

Jarle Brosveet

DATAFAGET - EN NYTTIG
ELLER BEVISST UNYTTIG DISIPLIN?

STS-arbeidsnotat 18/92

ISSN 0802-3573-62

arbeidsnotat
working paper

1. INNLEDNING

Hva er nytten av å utvikle datafaget som en egen disiplin? Er datafaget et selvstendig fag eller en del av andre fag? Hva er i tilfelle fagets egenart som skiller det fra andre fag? Er et av de fremste kjennetegnene at faget er preget av en bitfiklerkultur befolket med mennesker som lever i sin egen teknologiorienterte verden uten sans for reelle problemer?¹

Slike problemstillinger dukker opp fra tid til annen. Jeg tar ikke opp alle disse forholdene i denne rapporten, men konsentrerer meg mest om fagets opprinnelse og utvikling som akademisk disiplin. Denne utviklingen blir sett i sammenheng med tanker om nytteaspektene ved forskningen, som enkelte hevder er et vesentlig vilkår for faglig utvikling i våre dager. Enkelte går så langt som til å si at stor samfunnsnytte er en forutsetning for den fortsatte utviklingen på de fleste fagområder som universitetene dekker. Faren er at en slik betraktningssmåte kan bli for kortsiktig.

Til slutt i rapporten forsøker jeg å anvende et aktørnettverkperspektiv på utviklingen av datafaget for å antyde hvordan man kan fri seg fra de abstrakte nyttebetraktningene. Her er synet at nytte ikke er noe objektivt, men et begrep som deltakerne som utformer datafaget, kappes om å definere. Det gjelder å få gjennomslag for den definisjonen man selv synes best om. Fra akademisk hold vil man at nytte skal oppfattes som langsiktig, mens myndighetene gjerne framhever at nytte er kortsiktig. Det blir antydnet hvordan denne tautrekkingen former datafaget, og at den kanskje er den viktigste faktoren som både avgjør fagets innhold og dets plassering i den akademiske verden.

Solomon hevder at vitenskap er blitt en «vare» i løpet av det siste halve hundreåret, og at nytten av forskningen for samfunnet som helhet ble lagt sterkere vekt på mot slutten av 1960-årene enn ved utgangen av siste verdenskrig.

«Whatever the field of research and however distant from any short- or medium-term application (eg. high energy physics), it was axiomatic to advertise the multiple economic, social and political advantages that society would draw, as though by divine rights, from the increased investments.»²

¹ Se f.eks. Joseph Weizenbaum: *Computer Power and Human Reason*. Freeman, San Francisco, 1976.

² Jean-Jacques Solomon: «Science as a Commodity – Political Changes, Issues and Threats» i Michael Gibbons og Björn Wittrock: *Science as a Commodity. Threats to the Open Community of Scholars*. Longman, London, 1985, s. 80.

Det kan være grunn til å spørre om denne påstanden gjelder også for datafaget. Som nytt fagfelt etter den andre verdenskrigen skulle man tro at disiplinen var sterkt preget nettopp av det generelle nytteaspektet som Solomon hevder preger alle fagene i denne perioden.

Det er ikke bare den generelle tendensen mot å legge sterk vekt på nytteaspektet som kan ha spilt en rolle. For datafaget spesielt gjelder det at mange oppfatter det som en ekstremt nytteorientert disiplin. Det kommer av at man konstruerer informasjonssystemer for å oppnå gevinster i form av effektivitet, nøyaktighet, fullstendighet eller rutinisert likebehandling. Størst mulig nytte av informasjonssystemet målt med slike og lignende kriterier er ofte det man strever etter.³

Et tredje moment som trekker i retning av nyttetenking, kommer av at enkelte oppfatter datafaget utelukkende som støttefag for andre disipliner. Det kommer av at nytten ifølge mange ikke kan evalueres hovedsakelig etter datafaglige kriterier. De hevder at man ikke kan verdsette teknikk for teknikkens egen skyld, men for den nytten som teknologien har når den anvendes innenfor andre fagfelter.⁴

Ut fra slike nyttebetraktninger skulle man tro at behovet for evalueringskriterier preget teorier, metoder og teknikker for utvikling av informasjonssystemer. Det vil si at man i faglig sammenheng tar sikte på å utvikle og undervise teorier, metoder og teknikker som forventes å gi de beste resultatene i praktisk bruk.

I denne forbindelsen understreker jeg at begrepet nytte ikke er knyttet til diskusjonen som har forekommet bl.a. innenfor den filosofiske retningen som kalles utilitarismen. I mitt tilfelle er nytte bare knyttet til den økonomiske gevinsten og den økte effektiviteten som har sitt utspring i bedriftsøkonomisk og samfunnsøkonomisk tankegang i de siste tiårene. Ifølge denne tankegangen legitimerer all virksomhet, også vitenskap, sin eksistens ved å tilfredsstille økonomiske prinsipper til inntjening og innsparing.

³ Se f.eks. Arne Sølvberg et al.: «Skisse til et norsk program for forskning i informasjonsteknologi». Notat. Institutt for datateknikk og telematikk, Trondheim, 1985.

⁴ Se f.eks. Rolf Høyer: «Informasjonsvitenskap – et hjelpeseil på stormfylt hav.» *Forskningsnytt*, 24 (7), 1979.

2. FRAVÆRET AV EVALUERINGSKRITERIER

Innenfor datafaget er det lansert mange gode teorier om hvordan informasjonssystemer skal konstrueres, men det eksisterer få retningslinjer for å evaluere og avgjøre hva som er et godt informasjonssystem. På bakgrunn av de sterke forventningene om økonomisk nytte er det forbausende å slå fast at det nesten ikke finnes brukbare kriterier for å avgjøre hva som kjennetegner et godt informasjonssystem. I dag utvikles informasjonssystemer oftest uten at noen kan forutsi nytten av dem, f.eks. om de økonomiske fordelene ved de nye systemene oppveier utviklingskostnadene.

Det fins mange eksempler på dette. La meg her bare sitere ett eksempel, som stammer fra en rapport om EDB-bruken i Sandefjord kommune. Der var de endelige målene så lite målbare og oppfatningen av nytte så uklar at prosjektet ble redefinert litt etter hvert, tilsynelatende uten at noen var seg helt bevisst hva resultatet skulle bli til slutt:

«Når de tekniske mulighetene først forelå og var demonstrert, økte forventningene og kravene til de administrative stabsfunksjonene. Spesifikasjonsnivået i kommuneregnskap og eiendomsmatrikkel ble høynet, flere administrative rutiner kunne innbakes som trekkordninger i lønningssystemet. Lettere tilgjengelighet til rapportutskrifter o.l. førte til videre spredning, og dermed i sin tur til nye brukerkrav. På dette viset ble avdelingene pålagt utvidede arbeidsoppgaver, uten at dette ble formulert som nye oppgaver.»⁵

Slik brukte kommunen ressurser på en systemutvikling som ikke var planlagt, og som man ikke kunne forutse nytten av. Som så ofte i slike tilfeller sprengte prosjektet de økonomiske rammene som var satt. Kommunestyret var blitt forespeilt en betydelig økonomisk gevinst i tillegg til kvalitative forbedringer, men det endte med at administrasjonen la fram saken på nytt for kommunestyret, som aksepterte at den opprinnelige framdriftsplanen hvilte på urealistiske forutsetninger.⁶

I en av de mange lærebøkene i systemutvikling blir fraværet av retningslinjer for anslag av framtidig nytte begrunnet på denne måten:

«Det store problemet med å anslå forventede fremtidige gevinster ligger i å få fram anslag for styringsgevinster, altså gevinster som

⁵ Tron Espeli og Arne Maus: *EDB i Sandefjord kommune 1968-80*. Norsk regnesentral publikasjon nr. 701. Oslo, 1981, s. 33.

⁶ Espeli og Maus, s. 17

er et resultat av bedre beslutningsgrunnlag. I de aller fleste tilfellene er det her de virkelig store beløpene ligger.»⁷

Mer oppsiktsvekkende er det kanskje at man heller ikke har gode kriterier for å måle nytten etter at systemene er tatt i bruk. Derfor er det nesten aldri mulig å avgjøre hvor godt tilpasset formålet eller hvor lønnsomme informasjonssystemene er.

Eksemplene er mange også i dette tilfellet, men la meg igjen sitere fra den samme rapporten om EDB-bruken i Sandefjord kommune:

«Innføringen av Kommunedatas standardsystemer foregikk trinnvis (perioden 1968-72), og var heller ikke knyttet til noen stor éngangsinvestering fra kommunens side. Dette bidro nok til å dempe inntrykket av at man sto overfor beslutninger av stor økonomisk rekkevidde. Dessuten var omstillingene under oppbyggingen av den nye, felles kommuneadministrasjonen et argument for å innføre EDB akkurat da. Ved at disse to begivenhetene falt sammen i tid, høstet man en rasjonaliseringsgevinst i seg selv. Det er derfor knapt mulig å påvise det nøyaktige omfanget av produktivitetsforbedringer som direkte fulgte av EDB-innføringen i den første fasen.»⁸

I informasjonssystemenes livssyklus, slik den blir framstilt i store deler av systemutviklingslitteraturen, inngår faser som forandringsanalyse, analyse og utvikling av informasjonssystemet, realisering, implementering, forvaltning/drift og avvikling. Evaluering er ikke et så vesentlig punkt i denne sammenheng at det utgjør en egen fase. Selv innholdet i fasen avvikling, som uten tvil innebærer et sterkt innslag av evaluering, er lite utviklet ennå.

