Informatik — Et nyt Paradigme ?

Forstår vi det ret, og dermed ret at udnytte?

Dines Bjørner

12 November 1998

Synopsis: *Medierne kalder det informations--teknologi. Fagfolk nuancerer det og bruger relaterede, men forskellige termer: Datalogi (læren om de størrelser, der kan eksistere inde i datamaten og hvorledes de konstrueres)), datamatik (realisering af programmel på maskinel), informatik (matematik ⊕ datalogi ⊕ datamatik ⊕ anvendelser) og informationsteknologi (som primært omfattende datamatik, data-kommunikation samt alt det (især) elektroniske udstyr der kobles til datamater: sensorer og aktivatorer). Ordene dækker dermed over tilgrænsende og omsluttende, men dog distinkte fænomener. Termen informations--teknologi dækker, påstås det her, i gængs tale, et noget begrænset aspekt af det som datamater idag beror på og afstedkommer. Mao.: vi skelner mellem den mere omfattende "definition" givet ovenfor og hvad lægfolk, herunder medierne, almindeligvis forstår herved. Der er derfor brug for en mere nuanceret forståelse. Den del af den Danske befolkning, der idag beskæftiger sig med det område, der udgøres af de samlede områder der dækkes af termerne datalogi, datamatik, informatik og informations--teknologi, omfatter sikkert langt flere personer end der idag beskæftiges indenfor sundhedsvæsenet, eller indenfor det givetvis langt mindre område: fiskeriet. Dog nyder disse sidste områder en langt større politisk oa. bevågenhed: Større forsknings-midler, mere nuanceret opfattelse af feltet, oma. Disse, oa. discipliner repræsenterer en lang udvikling. Informatik — vor samlende betegnelse for de fire termer nævnt tidligere plus disses anvendelse på forståelsen af de genstands-områder i hvilke der bruges datamatik — betegner et næstent revolutionært paradigmeskift.*

Dette indlæg vil handle om dette: skiftet fra tidligere professioners materielle univers af kvantitet, målbare størrelser, til nutidens endnu ikke tilnærmelsesvis erkendte, og til den nærmeste fremtids intellektuelle univers af kvalitet.

• • •

Jeg har arbejdet indenfor informatik siden 1962: 36 år. Og jeg har været heldig: at vokse op i Danmark hvis datalogiske tradition, bl.a. beroende på nyligt pensionerede Prof. Peter Naur's banebrydende indsats i 1960'rne, er endog meget fin; og — i perioden 1962–1975 — gennem ved IBM at arbejde sammen med adskillige af verdens da førende dataloger: med Gerald Weinberg, vel den første, der indså vigtigheden af psykologiske aspekter i programmeringen; med Gene Amdahl, med-skaberen af IBM's 360 datamaskin-arkitektur; med John W. Backus, skaberen af det første egentlige bruger--vendte programmerings-sprog FORTRAN, og medskaber af ALGOL 60; med Ted (E.F.) Codd, skaberen af RELATIONS-DATABASE begrebet, som mange idag møder under navnet SQL, og med en spændende gruppe ved IBM laboratoriet i Wien: ledet af Heinz Zemanek og "bestykket" med bl.a. Cliff Jones, Peter Lucas og Kurt Walk.

Det var ikke tilfældigt at dette sidste forsknings-center lå i Wien: her skabtes i 1920'rne—1930'rne Wiener-kredsen omkring lingvistik og filosofi, med Schlitz, Carnap oa., og med Wittgenstein og Popper som kritiske iagttagere.

• • •

I den hidtidige tekst har jeg skiseret nogle få fikspunkter. I det næste skifter jeg tilsyneladende "retning".

• • •

I den ingeniørmæssige, teknisk/videnskabelige og moderne, professionelle udvikling af endog meget store programmel-systemer er det slående, at det vi frembringer stort set alene er skriftlige dokumenter. Vi laver ikke egentlige tegninger — ikke sådant rigtigt. Selvom vi iblandt laver nogle diagrammer, tjener de mest som en slags mellem-regnings-notater før vi fortsætter (hvor vi slap) med tekstuelle beskrivelser.

Først beskriver vi omhyggeligt det **domæne**, der eksisterer før vi begynder, subs. begyndte, at tænke på datamatisering. Når vi omhyggeligt har beskrevet dette (fremtidige datamatik--anvendelses-) domæne i mange, dog, desværre, sikkert ikke alle det parts-haver perspektiver, og ej eller alle dets komponenter og disses funktionalitet, de (øjeblikkelige) begivenheder de muligvis måtte være underlagt, og de (tidslige) opførsler de udviser — når vi omhyggeligt har beskrevet mangt og meget i dette — da kan vi så småt tænke, med reference til disse domæne--beskrivelser, på at skitsere, siden detaljere, de

krav vi måtte ønske at stille til evt. datamatisering. Og endelig kan vi påbegynde selve **programmel**-udviklings-processen. I denne process, som i de tidligere nævnte domæne– og krav-specifikations-udviklings-processer, kan der let blive tale om adskillige trin af forfining.

