionen kommer från. Den sätts automatiskt. Det finns möjlighet att på ett visst underkonto följa de olika transaktioner som finns på detta. Man kan alltså på koden direkt se om det är t ex lön, verkstadsdebitering, administrativ service etc. Period anger vilken månad summan hänför sig till. Det hör vidare närmast till undantagen idag att artikelnumret är med.

5.7.3 Vad som hände

Vi var redan tidigare bekanta med en del personal på Företaget. Tre personer på företagets dataavdelning hade tidigare gått på en av mina kurser på universitet. En annan av LUIS-gruppens medlemmar, Siv Friis, hade dessutom tidigare arbetat på Företaget och kände redan två av användarna på ekonomiavdelningen. Hon hade redan påbörjat ett projekt med dem. När vi fick indikationer om att man från STU:s sida föredrog ett fördjupat samarbete med ett företag istället för ett ytligt samarbete med flera var det naturligt att Företaget tillfrågades. Sålunda bildades en projektgrupp bestående av datachefen, två användare från ekonomiavdelningen, jag och Siv Friis.

Vår ansökan till STU var ambitiös och innebar nästan en total genomgång av företagets alla system. De forskningsmedel vi sedan fick räckte till att starta ett projekt, där vi skulle gå genom ekonomisystemet. Från början var det meningen att övriga projekt inom LUIS-gruppens forskningsprogram skulle delta, men man blev engagerade på annat håll. En del arbete gjordes dock inom IRIS-projektet. Vi forskare gjorde klart att vår roll var att stå till tjänst med metod- och teknikkunskap, idéer och kritik. Vi

betonade att vi inte fick betraktas som konsulter utan var forskare. Projektet organiserades på traditionellt sätt med en projektgrupp där en av företagets dataexperter ingick (han hade samma funktion som datachef, även om han inte kallades så), två representanter för användarna på ekonomiavdelningen och så forskarna. Datachefen var projektledare visavi Företaget och jag var projektledare visavi universitetet. Projektet startade på hösten 1983.

Problemet var att göra det befintliga ekonomisystemet mera tillgängligt. Därigenom skulle man lättare kunna ta ut tillfälliga rapporter. Idén var att bygga en databas bredvid det befintliga ekonomisystemet och göra denna nya databas lättare tillgänglig och sökbar med hjälp av något databasspråk eller applikationsgenerator. Den nya databasen skulle innehålla mera detaljerade uppgifter än den gamla i och med att den ligger närmare försystemen. En del av ekonomipersonalen ville även ha tillgång till vissa uppgifter i försystemen, men man bestämde att nöja sig med det som fanns.

Vi bestämde oss för att jobba med denna databas, vilken vi snabbt döpte till SOKRHATES-databasen. (H:et står där bara för att skilja det från forskningsprojekt SOKRATES). Uppgifterna i denna databas blir således mera detaljerade än i ekonomisystemet. I samband med överföringen måste uppgifterna noga kontrolleras för i annat fall kan man få problem med avstämning av uppgifter från SOKRHATES-systemet och nuvarande ekonomisystemet. Detta system ska fortfarande köras, men SOKRHATES kör parallellt bredvid.

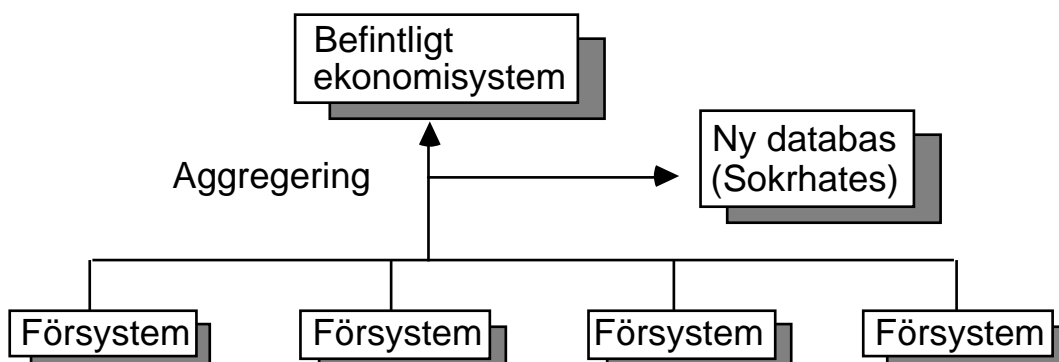


Fig 5.9 Utbyggnad av företagets ekonomisystem

Som en första upptakt hade jag och Siv Friis gjort en liten och enkel skiss till det kommande ekonomisystemet. Jag kallade den för *begriplighetsskiss*. Med detta menar jag en prototyp som realiserar av systemutvecklaren som kontroll av att vederbörande förstått en sak riktigt och fullständigt. Tanken är man vid realisering av denna skiss, vilket i regel innebär programmering i ett superhögnivåspråk eller användning av frågespråk mot databaser, måste tänka genom problemet så grundligt att eventuella oklarheter därigenom framstår.

Skissen kom dock att mera likna en prototyp, dvs ett förslag till en del av det nya systemet. Kravspecifikationsarbetet gick därmed över i arbete med prototyper. Prototypen bestod av en generell sökbild, med följande utseende (Fig 5.10):

Kostnadsställe:	<input type="text"/>
Kostnadslag:	<input type="text"/>
Belopp:	<input type="text"/>

Fig 5.10 Sökbild för skiss

Man kunde ange namn eller delar av namn på kostnadslag och kostnadsställe samt om beloppen skulle vara större än eller mindre än ett givet belopp. Utrymmet kunde också lämnas blankt. Det fanns ett transaktionsregister, som innehöll transaktioner med samma format som sökbilden ovan. Detta genomsöktes i sekvens och poster där träff med samtliga argument inträffade skrevs ut. Det fanns också två specialprogram som beräknade totaler för olika kostnadslag eller kostnadsställe. Kostnadsstället var uppbyggt som en hierarkisk kod. Jag antog att det fanns tre divisioner A, B och C, samt 3 fabriker 1, 2, och 3 på varje division. Varje fabrik hade i sin tur 3 avdelningar. Om jag t ex ville ha ut alla affärsomkostnader på division B, fabrik 2, avdelning 1 så skrev jag AFFO (eller bara FF, eller eller något annat entydigt) på kostnadslaget och B21 på kostnadsställe. Då kom önskade transaktioner på skärmen, dock utan budgetjämförelse eller summeringar. Skissen skrevs i Dbase II, som är ett databasprogram för persondatorer. Vid presentationen för projektdeltagarna gav den upphov till en livlig diskussion om hur det färdiga SOKRHATES-systemet skulle se ut.

Vi enades om följande huvudmeny :

1. Förutbestämda listor
2. Förutbestämda frågor
3. Spontana listor
4. Spontana frågor
5. Kataloger över befintliga frågor och listor
6. Ta bort gamla eller oönskade listor eller frågor
7. Rätta i register

Om en fråga antingen tar lång tid att besvara eller ger ett långt svar bör det bli en lista istället. I punkterna 3 och 4 ovan kan man tänka sig en funktion som anger förväntad söktid eller förväntat antal svar. Användaren kan då få möjlighet att ångra sig. Det är vidare tänkt att man skall kunna lagra sina frågor eller begäran om listor för att kunna använda dem fler gånger. Punkten 5 är en katalog över sådana lagrade utsökningar.

Användarna kan med hjälp av detta system analysera varför det gått som det gått. Det kan ibland leda till frågor till kostnadsansvariga. Dessa borde själva vara intresserade av att kunna ställa liknande frågor och i planerna ingick att systemet ska ställas till

övriga divisioners förfogande. Det fanns olika traditioner i handhavandet av listorna på de olika fabrikena. En del läser dem aldrig, andra lusläser dem. Vi krävde av systemet:

1. Användaren skall på förhand få besked om förväntad söktid och förväntat antal träff om dessa är av större omfattning.
2. 2. Automatisk backup
3. Städfunktion på sällan utnyttjade listor och frågor.
4. De enskilda användarnas sökmönster skall kunna lagras så att ett tradigt traskande i en mängd menyer undviks.
5. 5. Om det gäller att ta fram en spontan lista skall systemet tala om ifall någon annan nyss tagit fram samma (eller nästa samma) lista.
6. Vid de förutbestämda listorna skall det först tagas fram preliminära listor, som kollas vid källan innan de sprids vidare.
7. Om man efteråt konstaterar att en transaktion är felaktig måste alla analyser där denna transaktion ingår redovisas.
8. Systemet skall automatiskt göra budgetjämförelser. Dessa skall kunna räknas om till kr/enhet och/eller procentuella jämförelser.
9. Skall finnas möjlighet att uppdatera sökord.
10. Klartextåterkoppling på frågor.

Någon känslighetsanalys ansågs inte vara nödvändig. Budget och annan ekonomisk redovisning sker per månad så det fanns ingen anledning att ha någon annan uppdateringsperiod här.

Datachefen slutade på företaget vid årsskiftet 1983/1984. Han efterträddes som projektledare av en ny användare. Denne hade lång erfarenhet från Företaget och blev nu chef för den centrala ekonomiavdelningen. Datachefens efterträdare deltog periodvis i projektarbetet. Den nye användaren hade varit med i arbetet ända sedan mitten av december, så övergången blev ganska smidig. Vidare lämnade Siv Friis projektet i mitten av februari. Det berodde på att hon hade blivit mycket mer engagerad i ett annat projekt än som från början var tänkt.

Detta projekt kan antingen betraktas som ett lokalt projekt för ekonomiavdelningen eller som ett första prov på en helt ny datapolicy på Företaget. Sett som ett lokalt projekt är det inte rimligt att satsa några större summor. Frågan är om en åtkomst på transaktionsnivå från försystemen i detta fall är ekonomiskt möjlig. Det krävs i så fall c:a 350 MB i sekundärminne. För att nå framgång ansåg jag att en rejäl satsning, eventuellt i samband med övergång till ny datapolicy, vara nödvändig. Som pilotprojekt var detta i det närmaste idealiskt. Det lät sig lätt och till ringa kostnad byggas ut till fler användare.

Kännetecknande för den nya datapolicyn skulle kunna vara en satsning på interaktiv systemutveckling och redskap för sådan verksamhet. Den skulle möjliggöra en ökad decentralisering av dataverksamheten, samtidigt som den också skulle medföra ökad användning av datorer på Företaget och bättre användaranpassning av dessa. För

dataavdelningens vidkommande skulle det innebära att det inrättas en särskild enhet på dataavdelningen där man hjälper användarna komma igång med sin datasystemutveckling, ordnar kurser i handhavandet av sådana verktyg m m.

Om en ny datapolicy ska införas är det viktigt att i projektarbetet ta hänsyn till faktorer som bedöms underlätta denna nya datapolicy. Det innebär t ex att beslut måste tas angående val av genomgående maskinutrustning och operativsystem och som en förlängning av detta val av ett eller flera utvecklingsredskap. Vid ett möte i början av 1984 framhöll den nye datachefen att Företaget hade en klar policy vad gällde nya datorer. De skulle vara av ett visst bestämt märke. Detta var något oväntat, både för mig och för användarna. På detta möte sa datachefen också att man nog kunde tänka sig gå upp till ungefär en miljon i totalinvestering. Tydligt hade man kommit till samma slutsats som jag, nämligen att SOKRHATES måste bli ett koncernövergripande system och inte ett system dedicerat till ekonomiavdelningen.

Beslutet att satsa på ett större system fattades under april månad 1984. I mitten av maj månad hölls ett möte där projektgruppen, datachefen, ekonomidirektören och två controllers på divisionerna var närvarande. Syftet var att få krav och synpunkter från divisionerna på utökningen av ekonomisystemet. Följande hände på mötet.

Controllern för den största divisionen ville kunna gå tillbaka till den enskilda transaktionen i förssystemet. Detta gällde främst lönesystem och reparationssystem. Han tyckte inte heller om att det kom in främmande kostnader på hans konton. Detta beror på att det inte fanns någon spärr för felkonteringar. Vidare var listorna som kom ut svärbegripliga eftersom de inte innehöll någon text, bara ett kontonummer. Han hävdade också att om man rotar tillräckligt djupt i systemet för verkstadsdebiteringar, så hittar man alltid något fel. Ekonomidirektören sa att detta system ska bytas ut.

Controllern menade vidare att om man såg varje enskild post redovisad skulle man nog reagera mera än på en summa där stora plus- och minusvärden kan ta ut varandra. Om bara summan redovisas får man ju inte reda på stora kostnader. Han ville också att man skulle få informationen när det behövdes och inte någon dag senare. Då blir den aldrig framtagen.

Därefter redovisade kontrollern för den andra divisionen sina krav. Han förklarade att förmännen var kostnadsmedvetna och lämnade ett mycket gott underlag till budgeten. Följaktligen ville de gärna veta vad differenserna beror på. Detta kunde han dock inte förklara eftersom han inte kunde alla de kryptiska koderna. Han hade ingen lust att belasta sin hjärna med sådant. Controllern ville veta vad som låg bakom en viss siffra och det ville han veta genom att ha ett antal personer att fråga.

Vidare ville han på kostnadsfördelningen ha reell bruttokostnad och reell nettokostnad. Han tog som ett konkret exempel en forskningschef, som i sammandraget har ett hyfsat resultat men där stora kostnadsökningar balanseras av motsvarande taxehöjningar. Genom att bara redovisa sammandraget blir det ingen verklig kostnadskontroll. Man kunde också tänka sig att ett laboratorium gjorde fler analyser till

samma kostnad. Detta innebär då att det på något vis borde visas och "premieras". Detta ville inte den andre kontrollern och han tyckte att denna fråga på ett aldeles utomordentligt sätt belyste vad som händer i ett gemensamt ekonomisystem när båda tycker man har rätt men ändå måste jämka samman sig. Grundfrågan är hur kontrollern ska kunna göra sitt jobb när han inte får gehör för sina idéer?

Datachefen tog upp frågan om centralt eller divisionaliserat ekonomisystem. Den var något känslig men kontrollern för den största divisionen sa att om man på hans division inte fick gehör för sina krav låg det nära till hands att skaffa ett eget system. Den andra divisionen hade däremot inte alls några planer åt det hållet. Förhoppningsvis ska SOKRHATES-systemet göra det lättare för divisionerna att få gehör för framtida krav.

Sammanfattningsvis uppnåddes en viss enighet kring det förslag som lagts av SOKRHATES-gruppen. Denna databas skall i princip vara på transaktionsnivå, där transaktionerna hämtas från de olika försystemen. Varje division kan med hjälp av någon applikationsgenerator utforma sina listor och sina frågor precis som man vill, naturligtvis inom ramen för vad datamodellen tillåter.

Därefter arbetade vi parallellt med att göra en datamodell för SOKRHATESregistren och besöka en del leverantörer och för demonstration av olika redskap för interaktiv systemutveckling. I olika omgångar tittade vi på Mapper, Focus, Genius, Query på Ericssons minidatorer, SQL och QBE (Query By Example) vilka ingår i IBM:s QMF (Query Management Facility). Vårt allmänna intryck var snarare förvirring än klarhet. Användarna hade efter ett tag intrycket att alla var ganska lika. Möjligen tyckte man att Genius var något besvärligare än de andra. FOCUS hade en funktion som hette "Tabletalk" där maskinen ställde frågor i ett fönster på bildskärmen och samtidigt skrev programmet i ett annat fönster och gav hjälp i ett tredje. Detta imponerade stort på oss alla. Vidare tyckte man att QBE "var rätt roligt". Men det var inte så att man obetingat ville ge något redskap absolut företräde. Dessutom började mina andra undersökningar visa att valet av redskap inte var så kritiskt som jag hade trott tidigare.

Vi hade problem med datamodellen. För att konstruera en sådan krävs ett antal informationskrav. Några sådana gick emellertid inte att ställa eftersom man inte konkret kunde säga vilken information man ville ha tillgänglig. I princip var det all information i försystemen. Med detta som utgångspunkt gjorde vi en skiss i tredje normalformen. Dock fanns det många termer man var oklar över och i de olika registren i försystemen användes samma term i olika betydelser.

I juni 1984 fick Företaget kontakt med IBM genom ett seminarium där QMF presenterades. QMF är en påbyggnad på de två databasspråken SQL (f d Sequel) och QBE (Query By Example). Det bli lite enklare att lägga upp färdiga procedurer, register, rapporter och formulär. Man efterhörde redan då möjligheterna att lägga upp några egna register och testköra. Detta skulle gå bra.

Men först i början av november 1984 kom man igång. En särskild identitet lades upp, ett antal band (till slut blev det 16 stycken) skickades upp till IBM i Kista och användarna började, tillsammans med en person på dataavdelningen, lägga upp ett antal register med tillhörande kringrutiner. I stort sett konverterade man det gamla systemet. I början av maj 1985 var detta färdigt för demonstration för andra användare och direktion. Man hade en hel del problem under uppläggningsarbetet. Bl a hände det relativt ofta att systemet i Kista gick ner och att man fick vänta flera timmar på att det skulle komma ingång igen. Däremot var det inga problem att få in grunddata. Arbetet skedde parallellt med användarnas övriga arbete och tog därför tämligen lång tid. Bortsett från detta fungerade testinstallationen relativt bra och mottogs med entusiasm av företaget. Man har vid årsskiftet 1985/86 fattat beslut att installera ett sådant system plus ett nätverk som gör att även de enskilda divisionerna, med sina olika datasystem, kan använda databasen. Detta är som ett led i företagets decentralisering samtidigt som det möjliggör ett väl tillgängligt ekonomisystem för hela koncernen.

5.7.4 Kommentarer

Detta avsnitt skulle också kunna finnas i kapitel 6, men jag tycker det är mest logiskt att sätta det här, eftersom det direkt hänför sig till vad som hände på Företaget. I projektet har vi stött på några av de problem Andersson (1983) anger. Problemet med behov av mycket datakraft har vi stött på via två vägar, dels genom den stora datavolymen dels genom kostnaden att bygga ut den befintliga IBM-maskinen.

Kravspecifikation

I början av projektet använde vi oss också av något felaktiga systemeringsmetoder i och med att vi försökte få fram en traditionell kravspecifikation. Vi försökte använda följande metoder för att få fram kraven på den nya databasen:

- Missnöjesanalys av det nuvarande systemet.
- Ta reda på vad som redan finns i försystemen.
- Studera verksamheten och se vilken information som behövs för densamma.
- Utdataspecifikation.
- Prototypanvändning.

Utdataspecifikationen (dvs den traditionella metoden) förespråkades varmt av datachefen. Vid något tillfälle sa han att vi skulle ställa användarna mot väggen och tvinga fram en kravspecifikation. Siv Friis var rätt mycket för tanken att man skulle hitta någon flaskhals och börja nysta upp saker och ting därifrån, alltså en kombination av missnöjesanalys och utdataspecifikation. Själv trodde jag att verksamhetsanalys skulle vara möjlig att använda, men insåg snart att "information" och "verksamhet" förekom på minst fyra olika nivåer (fig 5.11). Att hålla isär alla dessa nivåer och de olika regler som gäller för varje nivå skulle vara alldeles omöjligt.

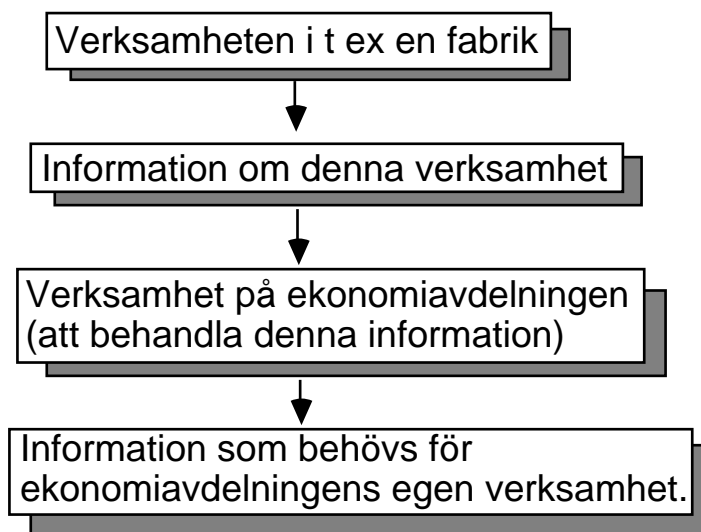


Fig 5.11 Olika nivåer av information i samband med ekonomisystemet

En av användarna förespråkade varmt att ta reda på det som redan fanns. Kravspecifikationen skulle i princip bestå i tillgång till allt som fanns i försystemen. Datachefen hade invändningar mot detta sätt och menade att det gav inga direkt nya infallsvinklar. Han antydde också att det skulle bli besvärligt att ta fram alla uppgifterna.