De mest brukte lærebøkene i systemutvikling sier lite om evaluering og evalueringskriterier. Derfor gir de heller ingen forklaringer på hvorfor EDB-systemer må avvikles eller hvordan man eventuelt kan forlenge bruksperioden for gamle EDB-systemer og på den måten minske investeringskostnadene.⁹

Vi kan fastslå at evalueringskriteriene mangler i stor utstrekning, men kanskje ikke først og fremst fordi det er umulig å utvikle slike. Årsaken ser heller ut til å være at få forsøk er gjort på å utvikle og ta i bruk evalueringskriterier. Det er typisk at nyere, videregående lærebøker i systemutvikling behandler evaluering først og fremst i forbindelse med de tekniske kravene som stilles til strukturen i de ulike modulene som et datamaskinprogram består av. Det er klart at det også er bruk for slike rent tekniske kriterier, men likevel er det nok så oppsiktsvekkende at det bare er de tekniske og ikke de sosiale og økonomiske kriteriene som systemutviklere får opplæring i å ta hensyn til.¹⁰

⁷ John Larsen: *Metoder og verktøy i systemarbeidet*. Systemarbeid. NKS-forlaget, Oslo, 1985, s. 199.

⁸ Espeli og Maus, s. 32.

⁹ Se f.eks. Erling S. Andersen: *Systemutvikling*. NKI-forlaget, Oslo, 1989.

¹⁰ Se f.eks. Lawrence Peters: *Advanced Structured Analysis and Design*. Prentice-Hall, Englewood Cliffs NJ, 1988.

De tradisjonelle og mest benyttede mål for kvaliteten i EDB-systemer går ut på å måle antall kodelinjer eller kostnad pr. feil under utviklingsarbeidet. Selv som mål på programmerernes effektivitet er dette noe tvilsomt, siden ikke alle løsninger lar seg programmere like lett. Dessuten tar målet ikke hensyn til om utviklingsverktøyet er konstruert for å generere mange kodelinjer (tredjegerasjonsspråk) eller få kodelinjer (fjerdegenerasjonsspråk).¹¹

Nyere mål som f.eks. justerte eller ujusterte «function points» forsøker å beskrive funksjonaliteten til EDB-systemet uavhengig av de metoder, verktøy og teknikker som blir brukt. Likevel tar «function points» i likhet med de fleste andre mål bare sikte på å måle den tekniske effektiviteten.¹²

Det kan nevnes at prosjektadministrative teknikker som inspeksjon og gjennomgang også spiller en viss rolle i undervisningen av systemutviklere. Men selv i denne sammenheng er det den tekniske kvaliteten av programmene som blir vurdert. Utover slike tekniske kriterier virker det som evalueringen av nytten fortsatt betyr lite for datafaget som disiplin.

En slik nedprioritering av mer generelle nyttebetraktninger vil for mange være et paradoks innenfor et fagfelt som er nytt, og som man skulle tro ville forsøke å stryke sin stilling både overfor andre fag og i publikums bevissthet ved å spille opp til forventningene om nytteeffekter som omgivelsene er opptatt av. Derimot har utviklingen innenfor faget gått i retning av å foreslå bedre teknisk effektivitet og bedre styring av EDB-prosjekter, bl.a. ved å bruke fjerdegenerasjonsspråk, prototyping, datamodellering og å legge større vekt på brukerkrav.¹³

Til tross for at teknisk effektivitet er målet, har de færreste forsøkt å vise hvor gode resultater selv slike nyere metoder og teknikker vil gi i praksis. Det gjelder til tross for at disse teknikkene ofte er mer eksakte og har et mer begrenset omfang enn de eldre metodene for systemutvikling. Derfor skulle en tro at de også ville være mer egnet for å forutsi de økonomiske og sosiale konsekvensene av EDB-systemene bedre enn de eldre metodene kan.¹⁴

Manglende evalueringskriterier hindrer ikke enkelte i å utnytte at utenforstående ikke er klar over denne svakheten ved datafaget. Figur 1 viser en annonse som spiller på forventningene hos intetanende personer om at det bør finnes sikre metoder for å måle effektivitet både under utvikling og under drift av datasystemer. Alle som kjenner faget godt, vet at garantien som loves i dette tilfellet, ikke holder. Det fins ingen sikre målemetoder som kan vise med sikkerhet at utviklingstid eller kostnader er 50 % lavere om man utfører arbeidet på den ene eller andre måten.

¹¹ James Martin: *Application Development Without Programmers*. Prentice-Hall, Englewood Cliffs NJ, 1982.

¹² A.J. Albrecht: *Measuring Application Development Productivity*. IBM Share Guide, 1979.

¹³ Se f.eks. James Martin: *Rapid Application Development*. Macmillan, London, 1991; Tom Gilb: *Principles of Software Engineering Management*. Addison-Wesley, Wokingham, 1988.

¹⁴ Se f.eks. Charles R. Necco et.al.: «Systems Analysis and Design: Current Practices», *MIS Quarterly*, 11 (4), 1987, s. 461-476.

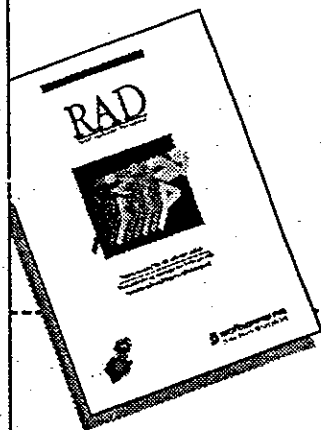
50% kortere utviklingstid på dine edb-prosjekter og 50% lavere kostnader - garantert!

La oss løse den neste utviklings-oppgaven din med Rapid Application Development metoden (RAD). Hvis vi ikke gjør jobben på under halvparten av den tiden du selv vil bruke med dine eksisterende verktøy, koster det deg ingenting!

RAD-metoden er bare en av utallige fordeler du oppnår gjennom et samarbeide med Software AG. Nettoresultatet er større produktivitet, og sterkt reduserte kostnader.

Send eller fax kupongen i dag, og få RAD-brosjyren som forteller deg om både fordelene og garantien.

Du kan også ringe direkte til Knut Bjerknes på tlf.: 22 83 73 40. Det er mange penger å spare!



SOFTWARE AG
SOLUTIONS WORLDWIDE

Software AG er et av verdens største software-selskaper. Vi er markedsledende på 4GL, CASE og database-teknologi. Du finner oss i mer enn 60 forskjellige land.

☐ **JA TAKK**

Send meg en RAD-brosjyre!

Firma: _____

Tittel: _____

Navn: _____

Adresse: _____

Postnr.: _____ Poststed: _____

Send eller fax kupongen til: **Fax: 22 83 73 44**
Software AG, Norge A/S, Munkedamsvn. 53 B, 0250 Oslo

Figur 1. Annonse i Dagens Næringsliv 4.3.1993.

3. BETINGELSENE FOR DET NYE FAGET

Hvorfor har ikke nytteaspektet slått sterkere igjennom innenfor datafaget? Er nyttebegrepet ikke viktig for de akademiske miljøene som forsøker å etablere datafaget som et nytt fagfelt? Eller er fagidentitet forbundet med at man beviser sin unytte i den forstand at man først og fremst utvikler teorier, metoder og teknikker på rent vitenskapelige premisser uten å ta nytteverdien i betraktning? For å besvare dette spørsmålet skal vi først se på forholdene som lå til grunn for etableringen av datafaget.

