

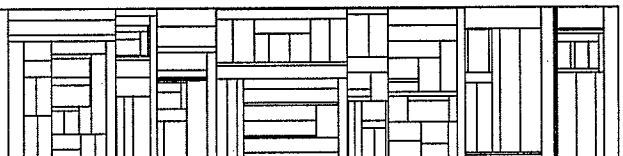
# PROGRAMMERINGSUNDERVISNING OG SYSTEMBESKRIVELSE

Lars Mathiassen

DAIMI PB-60

Juni 1979 (genoptryk)

Matematisk Institut Aarhus Universitet  
**DATALOGISK AFDELING**  
Ny Munkegade - 8000 Aarhus C - Danmark  
Tlf. 06-12 83 55



"Pædagogiske foranstaltninger er altid sekundære eller tertiære i forhold til andre politiske foranstaltninger. Primære er naturligvis de magtforhold, der eksisterer i et vist samfund til et vist givet tidspunkt. Pædagogiske foranstaltninger over en bred front er ikke i sig selv tilstrækkelige til at omdanne samfundet."

(Kallós: Pedagogikk og samhällsförändring,  
DSF 1972, side 219.)

## INDHOLDSFORTEGNELSE

I.	INDLEDNING .....	1
II.	OM UD GANGSPUNKTET .....	3
1	Didaktiske overvejelser	4
1.1	Generelle overvejelser	5
1.2	Specielt om edb-uddannelserne	11
2	Overvejelser om uddannelsesmæssig sammenhæng	16
2.1	Edb-arbejdere	16
2.2	Edb-undervisere	19
2.3	Edb-anvendere	19
3	Teknologiske overvejelser	20
3.1	Programmering, en præcisering	20
3.2	De teknisk betonedede programmeringsfærdigheder	23
3.3	Formalisering contra arbejdsform	25
3.4	Grænseområder	28
4	Sammenfatning	31
III.	SYSTEMBESKRIVELSE SOM INDFALDSVINKEL .....	33
1	Kort om systembeskrivelse	33
1.1	Generelt	33
1.2	Et eksempel på systembeskrivelse	36
2	Argumentation for systembeskrivelse som indfaldsvinkel	42
2.1	Kravet om problemorientering og deltagerstyring	42
2.2	Kravet om sammenhæng	44
2.3	Kravet om en fornuftig arbejdsform	44
2.4	Yderligere fordele	46
3	En skitse af et muligt undervisningsforløb	49
3.1	Organiseret som et særligt delforløb	49
3.2	Delforløbets tre faser	51
3.3	Det anvendte simple systembeskrivelsesværktøj	53
IV.	SAMMENFATNING .....	55
	Noter .....	56
	Litteraturliste .....	61

## I. INDLEDNING

Ud fra en kritisk, social synsvinkel<sup>1</sup> forsøger jeg i denne artikel at diskutere grundlaget for en praksis som programmeringsunderviser. Jeg diskuterer nogle grundlæggende problemstillinger i forbindelse med tilrettelæggelsen og gennemførelsen af en programmeringsundervisning. På grundlag af denne diskussion når jeg frem til et konkret forslag til, hvorledes en programmeringsundervisning kan gennemføres. Forslaget indebærer, at systembeskrivelse anvendes som indfaldsvinkel til selve programmeringsundervisningen.

Nogle overvejelser, som kan danne udgangspunkt for tilrettelæggelsen og gennemførelsen af en programmeringsundervisning, er behandlet i Afsnit II. Først ser jeg kort på det traditionelle udgangspunkt, som kommer frem i hovedparten af de artikler, som indtil nu er skrevet om emnet. Derefter opstiller jeg et alternativt udgangspunkt. Heri fremhæver jeg en række didaktiske (en præcis definition af ordet gives i Afsnit II) overvejelser, jeg ser på hvilke forskellige uddannelsesmæssige sammenhænge en programmeringsundervisning kan indgå i, og endelig trækker jeg en række teknologiske overvejelser frem.

På baggrund af overvejelserne i Afsnit II viser jeg i Afsnit III, hvorledes man med fordel kan anvende systembeskrivelse som indfaldsvinkel til en programmeringsundervisning. Afsnittet indeholder først en kort forklaring på, hvad jeg forstår ved systembeskrivelse. Derefter viser jeg, hvorledes systembeskrivelse kan anvendes til at opfylde de krav, som fremgår af overvejelserne i Afsnit II. Og endelig skitserer jeg, hvorledes en sådan undervisning kan realiseres.

I Afsnit IV trækker jeg artiklens hovedsynspunkter frem.

Bagest i artiklen findes en række noter og en omfattende litteraturliste. Artiklen bygger primært på Illeris' oplæg til en alternativ didaktik (Illeris74) og på overvejelserne omkring systembeskrivelsessproget DELTA (DELTA75). For imidlertid at give læseren mulighed for at få uddybet en række af de synspunkter, som fremsættes i artiklen, har jeg valgt at tilføje noter og den omfattende litteraturliste.

Artiklen er tænkt som en hjælp for den, som står overfor at skulle tilrettelægge og gennemføre en undervisning i programmering.

## II. OM UDGANGSPUNKTET

Gennem det sidste tiår har vi været vidne til en kraftigt voksende brug af elektronisk databehandling. Edb-systemerne er blevet store og komplekse og flere og flere funktioner er integreret i dem. Denne voksende brug af elektronisk databehandling har betydet tilsvarende stigende krav til kvalificeret arbejdskraft til konstruktion og drift af edb-systemer.

Det er på denne baggrund, at den kraftige faglige aktivitet omkring problemerne med at konstruere "gode" programmer og den samtidige interesse for problemerne i forbindelse med undervisning i programmering, skal ses. Nøgleord i den forbindelse er: struktureret programmering, programkorrekthed, programmeringssprog.

Efterhånden er der fremkommet en del artikler, der behandler problemerne i forbindelse med undervisning i programmering. En omfattende litteraturliste findes i Lecarme<sup>74</sup>. Hvert år afholder SIGCSE en konference over emnet 'computer science education'. Herfra udgives SIGCSE BULLETIN. Der har desuden været afholdt en arbejdskonference specielt om problemerne i forbindelse med undervisning i programmering i Zakopane i Polen (Turski<sup>73</sup>). Dette materiale indeholder mange forskellige - og ofte modstridende - synspunkter. Mange af artiklerne indeholder konkrete forslag til, hvorledes en programmeringsundervisning (i 90% af tilfældene for universitetsstuderende) kan gribes an. Ved gennemlæsning af materialet kan det være vanskeligt at vurdere, hvilke synspunkter der er brugbare i en given undervisningssammenhæng. En væsentlig grund hertil er udgangspunktet (eller snarere mangel på samme) i artiklerne.

Gries<sup>74</sup> indledes således med:

- An introductory course (and its successor) in programming should be concerned with three aspects of programming:

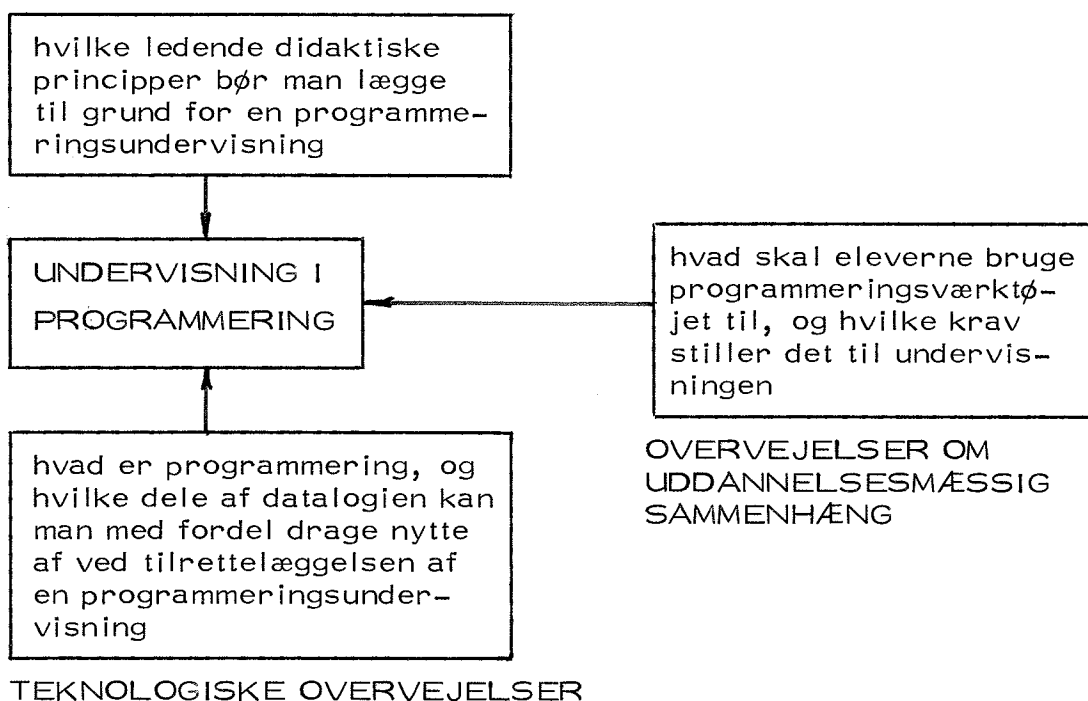
1. How to solve problems.
2. How to describe an algorithmic solution to a problem.
3. How to verify that an algorithm is correct.

Artiklen indeholder ikke nogen egentlig diskussion af, hvorfor det netop er disse tre aspekter, som er af hovedinteresse.

Ingen af artiklerne indeholder en diskussion af, hvorledes de kvalifikationer, som en given programmeringsundervisning giver eleven, relaterer til de krav om kvalifikationer, som den senere erhvervsv funktion stiller. På samme måde indeholder artiklerne heller ingen diskussion af den uddannelsesmæssige sammenhæng, som en given programmeringsundervisning indgår i. Det betyder for de første, at de resultater/konklusioner, som kommer frem i artiklerne, ofte er af tvivlsom værdi. For det andet betyder det, at det er vanskeligt for den læser, der ønsker at finde råd og vejledning med henblik på selv at gennemføre en programmeringsundervisning, at vurdere, hvilke af de mange forslag der er relevante i hendes situation.

I det følgende gennemgås de overvejelser, som jeg har valgt som udgangspunkt for tilrettelæggelsen og gennemførelsen af en programmeringsundervisning. Som nedenstående figur illustrerer, drejer det sig om didaktiske overvejelser. Det drejer sig om overvejelser om den uddannelsesmæssige sammenhæng, som programmeringsundervisningen indgår i – det vil sige overvejelser om, hvad de, man underviser i programmering, skal bruge dette værktøj til. Endelig drejer det sig om overvejelser af teknologisk art.

#### DIDAKTISKE OVERVEJELSER



En programmeringsunderviser må foruden indsigt i overvejelser af ovennævnte karakter også have indsigt i pædagogik og have praktisk erfaring i at håndtere den konkrete undervisningssituation. Mens didaktikken omhandler problemerne i forbindelse med tilrettelæggelsen af undervisning og på den baggrund opstiller generelle retningslinier for undervisningens gennemførelse, udtaler pædagogikken sig om selve undervisningssituationen, om de problemer som opstår, når nogle personer indgår i et lærer-elev forhold. Jeg vil ikke her komme nærmere ind på overvejelser af pædagogisk karakter, blot nævne at didaktikken og pædagogikken tildels er overlappende, idet de ledende principper for undervisningens gennemførelse, som de didaktiske overvejelser fører frem til, klart har betydning for relationerne mellem lærer-elev og elev-elev i selve undervisningssituationen.

Under tilrettelæggelsen af en given programmeringsundervisning må underviseren endvidere have kendskab til rammerne for den givne undervisning: givne formål, deltager-forudsætninger, arbejdsforhold etc.<sup>2</sup>.

Når jeg i det følgende taler om programmeringsundervisning og edb-uddannelser, hentyder jeg til de undervisningsaktiviteter og uddannelser, som eksisterer indenfor det offentlige uddannelsessystem. Det skyldes dels, at udviklingen går i retning af, at det offentlige skal varetage den væsentligste del af undervisningen på edb-området (flere steder er edb-assistent-uddannelsen i gang (Edb-assistent<sup>73</sup>), datanomuddannelsen er lige på trapperne (Datanom<sup>74</sup>), og datalæren er på vej ind i de grundlæggende uddannelser). Dels skyldes det, at det er uhyre vanskeligt at få overblik over de undervisningsaktiviteter, som foregår i privat regi.

### 1. Didaktiske overvejelser

Indenfor didaktikken er der udviklet forskellige værktøjer, som kan anvendes ved planlægningen af uddannelser og ved selve tilrettelæggelsen og gennemførelsen af en undervisning. De forskellige værktøjer giver udtryk for forskellige syn på, hvad didaktik er; men fælles for dem alle er, at de udtaler sig på et generelt plan. I det føl-

gende vil jeg forsøge at anvende de værktøjer, som er udviklet af Illeris (Illeris74), på de konkrete problemstillinger, som findes i forbindelse med undervisning i programmering. Illeris definerer didaktikkens emne som:

Ud fra de betingende samfundsmæssige og psykologisk-teoretiske forhold at opstille og nærmere uddybe nogle generelle principper vedrørende uddannelsernes indre struktur, det vil sige for uddannelsernes planlægning og for selve undervisningens tilrettelæggelse og gennemførelse.

### 1.1 Generelle overvejelser

Dette afsnit er i hovedsagen et resumé af Illeris' generelle overvejelser (Illeris74, s. 22-89).

Illeris ser på den ene side på, hvorledes den samfundsmæssige udvikling medfører ændrede krav til uddannelsessystemet. På den anden side ser han på forskellige indlærings teorier og finder på den baggrund frem til indlæringsformer, der modsvarer de krav, som stilles til uddannelsessystemet.

Udviklingen af produktivkræfterne<sup>3</sup> modsvares af en tilsvarende ændring i kravene til, hvilke kvalifikationer det offentlige uddannelsessystem skal give sine elever. For at kunne beskrive disse forskellige krav indføres forskellige kategorier af kvalifikationer.

Færdighedsmæssige kvalifikationer er de kvalifikationer, der er direkte nødvendige i en given arbejdsproces. De kan konkret defineres ud fra beskaffenheden af den pågældende arbejdsproces og de redskaber eller maskiner, der indgår. Kravet til færdighedsmæssige kvalifikationer varierer, ikke alene fra erhverv til erhverv og fra funktion til funktion, men også indenfor den samme funktion, hvis selve arbejdsprocessen ændres. Der kan skelnes mellem generelle og specielle færdighedsmæssige kvalifikationer, idet der dog er en glidende overgang mellem de to underkategorier.

- Generelle færdighedskvalifikationer er de kvalifikationer, som i et givet samfund søges bibragt individerne uafhængigt af, hvilken erhvervsfunktion de har eller kan forventes at få.



- Specielle færdighedskvalifikationer er de kvalifikationer, som kun søges bibragt individerne med henblik på den erhvervsfunktion, de har eller kan forventes at få.

Tilpasningsmæssige kvalifikationer er derimod principielt af generel karakter og fælles for alle funktioner i et givet samfundssystem. De vedrører den intensitet, hvormed arbejdet udføres: med bedre tilpasningsmæssige kvalifikationer stiger arbejdstempoet, eller man udfører flere funktioner samtidigt – produktionen stiger som følge af en øget arbejdsmængde, mens selve arbejdets karakter og den dygtighed, hvormed det udføres, ikke ændres. Der er således tale om holdningsmæssige egenskaber som flid, udholdenhed, påpasselighed – der modsvares meget præcist af det bestående samfunds normer og værdier. Men for at disse egenskaber kan være af værdi i erhvervene forudsættes en villighed til at bruge dem i arbejdet, til at acceptere og underordne sig de givne arbejdsprocesser. Det indebærer, at man villigt og lydigt gør, hvad man bliver sat til, så godt man kan. Men det omfatter også et mere vidtgående aspekt: at man på et generelt plan accepterer arbejdsforholdene og samfundsforholdene i det hele taget. Der kan skelnes mellem tre underkategorier af tilpasningsmæssige kvalifikationer.

- Aktive tilpasningskvalifikationer er de kvalifikationer, som bevirker at man udfører en given arbejdsproces med den størst mulige intensitet – typiske egenskaber er flid og udholdenhed.
- Direkte accepterende tilpasningskvalifikationer er de kvalifikationer, som bevirker, at man villigt og efter bedste evne udfører de arbejdsprocesser, man bliver sat til – typiske egenskaber er lydighed og pligtfølelse.
- Indirekte accepterende tilpasningskvalifikationer er de kvalifikationer, som bevirker, at man ikke på et generelt plan, uafhængig af de konkrete arbejdsprocesser, engagerer sig i aktiviteter, der kan være til skade for erhvervene – typiske egenskaber er ligegyldighed og apati.