De skisser och prototyper vi gjorde spelade en viss roll vid framtagning av kravspecifikationen, i och med att användarna satt och jobbade med dem och således blev tvungna att lägga in material och att ställa frågor på detta material. Men vi hade problem att testa idéerna rent praktiskt. Jag hade tillgång till en CP/M-dator med Dbase II som databassystem. Det var i och för sig möjligt att lägga upp databasen och testa ett antal olika kringrutiner, men det var alldeles orealistiskt att försöka testa med nägorlunda volymer av data. För att kunna göra ett realistiskt test och utvärdering krävdes i stort sett ett års ekonomiska data inlagda och sökbara. De upptog 400 MB, vilket ju var alldeles omöjligt att realisera på en mikrodator med 1 MB diskettutrymme!

Andra personer och forskare jag diskuterat med har ställt sig skeptiska till behovet och nyttan av hela denna volym jämförd med någon slags statistisk sampling, men ett sådant förslag föll inte i god jord. Användarna menade att det var fråga om ett bokföringssystem och då måste allting stämma. Intill sista kronan! Inte heller gick det att begränsa sig till bara ett litet område eftersom hela ekonomisystemet var så sammantvinnat. Att begripa hur det var uppbyggt var för övrigt omöjligt. Jag försökte flera gånger och trodde ibland att jag hade fångat det, men alltid gled det mig ur händerna. Det gjorde naturligtvis att de övriga användarna med ett litet lätt overseende leende mottog alla mina s k konstruktiva förslag till att gå vidare.

Om man startar personlig databehandling av ovan beskrivet slag försvinner den tekniska handhavandekunskap som behövs för att sköta dagens system. De nuvarande användarna behärskar dagens system men vid en övergång till mer användaroriente-

rat bruk behövs inte denna kunskap längre. Kodkunskap och kodsystém verkar i och för sig inte ha något större värde och en övergång till mera klartexttalande förefaller vara enbart positiv. Men kodsystémet ingår dock som en del i användarnas yrkeskunskap. Som forskare och lärare tycker jag att koder är något fruktansvärt och jag tar tillvara varje tillfälle att för mina elever visa hur man kan undvika dem. Men i detta fall riskerar jag hamna i ett visst etiskt bryderi. Åtminstone teoretiskt kan jag stödja en process som skulle kunna utarma mina medarbetares yrkeskunskap. Enda sättet att få reda på något om denna process är att initiera den. Men det verkar vara en irreversibel process. Jag har diskuterat saken med mina medarbetare på Företaget och till all lycka ser de ingen förlust alls i att mista kodsystémet. Viss yrkeskunskap kan man vara utan, det är bara frågan om vem som tar bort den, yrkesmänniskan eller dataexperten.

Att förstå ekonomisystem ett kommunikationsproblem?

Ett misstag jag gjorde som forskare var att jag försökte bli lika duktig som användarna i att använda det nuvarande ekonomisystémet. Då mina kunskaper i redovisning och kontoplaner är relativt begränsade, var det inte lätt för mig att förstå systémet. Ekonomipersonalen utgick också, helt omedvetet, från att jag kunde systémet lika bra som de och förklarade på en tämligen detaljerad nivå. Dessutom behärskade jag inte det speciella yrkesspråket inom företaget och gav de använda begreppen helt olika innebörder. Ett exempel är följande. Tidigare (fig 5.8) har jag talat om "högersida" och "vänstersida" i samband med huvudkonto. På vänstersidan finns det något som kallas "nivå". Detta hade jag hela tiden förknippat med "organisatorisk nivå", trots jag i flera rapporter och PM klart och tydligt har skrivit att det rör som om olika produkter. Nivåbegreppet var en produktorienterad hierarki.

Såvitt jag har förstätt är det befintliga ekonomisystémet i grunden mycket förändringsbart. Som en av användarna brukar säga: "Vi kan egentligen vränga ut och in på systémet precis som vi vill. Men det är ett djävla jobb!" Mycken tid gick åt till att försöka förklara egentligen alldeles oväsentliga saker och jag försökte hela tiden beskriva systémet för dess användare, i min terminologi. Detta var naturligtvis dömt att misslyckas och jag insåg också, fast inte förrän efter 5 månader, att det egentligen var alldeles onödigt att jag skulle förstå deras ekonomisystem lika bra som de själva när ju faktiskt användarna själva skulle göra utvecklingen. Det var inte min uppgift att lösa deras problem, jag skulle istället jobba mer övergripande och föreslå metoder att lösa problemen. Hade det rört sig om ett traditionellt projekt hade det varit annorlunda. Där förväntas dataexperten även delta i användarnas problemlösning. Beträffande dataexpertens kunskap i detta fall råder det delade meningar mellan mig och projektdeltagarna på företaget. Jag hävdade att dataexperten måste kunna användarnas arbete bättre än de själva för att på ett insiktsfullt sätt kunna ändra på det. Men det höll inte ekonomipersonalen med om.

I mina försök att begripa ekonomisystemet försökte jag med hjälp av min uppfattning om användarnas koder bygga upp först en hierarkisk datamodell och sedan en nätverksdatamodell. Men det visade sig hela tiden att jag missuppfattat väsentliga begrepp. Jag saknade en dokumentation på en mer övergripande och lättbegriplig nivå. Den jag fått av användarna var paradoxalt nog alldeles för dataorienterad för att jag (dataexperten!) skulle förstå den. Men när jag och Företagets projektledare satt och diskuterade dessa saker och jag klagade över att jag inte hade någon mellannivå, så vände han sig om och tog ner en pärm från bokhyllan på vilken det stod "Kontoplan". "Det skulle väl vara den här då!", sa han. Det var precis den översiktliga, men ändå detaljerade beskrivning jag sökt efter hela tiden.

Kravspecifikationer i teori och praktik

Under arbetet med prototyperna fick användarna ställa "frågor" till det tänkta systemet. Sedermera gjorde även Företagets projektledare en grov kravspecifikation. De krav som framfördes var av typen: "Kostnadsstruktur för bränsleförbrukningen på adv 4 i fabrik B på division A". Detta är inget krav, om man talar "allmänna dataspråket" men i det specifika yrkesspråket på Företaget var det ett krav. Själv undrade jag över vad "kostnadsstruktur" kunde tänkas innebära, men jag valde att inte dyka in i det vid ifrågavarande tillfälle, eftersom det lätt hade kunnat leda arbetet långt på villovägar. Men såvitt jag förstod var kravet snarare fråga om vad man vill göra med informationen än vilken information man vill ha. Så småningom begrep jag att "kostnadsstruktur" innebar någon form av uppdelning men exakt vilken vet jag inte. Denna kunskap hör till den specifika yrkeskunskapen hos användarna på Företaget. Dessutom vill man därefter ha ytterligare uppdelning av vissa kostnader och sedan ytterligare uppdelning. Vilket som ska uppdelas och hur det ska uppdelas kan inte sägas på förhand. Det ingår i den informella yrkeskunskapen, att direkt kunna "se" att en viss siffra är "mystisk". Liknande fenomen beträffande registreringsarbete har beskrivits av Göranson (1983) och beträffande laboratoriearbete av Sandström (1985). Traditionell kravspecifikation förutsätter att begreppen är välkända och väldefinierade från början och att alla uppdelningar är kända på förhand. Att så ofta inte är fallet kan vara en av orsakerna till att datasystem i allmänhet inte ger de förväntade resultaten. Detta sätt att göra kravspecifikation kan kallas det *verksamhetsorienterade* sättet.

Ett annat exempel är datamodellering enligt relationsdatamodellen. Ta t ex kontoklass 3, "Allmänna omkostnader". I denna kontoklass finns ett antal konton, från 30010 "Styrelse och revision" till 39690 "Skyddspatent varumärken övrigt". På vart och ett av dessa konton finns transaktioner, vilka representerar enskilda affärshändelser eller aggregeringar av affärshändelser. Om det är aggregeringar, kan man i motsvarande förssystem lösa upp dem i transaktioner representerande enskilda affärshändelser. Om detta skulle representeras i en relationsdatabas, skulle det bli ungefär följande:

- Kontoplan (Kontoklass, kontoklassbenämning, datum, belopp)
- Kontoklass (Kontonummer, kontobenämning, datum, belopp)
- Kontotransaktion (Transaktionstyp, försystem, datum, belopp)
- Transaktionstyp (Transaktionsnr, försystem, datum, belopp)

Detta ser ut att fungera fint, åtminstone på pappret. Men vad som inte representeras i denna modell är olika summeringar som äger rum mellan nivåerna. "Belopp" har olika innebörder i alla fyra tabellerna ty de är summeringar på olika nivåer av enskilda transaktioner. Inte heller representeras reglerna för konteringar och förflyttning mellan olika konton. Det finns stora skillnader mellan olika konton. Det finns även skillnader i organisationen av kontona. I kontoklass 7 t ex finns det vissa gruppkonton, som ligger på en nivå mellan kontoklass och enskilt konto. Förflyttningsreglerna och de olika summeringarna kan inte, åtminstone inte på ett enkelt sätt representeras i en relationsdatabas. Detta sätt att göra kravspecifikation kan kallas det *dataorienterade* sättet.

Det verksamhetsorienterade sättet och det dataorienterade sättet kan i viss mån sägas komplettera varandra. Det verksamhetsorienterade koncentrerar sig på vad som görs med informationen, medan det dataorienterade koncentrerar sig på vilken information som behövs. Det verksamhetsorienterade sättet är användarnas sätt, medan det dataorienterade är dataexpertens. Enbart kunskapen om de olika sättens kravspecifikation är bra, men det behövs någon metod för att kunna övergå från den ena till den andra. En tänkbar sådan har jag skissat på i ett tidigare forskningsprojekt (Flensburg 1981c), men denna gick inte att använda i föreliggande fall. Orsaken var att problemet bestod i att ta fram information, som inte fanns. Vilken var alldeles unikt för varje gång problemet uppkom och därför kunde man inte heller säga vad som ska göras annat än i mycket allmänna termer. Att närmare klargöra förhållandena mellan dessa olika sätt att specificera sina krav är en viktig uppgift för fortsatt forskning.

Den tredje innebörden av kravspecifikation, *maskinbehovet*, talade vi ganska mycket om, men ingen kallade det för kravspecifikation. Från början trodde jag nog att val av dator och av redskap skulle visa sig viktigare än det verkar bli. Det är faktiskt ett väldigt stort språng man tar, från satsvis arbetande system med separata filer till applikationsgenerator med databas.

Martin (1982) talar om fyra olika typer av "data environments":

- *Files* som innebär enskilda, flata filer för varje tillämpning. Ingen databashantare.
- *Application data bases* som innebär att man har separata databassystem för varje tillämpning.
- *Subject data bases* som innebär att man har databaser oberoende av tillämpning och avsedda för tunga, satsvisa bearbetningar.
- *Information systems* som är databaser organiserade för snabb on-line åtkomst och med goda frågemöjligheter.

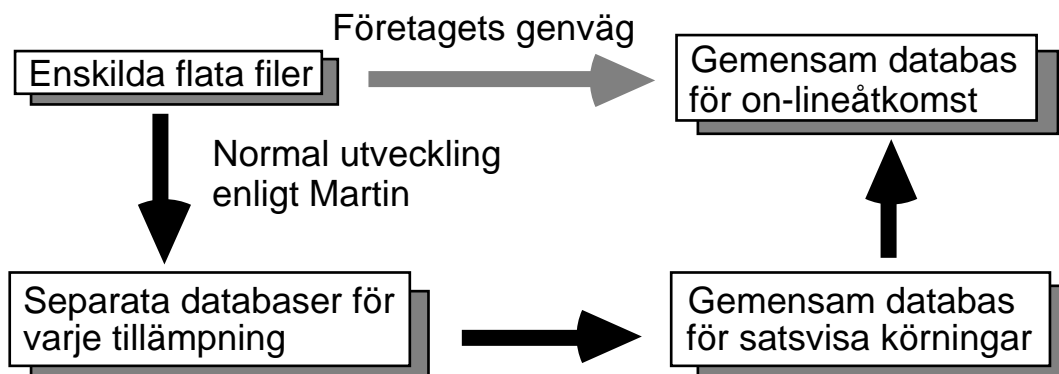


Fig 5.13 Utvecklingsvägar för databaser

I detta projekt försökte man gå en genväg från enskilda filer direkt till den sista typen, "on-line databaser" (Fig 5.13). En svårighet är naturligtvis att bestämma utseendet av den gemensamma databasen, speciellt som det inte finns något alls att bygga på. På ett projektmöte gjorde vi ett utkast, men kunde, då det saknades maskinresurser inte fullfölja. En prototyp realiserades dock i DBase II. Vi bestämde oss till sist för att skapa den gemensamma databasen successivt genom att allt fler och fler uppgifter plockas in från försystemen och sådana som inte används plockas bort. Det är meningen att det skall finnas särskild mjukvara för detta. Men det medför i inledningskedet att i stort sett samtliga uppgifter från försystemen kommer att föras över i den nya "Sokrhates-databasen".

Olika slags yrkeskunskap

Dessa upplevelser kan jag generalisera i form av olika sorters yrkeskunskap. Jag vill föreslå följande:

1. *Allmän yrkeskunskap* som inhämtas genom utbildning och är oberoende av företag. Kunskap som meddelas under t ex civilekonomutbildning eller sekreterarutbildning är exempel på sådan kunskap. I befattningsbeskrivningen för en viss befattning anges vilken allmän yrkeskunskap, som krävs.
2. *Specifik yrkeskunskap* som är en till det specifika företaget anpassad yrkeskunskap. Det är t ex kunskap om organisationens uppbyggnad, om kontoplan, om produktsortiment, om försäljningsvolym m m. Olika kategorier på företaget har olika specifik yrkeskunskap.
3. *Informell yrkeskunskap* som är kunskap om hur de övriga kunskapstyperna ska tolkas och användas. Man kan säga att det är kunskap om när man inte ska följa gällande regler. Denna kunskap är nära besläktad med det som bl a Göranzon kallar "tyst kunskap" (Göranzon 1983)

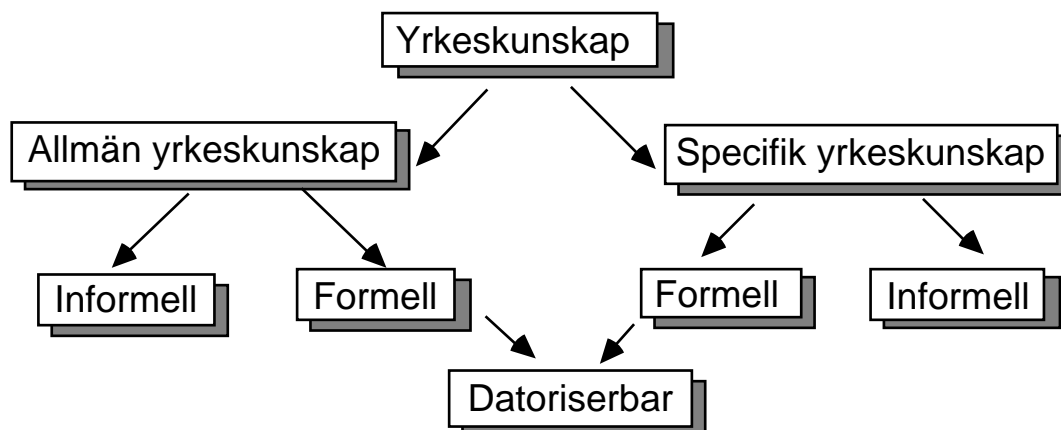


Fig 5.14 Olika yrkeskunskap

Alla dessa olika typer av kunskap uttrycks i var sitt språk med sina egna begrepp. Detta har bl a Høyer varit inne på (1974). Det är då inte märkvärdigt att jag i min specifika yrkeskunskaps begreppsvärld inte kunde göra mig förstådd i användarnas på Företaget specifika yrkeskunskaps begreppsvärld! Vi hade dock en delvis överlappande allmän yrkeskunskap, eftersom jag en gång i tiden läst ekonomi och två av användarna hade ganska god allmän datakunskap. De var bl a lärare åt Företagets egen personal vid företagsinterna datakurser. Men detta var inte till någon hjälp, snarare tvärtom. En gemensam allmän yrkeskunskap behöver inte alls medföra att den specifika yrkeskunskapen också är gemensam. Det istället många gånger vara ett direkt hinder, eftersom man tror att man är överens, man tror att man använder samma begrepp i samma betydelse. Ett bra sådant exempel är just användningen av "nivå"-begreppet. I min specifika yrkeskunskap var det "organisatorisk nivå" medan det för användarna överhuvudtaget inte kom för att "nivå" i detta sammanhang kunde betyda något annat än "produktivnivå". Bara för att jag är dataexpert, kan jag inte räkna med att begripa hur ett datasystem fungerar i användningshänseende. Rent tekniskt kan jag däremot väl förstå det, vilket kanske inte användarna gör alla gånger. En sekreterare som använder en ordbehandlingsmaskin vet t ex precis hur man gör för att få texten på ett visst sätt, vilka möjligheter som finns och vilka som inte finns, vilka som är praktiska att använda och vilka som inte är praktiska att använda. Som dataexpert kan jag mycket väl förstå hur programmet fungerar och vad som sker vid varje kommando. Sådant kunskap har sällan sekreteraren. Men för att jag har denna tekniska kunskap kan jag därför inte använda ordbehandlaren lika suveränt som sekreteraren gör. Det handlar i detta fall om olika yrkeskunskaper, men om samma företeelse. Denna hopblandning mellan teknisk nivå och användningsnivå har jag träffat på tidigare (Flensburg 1979) och det verkar vara en av orsakerna till missförstånd mellan användare och dataexperter

Beträffande det nuvarande ekonomisystemet kan jag identifiera tre nivåer av kunskap:

1. Kunskap om vad en viss transaktion *betyder för verksamheten*. Det är främst de resultatansvariga personerna som besitter denna kunskap, men även ekonomifolket har den i viss utsträckning.
2. Kunskap om hur man ska göra för *att få fram en viss information* ur ekonomisystemet. Denna kunskap har ekonomifolket, både centralt och lokalt. Resultatansvariga bör också ha det viss utsträckning.
3. Kunskap om *det tekniska handhavandet* av ekonomisystemet, vad alla koder betyder m m. Denna kunskap finns på den centrala ekonomiavdelningen men förmodligen inte på lokal nivå.

För att kunna utnyttja ekonomisystemet behövs idag kunskap inom alla tre nivåerna. Detta gör att centrala ekonomiavdelningen i många fall får rycka in och förklara även på lokal nivå vad det som står på listorna betyder. I det nya SOKRHATES-systemet skulle kunskapen på nivå 3 bli mycket mer tillgänglig och därmed skulle alltför personer kunna utnyttja ekonomisystemet på ett effektivt sätt. Betecknande är att kontrollern för en av divisionerna (han har för övrigt ett förflutet som datachef på Företaget) ville ha någon person att fråga istället för någon lista eller pärm.

En annan sak som kan tas upp i detta sammanhang är de skilda typerna av kunskap som tas fram i forskning och i "kontorsarbete". Vi forskare sysslar mycket med begreppsbestämningar och försöker skapa oss entydiga och väl definierade begrepp som alltid har samma innebörd. Vi strävar efter att producera explicit kunskap, klart och tydligt utsagd och tillgänglig för envar. Men mycken av den kunskap som används vid kontorsarbete är inte sådan. Den är istället av typen "informell yrkeskunskap" och yttrar sig t ex i en speciell "stil" eller möjligen "jargong" i arbetet. Det är en "tyst" kunskap, som inte kan inhämtas på annat sätt än genom praktiskt arbete på kontoret ifråga. Man kan rentav förmoda att det ligger i organisationens intresse att denna dolda kunskap förblir dold, eftersom en del av dess affärsidé ligger i denna tysta kunskap, i den värdegemenskap de anställda känner med företaget. Olika människor i samma organisation och på samma arbetsplats ger vanligtvis olika definitioner av samma begrepp. Detta har Sandström studerat (Sandström 1984, 1985). Forskare med ett annat kunskapsmotiv och en strävan att klarlägga saker, som kanske inte kan klarläggas eller bör klarläggas, kan i sådana situationer lätt uppfattas som en störningsfaktor. För att undvika detta måste man ställa upp på organisationens villkor på nästan samma sätt som om man vore anställd på firman.

Skisser och prototyper

I mina försök att begripa ekonomisystemet tog jag datorn till hjälp. Jag antog att först när jag löst ett problem kan jag anse mig ha begripit det. För att tvinga mig själv tänka genom problemet ordentligt försökte jag göra en skiss av det nuvarande systemet med hjälp av Dbase II. Min första avsikt var att göra skissen exakt som det nuvarande systemet, men den fick jag snart ge upp. Jag kunde ju inte alls beskriva systemet i

min dataterminologi. Dessutom var det alldeles för komplicerat för DBase II. Istället gjorde jag en mycket grov skiss till ett nytt system. Detta är egentligen inget annat än traditionellt dataexpertbeteende. Men mina skisser startade en livlig diskussion och under ett av mötena lyckades vi åstadkomma en hyggligt fungerande datamodell som kom att ligga till grund för den grova kravspecifikationen Företagets projektledare gjorde. Den dokumenterades första gången direkt genom att skapa motsvarande register i Dbase II!

