

## IT-historia

### **Självbiografi av**

Christian Ekvall, född 1944

### **Bakgrund, barndom**

Är född i Stockholm som enda barn och under uppväxttiden varit mantalsskriven på ett dussin ställen, i Stockholm (Årsta, Bromma, Kungsholmen, Sundbyberg, Högdalen, Kungsholmen), Västerås och Köpenhamn. Har också tidvis varit inneboende hos släktingar.

Mina föräldrar skilde sig när jag var sex år. De har alltid haft ett gemensamt uppträdande gentemot mig och visat mycket kärlek och tillit till mig och allt jag velat göra och gjort.

Fader: Servitör, försäkringsförsäljare, hålkortsoperatör, kontorsrationaliserare, ergonomi-intresserad. Omgift. Rötter från Stora Tuna i Dalarna.

Moder: Hemmafru, telefonist, tipstalongsoperatör. Omgift två gånger. Skild. Rötter från Bergen i Norge i andra generation samt från Stockholm.

Har haft mycket kontakt med många kusiner (yngre) och syslingar samt deras föräldrar.

Har bott längre tider hos barnlös faster med man (farbror) som också gett mig mycket kärlek och tillit. Han var administrativ direktör på ett antal stora industriföretag i Stockholmstrakten.

Skolgången belv mycket splittrad p.g.a. många adressbyten.

1:a klass i Åkeshov, Bromma

2:a klass i Västerås

3:e – 5:e hösttermin klass i Fredhälls skola, Kungsholmen:

- Problem med stavning. Dyslexi konstaterades så underbetyg i svenska blev ej tillåtet. Dyslexin gör att jag läser långsamt, eller rättare sagt så läser jag inte orden utan bokstäverna väldigt fort. Är mycket bild-, figur- och symbolorienterad istället.

5:e vårtermin – 7:e klass i Sundbyberg:

- Lyckades vinna en schackturnering i 6:e klass och vann även över min magister. Min mormor och far har lärt mig spela schack.
- Bedömdes i 6:e klass olämplig för högre studier (Realskola) p.g.a. endast godkänt (B) betyg i engelska.
- Lyckades med väl godkänd (Ba) i engelska i 7:an med annan lärare och privat stöd-undervisning.

Realskola i Sundbyberg:

- Tyska språket var besvärligt p.g.a. icke inspirerande lärare.
- Franska språket gick bra tack vare inspirerande lärare.
- Växlade medvetet koncentrationen mellan engelska och tyska för att inte vara underkänd två år i rad.
- Har alltid varit mycket bra i matematik, fysik, biologi, geografi, historia, samhällskunskap, ritning och gymnastik.
  - o Hade en mental krock med en fysiklärare när jag överbevisade honom inför hela klassen med en felaktig elektrisk koppling, vilket resulterade i endast godkänt betyg som jag absolut inte var van vid, tvärtom.
- Skolklassen var mycket homogen och i stort sett alla umgicks även på icke-skoltid.

Gymnasium ett halvår i Sundbyberg.

Gymnasium i ett år i Köpenhamn (Östre Borgedyd). Flyttade p.g.a. att min mor började sitt tredje äktenskap:

- Blev väl omhändertagen. Danska språket lärde jag mig löpande. Mina norska rötter var behjälpliga. När jag pratade danska så var det ingen som förstod mig, men när jag använde danska ord uttalade på svenska så gick det bra – precis som danskarna gör inför

## Självbiografi av

## Christian Ekvall

svenskar, fast tvärtom. Där var svenska språket obligatoriskt en timme i veckan! I Sverige hade jag dansk språkundervisning en timme ett läsår.

- En fascinerande erfarenhet av betygssättning fick jag framför allt i historia. Jag var van vid Med Beröm Godkänd (AB), men nu fick jag högsta betyg, stort A (A).
  - o När jag nästa år var tillbaka i svenskt gymnasium fick jag endast Godkänt betyg (B). Mitt förtroende för lärare försvann då. Jante-lagen visade sig.
- Jag tvingades gå om första klassen i Danmark p.g.a. att jag måste lära mig danska för att få godkänd studentexamen. Jag hade inte för avsikt att förvärvsarbeta i Danmark som vuxen, så jag flyttade tillbaka till Stockholm igen och min mor följde med.

Gymnasium och studentexamen i Stockholm, Kungsholms Högre Allmänna Läroverk:

- Jag började i första klass/ring igen, för det danska året ansågs inte meriterande.
- Jag träffade min blivande maka och mor till mina barn i denna klass. Vi pluggade ihop och kompletterade varandra förträffligt.
- I historia och svenska blev det totalkrasch. Läraren/klassföreståndaren satte kraftigt underbetyg på mina svenska uppsatser (C) och kluttade sedan till det till icke-godkänt (BC). Han gav mig endast godkänt i historia (B), enligt ovan. Han hotade min studentexamen. Svenskan räddade jag med att alltid skriva fackliga ämnen som jag fick mycket beröm för av facklärarna. Engelskan hade jag problem med.
- Detta innebar att jag fick göra ett tredje år i första årskursen!
- Äntligen fick jag en tysk lärare som inte tittade i backspegeln och som jag fick förtroende för. Jag hade också privat stödundervisning som resulterade i att jag blev Väl Godkänd (Ba)!
- Normalt var jag i mina klasser bland de bästa eleverna i matematik och bäst i fysik.
- Gymnastik har alltid gått bra, men jag är ingen tävlingsmänniska. Brukar inte hålla reda på min placering. Normalt är jag sist ut men väldigt ofta först framme.
- I grupparbete tar jag kommandot för en grupp endast om ingen annan är en bra ledare.
- Håller mig alltid väl informerad, helst från ursprungskällan. Jag vill alltid veta saker och ting, det räcker inte med att tro något.

Militärtjänstgöring

- Ville absolut köra lastbil. Blev lastbilsförare på pansarregementet P1 i Enköping som ABC-specialist (skyddskompetens vid Atom-, Biologiskt- eller Kemiskt anfall) först på spaningskompani sedan på stridsvagnskompani och slutligen på brigadstab.

## **Framtidsutsikter**

I min ungdom hade jag endast under enstaka perioder kompisar och lekt med dem ute men egentligen aldrig tagit med dem hem. Har alltid ägnat mycket tid åt läsläsning och dessemellan pulat hemma med Dinky Toys och en stor anläggning med Märklintåg samt hjälpt min under långa tider ensamstående och pyssliga mor med praktiska saker. Har haft möss som husdjur, vilka alltid var med mig. När min första mus dog efter fem(!) års leverne (normalt två, högst tre) hade det blivit 25 stycken i buren så jag avvecklade alltihop.

Varje sportlov åkte jag skidor någonstans. Varje sommarlov reste jag någonstans. Från 13 års ålder sommarjobbade jag med något undantag varje sommar i fyra veckor (cykelbud, bankkontor, tipstjänst) och ibland juljobb (posten).

Under gymnasietiden var jag privatlärare i matematik åt en högstadieelev som behövde stödundervisning.

Jag är både vänster- och högerhänt. Har aldrig tränat att skriva med vänster.

Mina yrkesval har jag alltid bestämt själv. Principellt ville jag alltid studera så mycket som möjligt. Genom mycket "självpulande", sommarjobb och juljobb var jag tidigt på det klara med att jag inte ville slita ut mig på grovarbete eller hantverk. Jag har alltid haft för avsikt att leva länge med god hälsa och goda inkomster för att må bra som pensionär. Mina generella mål var civilingenjör (såsom min mors andra make) eller läkare (såsom en kusin till min far). Mina morböder var egna företagare och min morfar och hans far har drivit en gardinfabrik i Stockholm, där bostadsområdet Minneberg nu ligger.