Kreative kvalifikationer kan teoretisk bestemmes som de kvalifikationer, der specielt er påkrævede i forbindelse med den for kapitalen nødvendige stadige udvikling af produktivkræfterne<sup>4</sup>. Der tænkes her i første omgang på udviklingen af produktionsmidlerne og hermed på de egentlige videnskabelige kvalifikationer, der kræves indenfor teknologien i bredeste forstand. Men samtidig bliver der tale om specielle krav til den menneskelige produktivkraft. Den videnskabelige produktion forudsætter typisk egenskaber som kritisk sans, selvstændighed og samarbejdsevne, og videnskabeliggørelsen af produktionen medfører, at tilsvarende kvalifikationskrav forplanter sig "nedad" i systemet. Indførelsen af et nyt og kompliceret maskineri (for eksempel datamaskinen) stiller i nogle tilfælde<sup>5</sup> øgede krav til kritisk tænkning og selvstændighed, ikke mindst når der bliver tale om fejlfinding, og i det omfang maskineriet forudsætter flere individers samvirke (det gør datamaskinen), øges kravene til samarbejdsevnen. Der kan skelnes mellem to underkategorier.

- Samfundsmæssige kreative kvalifikationer er de kvalifikationer, der kræves specielt i forbindelse med den stadige effektivisering af produktionsmidlerne og produktionens organisation – kvalifikationer til egentligt videnskabeligt, nyskabende arbejde.
- Personlige kreative kvalifikationer er de kvalifikationer, der kræves specielt i forbindelse med videnskabeliggørelsen af produktionen, for eksempel ved betjeningen af komplicerede maskiner – kvalifikationer til stadig at forny sig, at kunne indgå i nye funktioner, at kunne samarbejde.

Der vil i begge tilfælde typisk kunne henvises til egenskaber som kritisk sans, selvstændighed, åbenhed og evne til at samarbejde.

Disse kvalifikationskategorier kan benyttes til at beskrive udviklingen i kvalifikationskravene set i relation til den historiske og den aktuelle udvikling i produktionsforholdene. Jeg skal ikke her komme nærmere ind på den historiske udvikling, men blot referere Illeris' fremstilling af, hvorledes nogle få aktuelle, centrale tendenser indenfor kapitalismen og industrialismen relaterer til kravene til kvalifikationer. For det første er der tale om en tiltagende teknificering og automatisering. Det

har i første omgang betydet en reduktion i behovet for specielle færdighedskvalifikationer. Men i den videre udvikling er behovet for den helt "ukvalificerede" arbejdskraft formindsket: arbejderne skal kunne indstille, betjene, overvåge, finde fejl og reparere. De anvendte maskiner bliver mere komplicerede og dyrere. Det får alvorlige økonomiske konsekvenser, hvis maskinerne går i stå, hvorfor der stilles øgede krav om pålidelighed og ansvarlighed – typiske accepterede tilpasningskvalifikationer. For det andet er der tale om en samtidig differentiering og voksende kompleksitet i selve arbejdsprocesserne. Det betyder, at der af en stadig større del af arbejdskraften kræves overblik, samarbejdsevne og smidighed, og stadig flere må deltage i planlægningen. Dette stiller øgede krav om generelle færdighedskvalifikationer og i særdeleshed om personlige kreative kvalifikationer<sup>6</sup>. Samtidig betyder differentieringen, at der stilles øgede krav om specielle færdighedskvalifikationer, men disse er ofte så specielle, at de først indlæres "on the job". En vigtig opgave for uddannelsessystemet bliver at lære eleverne at lære. Dette hænger stærkt sammen med den tredie tendens: arbejdsprocesserne udvikler sig i en stadig hurtigere takt. Arbejdskraften skal være fleksibel. Dette stiller øgede krav om accepterende tilpasningskvalifikationer – arbejderne skal finde sig i ændringer – og for en dels vedkommende om personlige kreative kvalifikationer – arbejderne skal kunne forny sig.

Sammenfattende når Illeris frem til, at accelerationen i udviklingen af produktivkræfterne især stiller øgede krav til uddannelsessystemet om produktion af generelle færdighedskvalifikationer, accepterende tilpasningskvalifikationer, samt personlige og for en mindre del af arbejdskraften desuden samfundsmæssige kreative kvalifikationer.

Efter denne korte gennemgang af de betingende samfundsmæssige forhold vender Illeris sig mod de psykologisk –teoretiske forhold. For at kunne finde ud af, hvorledes uddannelsessystemet kan opfylde kvalifikationskravene, må disse sættes i relation til forskellige former for indlæring. I sin gennemgang af de psykologisk–teoretiske forhold når Illeris frem til tre forskellige former for indlæring.

Kumulativ indlæring er en indlæringsform, hvor indlæringsproduktet indgår som et isoleret element i elevens cognitive struktur (d.v.s. organisationen, stabiliteten og klarheden af den viden, som eleven sidder inde med). Denne cognitive struktur danner grundlag for elevens evne til at opfatte og forstå en given situation og derfor også for elevens evne til at handle i en given situation. Eksempler på denne form for indlæring er udenadslæren og indlæring af motoriske færdigheder (f. eks. cykelkørsel). Populært kan denne indlæringsform nærmest karakteriseres som dressur.

Assimilativ indlæring er en indlæringsform, hvor eleven tilpasser og indarbejder sine erfaringer som en udbygning og differentiering af allerede etablerede cognitive strukturer. Under denne indlæringsform hører størstedelen af den videns- og færdighedsindlæring, der traditionelt tilsigtes indenfor uddannelsessystemet. Populært kan denne form for indlæring betegnes almindelig skole indlæring.

Akkomodativ indlæring er en indlæringsform, hvor elevens cognitive strukturer ændres gennem nedbrydning, idet tidligere etablerede indlæringselementer frigøres fra den oprindelige indlæringssammenhæng og derefter frit kan indgå i nye strukturer. Eleven optrænes således til at opfatte og forstå situationer, som indgår i nye og anderledes sammenhænge. Denne form for indlæring medfører derfor, at eleven får øgede handlingsmuligheder til sin rådighed. Populært kan den nærmest betegnes som bevidstgørelse.

Indenfor uddannelsessystemet tilsigtes og etableres i dag overvejende assimilativ indlæring. Denne form for indlæring tager imidlertid ikke højde for kravet om produktion af kreative kvalifikationer. En forudsætning for, at et uddannelsesforløb kan leve op til kravet om produktion af såvel generelle færdighedskvalifikationer, accepterende tilpassingskvalifikationer som kreative kvalifikationer er, at det består af overvejende assimilative og overvejende akkomodative faser, som til stadighed afløser hinanden.

"I de assimilative faser tilføres og indarbejdes nyt materiale i individets etablerede cognitive strukturer, der dermed konsolideres og differentieres. I de akkomodative faser sker der en dissociering, en frigørelse af det indlærte fra de oprindelige sammenhænge, således at nye cognitive strukturer kan dannes. Disse må igen assimilativt konsolideres og differentieres, så man igen er "moden" til en akkomodativ nystrukturering, o. s. v." (Illeris74 side 77)

På grundlag af disse overvejelser omkring de betingende samfundsmæssige forhold og de psykologisk-teoretiske forhold opstiller Illeris to fundamentale didaktiske principper for uddannelsernes udformning. Disse principper tager højde for de opstillede kvalifikationskrav og bygger på de indlæringsformer, som modsvarer disse krav. Det ene grundlæggende princip er problemorientering. Det centrale ved den problemorienterede undervisningsform er, at udgangspunktet ikke tages i de traditionelle fag, men derimod i foreliggende problemer, som findes her og nu. Ved behandlingen af det pågældende problem inddrages de forskellige fags viden, metoder og teorier i det omfang, som det ud fra den givne problemstilling er relevant. Problemorienteringen bryder således med den traditionelle fagopdeling og muliggør derved et undervisningsforløb, hvori overvejende assimilative og overvejende akkomodative faser til stadighed afløser hinanden. Det andet grundlæggende didaktiske princip er deltagerstyring, hvor der med deltager menes de i undervisningen direkte involverede personer – eleverne, lærerne og eventuelt andre. Disse to grundlæggende didaktiske principper forudsætter gensidigt hinanden. De udtaler sig blot om to sider af samme sag, idet problemorienteringen angiver en måde at organisere undervisningens indhold på, mens deltagerstyringen derimod udtaler sig om organiseringen af selve undervisningsprocessen.

Illeris har således fremhævet, hvorledes de betingende samfundsmæssige forhold stiller bestemte krav til de kvalifikationer, som det offentlige uddannelsessystem skal bibringe eleverne, og hvorledes disse krav udmønter sig i krav til uddannelsernes udformning. Jeg vil i det følgende vende mig mod de faktiske forhold indenfor edb-uddannelserne og se på dem i lys af Illeris' generelle overvejelser.

## 1.2 Specielt om edb-uddannelserne

I dette afsnit vil jeg se på edb-uddannelserne som helhed. Senere vil jeg indskrænke mig til at se specielt på programmeringsundervisningen. Jeg starter med at se på målene for de forskellige edb-uddannelser. Derefter ser jeg på uddannelsernes faktiske udformning.

I målformuleringerne for edb-uddannelserne er der flere forhold, der underbygger, at der stilles øgede krav om generelle færdighedskvalifikationer, accepterende tilpasningskvalifikationer samt kreative kvalifikationer.

Datalære er på vej ind i de grundlæggende uddannelser (Folkeskole, Gymnasiet, HF, EFG). Flere steder kører undervisningen allerede på forsøgsbasis. I Johnsen-rapporten (Betænkning72) foreslås det, at datalæren indtil videre skal indgå som en valgmulighed i Folkeskolen, Gymnasie og HF, men at man på længere sigt bør overveje at gøre undervisningen obligatorisk. I EFG-uddannelsen indgår datalæren allerede obligatorisk i et vist omfang. Dette forhold viser, hvorledes der på det konkrete plan stilles øgede krav om generelle færdighedskvalifikationer.

Ser vi på mål- og indholdsformuleringerne for undervisningen i datalære indenfor de grundlæggende uddannelser, finder vi flere eksempler på krav om accepterende tilpasningskvalifikationer og kreative kvalifikationer. Som eksempel vil jeg trække en del af formålsformuleringen for EFG frem.

"Formålet med datalære er at ..... fremme den enkelte elevs mulighed for kritisk stillingtagen samt afmystificering af edb-teknikken og anvendelser heraf ....."

(EFG75) . I dette eksempel ses, hvorledes der såvel fremhæves kreative kvalifikationer (kritisk stillingtagen) som accepterende tilpasningskvalifikationer (afmystificering af edb-teknikken og anvendelser heraf).

Målet for edb-assistent uddannelsen er:

- a) at en edb-assistent skal kunne medvirke ved udførelsen af edb-arbejdets forskellige faser, som planlægning, programmering, kørsel og dokumentation,
- b) at en edb-assistent efter uddannelsen skal have bred viden, der muliggør en videregående uddannelse (Edb-assistent73).

Kravet om generelle færdighedskvalifikationer kommer her til udtryk, idet formålet udtrykker, at det ikke er tilstrækkeligt at give eleven de færdighedsmæssige kvalifikationer, som den senere erhvervsfunktion kræver i dag. Udviklingen af produktivkræfterne - specielt den teknologiske udvikling indenfor edb-området - foregår i et sådant tempo, at edb-assistenten må have et grundlag (for eksempel indenfor sprog, matematik og fysik), som gør det muligt via en videreuddannelse at holde trit med denne udvikling.

Ved edb-skolen i Aarhus - et af de steder, hvor man uddanner edb-assistent - er der mødepligt. Mødepligten administreres ved hjælp af et stempelur (Edb-skole75). Blandt andet på denne

måde er der i undervisningen indbygget lydighed og pligtfølelse – typiske accepterende tilpasningskvalifikationer. Sammen med eksamensbeviset kan eleven få en udtalelse fra skolen. Denne udtalelse indeholder udsagn om blandt andet elevens evne til at omgås andre og samarbejde med andre (Edb-skole75), hvilket viser, at der også lægges vægt på kreative kvalifikationer.

Ovenstående enkeltteksempler viser, hvorledes de generelle udviklingstendenser indenfor kvalifikationskravene, som Illeris opstillede på baggrund af en analyse af udviklingen i produktionsforholdene, genfindes i edb-uddannelserne. De indlæringsformer (forløb bestående af overvejende assimilative og overvejende akkomodative faser, som til stadighed afløser hinanden) og de generelle didaktiske principper (problemorientering og deltagerstyring), som Illeris på baggrund af en analyse af de psykologisk-teoretiske forhold opstiller som forudsætninger for opnåelsen af de ønskede kvalifikationer, kunne vi derfor forvente at genfinde i edb-uddannelsernes udformning. Det viser sig imidlertid ikke at være tilfældet.

Inden vi går over til at se på dette forhold, skal det understreges, at det ikke er givet, at mål- og indholdsbeskrivelserne for edb-uddannelserne afspejler de reelle krav til kvalifikationer indenfor edb-sektoren. En analyse af disse reelle krav til kvalifikationer må tage udgangspunkt i arbejdspladsstrukturens udvikling, idet arbejdspladsstrukturen netop afspejler, hvilken fordeling af kvalifikationer som de eksisterende arbejdspladser under den givne produktionsmåde kræver (Andersen73 side 16-18). Der er imidlertid visse vanskeligheder forbundet med at undersøge udviklingen af arbejdspladsstrukturen indenfor edb-sektoren. For det første er denne form for arbejdspladser, set i historisk perspektiv, af meget ny dato. For det andet foregår udviklingen indenfor området i et sådant tempo, at en undersøgelse let bliver forældet inden den er færdig<sup>7</sup>.

Selvom det således ikke er muligt på baggrund af eksisterende undersøgelser at underbygge de opstillede kvalifikationskrav med en analyse af de objektivt givne forhold indenfor edb-sektoren, mener jeg, at der er en del praktiske erfaringer, som taler for, at disse kvalifikationskrav er i overensstemmelse med de reelle krav til kvalifikationer. Deltagelsen i konstruktionen af de efterhånden store og komplekse edb-programmer kræver i vidt omfang samarbejdsevne og evne til selvstændigt at vurdere – altså udprægede kreative kvalifikationer. Samtidig kræver den hurtige teknologiske udvikling indenfor edb-området en række generelle færdighedsmæssige kvalifikationer af den enkelte edb-medarbejder, hvis hun skal kunne holde trit med denne udvikling. Endelig stilles der til stadighed krav om accepterende tilpasningsmæssige kvalifikationer, også indenfor edb-området.

Lad os herefter vende os mod edb-uddannelsernes faktiske udformning. Som allerede nævnt, er der her en del forhold, som går imod de grundlæggende didaktiske principper om problemorientering og deltagerstyring. Dette stemmer nøje overens med Illeris' egen kommentar om, at der idag indenfor uddannelsessystemet helt overvejende tilsigtes og etableres assimilativ indlæring (Illeris<sup>74</sup> side 76).

Datalæren foreslås indført enten som selvstændigt fag eller som en del af matematikundervisningen (Betænkning<sup>72</sup>). Dette strider mod det grundlæggende didaktiske princip om problemorientering.

Af uddannelsesplanerne for datanomuddannelsen (Datanom<sup>74</sup>) fremgår klart, at man også her lægger vægt på generelle færdighedskvalifikationer, accepterende tilpasningskvalifikationer samt kreative kvalifikationer. Således finder man på side 17

"Det der skal lægges vægt på er altså overalt (1) de praktiske aspekter ved de faglige problemstillinger, (2) evnen til både at arbejde selvstændigt og at samarbejde, og evnen til at fremstille, forklare og dokumentere faglige problemer eller resultater." Og to sider længere fremme finder vi under afsnittet Holdninger "datanomen bør være .... afslappet over for gamle vaner og forestillinger, men positiv over for tilegnelsen af nye." (datanomen bør være afslappet, hvis man på grund af udviklingen af teknologien ændrer forholdene på arbejdspladsen – min kommentar).

På trods af at man i den overordnede målformulering lægger vægt på at eleven skal bibringes kreative kvalifikationer – hvilket forudsætter et undervisningsforløb, hvori der indgår overvejende akkomodative faser – finder vi på side 24 i uddannelsesplanerne "Med hensyn til uddannelsesformen er det klart, at privatister vil foretrække en så modulær struktur



som muligt, mens en udpræget liniestruktur ikke harmonerer med denne gruppes ønske om kortvarige uddannelsesforløb. Da modulstrukturen ved passende valg af modulrækkefølge i realiteten omfatter en lang række af muligheder for forskellige liniestrukturer, finder udvalget, at en modulær struktur klart må foretrækkes ved udformningen af datanomuuddannelserne.

Hvad endelig undervisningsform angår, så vil en vis disciplinorientering være at foretrække, dels fordi dette harmonerer bedst med modulstrukturen, men også fordi privatisterne for en stor dels vedkommende vil møde med forholdsvis specialiserede ønsker om korte, intensive uddannelsesforløb inden for bestemte, afgrænsede emneområder."

Rammerne for edb-assistentuddannelsen er i kraft af den detaljerede indholdsbeskrivelse i Edb-assistent<sup>73</sup> og i kraft af den statskontrollerede prøve så snævre, at eleverne kun kan få indflydelse på indholdet af undervisningen i det fag, som hedder datamatisk samfundslære. Undervisningen i dette fag udgør for den 1-årige uddannelses vedkommende 38 ud af 1208 timer.

Disse eksempler viser, hvorledes den konkrete udformning af edb-uddannelserne gør det vanskeligt at lade de didaktiske principper om problemorientering og deltagerstyring danne grundlag for undervisningen.