Arbets sättet vid prototyparbetet var en form av trialand error. Eftersom det tar några timmar att lära sig Dbase och jag redan kunde det och användarna aldrig skulle komma använda det satt jag vid terminalen. Hade vi haft ett enklare redskap eller ett redskap som skulle användas senare hade jag givetvis inte gjort så. Jag producerade förslag, användarna sa vad som var fel och vad som behövdes ytterligare. Skissen förbättrades omgående och användarna såg att det kanske skulle vara på något annat vis istället. På så sätt kom vi gemensamt fram till den grova kravspecifikationen. Friis har utförligare diskuterat liknande arbeten (Friis 1986). Det kanske också ska sägas att man på Företaget vid ett tidigare tillfälle hade diskuterat samma sak, fast då med APL som hjälpmedel. Man hade tagit fram lite testmaterial och genomfört en testkörning.

En intressant iakttagelse jag gjort i andra sammanhang också är att användarna, dvs personer med verksamhetskunskapen plötsligt fördjupar sig i en diskussion kring vad en viss siffra eller viss transaktion eller viss term egentligen betyder. Jag har sett bibliotekarier, som mitt i arbetet med att beskriva hur man hittar i ett av deras kortregister har börjat fördjupa sig i en diskussion om klassifikationen av visst verk. Jag har sett Företagets användare, som helt plötsligt börjat undra om en transaktion på 13.62 kr verkligen var förd på rätt konto.

Användare i verkligheten

Användarbegreppet är ett besvärligt begrepp. Rent pragmatiskt har jag i detta projekt använt mig av en ostentativ användardefinition. De konkreta användarna av ett tänkt datasystem på Företaget har alltså varit följande:

Aug 1983 Dec 1983:	Företagets datachef 2 personer på centrala ekonomiavdelningen Per Flensburg från Lunds universitet
Jan 1984 Sep 1984:	3 personer på centrala ekonomiavdelningen Per Flensburg från Lunds Universitet (Controllers från två olika divisioner)

Parentesen om de sista användarna indikerar att de inte varit med vid projektmötena, men ändå på något vis haft tämligen stort inflytande på vad som gjorts. Möjligen kan man säga att de varit "immanenta användare".

Om vi ser på Företagets ekonomisystem så är det tänkt att betjäna hela företaget. Centrala ekonomiavdelningen tjänstgör, såvitt jag har kunnat förstå, som en slags förvaltare av ekonomisystemet. Man gör ändringar i systemet (t ex lägger till nya fördelningsregler, kanske rent av stugar om alla uppgifter för att få fram t ex försäljningsstatistik), man svarar på frågor, man kollar upp felaktigheter, man lägger in budget och man gör upp bokslut. Detta gör att centrala ekonomiavdelningen framstår som en slags förvaltare av systemet. Det rör sig då inte om datateknisk förvaltning, utan mer om innehållsmässig förvaltning. De konkreta förändringsåtgärderna görs i många fall av dataavdelningen. Man skulle kanske rent av kunna säga att centrala ekonomiavdelningen är "ägare" till det nuvarande ekonomisystemet. Detta är en typ av användaraspekt.

Men det är samtidigt inte alls givet att detta förhållande måste vara för tid och evighet. När vi forskare dök upp på scenen, såg vi genast att det måste ju finnas andra användare också, nämligen de resultatansvariga, de som utförde den verksamhet vilken var representerad av siffrorna i ekonomisystemet. Vi ville att även dessa personer skulle delta i projektet, för det var ju de som var de slutliga användarna, enligt vårt sätt att se. Dessa använder resultatet från systemet och kan därför kallas *resultatanvändare*. Det borde då också finnas *systemanvändare*. Detta visar på användningsaspekten av systembruk.

Men uppdelningen i resultatanvändare och systemanvändare är även den alltför grov. Centrala ekonomiavdelningen, som då skulle vara systemanvändare, är även i många fall resultatanvändare. Man skulle möjligen kunna säga att de är resultatanvändare gentemot vissa kontoklasser, men systemanvändare gentemot andra.

6. RESULTAT

Mitt arbete avser att bidra till en kunskapsutveckling. Törnebohm (1983) skiljer på kunskapsutvecklingar av olika ordningar. Kunskapsutveckling av ordning noll producerar data, t ex i form av mätresultat. En sammanfattning av sådana finns i kapitel 4 och 5. Kunskapsutveckling av ordning ett omfattar enstaka kunskaper i form av bekräftelser av hypoteser. Detta kan jag inte göra eftersom min undersökning är explorativ. Däremot kan jag ställa upp ett antal hypoteser, vilka än så länge är obekräftade, men som jag bedömer har viss sannolikhet att bli bekräftade i framtida undersökningar. Kunskapsutveckling av ordning två ska resultera i förklaringsmönster. Redovisning av ett typfall utgör ett sådant förklaringsmönster. Vad gäller kunskapsutveckling av ordning tre, speciella teorier, rör det sig här snarare om speciella teorifragment. Att en mellanhand uppkommer, att det rör sig om personlig databehandling är exempel på sådana. Kunskapsutveckling av ordning fyra, fundamentala teorier, finns det också något fragment av. T ex menar jag att uppdelningen av informationssystem i styr-, rutindataoch stöddatasystem utgör ett sådant. Nedan skall jag försöka följa denna ordning i min redogörelse av resultaten. Men först ska jag göra en sammanfattning av själva problemformuleringen. Som jag sade i inledningen är det ju först när problemet är löst som det fått sin slutliga formulering. Här vill jag på intet vis hävda att jag löst eller slutligt formulerat något problem, men en intermediär formulering är som följer.

6.1 Formulering av problemet

Datasystem uppfyller sällan de förväntningar som ställs på dem. Orsakerna härtill är flera och det går inte att utpeka en enda. Hittills tycks man ha antagit att bristerna i huvudsak beror på brister i de metoder eller tekniker man använder för att framställa datasystemet.

Men det syns mig som om många dessutom är överens om att *systemkonstruktörer och systemanvändare inte förstår varandra*. Många olika sätt har provats för att uppnå en ökad förståelse. Den kanske mest lovande ansatsen hittills är bruket av prototyper som kravspecifikationer.

Det har visat sig att *i de fall där användarna haft stort inflytande över systemutformningen har det blivit mera lyckade resultat*. En intressant fråga är då vad som händer om användarna får totalt inflytande över systemutformningen och -konstruktionen. Detta borde, enligt min förmodan, säkerställa ett relativt bra datasystem.

Ett annat (närbesläktat) problem är *underhållet av befintliga datasystem*. Alltmer av dataavdelningens resurser går åt till att genomföra ändringar i de befintliga systemen. Detta kan bero på underdimensionering av dataavdelningen men det även på ett omfattande reellt ändringsbehov.

Vidare finns det *nya tekniska möjligheter*, både hos datorer och programvara. Främst gäller detta de sk *applikationsgeneratorerna* och *de kraftfulla persondatorerna*. Dessa både kan och har använts för att sprida datakraft till användarna.

Att låta användarna själva utveckla sina system med hjälp av någon applikationsgenerator, *interaktiv systemutveckling*, anser jag bidrar till att lösa problemen ovan. Mitt forskningsproblem kan formuleras som följande fråga: *Hur kan interaktiv systemutveckling bidra till att skapa effektiva och användbara datasystem med hög grad av arbetstillfredsställelse?*

Jag har angripit problemet på två sätt. Det första är att undersöka några erfarenheter av interaktiv systemutveckling. Jag har valt att göra en explorativ undersökning i intervjuform eftersom det gäller att samla nya erfarenheter snarare än att testa färdiga hypoteser. Förutom denna explorativa ansats har jag valt en deltagande ansats, där jag försöker introducera interaktiv systemutveckling i olika organisationer i syfte att vinna djupare kunskap om processaspekter vid interaktiv systemutveckling. Jag har i dessa fall varit en form av förändringsagent, som har initierat en förändring och sedan låtit användarna genomföra den.

I denna avhandling rapporteras erfarenheter från 19 stycken personer som använt sig av applikationsgeneratorer i sitt arbete. Detta är underlaget för den explorativa delen. I den andra, deltagande delen, rapporterar jag erfarenheter från 9 st experiment med mer eller mindre långt genomförda interaktiva systemutvecklingsprojekt.

6.2 Obekräftade hypoteser

Nedanstående är en uppräknig av tänkbara hypoteser jag fått uppslag till i mitt arbete. I en del fall har jag kommenterat dem, men många gånger avstår jag. De är inte givna i någon speciell ordning även om jag försökt gruppera sådana jag tycker behandlar samma sak. En annan sak jag föregriper är ett successivt utbyte av termen "interaktiv systemutveckling" mot "personlig databehandling". Orsaken härför är att utveckling och användning vid detta arbetssätt sammansmälter. Därför tycker jag det är mindre lämpligt att beteckna det med en term som direkt är hänförlig till utveckling. "Personlig databehandling" är mera neutralt och ansluter dessutom till den internationella termen "end user computing".

- *De enskilda personerna använder personlig databehandling som stöd i sitt arbete.*

Det viktiga är stöd. Det innebär att personen har fullständig kontroll över detta stöd och själv bestämmer om han vill använda det eller låta bli.

- *För att själv kunna utforma sin personliga databehandling behöver en person både teknisk handhavandekunskap och modellkunskap.*

Med modellkunskap avser jag här kunskap om tillgängliga data och deras inbördes sammanhang. En förutsättning för att denna kunskap ska kunna användas på ett vettigt sätt är att personen har god verksamhetskunskap. Sådan kunskap behövs för att den enskilde personen ska kunna göra sig en modell av företaget och känna till dess begränsningar.

- *I samband med personlig databehandling finns en användningsnivå mellan körning av färdiga program och programmering.*

Jag tänker främst på sökning i register, framställning av rapporter och upprättande av kalkyler. För detta finns frågespråk mot databaser och kalkylspråk.

- *Användaren anser inte att kostnaden för den personliga databehandlingen spelar någon större roll.*

Detta gäller givetvis upp till en viss gräns. Hittills har personlig databehandling, åtminstone i större företag, varit något av en försöksverksamhet och har därför inte alltid debiterats ut till fullo.

- *Vid personlig databehandling används i huvudsak 10-15 olika kommandon.*

Liknande fenomen har också visat sig i användningen av andra system, t ex KOM-systemet. I en intervju i Datavärlden (Sandén, 1985) berättar dess skapare, Torgny Tholerus, om att i huvudsak 10-15 av de över hundra tillgängliga kommandona används. Liknande fenomen rapporterar Glimell (1975) från ett försök där användare fick välja mellan ett stort antal rapporter.

- *Vid personlig databehandling används flera redskap och inte bara ett enda.*
- *Vid personlig databehandling används företrädesvis kombinationerna Registerhantering-Grafik eller Kalkyler-Grafik.*

Denna hypotes ska ses som en ytterligare precisering av den förra. Jag menar att det i huvudsak rör sig om de ovan angivna kombinationerna. Därmed inte sagt att man inte alls använder t ex ordbehandling, memo-system eller alla tre i kombination.

- *Utbildning i personlig databehandling sker genom trial-and-error på eget initiativ av användaren.*

Det innebär att man ska avsätta *tid* för utbildning snarare än särskilda kurser.

- *Användarna upplever färre begränsningar i redskapen som används för personlig databehandling än vad dataexperterna gör.*

Detta kan ha sin naturliga förklaring i att dataexperterna (då främst personal på Inofomation Center) håller på att testa och utvärdera nya redskap.

- *För registerhantering inom personlig databehandling är en gemensam databas viktig. Detta är inte fallet vid kalkyleringstillämpningar.*

Registerhantering innebär att handläggaren hämtar information ur olika dataregister. Denna information används i hans arbete för t ex konstnads-kalkyler, leverantörsstatistik, försäljningsstatistik m .m.

- *Den enskilde handläggaren börjar med personlig databehandling av en tillfällighet.*

Detta ska jämföras med att utbildningen sker genom trial-and-error. Båda grundar sig på samma fenomen. Däremot introduceras personlig databehandling i *organisationen* inte av en slump! Förekomsten av Information Center och de anvisningar för dess skapande som finns (IBM, 1982) indikerar att det finns en tanke bakom.

- *Det behöver inte bli konflikter med dataavdelningen på företaget.*

Personer med anknytning till olika dataavdelningar hävdar, både i pressen (t ex Ahrnell, 1983) och mera informellt att sådan motsättning existerar. Men ingenstans i mina undersökningar kan jag finna något som talar för detta! Men de omfattar å andra sidan endast 8 företag med egna dataavdelningar. Därför är hypotesen relativt svagt formulerad. Med dataavdelning avser jag här vidare den vanliga dataavdelningen, dvs Information Center räknas inte dit.

- *Information Center eller liknande är en förutsättning för personlig databehandling i större företag.*

- *Att få igång personlig databehandling utan stöd från Information Center tar mycket lång kalendertid.*

Dessa två hypoteser verkar motsägande. Mindre företag har i regel inte möjlighet att etablera Information Center. Där är då den sista hypotesen möjlig att tillämpa. Alternativt kan man köpa tjänsterna från servicebyråer (Berg & Mattsson, 1986)

- *Den systemutvecklingsmodell som i praktiken används för personlig databehandling är "gradvis framväxt och förändring".*

Den strikta skillnaden mellan utveckling och användning är upphävd. Det rör sig mera om kontinuerliga förändringar.

- *Vinsten med personlig databehandling uppkommer inte genom effektivisering av rutinarbete. Det är genom nya möjligheter till beräkningar, sammanställningar och presentationer som de stora vinsterna görs.*

Jag har i mitt intervjumaterial dels rena exempel på kostnadsminskningar vid framställning av overhead-bilder, dels sådana tillfälligheter som inköpschefens kurva, för vilken jag redogjorde i kapitel två. I viss mening kan man förstås säga att de nya möjligheterna uppkommer tack vare en effektivisering av rutinberäkningar. Tack vare detta uppnås kvalitativa förbättringar.

- *De som sysslar med personlig databehandling är i stora organisationer handläggare på mellannivå, i mindre organisationer chefen själv.*

Det hävdas ibland att cheferna skulle ta över handläggarnas jobb med hjälp av personlig databehandling. Så har inte blivit fallet. Av detta följer:

- *Personlig databehandling tar inte bort några arbetstillfällen.*

Det är snarare tvärtom, för jag har vid åtminstone två tillfällen sett tjänster som inrättats direkt för en mellanhand.

- *Man jobbar vid personlig databehandling med kopior av produktionsregistren.*

Det beror på att redskapen kräver speciella register. En tänkbar orsak härtill kan vara säkerhetsskäl. Men numera finns applikationsgeneratorer som klarar av att jobba direkt mot produktionsregistren.

- *Vid personlig databehandling ökar arbetsmängden men trots detta blir kvaliteten på utfört arbete högre, både innehållsmässigt och presentationsmässigt.*

- *Applikationsgeneratorerna erbjuder vissa standardsätt att arbeta på. Detta påverkar användarnas sätt att arbeta.*

Även om det i princip skulle vara möjligt att anpassa applikationsgeneratoren till användarens tidigare sätt att arbeta väljer man ofta standardsättet. Detta medför:

- *Personlig databehandling med hjälp av applikationsgeneratorer kan medföra en viss standardisering av arbetet.*

- *Det tar inte speciellt lång tid att lära sig använda en applikationsgenerator.*

En förutsättning är att användaren är en viss typ av människa och har tidigare datorvana. Men jag kan inte specificera hur denna "vissa typ av människa" ska vara. Det närmaste jag kan komma är att vederbörande ska inte vara rädd för att experimentera och mer eller mindre hejdlöst kasta sig över applikationsgeneratoren och trycka på alla knappar och prova alla kommandon.

- *Handböcker till applikationsgeneratorer uppfattas som röriga och svåröverskådliga. Däremot uppskattas on-line hjälp.*

Det kan bero på att de personer, som "handlöst kastar sig över apparaten" inte är speciellt intresserade av att läsa handböcker.

- *Yrkesspråket förändras inte särskilt mycket vid personlig databehandling.*

- *De som sysslar med personlig databehandling känner sig uppskattade och får större inflytande.*

- *Personlig databehandling leder inte till ökad styrning och övervakning.*

Detta är en direkt följd av att personlig databehandling är stöd åt användaren i dennes arbete.

- *Kontakten mellan personalen ökar vid personlig databehandling om det finns en lokal dataexpert.*

Om alla har varsin dator kan man gissa att det omvända förhållandet skulle kunna gälla.

- *Att skicka meddelanden via datasystemet uppskattas.*

Detta kan möjligen ersätta minskade sociala kontakter, men det blir lätt ganska konstlade kontakter.

- *Skyddet mot obehörig åtkomst är inte speciellt utbyggt vid personlig databehandling.*

Möjligen kan detta vara bättre i de nyare typerna av applikationsgeneratorer.

- *Datapersonal besöker sällan de som sysslar med personlig databehandling.*

Detta kan förklara varför Information Center slipper göra färdiga tillämpningar, något som man klagat mycket över i Leitheser & Wetherbe (1984a).

- *Väntetiderna vid personlig databehandling är överlag långa, men accepteras.*

Det beror förmodligen på att det inte är tidskritiska behandlingar på samma sätt som t ex vid dataregistrering.

- *Att initiera personlig databehandling i form av en allmän datakurs i syfte att medvetandegöra personalen går bra.*

Det medvetandegörande syftet är väsentligt. Annars är risken att det blir mer eller mindre indoktrinering.

- *Det är möjligt för en grupp användare att med viss handledning under 3-4 timmar konstruera en relativt komplex gemensam databas i tredje normalformen.*

Handledningen består mest i undervisning av vad tredje normalformen innebär. Vad komplexiteten beträffar har jag alldeles för lite underlag för att ange några kvantitativa uppskattningar, men i kap fem finns exempel på en relativt komplex databas i samband med lönesystemet.

- *Vid start av personlig databehandling utan befintlig gemensam databas kan traditionell utvecklingsmetodik användas fast på en grov nivå.*

I regel tar det mindre än 12 timmar att gå genom alla faserna.

- *Att använda traditionell kravspecifikation vid personlig databehandling är oftast inte möjligt. Däremot är prototyping möjlig.*

6.2.1 Teserna

I detta sammanhang kan jag också ta upp de teser jag tidigare (i kap 3) diskuterade. I mina undersökningar har jag upptäckt saker som tyder på att nedanstående teser kanske skulle kunna verifieras.

1. *Det finns kunskap, som är av betydelse för verksamheten, men som inte går att formulera så att den är möjlig att överföra till datasystem (tyst kunskap)*

Denna tes har jag inte systematiskt försökt få bekräftad eller falsifierad, men en rad händelser (t ex omöjligheten att göra kravspecifikationer på Företaget) har inträffat som styrker mig i att den gäller.

3. *Tyst kunskap visar sig bl a genom kommunikationsproblem mellan systemutvecklare och systemanvändare.*

Mina svårigheter att förstå hur Företagets ekonomisystem fungerar visar klart att det är så. Men inte heller denna tes har varit föremål för systematiska undersökningar.

4. *Traditionell systemutveckling tar ingen hänsyn till tyst kunskap och det visar sig genom dålig anpassning av datasystemet till befintlig verksamhet.*

Detta är en hopsmältning av teserna två och fyra. Inte heller denna tes har varit någon av mina huvudteser, men de inblickar jag fått i befintliga datasystem visar på en, om än inte dålig, så åtminstone bristande anpassning. Framförallt innehåller systemen gamla delar som inte används, t ex termer i posterna.

5. *Ökad grad av användarmedverkan gör att problemen som följer av tes (1) reduceras.*

Mina undersökningar indikerar detta. I de system som användarna själva gjort har det inte funnits några tecken på sådana problem.

6. *Maximal användarmedverkan innebär att användarna själva gör sina system.*

Att detta är möjligt står helt klart. Förmodligen kan inte vilket system som helst göras av användarna. Avgränsar vi oss till personlig databehandling är utsagan i det närmaste en truism!

7. *Tillgång till applikationsgeneratorer är en nödvändig men icke tillräcklig förutsättning för att användarna själva ska kunna göra sina datasystem.*

Detta är en av mina huvudteser. Samtliga personer, både direkt intervjuade och indirekt omtalade har använt sig av applikationsgeneratorer i sin personliga databehandling. Eftersom direkt omtalade personer är ungefär 20 st och indirekt omtalade personer är över hundra anser jag att denna tes kan verifieras.