Nyttigheter som jag lärt mig i skolan fram till studenten, vilka jag aktivt använt mig av i mitt yrkesval är:

1. Ett studentbetyg som berättigade till högre studier var ju nödvändigt.
2. Fick lära mig "Den nya matematiken"/mängdlära via stencilerade kompendiekopior under ett halvår i första gymnasieklassen i Sverige. Under vistelsen i Köpenhamn 1962 och gymnasiet där fick jag den matematiklärare som hade skrivit kompendiet.
3. Engelska språket blev en nyttig kunskap i databranschen.
4. Det gav mig insikter i hur lite människor egentligen vet och hur mycket människor tycker.

Inriktning på mina fortsatta studier:

1. Mitt studentbetyg berättigade inte till högskolestudier i Stockholm till de linjer jag var intresserad av.
2. Min farbror (fasters man) föreslog 1966 att jag skulle bli "systemman", då ett helt nytt yrke. När jag frågade honom vad detta var sade han att jag fick ta reda på det. När jag frågade studievägledare så visade det sig att jag visste mera! Den enda rekommendation jag kunde få fram var att läsa 3 betyg (60 poäng) i matematik på universitetet samt 3 betyg (60 poäng) i numerisk analys, d.v.s. mycket om lite. (Det visade sig efter några år att de studierna gav mest arbetslöshet.) Min farbror ordnade så att jag fick träffa personalcheferna på Bonniers och IBM. De var samstämmiga om att jag skulle lära mig lite om mycket, så det blev ett hopplock av 2 betyg (40 poäng) i matematik, 2 betyg i (40 poäng) i administrativ databehandling (som under tiden blivit en ny studiemöjlighet), 1 betyg (20 poäng) i matematisk statistik och 1 betyg (20 poäng) i företagsekonomi. Det blev dock justerade planer då jag inte klarade den matematiska statistiken direkt. Jag rekommenderades av en kamrat att göra examensprov i vanlig statistik istället, med endast några extra böcker att läsa och utan lektioner vilket gav resultat direkt.

Under studietiden koncentrerade jag mig hårt på att klara universitetstiden på tre års studielån. Studentrevolter och andra yttringar märkte jag aldrig utan hörde endast talas om.

Några signifikativa händelser:

1. Mitt provresultat i den obligatoriska systemeringsuppgiften på ADB-institutionen hade den kvaliteten att jag blev erbjuden av Eva Lindencrona att bli handledare för andra studenter. Dock tackade jag nej för jag prioriterade att snabbt bli färdig med mina egna studier. Hon kommenterade det med att jag "inte var riktigt klok". Så jag antog erbjudandet. T ex träffade jag då också Janis Bubenko och Louise Yngström.
2. Sommarjobbade med att köra en Scania grusbil mellan Mälareöarna och Stockholm.

### **Introduktion till dator-/datatekniken**

Min första kontakt med och användning av en dator och de första uppgifterna var på ADB-institutionen på KTH 1968 i samband med programmeringsuppgifter i BASIC och COBOL. Lärde mig även ALGOL och ASSEMBLER.

Nedan följer min egna datatekniska utveckling via mina anställningar med för mig intressanta samarbeten med särskilda personer, om rutiner och speciella svårigheter, maskintyper och kodningstekniker:

### **Steg 1, år 1971-1974**

Mitt utbildningsval var helt rätt. Jag och min nyblivna hustru köpte hus på landet utanför Södertälje. Banken beviljade huslån baserat på min utbildning. Jag hade lätt att få jobb, ansåg de. Sökte tre olika jobb. Min farbror beordrade mig att inte ta det jobb jag verkligen siktade på direkt, utan skulle jobba tre till fem år någon annanstans först. Ett projektledarjobb tackade jag nej till, eftersom jag inte är bäst på det. En annan arbetsgivare var för senfärdig. **Min utbildningsinriktning var systemerare av administrativa system**, men grundlig som jag alltid varit, började jag med en **anställning 1971 som programmerare** på ett stort industriföretag (Electrolux) inom Wallenbergfären i Stockholm. De började precis använda en IBM360-dator, fyra nya hårddiskar (10MB/st) och två bandstationer med COBOL och ASSEMBLER. Efter ett par enkla listande applikationer började den egentliga utmaningen. Ett komplett nytt **order-, lager- och faktureringsystem samt kundreskontra och statistik** skulle byggas. Nytt var att programmen skulle byggas endast utifrån verksamhetens utdata/listlayouter och befintligt indata, men utan programbeskrivningar. Nytt var också att många systemtabeller med rubriker och andra metadata skulle användas. Det var tre systemerare som stod för verksamhetsbeskrivningen. Jag blev så småningom ansvarig programmerare för hela systemkomplexet som ytterligare fem hade bidragit till.

Tabelltekniken var så omfattande att jag utvecklade ett speciellt länkat program som bara hanterade de för stunden aktuella tabelldata. Idag kallar man det för cache-minne. Dessutom var utmaningen att vid utskrift av en faktura kunna fastställa om det skulle stå "Faktura" eller "Kreditnota" högst upp, eftersom det inte förrän när alla fakturans rader var på plats och beloppen summerade, som det visade sig om det blev ett negativt belopp – då ska det stå "Kreditnota". Lösningen blev att alla sidor måste byggas upp i minnet innan utskriften kunde göras.

När jag behövde göra stora systemtester, brukade jag "låna" hela datorhallen ensam någon lördag-söndag. Det var lärorikt att se var exekveringstiderna förbrukades; bandstationsläsning och utskrifter! Det fanns även en modemuppkoppling till centrallager för tidiga morgonleveranser.

Systemet exekverades varje natt, till en början i tre versioner, ett för varje dotterbolag. Med tiden blev det ytterligare tre. Därtill skulle en variant köras på egen datoranläggning i Danmark. Jag lyckades i alla fall att använda samma program för alla varianter trots att det var helt olika körningar/exekveringar. (Första steget till SOA-tänk som alltför få har förstått innebörden av. Det rätta är just gemensamma systemfunktioner som återanvänds.)

För mig viktiga relationer, i och utanför min organisation och arbetsuppgift, var behovet av att etablera en mycket nära kontakt med verksamhetens systemanvändare. När de hade förändringsönskemål, tog jag programlistan under armen och ändrade i den omgående medan vi pratade. Om det var en enkel förändring lovade jag att fixa den samma kväll och beordrade användaren att göra en testorder samtidigt. Principen var, att om jag stannade en extra timme, tjänade användarna dagar på detta, samtidigt som jag slapp testa något som jag ändå inte kunde bedöma. Jag lovade endast det jag kunde bedöma, d.v.s. att systemet inte skulle tvärstanna (dumpa) eftersom alla mina programutgångar var välspecificerade. Jag hade också ett antal inbyggda kontrollmätningar för att löpande veta att allt fungerade korrekt. Om allt gick bra kunde användarna redan nästa dag göra fullskalig användning av den/de nya systemfunktionerna. Reservlösningen var alltid att datoroperatörerna, om systemet dumpade, skulle göra omstart med de gamla programmen. Jag har upplevt datorhallens skyddade verkstad och rosenrasande användare nära hjärtinfarkt.