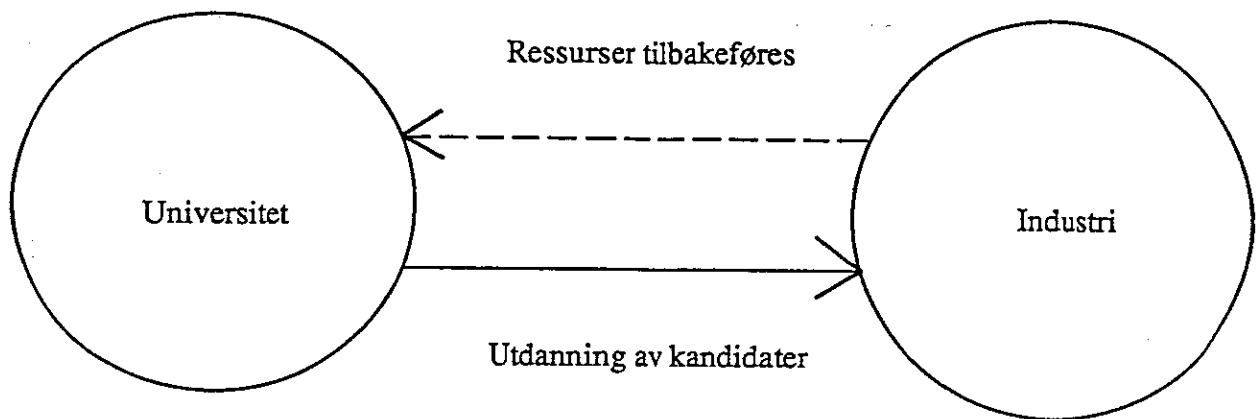
Stankiewicz tar opp de ulike gradene av kontakt som de akademiske miljøene kan ha med samfunnet rundt. Spesielt er han opptatt av forholdet til industrien. Han skisserer bl.a. en tradisjonell modell og en entreprenørmodell for dette forholdet. Den tradisjonelle modellen går ut på at universitetene er leverandører av kandidater til industrien, og at ressurser fra industrien blir tilført de akademiske miljøene direkte i varierende grad. Denne tradisjonelle modellen er skissert i figur 2.

Enkelte ganger satser industrien på bredere støtte til universitetsmiljøene, spesielt når det gjelder større teknologiske utviklingsprosjekter. Da etableres det i følge Stankiewicz en entreprenørmodell som er skissert i figur 3. Resultatet av støtten blir ofte at forskerne bryter ut av universitetsmiljøet etter en tid for å markedsføre de nye tekniske løsningene gjennom egne firmaer. Disse firmaene er ofte lite levedyktige. De kommer lett i økonomiske vanskeligheter og blir kjøpt opp av storindustrien. Siden grunnlaget ikke er til sted lenger for å fortsette de opprinnelige prosjektene ved universitetene, begrenser industrien nå sin støtte. Dessuten begynner industrien selv etter hvert å ivareta mye av den forskningen som universitetene stod for opprinnelig. Det fører til at entreprenørmodellen igjen blir avløst av den tradisjonelle modellen for kontakt mellom universitet og industri.¹⁵

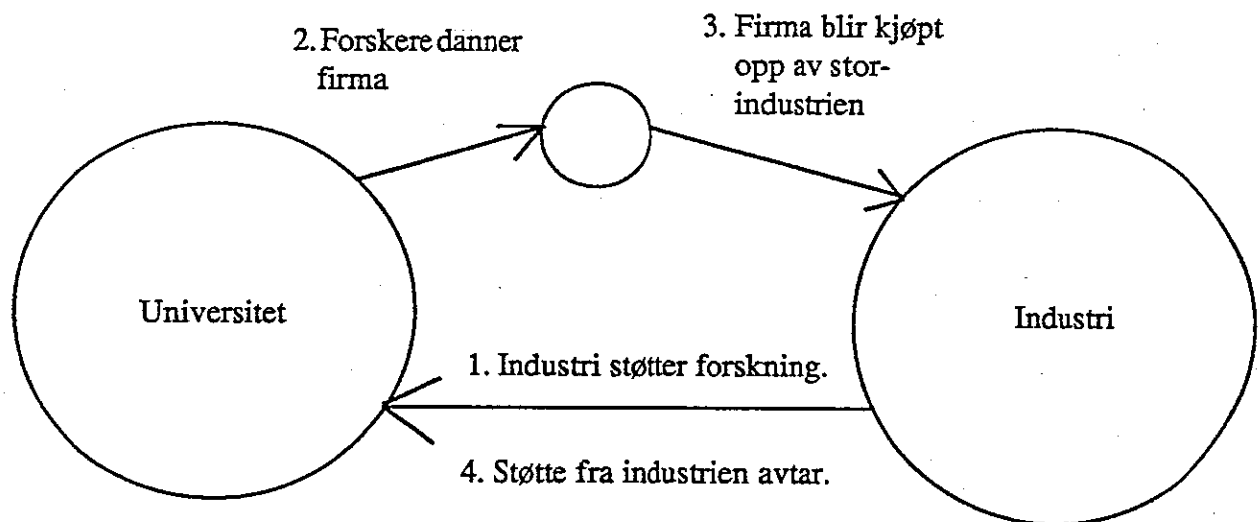
For å ta datamaskinen som eksempel var Eckerts og Mauchlys utvikling av ENIAC ved University of Pennsylvania i 1943-45 et typisk eksempel på entreprenørmodellen. De dannet et eget firma for å markedsføre maskinen, som i sin ferdige versjon ble kalt UNIVAC. Etter en del økonomiske problemer ble firmaet kjøpt opp av Remington Rand i 1951.¹⁶

¹⁵ Rikard Stankiewicz: *Academics and Entrepreneurs. Developing University-Industry Relations*. Frances Pinter, London, 1986.

¹⁶ Nancy Stern: *From ENIAC to UNIVAC: An Appraisal of the Eckert-Mauchly Computers*. Digital Press, Bedford MA, 1981.



Figur 2. Tradisjonell modell for kontakt mellom forskning og industri.



Figur 3. Entreprenørmodellen for kontakt mellom industri og forskning.

Det fins også andre eksempler på entreprenørkontakter mellom forskningsmiljøer ved de store amerikanske universitetene og det som etter hvert skulle bli de store datamaskinprodusentene.¹⁷

Entreprenørmodellen fungerer når forskningsprosjektene har et klart og veldefinert nytteaspekt. Det var tilfellet da de første kommersielle datamaskinene ble utviklet. Etter at datamaskinprodusentene overtok det vesentligste av forskningen selv, ble den tradisjonelle universitets-industri-modellen gjenopprettet på begynnelsen av 1950-tallet.

Til tross for at den moderne elektroniske forskningen kan spores tilbake til årene før siste verdenskrig, er datafaget forholdsvis nytt. Man regner med at det varte helt til begynnelsen på 1960-tallet før datafaget oppstod som egen disiplin.¹⁸

Det var behov knyttet til bruk av den nye generasjonen datamaskiner og ikke til utviklingen av elektronikken som lå til grunn for etableringen av datafaget. Litt av den konkrete bakgrunnen for at datafaget oppstod som disiplin, finner vi i behovet for å løse beregningsproblemer, reguleringsoppgaver og massedataproblemer.¹⁹

Etter at entreprenørmodellen ble avvirket, forsvant tydeligvis også en stor del av nyttebetraktningene som ellers kunne ha utgjort en del av grunnlaget for det gryende datafaget. I stedet ser det ut til at vekten hovedsakelig ble lagt på å kunne tilfredsstille behovet for et høyere utdanningstilbud og å produsere kvalifisert personale i tilstrekkelig stort antall.²⁰

Dessuten sprang det nye datafaget som regel ut fra andre deler av universitetsmiljøet enn der de første datamaskinene ble utviklet. Vekten på beregningsoppgaver førte som regel til at det ble matematikerne og ikke fysikerne som ledet an i etableringen av den nye disiplinen. Peters og Etzkowitz har kartlagt at størstedelen av den opprinnelige fagstaben ved datafaglige institutter på amerikanske universiteter hadde matematikkbakgrunn (38 %). Det andre sterkt representerte området var elektronikkfag (21 %). Bare en

¹⁷ Se f.eks. N. Metropolis et.al. (red.): *A History of Computing in the Twentieth Century*. Academic Press, New York, 1980; G.W.A. Dummer: *Electronic Inventions and Discoveries*. 2. utg. Pergamon, Oxford, 1978.

¹⁸ Lois S. Peters og Henry Etzkowitz: «The Institutionalization of Academic Computer Science.» Rensselaer Polytechnic Institute. Innlegg på 4S/EASST Conference, Amsterdam, 16-19. november 1988.

¹⁹ Håkon With Andersen: «Et kort riss av informasjonsteknologiens utvikling» i Per Morten Schiefloe og Knut H. Sørensen (red.): *Revolusjonen som forsvant? EDB, informasjonsteknologi og samfunn*. Universitetsforlaget, Oslo, 1986.

²⁰ Tove Håpnæs: *På sporet av en profesjon? Om framveksten av høyere EDB-utdanning i Norge*. STS rapport nr. 10. Senter for teknologi og samfunn, Trondheim, 1989; Ellen Ofstad: *Systemutviklingens tornefulle vei 1958-1963*. Moral-prosjektets skriftserie nr. 16. Institutt for informasjonsvitenskap, Bergen, 1985.

mindre del hadde annen ingeniørutdanning eller utdanning i økonomisk-administrative fag.²¹

En annen amerikansk forsker sier at den sammensatte bakgrunnen hos ansatte ved de datafaglige instituttene gjør at datafaget står i en særstilling sammenlignet med utviklingen av andre disipliner:

«It is unique in that it evolved from researchers from diverse backgrounds instead of emerging from an existing discipline.»²²

Med matematikken kom grunnleggende metoder og teknikker inn i datafaget, først og fremst fra de delene av matematikken som kalles kybernetikk og numerisk analyse. Datafaget ble i stor grad algoritmebasert i stedet for elektronikkbasert. Samtidig innebar det en dreining fra problemene knyttet til maskinvare og over mot utforming av programvare. Programmeringsteknikken ble det første store arbeidsområdet for det nye faget og preget størsteparten av forskningen og undervisningen både i USA og resten av verden gjennom 1960- og 1970-årene.

