

<b>1. PENSUM .....</b>	<b>2</b>
<b>2. BASIC TERMS AND DIFFERENT SCHOOLS OF INFORMATION SYSTEMS DEVELOPMENT .....</b>	<b>3</b>
2.1. Avison & Fitzgerald.....	3
<b>3. PARADIGMER .....</b>	<b>5</b>
3.1. Hirschheim & Klein .....	5
<b>4. A FRAMEWORK FOR INFORMATION SYSTEMS DEFINITION. ....</b>	<b>11</b>
4.1. Avison & Wood-Harper .....	11
<b>5. PARTICIPATION AND SYSTEMS DESIGN. ....</b>	<b>16</b>
5.1. Mumford .....	16
<b>6. REVIEWS AND TECHNICAL WALKTHROUGHS.....</b>	<b>19</b>
<b>7. FORMAL TECHNICAL REVIEW. ....</b>	<b>19</b>
7.1. Andersen et. al. Og Pressman .....	19
<b>8. HUMAN ACTIVITY SYSTEM &amp; SOFT SYSTEM METHODOLOGY .....</b>	<b>21</b>
8.1. Hvad er et human activity system? .....	21
8.2. Soft System metoden .....	22
8.3. SSM til udarbejdelse af informationssystemer .....	33
<b>9. STRUKTURERET ANALYSE. ....</b>	<b>35</b>
9.1. Introduktion - del I.....	35
9.2. Modeller og værktøjer .....	38
9.3. Teknikker og strategier .....	41
9.4. Værktøjer – del II .....	44
9.5. Entitets/reasons diagrammet.....	49
9.6. Tilstandsdiagrammer .....	52
9.7. Dataordbog .....	53
9.8. Procesbeskrivelser .....	54
9.9. EDB-støttet systemudvikling .....	56
9.10. Del III – basisstrategi for analysen .....	57
9.11. Fysisk model af eksisterende system .....	59
9.12. Hændelsesopdeling .....	59
9.13. Informationsanalyse-teknik .....	62
<b>10. SOCIO-TEKNISK ANALYSE OG ETHICS.....</b>	<b>63</b>
10.1. Ethics.....	63
<b>11. SUPPLERENDE LITTERATUR .....</b>	<b>67</b>
11.1. Budde et. Al.....	67
11.2. Bansler & Bødker .....	68
11.3. Floyd.....	74
11.4. Avison & Fitzgerald.....	75
11.5. Bansler.....	85
11.6. Kautz .....	88
11.7. Floyd et. Al. ....	90

## 1. PENSUM

**Avison & Fitzgerald**, Information Systems Development: Methodologies, Techniques and Tools, McGraw-Hill, Maidenhead, UK, 1995, pp. 1-15, 110-127, 353-364, 417-457

**Hirschheim & H. Klein**: Four Paradigms of Information Systems Development. In: Communication of the ACM, 32 (10), pp. 1199-1216

**Avison & Wood-Harper**, Multiview - An Exploration in Information Systems Development, Blackwell, Oxford, UK, 1990, pp. 20-35

**Mumford**, Effective Systems Design and Requirements Analysis - The ETHICS Approach. MacMillan, Information Systems Series, UK, 1995, pp. 12-26, 164-183

**Andersen et al.**, Professional Systems Development - Experience, Ideas and Action. Prentice Hall, London, UK, 1990, pp. 137-142

**Pressman**, Software Engineering - A Practitioner's Approach. McGraw-Hill, New York, USA, 1992, pp. 562-570

**Checkland & Scholes**: Soft Systems Methodology in Action, Wiley, Chichester, UK, 1990, pp. 13-58

**Delskov & Lange**, Struktureret Analyse – Integreret Systemanalyse, Teknisk Forlag, København, 1991, pp. 17-237

**Budde et al.**, Prototyping - An Approach to Evolutionary System Development, Springer, Berlin, Germany, 1992, pp. 33-48

**Bansler & Bødker**, A Reappraisal of Structured Analysis: Design in an Organizational Context. In ACM Transactions on Information Systems, 11 (2), 1993, pp.165-193

**Floyd**: A Comparative Evaluation of System Development Methods. In: Olle et al. (eds.): Information Systems Design Methodologies: Improving the Practice, North-Holland, 1986, pp. 19-37

**Bansler**, Systems Development in Scandinavia: Three theoretical Schools. In: Office: Technology & People, 4 (1989), pp. 117-133

**Kautz**, User Participation and Participatory Design: Topics in Computing Education, In Human-Computer Interaction, 11 (3) (1996), pp. 267-284

**Floyd et al.**, Out of Scandinavia: Alternative Approaches to Software Design and Systems Development. In Human-Computer Interaction, 4 (1989), pp. 253-350

## 2. BASIC TERMS AND DIFFERENT SCHOOLS OF INFORMATION SYSTEMS DEVELOPMENT

Pensum: Avison & Fitzgerald, pp. 1-15 (DIS kompendium S. 2-9).

---

### 2.1. Avison & Fitzgerald

Hovedemnerne er informationssystemer og hvordan disse udvikles så de er brugbare for organisationer. Hovedfokus er lagt på det menneskelige og organisatoriske aspekter i en udviklingsproces.

Ved identificering af systemer er det relevant at skelne mellem systemer, der kan være fælles for forskellige organisationer (lønningssystem, ordremodtagelse, projektplanlægningssystem osv.) og systemer der er så specialiseret, at de sjældent kan bruges i mere end én type organisation (stemmeregistrering ved lokale valg eller udlånssystem i et bibliotek).

#### Det menneskelige aspekt

Det er vigtigt at være specielt opmærksomme på det menneskelige og det organisatoriske aspekt, da det ofte er problemer med disse, der forårsager fiaskoer. Det kan fx være problemer med dårligt uddannede medarbejdere og dårlig anvendelse af metoder, teknikker og værktøjer.

Det menneskelige aspekt omhandler de mange former for brugere, der er involveret i udviklingen og som alt for ofte blot betegnes som ”brugere”. Tekstens synsvinkel er dog at denne betegnelse må anvendes langt mere nuanceret. Regular users (sekretærer osv.) skal fx trænes tilstrækkeligt så de benytter systemet korrekt og er godt motiverede. Casual users (direktionen osv.) har ofte ikke tid til at lære systemet at kende og intuitivt design er således vigtigere her.

Eksterne brugere dækker over fx eksterne revisorer og kunder, der har behov for tilgang til databasen. Professionelle brugere dækker typisk over teknikerne og superbrugerne.

#### Det organisatoriske aspekt

Selvom organisationer er forskellige er et fælles behov for udvikling af informationssystemer behov for ledelse. Fælles for udviklingen er behovet for en ”Informationssystems strategigruppe”, som har ansvaret for en projektplan og opdatering af denne i takt med at ændringer opstår.

”Steering committee” vil gennemgå hvert enkelt projekt indenfor planen og sikre at kravene fra strategigruppen bliver opfyldt. Steering committee har også til opgave at kontrollere/lede projektet.

#### Behovet for metode (methodology)

Virksomheder har altid haft behov for systemer, selvom det kun er på det seneste at man er begyndt at anvende EDB. I takt med at EDB-systemer blev brugt mere og ledelsen forlangte mere passende systemer for deres store udgifter, måtte nogle ændringer gennemføres:

- Der var en stigende interesse for analyse- og designdelen i systemudviklingen, samt systemanalytikerens og programmørens rolle.
- Udvikling væk fra engangsløsninger mod mere integrerede informationssystemer.
- Større forståelse for behovet for en accepteret metode for udvikling af informationssystemer.

Krav til metode<sup>1</sup>

---

<sup>1</sup> For en yderligere uddybning, se kompendium om IT-Kvalitet

Disse afviger fra projekt til projekt og afhænger blandt andet af hvilken indgangsvinkel, der tages. Nogle metoder lægger et mere humanistisk fokus, mens andre er mere videnskabelige eller pragmatiske<sup>2</sup>. Nogle krav til metoden kunne være:

- Nøjagtigt at registrere krav til informationssystemet
- At gøre effektiv overvågning af udviklingen mulig vha. en systematisk metode
- At færdiggøre et projekt indenfor en rimelig tid og til acceptable omkostninger<sup>3</sup>
- Producere et veldokumenteret system, der er nemt at vedligeholde
- Så tidligt som muligt at gøre opmærksom på evt. nødvendige ændringer
- At lave et system, der er værdsat af interessenterne

#### Diverse definitioner<sup>4</sup>

**Data** – ustruktureret fakta

**Information** – opstår ved at vælge og fortolke data.

Den essentielle forskel er således at data ikke bliver fortolket, hvorimod information har en betydning og er brugbar overfor modtageren i en bestemt kontekst. Nogle definerer information som explicit viden.

**Knowledge/viden** bygger på akkumuleret information. Viden for en person behøver ikke at være det for en anden.

**System** – indeholder relationer. Fx indeholder uddannelsessystemet relationer mellem lærere, skoler, elever, bøger osv. Systemer har også et formål, fx at levere relevant information til brugeren på det rigtige tidspunkt med det rigtige detaljeringsniveau. Et informationssystem kan defineres som:

”et system som opsamler, lagrer, behandler og leverer relevant information til en organisation...”<sup>5</sup>

**Metode** – en samling af procedurer, teknikker, værktøjer og ”hjælpe-dokumentationer”, der hjælper udviklere med at implementere et nyt informationssystem.

**Teknik** – en måde man gør en bestemt handling i udviklingsprocessen på. I forbindelse med udførelse af en teknik, kan en metode foreslå et bestemt værktøj (fx brainstorm, rich picture osv.)

---

<sup>2</sup> Se gennemgang af forskellige paradigmer i forbindelse med systemudvikling i næste afsnit.

<sup>3</sup> Bemærk her at man ikke tale om at blive færdig til tiden eller overholde budgettet – det har man forlængst opgivet, jvf. Peter Muyani's plancher.

<sup>4</sup> IB-kompendium s.8

<sup>5</sup> Frit oversat fra IB-kompendium s.8, se endvidere Karlheinz's plancher med forskellige definitioner

### 3. PARADIGMER

Pensum: Hirschheim & H. Klein, pp. 1199-1216 (DIS kompendium S. 10-27).

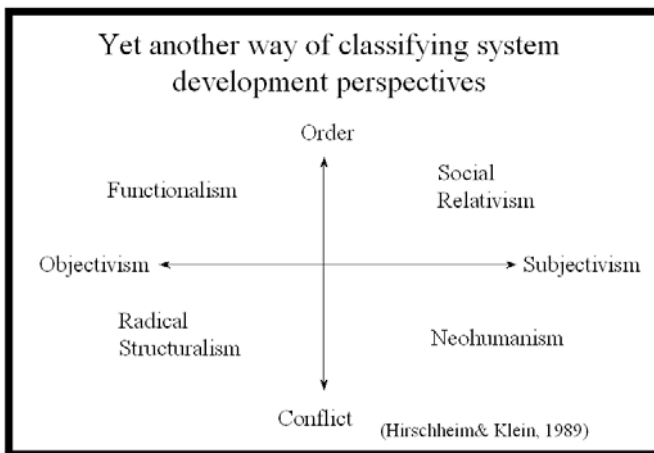
#### 3.1. Hirschheim & Klein

Tekstens hovedpointe er, at på trods af at systemer udvikles på baggrund af samme indgangsvinkler (fx OOAD og struktureret analyse) vil de samme problemstillinger ende med at blive behandlet på forskellige måder og resultaterne bliver således vidt forskellige. Grunden til dette skal findes i udviklerenes forskellige opfattelser og synsvinkler (paradigmer)<sup>6</sup>. Forfatterne bag teksten mener således, at det er vigtigt, at være opmærksom på dels hvilke paradigmer man selv og andre udviklere tilknyttet projektet er tilhængere af. Til at understøtte denne mere nuancerede udviklingsproces kan Multiview fx anvendes – se næste artikel. Det understreges dog også, at metoder, der anvender og kombinerer flere paradigmer (ISAC, SSM og Multiview), skal vurderes kritisk af netop samme grund<sup>7</sup>.

Da systemudviklere laver indgriben i systemets sociale omgivelser, som en del af selve systemudviklingen, er det naturligt at skelne mellem to typer af relaterede opfattelser:

- Ontologiske opfattelser – hvordan udvikleren anskuer de socio-tekniske omgivelser (Eksisterer virkeligheden inde i individets hoved eller findes der en fælles objektiv opfattelse).
- Epistemologiske opfattelser – hvordan indsamler systemudvikleren viden (Ved naturvidenskabelig metode eller ved fortolkning).

De to typer af opfattelser omkring verden (Ontologi) og viden (Epistemologi) gav Burrell og Morgan<sup>8</sup> anledning til at opstille to dimensioner (objectivism/subjectivism og order/conflict). Se eventuelt Burrell og Morgans ”Sociological Paradigms” fra IT-forandringsledelses kompendiet s. 22. De to dimensioner skaber grundlaget for 4 ”grund”-paradigmer:



Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ...

17

Figur 1 ISD Paradigmer

<sup>6</sup> “The most fundamental set of assumptions adopted by a professional community that allows its members to share similar perceptions and engage in commonly shared practices.” Hirschheim & Klein, s.12 i kompendiet

<sup>7</sup> IB-kompendie s. 25 nederst

<sup>8</sup> IB-kompendie, s. 12

Paradigmeopfattelsen kan ikke udtrykkes direkte og er baseret på en række common-sense antagelser og må derfor ikke betragtes som nogen endegyldig sandhed, ligesom de 4 beskrevne opfattelser ikke nødvendigvis er dækkende. En del projekter er blevet fulgt og studeret og er derefter blevet opdelt efter paradigme, hvilket har resulteret i 4 ”skole-eksempler” på paradigmer i udviklingsprojekter:

### **Funktionalist-paradigmet (Story 1)**

Her fokuseres der på at levere rationelle forklaringer på sociale sager. Udviklerens primære rolle er at være ekspert i teknologi, værktøj, metoder til system design og projektledelse. Udviklerens applikationer har til formål at gøre systemudvikling mere formelt og rationelt ved at placere mindre troværdighed på menneskelig intuition, dømmekraft og politik<sup>9</sup>. Udviklerens rolle er at udvikle et system, der modellerer virkeligheden så det kan anvendes som et værktøj af ledelsen til at opnå mål. Hovedformålet ved dette paradigme er profitmaksimering (shareholder value) – dvs. at det færdige system skal bidrage til dette. Ledelsen er den vigtigste beslutningstager.

Epistemologien<sup>10</sup> er positivistisk, forstået på den måde at udvikleren opnår viden om organisationen ved at lede efter målbare årsag-virknings sammenhænge (at ting bliver dokumenteret ved naturvidenskabelige beviser). Den ontologiske<sup>11</sup> opfattelse er realisme eftersom man her mener at en empirisk og objektiv organisatorisk virkelighed eksisterer.

### **Social relativisme-paradigmet (Story 2)**

Opfattelsen her er på mange måder modsat det første paradigme. Her mener man at viden om mennesker ikke er enkel at indsamle og opnå pga. dens kompleksitet og flygtighed. Der findes ikke en endegyldig og objektiv virkelighed, blot forskellige opfattelser af denne.

Udviklerens rolle er her at interagere med ledelsen for at finde ud af hvilket system og hvilken virkelighed, der giver mening. Der er intet objektivt succeskriterium – alt afhænger af hvad interessenterne tror på.

Den epistemologiske opfattelse er anti-positivistisk, da søgning efter empiriske forklaringer på sociale fænomener her menes at være misforstået og bør erstattes af sund fornuft og fortolkning af den konkrete situation. Den ontologiske opfattelse er nominalistisk, da virkeligheden ikke er ”givet” og statisk, men eksisterer inde i hovedet på det enkelte individ.

### **Radikal strukturalist-paradigmet (Story 3)**

Opfattelsen her er at der er en interessekonflikt mellem shareholders og arbejdere. Derfor er udvikleren her tvunget til at vælge side. I dette tilfælde vil udvikleren bruge systemer til fx at påvirke arbejdsintensiteten, arbejdsinstrumenter til arbejderens fordel. Den kunne eksempelvis være systemer, der sætter mere fokus på arbejderens håndværk og giver mere tid til den kreative proces og derved øger arbejdsmorale. Gevinst ved produktivitetsændringer skal komme arbejderne til gode: kortere arbejdstid, mere tid til planlægning og kreativt arbejde. Konkrete eksempler kan være udviklingsprojekter sponsoreret af fagforeninger.

Formålet ved systemudvikling i dette tilfælde skal således være at hjælpe arbejderne til at overvinde kapitalismens begrænsninger ved at støtte arbejds-aktivisme.

---

<sup>9</sup> IB-kompendie, s.14

<sup>10</sup> En gren af filosofien, der forsker i oprindelsen, adfærd, metode og grænser af menneskelig viden – frit oversat fra Websters

<sup>11</sup> ”en gren af metafysikken, der forsker i eksistens og det at være” – frit oversat fra Websters

Den epistemologiske opfattelse er her positivistisk, da der her anlægges et materialistisk syn på historie og samfund.

**Neohumanist-paradigmet (Story 4)**

Opfattelsen er her at organisationens liv ændres gennem system udvikling. Denne ændring er dog kraftigt begrænset af den sociale indflydelse som kanaliserer værdier, normer og opfattelse af deltagere. Samfundet og organisationen er arrangeret omkring arbejde, fælles forståelse og befrielse (emancipation) og her er det netop udviklerens rolle som befrier, der fokuseres på. Med befrielse menes at alle bindinger og alt tvang fjernes og man søger frihed og retfærdighed<sup>12</sup> – altså revolution. I denne proces har udvikleren brug for at opnå internt kendskab til organisation og deltagelse/interaktion er derfor altafgørende. I den forbindelse er der en række hindringer som udvikleren skal være opmærksom på:

- Autoritet og uretmæssig magt – kan få personer til at tilbageholde information
- Gruppe pres / ”gruppe tankegang” kan ”sløre” personlige holdninger
- Tid og ressource-begrænsninger
- Sociale forskelligheder – fx forskelligt uddannelsesniveau
- Sprogets begrænsninger – at sikre at budskabet forstås ens af alle parter

**Four schools of system development**  
(according to Hirschheim/Klein)

-

**A summary**

Paradigm	Developer archetype	Systems development proceeds	Elements used in defining IS	Examples
Functionalism	(Neutral) Expert	from outside, application of formal concepts, planned intervention with rationalistic methods	people, hard-, software, rules as objective entities	Structured analysis
Social Relativism	Facilitator	inside, improvement of subjective understanding by adapting to internal forces	subjectivity of meaning, sharing of meaning	ethnographic approaches (Florence)
Radical Structuralism	Warrior (for social progress)	from outside, raising of ideological consciousness, organized political action, adaptation of methods to class interests	people, hard-, software, rules as objective entities in the service of class interests	trade-union approaches (Utopia)
Neohumanism	Emancipator	inside, improvement of understanding and rationality, emancipation and liberation from social constraints	people, hard-, software, rules as objective entities for TKI, subjectivity of meaning for other KI	critical social theory

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ...

18

**Figur 2 Oversigt over 4 paradigmer**

Teksten behandler 2 projekter: et system til diagnosticering af flymotorer hos SAS og et projekt på et avistrykkeri – Utopia. Projektledelsen i sidstnævnte bestod af faglige repræsentanter og arbejdere og havde til formål at fokusere på det layoutmæssige håndværk i en avisproduktion. Nedenstående er en oversigt over forskellige resultater på 8 forskellige punkter alt afhængig af paradigme-opfattelse<sup>13</sup>.

<sup>12</sup> IB-kompendium, s.20

<sup>13</sup> Begge skemaer er frit oversat fra IB-kompendiet s. 23-24

Utopia Projekt	Functionalism	Radical Structuralism
Teknologisk arkitektur – hard- og software konfiguration og matching med organisationen	Tekstbehandling forbindes med layout-software for at eliminere manuel indtastning.	Journalisters og redaktørers tekstbehandling med layoutafdelingens server. Derved opsættes ekstra arbejdsstationer konfigureret til at støtte layout-processen.
Type af informationsflow – hvad skal informationen, der bliver behandlet her bruges til?	Fra redaktør og journalister til maskiner; produktivitetskontrol til ledelse	Feedback til journalister fra layout-medarbejdere før trykning. Produktivitetskontrol er mulig, men ikke hovedformålet.
Kontrol af brugere – hvordan sikrer systemet at fx magt ikke misbruges	Produktivtetskontrol for redaktør og journalister. Visuel kvalitetskontrol for layoutmedarbejdere reduceret eller elimineret.	Layoutmedarbejdere bevarer kontrollen med kvaliteten af deres produkt.
Kontrol af systemudvikling – hvem har kontrol over udviklingsprocessen?	Tekniske eksperter	Faglige repræsentanter og arbejdere
Adgang til informationer – hvem har adgang til den information som systemet stiller til rådighed?	Kun journalister og redaktører.	Journalister, redaktører og layoutmedarbejdere.
Uddannelse -	Basale computer og layout-færdigheder til journalister og redaktører.	Nye layout og computer færdigheder til layoutmedarbejdere
Primær eksistensberettigelse	Shareholder value	Fremhæve layoutarbejdets kvaliteter og levere et mere tiltrækkende produkt.

Tabel 1 Utopia Projektet

SAS udviklede i første omgang et system til diagnosticering af flymotorer, som skulle automatisere diagnosticeringen og derved minimere fejlprocenten. Det modsatte var imidlertid en realitet ved projektets afslutning. Man besluttede derefter at omdesigne systemet så beslutningerne i højere grad blev lagt i mekanikernes hænder og dette var en succes. Problemet var bl.a. at mekanikernes viden opnået gennem erfaring og uddannelse ikke let lod sig formalisere<sup>14</sup>. Nedenstående er en oversigt over forskellige resultater på 8 forskellige punkter alt afhængig af paradigme-opfattelse.

SAS engine Projekt	Functionalism	Social relativism/neohumanism
Teknologisk arkitektur – hard- og software konfiguration og matching med organisationen	Automatisk diagnosticering af motordele for at finde fejl.	Vurdering af servicebehov af erfarne mekanikere på baggrund af deres tavse viden.
Type af informationsflow – hvad skal informationen, der bliver behandlet her bruges til?	Instruktioner fra system til mekanikere.	Bruger (mekanikeren) beskriver problem/symptomer til computeren som foreslår løsningsmodeller.
Kontrol af brugere – hvordan sikrer systemet at fx magt ikke misbruges	Mekanikerne er kontrolleret af systemet.	Systemet kontrolleres af mekanikerne.
Kontrol af systemudvikling – hvem har kontrol over udviklingsprocessen?	Tekniske eksperter	Mekanikerne, deres faglige repræsentanter og ledelsen.
Adgang til informationer – hvem har adgang til de information som systemet stiller til rådighed?	Mekanikere og ledelse.	Mekanikere og ledelse.
Fejlhåndtering – hvordan håndteres fejl og hvem håndterer?	Fejlfinding på baggrund af statistik.	Bedre kvalitetskontrol af mekanikerne som føler ansvar for deres arbejde.
Uddannelse -	Begrænset til brugen af systemet	Diskussion af holdninger overfor et system. Forståelse af nøglekoncepter bag systemet og uddannelse i selve brugen.
Primær eksistensberettigelse	Ekspert system, der skal erstatte menneskelig dømmekraft for at eliminere fejl	System for eksperter, hvis bedømmelse vægter højere end systemets.

Tabel 2 SAS engine projektet

Hovedpointen er ikke så meget det at forfatterne mener at der eksisterer 4 paradigmer, men mere at der eksisterer forskellige opfattelser af Informationssystem-udvikling. Og at de forskellige opfattelser forårsager forskellige resultater. Ligesom forfatterne efterlyser alternativer til den traditionelle funktionalistisk indgangsvinkel og generelt systemudviklere, der er opmærksom og bevidst om andre paradigmer.

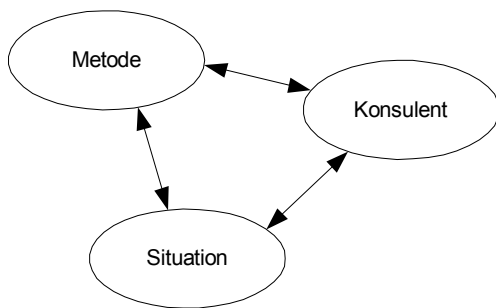
<sup>14</sup> De to projekter er beskrevet på side 11 i IB-kompendiet.

## 4. A FRAMEWORK FOR INFORMATION SYSTEMS DEFINITION.

Pensum: Avison & Wood-Harper, Multiview pp. 20-30 (DIS kompendium S. 28-35).

### 4.1. Avison & Wood-Harper

Denne artikel beskæftiger sig med metoden Multiview, og ser på forholdet mellem metoden, konsulenten der foretager analysen og situationen hvori analysen foregår. Det vil sige, at flere aspekter gør sig gældende under system-analyse og design, fordi de kan påvirke hinanden. Multiview metoden er så nuanceret, at den skelner mellem forskellige setups<sup>15</sup>, dvs. at konsulentens standpunkt (paradigme) eller situationen hvori hele setupet udspiller sig har indflydelse på det endelige resultat. Metoden er derfor en situationsorienteret metode. Se figur 1 nedenfor.



Figur 3

Resultatet afhænger således af projektet, menneskene der foretager analysen og hvilken metode der anvendes.

For at opnå et godt resultat ved System- analyse og design, er det vigtigt at være systematisk (jf. Kautz er systematisk ikke slavisk). Det er her "Multiview metoden" kommer ind i billedet. Metoden dækker over fem forskellige stadier<sup>16</sup> (Se også figur 3 eller 5) af System- analyse og design, hvor problemstillingen beskues fra forskellige synsvinkler/perspektiver. Multiview giver muligheden for, at se analyse- og design processen fra et socialt (menneskeligt perspektiv), politisk eller et teknisk perspektiv.

Artiklen fastslår at den perfekte metode er en illusion, og dette er netop grundlaget for Multiview. Som navnet "Multiview" antyder, kombineres multiple perspektiver eller metoder i én metode. Det vil sige, at metoden muliggør et mere fuldendt og nuanceret analyse- og designprodukt, netop fordi problemstillingen beskues fra flere perspektiver.

Essensen er, at forskellige konsulenter anvender forskellige metoder, afhængig af hvordan konsulenten vælger at se på situationen. Ligeledes afhænger resultatet af, hvilket standpunkt (paradigme<sup>17</sup>) konsulent har, dvs. hvordan konsulenten ser verden. Formålet med metoden er derfor, at beskrive et overordnet og mere fleksibelt stykke værktøj, der kombinerer vigtige aspekter fra flere store metoder og giver

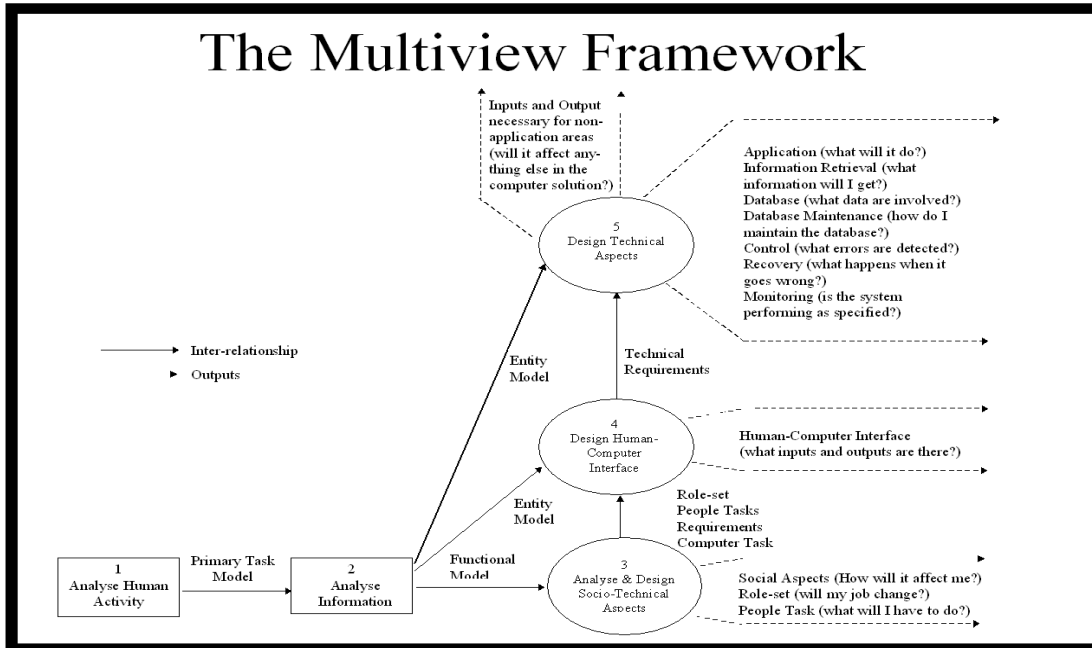
<sup>15</sup> Det skal forstås som forskellige kombinationer af udvikler-paradigme, metodevalg og situation (situation kan være forskellige projekter eller dele af et projekt. Dvs. analyse, design...).

<sup>16</sup> De fem forskellige stadier nævnes kun kort her, da de gennemgås senere i teksten. **Stadie 1.** Analysis of Human activity system. **Stadie 2.** Analysis of information – functions and entities. **Stadie 3.** Analysis and design of socio-technical aspects. **Stadie 4.** Design of human-computer interface. **Stadie 5.** Technical design.

<sup>17</sup> Se ovenfor om paradigmevalg.

en bredere forståelse for hele system- analyse og design processen. OOAD som vi kender fra 2. år, tager ikke hensyn til virksomhedens sociale konstruktion og de påvirkninger en systemforandring kan medføre, det gør Multiview.

De fem forskellige stadier i ”Multiview metoden” og de interne relationer mellem dem fremgår af nedenstående figur. Alle fem perspektiver er nødvendige, for at kunne forme et komplet (helt) system, såvel teknisk som menneskeligt. Ligesom et menneske også kun er helt, når det kender til både de menneskelige og de mere rationelle sider af sig selv.



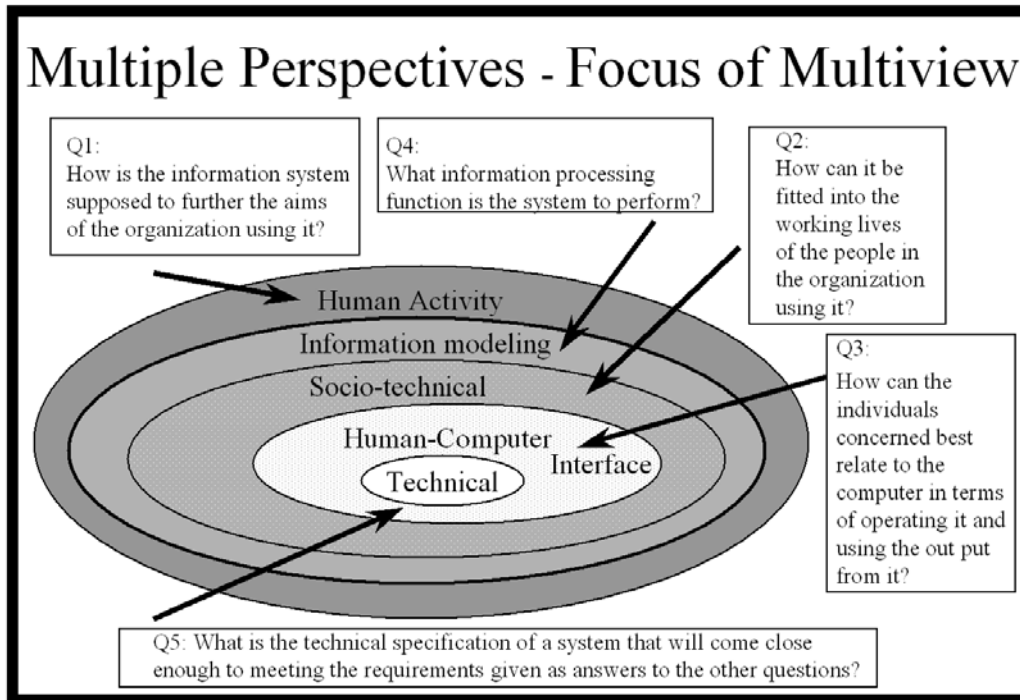
Copenhagen Business School, Karlheinz Kautz, DIS 3-4, Scandinavian ...

6

Figur 4 Strukturen i Multiview

Kasserne i figur 2 repræsenterer analysestadier og cirklerne designstadier. Fuldt optrukne pile illustrerer relationer og de stiplede pile beskriver output fra et givent stadie. Nogle outputs bruges på det efterfølgende stadie.

For at skaffe et hurtigt overblik over Multiview, findes en simplificeret udgave af modellen. Ved at besvare spørgsmålene på hver stadie i modellen beskrives den tilhørende aktivitet. Se figur 3 nedenfor.



Copenhagen Business School, Karlheinz Kautz, DIS 3-4, Scandinavian ...

7

Figur 5 Multiple perspektiver i Multiview metoden

Nedenfor beskrives aktiviteten på hver af de fem stadier mere dybdegående.

### Stadie 1. Analysis of Human activity system

Human activity er et generelt udtryk, der dækker over en hvilken som helst form for organisation. Det vil sige, et individ, en virksomhed, et department, en klub, en afdeling eller enhver der kunne overveje, at anvende en computer til nogle af deres informationssystemer.

På dette stadie er der fokus på, at bestemme virksomhedens ”Weltanschauung<sup>18</sup>”, hvilket danner grundlaget for den efterfølgende bestemmelse af virksomhedens systemkrav. Her må der ikke fejles, fordi verdensanskuelsen påvirker eller influerer resten af analysen og designet.

Softsystems metoden kaldes dette første stadie af Multiview. Det indeholder en model der er delt op i fire hovedemner, se figur 2.5 side 31 i kompendiet:

1. Erkend og forstå problemsituationen. Her produceres det rige billede. Pkt. 1-3<sup>19</sup>. Elementerne i det rige billede er, klienter af systemet, mennesker der er en del af systemet, opgaver der udføres, systemets omgivelser og opdragsgiveren af systemet. At konstruere et rigt billede er en subjektiv proces, hvor hovedopgaven er at give et holistisk billede af situationen.
2. Konstruer relevante systemmodeller. Her navngives relevante systemer, roddefinitioner for relevante systemer konstrueres analyser og korrigerer roddefinitionen med CATWOE og til sidst konstrueres og korrigeres den konceptuelle model. Pkt. 4-7. Flere relevante systemer bør udforskes og beskrives for at kunne beslutte, hvilket system der er mest brugbar. Metoden er her diskussion. Derefter bestem-

<sup>18</sup>”Weltanschauung” skal forstås som virksomhedens mission eller verdensanskuelse. Eks. Set ud fra en læges synspunkt: der skal behandles så mange patienter som muligt.

<sup>19</sup> Numrene refererer til substadierne i modellen figur 2.5 side 31 i kompendiet.

mer konsulenten og virksomheden i fællesskab, hvilket fokus og hvilket synspunkt der skal tages. Når der er taget kan roddefinitionen udvikles og revurderes. Til dette anvendes et kontrolkriterium - CATWOE - for at sikre, at alle roddefinitionens elementer er tilstede – ligesom vi kender det fra OOAD med FACTOR. Når konsulenten og virksomheden er kommet til enighed om, at roddefinitionen er veldefineret, konstrueres en konceptuel model som er en beskrivelse af systemet med ord.

3. Sammenlign systemmodellerne med virkeligheden. Det vil sige, at den udarbejdede konceptuelle model sammenlignes med virkeligheden som det er præsenteret i det rige billede. Pkt. 8
4. Vurder sammenligningen og implementer eventuelle ændringer. Pkt. 9-10

Indholdet af modellen bliver i dette kapitel kun skitseret, fordi formålet er at skabe overblik. En mere detaljeret gennemgang af aktiviteter på hvert stadie finder sted i de efterfølgende artikler. Dog skal det understreges, at der findes marginale forskelle i analysen af HAS i Multiview og den oprindelige SSM.

### **Stadie 2. Analysis of information – funktions and entities**

Formålet med dette stadie er, at analysere systemets før beskrevne entiteter og funktioner. Analysen bliver nu mere teknisk betonet, hvorfor situationen anskues fra et teknisk perspektiv. Udgangspunktet for analysen er roddefinitionen og den konceptuelle model fra foregående stadie. Metoden er struktureret analyse.

De to hovedaktiviteter på dette stadie er:

- udvikling af en funktionel model (funktionsmodel) og
- udvikling af en entitetsmodel (datamodel).

Værktøjer til at udarbejde disse modeller er, data flow diagrammer, procesbeskrivelser, E/R diagramm, data ordbog, tilstandsdiagrammer.