Kigger vi nærmere på edb-uddannelserne ser vi altså, at der på den ene side i mål- og indholdsbeskrivelserne i væsentlig grad lægges vægt på at bibringe eleven generelle færdighedskvalifikationer, accepterende tilpasningskvalifikationer samt kreative kvalifikationer. På den anden side gør uddannelsernes konkrete udformning det uhyre vanskeligt at lade de didaktiske principper, som netop er en forudsætning for at eleven virkelig bibringes de krævede kvalifikationer – nemlig princippet om problemorientering og deltagerstyring – at lade disse danne grundlag for undervisningen.

Den modsætning som således fremtræder skyldes flere forhold. For det første skyldes den, at den eksisterende produktionsmåde, den kapitalistiske produktionsmåde, har dobbeltkarakter som en samtidig arbejdsproces og værdiøgningsproces.<sup>8</sup> De modsætninger, der af den grund findes i produktionsprocessen, forplanter sig til modsætninger i kravene

til arbejdsstyrkens kvalifikationer, der "giver sig udtryk i vanskeligheder i at tilpasse kvalifikationsstruktur til arbejdspladsstruktur, og som dermed på samme tid begrundet og sætter grænserne for den statslige regulering af tilpasningen" (Andersen<sup>73</sup> side 42). Et eksempel på en sådan modsætning i kravene til kvalifikationer er kravet om på den ene side øgede accepterende tilpasningskvalifikationer og på den anden side øgede kreative kvalifikationer. Accepterende tilpasningskvalifikationer indebærer på den ene side at individet skal acceptere og underordne sig de eksisterende arbejdsforhold og samfundsforhold. Kreative kvalifikationer indebærer på den anden side at eleven er selvstændig og kritisk. Men en sådan selvstændighed og kritisk sans kan vanskeligt begrænses, så den ikke omfatter selvstændig stillingtagen og efterfølgende kritik af de eksisterende arbejds- og samfundsforhold. For det andet skyldes denne modsætning en kolossal træghed i selve uddannelses-systemet. For eksempel er lærerne i Gymnasiet i dag som regel kun uddannet i to fag. Sammenholdes dette med ressourcetildelingen til Gymnasiet, indses vanskeligheden af at arbejde for nedbrydningen af faggrænserne på det konkrete plan. Af andre hæmmende faktorer kan nævnes kravet om ekstern evaluering (som ikke harmonerer med princippet om deltagerstyring) og lokalernes konkrete udformning på uddannelsesinstitutionerne.

Sammenfattende kan vi ud fra en kritisk, social synsvinkel sige, at vi ved tilrettelæggelsen og gennemførelsen af edb-uddannelserne og herunder specielt programmeringsundervisningen bør benytte de generelle didaktiske principper om problemorientering og deltagerstyring i det omfang, som det er muligt. Hvorledes dette bedst lader sig gøre er behandlet i Illeris<sup>74</sup>. Problemorientering og deltagerstyring sikrer dog på ingen måde at undervisningen bliver tilfredsstillende ud fra en kritisk, social synsvinkel. Disse didaktiske principper, som er udledt af en række objektivt givne forhold, og som således er i nøje overensstemmelse med kapitalens ønsker og krav, er blot en forudsætning for at undervisningen blandt andet gennem en omhyggelig indholdsudvælgelse kan blive tilfredsstillende ud fra en kritisk, social synsvinkel.

## 2. Overvejelser om uddannelsesmæssig sammenhæng

I det foregående har jeg set på edb-uddannelserne indenfor det offentlige uddannelsessystem som helhed. I det følgende vil jeg indskrænke mig til at se på den del af disse uddannelser og andre uddannelser, der drejer sig om undervisning i programmering. For at skaffe overblik over hvilke forskellige uddannelsesmæssige sammenhænge, programmeringsundervisningen vil komme til at indgå i, vil jeg foretage en opdeling efter elevernes senere arbejde.

I en programmeringsundervisning vil deltagerne således enten være kommende

- edb-arbejdere
- edb-undervisere    eller
- edb-anvendere.

I det følgende vil jeg begrænse mig til ud fra denne simple opdeling at se på, hvilke kvalifikationer programmeringsundervisningen skal bidrage med. Jeg vil i disse overvejelser se bort fra det oplagte forhold, at programmeringsundervisningen skal lære eleverne at konstruere og indkøre programmer i et eller eventuelt flere programmeringssprog. I stedet vil jeg koncentrere mig om de yderligere kvalifikationer, som programmeringsundervisningen skal bidrage med.

### 2.1 Edb-arbejdere

Edb-arbejdere er karakteriseret ved, at de i deres arbejde deltager i anvendelsen af edb som værktøj. Edb-arbejdere omtales ofte som edb-specialister eller systemspecialister, og de omfatter systemanalytikere, systemprogrammører, programmører og operatører. Sædvanligvis fremstilles udviklingen af et edb-system som en proces, der er opdelt i faser. Disse faser modsvares af en tilsvarende arbejdsdeling indenfor edb-sektoren. Andersson fremfører, at disse faser ikke kan skilles klart ud fra hinanden, og slet ikke i tid (Andersson74). Jeg har derfor i dette afsnit valgt at se på anvendelsen af edb som én samlet proces.

Udviklingen af datamaskinen som produktivkraft har haft vidtrækkende konsekvenser. Udviklingen i anvendelsen af datamaskinen er gået fra små afgrænsede edb-systemer mod store og integrerede systemer. Disse edb-systemer har i stigende grad berørt problemstillinger, som er cen-

trale i forhold til magtforholdene og arbejdspladsforholdene i det hele taget indenfor såvel privat som offentlig virksomhed: selve beslutningsprocessen på virksomheden, arbejdstempoet, den enkeltes mulighed for at overskue sin arbejdssituation og solidariteten mellem de ansatte. De argumenter, som har dannet udgangspunkt for udviklingen af systemerne, har været af teknisk-økonomisk karakter. Ofte har de parter, som berøres af et givet system, været holdt udenfor beslutningerne under udviklingen af systemet. Et eksempel som illustrerer dette forhold, er udviklingen af det landsomfattende bibliotekssystem FAUST (Folkebibliotekernes Automations System).

## 1700 protester mod nyt biblioteks-edb

*Sagen behandles hos Kommunedata i Århus i dag for lukkede døre*



## Bibliotekets datacentral afslører låneres vaner

*Systemet skal give bedre service, men det er umuligt at hindre misbrug, hævder bibliotekarerne*

Ehn formulerer problemstillingen på følgende måde:

"... datateknologien vil medføre gigantiske muligheder for manipulation og kontrol ikke bare af de fysiske omgivelser, men også af mennesket selv. Om de autoritære og dehumaniserende muligheder vil blive realiseret, eller om andre potentielt mulige og frigørende muligheder vil blive realiseret, er ikke et spørgsmål om teknisk determinisme, men et spørgsmål om politiske muligheder. Datamaskinerne foretager sig ikke noget, uden at nogen beslutter, hvad de skal foretage sig, og spørgsmålet om hvem der skal have kontrollen, er ikke teknisk men politisk. Det burde være oplagt at den sociale udformning af disse systemer ejheller kan reduceres til et teknisk spørgsmål, men må betragtes som et socialt problem, et menneskeligt spørgsmål, et spørgsmål om vurderinger og interesser. Et sådant spørgsmål kan ikke løses af "eksperter", eller anderledes udtrykt, i et sådant spørgsmål er alle eksperter. Spørgsmålet har en politisk løsning." (Ehn74 side 12).

I uddannelsen af edb-arbejdere og specielt i programmeringsundervisningen skal dette forhold formidles. Programmering er et værktøj, som edb-arbejderen benytter i forskellige sammenhænge. Forskellige grupper vil have forskellige meninger om, hvordan dette værktøj skal anvendes i en given sammenhæng. Og da disse meninger ofte udtrykker modstridende krav, vil løsningen ikke eksistere.

Edb-arbejderen vil selv indgå i den sociale sammenhæng, som edb-systemet skal udvikles i og indgå i. Og edb-arbejderen vil gennem udviklingen og indføringen af edb-systemet forandre denne sociale virkelighed. Det er derfor muligt at drage en parallel mellem edb-arbejderens og sociologens situation. For sociologen gælder det, at den måde, hun opfatter samfundet på, indvirker på de teorier, hun konstruerer. Disse bestemmer atter, hvilke resultater hun når frem til. Med andre ord: Har vi valgt (og det gør vi, selvom vi ikke eksplicit giver udtryk for det) grundlæggende antagelser om menneskets natur, samfundets natur og forholdet mellem disse, har vi også indirekte bestemt, hvilken type af viden vi skaffer os om samfundet (Israel74 side 66). Tilsvarende må det derfor gælde om edb-arbejderen, at den måde hun opfatter samfundet og sine omgivelser på er afgørende for, hvilken type af viden hun skaffer sig som grundlag for konstruktionen af edb-systemer. Edb-arbejderens grundlæggende antagelser vil derfor indirekte være medbestemmende om, hvilke edb-systemer hun vil konstruere. Dette forhold skal belyses i undervisningen.

Undervisningen skal endvidere indeholde en diskussion af, hvorledes forskellige løsninger – forskellige edb-systemer – kan vurderes. Ud fra et kritisk socialt perspektiv

"... bør man sætte spørgsmålstegn ved ethvert forslag til et nyt edb-system, som

- ikke øger de berørte menneskers udviklingsmuligheder
- skaber et for de berørte mennesker usundt fysisk eller psykisk miljø
- for systemets introduktion kræver manipulation af de berørte menneskers holdninger eller bestræbelser".

(Ehn74 side 52)

Sammenfattende kan det siges, at det i programmeringsundervisningen er vigtigt at tage udgangspunkt i den sammenhæng, som edb-arbejderen vil komme til at benytte programmeringsværktøjet i.

## 2.2 Edb-undervisere.

De kvalifikationer, som en edb-underviser må have, afhænger i høj grad af på hvilket trin i uddannelsessystemet, vedkommende skal undervise. Generelt må edb-underviseren have kvalifikationer, som sætter hende i stand til at tilrettelægge og gennemføre en given undervisning, som indeholder edb-aspekter.

Da disse noter netop omhandler de grundlæggende problemer i forbindelse med undervisning i programmering, kan noterne som helhed siges at give udtryk for de synspunkter, som hører hjemme i dette afsnit.

## 2.3 Edb-anvendere.

I diskussionen af den tredje kategori vil jeg skelne mellem den undervisning, som finder sted indenfor de grundlæggende uddannelser (Folkeskolen, Gymnasiet, HF, EFG), og den som finder sted indenfor de videregående uddannelser (universiteterne, seminarierne, handelshøjskolerne, ingeniøruddannelserne, biblioteksskolen ....).

Indenfor de grundlæggende uddannelser skal der ifølge Johnsen-rapporten (Betænkning<sup>72</sup>) undervises i programmering i større eller mindre omfang. Tænker man på, at de grundlæggende uddannelsers hovedsigte er at give eleverne almene kvalifikationer (i modsætning til specielle kvalifikationer, som sigter mod den senere erhvervsfunktion), må der sættes spørgsmålstegn ved, om det overhovedet er på sin plads at undervise i programmering på dette trin.<sup>9</sup>

Indenfor de videregående uddannelser indgår undervisning i programmering som et led i flere af edb-anvenderuddannelserne. Programmering bliver for disse elever et værktøj, som de vil kunne bruge indenfor deres eget felt. I undervisningen i programmering er det derfor vigtigt, at man sætter eleven i stand til at kommunikere med sit eget fagområde. I undervisningen af kemikere må der indgå overvejelser

om hvilke typer af problemstillinger indenfor kemien, som egner sig til løsning ved hjælp af en datamat. Og de øvrige datalogiske værktøjer – for eksempel numerisk analyse og processtyring – som er af betydning for anvendelsen af datamaten indenfor det pågældende fagområde må belyses. I undervisningen af edb-anvendere som senere skal være lærere indenfor et eller andet fagområde, må der ligeledes indgå overvejelser om, hvorledes datamaten kan indgå som et pædagogisk hjælpemiddel i undervisningen.<sup>10</sup>

Sammenfattende kan man sige, at når det drejer sig om undervisning af edb-anvendere, må man overveje om det i den givne sammenhæng er på sin plads at undervise i programmering. I undervisningen i programmering af edb-anvendere er det vigtigt at tage udgangspunkt i sammenhængen, det vil sige edb-anvenderens eget fag- eller problemområde, og vise, hvorledes programmeringsværktøjet kan anvendes indenfor disse rammer.

### 3. Teknologiske overvejelser.

I det foregående har vi set på en række didaktiske overvejelser, og vi har set på overvejelser om den uddannelsesmæssige sammenhæng, som programmeringsundervisningen indgår i. I det følgende vil vi vende os mod overvejelser af teknologisk art – det tredje og sidste af de aspekter, som jeg i denne fremstilling har valgt som udgangspunkt for tilrettelæggelsen og gennemførelsen af en programmeringsundervisning.

Afhængig af hvem man skal undervise i programmering, stilles der forskellige krav til, hvorledes der skal undervises. Et fælles træk er imidlertid, at man i undervisningen må formidle værktøjet som en del af den sammenhæng, hvori det engang skal indgå. For at præcisere dette forhold vil jeg nedenfor definere programmering ud fra denne sammenhæng. Desuden vil jeg se på, hvad der er centralt i forhold til en formidling af selve programmeringsværktøjet.

#### 3.1 Programmering, en præcisering.

Programmering er en del af en edb-anvendelse, og en edb-anvendelse er en arbejdsproces, som foregår indenfor et system, og som har til formål at forandre dette system. Drejer det sig om konstruktion af et planlægnings- og styringssystem i en privat virksomhed omfatter systemet arbejderne, maskinerne, råvarerne, virksomhedsledel-

sen etc. og relationerne mellem disse. Drejer det sig om konstruktion af et program til styring af et kemisk forsøg omfatter systemet kemikeren, de indgående stoffer, måleapparatet etc. og relationerne mellem disse.

Ved et system vil vi forstå en del af verden, som vi vælger at betragte som helhed, adskilt fra resten af verden i en periode, en helhed som vi vælger at betragte som bestående af en samling komponenter som hver er karakteriseret ved en datastruktur og ved handlinger, som involverer den selv og andre komponenter.

Denne definition på et system er næsten den samme, som den der danner udgangspunkt for systembeskrivelsesværktøjet DELTA (DELTA75). Denne definition på et system skal ses specielt i sammenhæng med, at det vi ønsker at præcisere er, hvorledes programmering og edb-anvendelse skal forstås i en større sammenhæng. Den er lavet med det udgangspunkt, at systemet skal forandres gennem en edb-anvendelse. En edb-anvendelse anvender datamaskinen som det centrale værktøj. Og datamaskinen kræver, at den virkelighed, det system, som den skal arbejde med, kan måles og opdeles. Derfor ligestiller denne definition af et system mennesker, maskiner, råvarer, måleapparater etc. Den taler blot om komponenter, som hver er karakteriseret ved en datastruktur og et handlingsmønster. Den afspejler ikke, at mennesket er en komponent, som i mange væsentlige henseender er forskellig fra andre komponenter: Mennesket er et socialt individ, og mennesket har via sine handlinger mulighed for at forandre den virkelighed, det system, som det indgår i.

Denne definition siger også, at vi vælger at betragte en del af verden som en helhed adskilt fra resten af verden i en periode. Dette er en grov, men igen i denne sammenhæng nødvendig, forenkling. Praktisk taget alle de systemer, som bliver ændret gennem edb-anvendelser, er åbne systemer. Et åbent system er et system, som indgår i en vekselvirkning med et mere omfattende system. Er det system, vi betragter, for eksempel en virksomhed, må vi se på dette system som en del af en større helhed for at kunne forstå, hvorledes systemet fungerer. Vi må blandt andet forstå, at virksomheden er underlagt de gældende konkurrencebetingelser, og derfor ikke blot kan foretage vilkårlige lokale beslutninger.



Ved hjælp af datamaskinen vil man imidlertid altid betragte et system som værende adskilt fra resten af verden, idet datamaskinen kun kan arbejde på en endelig informationsmængde. Desuden vil man betragte verden som bestående af lutter målelige størrelser, idet datamaskiner kun kan forstå informationer, som kan omsættes til tal.

Denne definition på et system udtrykker således, hvordan man ved hjælp af datamaskiner opfatter virkeligheden. Den udtrykker de begrænsede muligheder, der ligger i anvendelsen af datamaskinen som værktøj. Som sådan kan den benyttes til at indplacere begreberne edb-anvendelse og programmering i en større sammenhæng.

Ved en edb-anvendelse vil vi forstå den arbejdsproces, som består i på en gang at

- afgrænse et system og
- ændre dette system ved at udvikle og indføre et edb-program, som via dets udførelse på et edb-anlæg, kommer til at indgå i det ændrede system.