Fra bygnings-arkitekten, f.eks. Utzon, der først skitserer et operahus og hans tegnestues mange arkitekters' rentegning og arkitektoniske detaljering, **via** bygnings-ingeniørernes mange tekniske trin: *fra* de der vedkommer de bærende konstruktioner — og beregninger herpå, *via* de, der drejer sig om varme–, vand–, sanitets– og ventilations-installationer — og beregninger "over" disse, *til* f.eks. elektricitets-installationer — **til** bygnings-håndværkerners fysiske arbejde, således er der også mange trin af datalogisk udvikling i forbindelse med programmel. Men her "skiller vandene": hvor de tekniske konstruktørers "sidste" tegninger af — og beregninger på (eller over) — bygnings, eller mekaniske, eller elektriske anlæg overgår til håndværkernes påfølgende arbejds--intensive arbejde, dér kan sidste trin af datalogens, cum programmørens beskrivelser ligge direkte til grund for eksekvering på en datamat: vores beskrivelser går fra de meget abstrakte, ofte i småbitte trin, og mange, til direkte at kunne danne grund for elektronisk bearbejdning! Vores beskrivelser er sprogligt tekstuelle — og kan narre betragteren ved, tilsyneladende at udgive sig for at være mere "alment sproglige" end andre ingeniørers umiddelbart mere formaliserede tegninger og diagrammer i forbindelse med hvilke det også tilsyneladende er nemmere at forestille sig tilknyttede matematiske beregninger.

Sådan lidt slagords-agtigt udtrykt: ALT DATALOGERNE GØR ER AT BESKRIVE.

• • •

Datalogi beskæftiger sig med egenskaber ved de størrelser, der kan "eksistere" inde i datamaten og hvorledes de konstruerers. Det er ikke naturlove, som for bygnings-, kemi-, elekronik- - og maskin-branchen, der bestemmer disse "størrelser". De skal mao. ikke i første omgang adlyde naturlovene. Istedet skal de adlyde matematiske lovmæssigheder. Der er ting der kan beregnes. Og der er ting man ikke kan beregne. Man kan foreskrive en beregning der f.eks. kontrollerer og analyserer et finansregnskab, og et program for en sådan beregning kan være fejlagtigt bl.a. derved at det "går i løkke": aldrig kommer ud af stedet, kører rundt og rundt. Men man kan ikke udvikle et andet program, der undersøger om programmer, som f.eks. det forrige, spilder tiden, går i løkke, og aldrig bliver færdigt. Gør det rigtige, det man forventer. Der er mange andre programmer man ej heller kan skrive.

Herved begynder forskellene at melde sig: forskelle mellem sædvanlige teknisk/-videnskabelige frembringelser, der beror på naturlovene, eller, som først nævnt nu: på finansbogholderi– & projekt-styring, der gerne skulle reflektere god praksis indenfor deres domæne: økonomi og ledelse, og datamatik. De første første beror på deres vel-etablerede fag-discipliners tradition, regler og love. De sidste, basalt på matematikken.

Sædvanlige teknisk/videnskabelige frembringelser kan objektivt "måles & vejes" — i fysisk forstand. Det nærmeste objektive man kan komme hertil hvad angår datalogens frembringelser er at afgøre om programmer opfylder visse egenskaber: At ræssonere sig frem, vha, matematisk argumentation. Selvflg. kan man "måle & veje" egenskaber ved udførelse af programmer, deres resource-forbrug: Tid og lager-plads, de maskinelle resourcer de lægger beslag på, men det er værd at bemærke at man operations-analytisk og teoretisk datalogisk kan (kompleksitets-teoretisk) ræssonere sig frem til langt mere dækkende estimater for ("målbart") resource-forbrug

• • •

Men hvis vi fortsat alene fokuserer på matematikken — som en forudsætning for korrekte og bruger-orienterede programel-systemer, ja så begår vi en udeladelses-synd. Foruden at opfylde matematiske — herunder især matematisk-logiske og algebraiske love — skal programmel-systemer også, som *fod-i-hose, hånd-i-handske,* reflektere deres genstansområdes lovmæssigheder. Hertil tjener at vi — som først nævnt — omhyggeligt beskriver dette (genstands–) domæne.