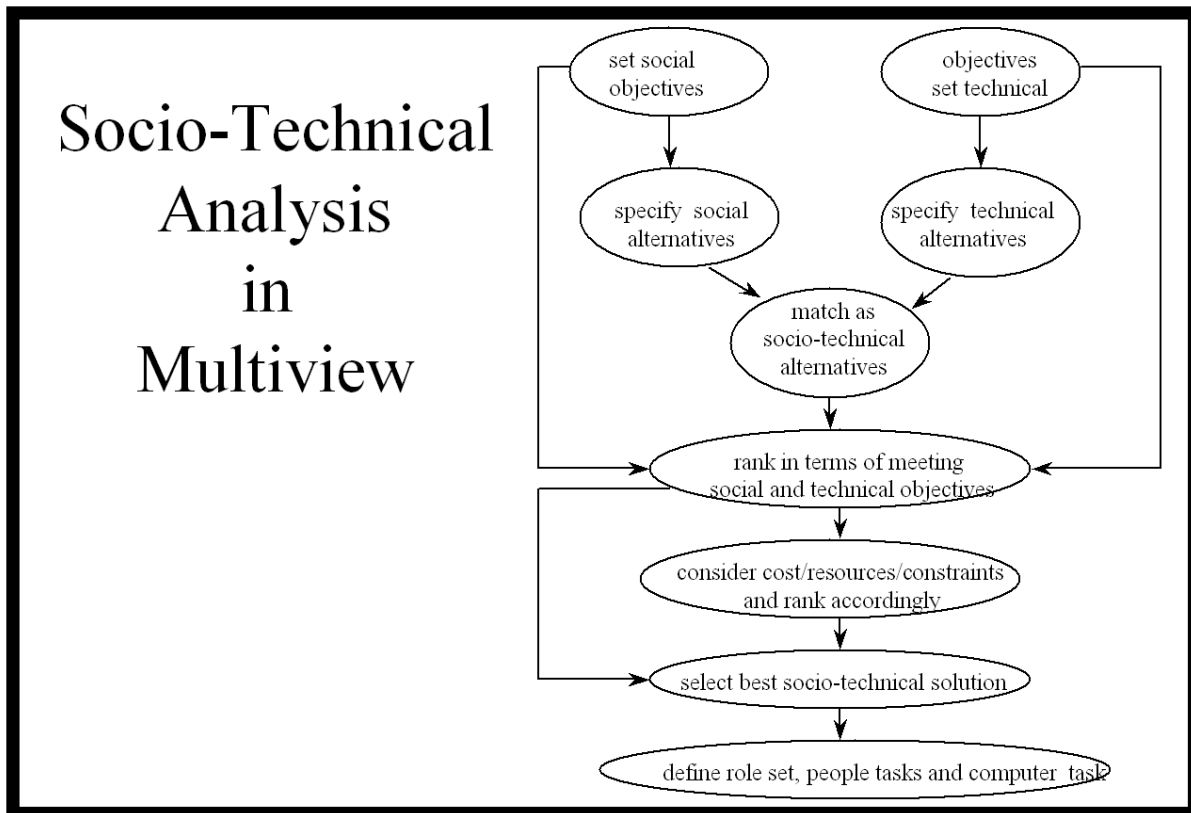
### **Stadie 3. Analysis and design of socio-technical aspects**

Hele essensen i Multiview er, at der tages hensyn til menneskelige værdier og deres behov. Arbejdsopgaven består derfor i, at udarbejde et design der tager højde for både de bløde værdier (menneskerne og deres behov samt omgivelserne de befinder sig i), og organisationens struktur, computersystemer og de nødvendige arbejdsopgaver der skal udføres.

Det helt centrale på dette stadie er, at identificere såvel sociale som tekniske alternativer der lever op til virksomhedens mål. Disse alternativer sammenholdes for at producere socio-tekniske alternativer. Den bedste socio-tekniske løsning findes derefter ved at rangere alternativerne, først efter hvorvidt de lever op til målet, og derefter kommer det økonomiske (omkostninger) aspekt ind i billedet.

Når den bedste løsning er bestemt, kan arbejdsopgaver for såvel computer som for menneskene i organisationen samt deres rolleset defineres. Dette er outputtet fra stadie 3.

Se nedenstående figur for en grafisk illustration af den socio-tekniske analyse.



Copenhagen Business School, Karlheinz Kautz, DIS 3-4, Scandinavian ...

12



Figur 6 Socio-technical analysis

#### Stadie 4. Design of Human-Computer Interface

Nu er det tid til at udvikle interfacet. Molich og Lauesen kommer derfor på banen igen med menneske/maskine interaktion, som vi kender fra andet år. Dette emne er ikke aktuelt eller en del af pensum i år. Inputtet til arbejdsopgaverne på dette stadie er outputtet fra stadie to og tre.

#### Stadie 5. Technical design

Inputtet til sidste stadie af Multiview er entitetsmodellen fra stadie to og de tekniske specifikationer fra foregående stadie. Det er heller ikke en del af dette kursus, men der henvises til OOAD.

## 5. PARTICIPATION AND SYSTEMS DESIGN.

Pensum: Mumford, effective Systems Design and Requirements Analysis pp. 12-26 (DIS kompendium S. 36-43).

---

### 5.1. Mumford

En vigtig del af ETHICS er det engelske ord Participation – oversat til dansk: deltagelse. I artiklen beskrives hvad Participation betyder og hvad det indebærer, for at forstå hvorfor der bør være fokus på deltagelse i systemdesign.

Definitionen på Participation er:

*En proces hvor to eller flere parter informerer og påvirker hinanden i planlægning, politik, specifikationer eller beslutningstagning.*

Det er begrænset til at være beslutninger som har fremtidig effekt, på dem der tager beslutningen eller dem der er repræsenteret af beslutningerne.

Ganske enkelt drejer det sig om at komme til orde, deltage i fællesskabet eller at være en del af demokratiet. Deltagelse kan som vi kender det fra demokratiet, enten foregå ved blot at sige sin mening eller det kan foregå ved et formelt valg med stemmeret. Det er blot én måde at gøre sin holdning offentlig bekendt

Spørgsmålet der bliver belyst, er hvor og hvornår det er fornuftigt, at have en demokratisk beslutningsproces i projektarbejde og systemdesign. I resten af teksten anvendes ordet deltagelse og refererer til ovenstående definition på Participation.

Ovenstående afsnit har været en generel beskrivelse af Participation, hvor formålet i resten af teksten er at relatere det til projektarbejde og systemdesign. Hele snakken om deltagelse skal ses i forhold til forandringer i organisationen. Det vil sige, de tiltag der automatisk følger implementeringen af nye systemer.

#### Hvorfor deltagelse?

Der er forskellige interesser og motivation for deltagelse i projektarbejde afhængig af synsvinkel eller position i organisationen. Nogle ledere mener, at det moralsk set er mere rigtigt at lade folk deltage og dermed have medbestemmelse over deres egen fremtid. Men det er ikke eneste årsag til, at ledelsen er motiveret for at få medarbejderne til deltagelse. Deltagelse kan også medvirke til, at virksomhedens mål opnås mere effektivt fordi nogle eksperter måske er med til at tage beslutninger.

En ledelse kan også opfordre til deltagelse med negative hensigter. I sådanne tilfælde anvendes deltagelsen som et manipulerende værktøj til at overtale projektdeltagere, som ellers ville have forkastet forslaget.

Medarbejderne har ligeledes forskellige grunde til at deltage. Blandt andet kan det være for at beskytte sig mod uønskede tiltag i organisationen. Deres medbestemmelse kan også lede til mere interessante jobs, eller det kan medvirke til forbedrede arbejdsvilkår.

### **Deltagelsesstruktur - indhold og proces**

Alle interessentgrupper skal have en fornemmelse af at alle deltagere/medlemmer blive tilgodeset eller at alle får nogle fordele. Det er fuldstændig det samme der sker ved politiske udvælgelsesprocesser. Participation er altså et produkt af et institutionaliseret politisk system, med demokratisk udvalgte repræsentanter eller et stemmesystem. Man får mest igennem, ved at have samme interesser eller holdninger i én gruppe.

Hvor der er ens magt, men forskellige interesser kan man tale om at tilstræbe en meta-løsning. Det betyder at alle interessenters målsætning bliver opfyldt.

Når der tages stilling til en deltagelsesstruktur handler det om at vælge en struktur der passer til den grad af influering en ledelse vil tillade. Eller det udsnit af virksomheden der skal vises. Ser vi på projektarbejdet handler det i bund og grund om at argumentere for, hvem der skal interviewes eller afprøve prototypen og hvorfor. Her tror jeg også at de velkendte teorier om data- og informationsindsamling fra Ib Andersen i fællesskab med artiklens emne gør sig gældende. Det vil sige, undersøgelsesdesignet.

Deltagelse anvendt i systemudviklingsopgaver involverer en proces over tid og ikke så meget overleveringen af information eller en mening på et bestemt tidspunkt i tid.

### **Muligheder for at anvende deltagelse**

Når det at deltage, ses som en proces der involverer delt problemløsning (mellem flere parter), over længere tid, er der adskillige muligheder at vælge i mellem.

1. Skal deltagelse være en ret eller noget der skal tillades?
2. Hvornår i projektforsløbet skal deltagere involveres i beslutningstagning?
3. Hvem skal deltage og involveres i processen?
4. Hvordan skal deltagerne udvælges?

Der er ikke nogen entydige svar, men det er spørgsmål der skal overvejes og der skal argumenteres. Som Karl Heinz ville have sagt, er alt på femte semester tilladt så længe argumenterne er i orden. Små men vigtige beslutninger tages på alle stadier af systemudvikling.

### **Forskellige former for deltagelse**

System design kan finde sted på tre forskellige organisatoriske niveauer. Toppen hvor det handler om strategisk planlægning, midten hvor det er taktisk og nederst hvor det er operationelt. Participation approach kan anvendes på alle niveauer.

Consultative participation– er den mest passende tilgang såfremt der ønskes enighed på det strategiske niveau. Ansatte kan under denne form bidrage med idéer til designprocessen, men den endelige løsning forberedes af en specialistgruppe og den endelige beslutning tages af topledelsen.

Representative participation– Når det valgte projekt påbegyndes kan der være mange interessenter der ønsker at influere. Der er her mulighed for en repræsentativ tilgang. Det er en form hvor udvalgte brugere og udviklere udvælger en designgruppe der enten alene eller sammen med ledelsen tager beslutningen. Især i startfasen af systemdesignet er denne tilgang anvendelig fordi mange forskellige har interesse i systemdesignet. Det giver selvsagt et repræsentativt billede af organisationens interesser. Progressive virksomheder er tilhængere af denne form.

Consensus participation – er en form hvor en designgruppe forbereder alternativer og præsenterer dem, men alle i organisationen er med til at tage designbeslutningen – en slags accept beslutning. Flertallet vinder.

I virksomheder med en flad struktur kan udgangspunktet være en konsensus form, men i store virksomheder vil denne form aldrig kunne lade sig gøre.

### **Problemer med at anvende en participative design tilgang**

Alle forandringsstrategier har deres problemer med deltagelse også systemdesign, hvor deltagelse indrages. Nedenfor nævnes nogle af forfatterens erfaringer hvor deltagelse har givet problemer.

Der kan være problemer forbundet med etablering af tillid, eller hvordan designgruppen skal findes ved election vs. selection. Der kan være interessekonflikter afhængig af hvilken gruppe eller afdeling man tilhører. Man kan blive udsat for pres uden at være vant til det i en deltagende designgruppe. Nogle systemdesignere kan føle sig truet på deres autoritet og status fordi de ved at vælge en designgruppe får en rolle som konsulenter. Idéelt set bør managers have tæt tilknytning til designgruppen, men det er ikke altid muligt. Derfor øges rapporteringsbehovet væsentligt fordi lederen hele tiden skal opdateres.

I designgruppen bør der være en ”facilitator” (proceskonsulent) der har neutral status i forhold til projektet og ledelsen. Opgaven er at hjælpe designgruppen med at styre projektet, ved blandt andet at lære dem metoden. Derudover er vedkommendes opgave at motivere og udrede eventuelle konflikter.

### **Generelt om miljøet i en designgruppe**

Som vi kender det fra første år, er kultur et vigtigt begreb der ikke må negligeres. Alle grupper er forskellige og kommunikation tager forskellig form i forskellige lande eller forskellige afdelinger. Det kan være årsag til mange misforståelser, hvorfor ”the facilitator” hurtig skal accepteres af designgruppen.

En designgruppe går gennem forskellige psykologiske stadier i projektførelsen, i starten er alle motiverede, men til sidst kræver det sin mand at bevare koncentrationen. Også her er det ”facilitatorens” opgave at holde gruppens moral høj. De forskellige stadier er:

- Bekymring og usikkerhed
- Så er der en stigning i gruppens tillid og moralen er høj
- Herefter kan der være en snert af demoralisering i analysefasen på grund af problemer
- Til sidst hvor projektet går ind i designfasen og stiger moralen ofte igen og holder projektet ud

En facilitator kan sammenlignes med en projektleder der holder alle trådene sammen og ved hvor han har medarbejderne i gruppen.

### **Konklusionen på deltagelse**

Succesfuld forandring i en organisation kræver identificering og opløsning af interessekonflikter. Interessekonflikterne findes alle steder i organisationen, mellem personale med forskelligt ansvar og forskellige funktioner. Den idéelle løsning på problemet er, når alle parter tilgodeses. Denne deltagende tilgang er blot én acceptabel måde at gennemføre en forandringsproces på, men som også beskrevet ikke uden problemer.

Argumenter mod at anvende en Participative tilgang er, at det er dyrt og tidskrævende. Men erfaringer viser, at det samlede antal manddage er nogenlunde det samme, fordi implementeringen er nemmere og hurtigere.

### **Perspektivering**

Deltagelsesstruktur afhænger naturligvis også af de helt grundlæggende holdninger der er bestemmende for eksempelvis forandringsstrategien. Er der en meget objektiv og rationel indgangsvinkel til systemudviklingen er der måske ikke så meget fokus på at inddrage brugere.

## **6. REVIEWS AND TECHNICAL WALKTHROUGHS.**

Pensum: Andersen et.al., Professional Systems Development - Experience, Ideas and Action. pp. 137-142 (DIS kompendium S. 44-48).

## **7. FORMAL TECHNICAL REVIEW.**

Pensum: Pressman, Software Engineering - A Practitioner's Approach. pp. 562-570 (DIS kompendium S. 49-57).

---

### **7.1. Andersen et. al. Og Pressman**

Formålet med denne artikel er, at skitsere nogle guidelines for hvorledes et reviewmøde bør afholdes, og hvordan evalueringen bliver mest effektiv.

Formålet med et review er, på en struktureret og effektiv måde at evaluere et produkt. Et review er derfor blot et møde, hvor kvaliteten af produktet evalueres. Under reviewet er det vigtigt at bevare fokus, det betyder at problemer skal identificeres og udpeges, men ikke løses.

Der skelnes mellem et "dokument review" (DR) og et "Formal Technical Review" (FTR). Den store forskel ligger i produktet og dermed, hvad der bliver reviewet.

Hvad er et produkt? Én produkttype kan være et analyse- eller designdokument, en kravspecifikation, en projektplan, eller brugermanualer. Til evaluering af disse produkter afholdes et DR. En anden produkttype kan være software. Det vil sige, en stump eller helt færdig kildekode. Til evaluering af softwareprodukter knytter der sig et FTR.

Rammerne for reviewet er maks. 3 til 7 reviewere med teknisk indsigt i produktet og en tidsbegrænsning på maks. 2 timer. Er to timer ikke tilstrækkeligt, arrangeres flere reviews. Deltagerne i reviewgruppen må selvsagt ikke være projektleder eller på anden måde have relation til det udarbejdede produkt. Inden mødet påbegyndes bør produktet sammenlignes med en eventuel kravspecifikation.

Formålet med et DR er:

1. at udpege eventuelle behov for dokumentforbedringer
2. at godkende (forkaste) produktet
3. at skabe et godt produkt med høj kvalitet
4. at uddanne, oplære eller træne

Formålet med et FTR er:

1. at identificere funktions-, logiske-, eller implementeringsfejl
2. at verificere at softwaren lever op til dets krav
3. at sikre at softwaren lever op til definerede standarder
4. at sikre at softwaren fuldendes i en ensartet form
5. at gøre projekter nemmere at lede

FTR er med til at sikre en høj kvalitet af et software produkt, hvilket er en del af pensum i IT-ledelse (Kvalitetssikring). Til et FTR møde anvendes forskellige metoder, blandt andet Technical Walkthroughs (se nedenfor), inspections (IT-ledelse 2. år) og round robin reviews (IT-ledelse 2. år).

Ved begge typer af reviews opnås det bedste resultat, ved en nøje planlagt og struktureret agenda. Derfor har Reviewerne forskellige roller. Der skal være en review leder, der er ansvarlig for gennemførelsen af reviewet, samt står for tjeklister, distribution osv.. Det vil sige, at vedkommende før mødet distribuerer relevant materiale og er en form for ordstyrer ved selve mødet.

Reviewerne (ca. to personer) er forberedte og er dem, der har styr på indholdet af produktet i detaljer. Med andre ord, kritiserer de produktet, både positiv som negativ kritik.

Derudover skal der være en rapporter til stede. Vedkommende har ansvaret for at notere undervejs. Til denne opgave er det ikke et krav, at være udenforstående, vedkommende kan derfor godt være projekt-deltager.

Og endeligt skal dem der har lavet produktet være tilstede. Deres eneste opgave under reviewet er, at give deltagerne en introduktion til produktet og besvare opklarende spørgsmål.

Review rapportering finder efterfølgende sted, ved at en reviewrapport udfyldes. I hovedtræk minder den om vores projektdagbogsdokument (kompendium side 51).

**Technical walkthroughs:** Den væsentligste forskel på reviews og Technical Walkthroughs er, at det udelukkende er projektmedlemmerne der deltager i Technical Walkthroughs. Eventuelt som opfølgning på et review, hvor produktet er blevet kritiseret. Produktet bliver præsenteret for de andre projektdeltagere, med henblik på, at opnå et bedre kendskab og indsigt i produktets helhed. Målet er som ved reviews at få kritik, med den undtagelse, at nye løsningsforslag er velkomne.

Eksempler på review- og Technical Walkthroughs lister findes på side 47 og 48 i kompendiet. FTR tjeklister findes på side 54 – 57 i kompendiet.

For begge møder anbefales det, at udarbejde egne metoder og tjeklister.

## 8. HUMAN ACTIVITY SYSTEM & SOFT SYSTEM METHODOLOGY

Dette afsnit vil behandle den første af de fem faser i multiview metoden: analyse af human activity system (herefter HAS).

Indledningsvis vil det blive gennemgået, hvad et HAS er, og derefter hvordan man kan analysere det ved hjælp af soft system metoden SSM.

---

### 8.1. Hvad er et human activity system?

Et HAS er en begrebsramme for beskrivelse af organisationer, det væsentlige er måden, hvorpå man betragter organisationen. En organisation skal ses som et system<sup>20</sup> bestående af mennesker, der søger at opfylde et overordnet formål<sup>21</sup>. Dette formål, eller verdensanskuelsen<sup>22</sup>, er væsentlig i den videre analyse af systemet, da det ofte vil være eksistensberettigelsen for det pågældende system.

I forhold til den videre analyse er der en række karakteristika ved organisationer, som får betydning i forhold til en HAS synsvinkel<sup>23</sup>:

- De har et formål – der gennemføres en form for transformation. Eksempelvis uddannelse af elever, bearbejdning af råvarer, behandling af patienter m.m. Det er her væsentligt at se relationer mellem verdensanskuelsen og transformationen.
- Det er muligt at måle effektiviteten af transformationen. Eksempelvis kan mængden eller kvaliteten af output måles, og det er muligt at vurdere, om det er tilfredsstillende eller ej.
- Der eksisterer en kontrolmekanisme. Dette vil ofte være ledelsen, men man skal være opmærksom på, at dette kan være en integreret del af organisationens opbygning.
- De består af del-komponenter, der hver især kan betragtes som et selvstændigt system. Her kan der være tale om de traditionelle afdelinger, men uformelle systemer kan være lige så væsentlige for analytikeren.
- Del-komponenterne er relaterede og de interagerer.
- Systemet indgår som en del af en større helhed. Dette er væsentligt, da det betyder, at man ikke kan betragte noget isoleret uden skelen til relationerne til omverdenen.
- Der er en grænse, der definerer, hvad der er en del af systemet, og hvad der ikke er. Denne grænse defineres af analytikerne, i samråd med de interessenter de prioriterer. Hvad der skal betragtes som en del af systemet, og hvad der skal betragtes som omverden bliver dermed tæt forbundet med analysegruppens valg af paradigme.
- Systemet har sine egne ressourcer. Eksempelvis medarbejdere, penge, bygninger m.m.
- Der er en forventning om kontinuitet i systemet.

Formålet med at analysere HAS, inden man går i gang med de næste faser af Multiview metoden er at sikre, at man har organisationens formål i fokus, når man arbejder med design af systemer - eksempelvis til løsning af konflikter.

Analysen af HAS gennemføres ved hjælp af soft system metoden, der er gennemgået nedenfor.

---

<sup>20</sup> Der er ikke tale om nogen form for IT, når der diskuteres systemer. Begrebet forklares nærmere i gennemgangen af soft system metoden.

<sup>21</sup> Avison & Wood-Harper – DIS kompendium S. 30

<sup>22</sup> Verdensanskuelse er løst oversat fra "Weltanschauung". Oversættelsen er ikke dækkende, da det oprindelige begreb rummer fortolkninger som "grundlæggende antagelser", "overordnet formål", "eksistensgrundlag" m.m. Herefter dækker begrebet verdensanskuelse alle disse fortolkninger. (frit fortolket fra undervisningen – ano)

<sup>23</sup> Undervisningsplancher til lektion 7 & 8. Karlheinz Kautz

## 8.2. Soft System metoden

Pensum: Checkland & Scholes, kap 2 S. 13-27 (DIS kompendium S. 71-78)

Soft system metoden kan på et lidt højere abstraktionsniveau ses som to forskellige ting. Dels er det en fastlagt metodisk indgangsvinkel til analyse af HAS, og dels er det en tankegang – eller verdensanskuelse om man vil – hvormed man betragter organisationer, konflikter, aktiviteter m.m.

Dette afsnit vil indledningsvis prøve at beskrive nogle grundelementer i denne tankegang. Derefter gennemgås de 7 trin i ”opskriften” til soft system analyse.

### 8.2.1. Soft system thinking

Soft system tankegangen udspringer af et ønske om at operationalisere og systematisere ledelses beslutninger. Dette udgangspunkt havde meget ringe succes, da det hurtigt viste sig, at problemløsning skulle ses i konfliktens sammenhæng og dermed ikke som en standard problematik. Dette resulterede i, at fokus flyttede sig fra operationalisering af beslutningsprocessen til en form, hvor fokus er på processer i stedet for resultater og læring i stedet for optimering<sup>24</sup>.

Soft system tankegangen går ud på at betragte ting i deres sammenhæng<sup>25</sup>. Det giver en meget subjektiv vurdering af problemer, og hænger sammen med en grundlæggende antagelse om, at problemer i organisationer er ustrukturerede og betragtes forskelligt afhængigt af, hvem der betragter dem. HAS er begrebsrammen for den helhedsopfattelse, man anlægger når man er soft system analytiker. Det er vigtigt at man ikke betragter HAS som om de findes<sup>26</sup> – de er kun billeder af verden, som kan bruges til beskrivelse og sammenligning. Dette er en vigtig pointe, fordi det understreger, at man ikke kan lave en objektiv beskrivelse af verden, når man er soft system analytiker – alt er mere eller mindre subjektivt.

Dette gør, at soft system tankegangen står i skarp kontrast til hard system tankegangen, hvor alt netop er objektivt og med mulighed for en endelig og korrekt løsning. Ved soft system tankegangen er det ikke muligt at finde en korrekt løsning på en problemsituation<sup>27</sup>, men man kan finde en løsning, der tilfredsstillende en eller flere individer.

### 8.2.2. Soft system methodology

SSM er en metodisk indgangsvinkel til analyse af HAS. Fremgangsmåden er illustreret i figuren nedenfor. Figuren skal opfattes således, at man i nogle faser arbejder i den verden man er i færd med at beskrive<sup>28</sup>, i andre faser er man i en analytiker verden, hvor man kun fokuserer på de ting, man kan *forestille* sig kan gøre en forskel i organisationen<sup>29</sup>.

Denne fremgangsmåde har betydning for, hvorledes beslutningsprocessen i de enkelte faser kommer til udtryk. Når man arbejder i organisationen, skal man nøje overveje, hvilke interessenter der har indflydelse på resultaterne<sup>30</sup>. Når man arbejder for sig selv eller i en projektgruppe, er det alene projektgruppens egne tanker og konklusioner, der er relevante.

<sup>24</sup> DIS kompendium S. 71-72

<sup>25</sup> Dette kaldes også en holistisk indgangsvinkel. (Holistisk udledt af ’holon’, der er et opfundet ord for en beskrivelse af en helhedsopfattelse. DIS kompendium S. 76)

<sup>26</sup> DIS kompendium S. 76

<sup>27</sup> Dette skal ikke forveksles med en problemformulering eller et problem, da disse begreber knytter sig til hard system tankegangen.

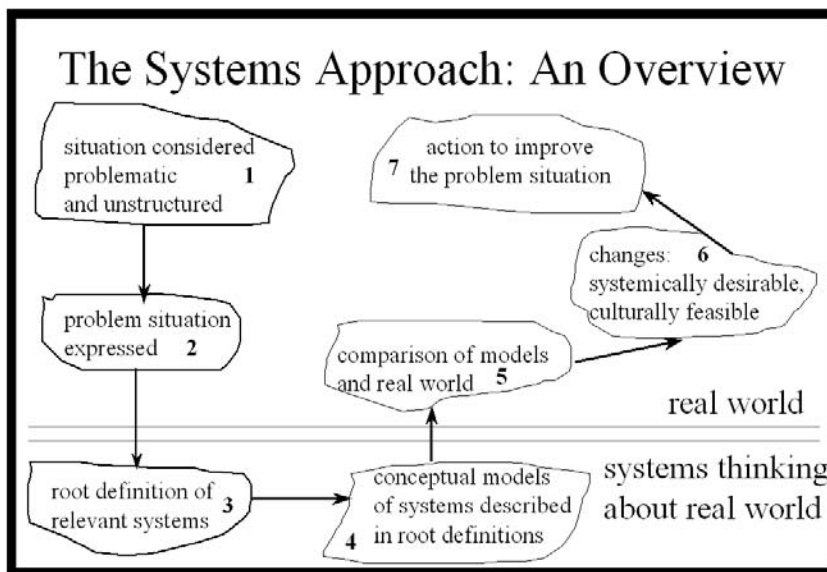
<sup>28</sup> Faserne 1-2, 5-7

<sup>29</sup> Faserne 3-4

<sup>30</sup> Dette hænger sammen med projektgruppens valg af paradigme, da dette er styrende for hvis side man ser tingene fra.

Metoden er bygget op over disse 7 faser – faserne 2-7 er beskrevet mere uddybende nedenfor:

1. Den første fase er en erkendelsesfase, hvor man står med en situation, der betragtes som problematisk og ustruktureret.
2. Anden fase går ud på at beskrive problemsituationen. I erkendelsen af at problemsituationen er ustruktureret, nytter det ikke noget at finde en egentlig problemformulering – dette ville være fremgangsmåden ved hard system tankegangen. I stedet beskriver man problemsituationen med et rigt billede.
3. I tredje fase går man ind i analytiker verdenen og begynder at beskrive systemer, man mener kan gøre en forskel i organisationen. Disse beskrivelser kaldes rod definitioner.
4. Fjerde fase er en beskrivelse af de enkelte rod definitioner ved hjælp af illustrationer af input og output, samt faser i systemerne.
5. Femte fase er i organisationen igen, hvor de enkelte rod definitioner sammenholdes med den virkelige verden. Her kan det eksempelvis vise sig, at der er nogle af de tænkte løsninger, der af praktiske årsager ikke kan bruges, eller som af sociale årsager ikke vil kunne implementeres.
6. I sjette fase vurderes om forslagene til ændringerne er realiserbare. Her finder analytikerne ud af om de anbefalede ændringer er systemisk ønskelig og kulturelt gennemførlig. Se nedenfor under kulturanalysens konklusion.
7. Sæt i værk!



Copenhagen Business School, Karlheinz Kautz, DIS7-8, Systems Approach - Rich Pictures ...

6

Figur 7 The systems approach

### 8.2.3. Rigt billede

Pensum: Avison & Fitzgerald S. 110-117 (DIS kompendium S. 67-70)

Udarbejdelse af et rigt billede er en måde, hvorpå man kan illustrere problemsituationen i en organisation. Metoden er god i forbindelse med SSM, fordi man her kan udtrykke interessenternes subjektive vurderinger og bekymringer med illustrationer, uden nødvendigvis at skulle formulere dem som et egentligt problem.

Der er som udgangspunkt ingen regler i forbindelse med udarbejdelsen af billedet, men det er væsentligt, at det fremstår intuitivt forståeligt, samt at det bidrager til overblik over situationen<sup>31</sup>. I forbindelse hermed er det en god idé at præsentere det vigtigste i midten, da dette ofte vil blive opfattet som omdrejningspunktet, af dem der ser på billedet. Nedenfor er givet en række råd i forbindelse med udarbejdelsen, som kan være med til at gøre billedet bedre anvendeligt i den videre analyse.

Det bør klart fremgå af billedet, hvad der er organisationens primære opgave. Dette skal sikre at den videre analyse er målrettet mod at imødekomme og hjælpe til opnåelse af dette mål.

Der bør være en klar grænse i billedet, der illustrerer, hvor grænsen går mellem det system, man er i færd med at beskrive, og omverdenen. Dette medvirker til at skabe overblik over, hvor eventuelle problemer opstår – er det udenfor eller indenfor systemet.

Billedet bør indeholde de aktører, der er relevante i forbindelse med beskrivelsen af den pågældende situation. Det er imidlertid ikke nok kun at beskrive de faste organisatoriske dele af organisationen – man skal også inddrage de sociale roller, som aktørerne mener er vigtige, eller som de bekymrer sig om. Arbejdsprocesser, kommunikationsflow m.m. bør også indgå, idet man dog holder for øje, at det ikke skal udvikle sig til et dataflow diagram – det vil være en misforståelse af opgaven.

Valg af symboler er helt frit, men man skal sørge for, at der er konsistens i anvendelsen af symboler, så man ikke anvender tre forskellige symboler til at illustrere en konflikt, hvis det samme kunne bruges i alle tilfælde.

Vægten mellem tekst og billeder afhænger af den pågældende situation.

Hvis ikke man kan tegne hele det rige billede på et stykke papir, kan man vælge at tegne flere rige billeder med forskellig detaljeringsgrad.

Da forskellige interessenter i systemet sandsynligvis opfatter situationen forskelligt, er det væsentligt, at man som analytiker beslutter sig for, hvilke ting man vælger at inddrage hvordan. Dette betyder ikke, at man skal lave en entydig fremstilling, hvor man kun inddrager en interessant – det vil være en total misforståelse. Man skal blot ikke forvente, at man kan få alle til at acceptere det rige billede man udarbejder. Her må man som analytiker prøve at sikre sig, at man har de forskellige konflikter med, og så sætte sig ud over om de enkelte er enige i, at det er en rigtig fremstilling af lige netop deres opfattelse.

#### **8.2.4. Kulturanalyse**

Pensum: The stream of cultural enquiry, Checkland & Scholes S. 44-53 (DIS kompendium S.83-87)

Dette afsnit handler om analysen af den virksomhedskultur, som et fremtidigt system skal indgå i. Kulturanalysen fremgår ikke klart af ”standard” modellen for SSM. Det er imidlertid vigtigt at få den med, da den kan have afgørende betydning for et fremtidigt system.

Kultur analysen bør startes op sideløbende med udarbejdelsen af det rige billede, men den stopper i princippet ikke før det endelige produkt er færdigt.

Kulturanalysen består af tre dele – de er hver især gennemgået mere indgående i afsnittene nedenfor:

- *Analyse af indgriben.* Her er der tale om at analysere den indgriben man selv som analytiker er ved at foretage i problemsituationen.

---

<sup>31</sup> DIS kompendium S. 67

- ”Social system” analyse. En analyse af roller, normer og værdier.
- ”Political system” analyse. En analyse af beslutningsprocesser og magt.

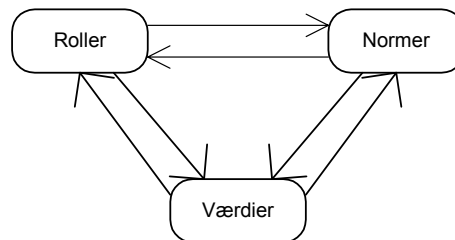
#### Analyse af indgriben

I denne fase skal der identificeres tre forskellige roller i forbindelse med det projekt man er i gang med:

- *Klient*. Her er der tale om at identificere hvem der har iværksat analyseprocessen. I forbindelse med denne del, er det væsentligt at have fokus rettet mod vedkommendes motiv for denne iværksættelse. Der kunne eksempelvis være tale om en leder med en skjult dagsorden, som man skal passe på at man ikke lader sig styre af.
- *”would-be problem solver”*. Her er der tale om at identificere alle dem, der er villige til at putte resourcer i arbejdet med at forbedre problemsituationen. Denne gruppe kan være sammenfaldende med klient gruppen.
- *Problemejer*. Set fra problemsituationen vil der som udgangspunkt ikke være nogen egentlig problemejer. Det er derfor analytikernes opgave at kortlægge, hvem der er de egentlige problemejere. Denne liste kan laves således, at den som udgangspunkt indeholder klienterne, og dem som er villige til at gøre noget ved problemsituationen. Listen bør dog ikke begrænses til disse grupper. Problemejer-listen er et godt udgangspunkt når man skal til at definere relevante systemer i problemsituationen.

#### ”Social system” analyse

Denne analyse går ud på at analyse det sociale miljø, der er en del af problemsituationen. Denne analyse kan gennemføres ved hjælp af nedenstående model, der er en stærk simplificering af virkeligheden, men som dog trods alt er anvendelig.



**Figur 8 Model til analyse af det sociale system**

Modellen skal forstås således, at det sociale miljø består af disse tre faktorer, der hver især påvirker og definerer hinanden.

Rolle defineres som en social position, der opfattes som væsentlig af aktørerne i problemsituationen. Denne position kan være defineret formelt, eller opstå som uformel rolle.

En rolle karakteriseres af de forventninger, der knyttes til den – dette udtrykkes i normer.

En rolle vurderes på baggrund af en række værdier, der kan indeholde opfattelser omkring menneskelig adfærd, som værende godt eller dårligt.

Modellen er god, hvis man husker at man ikke får et komplet eller statisk billede af det sociale miljø.

Selve gennemførelsen af analysen kan ikke gøres som en direkte forespørgsel på de konkrete begreber eller deres indhold. Analytikeren er nødt til at have denne analyse i baghovedet hver gang der foretages dataindsamling i organisationen. Der er således tale om data, der skal observeres og udledes af andre data, da den direkte forespørgsel sandsynligvis vil føre til svar der indeholder de officielle myter omkring det pågældende spørgsmål. Det vil i praksis sige, at folk svarer det de ønsker at situationen skal være, eller det de officielt taler om og dermed ikke det reelle svar.

#### ”Political system” analyse

Denne analyse går ud på at finde frem til, hvorledes forskellige interessenter med forskellige holdninger opnår enighed. Dette er i praksis en analyse af beslutningsprocesser og magt.

En indgangsvinkel til at få gang i analysen kan være at spørge til hvordan magt kommer til udtryk, hvilke virkemidler man har som leder og hvordan man anskaffer sig disse virkemidler. Begrebet virkemidler knytter sig primært til formelle magtredskaber.

Ved at gennemføre en magtanalyse, vil man sandsynligvis kun få den virkelighed at se som de enkelte interessenter ønsker man skal se. Det er derfor nødvendigt at kombinere interview med observation, ligesom spørgsmål der ikke direkte vedrører magt kan anvendes. Disse spørgsmål kan bruges til at skabe et billede af den uformelle magt, som muligvis ikke kommer frem ved almindelige spørgsmål. Den uformelle magt kan være knyttet til respekt for erfaring, uformelle roller m.m.

Man skal være opmærksom på at hvis man ekspliciterer de uformelle magtstrukturer kan man være årsag til at de ændres eller manifesteres yderligere. I forlængelse heraf skal man nøje overveje, hvilke dele af en magt analyse man vælger at vise til hvem.

#### Konklusion på kulturanalyse

I SSM tankegangen er det ikke muligt at lave et rigtigt HAS. Derfor er det heller ikke muligt at lave en forbedring, der er korrekt. De parametre man kan måle på er i stedet, hvorvidt en ændring er *systemically desirable* og *culturally feasible*.

Den første parameter kan oversættes til hvorvidt aktørerne i det pågældende HAS opfatter en ændring som relevant – dette kan der argumenteres for og imod, og det bliver således mængden og vægten af de enkelte argumenter, der kan bruges til denne vurdering.

Den kulturelle accept, hænger ikke sammen med graden af ændring. Den hænger alene sammen med om den pågældende ændring opfattes som meningsfuld i den pågældende organisation. Gør den ikke det, vil den ikke blive gennemført.

Disse parametre udtrykkes i teksten som kernen i SSM. Risikoen er at bøje dem så de i stedet bliver til et spørgsmål om at finde ud af om en ændring er *systemically feasible* og *culturally desirable* – Altså om ændring er mulig at gennemføre og ønsket af organisationen. Denne fælde hedder hard system thinking.

### **8.2.5. Rod definitioner.**

Pensum: Avison & Fitzgerald S. 117-122 (DIS kompendium S. 79-82)

I denne fase skal analytikerne beskrive relevante systemer, der kan medvirke til opfyldelse af organisationens primære opgave. Beskrivelsen, der er verbal, skal kort beskrive det allermost væsentlige ved et system, der kunne medvirke til en forbedring af problemsituationen.