Afgrænsningen af systemet og udviklingen af edb-programmet skal forstås som samtidige processer. Ved mange edb-anvendelser forholder det sig sådan, at det på forhånd er umuligt at afgrænse den stump virkelighed, som er central i forhold til edb-anvendelsen. Først gennem udviklingen af edb-programmet bliver der skabt klarhed over, hvilket system man vælger at betragte. Og det er faktisk først i det øjeblik edb-programmet er færdigudviklet og indført, at systemet er endeligt afgrænset. En af fejlene ved en række af de ofte anvendte metoder til edb-anvendelse er netop, som nævnt tidligere, at de betragter arbejdsprocessen som opdelt i faser: Systemanalyse, systemudformning og systemindføring.

Ændringen af systemet skal forstås bredt. Såvel under udviklingen som under indførelsen af edb-programmet ændres både de i systemet indgående komponenter og deres indbyrdes relationer og den virkelighed ( det sociale system), som systemet er en del af. Ændringen sker for eksempel gennem de samtaler og forhandlinger, som føres under

udviklingen af edb-programmet. Ændringen sker også gennem den undervisning, som stables på benene i forbindelsen med udviklingen og indførelsen af et edb-program i en privat eller offentlig virksomhed, og som har til formål at give medarbejderne en forståelse for de trufne dispositioner. Eller ændringen sker i det øjeblik kemikeren ændrer i forsøgsopstillingen med henblik på lettere at kunne skabe kommunikation mellem den kemiske proces og datamaskinen.

Ved en programmering vil vi forstå den del af en edb-anvendelse, som består i ud fra et givet oplæg ved hjælp af et formaliseret sprog at udarbejde et program, som når det udføres på et edb-anlæg, opfylder de krav, som er specificeret i oplægget.

Denne definition på programmering er bred. Ganske vist præciserer den klart, at en programmering skal ende op med et program, som kan udføres på et edb-anlæg, og som opfylder de ønskede krav. Men udgangspunktet for en programmering er et givet oplæg, som kan være alt lige fra et ønske om et kildeskattesystem til et detaljeret rutediagram over en algoritme til beregning af gennemsnittet af en række heltal. Medmindre det givne oplæg er fuldt specificeret indebærer programmeringen beslutninger, som er afgørende for hvilket system, der afgrænses, og hvilke ændringer edb-anvendelsen medfører. Disse beslutninger kommer til udtryk i valget af inddata, algoritmer og uddata.

I programmeringsundervisningen skal dette forhold formidles.

### 3.2 De teknisk betonedede programmeringsfærdigheder

Programmeringsundervisningen skal selvfølgelig lægge vægt på indlæringen af de teknisk betonedede programmeringsfærdigheder: Strukturering af data og algoritmer, repræsentationen af disse i et formaliseret sprog, udviklingen af programmet på et edb-anlæg m.m. Der findes megen litteratur, som omhandler problemerne i forbindelse med indlæring af disse færdigheder.

Hvad er centralt i forhold til denne indlæring? Er det indlæringen af det formaliserede sprog, er det forståelsen for datamaskinens funktioner, er det indlæringen af måder at tænke på, eller er det arbejdsmetoder, som er det centrale?

Wirth siger herom:

My primary motivation is to introduce programming as the art or technique of constructing and formulating algorithms in a systematic manner, recognizing that it is a discipline in its own right. Algorithms, in the most general sense, are recipes for classes of data processing and control processes. They must be conceived as solid structures consisting of logically, reliably, and suitably designed building blocks.

The student should be educated to proceed methodically and systematically in the design of algorithms through the demonstration of problems and techniques that are typical of programming but independent of the area of application. For this reason, no specific area of application is emphasized as an end in itself; exercises and examples have been selected as demonstrations of generally valid problems and methods of solution. In the same spirit, the notation or programming language is deemphasized; the language is our tool but not an end in itself. The primary aim of programming courses should not be to teach perfection in the knowledge and use of all features and idiosyncrasies of a specific language. Rather, the language should mirror the fundamental and most important concepts of algorithms in an obvious, natural, and easily understandable way. It should also take into account the inherent properties and limitations of digital computers. (Wirth73 side xi).

Gries og Conway siger om det samme spørgsmål:

A program is a description of a problem solution that is written in a language and form intelligible to a computer. If someone approaching the programming process for the first time already has a systematic, well-disciplined approach to problem analysis in general, then learning to program a computer just means learning the nature and syntax of a programming language. (Conway73, preface).

Og endelig mener Dijkstra:

The purpose of this little monograph is to assist the programming reader in cleaning up his own thinking, to transmit to him some mental disciplines by sticking to which he can avoid making his job unnecessarily difficult. It is born out of dissatisfaction with the usual kind of programming course, which now strikes me as like the type of driving lessons in which one is taught how to handle a car instead of how to use a car to reach one's intended destination. This monograph is intended as a complement to such courses .....

For those readers who identify the programmer's competence with a thorough knowledge of the idiosyncrasies of one or more of the baroque tools into which modern programming languages and systems have degenerated, the book will also be very incomplete, because I won't describe any programming language –not even the one I use– to any degree of detail. I shall use some sort of programming language, as "a communication language" say, not for the communication of algorithms, but for the communication of ways of thinking, as a vehicle for programming style. (Dijkstra71 side 2).

Ved indlæringen af de teknisk betonedede programmeringsfærdigheder ser vi, at der blandt de citerede forfattere er enighed om, at man bør lægge vægt på indlæring af en fornuftig arbejdsmetode, som tildels er uafhængig af det formaliserede sprog, som anvendes. Indlæringen af et eller flere formaliserede sprog må nærmest betragtes som en nødvendighed. Samtidig ser vi at vore tidligere overvejelser om, at det er vigtigt at formidle en forståelse for den sammenhæng, hvori værktøjet kommer til at indgå, modarbejdes af Wirth. Han betragter programmering som "a discipline in its one right" og understreger, at de opgaver, som skal indgå i undervisningen, skal være "independent of the area of application." <sup>11</sup> Med vores udgangspunkt må det netop være vigtigt gennem opgavevalget at inddrage sammenhængen.

Sammenfattende kan vi sige, at det i indlæringen af de teknisk betonedede programmeringsfærdigheder er

væsentligt at formidle en fornuftig arbejdsform, og gennem et omhyggeligt opgavevalg at formidle en forståelse for den sammenhæng, hvori programmeringsværktøjet vil komme til at indgå. Og det er

nødvendigt at lære eleverne at beherske et eller flere formaliserede sprog, som kan forstås af en datamaskine. Samtidig er det nødvendigt at lære eleverne at beherske datamaskinen i det omfang som de(t) formaliserede sprog kræver, og at give dem en forståelse for, hvilket grundlag der er for de forskellige sprogs design, og hvordan dette grundlag påvirker løsningernes endelige udseende.

### 3. 3 Formalisering contra arbejdsform

I en programmeringsundervisning er det altså på en gang væsentligt at bibringe eleverne en fornuftig arbejdsform, og nødvendigt at lære eleverne at beherske et formaliseret sprog, som kan forstås af datamaskinen.

I den traditionelle form for programmeringsundervisning undervises der fra starten i et eller andet programmeringssprog. Målet med undervisningen er at lære eleverne alle dette sprogs særheder og specielle egenskaber. Måske vil der henimod slutningen af undervisningen blive inddraget nogle overvejelser om metodik, dokumentation etc. Indlæringen af arbejdsformen foregår gennem løsningen af forskellige opgaver, og overlades stort set til eleven selv.

Vil man lære eleverne en fornuftig arbejdsform, må indlæringen af denne arbejdsform inddrages som en langt væsentligere del af undervisningen. Tilmed må man i undervisningen starte med indlæringen af arbejdsformen og vente med indlæringen af selve programmeringssproget. Startes undervisningen med indlæringen af et programmeringssprog vil det næsten automatisk blive dette sprogs særheder og finesser, kørslen af små programmer i dette sprog og de tekniske problemer det indebærer, der vil komme til at optage elevens opmærksomhed. Og selv om man i undervisningen lægger vægt på indlæring af en fornuftig arbejdsform, vil det være uhyre vanskeligt at motivere eleverne for den. Sagt med andre ord vil indlæringen af et programmeringssprog let virke blokerende for indlæringen af en fornuftig arbejdsform. Denne påstand underbygges af de erfaringer man har gjort med praktisk programmeringsundervisning. Cashman og Mein siger herom

Problem-solving ability is fundamental and essential to the succes of any computer scientist. Many computer science curriculums, while giving emphasis to the teaching of programming languages, compiler writing and numerical methods, do not directly attempt to produce enhanced problem-solving proficiency. Such an omission is quite serious since successful application of the computer in scientific, business and systems applications requires a resonably high level of problem-solving ability. (Cashman<sup>75</sup> side 43).

Og de går så vidt som til at foreslå, at en forudsætning for at kunne følge det introducerende programmeringskursus, skal være at man har fulgt et kursus i problemløsning.

Selvom det er det fornuftigste at vente med indlæringen af selve pro-

grammeringssproget, hvis man ønsker at bibringe eleverne en fornuftig arbejdsform, er der dog en række problemer forbundet med denne fremgangsmåde. For det første er det vanskeligt at holde eleverne væk fra datamaskinen. Den motivation, der for eleverne ligger i at se deres eget program blive udført på datamaskinen, bliver da også ofte brugt som argument for, at man bliver nødt til at starte programmeringsundervisningen med indlæringen af et programmeringssprog ("Det er umuligt at holde eleverne væk fra maskinen"). Der findes imidlertid andre løsninger. En af disse er at lade eleverne benytte programmer fra programbiblioteket til at løse forskellige problemer (anvendelsen af et færdigt program er også en form for programmering. Her er edb-anlægget blot udvidet med endnu et programmel-lag). For det andet vil kursusrammerne ofte sætte en grænse for, hvor lang tid man kan lade gå, inden indlæringen af selve programmeringssproget må starte. Det er nok i de færreste tilfælde, at der er tid til at holde et selvstændigt kursus i problemløsning, arbejdsform eller hvad man ønsker at kalde det. For det tredie må det overvejes, hvad der er en fornuftig arbejdsform. Hvad er problemløsning? Hvilken type af problemer drejer det sig om? Disse begreber – fornuftig arbejdsform, problemløsning, problemtyper – er begreber, der til dels er knyttet til den sammenhæng, som eleven senere skal anvende programmeringsværktøjet i. Den fysiker, som gerne vil anvende datamaskinen til at foretage besværlige udregninger, har ikke brug for de samme arbejdsformer som den edb-arbejder, der skal indgå i et team, som får til opgave at deltage i udarbejdelsen af et edb-system til brug i de danske folkebiblioteker. De problemer, som edb-arbejderen vil møde, og de som fysikeren vil stå overfor, er ikke af samme type. Begreberne problemløsning og fornuftig arbejdsform har dog også et væsentligt indhold, som er uafhængigt af sammenhængen. Dette indhold er karakteriseret ved forskellige teknikker: Stepwise refinement, top-down, bottom-up og struktureret programmering.

Når det er fastlagt, hvilken arbejdsform, der i et givet tilfælde er tale om, må det overvejes, hvordan denne arbejdsform kan indlæres. Jeg har selv givet et forslag til, hvorledes man kan anvende et simpelt algoritmebeskrivelsesværktøj til at lære gymnasieelever problemløsning (Migselv76). Dette beskrivelsesværktøj forsøger via en delvis formalisering at for-

hindre, at eleverne arbejder ustruktureret og uden omtanke. Samtidig tillader beskrivelsesværktøjet, at beskrivelserne indeholder næsten udelukkende almindelige danske formuleringer. Tanken er at dette beskrivelsesværktøj på den ene side skal støtte indlæringen af en struktureret arbejdsform (som for gymnasieelevers vedkommende vil kunne anvendes ved stilskrivning, rapportskrivning, strukturering af et givet emneområde etc.) og på den anden side skal illustrere vanskelighederne ved med et kun delvist formaliseret beskrivelsesværktøj at lave præcise, entydige beskrivelser (hvilket datamaskinen jo i sidste instans kræver). Denne ide vil blive taget op i afsnit III, idet jeg dog her vil anvende et andet delvist formaliseret beskrivelsesværktøj, som er mere velegnet i en programmeringsundervisningssammenhæng.

### 3.4 Grænseområder

Som en sidste del af de teknologiske overvejelser vil jeg se på, hvilke emneområder indenfor datalogien, som bør indgå i overvejelserne omkring undervisning i programmering. Jeg vil her indskrænke mig til at trække de emneområder frem, som er centralt placeret i forhold til problemerne i forbindelse med en programmeringsundervisning.

Programmeringssprog. Dette emneområde er klart centralt, idet enhver programmeringsundervisning vil omfatte indlæringen af et eller flere programmeringssprog.

Der udvikles på universiteterne en del programmeringssprog, som bygger på nye og anderledes ideer. Men disse sprog bliver ikke benyttet i praktiske edb-anvendelser udenfor universiteterne. Der er flere grunde hertil. En af grundene er universiteternes manglende evne til at forstå de problemer, som findes i forbindelse med praktiske edb-anvendelser. Lad mig nævne et eksempel.

PASCAL, som er udviklet af Wirth (Wirth70), anses i universitetskredse for at være velegnet til at understøtte struktureret programmering, og som sådant for at være velegnet til at indgå i en programme-

ringsundervisning. Men dette sprog benyttes stort set ikke udenfor universiteterne. En af grundene finder vi i PASCAL's mangel på faciliteter til ind- og udlæsning af data fra såvel ydre enheder som fra forskellige former for hjælpelagre. Som Wirth selv udtrykker det, "programming is a discipline in its own right", og derfor kan man betragte problemer, som er "independent of the area of application". Det betyder, at der under konstruktionen af PASCAL ikke er blevet lagt tilstrækkelig vægt på, at sproget skal kunne klare den type af problemer, som løses ved hjælp af edb udenfor universiteterne. Her drejer det sig om at håndtere kolossale datamængder. Og problemet ligger ikke i selve beregningerne, men i udvælgelsen, omfanget og kompleksiteten af de data, som skal behandles.

Samtidig skal det siges, at de programmeringssprog, som anvendes udenfor universiteterne (COBOL, FORTRAN og forskellige assembler-sprog) ikke er særlig velegnede til at understøtte en fornuftig arbejdsform, og derfor ikke egner sig til at indgå i en programmeringsundervisning.

Systemarbejde omhandler problemerne i forbindelse med udviklingen og indførelsen af edb-baserede systemer. Disse edb-baserede systemer spænder fra informationssystemer og planlægnings- og styringssystemer i private og offentlige virksomheder over systemer til undervisningsbrug i uddannelsessystemet og videre til operativsystemer og oversættere. Det er centralt indenfor dette emneområde at se på hvilke arbejdsformer og hvilke værktøjer, der i en given sammenhæng vil være velegnede.

Programmeringsmetodik. Indenfor dette emneområde forsøger man at udvikle arbejdsformer, som er velegnede under konstruktionen af programmer. Det adskiller sig fra systemarbejde ved ikke i så høj grad at tage udgangspunkt i de sammenhænge, hvori programmeringsværktøjet indgår. Nøgleord indenfor dette emneområde er: problemløsning, struktureret programmering, dokumentation, aftestning. Der findes kolossale mængder af litteratur, der omhandler problemstillinger, som kan siges at falde ind under dette emneområde.



Programkorrekthed og formel semantik. Disse to emneområder er tildels overlappende og bygger begge på et formelt, teoretisk grundlag. De beskæftiger sig med sammenhængen mellem forskellige former for total formalisering. Programkorrekthed drejer sig om udviklingen af metoder (gerne automatiserbare), som kan bruges til at vise egenskaber ved programmer – for eksempel at et givet program, hvis det udføres, virkelig gør det, man forventer det skal gøre. Formel semantik drejer sig om udviklingen af formelle beskrivelsesværktøjer til beskrivelse af semantikken af et programmeringssprog. Lad mig give et par eksempler, der viser, hvorledes disse to emneområder har direkte indflydelse på overvejelserne i forbindelse med tilrettelæggelsen og gennemførelsen af en programmeringsundervisning.

Under udviklingen af den aksiomatiske metode (Hoare69) indførte Hoare begrebet invarians. Begrebet blev benyttet ved opstillingen af aksiomet for en løkke. Dette begreb har ført til en ny måde at forstå en løkke på, som vil være relevant at benytte i en programmeringsundervisning ved forklaringen af, hvad en løkke er, og hvorledes en løkke konstrueres. I stedet for at forstå en løkke i termer af det den forandrer, forstår man den i termer af det den ikke forandrer, det den holder invariant.

I de senere år er man indenfor disse emneområder begyndt at interessere sig for problemerne i forbindelse med korrektheden af de konkrete repræsentationer af de indgående data, som findes i et program. Dette har for det første ført frem til nye værktøjer til beskrivelse af de indgående data (Hoare72, Hoare73, Liskov74). Disse beskrivelsesværktøjer tvinger programmøren til at tænke på de anvendte datatyper i abstrakte termer i stedet for i termer af deres konkrete repræsentation.<sup>12</sup> De abstrakte datatyper beskrives ved hjælp af de operationer, som definerer dem. Efter at have givet en konkret repræsentation af disse operationer, har programmøren kun adgang til et element af en given datatype via de definerende operationer. Samtidig har dette arbejde bidraget til en større forståelse af, at det under konstruktionen af et program fra starten er mindst lige så vigtigt at gøre sig grundige overvejelser over strukturen af de indgående data, som over strukturen af den indgående algoritme.