Jag ville dokumentera systemen i ett Fipotek, såsom jag lärt mig på ADB-institutionen (FiPoTeK = Fil-, Post- och TermKatalog, senare kallat t ex Data Dictionary). Mitt motiv var att dessa beskrivningar och relationer fanns bara i mitt huvud och utspritt i alla programmen vilket blir svårtillgängligt för andra. Dock blev uppdraget nedprioriterat av andra ”viktigare” uppdrag.

## Steg 2, år 1975-1980

Efter fyra år ville jag inte vara programmerare längre samt ville inte fortsätta på Electrolux. Jag var trött på att bli uppringd på nätterna p.g.a. felstrukturerade program och svulstig dygnsorienterad multisystemlösning med hålkort och tidsödande filskyffling mellan 75 program. Jag var nyfiken på databasteknik och terminaler. Jag svarade **1975** på en annons med två tjänster i **systemering** på Scania i Södertälje, vilket varit målet under hela tiden. Hemma hade jag en ettårig son. (Son nummer två såg dagens ljus 1977.)

Annonsen handlade om systemering av ett nytt ekonomisystem, alternativt ett nytt personalsystem. Dock roade inte någon av dessa mig. Men i ”pipeline” fanns behovet av ett nytt **order- och faktureringsystem**. Det passade mig. Datoriserade ordersystem var för övrigt i ropet på den tiden när datorkapaciteten började matcha verksamhetskraven.

## Nu fick jag möjligheten att utforma ”systemet” som jag tyckte att det skulle fungera för människor och datorer.

Mina ledmotiv var:

- Nöjda användare och IT-folk som skulle vara glada varje dag, år ut och år in.
  - o En motiverad människa presterar minst fem gånger mer än som omotiverad.
  - o Det blir en hög utväxlingseffekt med flera motiverade människor.
  - o Fenomenet är ”smittsamt” (i positiv mening)!
- Kvalitet är viktigare än kvantitet
  - o Kvalitet ger kvantitet (mer eller mindre automatiskt)
  - o Kvantitet ger **inte** kvalitet (bara med ren tur)
- Lovar aldrig något som inte är genomförbart.
- Lovar färdigtidpunkt med optimistkoefficient
  - o Schablonen är ’pi’ (3,14), i komplexa fall ’pi’ i kvadrat (10)
  - o Mottagaren blir allvarligt besviken om det efterfrågade inte är färdigt vid leveransdagen.
  - o Mottagaren blir ”själglad” om det är färdigt före utsatt tid.
    - Ännu gladare blir han om kostnaden också blir billigare, vilket den normalt blir med kortare tidsåtgång.
- Alltid måldrivna resultat
  - o Ett resultat har inget självändamål,

- Fokusera på de efterfrågade resultaten som behövs för att nå målet.

Mina principer var:

- Beskriv ”en sak en gång” - och återanvänd!
  - Denna ledstjärna upprepades flera gånger om dagen under flera år.
  - Det långa uttrycket är: att motiverat göra(/ha), rätt sak, på rätt sätt, på rätt ställe, av rätt person/organisation, vid rätt tillfälle, i rätt ordning.
- Varje åtgärd har sina konsekvenser
  - Det som är bra väcker intresset
  - Det som är dåligt måste riskbedömmas, först i tidsdimension och sedan i kostnadsdimension
    - Om störande, förfina systemstödet/lösningen
    - Om oacceptabelt, måste annan lösning hittas.
- Tänk positivt
  - Se möjligheterna, omvandla problemen till möjligheter!
  - Negativa tankar blockerar allt och mer därtill!
  - Människans hjärna har svårt att förstå negationer.
    - Med flera negationer på varandra gör tankegångarna ”kullerbytta”.
- Fokusera beroenden
  - Människor vill ha kontroll på saker och ting.
    - Nio av tio gör detta genom att styra utan att ha någon kontroll på vad man styr, d.v.s. fokuserar endast på att göra saker på rätt sätt.
      - Det blir snabbt fler än tre nivåer och man tappar kontrollen totalt.
    - Fokusera istället på vad man är beroende av, d.v.s. rätt saker.
      - Det visar sig att det inte blir så mycket att hålla reda på samt det ger en betydligt stabilare grund.
      - Dessutom är det mer logiskt korrekt p.g.a. livslängdsresonemang.
  - Ordning och reda
    - Förutsättning nummer ett är att varje sak fått sin givna plats.
    - Då blir det lättare att hantera.
    - Då blir de användbara att återanvända
- Ta ett steg i taget – sätt upp delmål
  - Starta småskaligt, skala upp med återanvändning, samt förfina successivt alltefter efterfrågan.
  - Starta med det befintliga och låt ”systemet” växa med användningen.
- Undvik parallellkörningar, de skapar flerfald fler problem än antalet problem de löser p.g.a. att (de outtalade) förutsättningarna är olika.
- Bygg in simuleringsmöjligheter och eliminera tester.
  - Människor gör fel, lär av sina misstag, måste få pröva sig fram, hela tiden.
  - Slumpvisa tester slukar bara tid – nio av tio ger inget mervärde.
  - Eventuella tester skall vara välplanerade, d.v.s. ge ett givet svar ’ja’ eller ’nej’.
  - Ett säkert systembygge behöver inte testas. Det har alla utgångar fördefinierade.
    - Det enda som behöver testas, är om det fungerar eller inte fungerar!
    - Passera aldrig kontrollpunkter okontrollerat. Återstartsmöjlighet måste alltid finnas.
  - Det är användarens acceptans av det nya/förändrade systemets stöd till sin verksamhet som är helt avgörande.
- Minimalt med instruktioner, endast generella rutinbeskrivningar
  - Människor läser inte, än mindre förstår vad de läser.
  - De flesta människorna är händelsedrivna (reagerar på händelser), få är planerade inför händelser (proaktiv)
    - När oväntade händelser inträffar, gäller det att hinna fatta rätt beslut och att åtgärda inom rimlig tid före katastroftidpunkten.

**Christian Ekvall**

- Detta förutsätter extra systemstöd och/eller övningar.
- Hur ”systemet” används är individuellt och sättet förändras hela tiden
  - Information är färskvara
- Vad ”systemet” är måste vara självförklarande och det genom att användarna själva måste vara en del av ”systemet”. Det blir de när de själva deklarerar och definierar förutsättningarna, fakta, regler och modeller.
  - Väldefinierade data är långlivade, men kan förändras i tiden (i sju-årscykler).
  - Behörighet och integritet är fundamentalt för tillförlitlighet och förtroende.