Hvilke systemer der betragtes som relevante, er en vurdering som skal gøres af analytikerne i den enkelte situation. Det er blot væsentligt, at man sikrer konsistens i beslutningerne, så man ikke anlægger forskellige kriterier forskellige steder i de enkelte faser.

Det er væsentligt at understrege, at der ikke længere er tale om beskrivelse af den virkelige verden, men derimod beskrivelse af nogle systemer udtænkt af analytikerne med henblik på fremtidig implementering. Det er altså ikke systemer som allerede eksisterer!

Med henblik på at sikre at rod definitionen er komplet, kan man anvende CATWOE kriteriet som en kontrol instans. CATWOE beskriver de ting, der bør være med i en rod definition. Nedenstående bør således kunne udledes af enhver definition, hvis den skal betragtes som komplet<sup>32</sup>:

- **Client.** Hvem får gavn af systemet (positivt såvel som negativt)
- **Actor.** Den som anvender systemet til at gennemføre en transformation.
- **Transformation.** Den forandring systemet frembringer. Dette er kernen i definitionen, da systemet ikke har nogen berettigelse uden der sker en form for ændring.
- **Weltanschauung.** Dette er den verdensanskuelse, der gør at systemet giver mening.
- **Owner.** Dette er den eller dem der ejer systemet og dermed er i stand til at stoppe frembringelsen af systemet, eller som i kraft af deres ejerskab kan bestemme udformningen.
- **Environment.** Omgivelserne som systemet indgår i. Dette hænger nøje sammen med det som i det rige billede er placeret udenfor det aktuelle HAS.

Når man skal lave sin rod definition, vil det ofte være en god idé at udarbejde en CATWOE beskrivelse først. CATWOE beskrivelsen og organisationens primære opgaver er de vigtigste redskaber til at målrette definitionen. Hvis man får flere forskellige CATWOE beskrivelser, fordi man betragter situationen fra forskellige synsvinkler, skal man udarbejde flere forskellige rod definitioner. Dette kan eksempelvis være tilfældet, hvis man betragter situationen set fra forskellige interessenters side. Derved kan man komme i en situation, hvor man har forskellige verdensanskuelser, eller forskellig opfattelse af, hvilken transformation der skal finde sted.

Det kan også være, at man har identificeret flere konflikter i problemsituationen, og ønsker at definere systemer til hver af disse konflikter.

Når man udarbejder en rod definition, er det vigtigt, at man udtrykker sig kort, klart og entydigt. Som tommelfinger regel, kan man sige, at hvis ikke man kan skrive rod definitionen i en sætning, har man ikke gjort klart for sig selv, nøjagtigt hvad systemet skal kunne.

En hjælp til selve udarbejdelsen af rod definition kan være at lave en sætning, der er udarbejdet således<sup>33</sup>:

*Et system ejet af [indsæt ejer] til at [Indsæt hvad systemet skal gøre], ved hjælp af [indsæt hvad systemet har til rådighed], med henblik på at [Skriv den primære opgave systemet bidrager til at opnå].*

Godkendelsen af definitionen, vil ofte være svær, da de forskellige interessenter har deres egen opfattelse af det kommende system. Derfor kan man blive nødt til at træffe beslutning om, hvilken version man fortsætter med, til trods for at nogen vil føle sig overhørt og forbigået. Det er væsentligt, at man

---

<sup>32</sup> DIS kompendium S. 79.

<sup>33</sup> Løst oversat fra: a system to do X, by the means of Y, in order to achieve Z.

ikke træffer beslutningen på forhånd og kun udarbejder rod definitioner, man på forhånd ved, man selv vil kunne godkende – dette vil resultere i konflikter, der ikke kommer frem i lyset, hvilket de bør<sup>34</sup>. En antagelse fra analytikerne om at de selv objektivt udarbejder rod definitioner uden skjulte antagelser er lige så fejlagtig og ligger absolut ikke i forlængelse af soft system tankegangen<sup>35</sup>!

### 8.2.6. Konceptuelle modeller

Pensum: Avison & Fitzgerald, S. 122-127 (DIS kompendium S. 91-93).

Rige Billeder og Root Definitions er medvirkende til at give et overblik over en organisations informationsbehandling – denne behandling vil muligvis allerede ske elektronisk i et eller andet omfang. For at gøre analysen af HAS komplet er det nødvendigt at bygge en model, der illustrerer, hvordan forskellige aktiviteter er relateret til hinanden eller hvordan de i en eller anden grad burde være relateret til hinanden – dette kaldes en Conceptual Model (herefter CM).

Er der stor forskel på hvad der faktisk sker i organisationen og hvad der burde ske, kan det være vigtigt at synliggøre det overfor organisationen for dermed at initiere ændringer i organisationens HAS – men husk at formålet er at optimere

Hvad er formålet med en konceptuel model:

- Er essentielt for konsulentens design aktiviteter
- Kommunikationsmedium mellem arkitekter og kunder der har til formål, at sikre at det bedste design vælges.
- Instruktionssæt for softwareudvikleren

En analogi hertil er byggebranchen hvor en lignende model har været anvendt i mange år – netop til disse formål. Modellen er derfor langt mere anvendt og testet i denne branche end det er tilfældet ved udvikling af informationsbehandlingssystemer.

Ved systemudvikling er der ikke en klar skelnen mellem konsulentens indtryk og udviklerens ”blueprints” – hvilket betyder, at alle parter skal forstå den konceptuelle model. Det betyder ikke at modellen ikke kan være kompleks – sådan er livet jo engang ☺

Den konceptuelle model udarbejdes på baggrund af den valgte Rod definition som følger:

- Dan et indtryk af hvorvidt systemet skal udføre en fysisk eller abstrakt transformation på baggrund af Rod definitionen.
- Udvælg et lille antal verber der beskriver fundamentale aktiviteter i det definerede system.
- Modellen skal udvikles på baggrund af, hvad systemet skal kunne udføre, hvordan stillede krav opnås og til sidst hvordan det er overvåget og controlled.
- Strukturere aktiviteter sammen i grupper.
- Anvend pile til at forbinde disse aktiviteter, hvor de er logisk forbundet med hinanden – enten ved information, energi, materiale eller andet.
- Verificer modellen ved at sammenholde den med den opnåede virkelighed beskrevet i problem-situationen<sup>36</sup>.

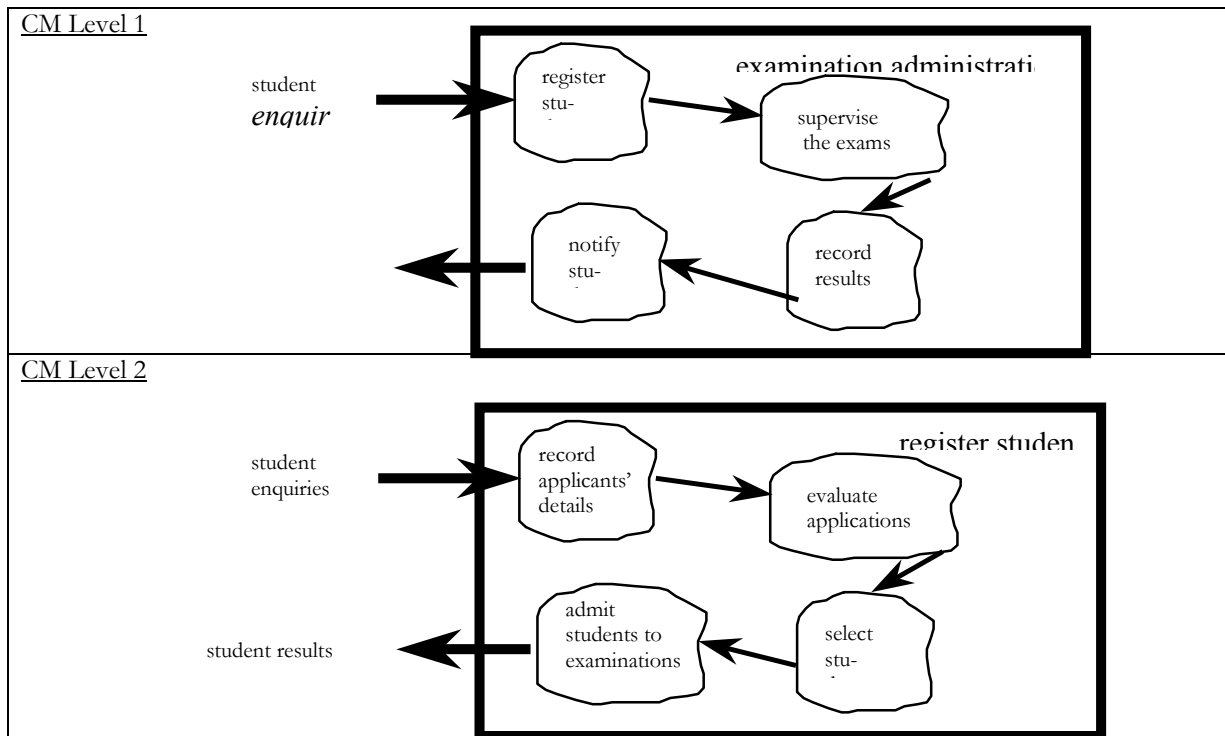
---

<sup>34</sup> DIS kompendium S. 80n.

<sup>35</sup> DIS kompendium S. 82ø

<sup>36</sup> Verifikationen sker i det femte stadie i SSM.

Nedenfor er gengivet et eksempel på en CM's 1. & 2. Level.



Figur 9 Et eksempel på konceptuelle modeller

For at sikre, at de mest anvendte subsystems er identificeret og forstået korrekt, vælger man at beskrive disse mere detaljeret – både i ord og diagrammer. Med andre ord en funktionel dekomponering af udvalgte CM'er (se eksemplet ovenfor).

Dette detailniveau er også vigtigt for at sikre at der er fuld forståelse mellem problemejer og problemløser – netop for at problemløser kan forstå subsystems. Som problemløser her er vi kun interesseret i hvad der er "conceptually necessary".

Igen – den CM udledes af Rod definition og er en model af HAS – elementerne i den CM er derfor aktiviteter som kan udledes via verberne i Rod definitionen. Normalt er antallet af aktiviteter i omegnen af 5 til 9.

Selve processen omkring den CM er abstrakt og formålet hermed er at udvikle et alternativt syn på problemområdet – så når det alternative syn er udviklet kan vi returnere til "real world" og teste modellen. Dermed illustrerer den CM hvad der bør ske i organisationen for at opnå de mål der er specificeret i Rod definitionen. Styrken ved den CM er, at den er god som kommunikationsværktøj til at skabe forståelse omkring modellen [DIS S.93].

I mindre organisationer er informationsbehandling ofte meget uformel – alle ved hvad der sker og folk går tæt op og ned af hinanden – informationsflowet fokuseres omkring de erfarne medarbejdere der bliver "walking databases"...hvilket både kan være med til at gøre det svært at for nye funktioner eller personer at få information – men også omvendt skaber store problemer når de forlader organisationen!

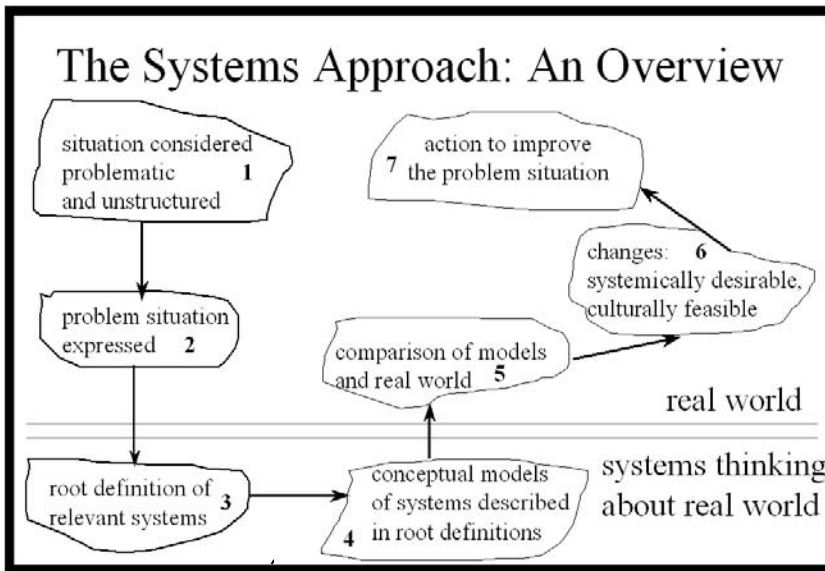
Når man sammenligner sin CM med ”real world” er det vigtigt at stille sig selv spørgsmålet → Does the information flow smoothly? – dermed bliver den CM en teknik til at tænke over hvordan subsystems skal organiseres for at opnå mål beskrevet i Rod definitionen [DIS S. 93].

### 8.2.7. The enquiring process which is SSM

Pensum: Checkland & Scholes, pp. 27-44 (DIS kompendium S. 94-103).

#### The Process Overall

Generelt præsenteres SSM som en proces der består af syv steps, hvilket også er illustreret ved den velkendte figur nedenfor



Copenhagen Business School, Karlheinz Kautz, DIS7-8, Systems Approach - Rich Pictures ...

6

Figur 10 The systems approach

Artiklen er en anden indgangsvinkel til anvendelse af SSM end den syvsteps model der er illustreret ovenfor. Se illustrationen [DIS S. 95]. Figuren illustrerer, at vi har en situation – der af mindst én person opfattes som problematisk -, hvilket skaber en følelse af, at dette der i dette område er behov for forbedringer.

Til håndtering af problemsituationen er SSM analytikerne med involvering fra relevante interessenter. Selvom forskellige analytikere anvender forskellige tilgange til at forbedre problemsituationen vil det i begge tilfælde lede til det samme resultat – nemlig en række forslag til ”tasks” (forbedringsforslag) og en række ”issues” (forskellige ting der er uenighed om).

I højre side af figur 2.6 [DIS S. 95] er der illustreret en række relevante modeller af HAS, udarbejdet med henblik på at eliminere problemsituationen. Dette sikres ved at sammenligne modellerne med ”real world” og strukturere debatten omkring forandring af problemområdet. Det vi som SSM analytiker er ”på jagt” efter, er en række forandringer som kan implementeres i ”real world” som et kompromis mellem forskellige interessenter.

Mange har beskrevet SSM som en logisk drevet "thinking and debate" der har været anvendt til at stille spørgsmål ved "real world". Hvilket en række succesfulde studier har gjort. Selvom "logic" er vigtigt i analysen af "human affairs" er det også nødvendigt at tage højde for det kulturelle aspekt af human situations.

I de seneste år er den kulturelle strøm, der er illustreret til venstre i figur 2.6 og består af tre eksminer-inger af problemområdet, blevet udviklet til SSM:

- Interventionen i organisationen
- Problemsituationen som et socialt system
- Problemsituationen som et politisk system

Det er klart at den logiske og den kulturelle strøm vil spille sammen og informere hinanden. Eksempelvis vil de udvalgte "relevante systemer" fortælle os noget om kulturen i organisationen – men omvendt vil viden om organisationen også hjælpe os med at udvælge relevante systemer.

Ændringer implementeret som et resultat af brugen af SSM, med det formål at ændre problemsituationen som oprindeligt anskuet, initierer en ny problemsituation og en ny læringsspiral er påbegyndt og metoden kan påbegyndes forfra – never ending story...

### **The Stream and Logic-based Enquiry**

Afsnittet beskriver, hvordan den logisk-baserede tankegang hvor relevante systemer er udvalgt, navngivet, modelleret og sammenlignet med opfattelsen af "real world" problemsituationen ser ud.

### **Selecting Relevant Systems**

Intet HAS er instinktivt relevant til problemsituationen – valget er altid subjektivt. Derfor er det nødvendigt, at træffe en række valg og se hvad disse fører os til og derigennem lære vejen til det sande relevante system... Som SSM analytiker er der nødvendigt at acceptere, en subjektiv indgangsvinkel til udvælgelsen af relevante systemer. Det er ikke altid er nemt for systemudviklere (hard thinkers 😊)

Der kan skelnes mellem to typer af valg omkring relevante systemer:

1. *Primary-task system*: Er aktiviteter der er direkte synlige i organisationen – valget sker uden refleksion og er typisk den eneste type af valg der foretages i "hard system thinking". Grænsefladerne har typisk sammenhæng til de der er i organisationen.
2. *Issue-based*: Denne type af systemer kan vi ikke nødvendigvis forvente at finde formelle systemer af i "real-world", men det kan i stedet være "a system to resolve disagreements on resource use" og er typisk kaldet "issue-based relevant systems". Generelt har grænsefladerne ikke sammenhæng til organisationen.

Eksperimenter har vist, at det er vigtigt at træffe valg af begge typer, men sondringen mellem disse er ikke skarp eller absolut – en metode til at skelne mellem dem kan dog være, at "primary task" har sammenhæng til de formelle organisatoriske arrangementer, mens "issue-based" har sammenhæng til mentale processer som ikke direkte er formaliseret i "real world" [DIS S. 97].

### **Naming relevant systems**

I udviklingen af SSM blev det hurtigt klart, at det er nødvendigt, at være meget opmærksom på hvordan vi vælger, at navngive relevante systemer. De skal navngives på en måde, så det er muligt, at bygge en model over det navngivne system. Navnene er kendt som "Rod definitioner".

En Rod definition beskriver formålets kerne af et formålstjenligt aktivitets system, hvor formålets kerne altid udtrykkes som en transformationsproces hvor input forandres til output. Eksempel herpå er i artiklens figur 2.7 [DIS S. 97]. På baggrund af historiske analyser har Checkland foreslået, at en velformuleret Rod definition skal indeholde elementerne i figur 2.8 – der tilsammen udgør CATWOE, hvor kernen er T (Transformation) og W (Weltanschauung) som gør den meningsfuld. Øvrige aktiviteter tilføjer idéer der underbygger aktiviteterne.

Root Definitions: CATWOE - The Completeness Test	
• <u>C</u> ustomer	- <b>whom</b> (victims or beneficiaries of T)
• <u>A</u> ctor	- those <b>who</b> do T
• <u>T</u> ransformation	- <b>what</b> (input -> output (T))
• <u>W</u> eltanschauung	- assumptions (the worldview that makes the T meaningful in context)
• <u>O</u> wners	- answerable (those with the power to stop T)
• <u>E</u> nvironment	- environmental constraints (elements outside the system which are taken as given, but nevertheless affect its behaviour)

Copenhagen Business School, Karlheinz Kautz, DIS9-10, Systems Approach - Relevant...

5

CATWOE - An Example	
A university owned and operated system to award degrees and diplomas to suitably qualified candidates (X), by means of a suitable assessment (Y), in conformance with national standards, in order to demonstrate the capabilities of the candidates to potential employers (Z).	
• <u>C</u> ustomer	- candidate students
• <u>A</u> ctor	- university staff
• <u>T</u> ransformation	- candidates -> degree holders
• <u>W</u> eltanschauung	- the belief that awarding diplomas is a good way of demonstrating the qualities of candidates to potential employers
• <u>O</u> wners	- the university governing body
• <u>E</u> nvironmental constraints	- national educational and assessment standards

Copenhagen Business School, Karlheinz Kautz, DIS9-10, Systems Approach - Relevant ...

6

En Rod definition's kernetransformation vil blive "a system to do X by Y in order to achieve Z" – hvilket er godt at huske når man skriver sine "Rod definitioner".

### Modelling Relevant Systems

Afsnittet behandler udarbejdelsen af konceptuelle modeller. Modellering af relevante systemer tager udgangspunkt i Rod definitionen og CATWOE. På den baggrund udarbejdes et subsystem der beskriver sammenhængen mellem aktiviteter på overordnet niveau. Hertil skal tilføjes processer til at overvåge og kontrollere aktiviteterne.

En logisk analyse af transformationen skal afklare hvorvidt input forvandles til output på en succesfuld måde. Dette gøres på tre niveauer:

- Efficacy: Virker systemet?
- Efficiency: Er systemet effektivt, dvs. sker transformationen med et minimum af ressourcer?
- Effectiveness: Kvaliteten af den transformation der opnås (langsigtede mål).

Masterspørgsmål: Hvorfor er langsigtede mål vigtige? → det er de fordi vi ellers skal igennem hele processen igen. Se modeller figur 2.10 & 2.11 [DIS S. 100-101]. De illustrerer hvor målet for Effectiveness er på et andet niveau en de to øvrige.

Der udvikles forskellige modeller baseret på forskellige "worldviews" og "transformations". Herudover er det muligt at udarbejde mere komplekse modeller hvor der foretages mere end én transformation.

Fordelen ved at anvende mere komplekse modeller er at de kan berige diskussionen når modellerne skal sammenlignes med real world. Ulempen er, at det kan påvirke os til at tænke mere på denne debat og hvad der kan lade sig gøre i real world.

En test af modellen – rent praktisk – er at sikre, at der er konsistens mellem CATWOE, Rod definitionen og den konceptuelle model. Tabel 2.1 [DIS S. 102] viser en oversigt over forløbet og mulige afvigelser i forhold til SSM.

### **Comparing Models with Perceived Reality**

Til sammenligning af den konceptuelle model og real world, er der en række metoder.

Modeller er alene en struktureret metode til at finde frem til hvordan den kan forbedres bl.a. baseret på forskellige verdensanskuelser. Checkland beskriver fire metoder til at sammenligne konceptuelle modeller med real world.

- Uformeldiskussion af modellen over en kop kaffe.
- Formel stillingtagen til modellen via udspørgning.
- Storytelling – modellen fortælles med andre ord til omverdenen (svær).
- Comparing model and reality (matrix).

Lige gyldigt hvordan modellerne anvendes til at sammenholde den konceptuelle model med real world, er formålet ikke at forbedre de eksisterende modeller. Formålet er at finde ”accommodation” mellem forskellige interessenter som man kan argumentere for, vil være en forbedring til den oprindelige problemsituation.

---

## **8.3. SSM til udarbejdelse af informationssystemer**

Pensum: Checkland & Scholes S. 53-58 (DIS kompendium S.87-90)

Dette afsnit handler om hvorledes SSM kan bidrage til udarbejdelsen af informationssystemer. Afsnittet er ganske kort og omhandler kun nogle overordnede emner. Indgangsvinklen er, at implementering af IT har været behandlet som en stor investering, og fokus derfor har været på rentabilitet og livscyklus i stedet for ændring af organisationer. SSMs bidrag til denne problematik er at fokusere på, hvilke informationssystemer der er relevante uden kun at fokusere på økonomiske termer.

### **Data og informationssystemer**

Teksten definerer grænsen mellem data og information til at være henholdsvis fakta og fakta med en tilknyttet kontekstafhængig mening. I forlængelse heraf udledes formålet med informationssystemer – de understøtter handling. Eksempelvis ved at hjælpe i beslutningsprocesser, gennemførelse af beslutninger m.m.

Disse definitioner giver et problem i forbindelse med navnet informationssystemer. Dette navn lægger op til, at informationssystemer – her opfattet som computerbaserede systemer – både behandler fakta og knytter en mening til. Denne ”menings tilknytning” er en menneskelig handling og vil kunne gøres forskelligt afhængigt af, hvem der gør det. Der opstår derfor en konflikt, hvis et system vil knytte en anden mening til fakta end den pågældende bruger vil.

### **Fra weltanschauung til datamanipulation**

Det er i konflikten mellem systemets levering af information og brugerens opfattelse af samme, at SSM kan anvendes. Brugen af SSM skal sikre at der kommer fokus på, hvorledes brugerne ser på deres verden og dermed hvordan de vil tolke fakta med henblik på at udlede en mening. Ligeledes skal SSM medvirke til at der bliver set på de handlinger systemet skal medvirke til at underbygge, da dette er væsentligt for at systemet senere opfattes som meningsfuldt af brugerne.

I den praktiske analyse af HAS kan man hjælpe sig selv ved at spørge, hvilken information der bruges til at understøtte de forskellige handlinger man finder, ligesom man kan spørge hvorfra denne information kommer, i hvilken form og hvor ofte. Derved har man skabt et fundament til udarbejdelsen af datastrømdiagrammer<sup>37</sup>.

Ved at anvende denne indgangsvinkel, hvor man starter med at analysere HAS i organisationen, inden man går i gang med struktureret analyse, skulle bidrage til at systemet bliver fokuseret mod brugernes opfattelse af den verden de indgår i. Og derudover at det målrettes til at understøtte de handlinger det er tiltænkt – alt sammen noget der bør bidrage til et relevant system, der giver mening i den kultur det skal indgå i.

---

<sup>37</sup> I denne proces kan man nok med fordel skele til afsnittet nedenfor om struktureret analyse, hvor dette gennemgås mere uddybende.

## 9. STRUKTURERET ANALYSE.

Pensum: Delskov & Lange, Struktureret analyse 1991-. pp. 17-66

---

### 9.1. Introduktion - del I

#### 9.1.1. Introduktion

Bogens formål er at give en præcis gennemgang af integreret systemanalyse som er en videreudvikling af struktureret analyse<sup>38</sup>. Struktureret analyse svarer til en funktionel analyse som vi kender fra 2. år. I relation til Multiview er dette begyndelsen på andet stadie ”Analysis af Information” se Multiview også kaldet ”hard system analysis”.

Oprindeligt tog man udgangspunkt i systemernes funktioner, men udviklingen og forfatterens erfaring med struktureret analyse metoden, er årsagen til at fokus i lige så høj grad er på analyse af data. Det vil sige, at informationsanalyse sidestilles med funktionsanalyse og det er på den baggrund metoden kaldes integreret systemanalyse.

Denne første del I indeholder en oversigt over metoden integreret struktureret analyse (ISA) samt en præsentation af vigtige begreber. I de efterfølgende dele II – IV beskrives værktøjerne og deres anvendelse mere indgående.

#### 9.1.2. Fordele og ulemper ved ISA

Systemudvikling er ofte kendetegnet ved:

- At ingen har fuldt kendskab til systemets mål og krav.
- At udviklingen kræver, at forskellige personer med forskellige perspektiver og fagsprog skal tale sammen<sup>39</sup>.
- At systemer ofte er store og uoverskuelige.
- At virksomhedens succes er afhængig af, om resultatet bliver godt og er indenfor de økonomiske og tidsmæssige rammer.

Efterfølgende stilles der i bogen nogle spørgsmål, der fortæller hvorvidt ISA er et godt stykke værktøj, hvor svær metoden er at bruge og om den tager højde for ovenstående problemer. I teksten fremhæves blot argumenterne for og imod.

Argumenter for:

Systemer skal være overskuelige og fleksible<sup>40</sup>, så ændringer kan implementeres med mindst mulige omkostninger og så smertefrit som muligt. Det særlige ved integreret systemanalyse er, at det ikke er en stringent og ufravigelig metode, men et instrument der stræber efter at tilpasse retningslinierne efter forholdene. Metoden kommer mest til sin ret ved meget komplekse systemer.

Metodens begrænsninger ligger i, at den for det første er afgrænset til kun at behandle retningslinier i analysefasen af systemudvikling og for det andet, at den kun analyserer det rent informationsmæssige. Det betyder, at metoden som den er beskrevet i bogen, er bedst anvendelig til administrationssystemer,

---

<sup>38</sup> Struktureret analyse er oprindelig udformet af Yourdon og DeMarco (1978) og videreudviklet af S. MCMenamin og J. Palmer (1984).

<sup>39</sup> Det er med andre ord, paradigme spørgsmålet og diskussionen om participation i samarbejde med kultur forskelligheder.

<sup>40</sup> Hermed menes, en lav grad af kobling som vi kender det fra DØK 2. år

men at den ved anvendelse af supplerende værktøj også kan bruges til modellering af processtyrings-systemer.

Metoden lægger vægt på at udviklerne afdækker informationsbehandlingsformål og indhold, ved en struktur der følger problemet frem for den hidtidige teknologi. Ved et tage udgangspunkt i en problemorienteret nedbrydning af problemområdet og ikke teknologien<sup>41</sup> tror forfatterne på at analysemetoden er kommet for at blive. De mener derudover, at metoden ikke er specielt svær at bruge og at hele metoden ikke nødvendigvis skal bruges for at opnå dens fordele. De stiller et lille regnestykke op der siger: hvis halvdelen af metoden anvendes, opnås en fjerdedel af dens fordele.

Har man sagt ja til integreret systemanalyse, har man også sagt ja til, at skaffe al viden om et (del)system inden programmering og at dokumentere denne viden. I analysefasen inddrages i høj grad brugere<sup>42</sup> hvilket er meget naturligt, når man tænker på, at der er dem, der skal bruge systemet bagefter. Disse to argumenter er medvirkende til, at analysefasen forholdsmæssigt bliver mere omfattende en programmeringen. Det samlede tidsforbrug er det samme, men den store gevinst ligger i vedligeholdelsen og omkostningerne forbundet hermed. Ved implementering af denne analysemetode i virksomheden, flyttes ressourcetrækket altså til et tidligere tidspunkt i systemudviklingen.

### 9.1.3. Systemanalysens platform

Som udgangspunkt skal der defineres nogle begreber, der anvendes i metoden.

*System* – er et informationssystem der modtager informationer fra omverdenen, behandler informationerne og afgiver andre informationer til omverdenen<sup>43</sup>.

*Omverden* – et systems omverden er enheder uden for systemet, det kommunikerer med, men ikke kan bestemme over.

Et konkret system er opbygget af processer, data og teknologi.

- *Proces* – en proces modtager data, der bearbejdes til andet data (informationer).
- *Data* – er altså informationer<sup>44</sup>. Der skelnes mellem lagrede data (data der er lagret i systemet) og kommunikationsdata (data der kommunikerer mellem omverdenen og systemet, eller internt i systemet).
- *Teknologi* – er alle tænkelige midler der anvendes til at udføre databehandlingen (mennesker, maskiner og medier).

Der analyseres kun forudsigelige behov, med henblik på at de skal automatiseres. Uforudsigelige behov henvises til adhoc behandling.

### 9.1.4. Et systemudviklingsforløb

Der skelnes mellem udførende funktioner<sup>45</sup> og styrende funktioner<sup>46</sup>. Den udførende del af systemudviklingsforløbet kan deles op i fire hovedfunktioner: Foranalyse, systemanalyse, systemdesign, implementering og vedligeholdelse. Hver hovedfunktion beskrives i det følgende.

---

<sup>41</sup> Teknologien forandrer sig meget hurtig, hvorfor en metode baseret på teknologi hurtig forældes.

<sup>42</sup> Det refererer til teksten om participation.

<sup>43</sup> Det er her væsentligt at se forskellen til SSM, hvor et system ikke har noget med IT at gøre.

<sup>44</sup> Der er i bogen ikke yderligere redegjort for begrebet data. Forskellen fra de oprindelige definitioner på data og information er marginale. Men når en proces modtager data og behandler dette er det ikke nødvendigvis information efterfølgende, grundet detaljeringsniveauet. For den næste proces kan det være information, men for menneskene i systemet bliver det måske først til information når det yderligere er blevet behandlet og til slut præsenteres.

<sup>45</sup> Målet er det system der skal udvikles.

Foranalysen er en rapport/projektkommissorium der danner beslutningsgrundlag for projektets gennemførlighed og er udgangspunktet for planlægning og gennemførelse af systemanalysen. Den skal afdække mål og krav, angive forskellige mulige løsninger samt vurdere fordele og omkostninger.

Systemanalysen skal fastlægge mål og krav og beskrive det nye system (problemområde). Formålet er altså at skaffe oplysninger om virksomhedens behov for informationsbehandling i en struktureret form. Forud for analysen skal virksomhedens (ledelse og medarbejdere) politik på forskellige områder defineres. Det vil sige, hvilke regler skal der gælde for givne områder.

En grundlæggende idé i ISA er, at et problem skal forstås, før en løsning udformes. Derfor specificeres på dette stadie kun krav der danner grundlag for konstruktion. Systemanalysens resultat bliver en kravspecifikation indeholdende en essentiel model af det fremtidige system og en designskitse.

Systemdesign indebærer konstruktion af det nye system, med bedst mulig udnyttelse af såvel tekniske som organisatoriske muligheder. Herunder også valg af maskinel, konstruktion af programmer og databaser.

Implementering er en fase hvor systemet realiseres, det vil sige, programmering, test og eksisterende data konverteres og nødvendig driftsdokumentation (manualer) udarbejdes.

Vedligeholdelse er noget der sker løbende, rettelser af fejl og tilføjelse af nye faciliteter.

### 9.1.5. Vigtige begreber

Afsnittet viser sammenhængen mellem nogle systemudviklingsbegreber. Mange af begreberne kender vi allerede, men for en god ordens skyld nævnes de lige igen.

*Systemudviklingsmetode* – er en fremgangsmåde til gennemførelse af systemudvikling.

*Teknik* – henviser til nogle værktøjer og angiver hvorledes en arbejdsproces kan udføres. Strukturerede teknikker refererer til teknikker defineret under Struktureret - analyse, -design og -programmering.

*Strategi* – er en overordnet fremgangsmåde, der besluttet på grundlag af opgavens karakter.

*Værktøjer* - er et hjælpemiddel til at understøtte nogle arbejdsprocesser.

*Model* – er en forenklet gengivelse af den virkelighed den beskriver. Der skelnes mellem modeller af det eksisterende system og det fremtidige. Derudover skelnes der mellem fysiske modeller<sup>47</sup> og essentielle modeller<sup>48</sup>.

*Systemdokumentation* – systemmodellerne udgør hovedparten af denne dokumentation.

*Blivende dokumentation* – danner basis for drift og vedligeholdelse.

### 9.1.6. Systemanalysens aktører

Kort skitseres her, hvilke færdigheder aktører tilknyttet projektet skal have.

- Indsigt i det forretningsområde edb-systemet skal betjene (I Pist'ase skal der være kendskab til ski/rejsebranchen).
- Indsigt i edb-teknologiens muligheder og begrænsninger.
- Indsigt i udviklingsprocessen (kende til hvordan man analyserer, designer, realiserer)
- Indsigt i IT-ledelse

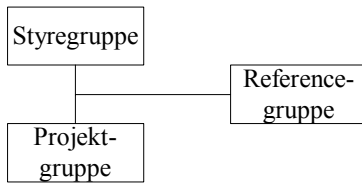
---

<sup>46</sup> Har til formål at styre fremstillingsprocessen.

<sup>47</sup> Viser hvorledes et system implementeres.

<sup>48</sup> Viser informationsbehandlingen i optimal forstand uafhængigt af en bestemt teknologi.

Nedenfor illustreres hvordan en projektorganisation kan se ud. Den er bemandet på tværs af organisationen og fastlægger retningslinier for de impliceredes opgaver og kompetence.



Figur 11 Organisering af projektet

*Styregruppen* fastlægger rammerne for projektet som kan være estimering af budget og tidsramme.

*Projektlederen* leder projektet og er bindeled mellem styregruppe og projektgruppe.

*Projektdelegerne* arbejder.

*Referencegrupper/-personer* står til rådighed med ekspert viden/sparring (review).

Ligeledes findes der på brugersiden i organisationen forskellige profiler.

*Systemejeren* betaler for gildet (Pist'ase).

*De funktionsansvarlige* har det daglige ansvar for de funktionsområder, der er omfattet af systemet (Hotelchef, kontorchef osv.).

*Daglige brugere* har kendskab til detaljerne og kan tage stilling til detaljerede krav til systemet. De kaldes også "hands on" eller slutbrugere.

*Systemanalytikeren* skal bringe viden frem så en kravspecifikation kan udarbejdes.

*Systemkonstruktør/programmerer* programmerer.

### 9.1.7. Forudsætninger omkring brugerdeltagelse

Jævnfør bogen, skal brugere der deltager<sup>49</sup> have kendskab til projektets rammer, metoden der anvendes samt kende deres rolle i hele processen. Relateres disse holdninger til Pist'ase ski eller andre mindre projekter, er det utopisk at opnå alle kriterier til fulde. Et fyldestgørende kendskab til metoden vil være det sværeste kriterium at opnå, ofte på grund af omkostninger i forbindelse med ressourceforbrug.

## 9.2. Modeller og værktøjer

### 9.2.1. Fysiske og essentielle modeller

Som indledning til kapitlet er der i bogen et meget beskrivende eksempel om modeller, som der opfordres til selv at læse<sup>50</sup>.

Ved systemudvikling skelnes der skarpt mellem analyse<sup>51</sup> og design<sup>52</sup>.

#### Essentiel model

En essentiel model er et optimalt billede af systemet. Systemanalysens resultat er den essentielle model. Modellen er teknologineutral og viser kun de funktioner, der skal udføres, og de data der skal lagres. Kravene til systemet ændres ikke ligeegyldigt hvilken teknologi der senere vælges. Essentielle krav til systemet er sjældent nye, oftest svarer kravene til de funktioner, der allerede bliver udført, de skal bare struktureres. Inden der tages stilling til systemets design, skal de essentielle krav verificeres.

<sup>49</sup> Ud over det beskrevne her er der aspekter fra teksten om Participation der gør sig gældende.

<sup>50</sup> Delskov og Lange: *Struktureret analyse*, punkt 4.1 side 37.

<sup>51</sup> Beskriver forfatterne som det at skaffe sig viden og synliggøre den.

<sup>52</sup> Beskriver forfatterne som det at træffe teknologiske beslutninger.