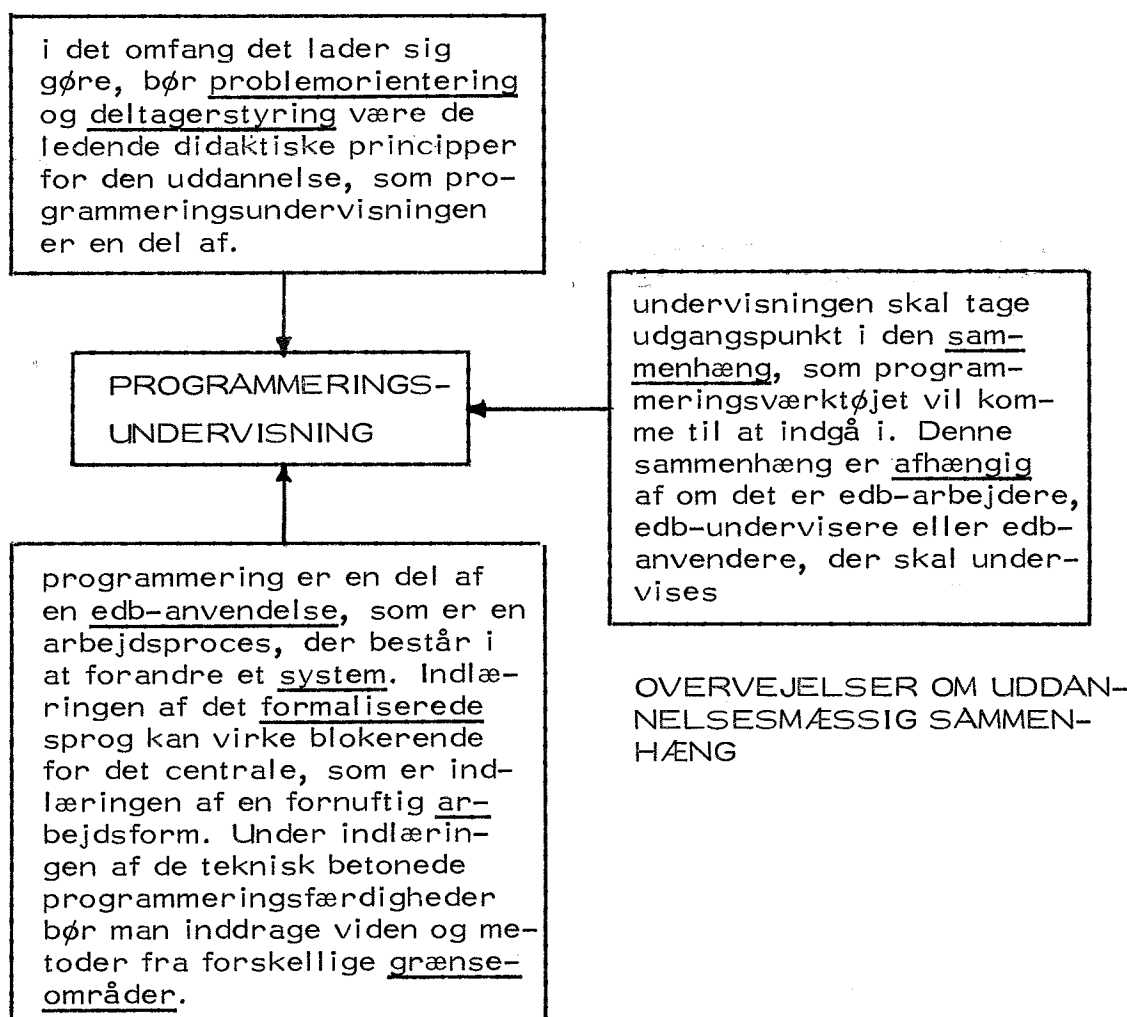
Datamaskinen som værktøj dækker over flere forskellige emneområder indenfor datalogien, som er af betydning for problemerne i forbindelse med undervisning i programmering. Datamaskinen som værktøj skal i denne sammenhæng ses ud fra to synsvinkler. For det første skal der i forbindelse med tilrettelæggelsen af en programmeringsundervisning træf-

fes en række beslutninger om, hvorledes datamaskinen skal indgå i undervisningen (såfremt der er nogen valgmulighed): Hvilket teknisk udstyr skal anvendes, og hvilken kørselsform skal man benytte. For det andet skal der i selve programmeringsundervisningen undervises i datamaskinen som værktøj. I det mindste må undervisningen indeholde de aspekter, som er af betydning for at kunne anvende det eller de indgående programmeringssprog.

#### 4. Sammenfatning

De synspunkter, som jeg har valgt som udgangspunkt for tilrettelæggelsen og gennemførelsen af en programmeringsundervisning, er sammenfattet i nedenstående figur. Figuren svarer fuldstændig til figuren på side 3, idet spørgsmålene blot er erstattet med de tilhørende svar.

#### DIDAKTISKE OVERVEJELSER



#### TEKNOLOGISKE OVERVEJELSER

Disse tre synsvinkler, som danner udgangspunkt for overvejelserne omkring problemerne i forbindelse med undervisning i programmering, skal ikke betragtes isoleret. Lad mig illustrere dette.

I en programmeringsundervisning er det væsentligt at lægge vægt på indlæring af en fornuftig arbejdsform. Denne arbejdsform omfatter såvel de teknisk betonedede programmeringsfærdigheder som det at kunne anvende programmeringsværktøjet i en given sammenhæng. Denne sidste del af arbejdsformen er selvfølgelig afhængig af sammenhængen. Men et fælles krav, som går på arbejdsformen – uafhængig af sammenhængen – er kravet om kreative kvalifikationer. Indlæring af kreative kvalifikationer forudsætter et overvejende akkomodativt indlæringsforløb. Dette fører frem til kravet om problemorientering og deltagerstyring, som igen falder tilbage på selve tilrettelæggelsen og gennemførelsen af programmeringsundervisningen.

De krav, som fremkommer under én synsvinkel, skal forstås i sammenhæng med krav fra de to andre synsvinkler. Opfyldelsen af ét krav fra én synsvinkel forudsætter opfyldelsen af et andet krav fra en anden synsvinkel. Det er netop kombinationen af de tre synsvinkler, som tilsammen danner grundlaget for tilrettelæggelsen og gennemførelsen af en programmeringsundervisning.

### III. SYSTEMBESKRIVELSE SOM INDFALDSVINKEL

I afsnit II har jeg diskuteret hvilke aspekter, der er væsentlige ved rettelse og gennemførelsen af en programmeringsundervisning. I dette afsnit vil jeg vende mig mod et forslag til, hvordan de krav, som udsprang af denne diskussion, kan realiseres. Et vigtigt kriterium ved vurderingen af dette afsnit, må derfor være praktisk anvendelighed.

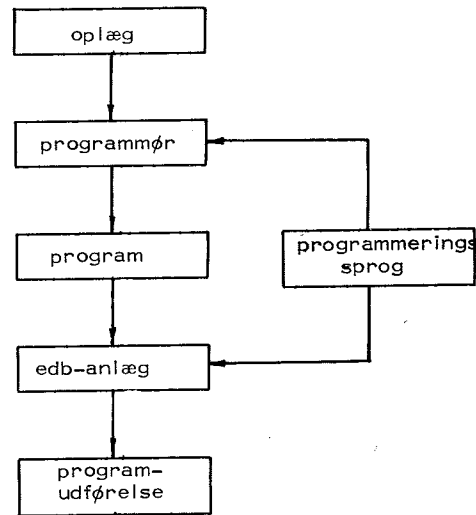
Afsnittet indeholder først en kort beskrivelse af, hvad vi vil forstå ved systembeskrivelse. Denne måde at opfatte systembeskrivelse på bygger på de ideer, metoder og værktøjer, som er udviklet i forbindelse med DELTA-projektet <sup>13</sup>. Dernæst indeholder afsnittet en argumentation for, at denne indfaldsvinkel til en programmeringsundervisning kan hjælpe til opfyldelsen af de krav, som blev opstillet i afsnit II. Og endelig findes der sidst i afsnittet en skitse af et undervisningsforløb, hvori systembeskrivelse anvendes som indfaldsvinkel til programmeringsundervisningen.

#### 1. Kort om systembeskrivelse

##### 1.1 Generelt

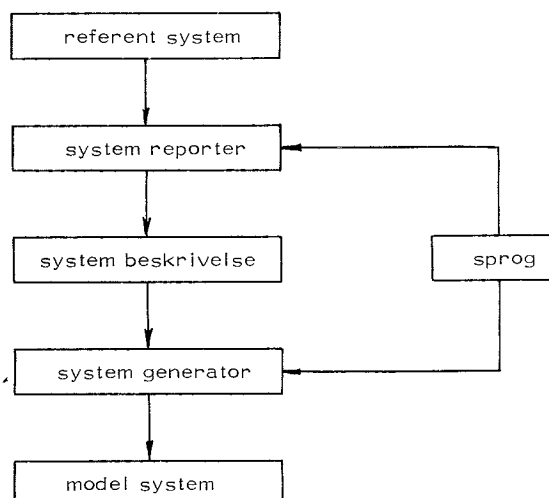
Vi kan opfatte et program som et eksempel på en systembeskrivelse.

Ifølge vores overvejelser er programmering den del af en edb-anvendelse, som består i ud fra et givet oplæg ved hjælp af et formaliseret sprog (programmeringssprog), at udarbejde et program, som når det udføres på et edb-anlæg, opfylder de krav, som er specificeret i oplægget. Dette forhold kan vi illustrere ved hjælp af nedenstående figur.



Vi kan opfatte programmeringen som en kommunikationsproces, hvori programmøren ved hjælp af et formaliseret sprog kommunikerer oplægget (problemløsningen) til edb-anlægget. Kommunikationen foregår på grundlag af en beskrivelse af dette oplæg, nemlig programmet. Edb-anlægget læser denne beskrivelse af oplægget og udfører den. Programudførelsen udgør således en model af dette oplæg.

Denne kommunikationsproces kan beskrives i mere generelle termer. En system reporter kommunikerer information om et givet referent system ved i et eller andet sprog at udfærdige en system beskrivelse af referent systemet. System generatoren modtager system beskrivelsen og generer på grundlag heraf et model system.



Denne kommunikationsproces omfatter betydeligt mere end blot kommunikationen mellem programmør og edb-anlæg. Den gruppe studenter, som på et kursus bliver sat til at skrive en rapport om "En programmørs hverdag på Jydsk Telefon A/S", optræder som system reportere. Referent systemet udgøres af programmørens hverdag: Jydsk Telefons bygninger, de ansatte, edb-anlægget, de opgaver der arbejdes med m.m. og relationerne mellem disse. Rapporten er en beskrivelse af dette system. Og de øvrige deltagere på kurset er system generatoren, i det øjeblik de læser rapporten. Samtidig med at de læser denne, dannes der i deres hjerner en model af det oprindelige system. Det sprog, som i dette tilfælde benyttes ved kommunikationen, er almindelig dansk udvidet med en række specielle fagudtryk (interaktiv, debugging etc.)

Jeg har tidligere præciseret, hvad vi vil forstå ved et system (s. 21). Det er klart, at den model, som genereres af system generatoren, også vil kunne betragtes som et system. Vi vil skelne mellem to former for systemer – manifeste og mentale systemer. Begge former for systemer er altid en del af den fysisk eksisterende verden. Mentale systemer eksisterer som en del af et menneskes bevidsthed, fysisk materialiseret som tilstanden af cellerne i menneskets hjerne. Systemer, som er ydre i forhold til menneskets bevidsthed, kaldes manifeste systemer. De forestillinger, som programmøren gør sig om udførelsen af det program, som skal udarbejdes på grundlag af et givet oplæg, udgør et eksempel på et mentalt system.

Hvad opnår vi ved at lave en system beskrivelse? For det første skaber system reporteren herved et grundlag for kommunikation omkring referent systemet. I den forbindelse er det vigtigt at gøre sig klart, at afhængig af hvem system reporteren er, og hvem der skal indgå i kommunikationen omkring referent systemet, vil der blive tale om forskellige systembeskrivelser. Lad os tænke os at der som et led i en edb-anvendelse på en virksomhed skal konstrueres en systembeskrivelse, der skal tjene som kommunikationsgrundlag i forbindelse med udviklingen og indførelsen af edb-systemet. Virksomhedsledelsen vil ønske at denne systembeskrivelse giver forbedrede muligheder for at vurdere de økonomiske konsekvenser. Arbejderne i produktionen vil derimod ønske at systembeskrivelsen skal klarlægge, hvorledes edb-anvendelsen vil ændre deres arbejdssituation.

Og andre grupper vil have andre behov. System reporterens grundlæggende antagelser om samfundet, virksomheden, de ansatte og relationerne mellem disse vil imidlertid sætte grænser for, i hvor høj grad forskellige grupper behov for systembeskrivelse vil kunne blive tilgodeset.

For det andet opnår system reporteren selv gennem konstruktionen af system beskrivelsen en bedre forståelse af referent systemet.

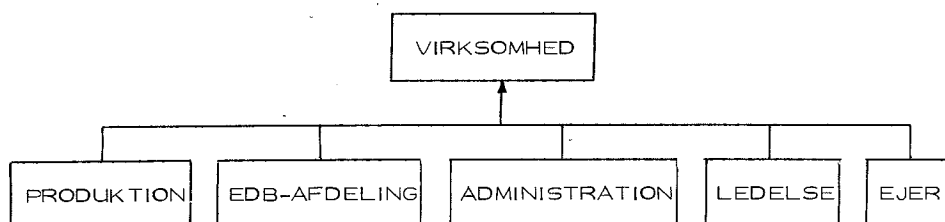
Et generelt krav til en systembeskrivelse, der skal danne grundlag for en kommunikation mellem personer, er, at systembeskrivelsen skal give de indgående personer øgede muligheder for at handle konstruktivt i forhold til referent systemet og dets omgivelser. Hvorvidt dette krav opfyldes af en given systembeskrivelse, vil afhænge af den sammenhæng, som systembeskrivelsen skal indgå i <sup>14</sup>.

DELTA-projektet omfatter blandt andet udviklingen af et delvist formaliseret sprog, som skal kunne anvendes ved beskrivelse af systemer i forbindelse med edb-anvendelser. I det følgende vil jeg give et eksempel på anvendelsen af et DELTA-lignende sprog til beskrivelse af et system. I dette eksempel vil jeg selv optræde som system reporter og læseren vil, i det øjeblik eksemplet gennemlæses, optræde som system generator.

### 1.2 Et eksempel på systembeskrivelse

Referent systemet udgøres af en privat produktionsvirksomhed. I beskrivelsen ønsker jeg at vise, hvorledes forskellige grupper på virksomheden – specielt edb-afdelingen – indgår i forhold til hinanden. Beskrivelsen er skitse-mæssig og afspejler min personlige holdning til referent systemet.

Jeg vælger at opdele virksomheden i 5 komponenter, som vist i nedenstående figur.



Denne figur kan jeg udtrykke i en overordnet, meget simpel systembeskrivelse.

#### VIRKSOMHED SYSTEM

BEGYND VIRKSOMHEDen består af en EJER, en LEDELSE, en ADMINISTRATION, en EDB-AFDELING og en PRODUKTION

#### SLUT VIRKSOMHED SYSTEM

I beskrivelsen er der anvendt en blanding af almindelig dansk og en række formaliserede sprogelementer. De formaliserede sprogelementer er skrevet med stort og understreget. Desuden er de indgående komponenter skrevet med stort. Disse komponenter og deres indbyrdes relationer kan beskrives yderligere ved angivelse af et handlingsmønster og en datastruktur for hver af komponenterne. Herved fremkommer en mere detaljeret beskrivelse af referent systemet.

I denne sammenhæng har jeg valgt at give følgende beskrivelse, idet jeg ønsker at lægge vægt på edb-afdelingens placering i forhold til de øvrige komponenter:



VIRKSOMHED SYSTEM

BEGYND VIRKSOMHEDen består af  
en EJER KOMPONENT

BEGYND

DATASTRUKTUR " EJERens navn er Jens Hansen  
Størrelsen af hans samlede kapital er ikke kendt,  
men han anses for at være meget rig. Han er medlem  
af det lokale bystyre. Og af uddannelse er han  
cand. merc.

HANDLINGSMØNSTER Han køber arbejdskraft ved hjælp  
af sin kapital og udbytter de arbejdere, hvis arbejds-  
kraft han køber. Han opstiller de langsigtede mål for  
VIRKSOMHEDen og har den endelige afgørelse i alle  
sager, der vedrører denne. To gange om ugen mødes  
han med LEDELSEn for at holde sig underrettet om  
de nyeste udviklinger på VIRKSOMHEDen og for at  
videregive sine direktiver. Det meste af sin energi  
anvender han på lokalpolitik.

SLUT EJER KOMPONENT

en LEDELSE KOMPONENT

BEGYND

DATASTRUKTUR LEDELSEn består af 3 personer.  
En cand. merc. og to teknikum ingeniører. LEDELSEns  
gennemsnitsløn er 150.000 kr. om året. LEDELSEn  
forsøger i enhver situation at handle i overensstemmelse  
med EJERens interesser.