Det ble raskt klart at skriving av programmer måtte suppleres med teknikker og metoder som gjorde at programmene kunne knyttes sammen i større systemer. Man begynte å snakke om konstruksjon og bygging av informasjonssystemer analogt med tegning og bygging av hus eller tekniske konstruksjoner. Analogien var så åpenbar at man begynte å hente metoder og teknikker fra ingeniørfagene for å ta dem i bruk i det fagfeltet som ble kalt «systems analysis and design» eller systemutvikling.²³

Utover i 1970- og 1980-årene ble man mer bevisst de samfunnsmessige konsekvensene som bruken av datateknikken hadde. Personvern, arbeidsløshet, arbeidsmiljø og brukerkrav er sentrale stikkord i den forbindelse. En rekke land innførte personregisterlover, og i vår del av verden fikk arbeidsmiljølov, hovedavtaler i arbeidslivet og oppnevning av datatillitsvalgte en sterk innvirkning på bruken av den nye teknologien. Det førte til at samfunnsinformatikk ble etablert som eget fagfelt innenfor datafaget. Dette fagfeltet tok opp teorier fra samfunnsvitenskap, spesielt fra organisasjonsfag.²⁴

Er datafaget fullt etablert som disiplin nå i første del av 1990-årene eller vil det også i framtida bestå av de komponentene jeg har skissert i figur 4? Det jeg synes å merke, er en sakte, men sikker tendens bort fra de fagfeltene som opprinnelig utgjorde hoveddelen av datafaget. Som sagt var elektronikk-komponenten svak alt på 1960-tallet. Matematikk-komponenten ser ut til å ha

²¹ Peters og Etzkowitz, s. 5.

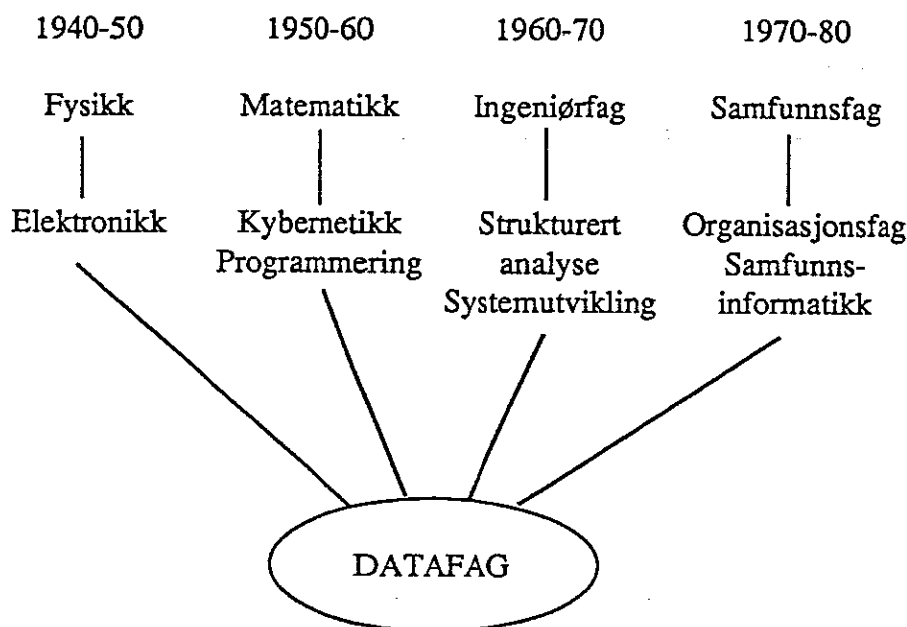
²² John E. Hopcroft: «Computer Science: The Emergence of a Discipline.» *Communications of the ACM*, 30 (3), 1987, s. 202.

²³ Se f.eks. Börje Langefors: *Theoretical Analysis of Information Systems*. 2. utg. Studentlitteratur, Lund, 1967.

²⁴ Se f.eks. Per Morten Schiefloe og Knut H. Sørensen (red.): *Revolusjonen som forsvant? EDB, informasjonsteknologi og samfunn*. Universitetsforlaget, Oslo, 1986; Eivind Jacobsen et.al.: *Samfunnsinformatikk. Om samspillet mellom EDB og samfunn*. NKS-forlaget, Oslo, 1986.

blitt svekket i løpet av 1980-tallet, mens samfunnsinformatikken kan få sterkere betydning utover mot år 2000. Denne utviklingen ser ut til å skje så raskt at datafaget fortsatt virker som en lite etablert disiplin.

Bevegelsen i retning av samfunnsinformatikk er ikke like synlig overalt. Peters og Etzkowitz har funnet at datafaget ved amerikanske universiteter enten har en tendens til å bli plassert sammen med de tekniske fagene eller til å bli et frittstående fag. Denne konklusjonen bekrefter hvor vanskelig det fortsatt er å trekke sikre konklusjoner om datafagets status i relasjon til andre fag og hvordan det vil utvikle seg i tida framover. Vi bør også ta i betraktning at Peters og Etzkowitz bare har undersøkt datafagets organisatoriske plassering ved universitetene og ikke fagets innhold.²⁵



Figur 4. Datafagets opprinnelse.

²⁵ Peters og Etzkowitz, s. 12-13.

4. DEN LANGSIKTIGE UTVIKLINGEN AV FAGET

Som man kan forstå av det som er sagt hittil, er det spørsmål om ikke datafagets utvikling har gått for raskt eller rettere sagt er kommet for kort til at man har rukket så svært langt når det gjelder å etablere et allment akseptert grunnlag av teorier, metoder og teknikker. Man kan heller ikke si at det er utviklet en allment akseptert fagterminologi ennå. De mange betegnelse som disiplinen har – «computer science», informatikk, «informatics», informasjonsteknologi, datateknikk, EDB-studium osv. – tyder på det. Man kan også finne en rekke definisjoner av datafaget. Definisjonene kjennetegnes ved at de er nokså uklare. Ofte er de så generelle at de også passer som beskrivelse av andre fag.²⁶

Erfaringen fra andre disipliner tilsier at utviklingen av et akademisk fag kan ta lang tid. Det kan nevnes at verdens første psykologiske laboratorium ble etablert i Leipzig i 1879, og at Psykologisk institutt ved Universitetet i Oslo ble etablert i 1909. I denne pionertiden gjaldt det å få trukket opp grensene mellom vitenskapelighet og amatørvirksomhet, gi faget et vitenskapelig innhold, utvikle begrepsapparatet, trekke opp grensene mot andre fag og få etablert yrkesroller.²⁷

Skard mener at norsk psykologi ble «moden» først mot slutten av 1930-tallet, altså knappe 30 år etter at faget ble etablert ved Universitetet i Oslo. Mange vil nok tvile på at denne påstanden er riktig, bl.a. fordi profesjonsstriden med legene først ble løst da psykologtittelen ble lovfestet. Det skjedde så sent som i 1973, mer enn 60 år etter at faget ble etablert i vårt akademiske miljø.

Grunnen til at fagutviklingen går sent, ikke bare i psykologi, men i akademiske fag generelt, beror på at kunnskapsutviklingen skjer gjennom en meningsbrytning. Latour framstiller det som en konfliktmodell i det som kalles den konstruktivistiske tilnærmingen til vitenskapsstudier. Han legger vekt på å vise hvordan vitenskapelig litteratur kan analyseres som retorikk for å vise hvilke retoriske grep forskerne bruker for å virke overbevisende.²⁸

Akademisk meningsutveksling tar tid, og kunnskapsutviklingen skjer heller ikke alltid kontinuerlig, men i rykk og napp. Det er motetendenser også i vitenskapen, og enkelte problemer kan bli liggende uten å få større

²⁶ Peter J. Denning et.al.: «Computing as a Discipline.» *Communications of the ACM*, 32 (1), 1989, s. 11-12.

²⁷ Åse Gruda Skard: «Psykologien i Norge», *Psykologisk tidsskrift*, 43 (5,6 og 7), 1959.

²⁸ Bruno Latour: *Science in Action*. Open University Press, Milton Keynes, 1987.

oppmerksomhet i lengre perioder av gangen. I datafaget ser vi dette tydeligst på måten som diskusjonen har skiftet mellom emner som strukturerte metoder, fjerdegenerasjonsspråk, kunnskapsbaserte systemer og åpne systemer, for bare å nevne noen stikkord fra de siste årenes debatt. Med en slik skiftende konsentrasjon om ulike emner er det klart at det hittil har rukket å bli liten enighet om teorier, metoder og teknikker på mange felter.