## Fysisk model

Systemudvikling betyder systemændring. Langt de fleste ændringer er teknologiske ændringer hvilket vil sige, at manuelle funktioner eller opgaver ændres til at være maskinelle<sup>53</sup>. Idet det netop er teknologien man ændrer, skelnes der skarpt mellem ”den gamle fysiske model” (er et billede af det eksisterende system) og ”den nye fysiske model” (er et billede af systemet efter ændringen). Den nye fysiske model er en designmodel, fordi den viser, hvilke teknologiske beslutninger der er truffet. Det første step i ”Multiview” også kaldet ”soft system analysis” bidrager til bedre at beskrive kunne beskrive den gamle fysiske model. Det er ikke et krav, hverken at udforme en ny eller gammel fysisk model. Men i nogle projekter kan det være en hjælp, at udarbejde den essentielle model på baggrund af ”en gammel fysisk model”. Og det kan inden designfasen være praktisk at udarbejde løsningskitser på baggrund af ”den nye fysiske model” som beslutningsgrundlag.

### 9.2.2. Værktøjer

Den færdige essentielle model tjener to formål, dels er det en kravspecifikation<sup>54</sup> og dels er det en vidensbase, der skaber overblik. Husk at kravspecifikationer skal verificeres ligesom kode m.m..

En essentiel model udtrykkes ved hjælp af fem forskellige værktøjer<sup>55</sup>, dog kan de ikke stå alene og udgør tilsammen en model af systemet. Nedenfor findes kort gennemgang af de fem forskellige værktøjer. Der kan til flere af værktøjerne drages paralleller til værktøjerne som vi kender dem fra OOAD. De bliver kun kort nævnt her og i de følgende kapitler beskriver værktøjerne i detaljer.

*Datastrømsdiagrammer* (DS-diagrammer) er hierarkisk opbygget og viser sammenhængen mellem data og processer samt mellem processer indbyrdes. En funktion kaldes en proces og symboliseres med en cirkel (boble). Da vi endnu ikke tænker på teknologiske midler kan proces, datalager osv. Være både elektronisk og menneskelig.

- Datastrømme (kommunikationsdata)
- Datalager (to parallelle streger)
- Pile til (skriv) og fra (læs)

Systemet brudes ned i et antal mindre dele indtil det kan overskues, derefter beskrives hver del ved hjælp af en procesbeskrivelse, se nedenfor. Det beskriver med andre ord, hvordan data flyder.

*Entitets/reasons-diagrammet* (E/R – diagram) (kalder de systemets spejl) viser opdeling og sammenhænge mellem entiteter set fra informationssiden. Sammen med dataordbogen (nedenfor) kaldes de to modeller for datamodellen/informationsmodellen. Data er som vi ved en vigtig ressource i dag i virksomhederne<sup>56</sup>, informationsmodellens rolle er derfor at lagre og synliggøre denne ressource. Informationsanalysen har to formål, dels at opnå en fælles begrebsopfattelse i virksomheden og dels at danne grundlag for et optimalt databasedesign.

Med fælles begrebsopfattelse menes, at data bliver til information når indholdet tydes. Det er netop argumentet for at data skal beskrives entydigt, så det forstås ens af alle brugere.

I bogen er der et afsnit om optimalt databasedesign, det vil jeg ikke komme nærmere ind på, men henviser til Database bogen af Lars Frank<sup>57</sup>. Med andre ord, fortæller diagrammet hvilke entiteter der er relateret og hvordan.

<sup>53</sup> Eventuelt konvertering af data fra kartoteker til databaser.

<sup>54</sup> Som kravspecifikation skal den sikre at systemet er korrekt, komplet og konsistent.

<sup>55</sup> Det er en kombination af grafik (giver overblik og viser sammenhænge) og ord (beskriver bedst detaljer). De forskellige værktøjer: datastrømsdiagrammer, entitets/reasons-diagram, tilstandsdiagram, dataordbog og procesbeskrivelser.

<sup>56</sup> Der er fokus på videnssamfundet og virksomhedernes evne til at styre denne viden.

<sup>57</sup> Se også en kort skitsering af kriterier for optimal databasedesign i struktureret analyse side 46 nederst.

*Tilstandsdiagrammet* (kalder de systemets bånd) viser en entitets livsforløb. Livsforløbet påvirkes af en hændelse som kan ændre en eller flere entiteters tilstand<sup>58</sup>. For hver entitet der har en livscyklus, tegnes der et diagram.

*Dataordbogen* (kalder de systemets sprog) definerer datanavne og beskriver indholdet af de to dataformer. Ordbogen sikrer entydighed og konsistens på tværs af diagrammerne i den forstand, at alle datastrømme og datalagre (entiteter og attributter samt relationer) beskrives i detaljer. Det vil sige, datastrømmens indhold beskrives, ligeledes gør de lagrede data og de enkelte dataelementer defineres. Ordbogen kan til dels sammenlignes med en relationel model (1. år). Hvert navn fra DS-diagrammerne og E/R-diagrammerne skal have sin egen indgang og fordi det er en ordbog skal den naturligvis også være i alfabetisk orden.

*Procesbeskrivelserne* (kalder de systemets bund) udgør den egentlige beskrivelse af systemets funktioner og beskriver behandlingsregler knyttet til de enkelte processer (med andre ord pseudokode). Processens opgave er at omforme inddata til uddata ved hjælp af det sæt behandlingsregler. Det eneste krav struktureret analyse stiller til procesbeskrivelserne er, at reglerne skal beskrives entydigt. Det stiller systemanalytikeren i den fornemme situation, selv at kunne vælge beskrivelsesmetode så længe den er entydig.

Der findes en række alternative metoder til beskrivelse af behandlingsregler. Almindeligt sprog, struktureret sprog, beslutningstabeller og beslutningstræer er mest egnet til analysefasen hvorimod Nassi Sneiderman diagrammer, Interface kontrakter, JSP (Jackson Structured Programming) og Rutediagrammer er bedst egnet til designfasen. Der findes flere end de nævnte, men de mest anvendte i struktureret analyse er almindeligt- og struktureret sprog.

### 9.2.3. Værktøjerne til sammen

Systemet angribes skiftevis fra funktionssiden (DS-diagrammer) og fra informationssiden (E/R-diagrammer), men der arbejdes med alle værktøjerne sideløbende. Ved hjælp af DS-diagrammer nedbrydes systemet, og de nederste bobler beskrives med procesbeskrivelser. Datastrømme og datalagre defineres løbende i dataordbogen. Ved hjælp af E/R-diagrammer ordnes systemets begreber i entiteter og relationer. De enkelte entiteter og relationer defineres i dataordbogen. Sammenhænge mellem entiteter og hændelser beskrives ved hjælp af et tilstandsdiagram og hver tilstand defineres i dataordbogen<sup>59</sup>.

Det er et krav til en essentiel model at processiden og datasiden hænger sammen, således at DS-diagrammernes datalagre refererer til entiteter og relationer på E/R-diagrammet. Den ovenfor beskrevne sammenhæng er struktureret analyses stærke side og er kendetegnet ved, at hvert navn kun findes et sted i modellen. Skal et navn ændres, er det kun nødvendigt at ændre et sted i modellen. Det er meget vigtigt at sikre konsistens mellem de forskellige værktøjer. Ellers kommer KK og hans påspændings...fætter efter dig.

Dokumentationen og dermed modellen er produktet af selve analysearbejdet, det er ikke længere et ekstra stykke arbejde der skal udføres. Når metoden er indlært og forstået kan der hentes hjælp til at tegne diagrammer, vedligeholde dataordbog og foretage visse konsistenstjek med CASEværktøj.

---

<sup>58</sup> Entiteter er informationsmodellens stabile elementer og hændelser er informationsmodellens stabile elementer.

<sup>59</sup> Dette afsnit er citeret direkte fra Delskov & Lange: *Struktureret analyse*, pkt. 4.8 side 50. Hele modellens sammenhæng og overblik skulle gerne skinne igennem i dette afsnit.

## 9.3. Teknikker og strategier

### 9.3.1. Essentiel analyse

Den essentielle analyse skal sikre et godt system, og repræsenterer en abstrakt model. Multiviews styrke er at den tillader forskellige synsvinkler afhængig af situationen. Dette er en funktionsanalyse, hvorfor vi helt naturligt bør se på problemsituationen ud fra et funktionalistisk perspektiv. Abstrakt skal i denne sammenhæng forstås, som en model af et system uden teknologiske begrænsninger – deraf ordet teknologineutral. Det vil sige, at modellen skal være gældende ligegyldigt hvilken teknologi, der senere vælges ved implementeringen. I modsætning til den fysiske model der også tager højde for teknologien, beskriver en essentiel model kun data og processer. Det essentielle er altså stabilt hvor det fysiske ofte ændrer sig.

Det er nemmest at skelne mellem hvad der er essentielt og hvad der er fysisk ved at give et eksempel. Essentielle funktioner og data fra en bank, er at indsætte penge og at hæve penge. Det er blevet gjort på mange forskellige måder igennem tiden, i takt med at teknologien har ændret sig. Men det essentielle er stadig at indsætte og hæve penge.

Et komplet system består af data, processer og teknologi.

En fysisk model svarer på spørgsmål som: hvordan udføres, hvordan lagres, hvornår udføres, hvem eller hvilken maskine udfører, hvor udføres og hvor lagres? En essentiel model besvarer spørgsmålene hvad udføres og hvad lagres?

Der er tre grundlæggende karakteristika for en essentiel model: at den er **teknologineutral**, at den er **hændelsesopdelt** og at der er **dataintegritet**.

**Teknologineutral** betyder funktioner og data indenfor et forretningsområde, der er stabilt selvom teknologien bliver en anden. Har man først udarbejdet en essentiel model for et forretningsområde skal arbejdet ikke gøres igen, modellen skal dog løbende holdes "up to date". Modellen er med andre ord ikke afhængig af tidens teknologi. Det essentielle er derfor stabilt hvorimod det fysiske kan ændre sig meget med tiden.

Teknologineutralitet har to formål:

1. at sikre et rationelt system. Det nye system må ikke indeholde overflødige elementer eller være unødigt komplekst.
2. At sikre et stabilt billede af systemområdet (forretningsområdet).

Med hensyn til **hændelsesopdeling** menes der, at systemet skal opdeles så der er så få datastrømme mellem systemdelene som muligt. Det betyder det samme som "så lav kobling som muligt".

Der findes to forskellige hændelsestyper, eksterne hændelser og tidsafhængige hændelser.

Eksterne hændelser sker uden for systemet, men får systemet til at reagere. På kontekstdiagrammet blev omverdenen symboliseret med rektangler "eksterne processer". En ekstern hændelse sker i en ekstern proces<sup>60</sup>. Den giver i form af en triggerdatastrøm systemet besked om at en hændelse er indtruffet. Udover at give systemet besked om at en hændelse er indtruffet, indeholder triggerdatastrømmen også almindelig data.

<sup>60</sup> Der henvises til kontekstdiagrammet for en grafisk illustration. En ekstern proces kan eksempelvis være en kunde.

Tidsafhængige hændelser igangsættes af tiden. Indlysende! Også her, er det omverdenen der ønsker at systemet skal reagere. Det kan eksempelvis være hver måned, to gange om året eller på en fødselsdag at systemet skal reagere.

En hændelsesreaktion er summen af processer, der udføres som reaktion på en hændelse. Det vil sige, summen af det antal processer der er igangsat af en ekstern- eller en tidsafhængig hændelse. Ved at gå et skridt højere op på stigen, kan en essentiel aktivitet defineres som hele reaktionen på en, og kun en hændelse. På datastrømdiagrammerne afspejler niveauopdelingen (hvert niveau) de essentielle aktiviteter. Grupperes processer der reagerer på de samme hændelser til en essentiel aktivitet opnås en række delsystemer der ikke kommunikerer med hinanden.

Der er mange fordele ved hændelsesopdeling, det er samme fordele som ved et objektorienteret system:

- Modellen bliver mere overskuelig, fordi den er opdelt i uafhængige ”delsystemer”.
- Modellen er nemmere at gennemskue og nemmere at læse.
- Modellens væsentligste pointe er at få et fornuftigt opbygget system, designet bliver nemt at overskue, gennemskueligt og nemt at vedligeholde.

Ligesom objektorienteret programmering kan hver aktivitet udvikles, programmeres og testes uafhængigt af hinanden. Med hændelsesopdeling sikres derfor en struktur, der er fleksibel overfor ændringer og tilføjelser.

**Dataintegration:** nu snakker vi om data.

I en gammel fysisk model indgår der ingen informationsmodel<sup>61</sup>. Systemet beskrives her som det er inklusive fejl og mangler. Alle systemets data i systemets kartoteker er medtaget og den er derfor ikke optimal som en essentiel model skal være. Som før beskrevet skal også data i en essentiel model være optimale. Ved at foretage en informationsanalyse elimineres derfor tre typer af uhensigtsmæssigheder. Gamle kartoteker indeholder ofte redundante data, der gemmes unødvendig data<sup>62</sup> og kartotekerne mangler nødvendige indgangsnøgler<sup>63</sup>.

Ved dataintegration sikres at lagrede data er struktureret optimalt, samtidig med at informationsmodellen sikrer en entydig tolkning af data. Der er først tale om dataintegritet når alle relationer og indholdet i modellen er konsistente.

### 9.3.2. Den essentielle models betydning.

Er der taget udgangspunkt i et eksisterende system, er den essentielle model først færdig, når den er ajourført og opdateret med brugernes nye essentielle krav. Den skal holdes ajour i hele systemets levetid. Analysefasen afsluttes ganske vist med en ”ny fysisk model”, men hovedproduktet af systemanalysen (informationsanalysen) er stadig den opdaterede essentielle model.

Det skal understreges at modellen ikke er et færdigt system, men grundlaget for systemdesignet. Den skal med andre ord, give et billede af det mest optimale system og sikre de bedste forudsætninger for systemdesignet.

### 9.3.3. Design af det nye system

Overgangen fra analyse til design har været genstand for megen polemik blandt systemdesignere. Bogens metode anviser en teknik der sætter lighedstegn mellem ”ny fysisk model” og overordnet system-

---

<sup>61</sup> Informationsmodel er resultatet af hele andet studie i Multiview.

<sup>62</sup> Mange data kan udledes af allerede eksisterende data eks. Sammentællinger, nøgletal osv...

<sup>63</sup> Manglende indgangsnøgler medfører at nogle funktioner må søge sekventielt – det koster til og ressourcer.

design<sup>64</sup>. Metodens værktøjer egner sig ikke til programdesign, men selvom design er et forbudt ord i analysefasen definerer struktureret analyse en metode til analyse og overordnet design.

Det næste skridt er at træffe nogle teknologiske beslutninger. Teknologien kan deles op i tre kategorier :

- processorer (menneske eller maskiner der kan udføre arbejdet),
- kommunikationsmedier (alle slags medier der transporterer data fra en processor til en anden),
- lagringsmedier og metode (Databaser, kartoteker, ringbind m.m.).

Det kan sidestilles med designfasen i OOAD, hvor der tages stilling til de samme problemstillinger. Hvor skal hvad køre og hvordan.

Beslutninger der skal træffes: hvor skal hvad køres og hvordan (online, batch ...) hvilke processer skal være manuelle og hvilke skal være maskinelle. Beslutningerne kan illustreres med DS-diagrammer. Der skal dog skelnes kraftigt mellem DS diagrammer her og under analysen.

Den essentielle model kan anvendes som udgangspunkt, til at nå frem til det mest optimale design, med den teknologi der er til rådighed. Opgaven ligger i at få ”den nye fysiske model” til at ligne den essentielle model mest muligt. Det vil sige, at tilføje så få teknologiske begrænsninger som muligt.

Den før beskrevne hændelsesopdeling i essentielle aktiviteter skal nu også overføres til designet. Det giver et naturligt grundlag for opdeling af systemet i programmer. En essentiel aktivitet kan med perfekt teknologi udføres umiddelbart eller ”non stop”, når hændelsen kommer til systemets kendskab. Skal hele den essentielle aktivitet udføres tidstro, bliver den essentielle aktivitet til et program. En fysisk aktivitet er de processer, men vælger skal udføres non-stop.

### 9.3.4. Integreret systemanalyse

Analysemetoder har traditionelt set været delt op i funktionsanalyse (også kaldet procedurale side) og informationsanalyse (også kaldet non-procedurale side). Integreret systemanalyse (ISA) er en integration af de to analysemetoder<sup>65</sup> fordi der arbejdes parallelt med begge sider. Viden fra den ene side anvendes til analyse af den anden og omvendt.

Integreret systemanalyse benytter sig af en hændelsesstyret fremgangsmåde. Systemets hændelser bruges med andre ord, som udgangspunkt for både funktions- og informationsanalysen.

ISA består altså af nogle teknikker der er sammensat i en basisstrategi. Disse teknikker kan også anvendes enkeltvis eller sammen med andre strategier.

Alternative strategier er:

- Blitzing
- Traditionel struktureret analyse
- 

Del III omhandler integreret systemanalyse og dermed basisstrategien. Del IV omhandler alternativerne nævnt ovenfor. Blitzing er beskrevet i detaljer, hvor traditionel strategi kun er beskrevet overordnet sammenlignet med basisstrategien.

---

<sup>64</sup> Det overordnede systemdesign er grundlag for grænsefladedesign, programdesign, databasedesign og design af manuelle rutiner.

<sup>65</sup> Det er en integration af både fremgangsmåde som dokumentation.

### 9.3.5. Gode råd

Gode modeller er et middel til at skabe gode systemer.

Brugen af værktøjer må ikke overskygge målet. Det vil sige, en diagramform eller måde at tale med en kunde på, sagtens kan tilpasses situationen og ikke nødvendigvis skal følge metoden. Modellen skal synliggøre hvad vi ved og ikke mindst synliggøre hvad vi ikke ved.

---

## 9.4. Værktøjer – del II

Dette afsnit indeholder indledningsvis en holdningsmæssig indgangsvinkel til det at bruge en stringent metode og de værktøjer der følger med. Efterfølgende beskrives de enkelte værktøjer i dybden, og slutteligt er der en kort beskrivelse af brugen af CASE værktøjer.

Den metodiske indgangsvinkel til selve udarbejdelsen af de enkelte diagrammer vil ikke blive gennemgået i dette afsnit – det kommer nedenfor i del III.

### 9.4.1. Metodeholdninger

ISA er en metodisk indgangsvinkel til software udvikling, som i høj grad er bygget op over brugen af modeller. Det er derfor vigtigt at gøre sig klart, at det ikke er modellerne i sig selv, der er målet med analysen.

Det vil i praksis sige, at det væsentlige i brugen af ISA ikke er at følge metoden stringent. Det centrale er at tilpasse metoden til den situation man arbejder med. Her opstår der en problematik omkring graden af situationstilpasning. Den totale situationstilpasning – også kaldt metodeanarki – stiller store krav til systemudviklerne, hvis det ikke skal ende med tilfældige resultater. Den anden grøft med ingen situationstilpasning, vil være udarbejdelse af modeller for modellernes skyld.

Det væsentlige i anvendelsen af værktøjer er derfor at sikre nogle beskrivelser, der er sammenlignelige og konsistente, men samtidig tilpasse dem til de brugere der skal læse dem. Det vil sige, at hvis man står i en situation, hvor en bruger ikke forstår diagrammerne, eller ikke kan tænke i de baner man gerne vil have, så må man tilpasse værktøjerne til brugeren<sup>66</sup>.

### 9.4.2. Datastrømdiagrammer

Dette afsnit beskriver DS-diagrammets opbygning, samt de regler der knytter sig til selve udarbejdelsen.

Datastrømdiagrammerne anvendes i funktionsanalysen, hvor de giver overblikket over resultatet. Helhedsbeskrivelsen opnås ved at se detaljerne i dataordbogen og procesbeskrivelserne.

Begrebet funktion knytter sig til en gruppering af opgaver - eksempelvis salgsfunktion. I funktionsanalysen nedbrydes funktionerne således til processer og aktiviteter for at beskrive, hvorledes de enkelte funktioner bliver udført.

### 9.4.3. DS-Diagrammets begreber

Et DS-diagram består af fire forskellige symboler:

- *Cirkler* – illustrerer en proces eller en aktivitet
- *Pil* – illustrerer en datastrøm, eller en samling af datastrømme afhængigt af detaljeringens niveauet.

---

<sup>66</sup> For en yderligere diskussion af software processer henvises til IT-kvalitets kompendiet.

- *To parallelle linjer* – illustrerer et datalager. Der er her tale om lagret data til forskel fra kommunikationsdata, der er vist med pilen.
- *En kasse* – illustrerer en ekstern proces. Altså en proces udenfor systemet. Eksteren processer kan have flere navne: interessenter, kilde/terminator eller afsender/modtager af data.

### Processer

En proces er kendetegnet ved disse karakteristika:

- Den udfører et stykke arbejde ved at transformere inddata til uddata efter et bestemt regelsæt.
- Den kan ikke huske data fra gang til gang
- Den aktiveres når der er data til den
- Skal den bruge ældre data, må den hente det på datalagret

I DS-diagrammet skal en proces navngives med et sigende navn. Er der tale om processer på et overordnet niveau, bør den navngives på en måde der bedst muligt dækker over de underliggende processer. Navnet kan med fordel opbygges med et udsagnsord i bydeform efterfulgt af et navneord - eksempelvis *gennemfør betaling*.

Ud over navnet skal processen nummereres – regler for nummereringen knytter sig til niveauopdelingen og er gennemgået nedenfor.

### Datastrømme

En datastrøm er en gruppe af data, der enten sendes internt mellem processer, eller mellem processer og datalager.

Datastrømme internt mellem processer kan enten være internt i systemet, eller mellem systemet og omverdenen.

Datastrømme skal navngives sigende i forhold til de data de indeholder, og den form de befinder sig i. Det kan eksempelvis være *salgsoplysninger* internt imellem processer i systemet og *faktura* til en ekstern proces. Det vil vise, at der i det ene tilfælde er tale om internt kommunikationsdata, og i det andet tilfælde et brev til en kunde.

### Datalagre

Datalagre indeholder data der skal gemmes over tid – og altså ikke kommunikationsdata. Alle datalagre skal navngives med sigende navne. I DS-diagrammet skal datalagre opdeles i entiteter og relationer, hvilket bevirker at der kan komme mange datalagre.

### Eksterne processer

Eksterne processer er uden for det system, der analyseres, men kan sagtens være inden for virksomheden.

De eksterne processer tegnes kun på det højeste diagram niveau – kontekstdiagrammet. Det vil i praksis sige, at på de underliggende niveauer peger datastrømme til eksterne processer ud i ingenting.

De eksterne processer kan navngives med navnet på den pågældende interessent – eksempelvis *kunde* eller *salgsafdeling*.

#### 9.4.4. Niveauer i DS-diagrammer

Funktionsanalysen munder ud i en række DS-diagrammer, der hver især beskriver forskellige niveauer af processer. Jo længere ned man kommer jo mere detaljeret bliver beskrivelsen. Dette kan sammenlignes med en træstruktur, der starter med roden for derefter at grene sig ud. På tilsvarende vis starter DS-diagrammerne med et kontekstdiagram.

Alle niveauerne af diagrammer beskriver hele systemet. Dette skal opfattes således, at der eksempelvis på niveau fire bliver beskrevet en masse processer på forskellige diagrammer, men lagt sammen giver de det samlede system. Ikke alle processer skal nedbrydes til samme niveau – det er situationsafhængigt fra proces til proces.

### **Kontekstdiagrammet**

Kontekstdiagrammet er specielt fordi det kun indeholder én cirkel – det samlede system. Det viser derimod en række kasser samt datalagre. Dette skyldes at diagrammets hensigt er at illustrere systemets afgrænsninger til dets omverden.

Hvis der er datalagre, der deles med andre systemer i virksomheden, er det væsentligt at få disse med, da en del af formålet med ISA er at integrere alle virksomhedens datalagre.

Ved udarbejdelsen af kontekstdiagrammet kan man støde ind i den situation, at der er en afdeling, der har processer både internt og eksternt i forhold til systemet. Hvis der kommunikeres med de eksterne processer, skal den pågældende afdeling optræde som ekstern proces på kontekstdiagrammet.

### **Oversigtsdiagrammet**

Oversigtsdiagrammet er den første nedbrydning af kontekstdiagrammet. Dette diagram kaldes også diagram-0.

Dette diagram giver et overblik over selve systemet, hvilket vil sige, at i den essentielle model indgår kun essentielle aktiviteter – aktiviteter der ikke har nogen direkte indbyrdes sammenhæng.

Datalagrene i oversigtsdiagrammet, vil typisk illustrere grupper af entiteter, da dette medvirker til overblikket<sup>67</sup>.

### **Yderligere opdeling**

Den videre proces med opdeling af DS-diagrammet foregår ved at tage hver enkelt cirkel og nedbryde den yderligere på det næste niveau. Denne proces gentages indtil man når de primitive processer, der er karakteriseret ved at være aktiviteter der skal gennemføres i en sammenhæng. Yderligere kendetegn ved de primitive processer er gennemgået nedenfor.

På de højere niveauer kan man samle flere datastrømme i en pil, men man skal være opmærksom på at navngivningen af datastrømme med flere forskellige data kan være svær.

Et datalager skal først med på DS-diagrammet når det anvendes af flere processer. Det vi i praksis sige at et lager der kun bruges internt af en proces på et højere niveau ikke tegnes på. Når den proces derimod bliver brudt ned og der bliver flere underprocesser der bruger det skal det tegnes på. Som beskrevet under oversigtsdiagrammet kan man anvende grupperinger af datalagre på de højere niveauer for overblikkets skyld.

### **Primitive processer – DS-diagrammets bund**

De primitive processer er målet for DS-analysen. De er karakteriseret ved overskuelighed og afgrænsethed. Det vil i praksis sige, at de skal være så ukomplicerede at navnet alene siger alt om hvad der foregår i cirklen. Derudover skal der være enkle grænseflader til de øvrige dele af systemet. Nedenfor er givet 4 ting der kan fungere som tommelfingerregler:

---

<sup>67</sup> Dette kunne svare til de klynger af data vi arbejdede med under OOAD

- *Processen kan overskues.* Her er der tale om at processen skal være så klart opbygget at der ikke er skjulte komplikationer i behandlingsreglerne – bogen siger at man som regel kan regne med at primitive procesbeskrivelser på en A4-side er passende. Det mener jeg nu er en anelse meget..
- *Processen udfører en veldefineret funktion.* I den essentielle model svarer dette til at processerne skal kunne defineres klart inden for én af disse tre kategorier:
  - *Fundamental proces.* Den skaber data ud fra andet data med henblik på indfrielse af systemets mål.
  - *Vedligeholdelsesproces.* Den lagrer eller sletter data.
  - *Grænsefladeproces.* Den formidler data mellem systemet og omverden i den fysiske form der er nødvendig.
- *Processen udfører arbejde der skal udføres uden afbrydelse, uafhængigt af teknologi.* Svarer til arbejde der skal udføres i en kritisk region af én processor<sup>68</sup>.
- *Fællesprocesser er skilt ud.* Her er der tale om at processer der går igen i systemet med samme ind-data og uddata skal skilles ud med henblik på genbrug af procesbeskrivelser. Fællesprocesser får forskellige numre men samme navn, de steder hvor de optræder, men de beskrives kun med procesbeskrivelse et sted. I DS-diagrammet markeres de med en stjerne i cirklen for at illustrere en fællesproces.

### Nummerering af processer

Hver proces og hvert DS-diagram har et entydigt nummer. Disse numre fordeles efter disse regler:

- Kontekstdiagrammet er diagram –1, og det indeholder proces nummer 0 – det vil sige det samlede system.
- Oversigtsdiagrammet har nummer 0, og det indeholder processerne nummereret med enkelte hele tal.
- De underliggende niveauer nummereres med den proces de beskriver. Eksempelvis vil det sige at diagram nummer 2, viser dekomponeringen af proces nummer 2. Dette diagram vil indeholde processerne 2.1, 2.2, 2.3 osv.

På denne måde kan man hele tiden øge detaljeringsniveauet ved at læse de underliggende niveauer. Hvis ikke der er flere niveauer, vil der være en procesbeskrivelse der viser den pågældende primitive proces.

Overblikket over de enkelte diagrammer skal være styrende for antallet af niveauer. Bogen anviser 7 processer +/- 2 per diagram som det ideelle, men antallet af datalagre er også styrende for dette tal.

### Konsistens

En del af udarbejdelsen af DS-diagrammet går ud på at sikre konsistens imellem de enkelte niveauer. Det vil sige at sikre at der ikke opfindes eller forsvinder data mellem niveauerne. Kontrollen kan ikke gennemføres ved at tælle pile på de enkelte niveauer – disse kan indeholde forskellige mængder data afhængigt af niveauet. Man må i stedet ind og se i dataordbogen for at kontrollere indholdet af alle pile. Dette gøres nemmest nedefra og op.

### 9.4.5. Supplerende regler

Dette afsnit indeholder en række praktiske regler der kan anvendes under udarbejdelsen af DS-diagrammer:

- *Undgå krydsende datastrømme.* Dette nedsætter læsbarheden og kan undgås med kopiering af datalagre. Disse markeres i givet fald med \*. Problemet kan ikke helt undgås.

<sup>68</sup> Kan både være menneske og maskine.

- *Dobbeltpile.* Man kan anvende dobbeltpile i to tilfælde:
  - Når den samme datastrøm går begge veje. Dette kan illustreres med en dobbeltpil navngivet med ét navn.
  - Hvis en datastrøm er svar på en anden datastrøm kan dette illustreres med en dobbeltpil med forskelligt navn i hver sin ende.
- *Pile der deler sig.* Dette kan bruges når den samme datastrøm går til flere forskellige processer. Navnet skrives kun én gang.
- *Afvisninger.* Datastrømme fra omverden skal kontrolleres inden brug. I de tilfælde hvor strømmen bliver afvist uden yderligere registrering eller behandling, kan dette vises med en lille bagud rettet pil. Dette vil ikke være tilfældet, hvis der blot er tale om dele af datastrømmen, der skal verificeres eller uddybes. Eksempelvis vil man ikke afvise datastrømme med mangelfulde betalinger – her vil man registrere indholdet og rykke for resten. Der er altså kun tale om klare afvisninger.
- *Undgå strømme mellem eksterne processer.* Dette vedrører ikke systemet og bør ikke fremgå. Krav om denne illustration er tegn på en forkert afgrænsning af systemet.
- *Accesser: læs og skriv.* Processernes adgang til datalagret kaldes en access. Disse kan antage to former: læs eller skriv.
  - En læs access henter data på lageret
  - En skriv access gemmer nye data, ændrer eksisterende data eller sletter data.

Navngivning af accesser kan udelades i essentielle modeller, hvis indholdet fremgår tydeligt af procesbeskrivelsen.

En pil tegnet fra processen til datalagret viser en skriv access. En sådan pil illustrer altid at data i lagret ændres – denne illustration bruges til trods for at processen er nødt til at læse fra lagret for at kunne opdatere.

Pile fra lagret til processen illustrer en læs access. Dette kan være tilfældet, hvis processen skal læse data som input til en beregning.

I dataordbogen, hvor datastrømmene beskrives skal man kun beskrive de data der overføres til eller læses fra lagret. Det vil sige, at i de tilfælde hvor en primærnøgle kun bruges som indgang i lagret udelades denne fra beskrivelsen – de indgår kun ved oprettelse.

- *Søgekriterier.* Som beskrevet ovenfor indgår identifikations felter ikke i beskrivelsen af datastrømme, hvis de kun anvendes som indgang til lagret. Hvis der indgår flere søgekriterier i en søgning, skal disse kriterier fremgå af dataordbogen.
- *Løkker.* Løkker forekommer mellem to processer når en proces får lavet en del af forarbejdningen af data af en anden proces. I dette tilfælde vil der gå en pil fra proces A til B. Når B har lavet sin del-beregning går der en strøm tilbage til A. Dette skal laves om så A bliver delt i to forskellige processer – en før strømmen sendes til B og en efter der er kommet et resultat tilbage.
- *Kontrolstrømme.* I de tilfælde hvor en datastrøm starter en proces eller afslutter den, kan der være knyttet betingelser til data. Disse betingelser kan være en fordel i visse typer af processtyrings-systemer, men bør som udgangspunkt udelades, da de forvirrer mere end de gavner.
- *Minimum data.* I en essentiel model må der ikke indgå overflødig data i strømmene. Der skal kun indgå det nødvendige for processerne. Det vil sige, at data der er med ”fordi det plejer man at gøre” skal udelades. Undtagelsen til denne regel er strømme til grænsefladerne – her kan man være tvunget af andre systemer til at sende de unødvendige data med.
- *Essentiel orden.* I den essentielle model må det kun være dataafhængighed der styrer rækkefølgen af processer. Det vil sige, at processer der ikke er afhængige af hinanden ikke må være anført i

sekventiel orden. Dette kan eksempelvis være tilfældet med *gennemfør handel* der fører til *opdater salgskonto* og *send faktura*. I dette tilfælde skal de to sidste være sideordnede processer, da de ikke er afhængige af hinanden.

Ved fysiske modeller er dette ikke tilfældet – her viser rækkefølgen den faktiske udførelses orden.

---

## 9.5. Entitets/reasons diagrammet

Dette afsnit omhandler E/R diagrammet der indgår som en del af resultatet af informationsanalyse. E/R diagrammet viser overblikket over informationsmodellen, imens dataordbogen indeholder de detaljerede beskrivelser.

Behovet for datalager er udledt af DS-diagrammet, men heraf er det ikke muligt at se sammenhængene mellem de enkelte entiteter – denne sammenhæng fremgår af E/R diagrammet. Derudover tjener diagrammet to formål:

- At skabe en fælles begrebsopfattelse i virksomheden omkring anvendt data
- At danne grundlag for databasedesign

### 9.5.1. Begrebsforklaring

Dette afsnit gennemgår de forskellige begreber der optræder i informationsanalysen. Til de enkelte begreber vil der være knyttet de regler der er nødvendige for udarbejdelsen af diagrammet<sup>69</sup>.

#### Entiteter

En entitet er et begreb virksomheden gerne vil gemme data om. Entiteter navngives med navneord og tegnes som rektangler i diagrammet.

Til de enkelte entiteter knyttes der en række egenskaber, eller attributter om man vil. Med henblik på at undgå redundans må egenskaber kun fremgå ét sted, ved en relation eller en entitet. De enkelte egenskaber defineres i dataordbogen. Dataordbogen skal ikke indeholde informationer om hvilke relationer den pågældende entitet har – altså ingen fremmednøgler.

Hver entitet vil have flere forekomster, der alle har de samme egenskaber, men indholdet af egenskaber er forskelligt<sup>70</sup>.

Alle forekomster defineres entydigt af en nøgle<sup>71</sup> – nøglen vises i dataordbogen som understreget. En nøgle kaldes for en intelligent nøgle, hvis den ud over identifikationen siger noget om entiteten. Dette kan give problemer hvis den pågældende egenskab kan ændre sig, men det tager man som udgangspunkt ikke hensyn til ved essentielle modeller – det er et design problem.

Der findes en speciel type entiteter, der kaldes *afhængige entiteter*. De opstår når forekomsten af en entitet er betinget af forekomsten af en anden entitet. Et eksempel på en afhængig entitet kan være *rykker*, der er afhængig af en entitet, der hedder *betaling*.

---

<sup>69</sup> Begreberne er som udgangspunkt de samme som i Databaser på 1.år, men der er knyttet lidt uddybende forklaringer til nogle af dem.

<sup>70</sup> Dette svarer til den logiske og fysiske opdeling fra Databaser på 1. år

<sup>71</sup> Begrebet nøgle svarer i virkeligheden til den fysiske udformning som ikke hører hjemme i den essentielle model, men er altså brugt alligevel.

En afhængig entitet identificeres med en sammensat nøgle, der består af identifikationen fra den entitet den er afhængig af, kombineret med en specifikation. Det vil i ovennævnte eksempel være identifikationen fra *betaling* kombineret med en dato eller et rykker nummer.

Relationen mellem afhængige entiteter tegnes ikke med en rombe, men med en pil med et C på, der peger mod den afhængige entitet.

En anden form for struktur mellem entiteter er subtype/supertype sammenhænge. Disse sammenhænge forekommer, når en entitet kan antage flere forskellige typer af specialisering. I de tilfælde vil man oprette en supertype, der indeholder de generelle egenskaber, og en række subtyper, der indeholder de specialiserede egenskaber. Relationer til hierarkier af entiteter kan både gå til supertypen og til subtyperne afhængigt af den enkelte relation.

### Relationer

Relationer viser den betydningsmæssige sammenhæng mellem entiteter. De skal kun fremgå af diagrammet, hvis de har betydning for systemet, og de defineres efterfølgende i dataordbogen.

Relationer navngives med et udsagnsord udformet som navneord – eksempelvis *værelsesreservation*. Hvis man er så elendig, at man ikke kan finde et passende navn på relationen, kan den overtage navnet fra de entiteter, der indgår i relationen ☺. I diagrammet er relationer tegnet som romber.

En relation kan have flere forekomster, tilknyttet et antal entiteter. Disse forekomster har ikke selvstændige identifikationsfelter, men låner i stedet nøglerne fra de entiteter, der indgår i relationen. Der kan være knyttet egenskaber til relationer, som siger noget mere om relationen, end nøglerne fra de enkelte entiteter siger. Det kan eksempelvis være tilfældet, hvis man har en kontooverførselsrelation mellem to konto-typer. Her kunne man koble en egenskab på, der siger noget om beløbet.

Der findes tre typer af relationer kaldt kardinaliteter:

- Én til mange
- Mange til mange
- Én til én

Kardinaliteten viser sammenhængen mellem forekomsterne af de involverede entiteter og noteres med 1 og M.