Ligesom det er utænkeligt, at man sender en satellit i omløb uden først at have bygget videre på Newton's celestiale mekanik (vedr. planetare bevægelser) og matematisk beskrevet satellittens mulige baner (trajektorier), således bliver det alt mere uprofessionelt at udvikle for eksempel programmel til understøttelse og delvis eller fuldstændig styring af jernbane-signallering — før man tilsvarende omhyggeligt og matematisk har beskrevet — og dermed givet et forløbigt, tilstrækkeligt svar på spørgsmålet — *hvad er et jernbane system (á la DSB & Banestyrelsen)* hhv. *hvad er en bank, et sundhedsvæsen, lufttrafik,* oma.

• • •

Ikke alene skal vi dataloger, i snævert samarbejde med de i genstands-området aktive medspillere: ejere, ledere, funtionærer, øvrige ansatte, kunder, leverandører, de regulerende myndigheder, oma., set ud fra hver af deres synsvinkler, beskrive **domænet**, men vi skal også tilsvarende omyggeligt beskrive, både fortællende (dvs. ikke-formaliseret) såvelsom som formaliseret — så meget, det vil egentligt sige ret så lidt, som vi nu er istand til det med dagens indsigt, de **krav** disse parts-havere måtte stille, de forventninger de måtte have. Og fra kravene skal vi så konstruere **programmel**, der opfylder kravene.

Hvor eletronik–, maskin–, bygnings– eller kemi-ingeniøren i mangt og meget, og ihvertfald traditionelt, har beroet på at frembringe materielt og fysisk (dvs. kvantitativt) måleligt (effektive) produkter: billigere, mindre, hurtigere, med større kapacitet, osv., dér er det nu sådan at datalogen forventes at frembringe programmel, der primært er kvalitets-betonet: Det skal være korrekt mht. krav — og denne form for korrekthed kan være svær at bedømme ved første brug. Det skal være "rart" at bruge — og dette "rare" har ikke meget med fysiologiske oa. umiddelbart målelige størrelser at gøre.

Måske læseren nu kan se linien: fra de materielle, umiddelbart fysiologisk sanselige (synbare, hørbare, lugtbare, følelige, mm.) kemiske, maskinelle, elektroniske og bygnings-mæssige kvantiteter — større, mindre, hurtigere, billigere, osv. — til de intellektuelle kvaliteter: korrekte, brugervenlige, anvendelses-affine, oma. Fra teknisk/ingeniør-videnskabelige kvantiteter, der beror på naturlovene, til datalogiske kvaliteter, der i første omgang beror på matematiske love.

• • •

Men vi har blot beskrevet eet, det første, dog vigtigste aspekt som dette indlæg skal omhandle.

• • •

Det blev nævnt ovenfor, og det skal nu blive hovedlinjen i det flg.: At ikke bare skal programmel adlyde matematikkens love, men det skal være funktions-dygtigt mht. et bestemt anvendelses-**domæne** — som derfor skal beskrives, analyseres og forstås, sådan nogenlunde rimeligt, førend vi begynder at tænke på **krav** til **programmel**, der skal yde funktioner i dette domæne.

På sæt og vis burde der egentligt ikke være noget ("revolutionerende") nyt i dette: Operations-analytikere har siden 1960'rne leveret matematiske modeller, af for eksempel transport, der så har dannet grundlag for konstruktion af komponenter i transport-systemer: Tidtabeller, bredde på motorveje, oma. Process-styre-ingeniører har siden 1940'rne leveret matematiske modeller af (typisk) mekaniske og elektro-mekaniske processer (turbiners optimale drift, lokomotivers energi-besparende opstart og nedbremsning, mmm.) — og disse modeller har dannet grundlag for passende styre-teknik (automatisering).

Men der er alligevel noget nyt. Vha. datamatikken: programmel plus datamaskiner, kan "processer", der ikke alene kunne beskrives vha. operations-analytiske eller alene ved styre-teoretiske metoder nu beskrives samlet vha. datalogiske metoder — og disse datalogiske metoder kan bringe de tidligere professioners konstruktioner sammen, lade dem "arbejde" sammen, samt bibringe dem nye elementer. Før var det svært at lade operations-analytiske planlægnings-rutiner veksle med styre-teknisk overvågning og indgriben. Nu, vha. datamaten, og de i den indlagte programmer, er det faktisk "såre enkelt"! Og dog gøres det tit og ofte altfor kompliceret.

Hvor de tidligere discipliner hver for sig behandlede et snævere gebet, kan datalogien bringe disse oma. sammen og, foruden at gå i dybden, også gå på tværs: koble de tidligere fænomener sammen, lade dem veksle, overgive observation og styring fra een til den anden — og meget mere.

• • •

Og nu er det så at vi kommer til konklusionerne.