HANDLINGSMØNSTER LEDELSEn mødes med EJERen to  
gange om ugen. På disse møder aflægger LEDELSEn  
rapport om udviklingen på VIRKSOMHEDen og den mod-  
tager direktiver fra EJERen. Til daglig udøver LEDEL-  
SEn EJERens aftalte ret til at lede og fordele ar-  
bejdet. På baggrund af de overordnede mål, som EJERen  
opstiller, forsøger LEDELSEn via en planlægning at  
styre PRODUKTIONEN med henblik på opfyldelse af  
målene. Hertil anvender LEDELSEn ADMINISTRATIO-  
Nen og EDB-AFDELINGen som værktøj.

SLUT LEDELSE KOMPONENT

## en ADMINISTRATION KOMPONENT

### BEGYND

DATASTRUKTUR I ADMINISTRATIONen er der ansat 14, 5 mænd og 9 kvinder. Gennemsnitsindtægten er 65.000 kr om året.

HANDLINGSMØNSTER ADMINISTRATIONen udfører løbende en række opgaver : lønregnskab og lønudbetaling, bogholderi, lagerregnskab, korrespondance osv. Derudover løser ADMINISTRATIONen andre mere specielle opgaver, som LEDELSEN stiller. I løsningen af nogle af disse opgaver samarbejder ADMINISTRATIONen med EDB-AFDELINGen.

### SLUT ADMINISTRATION KOMPONENT

## en EDB-AFDELING KOMPONENT

### BEGYND

DATASTRUKTUR I EDB-AFDELINGen er der ansat 8, 6 mænd og 2 kvinder. Gennemsnitsindtægten er 80.000 kr. om året. EDB-AFDELINGen råder over et middelstort edb-anlæg. EDB-AFDELINGen er fysisk isoleret fra PRODUKTIONen, men har under nogle edb-anvendelser behov for at være i kontakt med PRODUKTIONen. Derimod deler EDB-AFDELINGen bygning med ADMINISTRATIONen, hvilket letter kommunikationen med denne under de edb-anvendelser, som involverer ADMINISTRATIONen. EDB-AFDELINGen arbejder i perioder under hårdt tidspres. Det daglige arbejde ledes af afdelingslederen.

HANDLINGSMØNSTER EDB-AFDELINGen udfører opgaver, som formuleres af LEDELSEN. Under en konkret edb-anvendelse inddrager EDB-AFDELINGen PRODUKTIONen, ADMINISTRATIONen og LEDELSEN i det omfang det er nødvendigt for gennemførelsen af edb-anvendelsen. Hvis en gruppe på VIRKSOMHEDen har specielle ønsker i forbindelse med en konkret edb-anvendelse, skal EDB-AFDELINGen fremlægge disse synspunkter for LEDELSEN, og derefter handle i overensstemmelse med LEDELSENS afgørelse. Det daglige arbejde består i samtaler med forskellige grupper på VIRKSOMHEDen, konstruktion af oplæg til programmering, programmering og indkøring og drift af færdige programmer.

### SLUT EDB-AFDELING KOMPONENT

og en PRODUKTION KOMPONENT

BEGYND

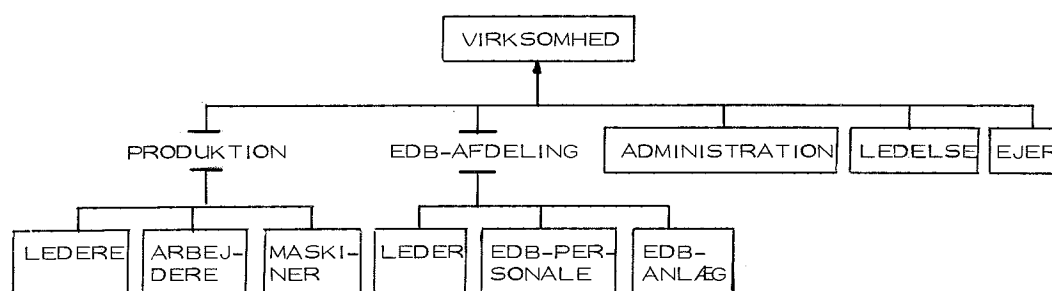
DATASTRUKTUR I PRODUKTIONen er der ansat mellem 200 og 300 arbejdere. Ca. halvdelen af disse er kvinder. Desuden er der ansat 5 værkførere og 1 planlægger. Gennemsnitsindtægten er 48.000 kr. om året. I PRODUKTIONen anvendes en række af avancerede maskiner. Hovedparten af de ansatte i PRODUKTIONen anser værkførerne, planlæggeren, ADMINISTRATIONen og EDB-AFDELINGen for at stå på LEDELSEns og EJERens side. De ansatte i PRODUKTIONen har vanskeligt ved at fremføre og få gennemført deres krav omkring konkrete edb-anvendelser. Kommunikationen med EDB-AFDELINGen er vanskelig. De ansatte i PRODUKTIONen er på en række punkter utilfredse med de forhold de arbejder under.

HANDLINGSMØNSTER PRODUKTIONen arbejder på grundlag af ordrer fra LEDELSEn. De fleste af disse ordrer kommer fra EDB-AFDELINGen, idet produktionsstyringssystemet automatisk genererer disse. PRODUKTIONen tilbagemelder til EDB-AFDELINGen om tilstanden i PRODUKTIONen. En del af de ansatte arbejder aktivt indenfor fagforeningen.

SLUT PRODUKTION KOMPONENT

SLUT VIRKSOMHED SYSTEM

Denne beskrivelse udtrykker – på et vist abstraktionsniveau – hvorledes jeg vælger at betragte virksomheden i denne sammenhæng. Jeg kunne opnå en mere detaljeret beskrivelse ved at foretage en yderligere opdeling af de indgående komponenter. Jeg kunne for eksempel vælge at opdele på følgende måde <sup>15</sup>.



Og derefter kunne jeg give en beskrivelse af hver af disse komponenters handlings- og tilstandsmønstre. Alt i alt fremtræder der således forskellige totalbeskrivelser af det samme referent system (foruden de forskelle, som vil fremkomme ved nyt valg af system reporter og ved ændring af de grundlæggende interesser), idet de adskiller sig ved det abstraktionsniveau, de beskriver systemet på.

## 2. Argumentation for systembeskrivelse som indfaldsvinkel

Jeg vil nu gå ind på en argumentation for, at systembeskrivelse, som det blev skitseret i foregående afsnit, med fordel kan anvendes som indfaldsvinkel til en programmeringsundervisning. De krav, som fremkom under diskussionen af udgangspunktet for tilrettelæggelsen og gennemførelsen af en programmeringsundervisning, vil i det følgende blive taget op. Jeg vil vise, hvorledes vi med systembeskrivelse som indfaldsvinkel vil kunne opfylde disse krav – for så vidt de ydre rammer for undervisningsforløbet gør det muligt. Desuden vil jeg se på, hvilke yderligere fordele vi kan opnå med denne fremgangsmåde.

### 2.1 Kravet om problemorientering og deltagerstyring

Mulighederne for at lade de didaktiske principper om problemorientering og deltagerstyring danne grundlag for tilrettelæggelsen og gennemførelsen af et uddannelsesforløb er stærkt afhængige af de ydre rammer for det pågældende uddannelsesforløb. Uddannelsens organisatoriske placering, de ressourcer, som er tildelt uddannelsen, kontrolbestemmelser – d. v. s. krav om extern evaluering, forløbsbestemmelser, der vedrører uddannelsesforløbets praktiske organisation og gennemførelse (for eksempel timeplaner, pensum og karaktergivning), og forskellige formålsformuleringer vil ofte virke begrænsende for problemorienteringen og deltagerstyringen.

En detaljeret diskussion af disse ydre rammer og af problemerne i forbindelse med tilrettelæggelsen og gennemførelsen af problemorienterede, deltagerstyrede undervisningsforløb i det hele taget findes i Illeris<sup>74</sup>. Jeg vil her indskrænke mig til at se på, hvorledes vi med systembeskrivelse som indfaldsvinkel, kan lade programmeringsundervisningen indgå i et problemorienteret, deltagerstyret uddannelsesforløb.

I den forbindelse er det vigtigt at gøre sig klart, at det ikke drejer sig om at lave et 14 dages kursus i FORTRAN-programmering problemorienteret og deltagerstyret. Det vil ikke være muligt, idet princippet om problemorientering netop bryder med den traditionelle fagopdeling, kræver tværfaglighed. Det, det drejer sig om, er at indplacere programmeringsundervisningen på en hensigtsmæssig måde i et problemorienteret, deltagerstyret uddannelsesforløb, og at udnytte de fordele dette indebærer i programmeringsundervisningen. I små, isolerede kurser, som ude-

lukkende har til formål at lære eleverne at programmere, vil betingelserne ikke på samme måde være tilstede for at anvende de didaktiske principper om problemorientering og deltagerstyring, og for på en meningsfuld måde at tage udgangspunkt i den sammenhæng, som programmeringsværktøjet vil komme til at indgå i.

Vi forestiller os altså et uddannelsesforløb, hvor der enten på forhånd i formålsformuleringen er specificeret et krav om, at eleverne blandt andet skal lære at programmere, eller hvor der under afviklingen af forløbet opstår et ønske blandt deltagerne om at lære at programmere. I en sådan situation vil det være naturligt, som en del af det problemorienterede uddannelsesforløb, at arrangere et særligt delforløb<sup>16</sup>, hvori eleverne lærer de teknisk betonedede programmeringsfærdigheder. Ved at anvende systembeskrivelse som indfaldsvinkel til dette særlige delforløb, vil det være muligt at tage udgangspunkt i problemerne i det egentlige forløb og i deltagerens forventninger med hensyn til delforløbet. Strukturen i dette delforløb kan være planlagt på forhånd, men opgavematerialet og eksempelmaterialet vil kunne konstrueres på stedet ud fra deltagerens ønsker og behov. Alt, som vi ønsker at betragte som et system, er nemlig et system. Så ligegyldigt om deltagerne er fordybet i "kemiens indflydelse på storindustrien" eller "edb som værktøj til styring af produktionsvirksomheder" vil det være muligt på stedet at fremtrække systemer, som tager udgangspunkt i disse problemkredse, som er egnede som opgave- og eksempelmateriale, og som samtidig vil kunne illustrere sammenhængen mellem den problemverden deltagerne befinder sig i og mulighederne for edb-anvendelse. Jeg vil senere mere detaljeret vise, hvorledes et sådant delforløb kan struktureres, og hvorledes samspillet med det egentlige forløb kan etableres.

Ved tilrettelæggelsen af deltagerstyrede forløb er det centralt at holde mulighederne åbne for, at deltagerens mål kan indgå som væsentlige styrende faktorer i det faktiske forløb. Ved at anvende systembeskrivelse som indfaldsvinkel til programmeringsundervisningen er dette muligt. I den traditionelle programmeringsundervisning, hvor det snarere er algoritmebegrebet og databegrebet end systembegrebet, der er det centrale, vil der ikke være de samme muligheder. Algoritmebegrebet og databegrebet er langt snævrere end systembegrebet – vi kan ikke vælge at betragte

alt som en algoritme – og disse begreber giver derfor ikke de samme muligheder for at inddrage deltagernes problemverden, som den former sig i selve undervisningsforløbet. Derimod vil vi med systembeskrivelse som indfaldsvinkel dels kunne tage udgangspunkt i denne problemverden og dels kunne inddrage algoritmebegrebet og databegrebet på en naturlig måde, idet databegrebet er nært knyttet til en komponents datastruktur, mens algoritmebegrebet er nært knyttet til komponentens handlingsmønstre.

## 2.2 Kravet om sammenhæng

Som det allerede er blevet fremhævet skal de forskellige krav til programmeringsundervisningen ikke betragtes isoleret (s. 32). Derimod er det kombinationen af disse krav, som udgør udgangspunktet for tilrettelæggelsen og gennemførelsen af en programmeringsundervisning. Således bidrager problemorienteringen i væsentlig grad til at det bliver muligt at tage udgangspunkt i den sammenhæng, som programmeringsværktøjet kommer til at indgå i. Netop ved at lade indlæringen af de teknisk betonedede programmeringsfærdigheder indgå som et særligt delforløb i det egentlige problemorienterede forløb, og ved at anvende systembeskrivelse til i delforløbet at tage udgangspunkt i situationen i det egentlige forløb, er der skabt gode muligheder for under indlæringen at lade programmeringsværktøjet indgå i en større og for eleverne meningsfuld sammenhæng.

I eksemplet på systembeskrivelse (s. 38 ff.) har jeg vist, hvorledes ét aspekt af edb-arbejderens senere arbejdssituation kan inddrages som eksempel. Andre eksempler ville kunne lægge op til diskussioner, som er relevante i en undervisning af edb-undervisere eller edb-anvendere.

## 2.3 Kravet om en fornuftig arbejdsform

Ved at anvende systembeskrivelse som indfaldsvinkel til programmeringsundervisningen kan vi således lette problemorienteringen og deltagerstyringen, og vi kan tage udgangspunkt i den sammenhæng, som programmeringsværktøjet kommer til at indgå i. Men kan eleverne samtidig – uden at lade programmeringsindlæringen optage en større del af undervisningsforløbet – lære det som er centralt i forhold til de teknisk betonedede programmeringsfærdigheder: en fornuftig arbejdsform?

Inden vi vender os mod dette spørgsmål skal det pointeres, at en fornuftig arbejdsform dækker over mere end blot en teknisk betonet programmeringsfærdighed. En fornuftig arbejdsform har også et indhold, som er afhængigt af den sammenhæng, programmeringsværktøjet skal indgå i. Edb-arbejderen og edb-anvenderen har behov for forskellige kvalifikationer, som sætter dem i stand til hver for sig at anvende de teknisk betonedede programmeringsfærdigheder i en given sammenhæng. Indlæringen af disse kvalifikationer vil kunne ske på baggrund af samspillet mellem det særlige delforløb og det egentlige problemorienterede forløb.

Den fornuftige, teknisk betonedede arbejdsform, som skal bibringes alle uanset hvilken sammenhæng, programmeringsværktøjet skal indgå i, kan karakteriseres ved hjælp af en række begreber og teknikker. Sammenfattende kan den betegnes som struktureret programmering. Den kan mere detaljeret beskrives ved hjælp af forskellige sæt af begreber/metoder: abstraktion overfor konkretisering, bottom-up overfor top-down, syntese overfor analyse og funktionelt overfor operationelt. Denne fornuftige, teknisk betonedede arbejdsform drejer sig om at håndtere det komplekse – om ved hjælp af strukturering og systematik at gøre det komplekse simple at håndtere. Den kan ikke indlæres ved at lære eleven de begreber og metoder, som kan beskrive den. Eller ved at vise eleven, hvordan det gøres. Den kan kun læres ved at eleven selv forsøger og derigennem giver de forskellige begreber og metoder et konkret indhold. I disse forsøg kan eleven ved hjælp af et fornuftigt værktøj støttes i selv at opnå en fornuftig arbejdsform.

De teknisk betonedede arbejdsformer, som benyttes under konstruktionen af systembeskrivelser, og de som benyttes under konstruktionen af programmer i et eller andet programmeringssprog, er – i det mindste som systembeskrivelse er skitseret i afsnit III.1 – tilnærmelsesvis identiske. Eller sagt på en anden måde: en væsentlig del af de problemer, som opstår i forbindelse med konstruktionen af systembeskrivelser, og de, som opstår i forbindelse med konstruktionen af programmer, er sammenfaldende.

Det vil derfor være muligt med systembeskrivelse som indfaldsvinkel allerede fra starten at lægge vægt på indlæringen af netop den arbejdsform, som

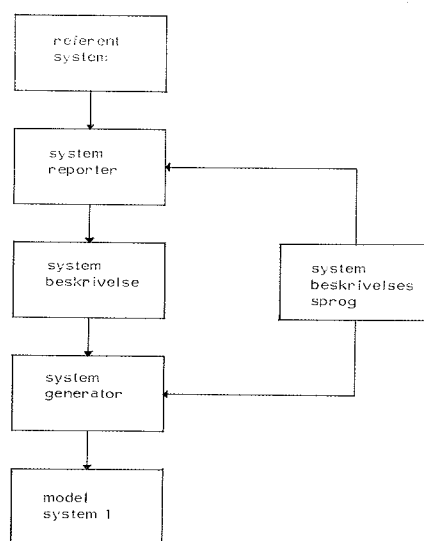


er væsentlig i forbindelse med indlæringen af de teknisk betonedede programmeringsfærdigheder.

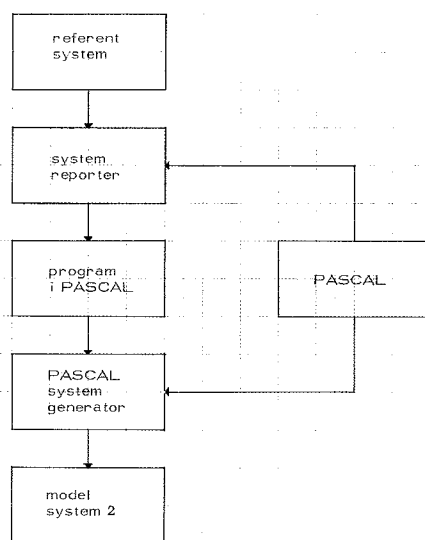
## 2.4 Yderligere fordele

Ved at anvende systembeskrivelse som indfaldsvinkel til programmeringsundervisningen vil vi kunne opfylde de krav, som blev opstillet i afsnit II. Men ud over dette opnår vi en række yderligere fordele.