Når man skal afgøre, om man skal arbejde med et bestemt begreb som entitet eller relation, kan man se på identifikationen. Hvis det ikke er muligt at identificere begrebet entydigt med lånenøgler, kan det være en idé at ”opgradere” relationen til at være en selvstændig entitet.

Det er ikke tilladt at lade to relationer være koblet direkte sammen – det vil give for mange lånenøgler.

Der findes en speciel typer relationer kaldt *rekursive relationer*. Dette er relationer, der opstår mellem forskellige forekomster af den samme entitet. En én til mange relation vil svare til en træstruktur og en mange til mange relation vil svare til et netværk.

### Beregnete data

Dette er data, der kan beregnes af andre data – eksempelvis saldo på en konto, der principielt kan beregnes ud fra summen af overførsler. Disse data medtages som udgangspunkt ikke i modellen, da de betragtes som kommunikationsdata. Undtagelsen til denne regel opstår, når flere forskellige processer skal anvende resultaterne.

### Alternative diagram former

Alternativerne til E/R diagrammet er at lave et Bachmann diagram, hvor der ikke indgår mange til mange relationer. Disse illustreres i stedet med to én til mange relationer til en ny entitet. Denne diagramform er mere mindet på det optimale database design, og ligger sig derfor mere op ad designet.

### 9.5.2. Illustration af virkemåde – kontrol af informationsanalyse

Formålet med informationsanalysen er at sikre at alle de relevante søgninger kan foretages, samt at alle de relevante ændringer kan registreres. Dette betyder imidlertid ikke at man skal lave 'nice to have' relationer. Der skal kun være det antal relationer der er nødvendigt. I praksis betyder det at man ikke skal lave relationer mellem entiteter hvis informationen fra den pågældende relation kan udledes af eksisterende relationer.

Under kontrollen af unødvendige relationer, skal man holde sig for øje, at anvendelsen af en relation er betinget af forekomsten af de enkelte entiteter. Det vil altså sige, at selvom man kan søge vej gennem tre forskellige relationer for at finde sit resultat, nytter det ikke noget, hvis man kan forestille sig denne søgning udført på et tidspunkt, hvor de relevante forekomster af entiteter ikke er oprettet. Det kan eksempelvis være tilfældet, hvis prisen på et abonnement er registreret i relationen mellem kundens adresse, standard leveringsbetingelser og det pågældende blad kunden abonnerer på. Her vil der opstå problemer, hvis kunden forespørger på prisen uden at have oprettet abonnementet. Et sådan tilfælde vil kræve andre relationer, eller ændring i egenskaber.

### 9.5.3. Strategier til informationsanalyse

Der er som udgangspunkt to måder at undgå redundans i informationsmodellen på:

- *Normalisering.* Denne metode er en mekanisk undersøgelse der går ud på at sikre overholdelse af 3. normalform: en egenskab skal knytte sig til identifikationsfeltet, kun identifikationsfeltet og intet andet end identifikationsfeltet.  
Metoden kritiseres netop for den mekaniske indgangsvinkel, hvor man ikke tænker over hvorfor man gør det – man gør det bare.
- *Semantisk informationsanalyse.* Denne strategi fører til det samme resultat som normalisering. Ideen er her at man analyserer nøjagtigt hvor en egenskab skal placeres for at give mest fornuft. En del af denne analyseteknik går ud på nøje at overveje hvad de enkelte databegreber dækker over. Kan man være kunde uden at have en konto? Slettes en konto når saldoen bliver nul? Hvornår ophører man med at eksistere som kunde osv. osv.  
Denne form for analyse anbefales til udarbejdelse af informationsmodellen, da den ligger godt i tråd med idéen om at udarbejde funktionsanalysen og informationsanalysen sideløbende.

### 9.5.4. Lokal eller global analyse – omfanget af informationsmodellen

Når man går i gang med informationsanalysen, skal man beslutte om den pågældende informationsmodel skal være en fælles model for hele virksomhedens systemer, eller kun for ét system. Hvis man vælger at lave en fælles model, skal man være opmærksom på at begreber kan opfattes forskelligt afhængigt af hvem der ser på dem. Eksempelvis vil en person blive opfattet som en kunde af sælgeren, men som en debitor hos regnskabsafdelingen.

Det vil altså sige at en global informationsanalyse, skal imødekomme krav til data fra flere forskellige systemer.

## 9.6. Tilstandsdiagrammer

Dette afsnit handler om formål og udarbejdelse af tilstandsdiagrammer. Et tilstandsdiagram viser hvilken tilstand en given entitet er i på et bestemt tidspunkt i en livscyklus, samt hvilke hændelser der bringer den fra en tilstand til en anden. Det er tiden, der er den afgørende faktor, da denne information ikke fremgår af de øvrige modeller.

Indholdet i tilstandsdiagrammerne udledes af funktions- og informationsanalysen, idet man holder for øje at man kun arbejder med tilstande man er interesseret i at registrere<sup>72</sup>. Det vil sige, at hvis man eksempelvis er interesseret i at registrere om en bil er afleveret på et værksted, skal den kunne antage en tilstand der hedder *afleveret på værksted* efterfulgt af *under reparation*. Hvis ikke man bruger den første tilstand til noget skal den ikke med. Det kunne være tilfældet hvis man kun vil registrere at kunden har bestilt tid på værkstedet – og man derfor er ligeglad med afleveringsregistreringen.

Tilstandsdiagrammer udarbejdes kun for de vigtigste entiteter. Entiteter der kun kan antage tilstandene *oprettet* og *slettet*, er uinteressante i denne sammenhæng.

Bogen nævner fire fordele ved tilstandsdiagrammer:

De skaber forståelse på et tidligt tidspunkt.

- *De kan bruges til kontrol af hvorvidt alle hændelser er husket.*
- *De kan bruges til kontrol af hvorvidt alle de ønskede tilstande kan registreres.*
- *De kan bruges til at udarbejde synkroniseringsregler*

De fire emner vil blive gennemgået yderligere under pkt. 9.2

### 9.6.1. Tilstandsdiagrammets symboler

Tilstandsdiagrammet indeholder to symboler:

En oval repræsenterer en tilstand

En pil repræsenterer en hændelse der medfører tilstandskifte.

Af dataordbogen skal det fremgå hvilke tilstande en entitet kan antage.

UML notationen er efter min mening bedre til tilstandsdiagrammer end den anført i bogen. Her åbnes der mulighed for start og slut markeringen, uden at dette er en tilstand. Endvidere mener jeg at man bør anføre hvilke egenskaber ved den enkelte entitet der ændres ved de enkelte tilstandskifte. Dette ligger ud over ISA fremgangsmåden, og skal således begrundes ved evt. anvendelse.

### 9.6.2. Tilstandsdiagrammets anvendelse

#### Forståelsen

Her tænkes der på kommunikationen med brugeren, der erfaringsmæssigt kan få et godt udbytte af tilstandsdiagrammer. I den forbindelse anbefales det også at man i kladde udarbejdelsen kan inddrage flere betydningsløse tilstande, da dette kan være en fordel til fremme af forståelsen. Jeg mener det er en uskik da det kan så tvivl om hvad der skal med i den endelige model og hvad der ikke skal.

Kontrol af informations- og funktionsanalysen

<sup>72</sup> Den sidste del er udledt af OOAD fra 2. år – alt logik siger dog at denne regel må være gældende såvel her som i OOAD.

Udarbejdelsen af tilstandsdiagrammerne kan give en anden indgangsvinkel til de øvrige diagrammer og kan derfor bruges som en kontrol instans til at sikre at man kan lagre de data der er nødvendige for at registrere de enkelte tilstande, samt om de hændelser der skal til for at skifte tilstand er til stede.

### Synkroniseringsregler

Synkroniseringsregler er regler for hvilken tilstand en entitet skal være i for at en hændelse må indtræffe. Disse kan i høj grad udledes af tilstandsdiagrammerne, og skal efterfølgende indgå i de enkelte primitive processers beskrivelse – der hvor de nu hører hjemme.

## 9.7. Dataordbog

Dette afsnit handler om dataordbogens indhold og opbygning. Dataordbogen er den del af værktøjerne der giver de detaljerede beskrivelser af alt det man ikke kan læse af diagrammerne. Indholdet kaldes dataspecifikationer og består af beskrivelser af:

*Datastrømme. Hver datastrøm i DS-diagrammet beskrives ud fra indhold og struktur*

*Datalagre – entiteter og relationer. Hvert datalager beskrives tilsvarende ud fra indhold og opbygning – dette gælder både entiteter og relationer.*

*Datagrupper. Datagrupper er grupperinger af data, der derefter kan genbruges i andre sammenhænge. Datagrupper beskrives ud fra deres opbygning og indhold.*

*Dataelementer. Dataelementer er de mindste data enheder i systemet – eksempelvis kontonummer, fornavn, alder, pris m.m.*

Datastrømme, datalagre og datagrupper kaldes sammensat data – disse beskrives mere uddybende nedenfor i afsnit 10.1. Dataelementer beskrives selvstændigt i afsnit 10.2

Selve udarbejdelsen af dataordbogen er beskrevet i afsnit 10.3

### 9.7.1. Sammensat data

For alle grupper af sammensat data gælder det at det skal fremgå, hvilke dataelementer der indgår, samt hvilken struktur de har. Der findes tre forskellige strukturer:

- *Sekvens.* Dette svarer til en rækkefølge af elementer
- *Selektion.* Dette svarer til et valg mellem forskellige muligheder af dataelementer
- *Repetition.* Her er der tale om en gentagelse af dataelementer.

En given mængde sammensat data kan bestå af flere forskellige strukturer inden i hinanden.

### Notation

Notationen for strukturer er således:

Symbol	Betydning
=	Består af, lig med
+	Og
[A   B]	Enten eller
0 { } 3	Gentagelse med nedre og øvre antalsgrænse
( )	Kan forekomme
**	Kommentar

Opstilling i dataordbogen ved brug af denne notation kræver nok lige lidt indøvelse, men man kan med fordel se på eksemplerne på side 106-107 i ISA-bogen.

### **Datagrupper**

Det kan være en fordel at samle data i datagrupper, for ikke at skulle gentage de samme beskrivelser flere gange. Det kan eksempelvis være tilfældet hvis der er mange datastrømme der sender kortoplysninger knyttet til en handel rundt. Der kan det være en fordel at definere et sted hvad kortoplysninger består af og derefter blot bruge dette begreb til de andre beskrivelser.

### **Undgå redundans**

Ved udarbejdelsen af dataordbogen skal man undgå at beskrive ting der fremgår af de øvrige diagrammer eller beskrivelser. Det vil sige at fremmednøgler ikke skal fremgå – de kan læses af E/R diagrammet, det samme gælder beskrivelse af hvilke datastrømme en datagrube indgår i m.m.

### **9.7.2. Beskrivelse af dataelementer**

Beskrivelsen af dataelementerne er inddraget for at sikre ens begrebsanvendelse i virksomheden. Ved nogle dataelementer der fremkommer ved simpel beregning kan det være en hjælp at angive formlen for frembringelsen, selvom dette principielt hører til under procesbeskrivelserne.

Der vil være dataelementer der er så entydige at de ikke skal beskrives – eksempelvis *alder*. Man skal dog være meget varsom med at udelade for mange beskrivelser.

Hvis der er dataelementer der defineres med forskellige navne forskellige steder i virksomheden, kan man vælge at give elementet en række synonyme. Det vil i praksis sige at man giver elementet et "hovednavn" og lader de andre synonyme henvise til dette element. Denne form bør begrænses til det minimale.

### **Udarbejdelsen af dataordbogen**

Oftentimes vil man være i stand til at finde eksisterende databeskrivelser i en virksomhed. Dette kan mindske arbejdet, hvis man kan overføre dette. Ellers anbefales det at man arbejder på dataordbogen sideløbende med alle de andre diagrammer.

---

## **9.8. Procesbeskrivelser**

Dette afsnit handler om procesbeskrivelser – hvordan de opbygges og hvilke valgmuligheder man har med hensyn til sprog.

Procesbeskrivelser er de præcise og detaljerede regler, der skal til for at danne uddata på baggrund af inddata. De udtrykker i høj grad virksomhedens forretningspolitikker for, hvorledes man ekspederer sager, behandler kunder og leverandører m.m.

Procesbeskrivelser laves kun for de primitive processer – de mindste enheder i DS-diagrammet. DS-diagrammet beskriver nøjagtigt, hvilke strømme der går til og fra processen, og dataordbogen beskriver indholdet af disse strømme. Denne information skal derfor ikke fremgå af beskrivelserne.

### **9.8.1. Procesbeskrivelsernes opbygning**

En procesbeskrivelse består af:

- Processens nummer

- Processens navn
- De regler der skal anvendes, for at processen kan gennemføre sit arbejde

Der er ingen formelle krav til, hvordan man beskriver behandlingsreglerne – blot de er entydige, komplette, korrekte og nemme at overskue og forstå.

Nedenfor er gennemgået to muligheder til sprogvvalg: struktureret sprog og almindeligt dansk. En tredje mulighed er pseudo-kode. Denne form er ikke gennemgået i bogen og har det minus at den ligger sig tæt op ad en teknisk løsning. Jeg mener dog at den vil være anvendelig, såfremt dem der skal bruge den har grundlæggende kendskab til de kode-termer man vælger at anvende. Dette skal derfor nøje overvejes ved dokumentation, der skal have en mere varig karakter – her må man kunne frygte at specielle kode-termer kan udgå over tid.

### 9.8.2. Struktureret sprog

Struktureret sprog består af de tre grundlæggende strukturer som også blev gennemgået ved dataordbogen<sup>73</sup>:

- *Sekvens*. Dette svarer til en fortløbende række af handlinger, der skal følge efter hinanden.
- *Valg*. Denne struktur forekommer, når der på baggrund af et kriterium skal gennemføres én af flere mulige handlinger<sup>74</sup>.
- *Repetition*. En handling der gentages et antal gange eventuelt afhængigt af et kriterium.

#### Sekvens

En sekvens består som udgangspunkt i den rækkefølge de enkelte linjer er skrevet i. Dette kan bestå af statements eller beregninger m.m.

#### Valg

Et valg sker på baggrund af en betingelse. De handlinger der knyttes til et valg rykkes ind, så det fremgår, at der er en sammenhæng.

#### Repetition

Gennemføres for et begreb eventuelt på baggrund af en betingelse. Eksempelvis:

```
For hver konto
  Hvis saldo er negativ
    Påfør gebyr
    Send gebyrnota
  Ellers
  ...
```

#### Komplicerede strukturer

De forskellige strukturer kan kombineres, så der opstår komplicerede procesbeskrivelser. Hvis dette er nødvendigt anvender man indrykning til at skabe overblik. Ovenfor er et eksempel på forskellige strukturer, hvor der indledningsvis gennemføres en repetition, den efterfølges af et valg, der kan lede til en selektion. Indrykningen viser overgangen til en ny struktur.

---

<sup>73</sup> Der er udmærkede eksempler på brugen af sproget i bogen S. 112-115

<sup>74</sup> Svarer til if () else() og switch

### Ordvalg i struktureret sprog

Ordforrådet i struktureret sprog er styret af få simple regler:

- Data beskrives ved hjælp af dataordbogens navne. Navne fra dataordbogen skrives med stort begyndelsesbogstav
- Udsagnsord skrives i bydeform. Eksempelvis *læs, beregn, skriv*
- Valg og repetitioner styres af betingelsesord. Eksempelvis *hvis, ellers, for hver*
- Betingelser indeholder typisk relationsudtryk. Eksempelvis *lig med, større end*

### 9.8.3. Almindeligt sprog

Anvendelsen af almindeligt sprog er ofte nemmere for brugerne end hvis man anvender struktureret sprog, der nok kan være nemt at læse, men som kan kræve lidt øvelse, hvis man skal skrive det.

Hvis man vælger at anvende almindeligt sprog, skal man blot være opmærksom på at skabe præcise formuleringer. Dette kan gøres ved at bruge navne fra dataordbogen, at skrive et aktivt sprog med brug af bydemåde og minimere fyldord samt ved at opdele behandlingsreglerne i punkter, der eventuelt gives en overskrift.

Man skal huske at få betingelserne for eventuelle repetitioner med ligesom man skal huske ellers grenen i valg.

### 9.8.4. Fremstillingen af procesbeskrivelser

Man kan godt blive overrasket over hvor lidt man faktisk skal skrive i en procesbeskrivelse. I dette afsnit gennemgås det kort hvilke relevante oplysninger der er at finde i de øvrige værktøjer, og hvad der derfor bør fremgå af beskrivelsen.

- *DS-diagrammet*. Heraf fremgår det hvilke indgående datastrømme der er, hvorfra der hentes data, hvor data lagres og hvilke uddata der er.
- *E/R-diagrammet*. Her kan man se hvilke relationer man kan anvende for at få adgang til de ønskede data.
- *Dataordbogen*. I dataordbogen kan man se definitioner af alle datastrømme og –lagre.

Det der ikke fremgår af de øvrige diagrammer er betingelserne for behandlingen af data. Hvornår skal data læses? Hvordan skal data beregnes? Hvordan dannes datastrømme? Hvornår skal data opdateres?

Det vil i praksis sige at man kan holde sig til disse regler:

- Man skal ikke skrive hvilke data der modtages fra andre processer, eller hentes i lagret
- Man skal beskrive hvordan data beregnes
- Man skal beskrive hvornår data læses på lagret for at indgå i behandlingen
- Man skal kun beskrive hvordan man danner en datastrøm, hvis det styres af betingelser

---

## 9.9. EDB-støttet systemudvikling

[Dette afsnit kan evt. læses ud fra PD forelæsningserne...Alternativet er de 5 sider der er i Delskov og lange]

## 9.10. Del III – basisstrategi for analysen

Dette afsnit handler om selve gennemførelsen af ISA. Indledningsvis bliver der gennemgået nogle filosofiske overvejelser, med henblik på at skabe en begrebsmæssig platform til den videre analyse gennemgang.

Selve analysestrategien vil blive gennemgået som en basisstrategi inddelt i syv overordnede emner – disse er gennemgået nedenfor i afsnit 14-20

### 9.10.1. To syn på et system

E/R diagrammet og DS-diagrammet er to grundlæggende forskellige måder at betragte det samme system på. DS-diagrammet viser dynamikken i et system, og udtrykker de handlinger der sker. E/R Diagrammet viser status, efter handlingen er sket. Dette kaldes også med fine ord for det *procedurale* og det *non-procedurale* syn på systemet. Det procedurale syn ser på dynamikken, og det non-procedurale ser på det statiske.

### 9.10.2. Hændelsesopdeling af den procedurale side

I den essentielle model er systemet delt op i essentielle aktiviteter, der svarer til den mængde processer, der sker på baggrund af en hændelse. En hændelse kan enten være tidsbestemt eller komme fra systemets omverden. De essentielle aktiviteter karakteriseres som den procedurale sides stabile elementer. Dette skal fremgå af høj-niveau diagrammerne, hvor disse aktiviteter illustreres som processer, der kun relateres indbyrdes via datalagrene – man kan betragte dem som små selvstændige systemer.

#### Nedbrydning af essentielle aktiviteter

De essentielle aktiviteter nedbrydes på de underliggende niveauer på deres egne aktivitetsdiagrammer. Her vil en proces enten starte med en trigger fra omverdenen, eller ved at hente data fra lagret. Disse to muligheder opstår, fordi de essentielle aktiviteter ikke kommunikerer indbyrdes.

#### Udgangspunkt for analyse

Man kan vælge mellem to muligheder når man skal lave sin ISA. Man kan starte med informationsmodellen, og lade den være udgangspunktet for en analyse af processer og hændelser. Alternativet er at starte med at se på hændelserne, og de krav de stiller til informationsmodellen.

Begge muligheder er lige gode, men det anbefales at man vælger en kombination. Indgangsvinklen ved at starte med informationsmodellen er, at det heraf fremgår, hvilke processer der skal til for at oprette forekomster af de enkelte entiteter og relationer. Dette kan føre til en analyse af, hvilke hændelser der iværksætter disse aktiviteter, og dermed er man i gang med funktionsanalysen. Denne indgangsvinkel kaldes entitets-orienteret.

Hvis man starter med at analysere hændelserne, vil informationsanalysen tage udgangspunkt i de krav, der stilles til informationsmodellen. Denne indgangsvinkel kaldes hændelses-orienteret.

### 9.10.3. Livscyklus

I den essentielle model repræsenterer informationsmodellen systemets hukommelse. Hermed menes der, at modellen lagrer information om alle de hændelser, der er indtruffet på det tidspunkt, man betragter modellen. Dette medfører også, at relationer ikke kan aflæses før de er oprettet.

Entitetens livscyklus

I forlængelse af afsnittet ovenfor, kan man sige, at systemet skifter tilstand, når det påvirkes af en hændelse. Det vil sige, at systemets tilstand repræsenterer summen af alle entiteternes tilstande.

Udarbejdelsen af tilstandsdiagrammer sker netop for at kunne forudsige disse tilstandsskifte. Ved den procedurale analyse anvendes tilstandsdiagrammerne til at identificere hændelser. Ved den non-procedurale analyse anvendes tilstandsdiagrammerne til at vise de tilstande, der skal indbygges i informationsmodellen.

#### **9.10.4. Aktiviteternes rangorden**

De essentielle aktiviteter kan rangordnes efter den evne, de har til at skaffe analytikerne viden om systemet. Denne orden udtrykker ikke vigtigheden for virksomheden.

1. Aktiviteter der opretter forekomster af entiteter og relationer
2. Aktiviteter der skifter tilstand på entiteter
3. Aktiviteter der tilføjer eller ændrer egenskaber
4. Aktiviteter der sletter forekomster af entiteter eller relationer
5. Aktiviteter der kun læser

Aktiviteter der sletter forekomster, vil ofte fungere således at de markerer en entitet for udgået. Senere vil der køre aktiviteter, der sletter som en form for batchkørsel. Hvis der er tale om en aktivitet, der markerer entiteten for udgået, skal den rangordnes som om den skifter tilstand på entiteten – tilsvarende er de egentlige slette aktiviteter rangordnet som nummer fire.

Ved at gennemføre analysen efter denne orden, får man hurtigst muligt information om systemet.

#### **9.10.5. Integreret systemanalyse**

Gennemførelsen af ISA kan gøres på flere forskellige måder – som beskrevet ovenfor. Om man starter med funktionsanalysen eller informationsanalysen, bør komme an på de forudsætninger man har som udgangspunkt, og den analyseform man vurderer i det konkrete projekt, der vil fungere bedst.

Figur 13.6 i *Struktureret analyse* S. 136 viser et typisk udviklingsforløb – slå op på den! Heraf fremgår det, at der er en række forskellige aktiviteter, der skal gennemføres i forbindelse med systemudviklingen. ISA beskæftiger sig kun med selve analysefasen, som bliver brudt ned i en basisstrategi, der ofte har vist sig at være effektiv. Denne strategi er ikke styrende for brugen af ISA – den viser blot en rækkefølge, der kan virke. Det er således muligt at vælge de samme teknikker, men en anden strategi og opnå et tilsvarende godt resultat.

##### **Analysefasen**

Figur 13.7 og 13.8 S. 138-139 viser et mere detaljeret billede af analysefasen. Det er disse modeller, der danner grundlag for udledning af basisstrategien:

1. Analyser eksisterende system (Afsnit 14)
2. Kortlæg hændelser (Afsnit 15)
3. Udarbejd informationsmodel(Afsnit 16)
4. Analyser essentielle aktiviteter(Afsnit 17)
5. Integrer modeller(Afsnit 18)
6. Indarbejd nye krav(Afsnit 19)
7. Udarbejd ny fysisk model(Afsnit 20)

## 9.11. Fysisk model af eksisterende system

Udarbejdelsen af en model af det eksisterende system tjener to formål:

- Etablering af gensidig forståelse og tillid mellem brugere og edb-folk
- At nemme arbejdet med den essentielle model ved at samle og systematisere viden om systemet

Udarbejdelsen af en komplet model vil være meget omfattende, og man bør således gøre sig nogle overvejelser om, hvor meget man ønsker at få med. Hvis man har god indsigt i aktiviteterne, kan man vælge helt at springe over og gå direkte til udarbejdelsen af den essentielle model. I springet fra gammel fysisk model til essentiel model, skal man huske, at den essentielle model skal abstrahere fra ”den gamle måde” at gøre tingene på.

### 9.11.1. Hvor meget?

Analysen af den eksisterende fysiske model bør fokuseres på de svært komplicerede dele. Ofte vil man støde på store mængder af redundante data, der ikke nødvendigvis skal inddrages.

### 9.11.2. Hvor detaljeret?

Det væsentligste er, at man kan komme ind til det essentielle i modellen. Det vil ofte kræve, at man går indgående ind i beslutningsprocesser og beregningsalgoritmer, da disse vil være de samme under udarbejdelsen af den essentielle model.

Tilsvarende kan man med fordel ligge ressourcer i datastrømme til omverden – de vil også være de samme i den essentielle model.

Datalagre bør kun beskrives meget overordnet, da disse vil blive analyseret i den semantiske informationsanalyse. At basere sin informationsanalyse på den fysiske model, svarer til en normaliseringsstrategi, der ikke kan anbefales.

### 9.11.3. Hvilken opdeling?

Der er grundlæggende to forskellige måder at dele sin fysiske model op på:

- *Processorer*. Dette svarer til en organisatorisk opdeling, hvor en processor er en maskine eller et menneske. En boble vil her svare til en person, en maskine, en afdeling eller et system.
- *Hændelser*. Denne modelform er gennemgået nedenfor i afsnit 15.

Det skal fremgå tydeligt, hvilke processer der udføres manuelt, og hvilke der udføres maskinelt. Tilsvarende skal man anføre, hvilken processor der udfører arbejdet, hvis man vælger en hændelses orienteret model.

Det anbefales at anvende en hændelsesorienteret model, hvis man kun bruger den fysiske model som et forstadium til den essentielle model. Ellers karakteriseres den som en hybrid mellem en fysisk model og en essentiel model.

---

## 9.12. Hændelsesopdeling

Dette afsnit handler om, hvorledes man udleder de forskellige hændelser i systemet. Indledningsvis gennemgås det, hvordan man finder dem i en fysisk model, derefter hvilke faldgrubber der er ved at identificere essentielle aktiviteter.

### **9.12.1. Hændelser i fysisk model**

Ved en hændelsesopdelte fysisk model vil de essentielle aktiviteter være samlet, her er der altså ikke det helt store arbejde med at finde dem.

I den processor opdelte fysiske model, vil de essentielle aktiviteter være spredt over flere forskellige afdelinger, hvorfor analysearbejdet er lidt mere omfattende. Det anbefales, at man starter med at finde essentielle aktiviteter til de eksterne hændelser – triggere der igangsætter en essentiel aktivitet i systemet. Metoden er herefter at finde en trigger, og derefter følge den række af aktiviteter, der sker, indtil man støder på et datalager, eller en ekstern datastrøm.

Når alle aktiviteter, der er knyttet til de eksterne triggere, er blevet identificeret, kan man starte med de resterende – det er dem, der bliver iværksat af tidsbestemte hændelser.

### **9.12.2. Hændelser i essentiel model**

Når alle hændelserne er kortlagt kan man begynde at tegne oversigtsdiagrammet. Her skulle man gerne kunne tegne alle de essentielle aktiviteter uden tværgående datastrømme. Fordelen ved denne form for opdeling af aktiviteter, hvor der ingen tværgående kommunikation er, er at man mindsker antallet af systemgrænseflader. Dette er en fordel, til trods for at der bliver mange datalager-grænseflader, da disse ikke betragtes som noget problem at lave.

### **9.12.3. Non-stop arbejdsgange**

Essentielle aktiviteter er samlingen af de processer, der reagerer på en hændelse. Dette fører til en konklusion om, at alle processerne udføres non-stop. I den fysiske model, vil man ikke kunne garantere at alt udføres non-stop, men i den essentielle model svarer de essentielle aktiviteter til den samling af processer, der udføres non-stop, når der ses bort fra teknologiske begrænsninger.

I den fysiske model svarer en aktivitet til den mængde processer, der rent faktisk kan udføres non-stop i systemet – det gamle såvel som det nye.

Nedenfor gennemgås en række faldgrubber i forbindelse med definerings af essentielle aktiviteter.

### **9.12.4. Mellemlagring**

Ved gennemgangen af den fysiske model, skal man ikke betragte en mellemlagring som afslutningen på en essentiel aktivitet, hvis denne har karakter af at være lagring til en batchkørsel eller lignende. I sådanne tilfælde skal det pågældende lager ikke med i den essentielle informationsmodel, og rækken af processer tilknyttet den pågældende essentielle aktivitet fortsætter ”på den anden side” af lagret.

Undtagelsen til denne fremgangsmåde er, hvis mellemlagringen er en del af en forretningspolitik – eksempelvis hvis der sendes faktura ud en gang om måneden og data skal lagres til denne aktivitet. Her vil der være tale om to hændelser – en trigger fra omverden og en tidsbestemt hændelse.

En af årsagerne til denne form for mellemlagring kan være eksisterende grænseflader mellem forskellige processorer – eksempelvis i form af indbakker eller blanketter, der senere skal indtastes i et system.

### **9.12.5. Forsendelse**

Som udgangspunkt bør fysiske ting ikke figurere som en del af en fysisk eller essentiel model. Der er dog den undtagelse, hvor behandlingen af den fysiske vare er en del af systemet, og hvor fremtidig teknologi ikke kan tænkes at muliggøre flytning indenfor systemet. I sådanne tilfælde vil man lave en eks-

tern datastrøm til en ekstern proces – eksempelvis Postdanmark. Fra denne eksterne proces kommer en trigger til en anden intern proces, der hvor varen modtages. I den essentielle model, bør dette ikke foretages med almindelige breve, da de kan tænkes overført internt i systemet.

### 9.12.6. Non-stop dialog

Dialog defineres som kommunikation med omverden – altså datastrømme til eksterne processer. Disse skal betragtes som én aktivitet, hvis dialogen ikke kan afbrydes. Ellers betragtes de som hver sin aktivitet.

### 9.12.7. Fælles proces

En fælles proces er en proces, der deles af flere aktiviteter. I diagrammet skal den markeres med en stjerne, og den skal inddrages på alle de diagrammer, hvor den udfører arbejde. Der udarbejdes dog kun en procesbeskrivelse.

Man skal være opmærksom på at man ikke erklærer en proces for en fælles proces på for højt et diagram-niveau. Hvis en proces kun udfører en del arbejde for den ene hændelse og en anden del arbejde for en anden hændelse, skal fælles processen nedbrydes yderligere inden den erklæres for fællesproces.

### 9.12.8. Andre faldgrubber

Hvis man anvender kontekstdiagrammet til at identificere triggere fra eksterne hændelser, skal man være opmærksom på at:

- En trigger kan bestå af flere datastrømme, der i så fald vil følges ad
- Nogle af de eksterne datastrømme kan være non-stop dialog – og altså ikke triggere jf. ovenfor.
- Datastrømme der er tegnet som én pil kaldes en datakanal. Sådan en kan indeholde flere triggere.

### 9.12.9. Kortlægning af essentielle aktiviteter

Som beskrevet ovenfor i afsnit 14, vil det være en stor hjælp, hvis man har udarbejdet en gammel fysisk model, når man skal analysere essentielle aktiviteter. Dog er der det problem, at modellen oftest ikke er hændelsesopdelt. I disse tilfælde har man ikke noget at anvende de øverste niveauer til – det er kun de nederste niveauer, der er interessante. Dette fører til et problem med en uoverskuelig mængde diagrammer, når alle processerne er brudt ned, hvorfor man med fordel kan vælge mellem disse to muligheder:

- *Tegne en ekspanderet fysisk model*, hvor alle de laveste diagram-niveauer er tegnet på et diagram. Denne løsning er meget omfattende ved større systemer.
- *Skrive henvisninger på diagrammerne*. Dette gøres ved at skrive procesnummeret på afsender eller modtager processen for alle datastrømme, der går ud eller ind i det pågældende diagram.

#### Kortlægningen

Kortlægningen af de eksterne hændelser sker ved at følge en trigger fra omverden gennem alle de aktiviteter, der påvirkes i non-stop kørsel.

Tidsafhængige aktiviteter kan findes ved at søge i den modsatte retning, når de eksterne hændelser er fundet. Ved denne metode finder man alle de datastrømme, der går ud af systemet og følger dem tilbage gennem en række af aktiviteter. På side 155-160 gennemgås et eksempel på udledning af hændelser i en gammel fysisk model.

### **9.12.10. Analyse uden gammel fysisk model**

Hvis man står uden gammel fysisk model, kan man gennemføre nøjagtigt den samme procedure ved at observere brugere og tale med dem om arbejdsgange.

### **9.12.11. Analyse af maskinelle systemer**

En ændring af et eksisterende maskinelt system, vil ofte medføre en større ændring for dem der bruger det end en omlægning fra manuelt til maskinelt system – til trods for at det kaldes vedligeholdelse og ikke nyudvikling.

Problemet for analytikeren opstår når vedkommende skal analysere et gammelt maskinelt system. Her risikerer man at systemet behandler flere forskellige triggere i den samme datastrøm og på den måde blander hændelserne sammen – man er altså nød til at se i gammel dokumentation eller kildekode for at finde essentielle hændelser.

### **9.12.12. Analyse af integrerede systemer**

Ved en analyse af en gammel fysisk model, skal man være meget opmærksom på integration med andre systemer. Datalagre der er fælles med andre systemer, skal tegnes på kontekstdiagrammet. Tilsvarende skal datastrømme til og fra eksterne processer tegnes på.

### **9.12.13. Basisstrategien for kortlægning af hændelser**

Som konklusion kan det ridses op at denne fase i basisstrategien består af tre aktiviteter, der udføres iterativt:

- Identificer hændelser. Denne aktivitet resulterer i en hændelsesliste<sup>75</sup>.
- For hver hændelse kortlægges den essentielle aktivitets forløb. Denne fremstilling kan ske ved at markere aktivitetens forløb i den gamle fysiske model, hvor sammenhørende aktiviteter tegnes med en bestemt farve.
- Aktiviteternes kommunikation med omverden fastlægges. Resultatet af denne fase er et hjælpe DS-diagram, som kan bruges som del resultatet i den videre analyse.

---

## **9.13. Informationsanalyse-teknik**

Der er ikke noter fra dette kapitel i Delskov og Lange. De skal læses i bogen fra s. 165.

---

<sup>75</sup> Se eksempel i ISA appendiks B3

## 10. SOCIO-TEKNISK ANALYSE OG ETHICS

---

### 10.1. Ethics

Pensum: Avison & Fitzgerald S. 353-364 (DIS kompendium S 104-110)

#### 10.1.1. Indledning

Dette afsnit handler om udarbejdelsen af socio-teknisk analyse samt systemudviklingsmetoden ETHICS.

ETHICS er en metode til systemudvikling hvor nøgleordene er brugerdeltagelse og i det hele taget fokus på det socio-tekniske paradigme, altså at det i høj grad er brugerne, der får lov at bestemme hvordan systemet skal udformes<sup>76</sup>. Dette skulle gerne resultere i en forbedring af medarbejdertilfredsheden. Medarbejdertilfredsheden defineres her som:

”...a good fit between what the employee is seeking from his work...and what he is required to do...”<sup>77</sup>

Forfatteren opsætter 5 faktorer til at måle graden af det gode “fit”:

- The knowledge fit – medarbejderen føler at hans færdigheder og viden bliver udnyttet, samt at han får mulighed for personlig udvikling.
- The psychological fit – jobbet skal passe til medarbejderens status og interesser (jfr. Herzberg’s motivationsteorier)
- The efficiency fit – dækker 3 områder:
  - Effort-reward bargain - hvor meget er arbejdsgiver villig til at betale for arbejdskraften
  - Work controls - graden af kontrol skal passe til medarbejderens forventninger
  - Supervisory controls - adgang til back-up, sparring og ekspertise
- The Task-structure fit – føler medarbejderen at hans opgaver er tilstrækkeligt krævende og tilfredsstillende, samt at han får tilstrækkeligt med tilbagemelding
- Ethical fit – stemmer medarbejderens værdiopfattelse overens med virksomhedens?

Den store grad af medarbejderdeltagelse under udviklingen bliver ofte af ledelsen opfattet som en måde at opnå ting og beslutninger, som ellers ikke var kommet igennem. Forfatterens holdning til dette er, at selvom det ikke er den optimale holdning, så er det bedre at bruge ETHICS i brudstykker end slet ikke. Mht. medarbejderdeltagelse (participation) skelner Mumford<sup>78</sup> mellem structure, content og process:

- Structure – strukturen kan være både konsultativ, repræsentativ eller enighed (consensus)
  - Konsultativ – deltager kommer med idéer, men udviklerne bindes ikke af disse
  - Repræsentativ – repræsentanter for forskellige interesser vælges, fx til udarbejdelse af en system definition.
  - Consensus – hvor alle deltager, dette deles dog i praksis op i flere grupper.

---

<sup>76</sup> Synspunktet ved denne tekst må findes i den nedre halvdel af Burrell & Morgan’s Paradigme-matrix (se Hirschheim & Klein), hvor udvikleren ikke nødvendigvis tager ledelsens side, men kan vælge at lægge udviklingen ud til medarbejderne. Brug evt. her trykkeri-eksemplet ”UTOPIA” (s.11)

<sup>77</sup> IB-kompendium, s.104

<sup>78</sup> Mor til ETHICS

- Content – drejer sig om hvilke aktiviteter og emner, der er indenfor brugernes beføjelser/(ansvars)område.
- Process – et eksempel kunne her være at sikre brugernes adgang til relevant viden. Ægte brugerdeltagelse betyder lige magt og adgang til viden for alle involverede grupper. Uddannelse af brugere er derfor essentielt i ETHICS.