Det er grunn til å tro at grunnleggende vitenskapelige trekk ved datafaget fortsatt er for lite utviklet. Det vitenskapelige innholdet i faget er ofte for dårlig definert, begrepsutviklingen er mangelfull, grensene mot andre fag endrer seg fortsatt raskt, og yrkesrollene er slett ikke avklart.

Dette siste punktet kan fortjene en ekstra kommentar, siden mange vil tro at yrkesroller som programmerer og systemutvikler er de egentlige datafag-yrkene. Til det er å si at de nåværende yrkene ofte er for begrensede til å representere hele disiplinen. Det som mangler, er et generelt informatikeryrke på samme nivå som yrkene psykolog, jurist, sosialøkonom osv.

5. NYTTE PÅ KORT OG LANG SIKT

Spørsmålet som ble stilt innledningsvis, var om datafaget er nyttig eller bevisst unyttig. Jeg har argumentert med at nyttebetraktningene i stor utstrekning er fraværende fordi evalueringskriteriene mangler. Etter å ha sett på betingelsene for det nye faget og den langsiktige fagutviklingen som kjennetegner akademiske disipliner, vil jeg påstå at evalueringskriteriene mangler fordi datafaget ennå ikke har nådd langt nok i sin utvikling. Først må de mer grunnleggende prinsippene for fagets akademiske eksistens avklares. Her tenker jeg på slike ting som begrepsutviklingen og grensene mot andre fag. Som sagt er det ennå ikke enighet om datafaget skal vurderes ut fra egne premisser eller ut fra premissene som legges til grunn i andre fag.

Slike grunner gir forklaringen på at datafaget ennå ikke er så nytteorientert som mange forventer. Men er faget *bevisst* unyttig? Det riktige vil nok være å si at alle som jobber med faget, er bevisst hvor kort fagutviklingen har kommet og i hvor liten utstrekning det er utviklet evalueringskriterier ennå.

Dette er forhold som vi ikke kan gjøre noe med på kort sikt, for all erfaring viser at akademiske disipliner bare kan utvikles gjennom en langsiktig, internasjonal meningsutveksling mellom forskere. Vi vet at det fører til stadige legitimeringsproblemer overfor dem som først og fremst ser etter nytteverdien i faget. Likevel er det grunn til å tro at myndighetene innser at fagutviklingen må gå sin gang, og at kortsiktige nyttebetraktninger ikke må tillegges så stor vekt at tilgangen på offentlige midler til forskning og utdanning begrenses. Grunnen er at begrenset ressurstilgang vil sette ned utviklingstakten, slik at det

tar enda lengre tid før det utvikles evalueringskriterier som nyttebetraktningene er helt avhengige av.

Jeg er ikke fornøyd med å fastslå at fagutviklingen er kommet kort, men vil også se nærmere på hvorfor den er kommet så kort. Utviklingen av maskinvaren har en over 50-årig historie, mens datafaget som disiplin er av langt nyere dato. Hva er årsakene til denne forskyvingen i tid mellom de første praktiske forsøkene og utviklingen av det teoretiske grunnlaget som faget har eller er i ferd med å få? Kan noe av årsaken være konstellasjonen av interesser og grupperinger og den tautrekkingen disse har vært med på i det siste halve århundret? Hva kan vi dokumentere av interessekonflikt, og hvem har vært de dominerende partene i konflikten?

6. DATAFAGET – «A THICK DESCRIPTION»

Man bør ikke feste seg for sterkt ved bitfiklerkulturen når utviklingen av datafaget skal beskrives. Den kan betraktes som et marginalfenomen og ikke som et av særtrekkene ved datafaget. Det har blitt hevdet at de fleste tekniske og vitenskapelige miljøer kan oppvise personer med en livsstil som har mye til felles med bitfiklerne.²⁹

Vi bør heller legge vekt på å følge de viktigste aktørene, som representeres ved de tunge teknologimiljøene, hovedsakelig i USA. I utgangspunktet har vi tre store aktører: universitetene som de opprinnelige utviklingsmiljøene, datamaskinindustrien, som overtok etter hvert, og forsvaret, som var en viktig finansieringskilde. Vi skal også se at forholdet mellom de enkelte forskerne og universitetene spiller en rolle i viktige sammenhenger.

Universitetene var tidlig i bildet som utviklingsmiljøer, i hvert fall i USA, hvor det var nær kontakt mellom universitetene, industrien og forsvaret. Et av de mest kjente prosjektene før andre verdenskrig ble skapt av alliansen mellom Harvard-universitetet, IBM og U.S. Navy. Det var Howard H. Aikens Mark I-prosjekt – den første programstyrte kalkulatoren. Bortsett fra at den kunne beregne matematiske tabeller, er Mark I blitt betegnet som en «blindgate» uten større betydning for utviklingen av datamaskinene. Forsvaret dekket 1/3 av utviklingskostnadene på \$ 500 000 og IBM resten, men ingen av dem fattet interesse for det elektromekaniske produktet med hele 750 000 deler som vi kjenner som Mark I.³⁰

²⁹ Knut H. Sørensen: *Sentrale datafag og sære datamiljøer?* STS arbeidsnotat 2/87, Trondheim, 1987.

³⁰ Stan Augarten: *Bit by Bit. An Illustrated History of Computers.* Allen & Unwin, London, 1985, s. 106-107.

ENIAC-prosjektet ved University of Pennsylvania ble det viktigste prosjektet under den andre verdenskrig. Forsvaret hadde tidligere samarbeidet med det samme universitetet om å få utviklet en type kalkulatorer som ble kalt «differential analyzer». Det var begrunnelsen for at forsvaret også støttet ENIAC-prosjektet. MIT var et av de få andre universitetene som hadde utviklet en tilsvarende maskin, men med private midler. Derfor ble MIT ikke en aktuell alliansepartner for forsvaret. Maskinen skulle beregne alle de ballistiske tabellene som man ikke maktet ved hjelp av «differential analyzer» eller for hånd da våpenproduksjonen pågikk for fullt under andre verdenskrig:

«ENIAC was not the product of a scientist or engineer's frustration with solving equations with mechanical calculators, but of a military imperative during a national crisis.»³¹

Forsvarets valg av samarbeidspartner ble ikke tatt nådig opp av de forsmådde universitetene. Derfor var University of Pennsylvania glad over å kunne invitere den kjente matematikeren John von Neumann til et opphold hos seg:

«As a highly respected scientist, von Neumann lent a badly needed air of legitimacy [sic] to the school's pioneering endeavors. Many scientists – in particular, Bush at MIT, Aiken at Harvard, and Stibitz at Bell Labs – regarded ENIAC as a foolish endeavor, bound for failure, and a waste of government funds that would have been better spent on the proven technologies of relay calculators and differential analyzers. There was a lot of bad blood between the University of Pennsylvania and its detractors.»³²

Skeptikerne ved de andre universitetene kunne ikke hindre at prosjektet fortsatte. Det som til slutt fikk forsvaret til å trekke tilbake sin støtte, var striden om patentrettighetene. Forsvaret ville at akademikerne ved University of Pennsylvania skulle gi fra seg disse rettighetene. Det samme ville også universitetets ledelse. Til slutt frafalt forsvaret sine krav. Det gjorde også universitetet etter en juridisk kranjel som viste at ansettelseskontraktene ikke gav universitetet rett til patentene. Men nå var det de øvrige universitetskollegene som grep inn. De kunne ikke gå med på at den enkelte forsker og ikke institusjonen fikk beholde rettighetene til oppfinnelser som ble gjort i prosjektene. Striden endte med at de to rettighetshaverne, Eckert og Mauchly, sa opp stillingene sine. Det førte til at University of Pennsylvania mistet sin tetposisjon i utviklingen av datamaskiner.

Striden gjaldt rettighetene til ENIAC, som ikke kunne lagre programmer. Oppfølgeren, EDVAC, var konstruert som den første datamaskinen med et lagret program. Her hadde von Neumann interesser, siden han utviklet den praktiske løsningen for hvordan lagringen skulle skje. I dette tilfellet utviklet det seg til en kamp om rettighetene mellom de tre partene Eckert og Mauchly, universitetet og von Neumann. Det endte med at ingen av dem vant, siden

³¹ Augarten, s. 110.

³² Augarten, s. 137.

fristen for å få patentert prinsippet om programlagring, var gått ut. Prinsippet om programlagring – og dermed rettighetene til prinsippet bak den moderne datamaskinen – var dermed frigitt.³³

Størsteparten av institusjonene som rett etter krigen sendte sine folk til datamaskinkursene ved University of Pennsylvania, var bedrifter, universiteter og offentlige organer som disponerte over de tekniske og økonomiske ressursene til å bygge datamaskiner – bl.a. IBM, General Electric, Bell Labs, MIT, Harvard og National Bureau of Standards:

The majority of the organizations eventually constructed their own machines, and some of them, such as IBM and General Electric, entered the computer business. By the end of 1947, six computers were under construction in America, three in academia and three in industry.»³⁴

Dermed var alliansen både mellom universitetene og industrien og mellom universitetene og forsvaret brutt. Fra nå av foregikk utviklingen av datamaskinene hovedsakelig i industrien.