Min lösning blev mycket okonventionell:

- Systemförutsättningarna var:
  - Ordersystemet skulle vara implementerat 1978.
  - Ett befintligt hållremsstyrt tio år gammalt ordersystem skulle avvecklas p.g.a. att leverantören ej längre ville göra systemsupport samt spritdupliceringstekniken gav en ohållbar arbetsmiljö.
  - Orderavdelningen skulle kunna decentraliseras.
  - Produktion på flera orter.
  - Möjliggöra kraftigt ökad produktionsvolym (med befintlig personal, blev mitt tillägg).
  - Ny produktgeneration skulle kunna hanteras, modulbaserad.
- Tillgängliga projektresurser var jag som systemerare, en kontorsrationaliserare som var väl insatt i befintliga rutiner samt en ovan projektledare. Med tiden fick vi tillgång till ett femtal programmerarresurser, totalt blev det ett dussin olika programmerare inblandade mer eller mindre.
- Det var framför allt fyra verksamhetskunniga som verkligen angav ambitionsnivån på systemstödet. Vid implementeringen av ordersystemet vinter - vår 1979 blev 40 personer utbildade av mig och kontorsrationaliseraren.
- Det fanns redan 20 terminaler på företaget, varav fyra på dataavdelningen. Detta projekt installerade 40 ytterligare terminaler. IBMs stordator hade 64kB i primärminne med möjlighet att ”swapa” (virtuellt minne) ytterligare 150kB. De modernaste hårddiskarna kom sommaren 1979 med 100MB styck, precis vad nya ordersystemet behövde.
- Databashanteraren inklusive transaktionshanteraren AROS/ROSAM var 1974 mer kompetent då, än vad Oracle:s eller MicroSoft:s motsvarigheter klarar än idag, avseende funktionalitet och prestanda. Påstådda brister är egentligen dålig insikt i hur en optimal systemarkitektur skall se ut, vilket är ett internationellt branschproblem. Detta systembygge är en hörnsten i mina egna framgångar. Det utvecklades av Christer Rosén, ursprungligen (60-talet) anställd på ASEA (idag ABB) i Västerås. Hans prestation då innebar att han blev utlånad till IBM i USA i slutet av 60-talet för att delta i utvecklingen av det som senare marknadsfördes av IBM som CICS/DL1. AROS/ROSAM är en vidareutveckling av dessa idéer där Christer förnämligt löste nätverkstrukturen i den hierarkiskt lagrade informationen samt med en avancerad transaktionshantering som möjliggjorde äkta on-line bearbetning mot företagsgemensamt data. AROS/ROSAM vidareutvecklade sin funktionalitet sedan i samarbete med bl.a. Scania.

**Året var 1976 då själva ordersystemets övergripande arkitektur kom på plats.**

- Skiktning av ”systemet” blir självklarheter om ”en sak bara skall finnas en gång”.
  - Antalet skikt är betydligt fler än tre! Men få människor orkar med vertikalt tänkande – en av fem orkar med högst tre stycken; den aktuella, den ovanför samt eventuellt den nedanför som oftast hanteras som ”självklar”.
  - Fler skikt får man genom att länka dessa tripplar med varandra över flera specialiserade människor som sällan klarar av att kommunicera med varandra –

Christian Ekvall

- de förstår inte varandra utan pratar bara förbi varandra, just p.g.a. nivåskillnaderna! ”Systemet” måste överbygga detta genom (generella) modeller.
- Dagens (2007) IT-folk är så långt inne i datorernas absoluta logik och uppfinner nya hjul hela tiden, därför alla andra hjul är så dåliga, att den efterfrågade fungerande helheten tappas bort, d.v.s. verksamhetsstödet såsom Informations-Systemet (IS), det som tillför mervärde för människorna i verksamheten och ”affären” med kunder.
  - Dagens (2007) IT-folk är dessutom så fixerade vid sina ”hjul” att de inte ser, än mindre vill förstå, det som borde lösas – det generella hjulet för alla hjul - det som gömmer sig bakom det som ”hjulet” är till för, nämligen någon lösning som kan snurra runt i all oändlighet, vilket är alla hjuls gemensamma **funktion** i efterfrågat resultat till verksamheten i en levererad och prissatt produkt till kund.
    - Senaste uttrycket för detta är SOA (System Oriented Architecture), men återigen är det alltför få som förstått essensen och hittat ”snurran”. Det mest anmärkningsvärda är att det är så många chefer, direktörer, myndigheter, nationer som accepterar och betalar för resursslöseriet som ger stressade och otillfredsställda människor samt en extremt ojämn kapitalfördelning. Hur kan man acceptera operativsystem som består av miljoner rader kod när det räcker mer än väl med några hundra.
  - Ordersystemets primära syfte var att stödja orderkontoren med att fånga en kunds/köparens krav och önskemål på produkt, vidimera det, beordra produktion och leverans av produkten samt att kunna fakturera den.
    - Möjlighet till förfrågningar och offerthantering var krav.
    - Beställningen skulle i huvudsak ske på egenskapsnivå.
      - Befintligt hållremssystem nyttjade maximalt för ett åttiotal egenskaper. Det nya systemet måste klara betydligt fler.
    - Ordersystemet skulle styra den kundorderdrivna monteringen av komponenter:
      - Komponenterna sammansattes av artiklar och förmonterade komponenter.
      - Sekvens- och kapacitetsplanering var tidskritiska och geografiskt spridda passager.
      - Sena orderändringar måste hanteras eftersom det var långa ledtider som en följd av produktens storlek och komplexitet.
      - Det var väsentligt att hålla isär den order som var kundens från den order som avsåg produkten och de order som avsåg komponentbeordringen.
      - Berörda parter måste vara direkt eller indirekt vara med i besluten när respektive order övergick i ny mognadsgrad.
        - ”Omstartsteknik” byggdes in som återstartade från senaste kontrollerade statusläge med hänsyn till nya ändringar.
    - Ordersystemet behövde vara tillgängligt åtminstone på kontorstid (on-line). Viss tid behövdes för tyngre bearbetningar på natten (batch-körningar).
      - Uppdateringar gjordes principiellt on-line utom de automatiska uppdateringar som kunde vänta till någon senare batch-körning.
      - Enklare informationssammansättningar gjordes on-line under förutsättning att svarstiden höll sig inom tre sekunder. I extremfall tilläts max 10 sekunder, men första respons måste vara inom tre sekunder och resten fortsatte i bakgrunden.
        - Med korta svars-/ledtider, baserat på icke-redundant dataöverföring får man automatiskt lägre kostnader – med låga kostnadskrav får sällan användbara systemlösningar. Observera att prispolitik kan ändras!
      - Tyngre och planerbar bearbetning gjordes på natten till nästa dag.
      - Tyngre ej planerbar bearbetning, t ex fakturering, gjordes ordervis med lågprioriterade program/transaktioner i on-linemiljö.

**Christian Ekvall**

- Små utlistningar gjordes på begäran på lokal skrivare för snabb distribution. Omfattande utlistningar på specialpapper, t ex kartong, gjordes i datorhallen.
- Sedermera möjliggjordes batchkörningar även i on-linemiljön med svarstider på 10 minuter.
- Frågespråk och rapportgenererande programfunktioner (planerbart och mallstyrt driftsattes 1980), trots förbud från datachefer, vilket löste mer än 50% av då efterfrågat informationsuttag. Idag (2007) genereras information till 85(!) andra system.
- Det gällde att hantera störningar så att de aldrig märktes.
  - Med konsekvensanalys och riskbedömning, avseende tillgänglig tid och resursåtgång för återställning, förfinades systemstödet för säker hantering.
- Det var oerhört viktigt att systemansvariga användare gavs möjlighet att planerat kunna finjustera sitt informationssystemstöd.
  - Parameter- och tabelldrivningen blev omfattande både avseende datordrift som systemdrift: rubriksättning, flera språk, valbara värdetexter, behörigheter, mm.
  - Driftstatistik insamlades löpande för att kunna trimma systemdriften och att kunna fördela och debitera driftkostnader.