Der findes flere versioner af systemudviklingsmodellen for ETHICS. Teksten gennemgår Mumford's 15-trins model<sup>79</sup>, der udspringer af en socio-teknisk analyse.. Modellen vil her blive gennemgået i overskrifter, idet der drages paralleller til de øvrige metodiske muligheder:

1. *Why change?* – Designgruppens første møde, hvor problemstillinger og muligheder diskuteres. Resultatet af denne fase er en erkendelse af de nødvendige ændringsbehov. Denne fase modsvarer af de indledende faser i SSM, hvor man erkender den ustrukturerede problemsituation. Umiddelbart virker SSM mere uddybende i denne fase, da den bruger en højere detaljeringsgrad i udarbejdelsen af de nødvendige ændringsbehov (konceptuelle produkter).
2. *System boundaries* – Gruppen identificerer systemets grænser/interfaces. Endvidere identificeres fire forskellige områder, der bliver berørt af ændringerne: Forretningsaktiviteter, eksisterende teknologi, afdelinger i organisationen og organisationens omverden. Dette svarer i høj grad til den grundlæggende struktur i det rige billede, udarbejdet under SSM.
3. *Description of existing system* – Gruppen uddannes i, hvordan det nuværende system fungerer og det analyseres vertikalt og horisontalt. Denne fase kan sammenlignes med udarbejdelsen af en gammel fysisk model i ISA og den efterfølgende udarbejdelse af den essentielle model.
4. *Definition of key objectives and tasks* – Dette punkt dækker reelt over tre faser. Formålet er at finde de primære målsætninger som en ændring skal imødekomme, og dette gøres ved at stille tre spørgsmål:
  - Hvilke forretningsområder/afdelinger er der – hvad er deres rolle og formål?
  - Hvad er deres ansvar, og hvilke funktioner udfører de?
  - I hvilken grad indfrier de nuværende aktiviteter de forventede aktiviteter?

Denne række spørgsmål bør føre til en liste af primære målsætninger, som det kommende system skal indfri. I forlængelse heraf udledes de opgaver, der skal gennemføres for at indfri disse målsætninger, og det tilhørende informationsbehov kortlægges.

Denne fase svarer til det at udlede primary tasks når man beskriver HAS i SSM.

5. *Diagnosis of efficiency needs* – svage led i det eksisterende system identificeres, med henblik på at finde tekniske behov. Det der her er tale om er en analyse af det eksisterende system med henblik på at finde steder, hvor systemet ikke handler i overensstemmelse med forventningerne. Disse uoverensstemmelser kan groft set antage to former: naturlige eller operationelle. De naturlige er uoverensstemmelser der ligger ”bag” systemet – eksempelvis uoverensstemmelsen mellem det at minimere lagerbeholdning og have korte leveringstider. Denne form vil man aldrig kunne komme til livs. De operationelle uoverensstemmelser stammer fra et dårligt design eller manglende tilpasning til ændrede forretningsgange. Denne form kan ifølge teksten fjernes totalt, men det ligger nok ikke helt i tråd med SSM og participation tankegangen at man får beskrevet systemets omgivelser så godt at der ikke opstår uoverensstemmelser mellem menneske og maskine.

Hvis denne fase udbygges med mulige løsninger på de tekniske problemer, kan den sammenlignes med den tekniske gren af den socio-tekniske analyse, hvor det netop er tekniske målsætninger der er emnet.

---

<sup>79</sup> IB-kompendium, s.107-109

6. *Diagnosis of job satisfaction needs* – tilfredsheden måles vha. et ETHICS-spørgeskema, som typisk tilpasses organisationen<sup>80</sup>. Til dette formål er der udarbejdet et specielt standardskema man kan tage udgangspunkt i, når man skal måle tilfredsheden<sup>81</sup>.  
Resultatet af denne analyse er en række målsætninger for de sociale aspekter, hvortil der knyttes mulige løsninger. Man skal holde for øje at ikke alle tilfredshedsproblemer skal løses med systemer – nogle kræve ændret personalepolitik m.m.  
Dette punkt svarer til socio-grenen af den socio-tekniske analyse.
7. *Future analysis* – gruppen vil forsøge at identificere potentielle ændringer og indbygge fleksibilitet overfor disse.
8. *Efficiency and job satisfaction* – her bliver effektivitets- og jobtilfredsheds mål og behov specificeret og vurderet. Mumford betegner denne som den vigtigste fase i metoden. Målene specificeres ud fra den foregående diagnosticering og disse bliver prioriteret. Det er væsentligt at så mange som muligt involveres i denne proces.  
Afslutningsvis opstilles målene som primære og sekundære mål, idet man holder fokus på de primære målsætninger man tidligere har udledt. Et kommende system skal indfri alle primære mål, og så mange af de sekundære som muligt.  
Denne fase svarer til det punkt i socio-teknisk analyse, hvor man holder tekniske og sociale målsætninger op mod hinanden, med henblik på at finde dem der bedst indfrier de enkelte målsætninger.
9. *The organisational design* – denne fase bør udføres parallelt med de tekniske valgmuligheder, da begge er tæt knyttet. Her specificeres de organisatoriske ændringer, der skal til for at opnå de for definerede mål. Der arbejdes videre med 3 til 6 muligheder, som bedømmes af medarbejderne ud fra kriterier om jobtilfredshed. Her pointerer forfatteren at ETHICS er en direkte modsætning til synspunkterne bag Taylorism, der udvælger nye systemer på grundlag af økonomisk optimering.  
I forhold til socio-teknisk analyse er der her den forskel at på dette tidspunkt er der ikke truffet beslutning omkring valg af teknologi – det er i ETHICS et design spørgsmål. I socio-teknisk analyse vil der på dette tidspunkt være truffet beslutning omkring hvilken teknologi en løsning skal designes i. Endvidere er der den forskel at løsninger udtænkes til den prioriterede liste af målsætninger – dette kan fjerne fokus fra de lavere prioriterede, hvorved man kan overse en række løsninger, der derved ikke bliver overvejet.
10. *Technical options* – forskellige muligheder vurderes. Denne vurdering omfatter hardware, software, kodesprog og design af brugergrænseflader m.m.  
De enkelte løsningsmuligheder sammenlignes med krav til jobtilfredshed, primære forretningsmålsætninger m.m.  
Slutteligt vælges den kombination af organisatoriske og tekniske muligheder, som bedst indfrier målsætningerne på den prioriterede liste.
11. *Detailed work design* – det valgte system er nu designet i detaljer (data flows, opgaver, grupper, individer, ansvar og relationer). Dette design reviews for at sikre at det stadig modsvarer de stillede krav.
12. *Implementering*
13. *Evaluering* – systemet checkes at det opfylder krav/mål, specielt i relation til tilfredshed.

Forfatteren forsøger at gøre op med myten om at brugere ikke har de fornødne færdigheder til at designe et system og at ledelsen ikke vil acceptere den ”magt-afgivelse”. Brugere kan nemt modtage den fornødne træning og derudover har de det største kendskab til deres arbejdsopgaver og er ”stakeholders”. Mumford pointerer at metoden ikke er let, men at den er besværet værd. Som eksempel på et

---

<sup>80</sup> Se forrige sides definition af tilfredshed som ”det gode fit”

<sup>81</sup> Se DIS kompendium S 111-120

projekt, der viser ETHICS' styrke nævnes en gruppe af kontorarbejdere, der udviklede 3 forskellige systemer til at håndtere den samme problemstilling på. Således var medarbejdertilfredsheden (her opnået gennem afveksling) prioriteret højere end økonomi.

## 11. SUPPLERENDE LITTERATUR

---

### 11.1. Budde et. Al.

Pensum: Budde et. Al., Prototyping pp. 33-48 (DIS S. 121-129)

#### 11.1.1. Prototyping

Prototyping beskrevet i denne tekst læner sig meget op ad 2.års pensum. Jeg vil derfor kun kort beskrive teksten, da den ikke bringer meget nyt. Teksten kan sagtens springes helt over, da den ikke skriver noget relevant. Halvdelen af den er et eksempel fra "the real world". :o(

#### Prototyping i softwareudviklingsprocessen

Der findes forskellige slags prototyper:

*Breadboard prototyping* – designes til at opklare tekniske vanskeligheder, relateret til udviklingsprocessen, men anvendes ikke i det endelige system

*Pilot system* – en prototype, der anvendes i det endelige system, som en slags "føler" for at vurdere om man er på rette spor.

Et pilot system kræver naturligvis langt mere udvikling end et breadboard system, da det skal implementeres, og ikke bare tegnes grafisk.

#### Mål med prototyping

Udviklingen af prototypen kan også gribes an på forskellige måder:

- Eksplorativ udvikling – Denne bruges når man ikke helt kender problemets omfang. De indledende skitser hjælper udvikleren og brugeren til at sikre at man er på rette spor. Ellers kan man hurtigt foretage ændringer.
- Eksperimentiel udvikling – Denne type fokuserer på den tekniske implementering af et udviklingsmål. Ved at eksperimentere kan brugeren hen af vejen specificere sine krav yderligere.
- Evolutionær udvikling – Ved denne 3. og sidste type, bruger man prototypen mere overordnet. Ikke kun til det enkelte udviklingsprojekt, men nærmere som et redskab der hurtigt kan omstille sig ved organisatoriske ændringer og anvendes over et længere forløb.

#### Horisontal og vertikal prototyping

Forskellen på de to er at i horisontal prototyping laver man oftest blot det grafiske udtryk, ofte som "paper mock-up", altså kun et enkelt lag af systemet.

I vertikal prototyping vælger man en "gren" af systemet og gennearbejder den så den har fuld funktionalitet.

## 11.2. Bansler & Bødker

Pensum: Bansler og Bødker, A reappraisal of Structured Analysis: Design in an Organizational Context. pp. 165-193 (DIS S.130-145)

### 11.2.1. Introduktion

Tekstens handler om forskellen mellem struktureret analyse som beskrevet i litteraturen og hvordan den i virkeligheden tager sig ud i "real life" organisationerne i praksis. Deres tese er at struktureret analyse i virkeligheden ikke anvendes som beskrevet i metoden. Når der her tales om struktureret analyse er det den oprindelige metode udarbejdet af Yourdon og DeMarco.

Først undersøges de implicite formodninger der er udtrykt i metoden såsom beskaffenheden af virksomheder samt hvorledes værktøjerne skal bruges. På baggrund af nogle studier omkring metodens anvendelse i praksis kommer de frem til, at der er et missing link mellem metoden som den præsenteres i litteraturen og som den føres ud i livet.

Gennem 80'erne har struktureret analyse været genstand for megen kritik og alligevel er det en af de mest velkendte og brugte metoder til udvikling af informationsbehandlingssystemer i Danmark.

### 11.2.2. Principperne i struktureret analyse

Struktureret analyse (SA) er blevet modificeret adskillige gange siden Yourdon og DeMarco første gang præsenterede metoden. En af de seneste modifikationer vi har stiftet bekendtskab med, er af Delskov og Lange. I alle versioner eller afskygninger af SA er de grundlæggende principper og formål dog nogenlunde de samme. De er beskrevet udførligt i ISA<sup>82</sup>, og alle værktøjerne vil derfor ikke gengives her<sup>83</sup>, men i stedet lægges der vægt på eventuelle forskelle mellem SA og ISA.

Nedenfor ridses grundprincipperne op.

Der anvendes i SA forskellige diagrammer til at modellere en organisation som et informationssystem. Det mest vigtige nævner de som værende Dataflow diagrammet, idet det anvendes som brobygger mellem brugerne og designerne<sup>84</sup>. I den oprindelige SA er der mest fokus på funktionssiden, hvilket er en mangel i forhold til ISA.

Ifølge SA skal man udarbejde fire forskellige produkter, 1) den gamle fysiske model, 2) den gamle logiske model, 3) den nye logiske model og til sidst 4) den nye fysiske model, for at gennemføre en komplet SA. I forhold til ISA er forskellen, at den gamle og nye logiske model er erstattet af den essentielle model. Hvert step bør ifølge SA (den gamle metode) bestå af en komplet beskrivelse i form af Dataflowdiagrammer, dataordbog og mini-specs (jeg tror det er procesbeskrivelser). DeMarco skelner altså mellem for det første modeller der beskriver det gamle og det nye system, og for det andet modeller der beskriver fysiske og logiske systemer. Det mangelfulde i DeMarcos metode er transformeringen mellem de forskellige faser, fra fysisk og logisk samt mellem gammel og ny. Palmer og McMenamin mener, at analysen kun bør afføde tre produkter fra de forskellige faser. Det er her begrebet essentiel model stammer fra.

---

<sup>82</sup> Integerret struktureret analyse af [Delskov og Lange, 1999]

<sup>83</sup> Bansler og Bødker side

<sup>84</sup> Her mener vi dog, at diagrammet i den henseende har sin begrænsning. Det er rigtigt at det er bedre at anvendes som kommunikationsmedie mellem brugere og designere fordi de er grafiske, men selv et dataflow diagram mener vi ikke at brugerne særlig nemt kan sætte sig ind i. Til dette formål anvender vi vores løst koblede tankegang som medfører prototyper.

Den essentielle model svarer til en slags logisk model, den er bare mere vel defineret end tidligere. Den essentielle model er teknologiafhængig og beskriver ikke hvordan, hvornår og af hvem en proces skal udføres blot hvad der skal udføres.

Som udgangspunkt for analysen til den essentielle model, anbefaler Palmer og McMenamin at gennemføre et blitzing forløb. Det sætter konsulenten i stand til at udarbejde en detaljeret analyse af relevante dele af det gamle system. Ifølge Palmer og McMenamin kan hele den fysiske model dog ikke undgås fordi den skal bruges til at verificere og færdiggøre den essentielle model. Yourdon siger, at en gammel fysisk model ikke skal, men kan udarbejdes såfremt den skal anvendes til at klarlægge hvad de essentielle processer er .

Selvom anbefalingerne afviger lidt fra hinanden er grundprincipperne de samme. Det første princip er, at modellere organisationer som informations proces systemer (dataflow). Det andet er at skelne mellem fysiske og logiske/essentielle aspekter. Og det tredje, lidt omdiskuterede princip er at tage udgangspunkt i en analyse af det gamle fysiske system til udledning af den essentielle model.

### **11.2.3. Organisationer og arbejde ud fra et struktureret analyse perspektiv**

Struktureret analyse danner et billede af en organisation som et informationsbehandlingssystem, i form af et netværk af processer som udveksler information i henhold til en række regler. Ifølge dette udsagn og som beskrevet ovenfor, er formålet at producere en detaljeret funktionel eller logisk beskrivelse af opgaver og operationer med fokus på flowet og processing af information.

Som beskrevet i paradigme teksterne<sup>85</sup> er der en vigtig og umiddelbar konsekvens af dette perspektiv, nemlig at mennesker opfattes som objekter og dermed bliver ligestillet med en komponent i systemet ( ting, maskine) og ikke som et selvtænkende, fejlbarligt menneskeligt individ. De grundlæggende opfatters og procedurer i SA er på mange måder de samme som under Taylorismen.

Mennesker har fysiske og psykologiske behov såvel som evner, hvilket ingen maskiner har. Behandles ansatte som maskiner stiger utilfredsheden, motivationen falder og de positive menneskelige aspekter miste. Det er gode argumenter for at tænke sig godt om, inden metoden umiddelbart adopteres og tages i brug.

Disse modargumenter for at anvende SA rejser en række spørgsmål omkring mangler i forhold til praktisk anvendelse af metoden. De nævnes nedenfor:

1. SA undervurderer medarbejdernes færdigheder.
2. SA ignorerer betydningen af tilfældige og uformelle samtaler mellem medarbejderne.
3. SA undervurderer hyppigheden og betydningen af fejl og undtagelser fra normen som unægtelig sker gennem arbejdsprocessen. Arbejdsprocesser sker aldrig så smertefrit og glat som beskrevet i DF diagrammer.
4. SA tager ikke problemer som forskellige interesser eller magt i betragtning.
5. SA tilbyder ingen hjælp til beskrivelse eller analyse af en organisation. Metoden indeholder ingen metoder til at modellere organisatoriske problemstillinger. Jobs og arbejdsopgaver er bløde værdier, hvilket jævnfør den funktionalistiske synsvinkel betragtes som en rest af de automatiserede procedurer.

### **11.2.4. Design proces ud fra et SA perspektiv**

Designprocessen ses som en problemløsningsaktivitet, startende med et veldefineret problem og et åbenlyst mål. SA tilhører en bred familie af designmetoder det er baseret på dekomponering af funktio-

---

<sup>85</sup> De tager fat i hvordan mennesker opfattes i en funktionalistisk verden.

ner, hvilket kaldes funktionel analyse. Grundprincippet er at et problem best forstås og løses når det splittes op i mindre dele som kan behandles enkeltvis.

Design handler i den fire fasede model om at transformere den gamle logiske model til den nye, fra en abstraktionsspecifikation til en anden.

Brugerens rolle er i denne proces helt passiv, brugere fungerer blot som informationsressourcer og som reviewere af de forslag vi som designer kommer med. De inddrages ikke som deltagere i det aktuelle design arbejde. Det mener, jeg er en anden holdning end den fremstillede i ISA<sup>86</sup>.

SA's design proces er baseret på tre grundlæggende opfattelser.

1. Der skal være et veldefineret og klart statet problem der skal løses fra starten. Mål og evalueringskriterier for de foreslåede løsninger er konsistente og det skal være muligt at verificere om løsningen lever op til målkriterierne (Kravspecifikationen).
2. Designeren forventes at have en rationel tankegang med fuld information om designvariable samt miljømæssige parametre og disses begrænsninger<sup>87</sup>.
3. Det skal være muligt at skille funktion fra implementation og dermed designe og beskrive systemets funktioner uafhængigt af tanker om implementationen.

Ifølge teksten har disse formodninger meget lidt med design i virkeligheden at gøre. Problemer er sjældent så specifikt defineret, hvilket ligeledes gælder for målene med systemet. Design processen er en fælles seance mellem brugere og designere som bliver forhandlet og mere klart defineret over tid.

Alternativt nævnes en teoretiker ved navn Malhotra<sup>88</sup>. Han har nedbrudt designprocessen i tre dele, som menes at være relevant i denne kontekst. De tre processer kaldes 1) goal elaboration<sup>89</sup>, 2) design generation<sup>90</sup> og 3) design evaluation<sup>91</sup>. Disse tre processer er delt op, men er stadigvæk stærkt relaterede, sammenhængene og ikke mindst forstærker de hinanden.

Fordi SA fokuserer så stærkt på funktionel analyse tilbyder metoden ikke designeren megen hjælp til hvorledes de tre processer skal føres ud i livet. Det medfører at SA overser at bruger participation til en vis grad er uundværlig i dele af designprocessen. Især når problemstillingen, mål og evalueringskriterier ikke er veldefineret fra starten af projektet. En vigtig pointe må her være, at denne participation ikke negligeres i samme grad i ISA. Delskov og Lange<sup>92</sup> ser trods alt participation som ikke uvæsentlig, især ved blitzing og prototyping, selvom det er en funktionalistisk synsvinkel.

Det medfører en række praktiske problemer, som skitseres nedenfor:

1. SA ignorerer konsekvent behovet for ”goal elaboration”, fordi det tages for givet, at problemet er veldefineret.
2. Vigtigheden af ”design generation” er velkendt, men der er ingen forslag til hvorledes det skal gennemføres. Der er dog nævnt prototyping som supportværktøj til generering og evaluering af designet. Prototyping underbygger eksperimenteren og den iterative proces, det nu engang er. To teoretikere nævnes<sup>93</sup> i denne sammenhæng, som kommer med et forslag om at lave fremtids-

---

<sup>86</sup> ISA[Delskov og Lange, 1999]

<sup>87</sup> Dette punkt er i stærkt strid med den løst koblede tankegang [Kristensen og Kreiner, 1999]. Hvor der ikke forudsættes at der er fuld information.

<sup>88</sup> Bansler og Bødker s. 174.

<sup>89</sup> Her diskuteres og states designets mål.

<sup>90</sup> Her forsøges der at fremkomme med et design som møder de stattede mål.

<sup>91</sup> Denne fase starter med introduktion af et nye design som efterfølgende skal evalueres for at forkaste eller fortsætte med forslaget.

<sup>92</sup> [Delskov og Lange, 1999]

<sup>93</sup> Madsen og Kensing.

workshops og metaforisk design til at stimulere genereringen af idéer til at designe computersystemer.

3. SA antager at ”*evaluation*” er et nemt og ligefremt spørgsmål. Designeren spørger ganske enkelt brugeren om meningsfuldheden af den nye logiske model, gennem dataflowdiagrammer. Det er ganske usandsynlig at en bruger er i stand til at forstå hvordan systemet fungerer ved at kigge på formelle beskrivelser i form af DF diagrammer, PD og så videre.
4. Generelt overser SA behovet for og vanskeligheden i at kommunikere med brugere. Brugerne spiller en meget passiv rolle i SA og de forskellige diagrammer er ikke designet eller udformet med brugeren - som den der skal evaluere designet - i tankerne.

Forfatterne tror på at denne formalisme som er præsenteret i SA er uegnet som værktøj og kommunikations instrument alle andre steder for designere og programmører.

### 11.2.5. Undersøgelses studie

Forfatterne har foretaget en undersøgelse i tre forskellige virksomheder på baggrund af nogle interviews. Det er for at se hvorfor og hvorledes metoden anvendes i praksis.

På den baggrund fremdrager de nogle generelle konklusioner om anvendelse af SA.

- SA er kun blevet anvendt fordi virksomhederne enten ikke har været bekendt med andre alternativer eller at de har haft kontakt til en virksomhed der anvender metoden.
- Årsagen til den store spredning af SA, forklarer de med at 1) den lover en kontrol og orden til den kaotiske systemdesignproces og 2) Dataflowdiagrammer opfattes af designerne som et fantastisk kommunikationsmedie og en god dokumentation.
- De synes at det er gode og let anvendelige værktøjer, men de anvendes ofte ikke som de foreskrives i SA.. Ifølge SA skal der udarbejdes tre eller fire modeller<sup>94</sup>, hvilket i langt de fleste tilfælde aldrig sker. Normalt udarbejdes der kun en model, nemlig en model af det nye computer system, hvor der hverken skelnes mellem logiske eller fysiske dataflowdiagrammer.
  - Designere analyserer ikke først det gamle system ved at lave en gammel fysisk model og transformere den til en logisk model. Disse to faser kan blitzing erstatte.
  - Ligeledes bygger designerne ikke først en ny logisk model for derefter at udlede eller transformere den til en ny fysisk model. I stedet ender det med en underlig klon af den logiske og fysiske model.
- Designere opfatter sig selv som tekniske designere og ikke som designere af arbejdsorganisering, jobs og arbejdsopgaver. Det forklarer hvorfor beskrivelser af manuelle procedurer og hvordan arbejde i praksis udføres er meget begrænsede.

Oven på denne undersøgelse er det jævnt forfatterne meget mere korrekt at sige, at dataflowdiagrammer, dataordbøger, mini-spec.. (procesbeskrivelser) i kombination med E/R diagrammer, skærm layouts, prototyper, organisationsdiagrammer og hvad man ellers kan finde på, end at **Struktureret analyse** anvendes til analyse og design af informationssystemer. SA kan derfor i den virkelige verden betyde mange ting.

### 11.2.6. Diskussion

I dette afsnit er der en uddybning af systemudviklingsmetoder som de bruges i praksis i modsætning til, hvordan de er præsenteret i tekstbøger. Det vil blive kigget på årsagerne til at det forholder sig således.

---

<sup>94</sup> Gammel fysisk-, gammel logisk- (essentiell), ny logisk- (essentiell) og ny fysisk model.

Ifølge deres studier viser nogle metodologiske problemer gennem udviklingsfasen sig som ”designer”<sup>95</sup> problemer og i implementeringsfasen viser nogle problemer sig som var det ”bruger”<sup>96</sup> problemer. De vil dog ikke komme nærmere ind på dette her.

### 11.2.7. Brugen af metoder

Deres studier at SA viser, at designere kun bruger SA delvis, hvordan og i hvor høj grad afviger fra organisation til organisation og fra projekt til projekt.

De nævner fire årsager til at det efter deres overbevisning forholder sig som beskrevet.

1. De studerede projekter kan have været for små. Yourdon anbefaler SA til alle projekter mellem 10.000 og 1 mio. linier kode, den dur også til mindre projekter selvom det måske er overkill.
2. En årsag kan have været utilstrækkelig træning i forhold til anvendelsen
3. En tredje årsag kan være organisatoriske politikker.
4. Den begrænsede brug af SA afspejler måske en naturlig metodologisk svaghed.

Når der ses på hvorledes designere fører metoden ud i livet er der identificeret fire væsentlige afvigelser i forhold til hensigten.

1. Designere bruger ikke DF-diagrammer i kommunikationen med brugerne.
2. Diagrammerne omfatter ikke manuelle procedurer.
3. Antallet af modeller er begrænset til én, i stedet for en model for hver fase.
4. Modellen understøttes altid af andre typer beskrivelser, diagrammer eller prototyper.

Det kunne derfor tyde på, at værktøjerne og procedurerne i SA er mangelfulde og besværlige at anvende i virkeligheden.

En forbedret ”*kommunikation med brugerne*” skulle ifølge DeMarco og Yourdon være en af de helt store fordele ved SA. Som nævnt tidligere afspejler diagrammerne et informations proces perspektiv, hvilket er meget fjern fra langt de fleste brugere.

”*Modellering af arbejdsprocesser*” SA er præsenteret som en generel metode til at beskrive og analysere informationsbehandlingssystemer, hvad enten det er manuelt eller automatiseret. Det giver mening at forstå et computersystem ud fra dataprocesser og dataflows, men det er tvivlsomt om det giver mening at forstå arbejde i de selv samme termer.

”*Begrænsning af antal modeller*” At udarbejde en detaljeret og vidtfavnende system model er enormt tidskrævende, hvilket er et dilemma idet designere altid er under et kæmpe tidspres for at levere et produkt. Designere finder det brugbart at lave en ny model, men ikke at udarbejde en gammel fysisk model. De færreste ser bestræbelserne godt givet ud, og at resultatet i stedet bliver en masse uanvendelig information. Det er et kendt problem af teoretikerne og kaldes ”Current physical tarpit”. De siger, at der generelt bruges en uacceptabel mængde ressourcer på at studere det gamle system og at der kommer alt for mange fysiske detaljer med.

Der er stort set enighed om, at det er nødvendigt at kigge på de eksisterende arbejdsprocesser for at forstå behovet for forandringer. De stiller bare spørgsmålet om SA er den rigtige metode til dette. I forhold til dette er de analyserede virksomheder skeptiske overfor en skelnen mellem logiske- og fysiske- modeller. De opfatter det meget nærliggende som en teoretisk og akademisk diskussion, uden nogen form for praktisk relevans.

---

<sup>95</sup> Defineres som problemer med at kommunikere med brugerne [Bansler og Bødker, s. 186].

<sup>96</sup> Hvordan uventede problemer i forhold til systemet skal håndteres.

”Using additional tool” Designerne I virksomhederne nævner to årsager til at anvende og supplere med andre værktøjer, 1) det er med DF-diagrammer, dataordbøger og mini-spec. umuligt at præsentere alle såvel tekniske som organisatoriske aspekter. Blandt andet mangler der specificering af datastrukturer, timing af events, adgangsrettigheder, brugergrænseflader, forretningsprocedurer og organisationsstruktur. Her skal det tilføjes, at der i ISA<sup>97</sup> tages højde for nogle af disse aspekter.

Til at beskrive datastrukturer tegnes der E/R- diagrammer, til brugergrænseflade tegnes der små skærm layouts og der laves små prototyper, organisationsstruktur vises med traditionelle org. Diagrammer og til resten af de ovenfor nævnte aspekter produceres der almindelig dansk tekst. På den måde illustreres et meget rigere billede af organisationen og faldgruberne i SA undgås.

Bansler og Bødker mener, at erfarne designere i praksis - i stedet for at følge SA’s regler og procedurer stringent - vælger mellem de mange forskellige muligheder metoden tilbyder og anvender dem i deres egen design proces. På den måde kompenseres der for metodisk mangelfuldhed og begrænsninger især i forhold til brugergrænseflader og ”human computer interaction”.

Fortolkningen af hvordan det i virkeligheden forholder sig når designere anvender SA, er konsistent med ”The Theory Building view” som er fastslået af en teoretiker ved navn NAUR om programmeringsudvikling. Han forslår at både designprocessen og implementering af programmerede løsninger skal ses som en ”theory building activity”<sup>98</sup>. Problemet med dette er, at alle elementer ikke er dokumenteret, men at de til en vis grad kun eksisterer i hovedet på nogle af de involverede mennesker. Der nævnes fordele som 1) at udviklerne ved hvordan løsningen forholder sig til de problemer systemet skal håndtere i virkeligheden, 2) at der kan forklares hvad hvorfor hver del af programmet er som det er og 3) der kan reageres konstruktivt på ethvert ønske om modifikation ved at have skabt dette mentale billede. I dette perspektiv er det programmørens færdighed og intuition der er omdrejningspunktet for programudviklingen og brugen af bestemte principper, regler og værktøjer spiller en sekundær rolle.

Udfra dette perspektiv er kernen i systemdesign at designeren får én indsigt ”Theory” der omfatter både tekniske som sociale aspekter af den aktuelle situation. Enhver specifikation eller model skal overvejes som underprodukter og kvaliteten af en given metode skal overvejes i forhold til, at opbygge designerens ”Theory”.

---

<sup>97</sup> Integreret struktureret analyse [Delskov og Lange, 1999]

<sup>98</sup> At skabe en ”Theory” svarer til at der dannes et mentalt billede af den programmerede løsning.

---

## **11.3. Floyd**

Pensum: Floyd, A Comparative Evaluation of Systems Development Methods pp. 19-37 (DIS S.146-155)

### **11.3.1. En komparativ evaluering af systemudviklingsmetoder**

## 11.4. Avison & Fitzgerald

Pensum: Avison & Fitzgerald; Methodologies: issues and frameworks pp. 417-457 (DIS S. 156-176)

### 11.4.1. Forskellige metodologier

#### Indledning

Denne artikel tager udgangspunkt i at fagområdet systemudvikling er begravet i metodologier, og at dette kan gøre det svært at afgøre, hvilken en man skal vælge.

Som oplæg til det væsentlige i artiklen gennemgås det hvad en metodologi er, og hvad rationalet er for deres eksistens. I forlængelse heraf gennemgås det hvad baggrunden for metodologier er, og hvad man får ud af at anvende dem.

Dette oplæg fører til det væsentlige, nemlig hvordan man kan sammenligne dem og slutteligt, hvilke emner man bør berøre, hvis man står i en situation, hvor man skal vælge mellem flere forskellige metodologier.

Afsnittet kommer ikke med nogen egentlig løsning på hvordan man vælger metodologi, men den kommer med en række anbefalinger til hvad man kan gøre. Det betyder i praksis at afsnittet vil påpege en række problematikker og behandle nogle emner på et mere filosofisk niveau, hvor det lades op til læseren selv at vælge, hvilken overbevisning man selv har og hvilken indgangsvinkel man ønsker at anvende.

Afsnittet lægger sig tæt op ad diskussionen omkring processer i IT-kvalitet, men kommer dog med en anden og mere nuanceret indgangsvinkel til hvordan man griber det an.

### 11.4.2. Hvad er en metodologi?

Der findes ikke nogen generel definition af begrebet, men der er dog enighed om at det er en skridt og procedurer man følger under udarbejdelsen af et informationssystem.

En måde at betragte det på er en definition givet af British Computer Society, der går på at det er en anbefalet samling af filosofier, faser, procedurer, regler, teknikker, værktøjer, dokumentation, ledelse og uddannelse for udviklere af informationssystemer.

Denne definition har ført til en operationel beskrivelse af hvad en metodologi i hvert fald skal indeholde:

- Hvordan projektet skal brydes ned i faser
- Hvilke opgaver der skal udføres i hver fase
- Hvilket output der skal laves
- Hvornår og under hvilke omstændigheder opgaverne skal løses
- Hvilke begrænsninger der skal inddrages
- Hvilke folk der skal involveres og hvordan de skal uddannes
- Hvordan projektet skal styres og kontrolleres
- Hvilke understøttende værktøjer der skal inddrages

Ud over disse meget konkrete emner, bør en metodologi behandle den bagved liggende filosofi, og de antagelser som forfatteren til metodologien tror på.

Det er i særdeleshed denne filosofi, som teksten understreger som væsentlig, og som det der gør en metodologi til mere end en "bageopskrift". Det er også dette punkt der adskiller en metode fra en me-

metodologi. Denne indgangsvinkel kan forklares med at en metodologi er en samling metodiske principper, der ved hver enkelt anvendelse skal konkretiseres til en metode tilpasset den specifikke situation.

Teksten giver en indgangsvinkel, hvor man betragter udviklingen af informationssystemer som en undersøgelse, der kan bygges op over et generelt skelet med tre forskellige aspekter:

- *Intellektuelle overvejelser*<sup>99</sup>. Her er der tale om tage stilling til nogle fundamentale filosofiske spørgsmål, der kan virke som retningslinje for undersøgelsen. Der er primært tale om to filosofiske dimensioner: den ontologiske dimension og den epistemologiske dimension. Ontologien er læren om det værende. Det vil sige nogle antagelser omkring den fysiske og sociale verden der omgiver os. Epistemologien er læren om viden. Det vil sige, hvordan man lærer, og hvad viden er.
- *Metodologi*. Dette er operationaliseringen af de intellektuelle overvejelser. Hvordan gennemfører man en undersøgelse hvor de filosofiske holdninger bliver gennemført.
- *Problemområde*. Her er der tale om det man gennemfører undersøgelsen på.

Denne indgangsvinkel lapper i høj grad over de indholdsmæssige retningslinjer der er givet ovenfor, men sætter dog stadig systemudviklingen ind i et lidt bredere undersøgelsesperspektiv.

### 11.4.3. Rationalet for anvendelse af metodologier

Dette afsnit handler om hvorfor man i det hele taget skal anvende en metodologi. Afsnittet handler om de samme ting som er behandlet i IT-kvalitet.

Teksten nævner tre fordele ved anvendelse af metodologier:

- *Man får et bedre produkt*. Her er der tale om et ønske om bedre kvalitet på grund af anvendelsen af en speciel metodologi. Det fører til to problemer: hvad er kvalitet og kan man bevise at det metodologien der har ført til netop den opnåede kvalitet. Til det sidste problem er svaret at det kan ikke bevises. Til spørgsmålet om kvalitet i informationssystemer lister teksten en række forskellige parametre man kan bedømme ud fra:
  - *Accept*. Indfrier systemet de ønsker der er til det.
  - *Tilgængelighed*. Om systemet er tilgængeligt; hvor og hvad kræves der.
  - *Sammenhæng*. Er der interaktion mellem forskellige komponenter. Passer systemet ind i forretningens strategi m.m.
  - *Kompatibilitet*. Er systemet kompatibelt med andre systemer og afdelinger.
  - *Dokumentation*. Er systemet veldokumenteret.
  - *Indlæring*. Er systemet let at lære – er det intuitivt forståeligt.
  - *Økonomi*. Er systemet cost-effective, og overholder det budgetterne.
  - *Effektivitet*. Om systemet på bedst mulig måde imødekommer de overordnede forretningsmål
  - *Efficiens*. Om systemet udnytter ressourcerne bedst muligt.
  - *Hurtig udvikling*.
  - *Fleksibilitet*. Om systemet er nemt at tilpasse nye krav.
  - *Funktionalitet*. Indfrier systemet kravene i specifikationen.
  - *Implementation*. Er overgangen fra gammelt til nyt system nem at gennemføre.
  - *Lav grad af kobling*. Kan komponenter udskiftes.
  - *Vedligeholdelse*. Er systemet nemt at vedligeholde.
  - *Mobilitet*. Kan systemet køre på forskellige platforme.
  - *Pålidelighed*. Mængden af fejl er lav og output er konsistent og korrekt.

<sup>99</sup> Dette er gennemgået mere i afsnittet nedenfor om hvilke emner man bør berøre i sit valg af metodologi.

- *Robusthed.* Systemet kan tåle at der opstår fejl.
- *Sikkerhed.*
- *Simpelt.* Ingen unødigt kompleksitet.
- *Testbarhed.* Kan systemet testes
- *Tid.* Giver systemet det krævede output til tiden, under såvel normale forhold som under spidsbelastninger.
- *Gennemsigtighed.* Fremgår det hvorfor output er som det er.

Mange af disse parametre er i direkte modstrid, hvilket umuliggør opfyldelse af dem alle. Det væsentlige er at finde en metodologi der kan fokuseres mod de kvalitetsparametre man mener er væsentlige i forbindelse med det system, man er ved at udvikle. Ifølge teksten er dette et problem, fordi de fleste metodologier ikke har taget eksplicit stilling til dette.

- *Man får en bedre udviklingsproces.* Dette knytter sig til fordelene ved at kunne kontrollere sin udviklingsproces. Dette emne hænger sammen med diskussionen af ISO, CMM m.m.
- *Man får standardiseret sin proces.* Dette punkt knytter sig til de fordele der opstår, når hele virksomheden følger den samme arbejdsmetode, og anvender de samme begreber. Dette muliggør at flytte medarbejdere rundt i virksomheden, og man får en langt højere grad af integrationsmuligheder, når produktet er lavet efter samme skabelon.