Suksessen til den første kommersielle datamaskinen, UNIVAC, kom av at teknologien ikke lenger var konsentrert om én maskin, men ble utvidet til et system eller en kombinasjon av ulike tekniske komponenter: skrivere, magnetiske lager osv., brukt i en rekke sammenhenger – en veritabel kombinasjon av teknikk og samfunnsforhold:

«UNIVAC included everything you needed to haul your paper-laden accounting department, factory, government agency, or university into the twentieth century. And it was a truly general-purpose computer, able to process both numeric and alphabetic data. It could print bills, sort accounts, and predict elections as well as solve the kind of scientific problems that led to the invention of computers in the first place.»³⁵

Mens datamaskinindustrien klarte å bli ledende i forsøkene på å lansere denne kombinasjonen av teknikk og samfunn i mange tiår framover, klarte ikke universitetene å tilpasse seg den nye virkeligheten og beholde posisjonen som «nyttige» i datamaskinsammenheng. Mye skyldtes at man lenge ikke hadde noe bestemt syn på organiseringen av forskning og undervisning i datafag etter at maskinvareutviklingen ble tatt hånd om av industrien. Man lot industrien stå for de fleste typer forskning og undervisning knyttet til datamaskiner. Forbindelsen mellom programvare, maskinvare og arbeidsmiljø var sterk, og industrien utnyttet den sterke forbindelsen for å knytte kundene fastere til seg:

«The formal training of computer staff was mostly in the hands of the computer manufacturers during the 1950s. The same business forces that induced them to provide applications programs with their machines also induced them to provide training for user

³³ Augarten, s. 151-153.

³⁴ Augarten, s. 153-154.

³⁵ Augarten, s. 156.

organizations. They would not sell computers without providing training. The training provided by IBM in particular was renowned for its high quality. Organizations trying to recruit would use the promise of IBM training as a lure for potential employees. On the other hand, computer hardware manufacturers, and especially IBM, provided staff as well as hardware and software to customers as part of their marketing strategy. Sometimes these people remained formally employed by IBM, but worked on-site at key customers for years. Many eventually joined user organisations formally, often taking key positions. With this strategy hardware manufacturers encouraged customer loyalty.»³⁶

Leverandørene beholdt sitt grep også over utdanningen av brukerne så lenge kunnskapen hovedsakelig var maskinavhengig og baserte seg på den individuelle brukerens oppfinnsomhet og innsikt. «On-site training», dvs. opplæring på arbeidsplassen, var den aksepterte løsningen på utdanningsproblemet langt inn i 1960-årene. Man regnet den gang med at perioden med forholdsvis uproduktiv læring for en nyansatt var rundt 18 måneder, og at det ikke eksisterte gode alternativer til denne formen for opplæring.³⁷

Situasjonen endret seg etter hvert som de såkalt strukturerte teknikkene ble utviklet:

«The impact of structured programming on the programming labour process lay in its ability to reduce the dependence on any individual programmer. Standardized working methods, as opposed to individual 'clever tricks', and increased clarity and readability of programs meant that programming work could be more easily transferred between one programmer and another. Similarly, top-down design and other procedures for program modularization also increased the scope for the division of labour in programming projects, although they do not necessarily specify the division of labour.»³⁸

Innføringen av strukturerte metoder innebar at den individuelle opplæringen av den enkelte programmerer, systemutvikler og driftsansvarlige fikk mindre betydning. Utviklingen av strukturerte metoder økte ledelsens kontroll over arbeidsprosessen, men samtidig ble leverandørenes innflytelse over brukerne svekket. Metodeutviklingen førte også til at stillingen til de uavhengige formidlerne av kunnskap, spesielt universitetene, ble styrket.

Framveksten av datafaget ble en følge av standardisering på mange plan. For det første innebar strukturerte metoder en standardisering av arbeidsprosessene som det ble undervist i, ikke minst innenfor programmering og systemutvikling. For det andre tok anbefalte studieopplegg sikte på å gi

³⁶ Andrew L. Friedman og Dominic S. Cornford: *Computer Systems Development: History, Organization and Implementation*. Wiley, Chichester, 1989, s. 92-93.

³⁷ Friedman og Cornford, s. 113.

³⁸ Friedman og Cornford, s. 135.

datafaget et klart definert innhold som skilte det fra andre fag. Samtidig var det tanken at anbefalingene skulle gjøre datafaget, eller i hvert fall de enkelte kursene som datafaget ble sammensatt av, mest mulig standardisert fra universitet til universitet.

Standardiseringen av arbeidsprosessene var nødvendig av flere grunner. Raskt økende kompleksitet i systemløsningene kombinert med lavt utdanningsnivå hos personalet skapte ofte forsinkelser, økte kostnader og EDB-prosjekter som ble en fiasko. Det skapte et behov hos ledelsen i mange virksomheter for økt kontroll med systemutviklingsprosjektene:

«This came to be known as the software crisis, or the software bottleneck, or the problem of lagging software productivity (and especially of unsatisfactory programmer productivity).»³⁹

Standardiseringen skjedde ofte ved at det amerikanske forsvaret stilte krav til produsentene. Slik oppstod det bl.a. standarder for programmeringsspråk, databaser og datakommunikasjon. Standardene førte med seg at en god del «on-site» opplæring i individuelle ferdigheter ble erstattet med grunnutdanning med vekt på bruk av standarder og maskinuavhengige løsninger.

Problemene med å få utviklet god programvare førte også til økt etterspørsel etter velutdannet arbeidskraft. Også i dette tilfellet kunne universitetene bidra til å dekke en stigende etterspørsel som leverandørene ikke var innstilt på å etterkomme.

På denne måten kan en si at det igjen var forsvaret og universitetene som ble viktige aktører, mens leverandørenes rolle avtok. Det styrket universitetenes stilling, først og fremst i undervisningssammenheng fra 1960-årene av. Det var en helt ny rolle sammenlignet med rollen som universitetene spilte da de første datamaskinene ble utviklet.

Den mest kjente standardiseringen av datafaget ble utarbeidet gjennom kursbeskrivelsene av datafaget som universitetsfag utgitt av Association for Computing Machinery (ACM), som er en ikke-kommersiell organisasjon hvor akademikere har hatt og fortsatt har stor innflytelse. Det er kommet flere versjoner av disse kursbeskrivelsene fra slutten av 1960-tallet og til i dag. De har betydd en god del for etableringen av datafaget som akademisk disiplin både i USA og i andre land, bl.a. i Norge.⁴⁰

Som sagt tidligere var det først på 1960-tallet at datafaget dukket opp som selvstendig akademisk disiplin. Peters og Etzkovitz har funnet at det første datafaginstituttet i USA ble organisert ved Perdue University i 1962, og at størsteparten av datafaginstituttene ble opprettet mellom 1962 og 1972.

³⁹ Friedman og Cornford, s. 90.

⁴⁰ «Curriculum 68: Recommendations for the Undergraduate Program in Computer Science.» *Communications of the ACM*, 11 (3), 1968; «Curriculum 78: Recommendations for the Undergraduate Program in Computer Science.» *Communications of the ACM*, 22 (3), 1979; «A Summary of the ACM/IEEE-CS Joint Curriculum Task Force Report: Computing Curricula 1991.» *Communications of the ACM*, 34 (6), 1991. Et eksempel på betydningen i Norge er nevnt i Tove Håpnes: *På sporet av en profesjon? Om framveksten av høyere EDB-utdanning i Norge*. STS-rapport nr. 10, Trondheim, 1989, s. 82.

Likevel ble rundt 30 % av instituttene organisert så sent som mellom 1973 og 1985. Enda i 1988 hadde enkelte viktige universiteter som Harvard og MIT ikke egne datafaginstitutter.⁴¹

I Norge kom man i gang relativt sent. Så sent som i slutten av 1960-årene fantes det overhodet ingen stillinger innenfor området ved NTH. Først i 1971 vedtok professorrådet ved NTH å opprette en egen studieretning i databehandling som spesialiseringsretning i de to siste årene av sivilingeniørutdanningen. Studieretningen ble lagt til linjen for teknisk fysikk ved allmenavdelingen ved NTH. Samtidig ble det opprettet tre universitetslektorer.⁴²

I 1972 ble informasjonsvitenskap opprettet som et grunn- og mellomfagstilbud ved Universitetet i Bergen. Samtidig ble tre stillinger, hvorav ett professorat, opprettet og lagt til både Det historisk-filosofiske og Det samfunnsvitenskapelige fakultet. Et informatikkstudium under Matematisk institutt ved Universitetet i Bergen ble opprettet først i 1979. Et kurs i numerisk analyse og EDB var blitt holdt første gang i 1963.⁴³

Tidspunktet for etableringen av datafaget ved de fleste universitetene bekrefter det vi tidligere har pekt på, nemlig at datafaget fortsatt er for lite utviklet. Enten datafaget hadde sitt utspring i matematikk, fysikk eller samfunnsfag, endrer grensene mot disse fagene seg fortsatt raskt. En viss standardisering av studietilbudet har skjedd, noe som ikke minst skyldes forslagene fra ACM. Likevel gjenstår mye arbeid med fagets innhold, metoder og teorier før vi kan si at datafaget har oppnådd en entydig identitet.