Ett bekymmer är naturligtvis att begreppet "applikationsgenerator" inte har någon allmänt accepterad och klar innebörd. I mina intervjuer figurerar personer som anser att den enda applikationsgeneratören värd att kallas så heter APL. Jag har också sett

handläggare som i BASIC gjort ganska avancerade stödsystem. Men detta är undantag.

8. Tillgång till applikationsgeneratorer i en organisation gör att datasystem av användarna kan användas som hjälpmedel i helt nya tillämpningar.

Det hade naturligtvis rent teoretiskt varit möjligt att på traditionellt vis utveckla samtliga de system jag studerat, men det hade kostat så mycket att det praktiskt är helt omöjligt. Kalkylredskap gör också att helt nya tillämpningar kunnat tas ibruk.

9. Om en person i sin verksamhet använder lämpliga applikationsgeneratorer får detta positiva effekter på vederbörandes arbete.

Tesen är försiktigt formulerad. Jag skulle kunna skärpa den genom att säga att personen utan ökad stress och med bibehållen eller ökad arbetstillfredsställelse, gör mer arbete och med bättre kvalitet under samma tid. Mina undersökningar indikerar att så är fallet.

6.3 Förklaringsmönster

6.3.1 Fallstudierna

Jag har här valt att göra två typfall, med alla i någon mening typiska egenskaper som finns i mitt material. Det är inte något statistiskt genomsnitt utan snarare *mångsidigt* (Arbnor & Bjerke 1978). Orsaken till att jag beskriver två typfall är att stora och små företag i viktiga delar skiljer sig ifråga om personlig databehandling. Jag beskriver dock typfallen parallellt.

Företaget

Det ena företaget är ett stort och har haft datoriserad databehandling sedan början av 60-talet. Man har egen datacentral med minst en stordator. Till denna datacentral hör ett Information Center, vilket varit i gång sedan ungefär 1982. Orsakerna till att ett sådant bildades är flera. En är allt större krav på ad hoc-rapporter från användarna, en annan svårigheten att utnyttja de nya databaserna med de gamla rapportgeneratorerna, en tredje är att IBM och andra leverantörer lanserat redskap, applikationsgeneratorer, avsedda att användas av slutanvändarna direkt. En fjärde orsak, slutligen, är förekomsten av enkel och lätthanterlig programvara på persondatorer. Detta har lett till ett tryck på dataavdelningen att tillhandahålla liknande redskap på sina stordatorer.

Det andra företaget är ett litet företag, där chefen upptäckt vilken nytta han kan ha av datorer och skaffat en persondator som han nyttjar för skiftande ändamål, främst kalkylering.

Redskapen

Redskapen har i de större företagen valts av Information Center. Det rör sig om flera, mer eller mindre möjliga att integrera. Man har i regel tagit över de redskap IBM lanserade till Information Center, dvs de APL-baserade ADI, ADRS och Graphpac. Både ADI och ADRS är olika former av registerhantering och rapportgenerering. Däremot tycks det inte ha funnits något kalkylredskap, utan sådant har man själv försökt skaffa. Det har nämligen visat sig finnas stort behov för sådana redskap. När man haft persondatorer har Lotus 1-2-3 varit det vanligaste redskapet.

I de små företagen kan man inte se något mönster. Antalet fall är alldeles för få. Jag noterar att problem med redskapsanvändningen uppkommit i två fall, i ett medelstort företag och i ett litet. Det lilla företaget hade problem både med ordbehandlingsystemet och med själva applikationsgeneratoren. Men att från detta isolerade fall dra slutsatsen att små företag riskerar misslyckas med val av applikationsgenerator i större utsträckning än stora företag är väl djärvt!

Tillämpningarna

Ofta tillkommer de av en ren slump (fig 6.1). Användaren träffar någon från Information Center och man börjar av en slump diskutera ett problem han har. Personen från Information Center visar då på närmaste terminal hur problemet skulle kunna lösas. Användaren skaffar sig behörighet att använda Information Center redskapen och börjar köra. Han eller hon upptäcker nya tillämpningsområden och blir snabbt en flitig datoranvändare.

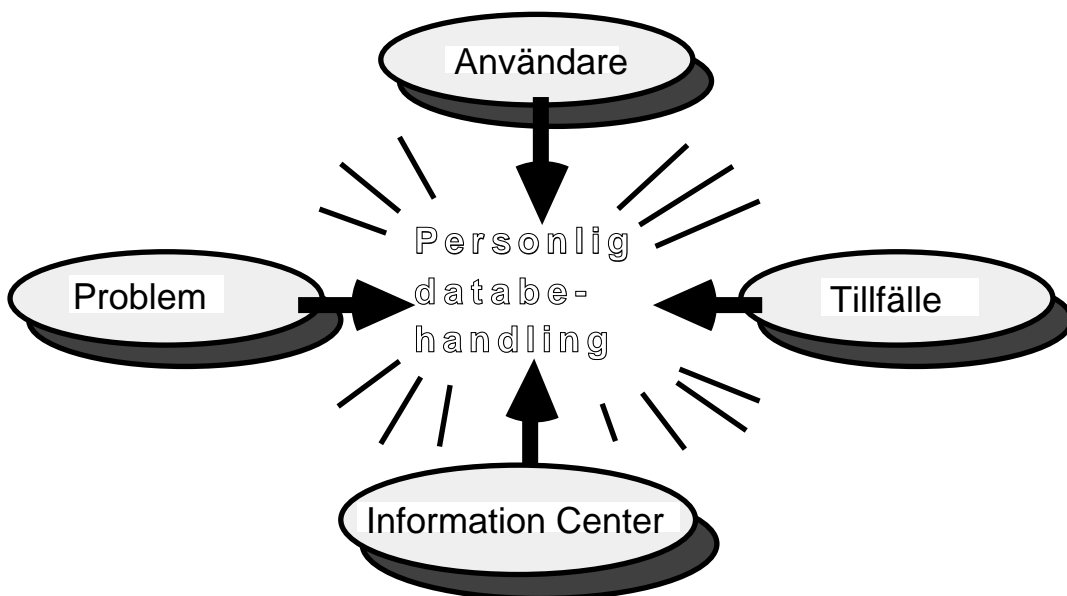


Fig 6.1. Vad som krävs för att användarutveckling ska börja.

Däremot tycks inte någon större rekrytering ske via kurser. Det kan bero på att man där inte har något aktuellt problem eller på att få av deltagarna är i en sådan position att de själva kan bestämma att börja med personlig databehandling.

Utbildningen verkar vara den enklast tänkbara, nämligen först visning och sedan egna försök ("trial and error"). Det medför mycket experimenterande och lekande med applikationsgeneratorerna. Men efter några månader har man lärt sig använda dem och övergår till att bruka dem som hjälpmedel och inget annat.

Det finns också tjänster inrättade speciellt för personlig databehandling. Det är i regel en tidigare befattning som kontaktperson med dataavdelningen, som ombildas till en sorts lokal expert. I båda fallen blir vederbörande mellanhand, dvs han eller hon hjälper, vid sidan om sitt ordinarie jobb, de övriga på avdelningen att ta fram information ur datasystemet. Orsaken till att denna person behövs är att det krävs ständigt aktualiserad kunskap om både applikationsgenerator och det system (dvs databas och färdiga, lagrade program och förmåga att uttrycka en fråga i termer hämtade från applikationsgeneratorerna) som finns tillgängligt. Exempel på tillämpningar är specifika utdrag ur ekonomiska register, budgetuppföljning samt olika sorters kalkyler. Omfattande beräkningar tycks vara en vanligare orsak till personlig databehandling än databashantering. Det tycks också vara så att databashantering sköts av specialpersonal, medan kalkyler görs av vem som helst.

I de små företagen tycks registerhantering överväga. Där är också något mera rutinartade tillämpningar i och med att det inte från början finns någon gemensam databas. Det kan då bli så att samma register ska användas både av företagsledaren och ett antal andra användare. Med en enda dator tillgänglig kan det bli åtkomstproblem. Utbildningen sker även i detta fall genom "trial-and-error".

Effekter på arbete och annat

Användning av applikationsgeneratorer medför att den enskilde tjänstemannen kan göra mer arbete på samma tid. Vederbörande tycker också att arbetet blir av högre kvalitet än tidigare. Kalkyler och prognoser, som det tidigare inte fanns tid att göra, hinns nu med. Resultaten presenteras i överskådliga, grafiska beskrivningar istället för omfattande tabeller. De som använder applikationsgeneratorerna känner sig mer uppskattade och får större inflytande.

När det ges bättre tid till uppföljningar kan det bli en ökad press på gruppchefer och liknande. Likaså kan mera kalkyleringsarbete leda till bättre beslutsunderlag och t ex bättre inköp. Detta är effekter, som kommer övriga organisationen till godo. Vad gruppcheferna tycker om den ökade pressen vet jag dock inte.

Den sociala samvaron har snarare ökat än minskat sedan applikationsgeneratorerna togs i bruk. Mellanhanden får ju hela tiden hjälpa de övriga användarna. Organisatorisk ställning för mellanhanden är "snäppet under avdelningschef". Det finns också andra användare på avdelningen vilka börjat använda applikationsgeneratorerna

själva, fast bara för mycket enkla tillämpningar. Även de får hjälp av mellanhanden. Jag har inte sparat något motsatsförhållande eller maktkamp dem emellan, men å andra sidan har jag bara pratat med mellanhänderna.

Bruket av applikationsgeneratorer och användarutveckling har, förutom inrättandet av Information Center, inte påverkat dataavdelningen nämnvärt. Det har inte förekommit några motsättningar, även om det ibland varit något trögt att få den ordinarie dataavdelningen att från produktionsregistren uppdatera även de register som används i personlig databehandling.

Det har inte heller blivit ökad jäkt och stress. Detta hänger till stor del hop med att de som använder applikationsgeneratorerna inte är styrda av dem. I själva verket upptar arbetet med datasystemen en förhållandevis liten del av deras arbete. Ett stressmoment skulle kunna vara långa väntetider. Det finns de som har väntetider på upp till 15 minuter! Men så tycks inte vara fallet. Tydligt har man lagt upp det övriga arbetet så att väntetiderna inte ska bli alltför påfrestande. Det torde väl också vara så att man i en kalkylering har lättare för att acceptera långa väntetider än man har vid t ex orderregistrering. Ett annat stressmoment är när den centrala datorn går ner eller något annat inträffar, som gör att man blir av med det man dittills gjort. I dagens läge kan man klara sig, tack vare att användarna fortfarande kommer ihåg det gamla arbetssättet, men hur blir det om några år, då man är helt beroende av datasystemet? Det kan t ex medföra att för ett beslut alldeles oombärlig information inte är åtkomlig och att företaget går miste om t ex en viktig order. Detta problem verkar ingen ha tänkt på.

I de små företagen gäller liknande förhållanden. Användaren tycker att han gör ett bättre jobb, men de sociala relationerna har blivit mindre påverkade än i större företag. Det beror på att chefen här är i stort sett ensam användare. Någon dataavdelning finns inte att ta hänsyn till och inte heller har det blivit ökad jäkt eller stress.

Säkerhetsåtgärder

Dessa är i det närmaste obefintliga. För att hindra obehörig åtkomst finns ett password och en sign-on kod, men för registren finns i regel inga spärrar. Har man väl kommit in i systemet är det fritt fram. Samtidigt klagar användarna på att det är för besvärligt att logga in sig i systemet. Som en av användarna kallt konstaterade, "...i ADI har man inga hemligheter!"

I de små företagen finns inte ens lösenord. Men där har man istället ett visst skydd i att övrig personal inte känner till vare sig redskap eller vilken information som är lagrad.

Övriga redskap och kostnader

Information Center tillhandahåller förutom applikationsgeneratorer även andra redskap för personlig databehandling som t ex ordbehandling, konferenssystem, Basic och möjlighet att rita OH-bilder. Detta medför att en del funktioner som annars skulle ha utförts av sekreterare istället utförs av sakbehandlaren. På sikt kan detta medföra en förskjutning av sekreterarnas arbetsuppgifter.

Kostnaderna för användarna kan upplevas som betungande. Men i regel kör man ohämmat. Information Center försöker sätta taxan så lågt som möjligt, men beroende på företagets policy, kan den ändå bli för hög. Dock anser användarna i regel att det betalar sig ändå. Men ingen tycks ha gjort någon utvärdering. Den huvudsakliga intäkten är inte genom effektivisering av rutinarbete utan genom möjlighet att göra analyser, som man inte kunde göra innan.

I de små företagen är användandet sedan man väl skaffat dator och redskap i stort sett gratis. Det största hindret för ökad användning är tidsbrist. I regel har man på de mindre företagen full uppsättning redskap, dvs registerhantering, kalkyler och ordbehandling.

Framtidsplaner

De applikationsgeneratorer jag studerat annonserades i början av 80-talet. Nu finns nya och bättre redskap. Speciellt för persondatorerna har det kommit fram mycket god programvara. Det är därför naturligt att man inom Information Center sneglar på dessa redskap och gärna vill förnya sig. Ett önskemål som framförs är möjligheten till integrering, att ta med sig resultat från t ex databashantering till kalkyler och sedan ta alltihop med sig till ordbehandling för att göra en rapport. Man hade själv lyckats integrera dessa funktionerna på ett par Information Center, men var intresserade att öka den ännu mer.

De små företagen hade inget mönster i sina framtidsplaner. Ett av dem önskade fler-användarsystem, ett annat önskade bättre fungerande utrustning. "Utbyggnad" kan möjligen ses som någon mera generell framtidsplan.

Stordator som persondator

På de stora företagen använde man i regel stordatorer. Det har fördelen att det finns viss kommunikationsmöjlighet mellan de olika användarna. T ex kan man skicka resultatet av en bearbetning till någon annans skrivare. Men samtidigt söker man i stordatorn skapa en miljö som påminner så mycket som möjligt om persondatorn. Det man främst uppskattar är stordatorns hastighet och möjlighet till kommunikation med andra tillämpningar, t ex en gemensam databas.

6.3.2 Övrig liknande forskning, validering

Här ska jag något diskutera frågan om dessa typfall kan anses i någon mening vara riktiga. Det rör sig inte om någon validering i statistisk bemärkning, utan om rimliga och förnuftiga skäl för och emot. Jag vill börja med att anknyta till kontorsautomation eftersom det även i kontorsautomation handlar om personlig databehandling, fast användarna möjligen har mindre möjlighet att själva påverka systemets utformning. En sak jag då noterar är den stora skillnaden mellan vad som står i reklamen och vad som är gällande i verkliga livet. I verkligheten är man långt efter. Detta intryck har bekräftats av flera andra (Asphjell 1984, Skjellanger 1984 och Skousen 1984).

Skousen (1984) redovisar forskningsresultat, som pekar på raka motsatsen till mina. Hans resultat rör dock ordbehandling utförd av sekreterare. Främsta orsakerna till missförhållandena är, enligt Skousen, bristande medinflytande på arbetsuppläggningsen och ökade krav från omgivningen. I de fall jag studerat har det varit stort, för att inte säga totalt medinflytande och några ökade krav från omgivningen har inte ställts. Så har jag inte heller noterat några av de negativa effekter Skousen talar om.

Skousen redovisar också tre olika steg i kontorsautomatiseringen:

1. Enskilda system
2. Nätverk
3. Systemintegrering

De tillämpningar jag studerat ligger någonstans mellan enskilda system och nätverk. Enligt Skousen kan större effekter väntas först i steg 2 och 3. Mina undersökningar visar inte på några större förändringar och bekräftas sålunda i detta avseende av Skousen.

Personerna på Information Center klagar också över att det tar lång tid att få ut användarutveckling i organisationen. Detta menar också Asphjell och Skjellanger. Asphjell menar att den långsamma ökningstakten till stor del kan förklaras av att den bromsas av "middle management". Avdelningschefer och direktionen ska enligt reklamen sitta med sina egna arbetsstationer och själva ta fram allt beslutsunderlag de vill ha. De handläggare vars huvudsakliga uppgifter är att ta fram sådant underlag skulle då bli arbetslösa. I de större organisationer jag och mina medarbetare studerat har samtliga chefer varit komplett ointresserade av datorer och har för övrigt inte haft tid att skaffa sig och vidmakthålla erforderlig kunskap om redskap och databaser. Detta leder till att handläggarna själva kan ta redskapen i bruk och därigenom öka sin service. Detta kan vara en förklaring till att jag inte sett några tecken till motstånd mot personlig databehandling. Dessutom tror jag Asphjell överskattar chefernas förmåga att hålla sig a jour med en omfattande datamodell över organisationen. Detta gör att handläggarna på mellannivån troligen kommer att finnas kvar under överskådlig tid framöver. Mina undersökningar skiljer sig alltså från Asphjells förväntan.

Ett annat sätt validera resultatet har varit att presentera typfallet för andra företag i vilka jag inte företagit någon undersökning. Detta har jag gjort på sammanlagt fyra konferenser. I samtliga fall har man känt igen sig och menat att det där typfallet är verkligen inget konstruerat fall, det är deras eget företag! Man har kanske inte riktigt tyckt att precis allt har stämt till hundra procent, men det har alltid varit några fenomen som har stämt precis på pricken. Jag har inte träffat någon som tagit avstånd och sagt att det där är helt felaktigt.

6.4 Specifika teorifragment

6.4.1 Att introducera personlig databehandling

Personlig databehandling är ett nytt fenomen. Att gamla metoder och tekniker inte är nödvändiga visar sig genom den slumpmässiga igångsättningen och det slumpmässiga handhavandet. Det förekommer således ingen planerad introduktion, ingen utvärdering och ingen egentlig utbildning. Jag skall nedan, med utgångspunkt i de kurser jag hållit, försöka skissa ett mera systematiskt tillvägagångssätt för att introducera personlig databehandling.

Fig 6.2 beskriver en tänkbar minimmodell. De streckade pilarna representerar själva starten, det som i traditionell mening kallas för utveckling. De svarta pilarna representerar det som traditionellt kallas användning, men som här även innebär ändringar av datasystemet. Som synes innebär detta att samtliga utvecklingsfaser ingår i den dagliga användningen. Den strikta skillnaden mellan användning och utveckling är upphävd.

Utgångspunkten är att det finns ett antal personer på en avdelning i ett företag (eller eventuellt hela företaget, om det är litet), vilka blivit intresserade att använda datorer i sitt arbete. Totala gruppen är mellan tio och tjugo personer. Personernas datakunskaper är också obetydliga. Vidare bör man ha tillgång till en dator med lämplig applikationsgenerator. Slutligen finns en utomstående person, konsult, katalysator eller forskare som är pådrivande kraft och så att säga "håller i" projektet.

Det börjar med en allmän datakurs för de blivande användarna. Denna kurs ska ge förståelse för datorer och datasystems begränsningar och egenarter i stället för att i traditionell mening gå genom datorns olika delar, de olika faserna i systemutvecklingen etc. I stället för att ge en ingående beskrivning av ett datasystems olika komponenter skall användarna få klart för sig vad ett datasystem är ett specialfall av. Kursen är alltså medvetandegörande. Av psykologiska skäl måste man nog åtminstone kortfattat gå genom hur en dator fungerar och hur ett program fungerar. Av någon anledning kan man tänka sig att se på TV eller prata i telefon utan att veta hur dessa apparater fungerar, men att använda en dator utan denna kunskap tycks vara värre. De klassiska sätten att introducera datautbildning är genom binära talsystemet och kontorsslagen med inoch utkorgarna. Jag menar att detta är att lura kursdeltagarna.

De har kommit dit för att lära sig hur en dator fungerar och får istället höra en massa snack om vanlig simpel kontorsteknik och skenbart meningslös sifferexercis! Därför brukar jag på mina kurser faktiskt gå ganska djupt in i det tekniska och försöka förklara hur en vanlig addition fungerar från den enskilda transistorn, via grinden, bus-sar till instruktionsavkodning med mikroprogram. Att gå genom detta tar vid tavlan c:a två timmar och då hoppas jag mina elever har fått ungefär samma vaga förståelsen som "folk i allmänhet" har för hur t ex bilar fungerar.