#### **11.4.4. Baggrunden for metodologier**

Dette afsnit gennemgår kort den historiske udvikling inden for metodologier.

Metodologier kan groft set deles ind i to kategorier:

- Udsprunget af praksis
- Udarbejdet på baggrund af teori

De praktisk funderede metodologier, er de mest udbredte. De er typisk opstået evolutionært, ved at en udvikler har fundet en god måde at gøre tingene på. Dette udspringer oftest i en enkelt teknik, der hen ad vejen bliver udviklet til at omfatte mere og mere. Som udgangspunkt har metodologien været brugt som et værktøj til at levere virksomhedens produkt, men hen ad vejen er metodologien blevet det produkt man sælger – denne udvikling medførte flere investeringer i udviklingen og det førte til en række fordele:

- *Mangler er blevet udbedret.*
- *Rammerne for metodologien blev udvidet.* Hvor fokus før var på nogle bestemte faser er dette blevet udviklet til at omfatte hele udviklingsprocessen.
- *Fokus på konkurrencemæssige fordele.* Man blev opmærksom på at informationssystemer skal understøtte en virksomhed og øge de konkurrencemæssige fordele.

Den anden kategori af metodologier udspringer af de højere læreranstalter, hvor man med tiden har bemærket at de eksisterende metodologier ikke var funderet i konkret teori. Denne udvikling har naturligt nok påvirket de kommercielle metodologier, men de teoretiske har haft svært ved at vinde indpas i den praktiske verden, fordi de har været svært håndgribelige.

#### **11.4.5. At indarbejde en metodologi**

Overvejelserne omkring anvendelse af en speciel metode går primært ud på hvad man får og om man er garanteret succesfulde informationssystemer i gengældelse.

Spørgsmålet om hvad man får ved at indarbejde en metodologi er vidt forskelligt afhængigt af leverandør. Spredningen er illustreret med disse punkter:

- Metodologier varierer fra at være en samling udviklingsprincipper til at være en færdig komplet bageopskrift
- Metodologier kan dække forskellige faser af udviklingsprocessen – nogle er strategiske andre operationelle.
- Nogle metodologier dækker kun for analysen – andre kun design af den fysiske model. Nogle dækker alle faser
- Nogle metodologier er lavet til en speciel type problemer – eksempelvis udarbejdelsen af administrative systemer. Andre dækker alle former for problemer.
- Nogle metodologier er lavet til specialister – andre til brug af alle. Andre igen er lavet så udvikleren selv definerer sin egen metodologi til det konkrete projekt.
- Nogle metodologier kræver mange mennesker til at løse en række forskellige opgaver – andre metodologier definerer ingen egentlige opgaver.
- Til nogle specielle metodologier er der udviklet et CASE-værktøj.

I forlængelse af alle disse variationer kommer prisforskellen. Nogle metoder leveres i tilknytning til CASE-værktøjer andre leveres gratis. Det væsentlige i denne forbindelse er imidlertid ikke hvor meget man betaler for metodologien, men prisen for at implementere ændringerne i organisationen. Disse omkostninger vil ofte være langt større, hvorfor det anbefales ikke at overlade beslutninger omkring metodologier til en IT-afdeling.

Til det andet spørgsmål om hvorvidt man er garanteret succesfulde IT-systemer er svaret klart nej. I mange tilfælde vil den samme metodologi anvendt af to forskellige udviklere ikke give det samme resultat – og dermed heller ikke den samme kvalitet. Jo mere ensartede resultater man får ud af metodologien jo mere detaljeret skal den være, og desto mere væsentligt er det at den definerer den bedste måde at løse problemet på, da man ellers står med en dårlig metodologi med ringe grad af selvbestemmelse. Diskussionen går altså på graden af frihed i udviklingsprocessen, og muligheden for at vælge den bedste måde at gøre tingene på – denne diskussion er gennemgået mere i dybden i IT-kvalitet under implementering af software processer.

#### **11.4.6. At sammenligne metodologier**

Der er som udgangspunkt to grunde til at sammenligne metodologier: en akademisk og en praktisk. Den akademiske knytter sig i høj grad til at forstå baggrunden og filosofien bag metodologierne og den praktiske knytter sig til at finde en egnet metodologi til et bestemt område.

Dette afsnit behandler det at sammenligne metodologier på en meget operationel måde. Det vil sige at der kommer en række forskellige emner der bør være indeholdt i en metodologi eller som man kan bruge som grundlag til at evaluere flere forskellige metodologier mod hinanden. De forskellige måder at gøre dette på vil lappe over hinanden, da de er udarbejdet af forskellige forfattere og i nogen grad udarbejdet som tillæg til andres arbejde.

Den første metode til sammenligning er lavet som en oversigt over hvad der bør være indeholdt i en metodologi. Sammenligningen må her gøres ved at finde mangler i de metodologier man arbejder med og vurdere hvorvidt det er væsentlige mangler eller ej. Listen er udarbejdet af en person kaldt Catchpole – listen kan springes over, hvis man ikke står med et konkret problem at løse..:

- *Regler.* En metodologi bør indeholde regler for hvorledes udviklingsprocessen faseopdeles og for hvordan værktøjer m.m. bruges.

- *Omfang.* En metodologi bør omfatte hele processen fra valg af udviklingsstrategi til implementering og vedligeholdelse.
- *Forståelse for informationsressourcerne.* Metodologien bør sikre god udnyttelse af de rådige ressourcer i form af data og processer til at behandle disse.
- *Dokumentationsstandarder.* Metodologien bør definere output for de forskellige faser der er forståelige af såvel brugere som analytikere.
- *Adskillelse af logisk og fysisk design.*
- *Validering af design.* Der skal være mekanismer i metodologien, der sikrer konsistens, korrekthed og fuldkommenhed af design i forhold til krav.
- *Tidlige ændringer.* Det skal være muligt at identificere ændringer til systemet så tidligt som muligt, da omkostningerne ellers bliver for store – hvordan dette gøres nævnes ikke...
- *Der skal være kommunikation mellem de forskellige udviklingsfaser.* Alternativt kan arbejdet i en fase ikke bruges i de næste faser.
- *Effektiv problem analyse.* Der skal være metoder der sikrer at man får dokumenteret problemer og målsætninger i organisationen.
- *Planlægning og kontrol.* Metodologien bør være udarbejdet på en sådan måde, at man kan planlægge og kontrollere udførelsen. Alternativt bliver omkostningerne for store. Planlægningsværktøjer kan være en del af metodologien.
- *Effektivitets måling.* Metodologien bør indeholde værktøjer der muliggør måling af kvaliteten af de produkter der bliver udviklet.
- *Øget produktivitet.* Metodologien bør føre til øget produktivitet.
- *Forbedret kvalitet.* Metodologien bør føre til bedre kvalitet.
- *Gennemsigthed.* Metodologien bør fungere således at man kan se informationssystemets udformning, og følge udviklingen.
- *Lærenem.* Metodologien skal være nem at lære, således at brugere og analytikere værdsætter de redskaber metodologien stiller til rådighed.
- *Grænser.* Metodologien bør muliggøre definerings af grænser for hvad det enkelte informationssystem indeholder og hvad der ikke er inkluderet. Disse grænser bør kunne defineres således at man ikke kun analyserer ting, der kan underbygges af IT:
- *Ændringer.* Det bør være en overskuelig opgave at ændre den fysiske og logiske model.
- *Kommunikation.* Metodologien bør understøtte en effektiv kommunikation mellem brugere og analytikere.
- *Simple.* Metodologien bør simpel at anvende. Hvis den er for svær at bruge bruger man den forkert og så får man forkerte informationssystemer.
- *Fortløbende relevans.* En metodologi bør kunne udvides med nye former for værktøjer og teknikker uden at miste sin konsistens.
- *Automatiserede udviklingsværktøjer.* Metodologien bør understøttes af CASE-værktøjer, da de øger produktiviteten.
- *Brugere.* Metodologien bør tage højde for brugernes ønsker og mål.
- *Participation.* Metodologien bør indfri målene om participation<sup>100</sup>.
- *Relevans for praktikere.* Metodologien bør ikke være så teoretisk at praktikere med erfaring inden for IT og sociologi ikke finder den anvendelig.
- *Relevans for problemområdet.* Metodologien skal understøtte den form for system der skal udvikles. (Administration, overvågning, styring...)

---

<sup>100</sup> Se note ovenfor om participation.

I forlængelse af denne liste er der kommet forslag til udvidelser:

- *Fremtiden.* Metodologien bør understøtte fremtidige ændringer og krav til systemet.
- *Integrering af teknisk og ikke-teknisk.* Det er ikke nok kun at arbejde med tekniske og ikke-tekniske problemer. Metodologien bør understøtte integrationen af disse forskellige problemstillinger
- *Åben for nye muligheder.* Metodologien bør ikke kun arbejde med eksisterende problemer, men bør åbne muligheden for at tage nye emner op.
- *Adskillelse af analyse og design.* Analysen af et system, bør ikke påvirkes af overvejelser omkring fysisk design.

En anden måde at gribe problemet an på er ved at behandle disse otte emner. Emnerne knytter sig i høj grad til værdier og det samfund systemet skal være en del af:

- Hvilket undersøgelses paradigme eller perspektiv danner grundlag for metodologien.
- Hvilke værdier findes implicit i metodologien
- Hvad er den kontekst hvor metodologien fungerer bedst
- I hvilken grad er ændringer tilskyndet eller mulige.
- Fungerer kommunikation og dokumentation på brugernes sprog, eller er det ekspert-sprog.
- Kan ting overføres.
- Behandler metodologien de sociale omgivelser og de problemer der knytter sig til dem.
- Opfordres der til bruger deltagelse, og understøttes det.

Listen er mere overordnet, hvilket måske gør den mere anvendelig, men man skal være opmærksom på at den bygger på en række antagelser – for eksempel at brugerdeltagelse er en god ting!

En tredje måde at gøre det på, ved at anvende denne metode. Metoden er baseret på systemtankegangen og evaluerer andre metoder op mod dette ved at stille spørgsmål inden for tre forskellige kategorier:

- *Problemsituationen* (den kontekst metodologien skal bruges i)  
Denne kategori af spørgsmål skal opklare hvorvidt metodologien hjælper til forståelse af problemområdet. Eksempelvis områder er:
  - Klienterne og deres forståelse, erfaring og tilhørsforhold
  - Problemejerne og deres bekymring og problemer.
  - Den situation brugerne af metodologien skal arbejde med og hvorvidt den opfattes som struktureret eller ustruktureret
  - Den måde hvorpå metodologien kan hjælpe på problemsituationen
  - Den kultur løsningen skal indgå i og de risici der er forbundet med brugen af metodologien i det pågældende sociale miljø
  - Interessenternes opfattelse af virkeligheden og hvorvidt dette harmonerer med metodologiens grundlæggende filosofi.
- *Problemløseren* (Brugeren af metodologien)  
Her stiller man spørgsmål om:
  - Hvad brugeren af metodologien tror på omkring værdier og normer og etik og hvordan dette stemmer overens med metodologien. Alternativt om metodologien kan ændres så det kommer til at stemme overens
  - Hvilken filosofi brugeren af metodologien har, og om dette stemmer overens med metodologien.

- Brugerens erfaringer og evner, og hvordan dette stemmer overens med de krav der stilles i metodologien.
- *Problemløsningsprocessen* (selve metodologien)  
Denne kategori stiller spørgsmål ved selve metodologien og hvordan den understøtter en række forskellige emner – eksempelvis:
  - Forståelsen for problemsituationen og det at sætte grænser herfor.
  - Gennemførelsen af diagnosen (identificering af problemer).
  - Definerings af problemer.
  - Definerings af teoretiske/mulige systemer. Hvordan gøres det – hvis det gøres overhovedet. Hvordan gemmes de.
  - Definerings af logiske og fysiske modeller. Hvem involveres i hvad, og hvordan fører det til forbedringer eller udvikling af problemområdet.
  - Implementering – hvordan sikres succes.

Med henblik på at evaluere sin metodologi anvendelse anbefales det at man stiller disse spørgsmål før, under og efter man har brugt metodologien. Derved kan man danne et indtryk af hvorvidt anvendelsen er gået godt eller skidt.

En fjerde måde at vælge metodologi på er ved at klassificere sin problemsituation i følgende fem kategorier og vælge metodologi derefter:

1. En veldefineret problemsituation, med klare krav. Her kan man anvende traditionelle systemudviklingsmodeller.
2. Som ovenfor, men med uklare krav. Her kan man vælge en metodologi med proces, eller data-modellering. En anden mulighed er prototyping.
3. Ustruktureret problemsituation med uklare målsætninger. Her kan man anvende en SSM indgangsvinkel.
4. Systemer med høj grad af brugerinteraktivitet. Her kan man med fordel anvende en indgangsvinkel som ETHICS.
5. Meget uklar problemsituation. Her kan man anvende en multiview metodologi der bruger flere forskellige indgangsvinkler.

Som afslutning har forfatterne til teksten udarbejdet deres egen 16 punkts model til sammenligning af metodologier:

- Hvilke dele af udviklingsprocessen dækker metodologien
- Hvilken overordnet model anvendes. Er der tale om en spiral model, en vandfaldsmodel, en livscyklusmodel ...
- Hvilken beskrivelsesform bruger den og hvilke modeller er anvendt
- Hvilke værktøjer og teknikker bruges
- Er metodologien veldokumenteret, så den kan anvendes og læres. Dette gælder ikke kun de enkelte udviklingsfaser, men også filosofien bag metodologien.
- Hvad fokuserer metodologien på. Er det eksempelvis data, mennesker, processer...
- Hvordan defineres resultaterne fra de enkelte faser
- Hvilke situationer og hvilke systemtyper er metodologien velegnet til
- Er metodologien videnskabelig, baseret på systemtankegangen eller...
- Forudsættes det at der skal laves en computer baseret løsning.
- Hvem spiller hvilke roller i udviklingen. Kræver det eksperter at anvende metodologien, og er det muligt at involvere brugerne
- Hvilke evner skal brugerne af metodologien have

- Hvordan håndteres forskellige opfattelser af problemer
- Hvilke kontrol mekanismer understøttes og hvordan estimeres succes
- Hvilke fordele tilskriver metodologien sig selv og hvordan dokumenteres det
- Hvad er de bagvedliggende filosofiske antagelser – hvad er det der gør denne indgangsvinkel til en legitim indgangsvinkel.

Forfatteren fremhæver det sidste punkt som det mest væsentlige.

#### **11.4.7. Komparativ analyse af metodologier**

Hvor det foregående afsnit behandlede sammenligningsmetoder, der var meget operationelle, behandler dette afsnit en mere overordnet indgangsvinkel, der måske er bedre i opgave sammenhæng, hvor det forrige var bedre i praktiske opgaver.

Modellen til denne komparative analyse er bygget op over 7 punkter der i nogle tilfælde brydes yderligere ned. Inden præsentationen af modellen skal det nævnes at det væsentligste i forbindelse med en komparativ analyse af metodologier er at gøre det fuldstændig klart hvad formålet med analysen er – dette er vigtigere end hvorvidt man analyserer et bestemt punkt eller ej.

Modellen er bygget op over mange af de emner der er behandlet ovenfor, men den prøver at definere en række begreber der kan bruges i de forskellige dele af analysen.

De syv punkter:

1. *Filosofi*. Filosofien er som beskrevet ovenfor de underliggende antagelser i metodologien som påvirker måden den er bygget op på. I denne model er filosofien beskrevet med fire forskellige punkter:

- a. *Paradigme*. Som udgangspunkt kan man vælge at arbejde med to forskellige paradigmer: et videnskabeligt paradigme og et system paradigme.

Det videnskabelige paradigme er det dominerende inden for hard system thinking, hvor en naturvidenskabelig syn på verden er dominerende. En del af denne indgangsvinkel er at man kan bryde alt ned og analysere delene hver for sig.

System paradigmet er karakteriseret ved en holistisk indgangsvinkel – altså at alt ses i en kontekst, som man ikke kan se bort fra i en analyse. Det vil sige at helheden er mere end de enkelte dele lagt sammen.

En anden måde at betragte paradigmer på er ud fra to forskellige dimensioner: epistemologi og ontologi.

Ontologien er læren om det værende – altså hvordan man ser på verden og de ting der er i den. Denne dimension har to poler: realismen og nominalismen.

Realismen siger at alt eksisterer i en bestemt objektiv form uafhængigt af hvem der observerer dem. Her kan man altså udlede en korrekt beskrivelse af ting.

Nominalismen er i den modsatte grøft og siger at alt er som man opfatter det – det vil altså sige at der ikke er en endelig definition på noget.

Den epistemologiske dimension handler om viden. Det handler om hvordan vi definerer viden og hvordan man mener man kan gennemføre analyser. Igen er der to poler inden for denne dimension: positivisme og tolkning.

Positivismen siger at der er en årsagssammenhæng mellem alt og denne kan udledes med analyse.

Tolkning vil sige at der ikke nødvendigvis er nogen endegyldig sandhed der kan udledes.

Disse to dimensioner kan kombineres i to overordnede paradigmer, der er henholdsvis subjektive og objektive. I det subjektive paradigme kombinerer man opfattelsen af at verden eksisterer i den enkeltes opfattelse med en videnskabelig indgangsvinkel baseret på tolkning.

I det objektive paradigme er der en endelig definition på verden og dens årsagssammenhæng kan udledes med naturvidenskabelig analyse.

Der er nok sjældent nogen der kan sige at de befinder sig i nogen af de ekstreme synspunkter, men paradigmerne kan alligevel bruges til at analysere hvorvidt man er af den samme opfattelse af ting eller ej.

Paradigme overvejelserne kan anvendes i langt større perspektiv end de gør i denne tekst – i IT-forandringsledelse anvendes de til at skabe grundlag for organisationsforandringer, hvilket i høj grad har relevans i denne sammenhæng.

- b. *Formål.* Dette punkt drejer sig om hvad metodologien sigter mod at opnå. Den primære skelen er mellem de metodologier der kun sigter mod computerbaserede systemer og de der sigter mod løsning af problemer i en større sammenhæng. Holdningen til dette punkt er at sigte mod en computerbaseret løsning er for snævert i en organisations-sammenhæng, der ofte kræver andre løsninger.
  - c. *Problemområde.* Det primære i denne del er omfanget af problemområdet. Der er metodologier der kun behandler bestemte typer af problemer, og dermed overlader andre typer til andre metodologier. Andre metodologier er åbne for alle typer af problemer (Eksempelvis SSM). Holdningen er her at man er nød til at gennemføre en top-down analyse af hele problemområdet og sikre sig at problemer løses i overensstemmelse med organisationens overordnede formål og strategi.
  - d. *Mål.* Her er der tale om hvilken type problemer metodologien sigter mod at løse. Punktet er også behandlet ovenfor i de mere operationelle indgangsvinkler og drejer sig om at nogle metodologier er bedst egnede til administrationssystemer, andre til overvågning.
2. *Model.* Dette punkt drejer sig om selve den model metodologien baseres på. En sådan model kan bruges til flere forskellige formål. Først og fremmest kan den bruges til at kommunikere mellem aktører i problemområdet og analytikerne. Dernæst kan den bruges til at sikre konsistens i arbejdet med et problem, så man kan gå fra problem til løsning og implementation. Sidst kan den tjene som værktøj til at få indsigt i problemområdet.

Modeller kan kategoriseres i fire forskellige typer:

- a. Verbal.
- b. Analytisk eller matematisk
- c. Billede, skema eller diagram
- d. Simulation

Modeller til informationssystemer vil oftest være skematiske eller diagrammer. Dette skyldes den vægt der lægges på kommunikationen mellem analytikere og brugere.

Man kan vælge at betragte en model som en abstraktion over problemet. Dette har den fordel at man fjerner alt fysisk og konkret fra problemet og kommer ind til kernen.

Man kan vælge at dele modeller op i forskellige niveauer. Teksten beskriver tre generiske niveauer:

- a. *Konceptuel model.* Denne model skitserer det højeste niveau. Den beskriver ikke nogen form for database eller proces aktivitet. En væsentlig risiko er at man ikke kan se forskel på konceptuel model og logisk model. Hvis der er risiko for dette, er det væsentligt at analytikeren gør sig umage med at forklare forskellen over for brugeren.
  - b. *Logisk model.* Denne model beskriver et informationssystem uden skelen til teknisk implementering. Formålet med denne model er at lave en kravspecifikation.
  - c. *Fysisk model.* Denne model svarer til den logiske model inklusive den tekniske løsning.
3. *Teknikker og værktøjer.* Dette beskrives som en væsentlig del af metodologien, men det vil ikke blive yderligere behandlet her.
  4. *Omfang.* Her er der tale om at analysere hvilke dele af et systems livscyklus metodologien beskæftiger sig med.
  5. *Output.* Dette punkt drejer sig om hvad metodologien har af output i de enkelte faser og i særdeleshed i den sidste fase. Her skal man skelne mellem om metodologien leverer en kravspecifikation, et fuldt implementeret system, eller noget helt tredje.
  6. *Praktik.*
    - a. *Baggrund.* Dette punkt knytter sig til metodologiens oprindelse som den er beskrevet ovenfor – altså om den oprindeligt er udarbejdet som en akademisk model, eller som en praktisk model.
    - b. *Brugere.* Her diskuteres primært participation problematikken.
    - c. *Players.* ???
  7. *Produkt.* Hvad får man som kunde ud af metodologien. Eksempelvis kravspecifikation, software, dokumentation...

## 11.5. Bansler

Pensum: Jørgen Bansler – Systems development in Scandinavia, three theoretical schools pp. 117-133 (DIS S. 177-193)

### 11.5.1. Indledning til Scandinaviske udviklingsmodeller

Denne tekst beskriver tre forskellige teoretiske indgangsvinkler til systemudvikling:

- Systemteoretisk indgangsvinkel
- Socio-teknisk indgangsvinkel
- Den kritiske skole

Dette afsnit vil først gennemgå de generelle træk ved de forskellige indgangsvinkler, hvorefter deres historiske baggrund bliver gennemgået hver for sig. Den sidste del vil i denne note blive skåret ned til et minimum.

### 11.5.2. Baggrund for udviklingsmodellerne

Udgangspunktet for en ny udviklingstradition er ifølge teksten bygget op over fire forskellige parametre:

- *Videnskabelig interesse.* De underliggende mål og intentioner som motiverer udviklerne – de ting de gerne vil opnå med deres systemudvikling. Teksten påpeger at dette er det væsentligste punkt når man skal finde forskelle på udviklingsstrategier.
- *Metafysiske antagelser.* Dette punkt dækker over en række antagelser omkring verden og dens oprindelse, som ikke kan eftervises med logik eller empiri – svarer nogenlunde til den ontologiske dimension beskrevet ovenfor.
- *Grundlæggende antagelser.* Dette punkt lægger sig tæt op ad de metafysiske antagelser. Det dækker over en række grundlæggende antagelser inden for det forsker felt der udvikler den pågældende udviklingsmodel. Tilsammen med de metafysiske antagelser, danner dette punkt ”de briller verden anskues med”. Svarer nogenlunde til paradigme overvejelser.
- *Praktiske resultater.* Dette punkt dækker over de modeller og systemer der bliver udviklet inden for den pågældende udviklingstradition. Emnet vil ikke blive behandlet yderligere i denne tekst.

#### Videnskabelig interesse

I dette punkt er de tre metoder vidt forskellige. Den systemteoretiske indgangsvinkel er fokuseret på brugen af IT som en optimeringsmekanisme. Det vil sige, at man er interesseret i at finde løsninger, der optimerer arbejdsprocesser med hensyn til arbejde, tid eller materialer.

Socio-tekniske udviklere arbejder med socio-psykologiske problemer forårsaget af systemudvikleres negligering af den menneskelige faktor i systemudvikling. De tror på at man ved at fokusere mere på bruger krav i systemudviklingen kan øge jobtilfredsstillelsen og produktiviteten samtidig.

Udviklere inden for den kritiske tradition arbejder med demokrati på arbejdspladsen. Det er deres hensigt at anvende IT til at styrke arbejderne og fagforeningernes magtbaser i relation til ledelsen.

#### Grundlæggende antagelser

- *Syn på organisationer.* Den system teoretiske indgangsvinkel er betragter organisationen som noget der skal styres. Det vil sige at der kommer fokus på kontrolmekanismer m.m.

Den socio-tekniske tradition betragter organisationen som en integrering af to forskellige systemer: et socio-system og et teknisk system. Vægten lægges på analysen af menneskelige relationer.

Den kritiske tradition ser organisationen som en social ramme om samarbejde og konflikt, mellem grupper – specielt ledelse og arbejdere. Fokus er på interesser og magt. Denne indgangsvinkel lægger sig tæt op ad den politiske ændringsstrategis organisationssyn, som er beskrevet i IT-forandringsledelses kapitlet.

- *Syn på medarbejdere.* Den systemteoretiske tradition betragter mennesket som en enhed i systemet – baseret på taylorismens medarbejdersyn. De to andre mener ikke medarbejderen er en ”del af maskinen”, men derimod at organisationen består af netværk og sociale relationer.
- *Arbejdet i organisationen.* Den systemteoretiske- og den socio-tekniske tradition mener at arbejdere og ledere har de samme interesser og arbejder mod de samme overordnede mål. Konflikttraditionen mener organisationen er præget af konflikt.

	System teori	Socio-teknik	Kritisk tradition
Videnskabelig interesse	Profit maksimering	Job tilfredsstillelse Deltagelse	Industrielt demokrati
Syn på organisation	Styret system	Socio-teknisk system	Ramme for konflikter
Syn på arbejderne	Enheder i maskinen	Individer	Individer
Syn på harmoni/konflikt	Harmoni	Harmoni	Konflikt

### 11.5.3. Historie

I dette afsnit gennemgås historien for de enkelte teorier kort.

#### Systemteoretisk tradition

Denne tradition udspringer i 60'erne, hvor der er stor tro på naturvidenskab og teknik.

Metoderne har traditionelt arbejdet med rationalisering, efficiens og kontrol. Til gengæld har man nedprioriteret tilfredsstillelse, helbred blandt arbejdere og sikkerhed. Dette har medført den samme kritik som taylorismen, hvor mennesket bliver nedprioriteret.

Enkelte gange har man set mennesket betragtet som noget andet end en enhed i maskinen, men dette har været ud fra en betragtning om at mennesket handler irrationelt og modsætter sig udviklingen, hvorfor medarbejdernes indflydelse skal nedprioriteres og styres.

#### Den socio-tekniske tradition

I 60'erne begyndte man i høj grad at implementere styringssystemer i virksomhederne, der skulle øge ledelsens indflydelse på arbejdet. Dette medførte store ændringer af organisationerne og mødte ofte stor modstand blandt medarbejderne. Nogle systemer fungerede ikke på grund af denne modstand.

Resultatet blev at man fokuserede mere på socio-psykologiske faktorer og Mumford's socio-tekniske system design blev et forbillede. Man valgte at gøre job tilfredsstillelse til den væsentligste succes parameter.

Den socio-tekniske tradition bryder med opfattelsen af mennesket som en enhed i maskinen og betragter Taylorismen som en fejl. Dette kan ifølge teksten få den ulempe at man bremser vejen for ny tekno-

logi, der kommer alle til gode, blot fordi man mener det er et produkt af ledelsens ønske om mere styring.

### **Den kritiske tradition**

Den kritiske tradition udspringer i 70'erne som en reaktion på de systemer, der blev introduceret i 60'erne. Fagforeningerne havde indledningsvis været tilhængere af nye systemer, men måtte efterhånden tage afstand fra den måde de blev brugt på.

Derfor startede de selv nogle projekter med systemer, der skulle øge demokratiet på arbejdspladsen – dette blev det primære mål, at opnå fuldstændigt demokrati på arbejdspladsen. Denne tankegang udspringer af en opfattelse om at der ikke er harmoni mellem ledelsen og medarbejdernes ønsker – de har forskellige mål og man må vælge side. Den kritiske tradition vælger side med arbejderne.

Det er tanken at det er fagforeningerne der skal hjælpe arbejderne med at få udviklet systemer, der understøtter dette demokrati. Problemet er bare at nutidens fagforeninger ikke gør dette, men i stedet fokuserer på løn, fritid og sikkerhed på arbejdspladsen. Tradition betragtes derfor som mere idealistisk end praktisk gennemførlig.

## **11.6. Kautz**

Pensum: Karlheinz Kautz – User participation and participative design pp. 267-284 (DIS S. 194-203)

### **11.6.1. Indledning til Systemudviklingsuddannelse**

Denne tekst er skrevet som et indlæg i en debat om sammensætning af systemudviklingsuddannelsen. Indledningsvis gennemgår teksten den matematik baserede uddannelse i system udvikling, hvorefter den ingeniør baserede indgangsvinkel gennemgås. Kritikken af disse to udviklingsformer er at de er for teknik baserede, hvorfor der introduceres et eksempel på hvorledes man kan undervise i brugerdeltagelse i systemudvikling.

Afslutningsvis gennemgås en uddannelse, der sigter på brugen af prototyper

Artiklen drejer sig om det samme som alle de andre paradigme artikler – det er ikke nok med den tekniske indgangsvinkel. I relation til vores undervisning er indholdet af begrænset værdi! (Med mindre selvfølgelig der er tænkt på noget ganske specielt, som vi ikke har fået gennemgået ☺)

Udgangspunktet for diskussionen er at der næsten udelukkende undervises i algoritmer, kodesprog, datastrukturer m.m. I denne undervisning er brugerdeltagelse ikke inddraget.

### **11.6.2. Matematikbaseret uddannelse**

Udgangspunktet i denne form for undervisning er, at programmer er komplicerede formler, der bearbejder tal ud fra matematisk logik. Målet er altså at lære at omdanne specifikationer til korrekte eksekverbare formler. En del af dette inkluderer uddannelse i grundlæggende matematik.

En indgangsvinkel der går ud på udelukkende at bygge systemudvikling op over matematisk logik er blevet kritiseret, idet der dog tages forbehold for at dette er en vigtig del af et uddannelsesforløb. En del af kritikken går ud på at det kan være svært at omdanne specifikationer til matematik, hvis man laver systemer, der er dårligt specificeret.

### **11.6.3. Ingeniørbaseret uddannelse**

Denne indgangsvinkel går ud på at uddanne folk der systematisk arbejder med planlægning, analyse, dokumentation og validering i deres systemudvikling.

Forbilledet er udarbejdelsen af almindelige maskiner, der ofte kan udarbejdes med meget faste kravspecifikationer. Man erkender her den matematiske logik som en del af de redskaber udvikleren skal besidde. Det påstås at der ikke er brug for kommunikation mellem udvikler og bruger – når først specifikationen er på plads, kan udviklerne udarbejde systemet.

Kritikken går naturligvis på, at systemer indgår i en kontekst og man derfor ikke kan se isoleret på systemudvikling, uden eksempelvis at arbejde med organisationsforandring.

### **11.6.4. Traditionel systemudviklingsuddannelse**

Hvis man ser på den eksisterende uddannelse i systemer, kan den deles i to dele: informationssystem udvikling og software udvikling. Den første del dækker over forretningsmål og indarbejdelse i organisationer, samt en smule bruger involvering – dette undervises primært på handelsskoler. Software udvikling er den ingeniørbaserede del, der primært undervises på universiteter.

Hvor den ene skole beynder og den anden slutter kan være svært at sige, men teksten påpeger at det væsentlige er at alle har fokus på at systemet skal indgå som en aktør i en organisation, når det er færdigt.

Det væsentligste problem har været en arbejdsdeling, hvor ledelsen udstikker retningslinjer for systemet og udvikleren har en rolle som ekspert der udarbejder dem – den eneste brugerinvolvering der har været har været lidt grænseflade design.

### **11.6.5. Brugerdeltagelse og participative design**

Udgangspunktet er her at man skal acceptere at systemudvikling dækker over alle de aktiviteter der involveret i at ændre organisationen ved hjælp af IT. Det vil sige at udviklerne skal prøve at forstå alle de organisatoriske, sociale og politiske aspekter af arbejdspladsen og tage ansvar for de ændringer de laver. En del af dette går ud på at involvere brugerne og lade dem forklare hvilke problemer og behov de har, frem for at lade ledelsen komme med en række faste krav.

Teksten kommer med tre grunde til at dette er en fordel:

- Øget viden som fundament for systemudviklingen.
- Reducering af modstanden mod systemet.
- Øget demokrati på arbejdspladsen.

Påstanden er, at denne form for brugerinvolvering gør participative design som udviklingsmetode overlegen i forhold til andre.

Som eksempel gennemgås uddannelses strukturen i Norge, hvor man har delt uddannelsen i tre moduler:

- *Computere og samfundet.* Dette kursus giver en gennemgang af konsekvenserne af systemimplementering. Dækker over modstand og forskellige interessegrupper.
- *Systembeskrivelse og sprog.* Her gennemgås problemer med at specificere systemer og der undervises i kodesprog.
- *Systemudvikling – teorier og modeller.* Baggrund for systemudvikling og overordnede modeller for systemudvikling gennemgås. Fokus på det organisatoriske perspektiv. En del af dette kursus har været prototyping – gennemgået nedenfor.

### **Prototyping – en evolutionær indgangsvinkel til systemudvikling**

Kurset er baseret på brugerinvolvering gennem aktivt brug af prototyper. Prototyperne er bygget op som beskrevet i (Budde, 1992 – vores tekst)(Floyd, 1984 – må være før vores udgaver...).

Kernen i kurset har været rollespil, hvor eleverne skiftes til at være bruger og udvikler. Konklusionen var at rollespillet var frugtbart og man stødte på en række konflikter, som ellers ikke var blevet gennemgået.

## **11.7. Floyd et. Al.**

Pensum: Floyd et. Al., Out Of Scandinavia. Alternative approaches to Software Design and System Development pp. 165-193 (DIS S.130-145)

### **11.7.1. Alternative tilgange til systemudvikling og software design**

Artiklen er meget beskrivende og historisk. De pointer som frembringes vil herunder blive beskrevet samt et superkort resume af artiklen.

Artiklen hedder Out of Scandinavia og lægger op til en vurdering om, hvorvidt Software Design og System Development er anderledes i Skandinavien end andre steder. Hvorledes er de sociokulturelle forhold i Skandinavien og hvorledes influerer de på Software Design og System Development? Nogle af de konklusioner som forfatterne fremdrager er at der lader til at være et tæt samarbejde mellem universiteter og fagforbund i Skandinavien, hvilket er med til at fremme den sociale udvikling, samt sikre teknologisk og teknisk udviklingsfremgang. Forfatterne mener ikke at de sociokulturelle forhold kan kopieres til andre steder i verden, men mener dog alligevel at den teoretiske og metodologiske tilgang, samt opfordring til brugerdeltagelse og humaniseringen af teknologien kan bruges andre steder end Skandinavien. Artiklen er ret lang og skriver meget om sociale forhold. Læs evt. hele konklusionen for et mere specifikt billede af artiklens indhold.

Det udtales i artiklen at flere forskellige skoler i Skandinavien forsøger at undervise i de samme værdier, som systemudviklere har. Dette sker for at lette overgangen fra skolebænken til den virkelige verden. Humanisering og demokrati er altoverskyggende mål, som tillægges større værdi end andre steder, når det drejer sig om design af computerbaserede systemer.

### **11.7.2. The spirit of the Scandinavian Approach**

Forskellen mellem systemudvikling i Skandinavien og andre steder kommer bedst til udtryk som følger:

- Systemet skal enten reflektere ejerne af systemets interesser eller – så godt som muligt – alle der bliver berørt af systemet. Dette viser en høj grad af demokratisering.
- Systemets primære funktion er at kompensere for menneskelige fejl og mangler eller at støtte menneskelige styrker. Dette viser en høj grad af humanisering.

Mange steder i Skandinavien lader man til at hælde mest til sidstnævnte punkt, altså humanisering. Det udtales, at hvor udviklingen af computer science i USA er kraftigt farvet af militærets indflydelse, er den Skandinaviske tilgang meget mere fokuseret på mennesket, jf. ovenstående punkter om demokratisering og humanisering. Ligeledes beskrives det, hvorledes man i Skandinavien lægger stor vægt på individet i arbejdsprocessen. Faktorer som tilfredshed og ”well-being” vurderes som værende præcis så vigtige som effekten af teknologiske, organisatoriske og strukturelle ændringer af de sociale grupper, set i relation til produktivitet og stabilitet i det sociotekniske system.

Således bør brugerne inddrages under systemudviklingen og man kan på en måde sige at udviklerne og brugerne sammen danner et team af forskellige eksperter. De kan hver især yde deres bedste for at sikre det bedst mulige produkt. Men man skal sikre et ligeligt mix, da en overdrejning imod enten udviklerstyret eller brugerstyret systemudvikling vil resultere i et produkt, der er ”farvet” af den dominerende ekspertskare og måske ikke er fuldt anvendeligt til slut.

Afslutningsvis vil jeg påpege at artiklens brugbarhed i forbindelse med vores pensum og fag ikke skinner tydeligt igennem. Den har ikke vist sig synderlig brugbar i forbindelse med vores projektarbejde og er heller ikke blevet gennemgået til nogen af forelæsningerne.

En lille hjælp kan være at læse konklusionen, som nævnt tidligere, da den følger godt op på indholdet af artiklen.