Det er idag muligt, vha. datalogiske metoder — principper og delvist matematisk baserede teknikker og værktøj — at "tackle" store, tilsyneladende komplekse, dvs. mangfoldigt sammensatte genstands-områder. Hvor vi tidligere så program-pakker der enkeltvist "tacklede" egentligt ret så "små-bitte" genstands-områder, ja der vil vi alt mere, vha. den moderne datalogi, kunne tackle større infrastruktur-komponenter: en hel transport-sektor, et helt sundhedsvæsen, den samlede offentlige administration sådan som borgeren oplever den, en hel fremstillings-industri, en hel finans-sektor, oma. Alene blot dette at tackle Danmarks jernbane-system: med Banestyrelsen og DSB som de to store komponenter er idag "lade-sig-gørligt"!

Man kan forudse en interessant ny periode i informatik-forskningen og udviklingen, en periode i hvilken forskerne arbejder snævert sammen med fagfolk indenfor hver af en lang række genstandsområder i bestemmelsen, den omhyggelige beskrivelse og analyse, af infra-struktur-komponent efter komponent. Sådan sagt lidt vel dramatisk: Hvor det svindende århundrede begyndte atom-fysikken og endte med det menneskelige gen-bestemmelses-projekt, der kunne man forestille sig at begyndelse af det næste århundrede bl.a. blev præget af endog meget omfattende sådanne infra-struktur-bestemmelses-projekter. Hvad med f.eks. systematisk at forsøge "fastlægge" alle (?) arbejdsgange i sundhedsvæsenet: de komponenter hvorom disse drejer sig (patienterne oa.), de funktioner der udøves, de begivenheder der opstår, og de opførsler der finder sted. Hvis man kan systematisere "totale" aspekter ved et jernbane-væsen, ja så må det blive magtpåliggende at forsøge sig på hvor langt man dog kan nå indenfor een af dagens samfunds (andre) "smerte-børn": Sundhedssektoren.

Men er disse infrastruktur-komponenters "ejere", subsidiært top-ledelse, idag "klædt på" hertil? Har De den fornødne "informatik pli". Besinder De sig på Deres "besøgstid"? Tilsvarende: Er den eksisterede programmel-industri, herunder mange konsulent-firmaer, "klædt på"? Det synes ikke så. Langtfra.

Forskerne er parate. Vi sender mange velforberedte kandidater ud, men får de en ledelse, der forstår deres egentlige værd? Jeg tvivler stærkt herpå.

• • •

Danmark har en dejlig overkommelig størrelse. Det burde være nemt at komme i samtale, at skabe kontakt. Det er iblandt pinligt at store dele af Dansk datalogisk forskning sættes mere i agt udenlands end herhjemme. Det er altfor ofte beklemmende at se to-tre (måske fire) gange så store forsknings-beløb blive brugt indenfor en relativ lille infrastruktur-komponent (som f.eks. Dansk fiskeri) som på informatik.

Måske det alt sammen er ganske naturligt: *TTT — ting tar' tid.* Måske vores efterkommere vil se store beløb anvendt på informatik-forskning, og måske det vil ske på et tidspunkt, hvor det ikke længere er så presserende!

Informatik pli, at forstå og at besinde sig på informatikkens væsen: dets beroen på *hvad der kan beregnes,* på den matematiske logik og moderne algebra, på formalisérbare, ikke altid nødvendigvist [fuldt] formaliserede, abstrakte modeller af **domæner, krav** og **programmel**, at kunne ræssonere — argumentere, "kalkulere" — baseret på, dvs. "over" sådanne modeller, disse og andre informatik facetter — informatik dannelse — vil nødvendigvis blive en del af enhver uddannelse — sådan som læsning, skrivning, regning, matematik og fysik var og er det !?

Hvortil indledningens personlige biografi? Jo: De mennesker jeg arbejdede for var ikke alene "teknologi"-pionerer, de funderede også over tingene. Foruden at vide meget og foruden store færdigheder, repræsenterede de holdninger — mao. var dannede fagfolk. Mediernes behandling af det nye paradigme: informatikken fokuserer stort set alene på de fysisk målelige teknologiske færdigheder.

Med dette indlæg har jeg forsøgt skitsere blot nogle få, indledende facetter ved informatikken — sådanne som også peger i retning af viden og holdnings-dannelser.

Dines Bjørner er professor på Danmarks Tekniske Universitet (1976-...) og arbejder med metodelære indenfor informatik: datalogi ⊕ matematik ⊕ anvendelser. DB har arbejdet i 13 år ved IBM's udviklings- og forsknings-laboratorier i Sverige, USA og Østrig (1962–1975). DB var første og grundlæggende FN Direktør for FN Universitetets' Internationale Institute for Programmel-teknologi i Macau (1992–1997).