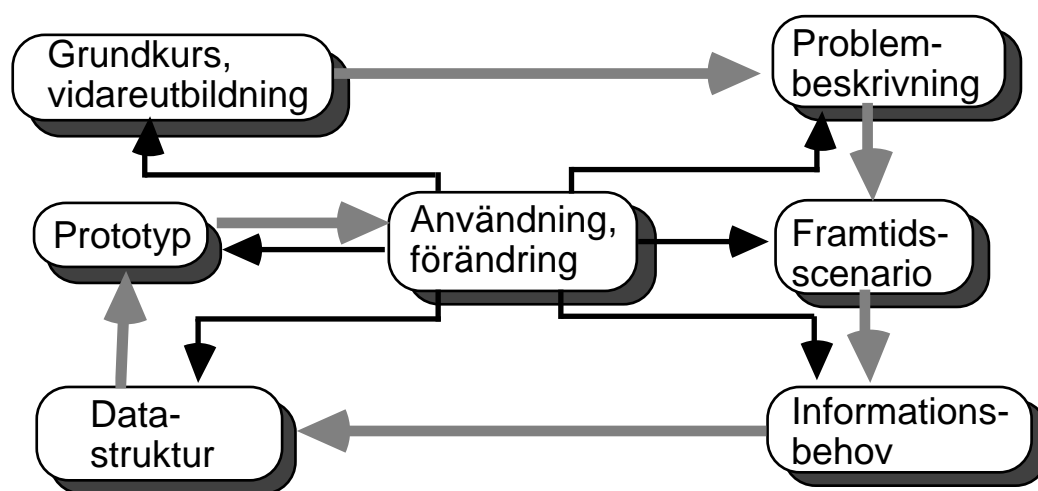


Fig 6.2 Tänkbar minimodell för användarutveckling

Kursen består av följande moment:

- Vad är datasystem specialfall av?
- Effekter av datasystem på arbete, frihet och demokrati.
- Genomgång av datasystemens tekniska delar.
- Praktisk övning genom att man gemensamt gör en systemskiss.

Ordningen har varit den ovan angivna, men jag tror det är bättre att starta med den tekniska genomgången. Det viktigaste momentet är emellertid det sista. Det är tänkt att man gemensamt ska göra en skiss till ett datasystem som kursdeltagarna kan ha nytta av i sitt arbete. Jag brukar följa en ganska traditionell metodik, fast jag gör det ganska snabbt och översiktligt. Följande steg tar jag upp, men med många iterationer:

- Problemformulering
- Målformulering, framtidsscenario
- Informationsbehov
- Strukturering av informationen (i tredje normalformen)
- Uppläggning av registerstrukturen på datorn.

- Prövning och testning av systemet.

Jag brukar under detta moment fungera som sekreterare och på tavlan skriva upp vad gruppen kommer fram till. Dessutom brukar jag använda mina egna ord för att därigenom kontrollera att jag förstått saker och ting riktigt. Om jag bara skriver ner med användarnas ord är det risk att jag inte rätt förstår och därigenom får problem. Risker är att jag förvanskar deras yrkesspråk, men erfarenhetsmässigt har vi efter några iterationer återgått till deras termer.

Jag brukar också kräva att alla som blir eller kan tänkas bli berörda av det blivande datasystemet ska delta i kursen. Detta kan ställa till problem för företaget om deltagarna samtidigt ska sköta sin ordinarie verksamhet. I något fall har därför kursen hållits på kvällstid, efter det ordinarie arbetets slut. I ett annat fall dubblerades kursen och hölls för halva arbetsstyrkan åt gången. Jag vill att alla ska vara med på kursen eftersom dess syfte är att medvetandegöra och inte bara att lära ut hur man handskas med en viss applikationsgenerator.

Själva kursen tar inte särskilt lång tid. Jag brukar räkna med åtta trettimmarspass. Men jag har en tendens att underskatta tiden för de sista momenten. Att lägga upp registerstrukturen, mata in en del data och arbeta med systemet tar för ovana personer ganska lång tid. 30-40 timmar är nog en mera realistisk beräkning.

Efter kursen är det meningen att användarna ska ha kommit igång så pass bra att de kan fortsätta själv. Möjligen behövs efter någon månad en fortsättningskurs som enbart behandlar möjligheterna hos den valda applikationsgeneratoren. Den skulle då lära användarna skriva mera avancerade procedurer, att ändra registerstruktur, lägga upp nya register etc. Denna kurs blir då mera som en traditionell programmeringskurs.

6.4.2 Krav på applikationsgeneratorer

I samtal med personer från dataavdelningar och Information Center har det framgått att det finns två typer av applikationsgeneratorer, nämligen de som är avsedda för användarna och de som är avsedda för dataexperter. Möjligen kan man hävda att det behövs särskilda applikationsgeneratorer för personlig databehandling, för rutinsystem (smör- och brödrutiner) och för administrativa kontrollsystem. Ser vi på de funktioner som finns i dagens applikationsgeneratorer och i vilka typer av system de kan tänkas användas får vi följande tabell:

Funktioner	Personlig databeh.	Rutinsystem	Kontrollsystem
Kalkyler	X		X
Databaser	X	X	X
Rapportgen.	X		X
Grafik	X		X
Procedurdef.	X	X	
Komm. med andra datasystem	X	X	

Tab 6.1: Funktioner hos applikationsgeneratorer

Vad denna tabell inte tar upp är behovet av helt nya funktioner. Dock tangerar kommunikationen med andra system detta. Jag avser här möjlighet att definiera en datamängd (t ex genom en rapportgenerator) och ange att dessa data ska sändas eller hämtas till/från en viss person via det gemensamma datanätet. Man kan tänka sig att det byggs upp lokala databaser på avdelningarna och att man där själv sköter både utveckling och underhåll. Detta ställer större krav på redskap och datanät än vad som möjligt att tillfredsställa idag. Framförallt är det inte ekonomiskt försvarbart.

En annan sak beträffande redskapen jag gärna skulle vilja ta upp är vad jag i brist på bättre kallar *sofistikationsnivå*. Schneider (1980) har varit inne på något liknande när han introducerar begreppet "user sophistication level". Han identifierar fyra olika nivåer: "Novice, Intermediate, Advanced and Expert". Järvinen (1985, 1986) och Järvinen m fl (1982) har vidareutvecklat dessa till "User Interface" som handlar om olika sätt att arbeta med datorn. I fig 6.3 presenterar jag en delvis omarbetat variant av deras bild. Ju högre upp på trappan man kommer, desto större frihet i interaktion.

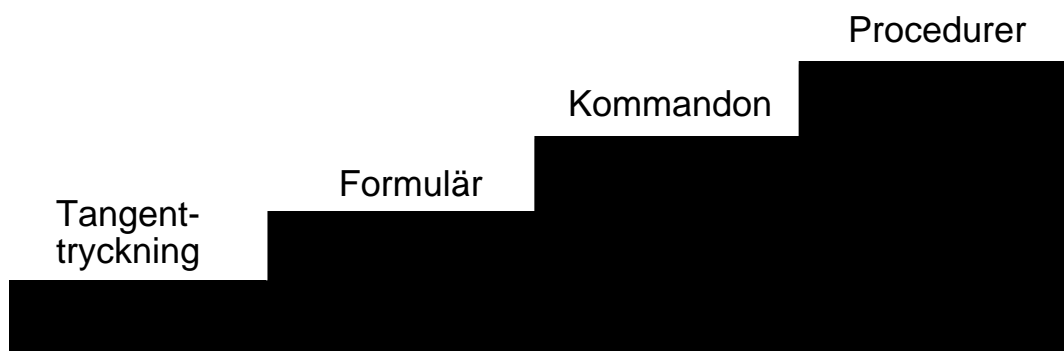


Fig 6.3 Olika interaktionsnivåer enligt Järvinen m fl.

Det lättaste sättet att interagera är genom enkla knapptryckningar. Det klassiska exemplet är menyer. På så vis förhindras en hel del misstag användaren annars skulle kunna ha gjort. Dessutom anses inte någon större utbildning behövas för att kunna

använda ett sådant system. Nästa steg är att fylla i formulär. Även här sker på samma sätt som i menyfallet en tämligen omfattande kontroll av inmatningen. Friheten består i att användaren i viss mån själv kan välja i vilken ordning han vill fylla i de olika fälten. Nästa nivå är att ge enstaka egna kommandon. Det är så många vanliga användare (dvs icke mellanhänder) i min undersökning använder applikationsgeneratorn. Sista nivån slutligen är att använda datasystemet för helt nya möjligheter, att helt och fullt ut behärska det. En del av mellanhänderna i min undersökning använder redskapet på detta sätt.

Emellertid kan man här lägga till ytterligare en dimension nämligen de möjligheter redskapet erbjuder. Detta kallar jag *redskapets sofistikonivå*. För att ta ordbehandling som exempel så har en radorienterad editor inte särskilt hög sofistikonivå, medan ordbehandlingsprogram där man på skärmen exakt ser hur den färdiga sidan kommer att se ut, har möjlighet till kolumnvis hantering, fotnoter, aritmetik m m har en hög sofistikonivå. Av egen erfarenhet vet jag dock att skillnaden, mätt i ökad produktion och ökad kvalitet på det skrivna inte är så stor som skillnaden mellan t ex en radeditor och Word-Star. Bland de redskap jag har träffat på i min undersökning finns det både mer och mindre sofistikerade redskap. Men de personer som använder dem tycker i båda fallen att de är bra. Det kan kanske förhålla sig så att "nyttan" man har av ett redskap inte hela tiden är proportionell mot hur pass sofistikerat detsamma är. Kanske nyttokurvan snarare ser ut som i fig 6.4. Då krävs det bara att man kommer över ett visst tröskelvärde för att få stor nytta. Mer sofistikon leder inte till lika stor nytta.

Nu kan man naturligtvis tänka sig att det finns flera olika trösklar och att helt nya möjligheter dyker upp. T ex när jag själv med hjälp av speciell programvara samtidigt kunde jobba med både skriv- och ritprogram på Macintosh, ökade mängden bilder i det jag skrev signifikant. Att nu återgå till det mera primitiva Word-Star eller ännu värre, en radorienterad editor, skulle upplevas som ett stort steg tillbaka. Jag tror man kan ha väldigt stor nytta, även av måttligt sofistikerade redskap, men att det är svårt att sänka sofistikonivån när man väl jobbat ett tag på en viss nivå.

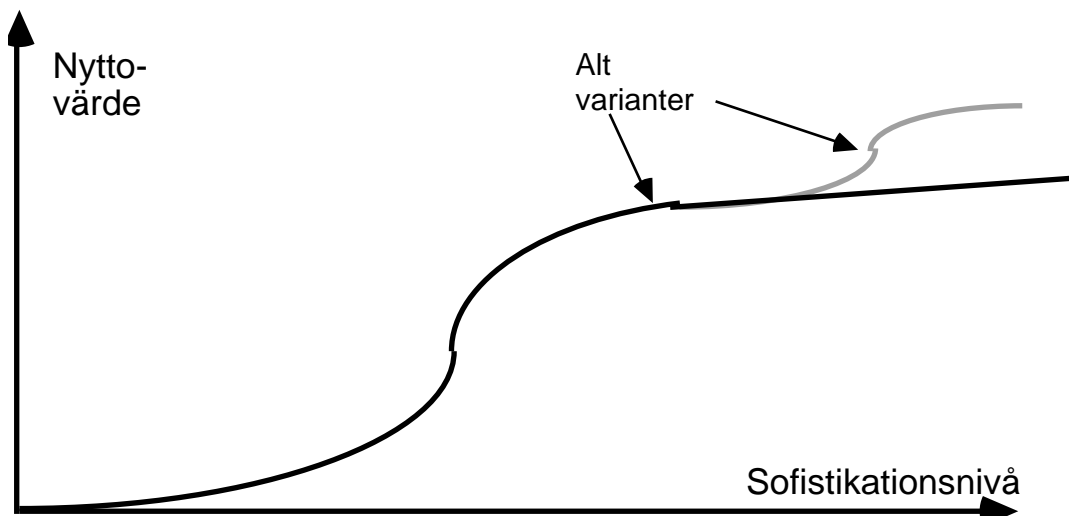


Fig 6.4 Nyttokurva för olika sofistiktionsnivåer.

6.4.3 Användarnas roller

Om den strikta åtskillnaden mellan utveckling och användning upphävs förändras därmed också användarnas roller. På användarsidan ligger traditionellt ett ansvar för att det som finns i systemets databas eller register är riktigt. Det formella ansvaret för indatas riktighet kan visserligen vara placerat hos någon som egentligen inte använder systemet. Det reella ansvaret ligger hos de personer som rapporterar indata. De kan vara helt andra människor än de som använder utdata. Att i ett sådant system åstadkomma en ökad integrering av utveckling och användning vara besvärligt. Bl a kan det vara svårt att motivera inrapporteringspersonal till noggrann inmatning om de inte klart ser sambandet mellan sitt arbete och organisationens arbete som helhet.

De flesta frågespråk mot databaser tillåter även användaren lägga upp egna register eller modifiera de existerande registren. I och med att vi nått detta stadium har vi påbörjat en integrering av utveckling och användning. Med hjälp av de moderna applikationsgeneratorerna är det möjligt än mer öka denna integrering. Programvaran är dessutom förhållandevis billig. I och med att man kan använda mikrodataorer så man har faktiskt i åtskilliga företag fört ut dessa redskap till användarna och tillåter dem jobba med sin egen systemutveckling/användning. Personlig databehandling uppkommer och den traditionella skillnaden mellan utvecklare och användare tenderar här att försvinna och begreppet "användare" blir oklart.

Utveckling görs traditionellt av dataexperter, systemerare och programmerare, i syfte att åstadkomma ett datasystem för användning av företagets personal. *Utvecklingsanvändare* är således dataexperter medan *användningsanvändare* eller kortare användare, inte är dataexperter. Användarna använder datasystemet som hjälpmedel eller redskap i sitt egentliga yrke. Dataexperternas yrke är utveckling av datasystem. I

fortsättningen undviker jag att kalla dataexperter för användare. Dataexpert och användare blir då två motpoler på samma skala. Skalans mätobjekt är kunskap om datasystem. Observera att när det gäller *användning* av datasystem har användarna ofta betydligt större kunskap än de dataexperter som gjort systemet ifråga.

I samband med personlig databehandling uppkommer en ny sorts användare, *mellanhanden*, med uppgift att hjälpa oerfarna användare att få ut önskad information ur datasystemet. Hjälpen kan bestå i att ta fram listor med hjälp av någon rapportgenerator, att använda något kalkylprogram för ett besvärligt problem eller kanske helt enkelt ta fram en snygg overheadbild över firmans likviditetsutveckling att presentera för VD på ett styrelsemöte. Detta är en av mellanhandens uppgifter och ingår i det ordinarie arbetet. Mellanhanden finns på användarsidan men har tagit ett steg på vägen mot dataexpert. På samma sätt finns det ofta *kontaktpersoner* från dataavdelningen som närmat sig användarna och ofta organisatoriskt tillhör användaravdelningen även om de i grund och botten är dataexperter. Vi får således en glidande skala från användare till dataexpert:



Denna skala är snarare kontinuerlig än en diskret uppdelning. Dataexpertens specifika kompetens kommer alltmer att omfatta utbildning och att hålla sig ajour med den tekniska utvecklingen.

6.5 Fundamentala teorifragment

6.5.1 Datasystem, informationssystem

Det kan i samband med personlig databehandling (inte bara på persondatorer) identifieras ett alternativt syfte med databehandling, nämligen som stöd åt enskilda personer i deras arbete. Ett traditionellt datasystem har i stället till huvudsyfte att kontrollera och styra verksamheten i en organisation. Detta har jag varit inne på tidigare i kapitel två. *Kontroll* och *stöd* blir således två olika syften med datasystem. I samband med dessa båda syften vill jag gärna ställa frågan om vad datasystem är specialfall av. Jag ser dem då som specialfall av kommunikation eller av arbete. Jag kan då konstruera en fyrfältstabell (fig 6.5) och skall nu försöka göra vad Törnebohm (1983, sid 244), med hänvisning till Weber kallar *idealtypskonstruktion*. Idealtypen är artificiell, en sammanställning av delar som finns var för sig, men inte förekommer tillsammans. Ett exempel på idealtyp är "renässansmänniskan", som har en mängd typiska drag och egenskaper, som knappast förekommit hos någon historisk person.

Vid traditionell systemutveckling (t ex om SIS/RAS tillämpas) är syftet att skapa ett styr- och kontrollsystem (Se kap 2). Meningen är att datasystemets användare ska utföra sitt arbete på ett i förväg bestämt och fastlagt sätt. Detta gäller både det manuella arbetet i direkt anslutning till datasystemet (t ex inmatning av data) och själva verksamheten. Planeringen av det arbete systemet kan utföra görs vid systemutvecklingen. Detta inkluderar även ändringar av vissa parametrar o dyl. Användarna och datasystemet betraktas gärna som ett samverkande system och användarna ses ofta "as components attached to the system". På så sätt betraktas användarna som betjänter till datasystemet (operatörer) med uppgift att göra de arbetsuppgifter, man inte lyckats datorisera. De negativa effekterna av detta tillvägagångssätt har länge påpekats av forskare med mera organisationsteoretisk inriktning (Bjørn-Andersen 1975, 1976, Høyer 1974, Hedberg & Jönsson 1976, Borum 1976 m fl). De datasystem som skapas i detta fall kallar jag *rutindatasystem* (det samma som basdatasystem enligt Brandinger & Norrby 1980).

"Fall av" Syfte	Kommunikation	Arbete
Kontrollera	IRM	Rutindatasystem
Stödja	Beslutsstöds-system	Human Scale Information Systems

Fig 6.5 Olika dimensioner hos datasystem

Om kontrollsytet bibehålls men datasystem istället betraktas som specialfall av kommunikation, blir förhållandet inte fullt så uppenbart som förra fallet. Det finns något som kallas IRM (Informations Resource Management, Svensson 1983, Swende 1980) och som innebär att information betraktas som en resurs och följaktligen är förknippat med kostnader och intäkter. Det behövs således planering och kontroll över tillgången på information och därmed också kommunikationen. Det kan innebära att det till en befattning knyts en viss informationsprofil. På så sätt sker genom ett kontrollerat informationsurval en styrning och kontroll av de som använder datasystemet. Detta har varit känt länge, redan H A Simon tog upp det (Simon, 1957). Goldkuhl och Lyytinen (1983) har visat hur man kan skapa ur användarsynpunkt dåliga datasystem genom att anlägga ett felaktigt kommunikationsperspektiv. Men om de som använder datasystem även får inflytande över dess information, vilket förordas av Swende (ibid), kan man uppnå stora fördelar med IRM.

Om istället syftet är stödjande men datasystem ses som specialfall av kommunikation har vi en typ av system som kallas beslutsstödsystem. Syftet med dessa är att förse

olika beslutsfattare med beslutsunderlag. Kring beslutsstödsystem finns en omfattande forskning. Det har gjorts många olika undersökningar (t ex Alter 1980, Bonzek et al 1981) av olika typer effekter, då mestadels ur ledningsperspektiv. Detta är också ett motiv för min avgränsning mot beslutsstödsystem sedda ur snäv synvinkel.

Om vi bibehåller det stödjande syftet, men istället ser datasystem som specialfall av arbete, får vi det som Nurminen (1982) kallar "Human Scale Information System". Nurminen påpekar att detta är en renodlad teoretisk konstruktion, som inte finns i verkligheten. Men de datasystem jag träffat på i mina undersökningar har passat in i denna kategori. De har utgjort stöd åt användarna i den viktiga bemärkelsen att stödet har definierats och utformats av dem själva. Det är inget stöd som har påförts utifrån utan de som vill ha stöd definierar själva vad som är stöd. Användarna har känt sig fria och obundna. Datasystemen kan vidare med fördel ses som specialfall av arbete, eftersom de utfört arbete, som användarna annars hade fått göra på annat sätt (t ex göra olika sorters kalkyler, göra olika rapportsammanställningar).

Jag medger att gränserna mellan de båda senare typerna än så länge ter sig flytande och oskarpa. Många av de system jag träffat på, kan med viss rätt betraktas som beslutsstödsystem. Därför är det i viss mening befogat att betrakta beslutsstödsystem och HIS-system som en och samma typ av datasystem, vilken jag i så fall kallar *stödsystem*.

Med ovanstående diskussion vill jag visa på olika *perspektiv* (Berglind, 1984) ur vilka datasystem kan betraktas. Det rör sig alltså om en renodling, en idealtypskonstruktion, för att därigenom framhålla väsentliga egenskaper. Samma arbetssätt har Nurminen (1982) tillämpat. Mitt bidrag här utgör kunskap om att HIS-perspektivet är möjligt att realisera.

Sammanfattningsvis har jag här renodlat tre olika typer av datasystem nämligen *rutindatasystem*, *kontrolldatasystem* och *stöddatasystem*. Men detta är en idealtypskonstruktion. I verkligheten innehåller varje datasystem lite grand av varje idealtyp. Det kan då beskrivas som en kropp ("potatis") i en tredimensionell rymd (fig 6.6). Kroppens relativa utsträckning och form indikerar då vilken huvudtyp av system det rör sig om.