**Året var 1977 då själva ordersystemets detaljerade arkitektur kom på plats.**

- Hela systemet är på utsidan traditionellt med transaktionsorienterade on-lineprogram respektive batchade programsekvenser. Dock är programmen uppdelade i programfunktioner, ibland i flera nivåer, som återanvändes genom länkning både här och där i de exekverande programmen.
- De olika orderkontoren var olika organiserade avseende arbetsrutiner. Detta löstes genom att alltid använda samma programfunktioner, men de länkades till varandra på olika sätt.
- Identifierande datastrukturer lades i databaser. (Sedermera gjordes även dessa dynamiska.)
- Den befintliga produktbeskrivningen var redan parametriserad åtminstone sedan 1968. Troligtvis fanns en hålkortsbaserad lösning före den hålremsstyrda. Det gällde för mig att vidareutveckla detta till att kunna hantera åtminstone 1000 parametrar per objekt.
  - En mycket stor styrka med dessa parametrar var att de var definierade för alla system över hela företaget och vårdades av produktspecialister inom respektive verksamhetsområde, d.v.s. inte som i traditionella system av programmerare som av naturliga skäl saknar insikt i företagets produktgenskaper.
  - Traditionellt brukade mask-/matchningsteknik användes, ibland även kallat beslutstabellteknik. Men jag upptäckte att det fort blev svårt att överblicka, vilket medförde att man sjufaldigt dubbellagrade samma parametrar och de blev därmed väldigt jobbigt att underhålla, då man inte klarade att göra kompletta förändringar. Med ett tre månaders halt i projektet, samt att jag under tiden listade ut lösningen till regeltolkningen, genomförde jag ett omvänt arbetssätt, vilket underlättade underhållet för produktbeskrivningen dramatiskt.
    - Jag fick mina kollegor på dataavdelningen att ”gå i taket” när jag påstod att befintlig beslutstabellteknik inte var användbar i on-linemiljö med svarstidskrav på 3 sekunder. Man ordnade ett stormöte med ett dussin inblandade personer inklusive systemchefer. Jag fick övertyga samtliga närvarande inklusive min egen projektledare om att den nya lösningen var en absolut nödvändig beståndsdel.
    - Jag upplevde också (1978) på ett högt beslutsforum att produktfolket inklusive systemerare hade klassat företagsledningens ambitioner att införa moduler (komponenter fanns redan) som ”omöjliga”. Detta preliminära protokoll justerade jag till ”möjligt”. Därefter hade jag fin kontakt med företagsledningen.

- Det visade sig också att även om parametrarna var grunddefinierade över hela företaget, så hade varje avdelning (konstruktion, försäljning, produktion, m.fl.) var sin egen begreppsuppfattning. Jag tillät dock att avdelningarna fick skapa sina egna parameterstrukturer mot att varje avdelning då måste definierade sig mot grunddefinitionerna. Sådana strukturer blev användbara i båda riktningarna. Varje avdelning kände sig motiverade att skapa en lokal beredningsorganisation redan när bara 80% av befintliga företagsparametrar räckte till.
- Denna teknik med att lyfta ur verksamhetens regelverk, som traditionellt fortfarande blir programlogik, till databaser gjorde att verksamhetens ansvariga kunde förändra systemets uppträdande på sekundnivå, som med åtföljande simuleringsmöjligheter gjorde att regelverken (den ackumulerade kunskapen) kunde göras driftmässiga på minuter eller den tid den organisatoriska beslutsprocessen kräver. Observera här att en verksamhets- eller produktförändring kan påverka flera olika ställen i ett regelverk, vilket gör att en speciell teknik krävs för att hålla ihop dessa punkter respektive införandeordning. I traditionella system måste en programmerare allokeras, normalt någon gammal erfaren nyckelperson, som måste få sin tid att förstå problematiken. Den tiden brukar normalt inträffa ett halvår senare i kombination med att problemet som skall lösas ofta har några komplexa undertoner. Dessutom skall ett dataprogram kompileras om till maskinkod, vilket gör att svårt att simulera på sekund/minutnivå...
- Jag har upplevt vid flera tillfällen under efterföljande år, förutom att ledtiden för systemutvecklingen dramatiskt minskat från halvårsvis till några timmar, hur kostnaden för anlitate programmerare minskat först med 70% och sedan med 99%. Även driftkostnaderna minskade med 70% med justerad systemarkitektur.
- Jag har upplevt hur buffertlager på verkstadsgolvet och hela system kring detta försvunnit med förändrad systemarkitektur. Jag har upplevt hur tio dagars materialledtid i verkstaden minskat till ett par timmar...

### **Året var 1978 då ordersystemets kärna och basala funktioner programmerades.**

- Parallellt med programmeringen påbörjades grundläggning av kunddata och produktdata.
  - Jag skrev 95% av programföresättningar. Jag var förbjuden att skriva programmen själv, så jag skrev på vanlig svenska istället för i COBOL.

### **Året var 1979 då ordersystemet implementerades och det driftsattes i juni.**

- Systemanvändarna utbildades i tre steg De var fyrtio stycken som skulle hantera och använda systemet:
  1. Användarna visste knappt vad en dator var. ABC80 eller PC var ännu inte uppfunnet. Utbildningen gjordes teoretiskt samt handfast på någon befintlig terminal under ett par timmar.
  2. En månad senare fick samtliga möjligheten att lära sig systemlösningen under 5 dagar vid "skolbänken". På de delar som var och en måste förstå fanns övningsuppgifter.
  3. Användarnas acceptanstest och mitt systemtest gjordes samtidigt med två insamlade äkta produktionsmaterial. Orderflödets förädlingsprocess genomgicks minutiöst hos varje systemanvändare som fått sina ergonomiskt korrekta arbetsplatser iordningställda genom (beordrat) fackligt engagemang (MBL-lagen var ju ny). Personliga användarinstruktioner till rutinerna printades ut (hard-copy) parallellt. Principiellt var jag aldrig framme vid tangentbordet utan stod bakom användarens stol tillsammans med programmerare (egentligen otillåtet) som hade order om att vara helt tysta utan skulle bara lyssna in alla suckar, stönanden och grymtningar. Jag besvarade användaren endast på direkt fråga när han/hon helt kört fast. Hjälpen bestod i att hjälpa användaren att själv förstå varför han/hon hamnat i situationen och hur man kan ta sig ur den och upp på spåret igen. Under tiden

använde jag också ett tidtagarur för att bekräfta att svarstiderna höll måttet inom 3 sekunder. Dessutom fick programmerarna bra exempel på justeringsbehov i sina program.

4. Användarna instruerades att aldrig bli arga på datorn, utan var de missnöjda eller arga skulle de höra av sig till mig, för det var jag personligen som satt bakom glasrutan på dataskärmen. Dessutom poängterade jag att
  - ”man får aldrig veta mer än man frågar”
  - ”man får svar såsom man frågar”.
5. När acceptanstestens exempel flöt smärtfritt till allas belåtenhet, beslutades en skarp driftstart som började någon måndag med en tom orderstock som fick växa vecka för vecka tills att hela orderstocken fanns inlagd i datorsystemet.
  - Principen var att verksamheten skulle starta med befintligt känt produktprogram och på det sättet lära sig den kunskapsbaserade modell- och regeldrivna systemlösningen.
    - i. Den gängse traditionella stansoperatrisbaserade hålkortslösningen hoppades över. Den användes endast för grundladdningen – observera att samma programfunktioner användes även för detta, men genom program som kunde läsa hålkort istället.