Det vil fortsatt pågå en tautrekking mellom ulike faggrupper, og det er ikke gitt hvor mye datafaget vil bli preget av f.eks. matematikk i framtida. Fortsatt hevder enkelte at løsningen på datafagets identitet ligger i å la det preges av matematiske metoder, som har et sterkt teorigrunnlag, eventuelt la det preges av naturvitenskapene generelt, fordi man der har et høyt abstraksjonsnivå med sterk vekt på modellbygging, eller av ingeniørfagene, som har sin styrke i systematisk analyse av problemene («design»). Men det er også hevdet at datafaget stiller så store krav at denne kategoriseringen basert på eksisterende faggrenser ikke er egnet:

«Many debates about the relative merits of mathematics, science, and engineering are implicitly based on an assumption that one of the three processes (theory, abstraction, or design) is the most fundamental. Closer examination, however, reveals that in

⁴¹ Lois S. Peters og Henry Etzkowitz: «The Institutionalization of Academic Computer Science.» Rensselaer Polytechnic Institute. Innlegg på 4S/EASST Conference, Amsterdam, 16-19. november 1988, s. 8-9.

⁴² Tove Håpnes: *På sporet av en profesjon? Om framveksten av høyere EDB-utdanning i Norge.* STS-rapport nr. 10, Trondheim, 1989, s. 84-85.

⁴³ Håpnes, s. 90 og 78.

computing the three processes are so intricately intertwined that it is irrational to say that any one is fundamental.»⁴⁴

Situasjonen blir ikke klarere dersom vi også trekker inn samfunnsfagene som et vesentlig grunnlag for datafaget, slik andre forfattere gjør, bl.a. Friedman og Bjørn-Andersen.⁴⁵ Da ender vi med den konklusjon at datafaget preges av nesten alle de fagtradisjoner man har i dag innenfor den akademiske verden. Det er mange meninger om dette er et godt utgangspunkt for et nytt fagfelt. De pessimistiske vil hevde at datafaget er for heterogent til at det er en ny disiplin. Slike tanker står i motsetning til det faktum at datafaget tross alt er etablert som et eget fagfelt ved størsteparten av universitetene over hele verden, og at denne utviklingen ikke ser ut til å snu. Sannsynligheten er størst for at tautrekkingen mellom profesjonene om å prege datafaget vil pågå i lang tid ennå så lenge datafaget utvikler sin profil som egen disiplin.

7. AKTØRNETTVERKPERSPEKTIV PÅ DATAFAGET

I det foregående kapitlet har jeg analysert framveksten av datafaget etter de samme prinsipper som Pinch og Bijker har benyttet for å beskrive hvordan sykkelen ble utformet. Riktignok er ikke datafaget en gjenstand («artifact») som sykkelen, men viten («facts»). Her er det snakk om en «seamless web» mellom samfunn og kunnskap, ikke mellom samfunn og teknologi. Uansett typen «seamless web» passer perspektivet som Pinch og Bijker legger til grunn for sin analyse, og som de kaller sosial konstruktivisme (SCOT).⁴⁶

Jeg har identifisert *de relevante sosiale gruppene* som deltok i utformingen av datafaget – universitetene, datamaskinprodusentene, forsvaret. Videre har jeg ved hjelp av en detaljert beskrivelse eller såkalt «thick description» redegjort for de viktigste *problemer og løsninger* som de ulike sosiale gruppene bidro med inntil datafaget ble realisert i den form vi kjenner i dag. I konstruktivistisk sjargong sier vi at datafaget er blitt *stabilisert* som et resultat av denne utviklingen, selv om vi likevel erkjenner at faget vil utvikle

⁴⁴ Peter J. Denning et.al.: «Computing as a Discipline.» *Communications of the ACM*, 32 (1), 1989, s. 10.

⁴⁵ Andrew L. Friedman og Dominic S. Cornford: *Computer Systems Development: History, Organization and Implementation*. Wiley, Chichester, 1989; Niels Bjørn-Andersen (red.): *The Human Side of Information Processing*. North-Holland, Amsterdam, 1980.

⁴⁶ Trevor J. Pinch og Wiebe E. Bijker: «The Social Construction of Facts and Artefacts: Or How the Sociology of Science and the Sociology of Technology Might Benefit Each Other» i Wiebe E. Bijker et.al. (red.): *The Social Contruction of Technological Systems. New Directions in the Sociology and History of Technology*. MIT Press, Cambridge MA, 1987.

seg videre etter hvert som forholdet mellom de ulike sosiale gruppene endrer seg.

Hvis vi skulle gjøre analysen litt tydeligere og vurdere aktørenes bidrag til utviklingen mer inngående, kan vi ta i bruk aktørnettverkteorien. I stedet for relevante sosiale grupper vil vi da konsentrere analysen om en *hovedaktør* som mobiliserer og stabiliserer det som blir kalt et *globalt nettverk*. Hensikten med det globale nettverket er at det støtter hovedaktøren når vedkommende skal drive det man kan kalle prosjektet eller *det lokale nettverket*. Prosjektet utgjør selve utviklingen dersom hovedaktøren lykkes med sin plan om å gjøre seg selv til *obligatorisk passeringspunkt* for de problemer og løsninger som prosjektet omfatter.⁴⁷

Fordelen med aktørnettverkteorien sammenlignet med SCOT er for det første at man fokuserer mer eksplisitt på handlingene som kan knyttes til den sentrale aktøren i stedet for å se på en rekke sosiale grupper, kanskje uten alltid å ta standpunkt til hvor sentrale de ulike gruppene er. For det andre trenger man i SCOT ikke være så opptatt av motivene til den sentrale aktøren eller hvordan vedkommende konstruerer det globale nettverket og oppnår støtte for å styre og kontrollere det lokale nettverket. For det tredje er SCOT ikke så opptatt av konflikter, ustabilitet og maktkamp i nettverkene, mellom nettverkene og mellom hovedaktøren og de andre aktørene. Vanligvis er det ikke avgjørende å vise hvordan de andre aktørene forsøker å utmanøvrere hovedaktøren som obligatorisk passeringspunkt, og begrunne hvorfor de andre aktørene kanskje selv ønsker å overta kontrollen.

Disse tre punktene innebærer at aktørnettverkteorien gir anledning til en grundigere analyse av dynamikken i utviklingen enn SCOT legger opp til. Det kan være en styrke for å oppnå en troverdig analyse av årsakssammenhengene. Ulempen med aktørnettverkteorien er at den krever en mer dyptpløyende analyse enn SCOT. Man må trekke opp flere konfliktlinjer, vurdere flere mulige alternativer eller scenarier og vise stor innsikt spesielt i hovedaktørens motiver og handlinger. I sin tur stiller disse vilkårene store krav til data-materialet, spesielt til dokumentasjonen av kontakten mellom de ulike aktørene. Ofte er det vanskelig å dokumentere interaksjonen mellom aktørene, eventuelt prøve å slutt seg til interaksjonen på grunnlag av foreliggende bakgrunnsdata. Ofte er mye av den viktige interaksjonen av uformell eller flyktig karakter, slik at den ikke går fram av skriftlige dokumenter eller kan påvises i etterhånd gjennom å intervju aktørene.

I mitt tilfelle er det, slik det ble vist i forrige kapittel, lettest å bruke SCOT-tilnærmingen. Det kommer av at jeg har bygd på kilder som gir liten innsikt i de flyktige relasjonene som har preget utviklingen. For meg er det ikke lett å si hvorfor IBM satset på Aiken, hvorfor Aiken kjørte sitt prosjekt inn i en blindgate, eller hvorfor von Neumann ble overbevist om at University

⁴⁷ John Law og Michel Callon: «The Life and Death of an Aircraft: A Network Analysis of Technical Change» i Wiebe E. Bijker og John Law (red.): *Shaping Technology/Building Society. Studies in Sociotechnical Change*. MIT Press, Cambridge MA, 1992.

of Pennsylvania var det beste stedet å oppholde seg for å få utviklet en «general-purpose» datamaskin.