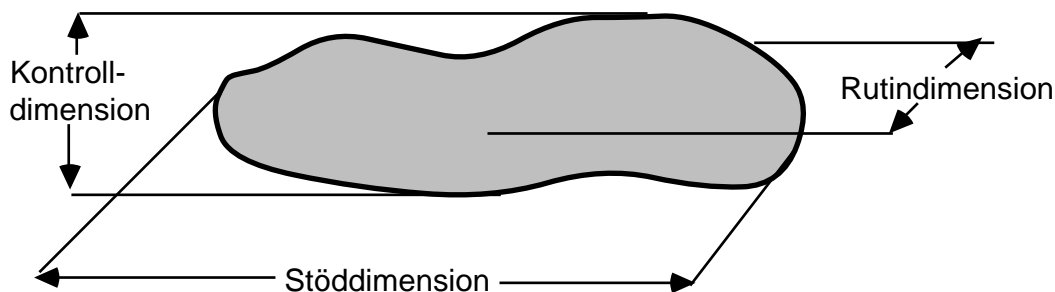


Fig 6.6 Ett ramverk för att beskriva olika typer av informationssystem

Det finns naturligtvis många andra perspektiv på datasystem man kan anlägga. Nissen diskuterar t ex (Nissen 1983) "Subject matter separability", som innebär en viss datamodells förmåga att skilja på olika aspekter av föremålet för systemutveckling. Järvinen redogör, som tidigare nämnts, för olika typer av användarinterface (Järvinen 1985, 1986 och Järvinen m fl 1982), vilket innebär användarnas möjlighet att samverka med det färdiga datasystemet. Mumford diskuterar arbetstillfredsställelse (Mumford 1983a,b) i samband med användning av datasystem, Sandström (1985) diskuterar vad de fenomen som uttrycks i ett informationssystem innebär för användarna, Nissen har i senare arbeten (1985b) betraktat informationssystem som specialfall av argumentation m m..

6.5.2 Olika typer av informationssystem

Holmberg har i Nissen et al 1982 beskrivit olika typer av information. Jag vill här tillämpa samma beskrivningssätt på kunskap (fig 6.7).

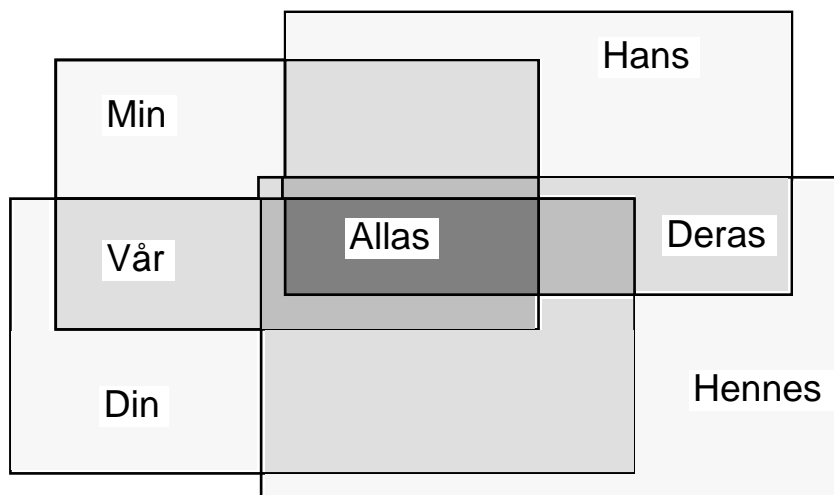


Fig 6.7 Olika typer av kunskap

Det innebär att det finns kunskap som bara berör en enskild person och dennes arbete (*min, din, hans* och *hennes* kunskap). Denna kunskap är "privat" och inte användbar för andra. Man kan också förmoda att den till stora delar är "tyst".

Emellertid förekommer det kommunikation mellan de enskilda personerna och den gemensamma grunden för kommunikation är kunskap som delas av två eller flera personer (*vår* och *deras* kunskap). Denna kunskap är meningslös för personer utanför gruppen och tycks ofta vara tämligen "meningslös" eller "egendomlig" för personer som inte tillhör gruppen (Sandström 1985). Eftersom kunskapen är gemensam är den delvis möjlig att formalisera och då delvis möjlig att behandla i ett datasystem.

Slutligen finns kunskap som är gemensam för alla i en organisation (*allas* kunskap). Ett datasystem som är gemensamt för hela organisationen innehåller uppgifter som finns i *allas* kunskap. Det är denna gemensamma kunskap som bildar grunden till det traditionella datasystemet.

Vid systemutveckling får de presumtiva användarna ange sitt informationsbehov. Givetvis anger de då delar av sin egen information. Systemutvecklaren begriper då inte allt, missuppfattar en del och lägger till en del ur egen fatatur (fig 6.8). På så sätt skapas ett gemensamt informationssystem som omfattar väsentligt mer än det borde göra. De enskilda användarna får anpassa sin egen information till en helt onödig gemensam standard. För organisationens förmåga att överleva stora och oförutsedda förändringar är det inte heller bra.

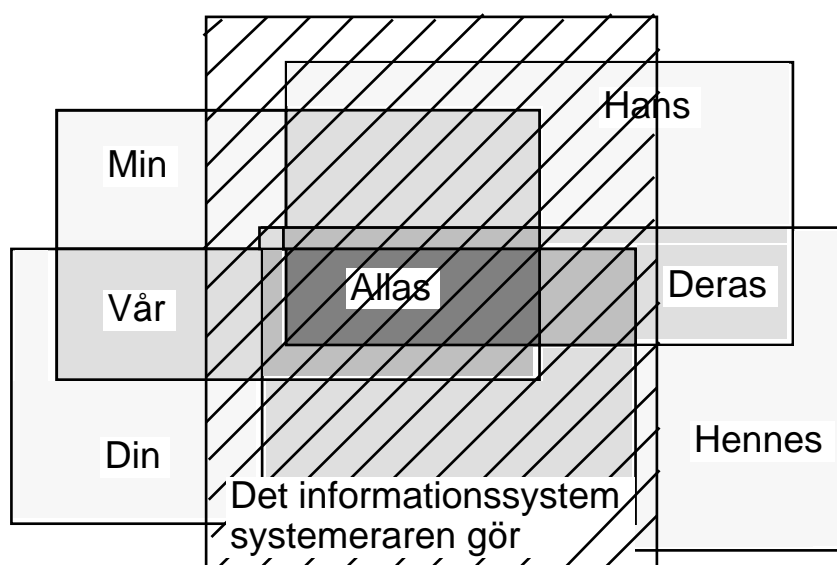


Fig 6.8 Vad som händer vid traditionell systemutveckling

Om nu istället en applikationsgenerator introduceras i företaget och enskilda personer får tillgång till denna för sin personliga databehandling, medför detta att databehandlingen kan ske på deras premisser, naturligtvis med den reservationen att det finns en omkringliggande organisation som sätter vissa gränser för handlingsfriheten. Följaktligen kan användarna nyttja datasystemet för de arbetsuppgifter det är lämpat för och hela databehandlingen ligger under deras kontroll. Ett sådant datasystem är ett typiskt stödsystem och bör inte utvecklas enligt traditionella systemutvecklingsmodeller. Detta har flera intervjupersoner påpekat, både bland användarna och på Information Center. Personlig databehandling är i själva verket ett helt annat fenomen än traditionell databehandling. Det möjliggörs tack vare applikationsgeneratorerna och kräver andra utvecklings- och användningsmodeller.

Givetvis innebär detta inte att den traditionella databehandlingen överges. Det finns fortfarande behov av gemensamma datasystem och företagsövergripande databaser. Men metoderna att utveckla dessa system kan möjligen förbättras i takt med att de blivande användarna bygger ut sin egen personliga databehandling.

En intressant fråga är om personlig databehandling även skulle kunna användas för rutindatasystem. Ett ordermottagningssystem kan ju mycket väl ses som ett stödssystem åt den enskilde ordermottagaren. I den mån en och samma person är ansvarig för hela arbetet från ordermottagning till leverans är det säkert möjligt. I regel är detta dock inte fallet och någon form av samarbete och koordination mellan skilda personer måste ske. I dagens applikationsgeneratorer finns inte någon sådan samarbetsmöjlighet, men tekniskt sett är det fullt möjligt att åstadkomma. Samordningen görs vid traditionell systemutveckling under utvecklingsprocessen och av systemutvecklaren i samarbete med representanter för de blivande användarna. Den blir därmed också tämligen rigid och oflexibel. Med hjälp av en applikationsgenerator, som har väl utbyggda kommunikationsmöjligheter, skulle de enskilda användarna själva kunna sköta, åtminstone en del av denna samordning. Den kan därmed bli flexibel och anpassningsbar.

6.6 Fortsatt forskning

De här redovisade undersökningarna har varit översiktliga och förberedande. Det faller sig därför naturligt att avsluta med förslag till fortsatt forskning inom området. Tidigare har en del uppgifter antytts.

Den första uppgiften innebär försök till verifiering av de i avsnitt 6.2 uppställda hypoteserna. Detta kräver fler och mer strukturerade undersökningar. Som ett första led i detta arbete kommer en ytterligare överarbetning av hypoteserna. Någon form av angelägenhetsgradering eller övergripande strukturering borde ingå i detta arbete. För att få ett tillräckligt stort och stringent undersökningsmaterial tror jag man måste använda någon form av enkätundersökning.

Man kan också tänka sig göra flera fallstudier liknande dem jag redan gjort. I så fall bör frågeschemat omarbetas och flera mindre företag medtagas. Det geografiska upptagningsområdet kan också utvidgas.

För att åstadkomma struktureringen av hypoteserna krävs något att strukturera efter, dvs den bakomliggande teorin bör artikuleras ytterligare. Detta är också ett arbete som behöver göras. De fragment som finns i denna skrift kan vara en första början till en sådan artikulering.

En annan uppgift är att följa upp företagen i den aktionsorienterade delen, både de där arbetet precis komit igång och de där det avslutats. Det är intressant att göra någon form av utvärdering efter något eller några år, för att se vad som har hänt och vilken effekt mina insatser fått. Denna utvärdering bör göras av andra forskare än mig. Givetvis bör det arbete, som redan påbörjats fullföljas.

Om man ser forskningen som ett samspel mellan induktion och deduktion kan man säga att jag här genomlöp en första induktiv fas. Nästa steg blir övervägande deduk-

tivt, dvs det gäller att förklara och föreslå åtgärder istället för att samla data och bilda teorier. Först med detta steg kan den fulla samhällsnyttan uppnås.

Ett annat intressant forskningsuppdrag är att spekulera över den ovan beskrivna "potatismodellen". Eftersom varje informationssystem i sig har någon stödkomponent, borde alltså någon del av av varje informationssystem kunna utvecklas genom interaktiv systemutveckling. Hur detta ska ske och kännetecknen på dessa delar är en viktig och intressant framtida forskningsuppgift.

7. SUMMARY

7.1 Background and problem

7.1.1 Work and tacit knowledge

Stig Lindholm (1979), a Swedish scientist, suggests when starting research that the following questions should be asked: "What type of phenomenon is this? What is it a special case of?". To this I have added a question of my own: "What does the special case consist of?"

In applying the questions to data systems, I have found two common answers. Firstly a data system is a special case of communication. This approach is taken by among others Langefors (1966), Nissen (1976, 1984), Goldkuhl & Lyytinen (1983) and Sandström (1985). It is a special case of communication, due to the fact that data systems transform the content of the message and store the messages over time. Nissen (1985a) goes one step further when he, based on Toulmin (1958), argues that data systems can be seen as a special case of argumentation.

The second answer is that a data system is a special case of work description. This approach is taken by Nurminen (1981), Göranson (1983a,b), Cooley (1980) and myself. It is a special case of work description because it is the complete formalization of it. The two answers are however complementary, not contradictory.

Thus I consider a data system as a complete and deterministic description of a certain work and the work will be performed when the system is working, i.e. when the programs are executed. A prerequisite for work to be described in programs is a totally predictive and deterministic work. The type of work that can be described this way are all types of mechanical work operations such as mass transactions, routine work and standardized operations.

Many of the shortcomings of data systems use are due to discrepancies between description of work and intended performance. In order to make a feasible data system, it must be decided if the actual work is possible to describe in a programmable way. Methods or techniques to achieve this are not available today.

What types of work can be described in such a way? The work of a conveyor belt surely can. But can for instance invoice work? Some manual work might be described in this deterministic way, but I do not think that very much office work can. This is due to the possibility of different interpretations of data. People performing office work are often themselves unaware of these different interpretations. Let me give an example:

As a systems consultant in an enterprise, I helped develop an invoice system. It worked with three files: a parts file, an invoice file and a customer file. Every customer was given a discount of a certain percentage on the total sum of every invoice. The people in the enterprise assured me that this percentage was fixed for every customer. It was decided therefore to supply this automatically by storing the percentage in the customer file. However, when the users saw the actual invoice they said: "This is wrong. The customer shall not have x % of discount, he should have y % in this case!" So the rules, which were supposed to be absolute with no exceptions were, when applied, found to be incorrect. In fact the system had to be changed so that the actual discount was specified by the user for every invoice and not performed automatically by the system.

The mysterious thing was that the users said the rules were absolute and obviously thought this was true. But when the rules were applied indiscriminately, they realized that some results became inadequate. The given rules were overridden by other rules, which the people only realized when they were violated.

Data systems can only take explicit rules into account. Thus data systems can correctly deal only with job steps that are guided by explicit rules. Work at a conveyor belt can be described that way, but not all office work. Office work is supporting human beings and since they are partly unpredictable, this work must have a certain degree of adaptation, that is, it has implicit rules. These rules being manifest as "management style", "organizational behavior" etc.

It seems as if this fairly obvious fact is often neglected in current research on information systems. Much research is devoted to the problem of constructing a technically efficient data system based on a given requirement specification. Also a lot of research is devoted to the problem of how to make the users of the system accept it and not complain too much about its shortcomings, but instead be happy about its merits. In fact almost every data system has a great merit in removing a lot of routine work. However, often other work that is yet more routinized is introduced. But the deep philosophical problem of what will happen when knowledge of a certain type is unconsciously replaced with knowledge of another type or simply removed is very seldom discussed.

7.1.2 The role of the users

Tacit knowledge not taken into account, when a data system is constructed, will cause "tacit misfits". The people best suited to detect and correct these misfits are the users. The correction often requires considerable changes in the data system. Due to the heavy backlog of most data departments, it often takes over a year before the changes are implemented (Martin 1982). This prolongs poor job performance, it causes a lot of problems for the users and the organization in adapting to a changeable environment. The cost of this inflexibility is often very high. The possibility for making rapid

changes is thus very important. The capacity for changing data systems must be increased. This can be done in three ways:

- By increasing the *productivity* of the system designers
- By increasing the *number* of system designers
- By both increasing the *productivity and number* of system designers.

The first one is possible since we today have powerful tools for speeding up the development. We have different types of structured design that might increase productivity by 10 to 20 % (Martin 1982). We also now have application generators or 4th generation languages that will typically reduce coding time 50 to 80 % (Ibid). Even non-data experts can use these tools, after a short introduction, to develop their data systems and thus achieve the third possibility. Today there exists a variety of such tools and since they have been in use for a number of years it would be an interesting research task to investigate the effects when using such a tool. This is my main problem. When people in an organization use application generators as support in their work I call this *personal data processing*.

7.2 Scientific approach

A traditional approach to such an investigation should be the verification of a set of hypotheses. This set must however be based on an understanding of the factors that are important. You do not get answers to questions that are not asked!

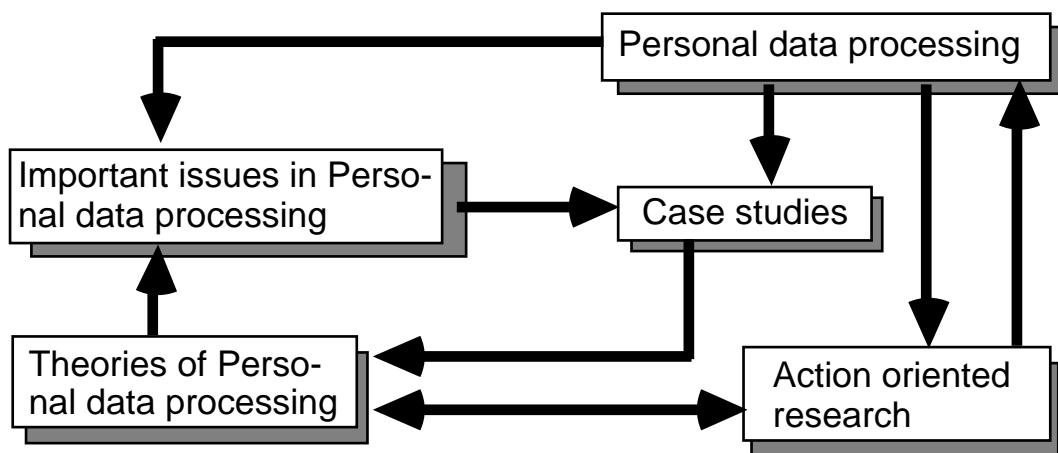


Fig 7.1 Relationships between the different parts of the research project.

The factors are parts of a theory, so we thus need a theory of user developed support systems in order to undertake a traditional scientific approach.

The whole research process can be described in fig 7.1, which is a modification of a similar figure in Checkland (1981, p 8). I suggest here two approaches in the study of personal data processing. The first one is based on collecting experiences from such

work and documenting them in case studies, the other one is based on taking an active part in such developments.

In the first approach I used a data collection technique called semistructured interviews that will be described later and in the active part of the investigation I used an action research oriented approach.

7.3 Case studies

Collecting experiences means in practice asking a lot of people about their experiences. I must have some ideas about what might be interesting but it is also important to allow people to talk about other things. Interviews with a few rather broad questions and possibilities of asking more detailed questions seemed to be an acceptable approach. Questionnaires would probably not be a valid technique here, since the purpose was to let the people concerned talk about their specific experiences. I thought case studies were a better way of reporting this type of research. Perhaps some common patterns might emerge and then, but not until then, the specific pattern could be investigated using harder data gathering techniques.

The problem was to find the broad questions. I could of course ask only one question: "Please tell me about your work with the computer." In fact I have started my interviews with exactly that question. Often people then talk about how it all began. However, I thought my interviews should reflect my understanding so far about the area of investigation. Therefore I put forward some areas I thought to be of interest, such as:

- The tools that are used
- The organization that uses the tool
- The historical background
- A description of present applications
- The effects on job satisfaction
- The organizational effects

These were the main areas I tried to cover in semistructured interviews with individual people. In order to make the so far very vague theory explicit, to both myself and other researchers helping me in the investigation, I developed a set of questions covering all six areas above. I had in total more than 80 questions that could be asked. However, and this is very important: *It was by no means my intention all these questions should be asked as a mechanical questionnaire!* When I made the interviews myself, I found that it is usually sufficient to ask the main question in each area, while the details concerning the other questions were answered without me having to ask them directly. I tried to maintain a conversation, rather than make an interview.

The data processing department was supposed to answer questions concerning the tools, at least the more technical ones. I thought, however, it was important to ask the users too, because in using the tools they might have gained this kind of knowledge, perhaps at a higher degree than people from the data processing department. It was also possible to compare the description of the use of a certain tool, first as the vendors described it, and second the description of its actual use.

Concerning the organization, that uses the tools, I have almost only considered fairly large organizations, which already had a working data processing department. The idea of user development had often come via the concept of the "information centre" as IBM (1982) suggested it. Mostly only a few people in the data processing department were working with user development. In small organizations it seemed to be only the manager that used the tools, mostly for calculation purposes. But I have only studied four such organizations.

The historical background was very important. It helped me better understand the phenomena I come across in the investigation. However, different people gave different interpretations of what happened and why it happened. I also came across some integrity problems, which of course were respected. Sometimes people did not know much about the story prior to their appointment, since they were employed to work with the computer system and thus they had not joined the organization until the system was in use. In this case I was forced to use second hand information.

I have used the framework of Mumford (1983a,b) concerning job satisfaction and asked people about effects on their job skill, their social relations, their psychological needs, their ethical dilemmas and their task structure.

An important organizational property was the possible support for decentralization in the organization. This could be one organizational effect. Another might be conflicts between data departments or user groups or information centre people.

The set of questions has been worked out in two steps. First two of my students made investigations in two companies, without any question set. Their conversations with the people in the companies were written down almost verbatim. I thus obtained a good impression of what took place. A certain pattern emerged: Historical description, actual application, properties of the tool and organizational effects were the main points. However, people did not say anything about the effects on job satisfaction. Perhaps they were unaware of them, so I supplied a set of questions relating to this area. Then I made a test interview myself and found the total set of questions useful. I also prepared a guide with a more detailed discussion of every question and explained my intention with the questions and the type of answers that I expected. I also tried to foresee the difficulties and identify possible ways of overcoming them. The purposes of this guide was twofold:

- To act as a guide for other researchers or students that were going to help me in the investigation.

- To document my own thoughts before the real data collection started.