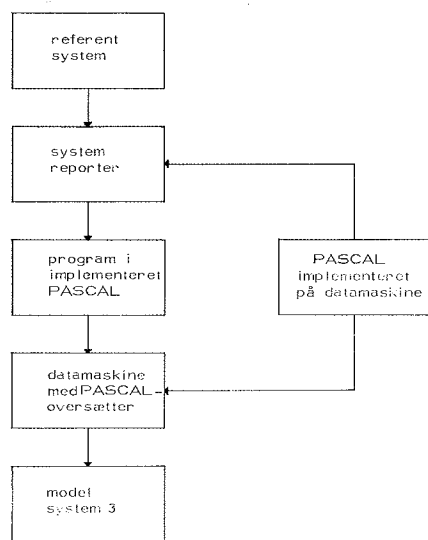
For det første vil det være muligt at diskutere, hvilke indskrænkninger og begrænsede muligheder, der ligger i anvendelsen af programmeringsværktøjet. En måde, at diskutere denne problematik på, er at tage udgangspunkt i det teoretiske fundament for datamaskinen. Ved hjælp af Turing maskiner eller partielt rekursive funktioner kan vi formelt definere, hvad det vil sige at et problem kan løses ved hjælp af en datamaskine<sup>17</sup>. Et andet udgangspunkt for diskussion af denne problematik er kommunikationsmodellen i afsnit III.1.1. Nedenstående tre figurer giver udtryk for tre forskellige kommunikationsprocesser, som er af betydning for forståelsen af de indskrænkninger og begrænsede muligheder, der ligger i anvendelsen af programmeringsværktøjet.



figur 1



figur 2



figur 3

På fig. 1 ligger de væsentligste begrænsninger i det anvendte værktøj – systembeskrivelsessproget – og i system reporterens begrænsede egenskaber. Disse begrænsninger gør sig gældende i alle tre tilfælde, men på grund af den større frihed i valget af sprog, kan de specielt sættes i relation til den konkrete virkelighed ud fra figur 1. I fig. 2 er der den yderligere begrænsning at programmeringssproget er totalt formaliseret og at det i sine egenskaber bærer præg af, at det skal kunne forstås og udføres af en datamaskine. Endelig træder datamaskinens begrænsede egenskaber frem i fig. 3. Den endelige lagerplads og datamaskinens hastighed medfører yderligere begrænsninger.

For det andet vil vi i uddannelsen af edb-arbejdere med fordel kunne anvende systembeskrivelsesværktøjet til at beskrive det datamatiske system. Ved at lave beskrivelser på forskellige abstraktionsniveauer og fra forskellige synsvinkler af det datamatiske system vil eleven få bedre mulighed for at opnå en forståelse for det datamatiske system og dets funktion.

For det tredie vil man på baggrund af arbejdet med systembeskrivelser kunne inddrage de begreber som er væsentlige for forståelsen af programmeringssprog og deres anvendelse. Data, datastruktur, variabel, navn, type og værdi er ligesom algoritme, kontrolstruktur og primitiv begreber, som har et konkret indhold i forbindelse med systembeskrivelse. På baggrund af arbejdet med systembeskrivelse vil der således være gode muligheder for ved hjælp af deltagernes begrebsforståelse at lette introduktionen af et programmeringssprog – og i denne introduktion at lægge vægt på de væsentlige momenter.

For det fjerde har vi med denne indfaldsvinkel til uddannelsen af edb-arbejdere bidraget til at give den enkelte edb-arbejder en mulighed for at kunne forstå sit arbejde i en større sammenhæng, og følgelig givet hende øgede muligheder for at få indflydelse på sin egen arbejdssituation. Ved i programmeringsundervisningen at tage udgangspunkt i den sammenhæng, som programmeringsværktøjet skal anvendes i, og ved at fremstille programmeringen, ikke som en isoleret og velafgrænset proces, men som en integreret del af en edb-anvendelse, har vi blandt andet givet edb-arbejderen en bedre baggrund for at kunne modarbejde tendensen til en stadig kraftigere arbejdsdeling indenfor edb-sektoren <sup>18</sup>.

### 3. En skitse af et muligt undervisningsforløb

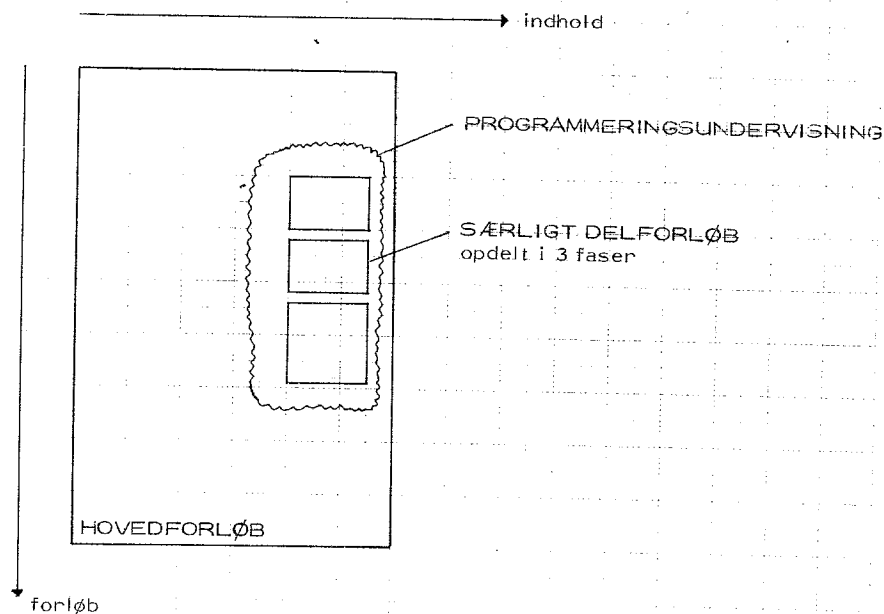
I dette afsnit vil jeg skitsere, hvorledes vi kan realisere en programmeringsundervisning, hvori systembeskrivelse benyttes som indfaldsvinkel, som en del af et problemorienteret, deltagerstyret uddannelsesforløb. Det er klart at denne skitse kun udgør ét forslag blandt mange mulige. Lige så klart er det, at et særdeles væsentligt kriterium ved vurderingen af denne skitse er praktisk anvendelighed.

Det er tanken, at afsnittet skal virke som en inspiration for den, som står overfor at skulle tilrettelægge og gennemføre en programmeringsundervisning. Samtidig skulle afsnittet konkretisere en række af de tanker, som jeg har givet udtryk for i det foregående.

#### 3.1 Organiseret som et særligt delforløb

Vi forestiller os et problemorienteret, deltagerstyret uddannelsesforløb, hvori det enten på forhånd er specificeret i formålene, at deltagerne bl. a. skal lære at programmere, eller hvor der undervejs opstår et behov for eller ønske om dette. I begge tilfælde vil det være nærliggende at arrangere programmeringsundervisningen som et særligt delforløb – et selvstændigt kursus – der indgår som en del af det problemorienterede uddannelsesforløb.

Den her angivne skitse kan betegnes som et halvfabrikeret kursus, der vil kunne indgå som et særligt delforløb i et problemorienteret, deltagerstyret uddannelsesforløb, og som åbner mulighed for en vis portion deltagerstyring og orientering i retning af de problemer, som indgår i selve hovedforløbet.



Som ovenstående tegning angiver er det særlige delforløb opdelt i 3 faser. Disse 3 faser er nærmere beskrevet i næste afsnit, og de angiver, hvorledes indlæringen af de teknisk betonedede programmeringsfærdigheder skal finde sted. De øvrige krav til en programmeringsundervisning, som vi har opstillet, vil blive tilgodeset, dels ved at problemstillinger fra hovedforløbet indgår som eksempel- og opgavemateriale i det særlige delforløb og de øvrige aktiviteter i selve hovedforløbet. Dette forhold er på tegningen angivet ved at programmeringsundervisningen omfatter mere end blot det særlige delforløb. Indlæringen af de teknisk betonedede programmeringsfærdigheder vil således komme til at indgå i en sammenhæng, som er eleverne vedkommende, og som illustrerer de problemer, der i forbindelse med den pågældende uddannelse findes ved anvendelse af edb.

Det halvfabrikerede kursus, som jeg her skitserer, vil kunne indgå i et problemorienteret, deltagerstyret uddannelsesforløb – uanset om deltagerne i det pågældende uddannelsesforløb er kommende edb-arbejdere, edb-undervisere eller en eller anden form for edb-anvendere. Det udmerker sig endvidere ved på den ene side at være så tilpas halvfabrikeret, at det åbner mulighed for deltagerstyring og samspil med selve hovedforløbet, og på den anden side at være så tilpas fastlagt, at vi vil kunne have

det "liggende i skuffen" og lade det indgå i et uddannelsesforløb ud fra et (uforudset) opstået behov.

### 3.2 Delforløbets tre faser

I det følgende vil jeg beskrive de tre faser i delforløbet én for én. Det er klart at det egentlige forløb og indholdet af disse tre faser vil variere afhængig af, hvilken større uddannelsesmæssig sammenhæng de indgår i. Jeg vil derfor i denne beskrivelse koncentrere mig om at vise, hvorledes systembeskrivelse skal indgå i faserne. Omfanget af de tre faser forholder sig ca. som 1:1:3, idet overgangen mellem faserne dog skal betragtes som udflydende.

#### Fase 1 – simpel systembeskrivelse og anvendelse af færdige programmer.

Deltagerne skal lave systembeskrivelser ved hjælp af et simpelt systembeskrivelsesværktøj. Dette værktøj anvendte jeg i afsnit III. 1.2, og det vil blive nærmere beskrevet i næste afsnit. Allerede fra starten lægges der således vægt på indlæringen af arbejdsformen. Det anvendte værktøj skulle støtte denne indlæring. De systemer, som skal beskrives vælger deltagerne selv ud fra selve hovedforløbet. Ligeledes henter læreren ved indføringen af systembeskrivelsesværktøjet sit eksemplermateriale fra hovedforløbet.

Samtidig hermed skal deltagerne stifte bekendtskab med det datamatiske system. De skal lære at anvende færdige programmer – muligvis findes der et programbibliotek, som indeholder programmer, der vil kunne anvendes ved løsning af problemer fra hovedforløbet. De skal lære det datamatiske system, som findes på stedet, at kende. Desuden skal de lære hvilke hovedelementer, som indgår i et datamatisk system.

I denne fase skal der etableres en forståelse for, at en yderligere formalisering af beskrivelsesværktøjet er nødvendig. Dette opnås dels gennem en forståelse for datamaskinens opbygning og funktion, og dels ved at påvise hvorledes deres egne beskrivelser indeholder uklarheder og tvetydigheder.

## Fase 2 – mere formaliseret systembeskrivelse med anvendelse af datamaskinen.

I denne fase skal deltagerne anvende et mere formaliseret systembeskrivelsesværktøj, som dog stadigvæk i væsentligt omfang skal tillade uformelle sprogelementer. Hvor formaliseret det nye værktøj skal være, og hvilke nye faciliteter det skal indeholde, kan variere fra undervisningsforløb til undervisningsforløb. Det nye beskrivelsesværktøj skal være en udvidelse af det, som blev benyttet i fase 1, og det skal gennem introduktionen af yderligere struktureringshjælpemidler understøtte den fortsatte indlæring af en fornuftig arbejdsform. Dette nye beskrivelsesværktøj kan således blandt andet indeholde faciliteter til strukturering af data og til beskrivelse af interaktion mellem komponenter, desuden procedurer og nogle kontrolstrukturer. Endvidere skal beskrivelserne nu være på imperativ form – dette var ikke et krav i fase 1, hvor de fleste naturligt vil vælge den beskrivende, berettende form frem for den imperative. Der vil således være lagt op til næste fase, hvor et egentligt programmeringssprog anvendes. Den detaljerede beskrivelse af handlinger og data skal stadigvæk være uformel.

De systemer, som beskrives i denne fase, kan enten være de samme som i fase 1, eller det kan være nye systemer hentet fra hovedforløbet.

De systembeskrivelser, som laves i denne fase, kan afvikles på datamaskinen. Dette kan ske ved, at vi lader datamaskinen checke syntaxen af den pågældende beskrivelse for derefter at give passende fejludskrifter <sup>19</sup>. En sådan afvikling af systembeskrivelsen på datamaskinen vil dels støtte deltagerne i indlæringen af brugen af de formaliserede sprogelementer, dels vil det supplere kendskabet til det datamatiske system.

## Fase 3 – programmering og kørsel af programmer.

Denne fase indeholder det, som en programmeringsundervisning i sin helhed traditionelt indeholder: konstruktion af programmer i et givet programmeringssprog og afvikling af disse programmer på et datamatisk system.

I denne sammenhæng har deltagerne imidlertid på dette tidspunkt af forløbet væsentlige forudsætninger: de har en (forhåbentlig) fornuftig arbejdsform, de har lært at håndtere et formaliseret sprog og de har stiftet bekendtskab med det datamatiske system. Disse forudsætninger vil i høj grad kunne anvendes i denne fase. Som Gries og Conway – måske lidt skarpt – formulerer det: "if someone approaching the programming process for the first time already has a systematic, well-disciplined approach to problem analysis in general, then learning to program a computer just means learning the nature and syntax of a programming language." (Conway73 preface).

Jeg vil ikke her komme nærmere ind på indholdet af denne fase. Blot vil jeg påpege, at opgave- og eksempelmaterialet vil kunne hentes dels fra de systembeskrivelser, som deltagerne allerede har lavet, dels fra andre problemstillinger fra selve hovedforløbet. Det vil således ikke her dreje sig om at flytte dronninger rundt på et skakbræt, eller at lægge farvede fliser i underlige mønstre. Problemstillingerne vil være taget fra en for deltagerne meningsfuld sammenhæng. Og de vil kunne illustrere den sammenhæng, som programmeringsværktøjet vil komme til at indgå i.

### 3.3 Det anvendte, simple systembeskrivelsesværktøj

Det systembeskrivelsesværktøj, som deltagerne skal benytte i fase 1 af det særlige delforløb, og som jeg benyttede i eksemplet i afsnit III.1.2., vil jeg her kort beskrive<sup>20</sup>.

Hovedideerne bag dette simple værktøj stammer dels fra DELTA-projektet, dels fra de erfaringer, der er gjort med det simple algoritmebeskrivelsesværktøj, som findes i Migselv76. Værktøjet skal på den ene side være simpelt, således at det ikke er noget problem at lære det. På den anden side skal det være af en sådan karakter, at det støtter indlæringen af en fornuftig arbejdsform. Dette opnås ved at vælge værktøjet således, at det tvinger brugeren til at benytte bestemte struktureringsmetoder, men ellers stiller vedkommende frit.



Et system beskrives således på følgende måde:

```

<navn>SYSTEM
BEGYND
    <beskrivelse af system>
SLUT <navn> SYSTEM

```

hvor <beskrivelse af system> er en beskrivelse af de komponenter, som systemet består af<sup>21</sup>. Hver af disse komponenter beskrives på følgende måde:

```

<navn>KOMPONENT
BEGYND
    DATASTRUKTUR
        <beskrivelse af datastruktur>
    HANDLINGSMØNSTER
        <beskrivelse af handlingsmønster>
SLUT <navn> KOMPONENT

```

Brugeren tvinges således til at beskrive et system som en række af komponenter, der hver har en datastruktur og et handlingsmønster. Men derudover er brugeren frit stillet.

Det er klart, at dette værktøj ikke sikrer at deltagerne får en fornuftig arbejdsform. Men i en undervisning, hvor der lægges vægt på indlæringen af en fornuftig arbejdsform, vil et sådant værktøj være en støtte.

#### IV. SAMMENFATNING

I denne artikel har jeg ud fra en kritisk, social synsvinkel diskuteret grundlaget for en praksis som underviser i programmering. Ved tilrettelæggelsen og gennemførelsen af en sådan undervisning foreslås det, at lade problemorientering og deltagerstyring være de ledende didaktiske principper, at inddrage den sammenhæng, som programmeringsværktøjet vil komme til at indgå i, og endelig under indlæringen af de teknisk betonedede programmeringsfærdigheder at lægge vægt på en struktureret og systematisk arbejdsform. Det foreslås videre at organisere indlæringen af de teknisk betonedede programmeringsfærdigheder som et særligt delforløb i et problemorienteret og deltagerstyret hovedforløb. Ved i dette særlige delforløb at anvende systembeskrivelse som indfaldsvinkel åbnes der mulighed for, at lade delforløbet indgå i et samspil med selve hovedforløbet og for at tage udgangspunkt i den sammenhæng, som programmeringsværktøjet vil komme til at indgå i. Samtidig skabes der gode betingelser for indlæringen af en struktureret og systematisk arbejdsform.