### Året var 1980 då ordersystemets alla grundfunktioner slutligen kom på plats.

- Frågespråk och rapportgeneratorm implementerades när det blivit lite volymer i orderdatabasen.
  - o Denna integrerade lösning gjorde sedermera att ”Data Warehouse”-lösningar var ointressanta.
  - o Generatorm var så intimt integrerad i systemlösningen att merkostnaden blev så låg att någon kostnadsdebitering aldrig gjorts.
- Historikhanteringen kom på plats. På grund av att livslängden på levererade produkter är mycket lång (mycket mer än 40 år) och reservdels-/servicebehov finns löpande, så måste historiska data sparas lika länge.
- Den väntade nya produktgenerationen skulle beskrivas. Kravet var att den skulle vara modulariserad.
  - o Nu hade vi lärt oss vad det nya ordersystemet klarade av. Nu tänkte vi till på hur en modulariserad produktbeskrivning skulle kunna se ut ”på data”.
    - Lösningen ur programmerarsynpunkt blev ”krångligare” och fler databasaccesser krävdes vilket gav ”längre” svarstider. Men det visade sig att den verkliga svarstiden inte blev längre, tvärtom.
      - Detta berodde på att den totalt lagrade generiska produktbeskrivande datamängden minskade med 80%, trots att antalet regler ökat jämfört med föregående produkt-generation,. Minskningen var väntad, men inte så stor.
      - Jag lyckades visa matematiskt för tillräckligt många fördelen med att skapa moduler som komplement till komponenter, genom att multiplikation har starkare momentum än addition.

**Systemet är fortfarande i drift (2007) i stordatormiljö och kommer att så göra några år till.** Jag blir mycket välkomnad ute på avdelningarna när jag kommer på besök hos dem. Vissa kallar mig för "Guru".

Administrativa system på 70- och 80-tal överlevde sällan 3-årsgränsen. Mitt system var för mig planerat att oförändrat fungera i 10 år. Och mycket riktigt dröjde det så länge innan verksamheten nådde taket för vald ambitionsnivå. Systemets kärna är fortfarande intakt, men användare har idag, med all rätt, större krav på funktionalitet idag.

I början av 90-talet gjordes en inventering över totala programstocken i stordatorn, samt statistik över dess belastning. Det visade sig att av totalt antal program, cirka 2000, var det bara en handfull program som stod för 80% av den totala belastningen. De var samliga tillhörande ovan nämnda ordersystem samt en modernare variant av dess regelmotor som driftsattes 1987, ett egentillverkat modellstyrt objektorienterat skal utanpå databashanteraren som användes mestadels till generiskt produktbeskrivande system.

1987 fick jag veta att mitt ordersystem hade en jämlike i dåvarande Digitals (uppköpt av Compaq, HP,...) konfigurator XCON som ansågs först i världen i sitt slag. Den innehöll 1990, den mest omfattande konfiguratorn i världen, 30 000 regler. Den utvecklades från 1978 och driftsattes 1982, d.v.s. några år efter mitt system. Motsvarande systemfunktioner började dyka upp hos bilindustrier först under de sista åren på 80-talet (GM). Traditionellt skrivs regler i programformat såsom IF...THEN...ELSE. Men det är ett förödande klumpigt beskrivnings sätt. De är hopplösa att underhålla och kapacitetstaket stoppar senast vid 6000 regler.

I samband med en systemkonvertering 1994, gjorde jag en uppskattning av totalt antal data-lagrade regler (rule sets) som var inblandade i kundorderspecifisering (ofta felaktigt kallat konfigurering som egentligen som ord behövs för någonting annat), konstruktionsgodkännande, artikelbehovsberäkning och monteringsbeordring. Ordersystemet hade då 150 000 regler och övriga produktbeskrivande strukturer tillsammans 850 000 regler. Ordersystemets svarstider var då under en sekund för varje stegning (cirka fem stycken) i specifikationen. Totalt antal teoretiskt möjliga produktvarianter var då uppe i  $10^{680}$ . Dock begränsade de 150 000 reglerna denna mängd till gissningsvis ett par miljarder tillåtna varianter. Och dessa regler förändras hela tiden varje dag av ett fåtal personer. Ett stort antal av dessa personer är så roade av sitt arbete att de stannat kvar med stolthet på företaget i åtskilliga år.

1997 gjordes försök att köpa ett "modernt" konfigureringsverktyg, som användes av Bang & Olufsen. Dock sprängdes det systemet redan innan vi ens hade första komponenten på plats.

## **Maskiner och tekniker**

Maskintyp och programmeringstekniker som jag har arbetat med är IBM-datorer och dess DOS och MVS-miljöer eftersom de dominerade inom tillverkande industrier på den tiden.

Dominerande programmeringsspråk inom administrativ databehandling var COBOL som därför var en allmän spridd kunskap.

COBOL i all ära, men, men:

- ”Allting” måste ju skrivas i samma program därför att annars var det ”svårt att läsa dumpar”. Själv praktiserade jag att länka programdelar.
  - o Av utrymmesbrist i primärminnet och av svarstidkrav mot diskminne skrev jag ett program på Electrolux som idag skulle kallas ”Cache”-minne. Exekveringstiden minskade från 40 minuter för ett likvärdigt program till 10 minuter.
  - o Jag återanvände samma programdelar/funktioner länkade till flera olika program, idag kallas det SOA.
  - o På Scania vidareutvecklade jag primärminnesutnyttjandet (64kB, med möjlighet att swapa till disk med ytterligare 150kB) genom att använda dynamisk och statisk länkning.

ASSEMBLER användes i två fall där COBOL kom till korta.

- o Dels för att återta kontrollen till ordersystemet från IBMs operativsystem, när det ville tvärstanna exekveringen (dumpa), även för mindre fel. En speciell funktion i ordersystemet utvärderade graden av allvarlighet. Den simplaste var vid ”0C4”-fel då operativet inte klarade av att hantera icke-numeriska tal i beräkningar. I dessa fall felmarkerades aktuellt fält i utskrifter samt felsignal skickades till en systemgemensam lista för senare åtgärd.
- o Dels klarade inte COBOL ”variabelt multi-fält”. Vi byggde en funktion, bl.a. bestående av ASSEMBLER, som kunde bygga och redigera innehållet i ett fält som presenterades på skärm eller lista bestående av utvalda databasfält enligt tabelldriven mall som systemansvarig i verksamheten själv underhöll.

Mina specifika val av maskin och programmeringsteknik blev:

1. IBM-datorer, eftersom de gällde inom tillverkande industrier på den tiden.
2. Ett nytt programmeringsspråk kan man läsa in på en vecka, men att lära sig att ”programmera” gör nästan inte någon.
  - a. De flesta ”systemutvecklingsmetoder” som finns är helt uppåt väggarna fel. De maximerar behovet av konsulter och konsultbolag. Det är systemarkitekturen för förväntat projektresultat är absolut viktigast!
  - b. När jag träffat andra specialister inom IT, har de alltid frågat mig om ordersystemets regelhantering var skrivet i PROLOG eller LISP! Mitt svar var då alltid COBOL, vilket är det korta svaret på den givna frågan men som inte förtäljer hur regelhanteringen egentligen var strukturerad. IBMs stordator och COBOL är väldigt klumpfotat sätt att lösa kunskapsstödet på, men det gick att genomföra med acceptabla svarstider. Den valda tekniken kunde ha trimmats ytterligare men det behövdes aldrig.