The interviews were conducted using a tape recorder to avoid having to take notes which might have inhibited the interview. Thus only references to the prepared set of questions was necessary. On conclusion the interviewee was provided with the set of questions to check if any important or interesting questions remained unanswered.

Currently I have two levels of documentation, firstly the tape-recorded interviews and secondly the interviews as subsequently written up in a coherent form. This requires some interpretation of the collected material. A typical interview took about 1 to 1.5 hour and occupied about 8 to 9 full pages of text when written up. When I had conducted the interviews I tried to write them up immediately. Some interviews have been done by students or other researchers and they were unable to do it this way, for example have some interviews been done at companies far away, so people have had to stay at an hotel over night. The written up versions were sent back to the interviewed people for comments and corrections. Then I called back after about a week to check if my documentation was correct and to see if any of the set of questions were unanswered. When they had agreed the written interviews became an official version and could be distributed.

The material has to be confidential as some interviews reveal information of sensitive nature that cannot be published or alluded to. I thus have to write in such a way that the actual applications are only very briefly described and the company and the products are not mentioned by name. In doing so I hope to guarantee the integrity of the organizations involved.

7.3.1 Some findings

At present I have 19 interviews from 13 companies and some patterns are emerging. The first finding indicates the existence of the local experts, that is people who know how to use the tools and help other people in their work. These local experts work with the computer tools as a part of their job. They frequently have a relatively high position in the organization, working close to the head of the department or maybe the company. In one case the local expert was the head of the purchase department. They were all enthusiastic about the possibilities this new way of working provided. They often became more influential and felt more valued by the head of the department or the company.

Another finding is that they use more than one tool. Typically there were tools for data bases (query languages), calculations, graphics and word processing. The tools are supplied by an information center at the data processing department. At this center two or three persons work as advisors and teachers concerning the use of these tools. APL is often used as a host or base language. This reflects the fact that IBM uses this in their information center idea.

Irrespective of the tools they are using, the people were satisfied with the choice and the way of working with them. On the other hand, they do not seem to perform any deep evaluation either of tools or of suitable applications. The choice seems to be more or less randomly. No company have any well planned strategy for introducing this new way of working. But the people using the support systems are all convinced about the profits, both in money terms and particularly in terms of quality of work performance. They all think they could do a much better job than before and also do things they had not previously conceived of doing.

I have found it relatively easy to detect effects on job skills since the people, without exception so far, think the use of a user-developed support system has greatly improved their ability to do a good job. Concerning the other job satisfaction factors, no effects have been detected. Some people admit they have more influence due to their improved job performance, but that is all.

I also distinguish two or maybe three levels of using the tools. The occasional user who uses a prewritten procedure in order to get a report from the data base or some computations in the spreadsheet program or a picture plotted with the graphical system can illustrate the first level of use. (S)he also might use the command mode of the query language or the spread-sheet program in order to answer some ad hoc questions. The local expert, who can write procedures, create new files and help other people with more complex questions, can illustrate the second level of use. People from the information centre who help the local expert with the more complex jobs or when the system has broken down might illustrate the third level.

People use the support system to produce reports, which they are working with at their desks. The terminals are not necessarily located on the users desks, but can be found in another room, enabling more people to make use of it. However, people working with large calculations seem to require the terminal at the desk.

Response times fluctuate widely. I have seen applications with surprisingly quick response times but also applications where response time could be up to 15 minutes. As a rule, people think they are rather slow, but it is not considered to be a serious problem.

Knowledge of the data model (the system) is more important than knowledge of the tools. The data bases used by the companies are usually complex and require great skill from the users of the support systems to make effective use of them. Thus the local experts are experts both on the tool and on the data model. Perhaps they can be considered as local data base administrators.

7.3.2 Comments to the case studies

The purpose of the research was to get a brief overview of empirical experiences and to describe what people thought about their job with user developed systems. Some

common patterns emerged, and hypotheses were formulated (section 7.4.1). What remains is designing experiments or investigations in order to verify or falsify them.

Since the interviews were quite free (rather conversations), I hope the most important effects and experiences from the users' point of view, have emerged. However, people saw me as a computer expert and they often thought I required certain answers. In such cases they often unconsciously gave me just these answers.

Another important question is how do I know that my six main areas (p 191) are the important ones. They are only a structure. I tried to collect and report the data in such a way that I was not restricted to these areas. But of course you cannot get answers to questions not asked. Other types of investigations must thus supplement this investigation.

I have considered individuals all the time, not organizations. This orientation is perhaps disadvantageous but my reason is I think not many investigations that have been undertaken from the individual perspective. It is, however, possible to compare people working in the same organization and perhaps find some common patterns due to organizational factors.

7.4 The action oriented approach

The case studies describe the phenomena of user development from the observer's point of view. They do not provide any inside or in depth knowledge of effects that might occur during the implementation of user development in an organization. I think that process too must be documented, in order to avoid pitfalls and to gain knowledge about the user development process. The question for whom I gather this knowledge and which Nissen (1984) discusses, is in this case a very important question. In the case study part of the project, I achieve a rather generalized knowledge. In the second part of the project, which I call the action oriented approach; my role as a researcher is not so clear-cut. I can choose the role of an observer only and document what happens without any intervention. This is suitable when I know nothing about the problem. But in this case I do. I have gained some knowledge from the case study part of the project and I have a strong belief that the users ought to have great influence. I believe it is wrong and ethically unacceptable to act only as an observer and not try to help people involved in the development.

The possibilities for achieving generalised results are almost zero, but the possibilities of achieving valuable knowledge for the people involved in the project are enormous! I try to act as a change agent and obtain acceptance for the idea of user developed support systems in the organisation. Often I feel like a salesman of ideas. I believe my ideas to be good for the organisation, but cannot prove it.

Another problem is identifying what the result of the research actually is. Usually research reports are considered to be the main result. In action research projects, however, actions taken by participants in the project might be considered as results of the research. Nygaard (1974) has described this dilemma in the NJMF-project. However these actions must be described and reported somehow. I think it is a fair demand from my research colleagues. My research cannot be reproduced but it can be described although the description must not reveal any trade secrets or other sensitive information.

7.4.1 Some results

I will in this section describe the action research oriented cases.

The Company

The Company is a very old company. It was founded in 1797 but its roots are further back in the 16th century. The Company has had problems in the later years. Recently almost half of the white-collar staff has had to leave the Company. A whole division was sold. However, things have begun to change and the result of 1983 was good enough and for 1984 it was even better. The problem concerned an accounting system from which it was rather difficult to obtain desired information. The information existed in the system, but it was impossible to get at it. The system was run as a batch system and produced lots of lists. It was developed as an in house system by the data process department. It was a centralised system supplied with information from a couple of presystems. These presystems were run on different computers. The communication with the central IBM 4331 performed via exchange of diskettes and tapes.

In the spring of 1983 a project group was formed consisting of the present head of the data processing department, two people from the central finance staff and two researchers from the university of Lund. One of the latter was myself and I acted as the scientific project leader. The project leader for the Company was the head of the data processing department. This changed about Christmas 1983, when he left the Company and another person from the central finance department replaced him. Also the other researcher was unable to participate further.

The project started as an ordinary systems development project. In the fall 1983 we undertook a problem investigation and a brief description of the company. I tried in vain to understand their accounting system. The researchers and the data processing head tried to obtain a requirements specification from the other two finance people, but this was also in vain. At Christmas I suddenly realized I was acting as a traditional data processing expert and not as a user oriented data researcher! I made a description of my own point of departure from this role in a planning report. In the companies view this was an unusual and embarrassing report. I realized this and tried to adapt as much as possible to the behavior of the company. Thus I hoped to gain their confidence and perhaps acquire more favorable response for my own ideas.

After Christmas 1983 things began to work much better and it became clear that it was a support system we were going to develop, not a control system. I also realized it was impossible to produce a traditional requirements specification of the desired system. In fact this directly led me to develop the concepts of support and control systems and the realization of the impossibility of specifying office work.

I found four different ways of developing requirements specifications. The first and the easiest was to *identify the inadequacies of the present system*. We used this method at the beginning but not systematically. The second was the traditional one of asking what *demands and information were required of a future system*. This was, as already said, impossible. The third was to describe the work, then *analyse which activities could be performed by computers* and after that find what information was needed for these activities. I have in previous research projects (Flensburg 1981a) suggested a method for doing such an analysis. This was also impossible since the people were not able to describe their work beforehand. The fourth was to *find the data structure of the existing presystems and then determine the data needed*. This method was suggested by one of the users in the central finance department.

I tried making the future system visible for the users (and myself) by constructing some simple prototypes with DBase II on a micro computer. As a result of this, we obtained an apprehension of a possible requirements specification. I also ran a short course on different data models for people in the data processing department. Finally we managed to develop a requirements specification by looking at the data descriptions for the existing presystems and chose the data terms we wanted. I translated these to a data model in the third normal form. It was our intention to use this model as a test case when we visited vendors, selling tools for development of support systems. We visited some vendors and the tools were demonstrated. However, after a couple of demonstrations we were more confused than helped. It also turned out that the investment needed to be rather larger than anticipated and thus a decision needed be taken at a higher level in the organization. The Company has a policy of buying computers from only one vendor and we investigated the tools this vendor offered. However, none was suitable and we must turn to other vendors. After a while we found the Query Management Facility of IBM, that was built on the data base languages Sequel and Query By Example, as a suitable tool. An almost full-scale prototype was developed by the users and demonstrated to the top management. It was accepted and the system will be implemented in the beginning of 1986.

The library system

This was the very first user development project I was involved in. It started as an inquiry from the region library about help in investigating possibilities for development of computerised information systems for local collections, for the library of Lund and the library of the Malmö region. A group of students developed two prototypes for such systems (Axelsson et al 1982, Bertilsson et al 1982) and after that I

held an introductory course about use and development of data systems for about 20 librarians from the region. We applied for development money but this was denied. A group around the head of the region library, however, continued searching for possibilities, and after some time (in fact about three years) the proposed system "LIMES" was a fact (Mundt-Almquist et al 1985). I acted as a start up person, provided some education and then the librarians carried through the project all on their own.

The salary system

In another area of local government they intended to introduce a new salary system. Fortunately one of my fellow researchers of the LUIS-group (Siv Friis) became involved in that project at a very early stage. We have collaborated and held a course entitled: "How to put requirements on data systems" for all the people in the salary department. In the course we introduced systems development as an example from their own work. We developed a rough sketch of a data base model in the third normal form for their work within the course. However it was not possible to fulfill the implementation due to political reasons. It was decided by the politicians not to develop a new salary system.

The employment exchange

Since the year 1980 I have had contacts with the regional employment exchange in the Malmö region. A certain institute, the "labour market institute" served as a qualified analysis resource for the local employment exchanges concerning people that were hard to place on the labour market. For such an institute a similar course was held. We developed, as was the case for the local government, a data base model in the third normal form. This database concerned the courses given at a trade school for unemployed people. But it was not possible to implement this database either, because computer resources were not available. An effort with another database was also in vain because there were no people who had time to do the development.

Some other efforts

A similar course was held at a department at the hospital of Lund. At least three researchers were involved in this project. The nurses and two doctors were educated but it turned out that it were the doctors that needed a system. Also in this case we had no machine resources, but it seemed quite possible to obtain such resources in a couple of months. However, due to very heavy time restrictions both on the doctors and the researchers, we had no chance to do anything.

Two other cases concerned unions. Since I had learned the importance of having computers we decided to obtain such first and then start the development project. We are still waiting. Another case concerned a small company that needed a new book-keeping system. The old one was run on a mechanic machine, which was almost a scrap heap. A new micro was bought with a standard bookkeeping system and we decided to develop an invoice system after a while, when people had more time. This has not yet been the case.

7.4.2 Comments to the action oriented research

The great problem with this type of research is to keep the primary material to a manageable size. It is not possible to know, in advance, what is important or not. Everything can be important and thus perhaps I should have recorded everything that happened on videotapes. This would be extremely expensive and probably impossible to use. I did tape record (on usual tape, not videotape) certain meetings and documented them afterwards in written form. Some meetings were only documented in my own notes and some were not documented at all. An example of the latter being some visits to vendors for demonstration of user development tools. The written documents are distributed to the project members as "meeting notes". In some cases I have produced planning reports or other more traditional reports intended for use by the management of the company. I also kept a diary, where I put personal and speculative thoughts.

The criteria for failure or success are also very important. Quite by chance a good success criteria evolved. One of my fellow researchers complained about people not giving her credit for her ideas. She said: "I suggested something two months ago that they are now using and they do not even mention my name now, but act instead as if it were an idea of their own!" In my opinion this is a very good success criterion. We are not working with people in order to get credit for our ideas; we are working in order to help those people. If they take one of our ideas as their own it is surely a very positive indication of our success.

7.5 Theoretical results

According to Törnebohm (1983) the results will be given in four levels of knowledge development:

Level 0: Measured data

Summaries of those are given in chapters 4 and 5.

Level 1: Verified hypotheses

Since this is an explanatory investigation no such hypotheses are given. But in section 7.5.1 I will give a set of unverified hypotheses.

Level 2: Explanation patterns

Such patterns are given above in sections 7.3.1 and section 7.4.1

Level 3: Special theories

Contributions to such theories are given below in section 7.5.2

Level 4: Fundamental theories

Contributions to such theories are given below in section 7.5.3

7.5.1 Unverified hypotheses

Since my investigation was an explanatory one, I was not able to provide any verified hypotheses. But one goal of such an investigation is to provide hypotheses for later verification. The following hypotheses are the most important in a greater set I think have chances of being verified, though I for the moment can't say anything about how to do that.

- *In order to do personal data processing, knowledge of both the tools and the underlying data model and model of the work are needed.*
- *In personal data processing you use mainly about 10-15 different commands.*
- *In personal data processing you use a set of tools and not a single one.*
- *Education in personal data processing is through trail-and-error at the initiative of the user.*
- *The single manager starts with personal data processing quite by chance.*
- *Conflicts with the data department are not necessary to occur.*
- *An Information Center or similar is almost necessary for starting personal data processing in big companies.*
- *The great benefits with personal data processing are not the rationalisation of routine work, it is through new possibilities to computations, collections and presentations.*
- *People working with personal data processing are middle management in great organizations, top management in small organizations.*
- *The work load increases at personal data processing, but the users think the quality of work increase, both in content and in typography.*
- *Manuals for application generators are seen as messy and hard to overview. On the contrary, on-line help facilities are greatly appreciated.*
- *Sending messages via the data system is much appreciated.*
- *The protection against unauthorized access is not much developed in personal data processing.*
- *It is possible for a group of users to construct a rather complex data base in the third normal form during 3-4 hours with a certain amount of handholding.*

- *It is not possible to use traditional requirements specification in personal data processing. Prototyping might on the contrary be quite useful.*

7.5.2 Contributions to fundamental theories

A common objective of a data system is to ensure that certain work is done according to the formulated explicit rules. This is achieved simply by letting the computer do the work. A systems analyst described it this way: "Why give the human a chance of doing a job wrong, when a computer can do it right?" I call this type of data system a *control data system*. In such a system the humans in the organization have to serve the computer with data in a certain format in order to make the computer do the work. The users are servants of the machine. There are, however, systems where this is not the case. I call these *support systems*. The purpose of such a system is to help the user to do his or her job. Thus the user is the complete master of the system. Decision support systems are one special case of support systems.

The functions of a support system are many. During empirical investigation I have found the following ones:

- To extract reports from common, or at least partly common, databases (*Query languages, report generators*).
- To produce statistical diagrams according to extracted or computed data (*Graphical packages*).
- To compute new figures from old ones according to an understood model of the organisation (*Spreadsheet, decision generators*).
- To retrieve information from other data banks or from an individual's own data collection (*Data base systems*).

The common database has strength, but also a weakness. A strength, since it is possible for people to communicate and exchange information, a weakness since the interpretation of the data representing the information is determined in advance by designers of the data base. It is thus a control data system. The common database might act as a gatekeeper, and not be able to provide the desired information if it has not been anticipated by the data base designers. A support system is a highly individual system. Using a common database perhaps violates the possibilities for the individual person to treat the data according to his or her own apprehension of reality. A common database implies a data model with certain information objects, properties of these objects and dependencies between them. Systems analysts and not the users of the data mostly build this model. Important tacit knowledge might disappear in this process. Alternative ways of building such models are suggested by some researchers (e g Lyytinen & Goldkuhl, 1983).

A management system for a common database usually presupposes integration of all data in the organization. Individual data or data used by just a few people do not need to be accessible by all other people in the company. The risk of misinterpretation is in such a case great. Only data that are common to all people in the company ought to be stored in a common database. Another way of expressing this is saying that only intersubjective data are to be stored in a common database. Nurminen (1982) has pointed out the disadvantages of too great a degree of integration.

Since a support system mostly supports a single user in his or her work and since every person has a working style and an apprehension of reality of his own, the system must be, at least to a large extent, individual. Obviously we cannot use a traditional systems development approach in developing such systems. Personal data processing, where the single users develop and maintain the systems, seems to be an appropriate approach to support systems.

Functions of such a support system might be:

- Data base operations for storing and retrieving data.
- Calculation operations in order to perform mathematical manipulations of numbers representing some sort of model of the company.
- Graphic operations in order to represent a certain set of numbers as curves or bars or pie charts.
- Word processing to support writing of memos, reports, letters etc.
- Two other operations might also be considered:
- Mail systems in order to exchange written messages with other users maybe far away or at times that suits the involved persons.
- Information retrieval in external data banks.

I consider information systems according to three dimensions, the *control* dimension, the *routine* work dimension and the *support* dimension. Every information system can thus be considered as a body ("potato") in this three-dimensional space (Fig 7.2). The main shape of the body indicates it's average purpose.

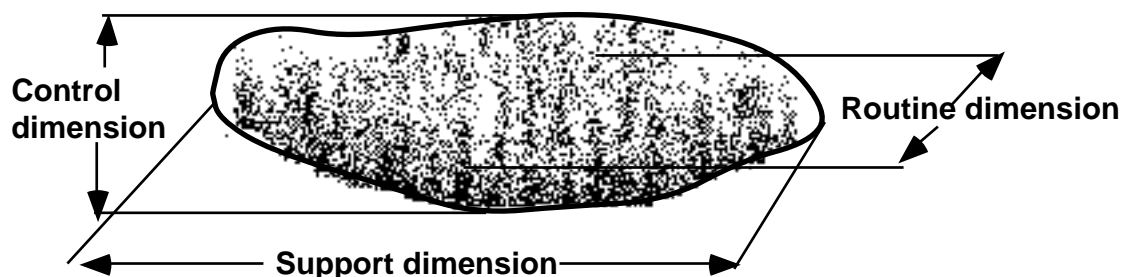


Fig 7.2 Different dimensions in support systems

Information can also be connected to certain persons, groups of persons or the whole organisation. This is the concept of *my information, your information, our information* and *everyone's information* that is described by Holmberg in Nissen et al 1982. Personal data processing is done in connection to personal information. Such information can not easily be processed in a traditional information system. This is due to its personal and tacit character. Since personal data processing supports the processing of personal information we have quite a new area of information processing.

I have so far not discussed the question of whether user development takes tacit knowledge more into account than traditional development. But the truth is, that none of the applications considered here have been developed in a traditional way. The data systems are used in quite a different way than is supposed in the traditional way. In the applications I have studied the people use the computer as a real tool, comparable to an electronic calculator (in fact one of the interviewed persons did exactly that). In traditional data systems people are either humble servants of the systems (data feeders) or passive receivers of thick paper lists (data needers) (cf Sandström 1985).

REFERENSER

Ahrnell B-M: Han studerar hur högnivåspråken fungerar ute på företagen ADB-avdelningarnas motstånd fördröjer användarnas inmarsch Datavärlden, 23 aug, 1983

Alter S L: Decision Support Systems: Current Practice and Continuing Challenge, Addison-Wesley, Mass. 1980

Andersen E, Paulsen M, Sörsven Å : Resultatene av de prosjektarbeid jag har vært med i, har vært lite tilfredsstillende på mange (alle) måter, Data nr 3, 1980

Andersson J: Några reflektioner kring praktiskt utnyttjande av fjärde generationens programmeringsspråk, SSI-aren nr 3, 1983

Arbnor I, Bjerke B: Företagsekonomisk metodlära, Studentlitteratur, 1977

Asphjell A: Kontorsautomatisering Det er kanskje bra det går så tregt? i Nordisk DATAnytt med data, nr 14, 5 nov, 1984.