Latour har vist at bruk av aktørnettverksteorien er mulig i beskrivelsen av Pasteurs innsats for å bekjempe miltbrann. Pasteur uteksperimenterte en vaksine og fikk overtalt franske bønder både til å bruke vaksinen og å ta diverse forholdsregler som hindret spredning av sykdommen. Latours beskrivelse av denne innsatsen er hovedsakelig basert på artikler i tre tidsskrifter mellom 1870 og 1919. I det ene tidsskriftet, *Revue Scientifique*, systematiserte han innholdet i artiklene ved å følge aktørenes samhandlingskjeder:

*«For each article I sketched the interdefinition of the actors and the translation chains, without trying to define a priori how the actors were made up and ranked. Without being exhaustive, I nevertheless recorded the great majority of the allusions, however distant, to Pasteur and his microbes throughout the pages of the Revue.»*⁴⁸

Ved å granske primærmaterialet la Latour grunnlaget for å kartlegge det han kaller «implicit translations». Det betyr at Latour kunne dra slutninger om aktørnettverket basert på aktstykker som sentrale aktører hadde etterlatt. I dette tilfellet var det mulig å gå fram på denne måten, fordi Pasteurs ideer vakte stor strid og til dels voldsomme følelsesutbrudd i samtiden. I samsvar med tidens ånd utvekslet både Pasteur og hans motstandere i stor utstrekning synspunkter gjennom tidsskriftene. Det gjør at ettertiden, i dette tilfellet Latour, blir i stand til å lese mellom linjene og rekonstruere nettverket rundt Pasteur og hans mikrober.

I mindre kontroversielle saker eller i forhold hvor aktørene ikke er like ordrike og debattglade som i Pasteurs kontroverser, kan det være vanskelig å anvende aktørnettverksteorien med større hell for å belyse historiske prosesser. De avgjørende handlingene blir gjerne skjult i historiens mørke, og forskeren blir henvist til gjetninger som er egnet til å vekke strid blant kolleger. Dette problemet blir historikerne ofte konfrontert med når kildene er sparsomme. Siden problemet er velkjent, representerer det ikke noen avgjørende innvending mot å bruke aktørnettverksteorien. Men det stiller større krav til forskeren om dokumentasjon som viser troverdigheten i framstillingen enn når man velger å bruke den mindre dyptpløyende SCOT-tilnærmingen.

La meg avslutningsvis forlate den historiske utviklingen av datafaget og se på dagens situasjon, der det er lettere å bruke aktørnettverksteorien. Mye tyder på at universitetene og myndighetene er de to viktigste aktørene i dag. Av disse to vil jeg påstå at universitetene er hovedaktøren.

De to aktørene stiller opp ulike scenarier. Myndighetene står for nyttebetraktningene om datafaget som «vare», slik jeg innledet denne rapporten med å fastslå, altså *nytteaspektet*. Universitetene står for kravet om behovet for

⁴⁸ Bruno Latour: *Pasteurization of France*. Harvard University Press, Cambridge MA, 1988, s. 11. Fransk original: *Les microbes: guerre et paix suivi de irréductions*. Editions A.M. Métailié, Paris, 1984.

større egenutvikling og en bredere rekrutteringsbase for datafaget, et langsiktig perspektiv som jeg for enkelhets skyld vil kalle for *unytteaspektet*. Unytteaspektet er bl.a. blitt formulert slik:

*«The demands of educational institutions and research laboratories for computer scientists are growing faster than the field can build the infrastructure necessary for producing the needed pool of talent. The shortfall in computer science faculty, in particular, has serious ramifications. Unless an investment is made to provide sufficient researchers in our educational institutions, we will fall even further behind in trying to meet this ever-growing demand for computer scientists.»*⁴⁹

Myndighetene sitter med kontrollen over bevilgningene, som i de fleste land er nøkkelen til en større ekspansjon av datafaget ved universitetene. Problemet for universitetene er at myndighetene forsøker å gi pengestøtten på sine egne premisser, altså nytteaspektet. Universitetene forsøker å få akseptert unytteaspektet, som man oftest oppfatter står i strid med myndighetenes ønsker. Denne situasjonen fører til at universitetene legger nye strategier for å unngå det som de ser på som den store faren, nemlig at de både mottar bevilgningene og må oppfylle myndighetenes ønsker. I aktørnettverksteoriens terminologi er det med andre ord en kamp om hvem som skal være hovedaktør og definere det obligatoriske passeringspunkt, dvs. beholde kontrollen på sine premisser.

Situasjonen åpner for en maktkamp som kan arte seg noe ulikt fra land til land. Hvis vi tar Norge som eksempel, fikk vi mot slutten av 1980-årene en nasjonal handlingsplan for informasjonsteknologi.⁵⁰

Sølvberg ved Institutt for databehandling ved NTH kom i 1984 med et første forslag til handlingsplan med et scenario som i sitt vesen tok utgangspunkt i unytteaspektet. Utdanningsinstitusjonene skulle spille en sentral rolle, siden det er på den siden midlene først må settes inn. For å gjøre dette perspektivet salgbart for andre aktører som skulle utgjøre det globale nettverket, bl.a. instituttene for anvendt forskning, IT-industrien og offentlig forvaltning generelt, ble unytteaspektet tilpasset en innovasjonsmodell sentrert rundt en kjede av prosesser som ser slik ut: grunnforskning → anvendt forskning → prototyputvikling → produksjon. Med andre ord ble unytteaspektet innbakt i det som kalles en «technology push»-modell.⁵¹

Dette er et nærmest klassisk eksempel på hvordan en aktør etablerer seg som obligatorisk passeringspunkt, nemlig ved å si at «gjennom å satse på oss oppfylles også målene dere har satt dere.» I dette tilfellet lyktes det å få de

⁴⁹ John E. Hopcroft: «Computer Science: The Emergence of a Discipline.» *Communications of the ACM*, 30 (3), 1987, s. 202.

⁵⁰ *Nasjonal handlingsplan for informasjonsteknologi. Resultater og planer. Statsbudsjettet 1990. Vedlegg til St.prp. nr. 1 (1989-90).*

⁵¹ Trond Bruland: *Forskningspolitikk nedenfra? Nasjonal handlingsplan for informasjonsteknologi blir til.* STS arbeidsnotat nr. 7/88, Trondheim, 1988.

andre aktørene med på denne tankegangen, kanskje mest fordi de ikke hadde noen alternativ plan som virket like samlende på interessene.

Ulempen med en innovasjonsmodell med en så lang kjede av prosesser er at man risikerer at mye av innsatsen foregår i de første prosessene, og at lite eller ingen ting kommer ut av den siste prosessen, i hvert fall på kort sikt. Slik også i dette tilfellet. Det er i ettertid lettest på peke på konkrete resultater av IT-planen for høyere utdanning og forskning i datafag.⁵²

På denne måten kan en si at universitetene lyktes med sitt scenario og fikk stor uttelling for unytteaspektet i denne sammenheng. Ulempen er at seieren bare varte i de årene handlingsplanen ble iverksatt. Unytteaspektet er langt fra akseptert på lengre sikt, og derfor kan vi se for oss en kamp mellom de to scenarioene også i årene framover.

Den norske handlingsplanen for informasjonsteknologi er et godt eksempel på hvordan aktørnettverksteori kan benyttes for å vise at nytte ikke er noe objektivt begrep, men resultatet av en tautrekking mellom ulike interesser. Selv om det jeg har kalt unytteaspektet vant fram denne gang, var det likevel på de fleste hold enighet om at scenarioet støttet opp under forestillingen av «nytte» fordi unytteaspektet var bakt inn i en «technology push»-modell. Slik blir intensjoner og forestillinger vridd og tilpasset nye konstellasjoner av aktører som vil fremme sine mål. Hva som er sant, avhenger av hva man i det enkelte tilfellet forhandler seg fram til som «sannhet».

Når en tenker nærmere etter, kan en på mange måter si at det er en parallell mellom den parlamentariske situasjonen i de nordiske land og situasjonen som universitetene befinner seg i. I begge tilfellene er alle aktørene i en mindretallsposisjon og har så ulike interesser at man fra sak til sak må finne løsninger som ikke er gitt på forhånd. Likevel er det mulig for en og samme hovedaktør å ha kontroll med aktørnettverket lange perioder i gangen ved hjelp av dyktig scenario-planlegging som forener mange aktørers interesser, men som likevel gjør at hovedaktøren får det største utbyttet. Forhandlingene følger ikke alltid vedtatte organisasjonsmønstre eller strukturer, men er ofte aktørdrevet på tvers av slike formelle skiller.

Dette perspektivet skulle tyde på at aktørnettverksteorien også i framtida vil være et nyttig hjelpemiddel til å forstå de sosiale og sosiotekniske prosessene som vi er vitne til både i utviklingen av datafaget og i en rekke andre sammenhenger. kanskje vil den bli vel så viktig som teorier som er utviklet spesielt for å analysere strukturelle eller institusjonelle forhold.

⁵² Trond Bruland: *Den store planen. Norges satsing på informasjonsteknologi 1987-90*. Trondheim, 1992 (under utgivelse som STS rapport), s. 347-366.