## **Kunskaper och färdigheter**

Krav på inhämtning av kunskaper, information och färdigheter för att bedriva mitt arbete har varit olika under skilda tidsperioder:

Systemlösningen på Scania 1977-79 önskades och krävdes. Jag skapade den ensam och motströms – precis som kärringen mot strömmen. Jag hade till och med projektledaren mot mig. Hela dataavdelningen behövde övertygas om nödvändigheten av annorlunda lösning. På den tiden blev man nöjda med att man fick en lösning som levererades i rätt tid till användarnas stora belåtenhet. Min egna personliga utvecklingsresa har varit att skapa en överskådlig struktur för människornas egna och gemensamma informations- och kunskapsstödsbehov. Ju mer komplext vuxit med antalet involverade människor, även geografiskt utspridda samt med högre krav på ledtider relativt tröghet i förändringar, ju finmaskigare har arkitekturen behövt bli. Ambitionen i den grundläggande systemstrukturen har hållit hela tiden. Det är behovet av utbyggd funktionalitet, baserad på redan utvecklade grundfunktioner, som behövt vara följsam till de utvecklingssteg som mognat fram stegvis, cirka vart sjunde år.

1980 blev jag systemerare åt systemerare, d.v.s. 'Dataadministratör' – inte 'Databasadministratör' som var gängse på den tiden, vilka vårdar databaserna till någon specifik databashanterare. Inte ens min närmaste chef förstod skillnaden. Programmerare kan sina program som hanterar data. Systemerare skall kunna hur databaser interagerar med program för att inhämta och leverera information till användare. Och detta skall ske inom givna svarstidskrav.

1978-1980, 1986-1988 samt 1994-1995 var jag kunskapsingenjör, d.v.s. hjälpte systemanvändare att strukturera och ladda kunskapsdatabaser. 1986 började jag använda "induktionsteknik" som AI-verktyg (Artificiell Intelligens) i detta arbete. 1989 kom jag nära "neurala nät", "genetiska algoritmer" och "fuzzy logic", men fick aldrig möjlighet att praktisera teknikerna.

1984-1988 var jag arkitekten bakom företagsledningens Informationsstrategiprojekt.

1995 blev jag utnämnd till "Senior Systems Analyst", en av två experter inom fackområdet samt expert på Scantias verksamhet. Min specialitet är "Product Data Descriptions, Structures and Rules".

2006, sista december, pensionerade jag mig.

Mina internationella kontakter har under mina yrkesverksamma år varit (länder, institutioner (högskolor, företag, myndigheter), sammanhang och funktioner):

Några exempel:

- Jag har varit på en veckas kurs med Charlie Bachman i London 1984.
- Jag etablerade ett samarbete med Computas, tillhörande Norsk Veritas i Norge 1987-88 och träffade t ex en dansk professor i AI-teknik och personer som jobbat vid CERN i Schweiz.
- Jag har ätit lunch med Edgar F. Codd i samband med ett seminarium av Oracle.
- Jag har varit på en veckas kurs i Objektorientering av Bill Olle/James Martin i Rom 1990.
- Har deltagit i och påverkat ISO10303 (STEP) 1995-1997 tillsammans med Volvo, Mercedes, Volkswagen, Porsche, General Motors, Renault, Peugeot, Citroën, Ford, Chrysler, m.fl.
- Har deltagit i EUREKA-projektet TIME 92-97 och reste runt i Europa till industrier, universitet och industridepartementet i Paris med Professor Erik Agerman från KTH. Han har erfarenhet från tidigt 70-tal på ASEA/ABB i Ludvika med regelbaserade produktbeskrivande tekniker kring högspänningstransformatorer. Träffade då också folk från Toyota.
- Jag hade så detaljerade frågor om ett webb-baserat regelhanterat verktyg, att jag fick resa till deras utvecklingskontor San José i Californien 2003.

Jag har bara verkat för kompetensförändringar inom ditt område inom Scania.

Det är en specifik konst att bygga kunskapsdatabasers modeller och regler på ett sådant sätt att de kan underhållas av verksamhetskunnig personal på ett tillförlitligt sätt samtidigt som kunskapshanteringsystemet skall ha relevanta svarstider. Även strukturella organisationsförändringar har genomförts.

## **Synen på dator- och datateknik**

Min erfarenhet och starka åsikt pekar på att 99% av allt dataarbete är fullständigt onödigt. Jag brukar inom varje upplevt problemkomplex kunna lova en rationaliseringspotential på 80% utan att studera problemområdet närmare. Tyvärr är tal större än 5% - 10% inte trovärdiga för beslutsfattare. Dessutom är alltför billiga lösningar inte heller trovärdiga för direktörer. Kunskapsbaserade administrativa system har relativt stor investeringskostnad för regelmotorn och grundladdningen, så stor att företagets styrelse kan blir inblandade. Därtill är systemtypen tidskrävande att införa, minst tre år mest p.g.a. mänsklig mognadsutveckling och konsensus i grupp går inte fortare, vilket upplevs som en oändligt lång tid för kvartalskapitalister. Men investeringen brukar betala sig på första implementeringen inom två år. Därefter är ytterligare implementeringar ”gratis” enligt modern ekonomisk kalkylering! Det svåra är dock att visa på alternativkostnaden, för detta berör områden som inte brukar kostnadsföljas. Men det brukar bli väldigt stora företagsvinster när samma berörda personal blir mer än 5 gånger effektivare. Det handlar om att systematisera konsensus. Och med fler inblandade verksamheter accelererar nyttan och därmed vinsten exponentiellt.

Den känsligaste delen är att hårt styra upp ansvar och befogenheter med ramstyrning så att flexibiliteten finns kvar och därmed kreativiteten får utrymme utan att ohämmat kunna springa iväg. Sedan 90-talet så är tekniken inte bromsen längre utan människans förstånd är bromsen. Jag är också förvånad över konservatismen och trångsyntheten även hos datafolk, som anser sig vara kreativa.

Jag är störd över den låga ambitionsnivå som ges i utbildningen. Det är än en gång, på tok för mycket fokus på processer, nu upplever jag en fjärde omgång under min snart fyrtioåriga yrkesbana. Man fokuserar på att göra saker rätt, men det är inte ett vinnande arbetssätt i långa loppet, bara bra för problemsökning. Det är ett arbetssätt som engagerar och ger förhoppningar, men man kommer aldrig in i mål någon gång, för det är bara en hägring man får se. Vad som händer är att riskkapitalisterna sätter stopp och för att nå ett delresultat så forcerar man en konserverande lösning på gamla uppfattningar utan att höja ribban till nya möjligheter. Man tar sig inte tid att tänka till på att göra rätt saker, än mindre tydligt uttrycka sina motiv. När sedan det ”nya” systemet kommit på plats där man i bästa fall sportat upp de trånga passagerna, kommer användarna redan andra driftdagen att vilja ha förändrad funktionalitet. Men eftersom datafolk inte fått lära sig hur man bygger flexibla system som kan förändras löpande, så blir det nya systemet snabbt ”spagetti” det också, som skapar behovet att byta ut även det nya inom ett år. Men med ett nytt traditionellt projekt kan användarna få ett nytt system tidigast efter tre år! Datafolk, färskas från skolan, vilka jag alltid intervjuat, kan i praktiken inget om äkta databaser, de kan bara relationstabeller som är, enligt min starka åsikt, bara ett intressant presentationssätt men totalt odugligt för datalagring vilket märks vid större volymer, särskilt när dessa relationstabellverktyg indexerar ihjäl sig. Än mindre har datafolk klart för sig vad Data Dictionary – Directory – Repository har för roll i ett systembygge. Och det största hotet är den nonchalanta hanteringen av data. Få har klart för sig skillnaden mellan information och data. Här har vi den Gordiska knuten. Det är få som vet hur man öppnar den. Skillnaden är densamma som skillnaden mellan kvantitet och kvalitet. Och förstår man inte skillnaden, ser man heller inte den kraft som ligger gömd i kunskapshanteringen. Då först upptäcker man att pudelns kärna egentligen är definitionerna av data vilket bl.a. inkluderar värdemängden och hur man måste hantera den. Det viktigaste värdet är ”tomma mängden”. Hur många har lärt sig den i skolan? Mängdläran är Akilleshälen i informations- och kunskapshanteringen! Då kommer vi att upptäcka hur svaga begreppsbyggnader människor har, och att de flesta människor agerar på troende istället för vetande. De flesta människorna bara kastar ur sig en massa (tvetydiga) ord som de knappt förstår själva, och de kontrollerar inte ens hur budskapet tas emot eller tolkas av åhörare. Sen blir man förvånad över reaktioner eller uteblivna reaktioner!