Asplund J: Om undran inför samhället, Argos, Uppsala 1971

Aubert V: Det skjulte samfunn, Pax, Oslo 1969

Austin J L: How to do things with words, Clarendon Press, London, 1962

Axelsson L, Axelsson O, Broman M, Ersin R, Hansson A, Jönsson K, Lindell J, Petersson H, Åkesson A: Prototyp till litteratursökningssystem för skånesamlingen vid Malmö stadsbibliotek, Lunds Universitet, Informationsbehandling-ADB, 1982

Berg G, Mattsson R: Information Center en jämförelse mellan teori och praktik, Lunds Universitet, Informationsbehandling-ADB, 1986

Berger P L, Luckmann T: The social construction of reality A Treatise in the Sociology of Knowledge, Anchor Books, New York, 1968

Berglind J: Om motiv och behov, kunskap, språk och handling i Nissen H-E (red): Systemutveckling av Vem, för Vem och Hur?, Arbetarskyddsfonden, 1984

Bergström : Objektivitet, Prisma, 1976

Berrisford T, Wetherbe J: Heuristic Development: A Redesign of Systems Design, MIS Quarterly, Vol 7, No 4, Dec 1983

Bertilsson B, Hägerfors A, Lloveras M, Nilsson C, Jansson G-Å, Norrman M, Nolin A, Karlén H-U: Ett informationssystem för Lundasamlingen, Lunds Universitet, Informationsbehandling-ADB, 1982

Bjørn-Andersen N: Fagforeningssejlvord at tage medansvar for godkendelsen af en kravspecifikation ved traditionel systemudvikling Arbejdsformer i Systemudvikling, Århus-rapporten 1975

- Bjørn-Andersen N*: The Organizational Aspects of Systems Design DATA, No 12, 1976
- Bonzek R, Holsapple C, Whinston A*: Foundations of Decision Support Systems, Academic Press, New York, 1981
- Borum F*: Brugerindflydelse på systemutveckling hvorfor og hvordan? NordData 76, 1976
- Borum F*: Systemkonstruktion og videnskabelig driftsledelse i Borum (red): EDB, Arbejdsmiljø og Virksomhedsdemokrati, Nyt fra Samfundsvidenskaberne, 1977
- Bosrup L, Holmberg S*: ADB investeringen en angelägenhet för företagsledningen. Liber, 1984
- Brandinger R, Norrby J*: ADB systemarbete, Studentlitteratur, 1980
- Brandt P, Gustafsson M R, Johansson L Å*: VIA-projektets slutrapport, Institutionen för informationsbehandling, Göteborgs universitet, 1978
- Brandt P, Johansson L Å*: Situationsanpassad och situationsanpassbar systemutvecklingsmodell, slutrapport ASAS-projektet, Institutionen för informationsbehandling, Göteborgs universitet, 1980
- Checkland P*: Systems Thinking, Systems Practice, John Wiley & Sons, Chichester, 1981
- Churchman C W*: The Design of Inquiring Systems, Basic Books, New York, 1971
- Ciborra C*: The Social Costs of Information Technology and Participation in System Design, IFIP conference on System Design, Riva del Sole, Italien, 1982
- Cooley M*: Architect or Bee?, Langley Technical Services, Slough, 1980
- Davis G*: Caution: User Developed Systems Can Be Dangerous to Your Organization, Management Information Systems Research Center, School of Management, University of Minnesota, Minneapolis, 1984
- Dickson G W, Leithheiser R L, Wetherbe J C, Nechis M*: Key Information Systems Issues for the 1980's, MIS Quarterly, September 1984, pp 135-159
- Ehn P*: Styrning ifrågasatt. Om möjligheter, hinder och gränser för demokrati i systemutveckling i *Göranzon B*: Perspektiv på systemutveckling, Studentlitteratur, Lund, 1978
- Ehn P, Sandberg Å*: Företagsstyrning och löntagarmakt, Prisma, 1979
- Engdahl H, Holmgren O, Lysell R, Melberg A, Olsson A*: Hermeneutik, Rabén & Sjögren, 1977
- Erikö H*: On Data Description Facilities in some Application Development Systems, 8th Scandinavian Research Seminar on Systemeering, Århus, 1985

Festinger L: A Theory of Cognitive Dissonance, Stanford University Press, 1957

Feyerabend P: Against Method, London 1972

Fitzsimmons A, Love T: A Review and Evaluation of Software Science, Computing Surveys, Vol 10, No 1, 1978, pp 1-18

Flensburg P: Systemutveckling med människan i centrum, Lunds universitet, Institutionen för informationsbehandling-ADB, 1979

Flensburg P: Grundföreställningar inom informationsbehandling, Lunds universitet, Institutionen för informationsbehandling-ADB, 1980a

Flensburg P : Systemutvecklarens nya roller på 1980-talet, Data nr 9, 1980b

Flensburg P: Towards User Development A dialogue in Kerola & Koskela (eds): Report from the 5th Scandinavian Research Seminar on Systemeering, Oulo, 1981a

Flensburg P: Systemutveckling Vad är det?, Lunds universitet, Institutionen för informationsbehandling-ADB, 1981b

Flensburg P: En social och kreativ beskrivningsteknik, Lunds universitet, Institutionen för informationsbehandling-ADB, 1981c

Flensburg P: Datorer på biblioteken, Tidskriften på biblioteken, Statens kulturråd, 1982

Flensburg P: Erfarenheter av användarutveckling planering av en undersökning, Lunds universitet, LUIS-gruppen, Informationsbehandling-ADB, 1984a

Flensburg P: Erfarenheter av användarutveckling Intervjuer med användare, Lunds universitet, LUIS-gruppen, Informationsbehandling-ADB, 1984b

Flensburg P: Erfarenheter av användarutveckling en sammanställning av ett antal intervjuer med användare, Lunds universitet, LUIS-gruppen, Informationsbehandling-ADB, 1984c

Flensburg P: Erfarenheter av användarutveckling tolkning av intervjusvar, synpunkter på en undersökningsmetodik, Lunds universitet, LUISgruppen, Informationsbehandling-ADB, 1984d

Flensburg P: Lägesrapport från ett aktionsorienterat forskningsprojekt, LUIS-gruppen, Lunds Universitet, Informationsbehandling-ADB, 1984e

Flensburg P: Effekter av formalisering, Lunds universitet, Informationsbehandling-ADB, 1984f

Flensburg P, Sandström G: Slutrapport från forskningsprojekt MOMS, Lunds Universitet, Informationsbehandling-ADB, 1980

Flensburg P, Holmberg K Å, Nissen H-E: Experiments on organization adapted, user developed systems, Lunds Universitet, Informationsbehandling-ADB, 1981

Floyd C: A systematic look at Prototyping, Working Conference on Prototyping, Belgien, 1984

Freire P: Pedagogik för förtryckta, Gummesson, Stockholm, 1972

Friis S: Prototyping en ny metod eller ett uttryck för ett nytt synsätt? i Nissen H-E (red): Systemutveckling av Vem, för Vem och Hur?, Arbetarskyddsfonden, 1984

Friis S: Tools for User Prototyping, IRFIS 6, Italien, 1985a

Friis S: Användarutvecklade kravspecifikationer med hjälp av prototypsystem, i Chrona, Goldkuhl, Nilsson (red): "Vägen till bättre informationssystem", RDF, Stockholm, 1986 (under utgivning)

Gingras L, McLean E R: Designers and Users of Information Systems: A Study in Different Profiles, Joint International symposium on Information Systems, IFIP TC 8, Sydney, 1984

Glimell H: Designing Interactive Systems for Organizational Change, BAS ek förening, Göteborg, 1975

Goldkuhl G: Modeling Communicative Acts in Information Systems, TIMS XXVI International Meeting, Copenhagen, June 17-21, 1984

Goldkuhl G, Lyytinen K: A language Action View of Information System i Ginzberg M, Ross C A (eds): Proceedings of the Third International Conference on Information Systems, Ann Arbor, Michigan, 1982

Goldkuhl G, Lyytinen K: Information System Specification as Rule Reconstruction, HUMOR, 1983

Greenbaum J: Arbetsdelning inom databehandlingsområdet, Monthly Review, vol 38, no 3, juli 1976 (svensk översättning i Zenith nr 59)

Göranzon B (red): Perspektiv på datasystemutveckling, Studentlitteratur, Lund, 1978

Göranzon B (red): Datorutvecklingens filosofi, Carlsson & Jönsson, 1983a

Göranzon B: Datorn som verktyg, Studentlitteratur, Lund, 1983b

Habermas J: Zur Logik der Socialwissenschaften, Phil. Rundschau, Feb 1967

Habermas J: Knowledge and Human Interests, Heineman, London, 1978 (German original: Suhrkamp, 1968)

Halldén S: Datorerna och den tysta kunskapen, Skolan och Datorn, nr 2, 1985

Hansson A, Pettersson H: Fallstudier över superhögnivåspråket BUSTER och FOCUS, Lunds universitet, LUISgruppen, InformationsbehandlingADB, 1982

Hedberg B, Jönsson S: Designing Semi-Confusing Information Systems in Changing Environments, FE-rapport nr 73, Göteborgs universitet, företagsekonomiska inst, 1976

Hedberg B, Mumford E: The Design of Computer Systems: Man's vision of man as an integrated part of the system design process, i Mumford E, Sackman H (eds): Human Choice and Computers, North-Holland, 1975

Heidegger M: Varat och tiden, del I, Doxa, 1981 (tyskt original: Sein und Zeit, Tübingen 1926)

Hellevik O: Forskningsmetode i sosiologi og statsvitenskap, Universitetforlaget, Oslo, 4:e uppl, 1980

Hiltz S, Turoff M: The Network Nation: Human Communication via Computer, Addison & Wesley, 1978

Høyer R (red): ... over til EDB, Tanum, Oslo, 1974

IBM: Implementation Guide for an Information Center, 1982

Israel J: Om en relationistisk socialpsykologi, Korpen, Göteborg, 1979

Israel J: Språkets dialektik och dialektikens språk, Esselte studium, Lund 1980

Israel J: Om konsten att blåsa upp en ballong inifrån, Korpen, Göteborg, 1982

Ivanov K: Quality-control of information: on the concept of accuracy of information in databanks and in Management Information Systems, Akademisk avhandling, KTH, Stockholm, 1972

Ivanov K: Systemutveckling och ADB-ämnets uppkomst, i Nissen H-E (red): Systemutveckling av Vem, för Vem och Hur?, Arbetarskyddsfonden, 1984

Jenkins M : Prototyping: A Methodology for the Design and Development of Application Systems, Spectrum, Vol 2, No 2, apr 1985

Josefsson I: Finns det nån plats för humanisterna i datavärlden?, Datavärlden 9 dec, 1985

Järvinen P: Levels for Utilization of Computing, Convexion informatique, SICOB, Paris, 1985

Järvinen P: Flexibility of Software as a Dimension of Information Systems Assessment, submitted to IFIP WG 8.2 Information Systems Assessment Conference, 27-29th August 1986

Järvinen P, Kirjonen J, Tylillä P, Vihmalo A: Analysis and Design of Jobs in a Man-Computer System, Department of Mathematical Sciences, University of Tampere, 1982

Kant I: Kritik der reinen Vernunft, Leipzig, 1924 (original 1781)

Kraushaar J M, Shirland L E: A Prototyping Method for Applications Developed by End Users and Information Systems Specialists, MIS Quarterly, sep 1985

Kuhn T: De vetenskapliga revolutionernas struktur, Doxa, Lund 1981, engelskt original "The structure of scientific revolutions", Chicago, 1962.

Kuvaja P, Similä J: A Selective Analysis of Application Generators in Systems Development Work, 8th Scandinavian Research Seminar on Systemeering, Århus, 1985

Langefors B: Theoretical Analysis of Information Systems, Studentlitteratur Auerbach, 1966

Langefors B: System för företagsstyrning, Studentlitteratur, 1968

Langefors B: Hermeneutics, Infology and Information Systems, TRITA-IBADB nr1052, Stockholm, 1977

Leavitt H : Organisationsändring strukturelle, teknologiske og personorienterede metoder i Borum (red): EDB, Arbejdsmiljø og Virksomhedsdemokrati, Nyt fra Samfundsvidenskaberne, 1977

Leitheser R L & Wetherbe J C: A Survey of Information Centers: Services, Decisions, Problems and Successes, Management Information Systems Research Center, School of Management, University of Minnesota, Minneapolis, 1984a

Leitheser R L & Wetherbe J C: Avoiding the pitfalls of End-User Computing, Management Information Systems Research Center, School of Management, University of Minnesota, Minneapolis, 1984b

Lewin K: Action Research and Minority Problems, Journal of Social Issues, No 2, 1946, s 34-46

Lie-Nielsen S, Colliander L: Principer för en ny generation systemutvecklingsverktyg, RDF's rapportserie nr 16, 1983

Lindholm S: Vetenskap verklighet och paradigm II, UMIL-rapport nr 12, Stockholms Universitet, Pedagogiska institutionen, 1979

Lindholm S: Vetenskap, verklighet och paradigm, AWE/Gebers Stockholm, 1980

Lindholm S: Kunskap från fragment till helhetsyn, Kontenta, 1985

LUIS-gruppen: Forskningsprogram, Användarstyrd Systemutveckling och Systemanvändning empirisk forskning kring informationssystem, Lunds universitet, Institutionen för informationsbehandling-ADB, 1985

Lundeberg M, Goldkuhl G, Nilsson A: Systemering, ISAC, Studentlitteratur, 1978

Lyytinen K: Reality Mapping or Language Development a tentative analysis of alternative paradigms for information modelling, SYSLAB, WP no 27, 1983

Martin J: Programming without programmers, Prentice-Hall, New Jersey, 1982

Masuda: Informationssamhället, Liber, 1984

Mumford E: Designing Human Systems, Manchester Business School, 1983a

Mumford E: Designing Secretaries, Manchester Business School, 1983b

Mumford E, Henshall D: A Participative Approach to Computer Systems Design, Associated Business Press, 1979

Mundt-Almquist G, Andersson A, Lindström B: Slutrapport från projekt LIMES, Malmö stadsbibliotek, 1985

Neergaard P: Er brugerrepræsentanter løsningen på brugerne modstand mod forandring? i Borum (red): EDB, Arbejdsmiljø og Virksomhedsdemokrati, Nyt fra Samfundsvidenskaberne, 1977

Nissen H E: On Interpreting Services Rendered by Computerized Information Systems, Dissertation, University of Stockholm, 1976

Nissen, H-E: Towards a Multi Subject Groups Conception of Information Systems, Lunds Universitet, Informationsbehandling-ADB, 1980

Nissen H-E: Subject Matters Separability in Information Systems Design Methods i Olle T W, Sol H G, Tully C J (eds): Information Systems Design Methodologies A Feature Analysis, North Holland, IFIP, CRIS 2, 1983

Nissen H E: Acquiring Knowledge of Information Systems Research in a Methodological Quagmire, Department of Information and Computer Sciences, Lund, 1984

Nissen, H-E: A Theor-Ethical Basis for Studies of Information Systems Use and Development, 8th Scandinavian Research Seminar on Systemeering, Aarhus, 1985a

Nissen H-E: Kräv datasystem du kan begripa det har du rätt till! "Vägen till bättre informationssystem", RDF-konferens, 23-24 okt 1985b

Nissen H E, Carlsson S, Flensburg P, Holmberg K Å, Sandström G, Wormell I: User Oriented Information Systems A research program Lunds Universitet, Informationsbehandling-ADB, 1982

Nurminen M: Against Systems, in Kerola & Koskela (eds): Report from the 5th Scandinavian Research Seminaar on Systemeering, Oulo, 1981

Nurminen M: Human-Scale Information Systems, Institutt for Informasjonsvitenskap, University of Bergen, 1982

Nygaard K: Fagbevegelsen en ny oppdragsgiver in Høyer (ed): ... over til EDB, Oslo, 1974

Olerup A: A Contextual Framework for Computerized Information Systems, Akademisk avhandling, Lunds universitet, Informationsbehandling-ADB, 1982

Olerup A: Prescriptive Approaches for Developing Information Systems Introducing the notion of strategic choice, presenterat vid Working conference on Development and Use of Computer-Based Systems and Tools in the context of democratization of work, Aarhus, Danmark, 1985

Polanyi, M: The Tacit dimension, Anchor Books, New York, 1966 (första upplagan 1958)

Persson M: Användarerfarenhet av applikationsgeneratorer på persondatorer, Lunds universitet, Informationsbehandling-ADB, 1985

Rapoport R N: Three Dilemmas in Action Research with special Reference to the Tavistock Experience, Human Relations, Vol 23, No 6, 1970, s 499-513

Revay P: RAS SIS' handbok 113 i teoretisk och praktisk belysning, akademisk avhandling, Institutionen för administrativ databehandling, Stockholms universitet, 1977

Rhenman E: Företaget som ett styrt system, Stockholm, 1964

Rivard S, Huff S L: User Developed Applications: Evaluation of Success from the DP Department Perspective, MIS Quarterly, Mars 1984, sid 39-50

Romanovsky R: Sammanställning av intervjufrågor och svar Fallstudier, manus, Lunds universitet, Informationsbehandling-ADB, 1985

Rosenbrock H H: Alternative Paths for Technological Development, UMIST, Control Systems Center, Manchester, 1984

Ryle G: The concept of Mind, Oxford, 1949

Rzewski G: Prototypes versus Pilot Systems: Strategies for Evolutionary Information Systems development, Working Conference on Prototyping, Belgien, 1984

Röstlinger A, Selldén J: How to perform an Empirical Study on a Change Analysis Method, i Nurminen M, Gaupholm H T (eds): Report from the sixth Scandinavian Research Seminaar on Systemeering, Øystese, 1983

Sandberg Å (red), Abrahamsson B, Ehn P, Swedner H: Utredning och förändring i statsförvaltningen, Stockholm, 1978

Sandemose A: En flykting korsar sitt spår, 1933

Sandén M: Torgny Tholerus, mannen bakom KOM: "Världsrevolutionen börjar 1995", Datavärlden nr 18, 11 nov 1985

Sandström G: Improved Retrieval from Information Systems, Department of Information and Computer Sciences, Lund, 1984

Sandström G: Towards Transparent Data Bases How to Interpret and Act on Expressions Mediated by Computerized Information Systems, Akademisk Avhandling, Lunds Universitet, Informationsbehandling-ADB, 1985

Sandström G, Wormell I: Studier av modeller för systemutveckling hos lantbrukskooperationen, Lunds Universitet, Informationsbehandling-ADB, 1980

Sandström G, Nissen H-E, Carlsson S, Flensburg P, Friis S: Användarnära bruk och utveckling av beslutsoch informationssystem, LUIS-gruppen, Lunds Universitet, Informationsbehandling-ADB, 1983

Schneider M: Models for the Design of Software User Assistance, Sperry Univac, Blue Bell, 1980

Searle J R: Speech Acts, Cambridge University Press, 1969

Semprevivo P: Systems Analysis Definition, Process and Design, 2nd edition, Chicago, 1982

Shneidermann B: Software Psychologi Human Factors in Computer and Information Systems, Winthrop Publishers, Mass,

Simon H A: Administrative Behavior. A Study of Deciscion-making Processes in Administrative Organizations, The Free Press, New York, 1957

SIS: Riktlinjer för Administrativ Systemutveckling, Handbok nr 145, 1979

Skjellanger Svein H: Fremtidens kontor hvor ble det av? i i Nordisk DATAnytt med data, nr 14, 5 nov, 1984.

Skousen Thomas: Sociale konsekvenser af næste generations kontormiljø i Nordisk DATAnytt med data, nr 14, 5 nov, 1984.

Susman G I, Evered R D: An Assesment of the Scientific Merits of Action Research, Adminstrative Science Quarterly, vol 23, Dec 1978, s 582-603

Svensson N G: Vad är IRM/IA?, Liber, 1983

Swanson B: Information system approaches: directions for research and practice, Management Datamatics, vol 25, no 4, pp 155-163, 1976

Swedner H: Sociologisk metod, (2:a upplagan) Gleerups, 1970

Swedner H: Sociologisk metod, (4:e upplagan) Liber, 1982

Swende E: SASMO -SAS data models, SAS, Data planning group, 1980

Toffler A: Framtidschocken, 1971

Toulmin S: The Uses of Argument, University Press, Cambridge, 1958

Törnebohm H: Studier av kunskapsutveckling, Doxa, Lund, 1983

Törnebohm H: Undersökande system, paradigm och tematiska diskussioner, Avdelningen för vetenskapsteori, Göteborgs Universitet, 1975

Wittgenstein L: Tractatus Logico-Philosophicus, Routledge & Keagan Paul 1961 (Original 1921)

Wittgenstein L : Philosophical Investigations, Blackwell, 1976

Weizenbaum J: Computer Power and Human Reason, Freeman & Co, San Francisco, 1976 (Penguin 1985)

Wood-Harper T, Fitzgerald G: A Taxonomy of Current Approaches to Systems Analysis, The Computer Journal, vol 25, no 1, 1982, pp 12-16

von Wright G H: Den logiska empirismen - en huvudriktning i modern filosofi, Söderströms, Helsingfors, 1943