Synspunkterne i artiklen skal ikke betragtes som afsluttede, korrekte overvejelser. Tværtimod. Artiklen giver udtryk for mit personlige syn på problemerne og er således begrænset af mit nuværende erkendelsesniveau og min nuværende viden. En væsentlig del af synspunkterne er opstået i dialog med deltagerne på kurset "Programmeringsundervisning - hvem skal undervises hvordan?" afholdt på Datalogisk Afdeling, Århus Universitet efteråret 1975.

## Noter

1. Grundlaget for overvejelserne i denne artikel er mit personlige syn på datalogien og dens anvendelser, på samfundet og på relationerne mellem dem. I det kapitalistiske samfund (specielt i Danmark) udnytter et mindretal den skæve magtfordeling til at undertrykke flertallet. Denne undertrykkelse kommer konkret til udtryk i en række sammenhænge – blandt andet i forbindelse med anvendelsen af edb indenfor privat og offentlig virksomhed. Disse anvendelser medfører forandringer af social karakter og ændringer i enkeltpersoners og gruppers handlingsmuligheder. Som datalog anser jeg det for min opgave i mit arbejde at forholde mig kritisk til denne undertrykkelse og gennem min praksis at arbejde for en forandring af samfundet i frigørende retning, imod socialisme.
2. En detaljeret gennemgang af disse rammer findes i Illeris<sup>74</sup> s. 90–106.
3. Produktivkræfterne omfatter naturressourcer, for eksempel jord, mineraler og andre naturrigdomme, maskiner, værktøj kombineret i en bestemt teknologi, og menneskelig arbejdskraft, organiseret på en bestemt måde afhængig af teknologien. Produktionsforholdene er de relationer, mennesker etablerer med hindanden (for det meste uden eget ønske) som følge af den funktion, de har indenfor produktionsprocessen. Produktionsforholdene har at gøre med, hvem der ejer produktionsmidlerne (bygninger og maskiner), og hvem der kontrollerer selve processen og udøver magt. Produktivkræfterne og produktionsforholdene udgør tilsammen den historisk bestemte produktionsmåde (Israel<sup>74</sup> side 102).
4. Det der motiverer enkeltkapitalen til at forøge arbejdsproduktiviteten gennem indførelse af arbejdsbesparende maskineri, er udsigten til at opnå ekstraprofit. Deltagelsen i kampen om ekstraprofit er imidlertid ikke nogen frivillig sag for enkeltkapitalen, for hvis denne ikke til stadighed bestræber sig på at indføre de mest moderne og de mest arbejdskraftbesparende produktionsmetoder i virksomheden, vil det netop være den selv, der bliver udkonkurreret på markedet. I selve det kapitalistiske samfunds grundstruktur i konkurrencen mellem de mange enkeltkapitaler, ligger der altså en indbygget tendens til udvikling af produktivkræfterne (Linde<sup>75</sup> side 16).
5. Det vil sige til dele af arbejdsstyrken. I Otten<sup>73</sup> udtrykkes dette forhold således: "Når f. eks. produktiviteten som helhed øges, så forudsætter dette at totalarbejdskraftens kvalifikationer øges proportionalt – men ikke, at kvalifikationer hos de enkelte dele af denne totalarbejdskraft forøges.

Hvis arbejdspladsstrukturen (d. v. s. den fordeling af kvalifikationer, som de eksisterende arbejdspladser under den givne produktionsmåde kræver – min kommentar) forvandles under kapitalen, så følger heraf kun, at også den enkelte arbejdskrafts kvalifikationsstruktur på en eller anden måde må tilpasses denne forandring. Ellers ville man snart nå den tilstand, hvor der ikke mere fandtes konkret arbejde, som var i stand til at betjene

arbejdsmidlerne i den givne arbejdspladsstruktur. Ud fra de hidtidige præmisser ville det være en fejlslutning, hvis man antog, at der med stigende produktivitet også nødvendigvis måtte følge stigende krav til viden og færdigheder. Man kan i stedet for antage, at produktivitetens udvikling under kapitalen kræver forandrede, men ikke forbedrede kvalifikationer. (Ottens73 side 42-43).

6. På dette punkt stemmer Illeris' fremstilling hverken overens med de konkrete erfaringer, som er gjort i forbindelse med anvendelsen af edb som værktøj til planlægning og styring af produktionen, eller med visse resultater indenfor den nyere kvalifikationsforskning.

Illeris skriver: "... samtidig med teknificeringen og differentieringen er mange arbejdsprocesser blevet mere komplicerede. Der skal tages hensyn til mangeartede forskellige forhold, som kan spille ind - der kræves i stadig stigende omfang og stadig længere "ned" i systemet overblik, planlægning, samarbejde, kommunikation og smidighed. Det giver et øget behov for mere generelle færdigheds-kvalifikationer, men først og fremmest er det de personlige kreative kvalifikationer, der med denne udvikling kommer i søgelyset.

Den stigende differentiering og kompleksitet stiller således øgede krav til både generelle og specielle færdigheds-kvalifikationer, og ikke mindst til personlige kreative kvalifikationer. Men dermed ændrer arbejdet også på en anden måde karakter. For den stadig større andel af arbejdskraften, der indgår i de mere differentierede og komplicerede funktioner, opleves arbejdet mere meningsfuldt, mere personligt tilfredsstillende." (Illeris74 side 38-39).

Derimod konstaterer Otten i sin gennemgang af 'kvalifikationens rolle i kapitalens udviklingsproces og dens indvirkning på uddannelsesvæsenet (med Tyskland som eksempel)': "Den industrielle produktions struktur forvandles i stigende grad i retning af, at bestemte typer af regulations-funktioner automatiseres, samtidig med at der opstår nye kontrol-funktioner på ingeniør-niveau.

På den ene side fremstår således behovet for stigende kvalifikationer i et voksende antal delarbejder-funktioner. På den anden side sker der også en kvalifikationsformindskelse for massen af produktive arbejdere, som følge af stadigt faldende krav til den produktive massearbejdsevnes kvalifikationer." (Ottens73 side 150).

Ottens fremstilling understøttes af de konkrete erfaringer, som er gjort i forbindelse med anvendelsen af edb som værktøj til planlægning og styring af produktionen. Med edb som værktøj er det blevet muligt at overskue omfattende og komplekse sammenhænge. Indførelsen af produktionsplanlægnings- og styringssystemerne har derfor muliggjort en fjernelse af en lang række af planlæggende og styrende funktioner fra fabriksgulvet til centralt placerede planlægningsfunktioner. Ligeledes er det blevet muligt for eksempel ved hjælp af de såkaldt numerisk styrede værktøjsmaskiner at fjerne en række planlæggende og styrende funktioner fra arbejdet ved maskinen (omsætningen af en arbejdstegning til valg af skærestål, omdrejningstal etc.) til en centralt placeret programmerings-

funktion. I stedet for en arbejdstegning får arbejderen ved indførelsen af numerisk styrede værktøjsmaskiner udleveret en hulstrimmel til maskinen.

Illeris' konklusion om øgede krav om generelle færdighedskvalifikationer og i særdeleshed om personlige kreative kvalifikationer stemmer derimod udmærket overens med Ottens fremstilling af – som han kalder det – de nødvendige basiskvalifikationer. (se Otten73 side 43ff).

7. Et forsøg på at beskrive udviklingen i arbejdspladsstrukturen indenfor edb-sektoren findes i Edb-rap74.
8. Den kapitalistiske produktionsmåde har dobbeltkarakter, "idet den dels er en arbejdsproces – den proces, hvor arbejderen fremstiller et produkt (en vare) ved at bearbejde et eller andet råstof eller materiale oftest under anvendelse af forskellige værktøjer og/eller maskiner, og dels er en økonomisk proces – den proces, hvor der investeres nogle penge (i råmaterialer, maskiner m.m. og arbejdskraft) med det ene formål at få nogle flere penge ud af det i den anden ende. Kapitalisten må "logisk" være interesseret i at udforme arbejdsprocessen således, at den sikrer flest mulige penge ud i den anden ende – d.v.s. størst mulig profit, og han er "historisk" nødt til det for at klare sig i konkurrencen; men det kan kun ske på bekostning af arbejderne – d.v.s. øget udbytning i form af hårdere arbejde og/eller en teknologi, som reducerer arbejdsprocessens krav til arbejdskraftens kvalifikationer. En sådan teknologi stiller arbejderklassen svagere over for kapitalejerne, fordi den enkelte virksomhed så ikke er afhængig af en bestemt (begrænset) gruppe faglærte arbejdere. Kampen skal imidlertid ikke rettes imod flere maskiner, men imod udformningen og udnyttelsen af dem." (Mathiesen76 side 15).
9. En omfattende diskussion af, hvilken grundlæggende viden om edb og edb-systemer de grundlæggende uddannelser bør indeholde findes i Århuskonf75 side 632–658.
10. Se blandt andet Rapport75.
11. På den ene side er den arbejdsform, som fremstilles i Wirth73, fornuftig.  
På den anden side indskrænker den sig til en rent teknisk betonet programmeringsfærdighed, idet programmeringen er fremstillet som en isoleret, velafgrænset proces løstrevet fra enhver anvendelsessammenhæng.  
Ved tilrettelæggelsen og gennemførelsen af en programmeringsundervisning er der derfor – specielt med henblik på indlæringen af de teknisk betonedede programmeringsfærdigheder – meget at hente i Wirth's fremstilling (og i de andre citerede fremstillinger), selvom den set ud fra et kritisk, socialt synspunkt er mangelfuld.

12. Denne anvendelse af begreberne abstrakt og konkret afspejler et teknisk orienteret syn. Ud fra en anvendelsesorienteret synsvinkel må det, som her betegnes den abstrakte datatype (for eksempel et kartoteksskab med et tilhørende sæt af operationer: finde person, indsætte ny person etc.) netop betegnes konkret, mens repræsentationen af datatypen (ved hjælp af et array og nogle operationer på dette array) må betegnes abstrakt. Denne problematik er mere udførligt behandlet i arbejdsrapporten "Anvendelsesorienteret diskussion af formaliserede sproglige udtryksmidler", som jeg i skrivende øjeblik har under bearbejdning til publikation i samarbejde med Morten Kyng.
13. En detaljeret fremstilling af de foreløbige resultater fra dette projekt findes i DELTA75.
14. En diskussion af forskellige behov for systembeskrivelser og forskellige krav til systembeskrivelser findes i Århuskonf75 side 96-103.
15. Formatet af disse tegninger er taget fra DELTA75. I den øverste kasse angives systemkomponenten. De øvrige kasser angiver de komponenter, som vi på dette abstraktionsniveau har valgt at lade systemet bestå af. Desuden kan vi mellem to vandrette streger angive elementer, som optrådte som komponenter i en tidligere, mindre detaljeret beskrivelse, men som vi på dette abstraktionsniveau har valgt at erstatte af andre komponenter.
16. "En problemorienteret og deltagerstyret organisation af et uddannelsesforløb behøver ikke at betyde, at alle uddannelsesaktiviteter i sig selv organiseres problemorienteret og deltagerstyret som projekt - eller temaforløb. Tværtimod indebærer deltagerstyringen, at det står deltagerne frit for at indskyde forskellige delforløb af mere eller mindre fagorienteret og lærerstyret karakter og mere eller mindre afgrænset fra de øvrige dele af forløbet." Et sådant delforløb kan være et kursus, "der er tidsmæssigt afgrænset, og som koncentrerer sig om et bestemt emne, en bestemt problemstilling eller en bestemt funktion." (Illeris74 side 202ff).
17. Se for eksempel Schmidt75.
18. I Edb-rap74 er edb-personalets arbejdsforhold beskrevet. Personalet er her inddelt i 5 kategorier efter dets arbejdsfunktioner:
  - en ledelse
  - systemanalytikere og systemprogrammører
  - programmører
  - operatører
  - registreringspersonale

Meget tyder imidlertid på, at den videre udvikling vil føre til en endnu kraftigere arbejdsdeling. Således medfører de af IBM foreslåede "chief

programmer teams", at programmeringsarbejdet opdeles i

- en overordnet planlægning: opdeling i mindre moduler, uddeligerig af de enkelte moduler og programmering af hovedprogrammet, som anvender de mindre moduler (the chief programmer).
- kodning af moduler (additional programmers).
- administration, dokumentationsvedligeholdelse (the librarian).

19. En sådan syntax-checker vil kunne laves automatisk ved hjælp af en parsergenerator - se for eksempel BOBS74.
20. På dette sted beskrives syntaxen af det anvendte systembeskrivelsesværktøj. I DELTA75 findes en semantisk beskrivelse af et sådant værktøj. Jeg har ikke i denne fremstilling taget stilling til, hvorledes værktøjets semantik bør fremstilles i undervisningen. I "Anvendelsesorienteret diskussion af formalisrede sproglige udtryksmidler" (se note 12) findes en diskussion af denne problemstilling.
21. Jeg har i denne fremstilling valgt at lade system komponenten bestå af lutter simple komponenter, det vil sige komponenter, der ikke indeholder andre komponenter. Med andre ord optræder alle indre komponenter på samme niveau.

I DELTA75 hedder det derimod, "we will conceive a system as a nested structure of components."

I praktiske anvendelser er jeg imidlertid ikke stødt på brugen af flere niveauer af komponenter i samme beskrivelse, og har derfor i denne fremstilling foretaget ovenstående valg.

# Litteraturliste

- Andersen73      Industri, stat og teknisk naturvidenskabelig uddannelse, E. Andersen, DSF 1973.
- Andersson74      Industriföretagets produktionsstyring, John Andersson, Kungliga Tekniska Högskolan, Stockholm 1974.
- Betænkning72      Betænkning om edb-undervisning i det offentlige uddannelsessystem, Betænkning nr. 666, 1972.
- BOBS74      A short description of a translator writing system (BOBS-system), B. Kristensen m. fl., DAIMI PB-41, Århus Universitet 1974.
- Cashman75      On the need for teaching problem-solving in a computer science curriculum, W. Cashman and W. Mein, SIGCSE BULLETIN 1975.
- Conway73      An introduction to programming, R. Conway and D. Gries, Winthrop 1973.
- Datanom74      Uddannelsesplaner for datanomuddannelsen, oktober 1974.
- Delta75      Systemdescription and the DELTA language, DELTA report no. 4, E. Holbæk-Hanssen, P. Håndlykken og K. Nygaard, Norsk Regnesentral 1975.
- Dijkstra71      A short introduction to the art of programming, E. W. Dijkstra, EWD 316, Technologische Hogeschool, Eindhoven, 1971.
- Edb-assistent73      Rapport om etablering af en 1-årig edb-assistentuddannelse, Forsøgssrådet for de erhvervsfaglige grunduddannelser, 1973.
- Edb-skole75      Rapport om edb-assistentuddannelsen, B. Madsen, S. Clausen, P. Andersen, H. Andersen. Lavet i forbindelse med kursus på DAIMI 1975.
- Edb-rap74      Edb-rapporten, DSF 1974.
- EFG75      Undervisningsvejledning til timeplaner og læreplaner for fællesfag ved EFG, Forsøgssrådet for de erhvervsfaglige grunduddannelser, August 1975.
- Ehn74      Bidrag till et kritisk socialt perspektiv på utvecklingen av datorbaserade informationssystem, P. Ehn, intern kommunikation, Stockholm 1974.
- Gries74      What should we teach in an introductory programming course, D. Gries, SIGCSE BULLETIN vol 6, 1, 1974.
- Hoare69      An axiomatic basis for computer programming, C. A. R. Hoare, CACM 12 (1969), pp 576-583.



- Hoare72 Proof of correctness of data representations, C.A.R. Hoare, Acta Inf., 2 (1972), pp 271-281.
- Hoare73 Recursive data structures, C.A.R. Hoare, University of Stanford, STAN-CS-73-400, 1973.
- Illeris74 Problemorientering og deltagerstyring - oplæg til en alternativ didaktik, K. Illeris, Munksgård 1974.
- Israel74 Sociologisk Grundbog I, J. Israel, Gyldendal 1974.
- Lecarme74 Structured programming, programming teaching and the language PASCAL, O. Lecarme, SIGCSE BULLETIN vol 6, 2, 1974.
- Linde75 Kapital, krise og stat, B. Linde og S. Fink, Fagtryk 1975.
- Liskov Programming with abstract data types, B. Liskov og S. Zilles, SIGPLAN Notices 9 (1974), pp 50-60.
- Mathiesen76 Uddannelse og produktion, Anders Mathiesen, Munksgaard 1976.
- Migselv74 En datalære bog, Lars Mathiassen, Gyldendal 1976.
- Otten73 Kapitaludvikling og kvalifikationsudvikling, Dieter Otten, Kurasje/Modtryk 1975.
- Rapport75 Om edb i undervisningen, Rapport fra efteruddannelseskursus for gymnasielærere, efteråret 74, Århus 1974.
- Schmidt75 Beregnelighed, E. Meineche Schmidt, DAIMI FN-11, Århus Universitet 1975.
- Turski73 Programming teaching techniques, W.M. Turski editor, North-Holland/American Elsevier 1973.
- Wirth71 The programming language PASCAL, N. Wirth, Acta Inf., 1 (1971), pp 35-63.
- Wirth73 Systematic programming: an introduction, N. Wirth, Prentice-Hall 1973.
- Århuskonf75 Århus konferencen januar 1975, Arbejdsformer i systemudvikling, DAIMI PB-46, 1975.