På senare år har Extrakt – Data Warehouse-fenomenet fått ett nytt ”Buzz-word”, nämligen ”Beslutsstöd”. Återigen har icke-förstå-sig-på-are varit framme och förstört ett ord. Dessa systemanvändare håller bara på med att sammanställa ny statistik ur historisk information som är skapat

med tveksam kvalitet. I bästa fall kan de bara få fram ett underlag till mänskliga beslut. Statistik är väldigt beroende av källans kvalitet, annars går det att "bevisa" vad som helst.

Jag har ägnat mig åt **äkta beslutsstöd** sedan 1977. Ovan nämnda "beslutsstöd" bakade jag in i mitt frågespråk- och rapportgenerator 1979-1980. Ett äkta beslutstöd används vid t.ex. order-specificeringen som handlar om att ge stöd, exakt på sekunden vid behovstillfället, för att godkänna/besluta att en specifikation är acceptabel (kompletterad och validerad) baserat på fastställt regelverk av tidigare insamlade erfarenheter, gärna av flera berörda parter. Detta är fundamentalt väsentligt, som en stomme i tillförlitlig kvalitet, att ägna sig åt rätt sak innan man gör saken rätt. När detta också baserar sig på lokal acceptans relaterad till ett och endast ett gemensamt data inom fördefinierad organisation, ju större desto bättre, så frigörs en enormt stor informationskraft. Jag brukar jämföra skillnaden i storlek med att förlösa energi med att elda ved eller med att starta en kärnreaktion.

Dator- och datateknikens status har förändrats, anser jag.

Tekniken har kommit för att stanna för den ger oanade nya möjligheter. Hittills har den tagit över mycket i informationsspridning, inom företag, i samhället och personligt: orderhantering, fakturering, reklam/spam, e-mail, hemsidor, skattedeklaration, m.m. Det är oerhört effektivt för företagare som inte längre behöver betala porto för sina utskick samtidigt som utskicken skickas på några sekunder jorden runt och svar kan komma lika snabbt om de inte sover på andra sidan jorden. Det är en ny kommunikationsmöjlighet jämfört med en beriden postiljon med brev på 1600-talet, senare tåg- och bilburen och telefon- och faxtekniken från förra århundradet, vilka har fått en kompletterande möjlighet tillsammans med internet.

Den enskilda människan "drunknar" redan i informationsfloden och sitter helt ensam framför sin PC och blir frustrerad över långa svarstider, virushot och att allt är krångligt och ingenting fungerar. Och inte vågar man fråga någon eller säga något, för då kan man ju "avslöja sin dumhet" och "tappa ansiktet" eller stämplas som "besser-wisser", enligt jante-lagen.

Eftersom tekniken är mer eller mindre överallt idag, så fungerar inte ett modernt samhälle utan den. Den är starkt beroende av tillgång till säker elektricitet. Samtidigt är informationskvaliteten så låg att datafolk snart hamnar i strykclass. Som tur är det okunniga kundmottagare som i regel får ta den "smällen" och stressen.

MicroSoft-tekniken har t.o.m. hamnat i bilars realtidsystem, den teknik som är mycket instabil i vanliga PCn! Ansvariga beslutsfattare är otroligt blåögda för konsekvenserna. För några år sedan t ex så fanns en datoriserad (chip) farthållare. Det visade sig att när den "dumpade" så blev det med "gasen i botten", med den följd att bilen "skenade" okontrollerat. Få människor hade kallsinnet att stänga av motorn genom att vrida tillbaks nyckeln för att sedan kunna vrida på igen (reset), vilket troligtvis hade nollställt chipet.

Den nya tekniken är så pass komplex att bara ett fåtal människor kopplar verklig orsak och verkan.

Sedan även internettekniken blev allmän under 90-talet ser jag inga tekniska begränsningar längre, utan endast mänskliga attityder, bristande fantasier och kompetensbegränsningar, för möjligheterna att med datateknikens hjälp förbättra mänsklighetens förväntade behov av kvalitativt kunskapsutbyte för effektivare resursutnyttjande, vilket är en högre ambitionsnivå än bara informationsutbyte.

## ***Livet i övrigt***

Mitt liv utanför ditt yrkesarbete:

- Min kära tycker jag sitter alldeles för mycket framför min dator, där jag samlar allt.
- Läser inte så många böcker, men många tidningsartiklar i t ex Forskning & Framsteg, Illustrerad Vetenskap, Populär Historia och Svenska Dagbladet. Jag lyssnar mestadels på P1 och föredrar radio- och TVprogram som är vetenskapliga, naturinriktade eller dokumentära.
- Se också nästa punkt.

Mitt familjeliv, mina fritidsintressen, föreningsengagemang:

- Är skild och bor ihop med min sambo sedan 1990. Hon har två döttrar och fyra barnbarn. Jag har fyra barnbarn också. Dessa fångar mycket av vårt intresse. I övrigt reser vi ofta, lyssnar på musik på olika sätt och är ofta ute och promenerar. Vi tycker om ett vackert hem och god mat.
- Vi har också ett handelsbolag tillsammans som ger tillgång till miljövänliga produkter från ett internationellt företag som bygger på anti-jante-principer!
- De gånger jag engagerar mig i föreningsliv, t ex bostadsrättsföreningar, brukar jag vara kassör eller revisor.
- Jag älskar att köra bil, det är avkopplande. Ju mer snö, desto roligare blir det.

Mitt yrke har påverkat valet av bostadsort. Yrkesvalet gav begränsat antal arbetsgivare, privata större industriers huvudkontor. Om det bara fanns en sådan arbetsgivare på orten och avståndet till nästa var mer än en timmes restid i enkel riktning, måste jag byta bostadsort om jag inte skulle trivas med arbetsgivaren. Antalet orter var därmed mycket begränsat. Dessutom var jag huvudsakligen uppvuxen i Stockholmstrakten och hade de flesta nära släktingarna där.

Mitt liv idag:

- Är nybliven pensionär och startat min enskilda firma i avsikt att få friare arbetstider. Enligt min doktor är min hälsa på topp.
- Någon dag i veckan roar jag mig med att vara ”Demochaufför” på Scania, d.v.s. sitta bredvid en potentiell lastbils- eller busskund eller annan intressent som försöker framföra t ex ett 60 tons fordon på 25 meter och vara beredd att ta tag i ratten eller handbromsen, om körningen går över styr.
- Ibland lånar jag en lastbil och åker till mina barnbarn som bor på en bondgård i Värmland.
- Jag läser och funderar mycket om dagens snabba upptäckter, om och kring genomet och hur liv, särskilt mänskligt liv, uppstår och fungerar. Jag hittar förvånansvärt många likheter i livs-systemets arkitektur, med en åtminstone en miljard års erfarenhet på jorden, med mitt Datasystems arkitektur.