# Definition of Information Systems

## Basic Terms and Different Perspectives of Information Systems Development

## Organization

- Lecture/Repetition
- Pensum
- Group work and group presentations
  - as part of class and outside class
    - pensum papers
    - case studies
    - project-related exercises
- Project work

## Definition of Information Systems

*Why are concepts important?*

What is information (as opposed to data)?

**Data:**

a representation of facts or ideas in a formalized manner capable of being communicated or manipulated by some process.

**Information:**

the meaning that a human being assigns to data by means of known conventions applied to that data.

## What is a system ?

- A classical definition of system

a collection of objects, called parts, which are correlated in some way (Langefors, 1973)

- A “political” definition of system

a part of the world which we choose to regard as a whole, separated from the rest of the world during some period of consideration, a whole which we choose to consider as containing a collection of components, each characterized by a selected set of associated data items and patterns, and by actions which may involve itself and other components (Holbæk-Hansen, Håndlykken & Nygaard 1975)

## What is an information system?

- Definition of a general information system

a system to collect, process, store, transmit, and display information (Avison & Wood-Harper, 1990)

- Definition of a formal information system based on information technology

an integrated man/machine system for providing information to support the operations, management and decision making functions in an organization. The system uses computer hardware, software, manual procedures, management and decision models and a data base. (Davis, 1974)

## What is an information system?

- Definition of a formal information system which does not necessarily include information technology

An information system is any system which assembles, stores, processes and delivers information relevant to an organization (or to society), in such a way that information is accessible and useful to those who wish to use it, including managers, staff, clients and citizens. An information system is a human activity (social) system which may or may not involve the use of computer systems. (Buckingham et al., 1987)

## Information Systems Development: Methodologies, Techniques and Tools

What is a methodology?

... a collection of procedures, techniques, tools and documentation aids which will help some actors to perform a task in a systematic way ...

**In our case:**

to help system developers to implement (an) information system(s).

***Beware!***

.... methodology vs. method/method vs. technique ....  
(UK. English vs. US English/Scandinavian)

## Information Systems Development: Methodologies, Techniques and Tools

Why should we use a methodology ?

- to achieve a standardized, repeatable process representing best practice
  - a better development process
  - a better end product

## Information System Development

What is the difference between software engineering and system development?

System development consists of all activities which aim at changing an organization through use of information technology.

It is not just a process of technical change, but means organizational development as well.

In system development emphasis is on using the computer for information processing.

## ISD and Methodologies: *A question of perspective !*

System development can be seen as:

- construction process
- organizational change process
- learning process
- political process

This is expressed in different perspectives:

a traditional perspective  
(hard) systems perspective

non-traditional perspectives  
social-technical perspective  
critical perspective

## A traditional perspective on ISD

organizations are seen as unitary structures with a manifest and rational reality

the organizational reality consists of objects, properties, and processes which are observable and 'just' have to be discovered

requirement specifications are clear-cut documents based on an objective reality

## A traditional perspective on ISD

system developers are neutral experts in analysis and design methodologies and project management

management provides the project objectives, system developers turn them into products

the resolution of contentious objectives is seen as a task for management and not in the domain of the developers

system developers use methodologies that allow them only to analyze and describe the technical features of organizations

the organizational, social and political context is abstracted away to pure information processing aspects

## The traditional perspective - Some Drawbacks

- limited to catch less formalized activities, structures and more dynamic organizations
- outdated requirement specifications which do not capture the 'real' working practices

Others.... ?

## Beyond the traditional perspective ...

organizations are seen as pluralistic structures where people with different interests work

requirements are not objective, but depend on the stakeholders' perceptions

requirements are fuzzy and hidden in work practice and they may change in the course of time

## Beyond the traditional perspective ...

system developers are not neutral experts, but take a stand and act as facilitators and/or change agents

the center of interest is moved to everyday work practice of the people in organizations

development is seen as a process to understand, define and solve user problems and needs, with active involvement of users

methods which support this process are used

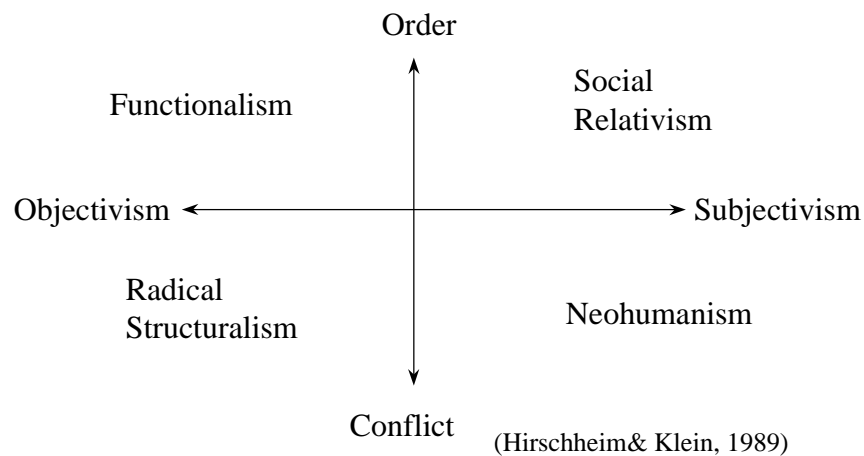
## Group work:

Explain Hirschheim&Klein's model!

How can it help us in our practical work as users of methods and in our project work?

What is its relationship to what you have learned in the first 2 years?

## Yet another way of classifying system development perspectives



Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

17

## Four schools of system development (according to Hirschheim/Klein)

### A summary

Paradigm	Developer archetype	Systems development proceeds	Elements used in defining IS	Examples
Functionalism	(Neutral) Expert	from outside, application of formal concepts, planned intervention with rationalistic methods	people, hard-, software, rules as objective entities	Structured analysis
Social Relativism	Facilitator	inside, improvement of subjective understanding by adapting to internal forces	subjectivity of meaning, sharing of meaning	ethnographic approaches (Florence)
Radical Structuralism	Warrior (for social progress)	from outside, raising of ideological consciousness, organized political action, adaptation of methods to class interests	people, hard-, software, rules as objective entities in the service of class interests	trade-union approaches (Utopia)
Neohumanism	Emancipator	inside, improvement of understanding and rationality, emancipation and liberation from social constraints	people, hard-, software, rules as objective entities for TKI, subjectivity of meaning for other KI	critical social theory

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

18

# Definition of Information Systems:

Combining Perspectives: A Contingency Approach

## Scandinavian Approaches

-

### A look from the outside

Floyd et al. (1989) say about the Scandinavian tradition:

*“We have come to identify such an approach in certain common features shared by different schools of thought. These includes efforts toward humanization and democratization as overriding design goals, in keeping with the aim of building an egalitarian society.”*

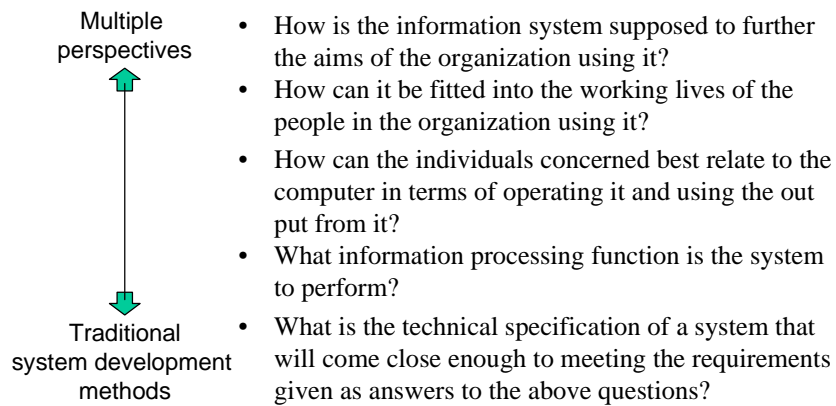
## Combining Multiple Perspectives in one Methodology

-

### The Multiview Example

- *MV* addresses problems associated with the *analysis* and *design* activities of *information systems definition*
- *MV* is a methodology to *structure the tasks* for the *analysts* and *users* during these activities
- *MV* is a contingency -- situation-oriented -- methodology
- *MV* is a flexible framework for the application of different other methodologies and techniques
- *MV* includes understanding of different stakeholders
- *MV* includes different perspectives of one situation (social, political, technical, ....)

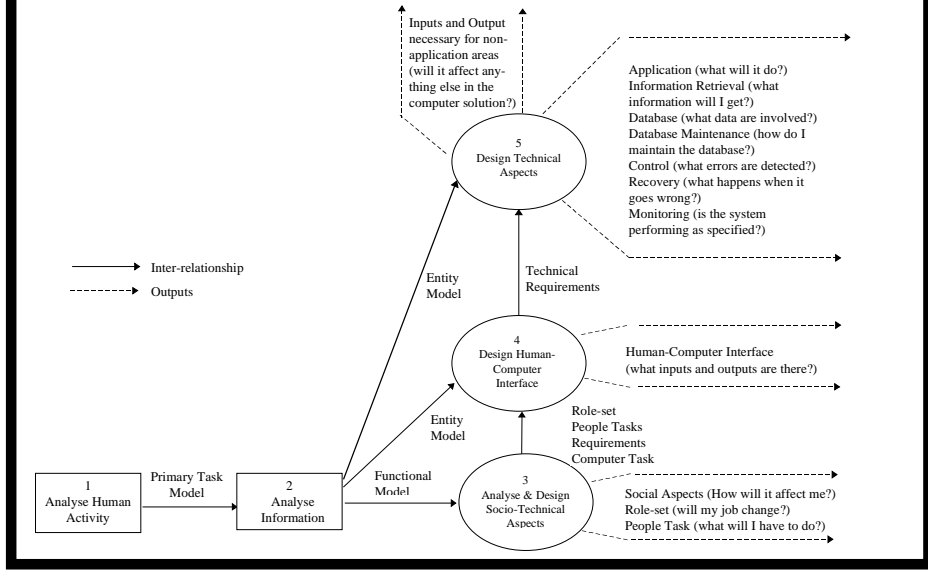
## Multiple Perspectives & Traditional Methodologies



## Elements of a Multiperspective Approach

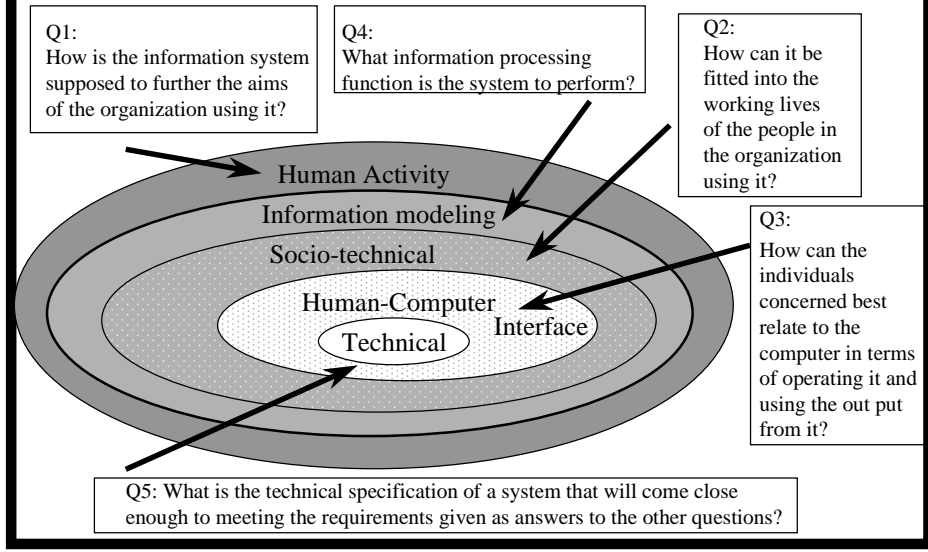
1. Analysis of the human activity system
2. Analysis of information/information modeling
3. Analysis and design of socio-technical aspects
4. Design of the human-computer interface
- [5. Design of technical aspects]

# The Multiview Framework



Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

# Multiple Perspectives - Focus of Multiview



Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

## Analysis of Human Activity System

- What is a human activity system: ‘any sort of organization’
  - an individual, a company, a department, a club, ...
- Weltanschauungen
  - different perceptions of the same situation as basis for the description of system requirements
- substages within the analysis of human activity systems:
  - perceiving the problem situation/ producing Rich Pictures
  - constructing systems models/ Root definitions and Conceptual Models
  - comparing the systems models to perceived reality
  - deciding on the comparison and then implementing the consequences of these decisions



**Soft Systems Methodology**

## Analysis of Information - Functions and Entities

- development of a functional model
  - main function can be derived from root definition of chosen model (as a start)
  - functional decomposition
  - data flow diagrams
- development of an entity model
  - entities can be derived from rich pictures (as a starting point)
  - entity-relationship-diagrams
- process descriptions
- data descriptions
- state-transitions



**Structured analysis - integrated analysis**

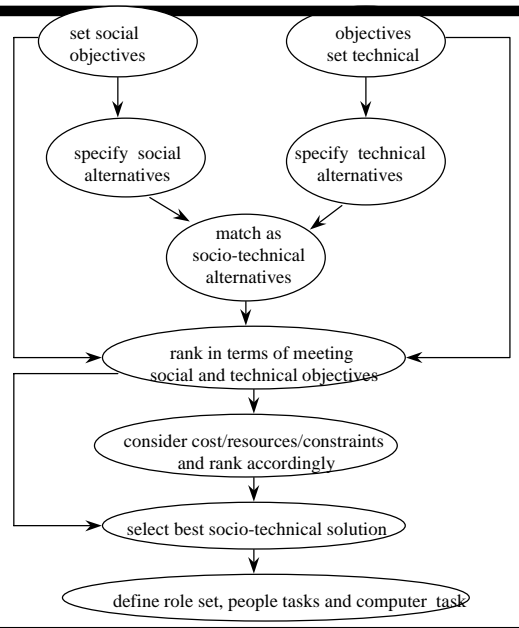
## Analysis and Design of socio-technical Aspects

- Fit between social and technical objectives and possible solutions
- Identification of alternative solutions and selection of one alternative

## Socio-Technical Analysis and Design: ETHICS - An Example

Rationale for Change/Definition of System Boundaries		
Description of existing System	→	<b>SSM</b>
Definition of Key Objectives and Tasks		
Diagnosis of Efficiency Needs	→	<b>SA/IA</b>
Diagnosis of Job Satisfaction Needs		
Future Analysis/ Specifying and weighting efficiency and job satisfaction needs and objectives	→	<b>ETHICS</b>
Organizational design of new system		
Technical Options	→	<b>SD/SP</b>
Detailed (work) design/Implementation		
Evaluation	→	<b>ETHICS</b>

## Socio-Technical Analysis in Multiview



## Design of Human-Computer Interface

- Definition of interaction
  - screen layout, window layout
  - command input
  - etc.. ....



more details in the specific module on the topic

## Technical Design



Not a subject of this course

## Group Discussion

How can Multiview be characterized in the framework of Hirschheim and Klein?

How do the elements of the multiple perspective fit together?

Is the sequence of elements as proposed by Multiview appropriate?

### **Group Work:**



Prepare the bank case study!

## Definition of Information Systems

-

### Participation and Project Work



### User Involvement and Document Review

## Participation and User Involvement

Participation is a process in which 2 or more parties inform and influence each other in making plans, politics, specifications, or decisions. It is restricted to decisions that have future effects on those making the decisions or on those being represented by them.

### Management's motivation

- participation is morally right
- helps to build interested and committed design groups
- involves those who the job best
- persuades to accept changes

### Employees' motivation

- provides protection
- leads to more interesting jobs
- improves working life quality

### Others' motivation

- develops responsible people

## Participation - Basic Concepts

- Each stakeholder group should have the feeling that all its members benefit
- The aim is to acquire knowledge to make informed decisions
- Participation in systems development involves a process over time and not merely the giving of information or an opinion at one moment in time



more than just one interview

## Forms of Participation

- consultative
  - users are not members of the design group, but contribute with ideas, decisions are made by design group or management
- representative
  - selected 'user' representatives and developers build the design group, contribute and prepare or make decisions, taken either by the group or management
- consensus
  - design group prepares alternatives, presents them,

but all those affected take decision

## Techniques for User Involvement

### Developer-User Communication

#### Areas of Knowledge:

- concrete experience with users' present work (1)
- relevant structures of present work (2)
- overview over technical options (3)
- concrete experience with technical options (4)
- concrete experience with design proposals (5)
- visions and design proposals (6)

#### Techniques

#### Areas of Knowledge

	1	2	3	4	5
observation	1				
interviews	1				
surveys	1				
developers doing users work	1				
video recording	1				
mock-ups	1				
think aloud experiments	1				
rich pictures	1	2			
conceptual modeling	1	2			
culture analysis	1	2			
sa/oo-analysis		2			5
sa/oo-design					5
event lists		2			5
e-r diagrams		2			5
wall graphs		2			5
mapping		2			5
future workshops		2			5
metaphorical design		2			5
data flow diagrams		2			5
language analysis		2			5
card games		2			5
prototyping			3		5
visits to other installations			3	4	
literature study				4	
study standard software			3	4	
forum theater					

## Issues of Participation

- Trust and Motivation
  - Motivating developers to work with users and vice versa
- Election vs. Selection
  - Identifying appropriate users and obtaining access
- Conflicts of Interest
- Stress
- Communication Skills
- Role of System Analysts/Designers
  - Facilitator and Expert
- Role of Management

## Project Work - Document Reviews

- Subject
  - to evaluate a finished product
    - analysis report, requirement specification, design document, code document
    - project plans
  - not the process of production or style of the producer
  - to point to problems, not to solve them
- Objectives
  - to point out need for improvement
  - to approve a product
  - to create homogeneous quality
  - to train

## Document Reviews

### Participants (between 3 and 7)

- review leader (from outside the project)
  - distributes review guidelines, checklists, and material
  - schedules meeting (no longer than 2 hours)
  - controls agenda
- reviewers (from outside the project)
  - are knowledgeable and prepared (no longer than 2h.)
  - give positive and negative comments
- reporter (from outside or inside the project)
  - creates basis for report
  - takes notes on paper and blackboard
- producers
  - give an introduction, but are otherwise silent, and
  - answer only questions for clarification

### Results

- report
  - list of issues
- review decision
  - acceptance without changes
  - acceptance with minor changes, no further review needed
  - rejection with major changes, new review needed

## Group Work

### I. User Involvement - ALL:

- develop a rough plan for user involvement in your project
- whom do you want to involve?
- which approach and technique do you want to apply?

### II. Reviews - 1 group

- develop review guidelines

### III. Reviews - 8 groups

- 2 groups to be reviewed (soft systems analysis)
- 2 groups to be reviewed (structured analysis)
- 4 groups reviewing the documents

## Definition of Information Systems

The Systems Approach -  
Rich Pictures and Relevant Systems



## The Systems Approach:

What has it to do with information systems development?

“We need a new system for the administration of students”

- organizational and management problems are seldom clear-cut and well defined
- organizations are unpredictable as human behaviour is unpredictable
- organizations are complex and not closed and self contained

## The Systems Approach:

What has it to do with information systems development?

- the systems approach (systems thinking) provides an opportunity to understand organizations in a more holistic way
- human activity system is a concept to think about an organization in which an information system may be used in a more general way

## The Systems Approach

### Human Activity System

- a system has a purpose - it exists for a reason and achieves some change, or 'transformation'
- its performance can be measured, and it can be shown to be more or less efficient
- there is a mechanism for control - a decision making process
- it has components - which can themselves be taken to be systems

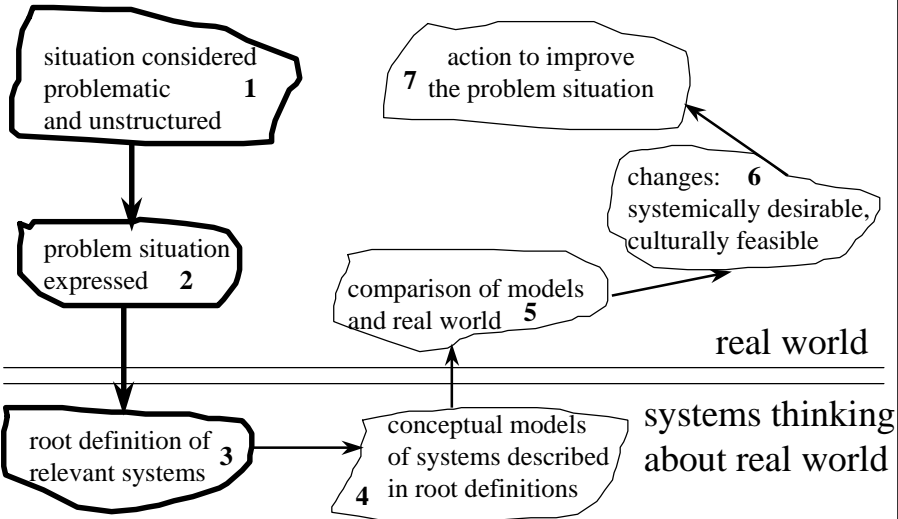
### Example: a university

- a university educates, promotes learning - it 'transforms' undergraduates into graduates
- student/staff ratios, pass rates, research papers published
- management structure
- faculties, departments, institutes

## The Systems Approach

- its components are related and interact
- it exists as a part of a wider system or systems - its environment, with which it must interact
- it has a boundary - which defines what is and what is not part of the system
- it has its own resources
- it has an expectation of continuity, and it can be expected to adapt to or recover from disturbances
- board meetings shared courses
- education system, community
- the dean is part of the university, the mayor not; the H-building is, the town hall is not
- staff, finance, buildings
- the university expects to exist in 10 years time, though it may have to accommodate government spending cuts

## The Systems Approach: An Overview



## Understanding the Problem Situation

### Problem Definition - The first Step: a Rich Picture

an unstructured problem situation

#### The purpose of a rich picture

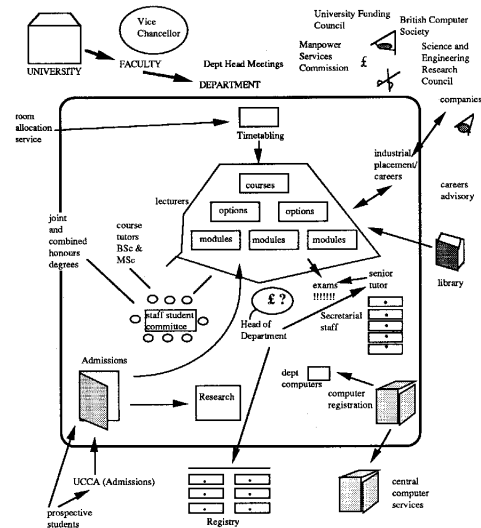
- support communication between all involved
- support an agreement on a starting point between analyst (developer), client (mgt), and problem owner ('user')
- show important persons, activities, tasks, issues and problems
- reveal worries and potential conflicts

## Rich pictures

- get hold off as much background material as possible: interviews, published documents, observations
- depict structures, process, and relationship between structure and process ('climate')
- depict 'hard' and 'soft' facts
- no 'fixed' syntax of symbols:
  - cross swords - conflicts, 'thinks' bubbles - worries, arrows - relationships, etc.. ...

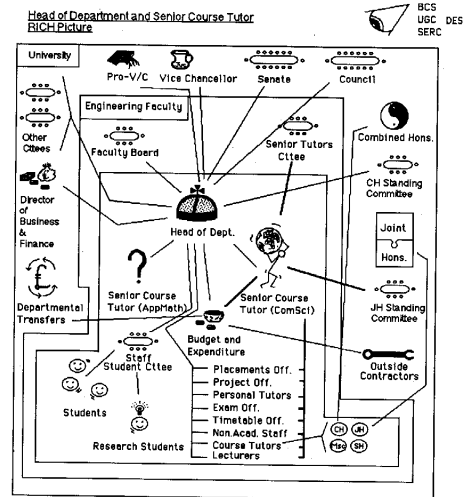
## Rich Picture: An example

A university  
department



## Rich Picture: An Example

### - Concerns of Head of department and Senior Tutor



Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

55

## A Rich Picture Exercise

### Convert the description to a rich picture!

At the center of this problem situation is a large recruitment agency with a poor reputation amongst its customers, both people seeking work and companies using the agency's services. A recent survey conducted by a newly appointed management team has discovered the following:

- 'the sign-on fee is too high - what do you do for my money'
- 'non one ever gets back to me'
- 'non one knows who I am'
- 'you send my CV out without my permission'
- 'you do not listen to what I want'
- 'you send me to unsuitable interviews'
- 'you do not check on the real requirement of the job'
- 'once I am in a contract you forget about me'
- 'many of the applicants are quit unsuitable'

The agency is also facing increasing competition from a number of other agencies operating in the area and the management team feel that some sort of differentiation might be needed in the future.

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

56

## Human Activity System - Towards Root Definitions: Defining Relevant Systems

- A rich picture represents a holistic view of a system(s)
- extract those parts which we want to examine further:  
extract relevant systems - decide which systems to examine further
- relevant systems because they are relevant to the problem situation
- relevant systems may represent different perspectives on (perceptions of) the 'same' (problem) situation
- they may cover primary task(s) of or issues within the organization under investigation

## Group Work

- produce a Rich Picture of the recruitment agency
- extract relevant systems



## Definition of Information Systems

### The Systems Approach - Relevant Systems and Root Definitions



## Human Activity System - Towards Root Definitions: Defining Relevant Systems

- A rich picture represents a holistic view of a system(s)
- extract those parts which we want to examine further:  
extract relevant systems - decide which systems to examine further
- relevant systems because they are relevant to the problem situation
- relevant systems may represent different perspectives on (perceptions of) the 'same' (problem) situation
- they may cover primary task(s) of or issues within the organization under investigation

## Root Definitions

- a root definition is a concise, verbal description of the purpose of a relevant system
- it is not the real world that is being modeled, but potential systems
- root definitions often have the following form:  
A system to do **X**, by (means of) **Y** in order to (do) **Z**
- This tells us  
*who* is doing *what* for *whom* and *to whom* are they answerable,  
*what* assumptions are being made and in what *environment* this is happening  
(Hvem gør hvad for hvem og overfor hvem er de ansvarlige, hvilke antagelser er der gjort, og i hvilke omgivelser sker dette?)

## Root Definitions: Examples

### A primary task-based root definition:

A university owned and operated system to award degrees and diplomas to suitably qualified candidates (**X**), by means of a suitable assessment (**Y**), in conformance with national standards, in order to demonstrate the capabilities of the candidates to potential employers (**Z**).

### An issue-based root definition:

A university owned and operated system to implement a quality service (**X**), by devising and operating procedures to delight its customers and control its suppliers (**Y**), in order to improve its educational products (**Z**).

## Root Definitions:

### CATWOE - The Completeness Test

- Customer - **whom** (victims or beneficiaries of T)
- Actor - those **who** do T
- Transformation - **what** (input -> output (T))
- Weltanschauung - assumptions (the worldview that makes the T meaningful in context)
- Owners - answerable (those with the power to stop T)
- Environment - environmental constraints  
(elements outside the system which are taken as given, but nevertheless affect its behaviour)

## CATWOE - An Example

A university owned and operated system to award degrees and diplomas to suitably qualified candidates (X), by means of a suitable assessment (Y), in conformance with national standards, in order to demonstrate the capabilities of the candidates to potential employers (Z).

- Customer - candidate students
- Actor - university staff
- Transformation - candidates -> degree holders
- Weltanschauung - the belief that awarding diplomas is a good way of demonstrating the qualities of candidates to potential employers
- Owners - the university governing body
- Environmental constraints - national educational and assessment standards

## Differing Weltanschauungen result in different root definitions - an example

	PATIENT	DOCTOR	ADMIN
Customer	me	patients	doctors
Actor	doctor	me	me
Transform ..	treatment	treatment	enable doctors to optimise waiting lists
Weltan ...	as tax payer entitled to treatment	treat as many people as possible	create a bigger hospital within budget
Owner	the taxpayer (the 'system')	hospital administration	Dept. of Health
Environ ..	the hospital	National Health Service/private life	politics

## Differing Weltanschauungen may result in different Root Definitions - An Example

- A hospital is a place that I go to in order to get treated by a doctor. I am entitled to this because I am a tax payer, and the system is there to make sure that taxpayers get the treatment they need.
- A hospital is a system designed to enable me to treat as many patients as possible with the aid of specialised equipment, nursing care etc.. Organizational decisions are made by the hospital administrators against a background of the NHS politics and my visions of weekends off with my family.
- A hospital is an institution in which doctors are enabled by administrators to provide service which balances the need to avoid waiting lists with that to avoid excessive spending. Ultimate responsibility rests with the DoH and the environment is very political.

## Group Work

- use your Rich Pictures from the recruitment office case
- develop root definitions for the extracted relevant systems



# Definition of Information Systems

The Systems Approach

-

Root Definitions and Conceptual  
Models



## Root Definitions

- a root definition is a concise, verbal description of the purpose of a relevant system
- it is not the real world that is being modeled, but potential systems
- root definitions often have the following form:  
A system to do **X**, by (means of) **Y** in order to (do) **Z**
- This tells us  
*who* is doing *what* for *whom* and *to whom* are they answerable, *what* assumptions are being made and in *what environment* this is happening  
(Hvem gør hvad for hvem og overfor hvem er de ansvarlige, hvilke antagelser er der gjort, og i hvilke omgivelser sker dette?)

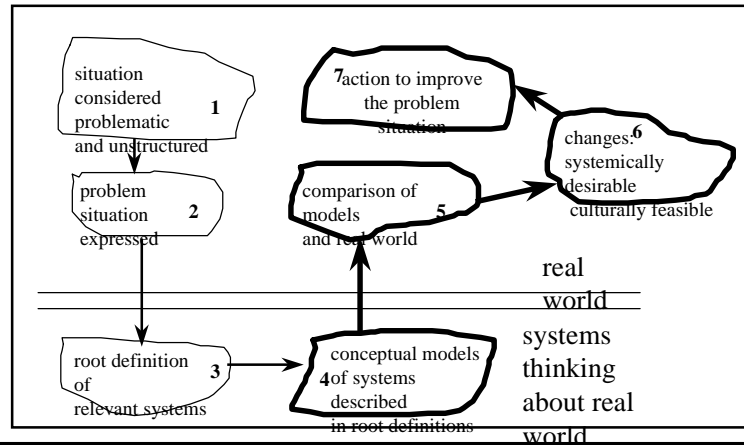
## Root Definitions:

### CATWOE - The Completeness Test

- Customer - **whom** (victims or beneficiaries of T)
- Actor - those **who** do T
- Transformation - **what** (input -> output (T))
- Weltanschauung - assumptions (the worldview that makes the T meaningful in context)
- Owners - answerable (those with the power to stop T)
- Environment - environmental constraints

## Definition of Information Systems

### • Conceptual Models and Reality



Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

73

## A conceptual model

is formed as follows

- identifying significant aspects from the root definition
- assemble a small number of verbs which describe the most fundamental activities in the defined systems
- decide what the system has to do & how it would accomplish the requirement (and how it would be monitored and controlled)
- group similar or related activities (name subsystems with verbs)
- connect activities which are logically related with arrows

~~and verify the model against the 'real world'~~

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

74

## More about conceptual models

The elements of a conceptual model are activities and can be found by extracting all verbs from a root definition.

The number of activities should be seven plus/minus two.

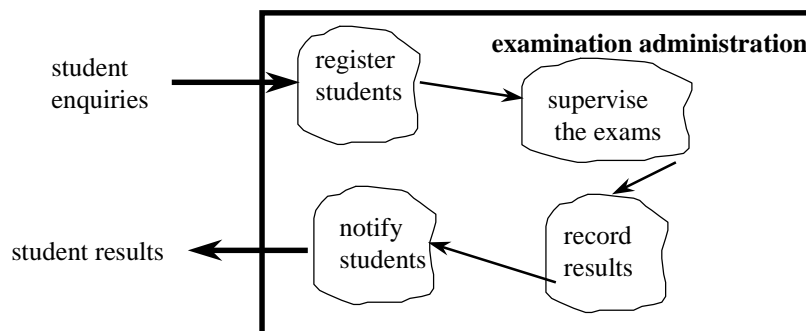
Again: the conceptual model illustrates what ought to be happening to achieve the objectives, not what actually happens

The conceptual model is used as a technique to think of and to discuss (which and) how the subsystems should be organised

## A conceptual model - simple example

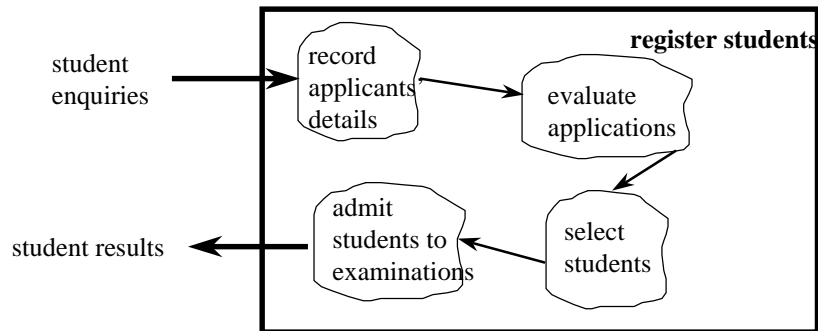
### Root definition:

A system owned and operated by the university to administer the examinations by registering, supervising, recording and notifying students accordingly.

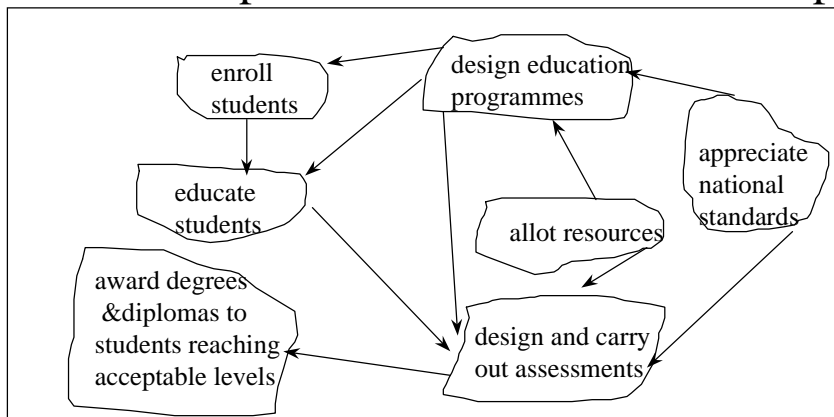


## More on conceptual models: level 2 models

To ensure that the most useful subsystems have been identified they are described in more detail - diagrammatically and verbally (level 2 conceptual models)



## Conceptual Models - Another Example



A university owned and operated system to award degrees and diplomas to suitably qualified candidates (X), by means of a suitable assessment (Y), in conformance with national standards, in order to demonstrate the capabilities of the candidates to potential employers(Z).

## Conceptual Models : Check against formal model

### A formal system model

- a system has a purpose - it exists for a reason and achieves some change, or 'transformation'
- **its performance can be measured, and it can be shown to be more or less efficient**
- **there is a mechanism for control - a decision making process**
- it has components - which can themselves be taken to be systems
- its components are related and interact
- it exists as a part of a wider system or systems - its environment, with which it must interact
- it has a boundary - which defines what is and what is not part of the system
- it has its own resources
- it has an expectation of continuity, and it can be expected to adapt to or recover from disturbances

## Conceptual Models - Monitor and Control: Measures of Performance

E<sup>1</sup> - efficacy - does the system work ?  
- is the transformation achieved?

E<sup>2</sup> - efficiency- is the system worthwhile?  
- are the resources used in an appropriate way?  
(a comparison of the value of the output of the system and the resources to achieve that output)

E<sup>3</sup> - effectiveness - does the system achieve its long term goals?

## Measures of Performance: An Example

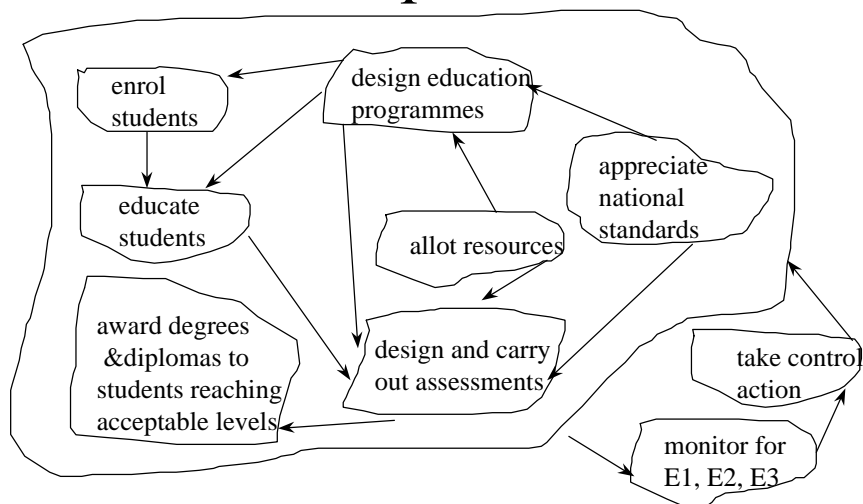
E<sup>1</sup> - efficacy - are degrees and diplomas awarded ?

E<sup>2</sup> - efficiency- how many degrees and diplomas, of what standard, are awarded for the resources consumed?

E<sup>3</sup> - effectiveness - do employers find the degrees and diplomas a useful way of assessing the qualities of potential employees?

## A complete conceptual product : part

I



## A complete conceptual product : part

### II

A university owned and operated system to award degrees and diplomas to suitably qualified candidates (X), by means of a suitable assessment (Y), in conformance

with national standards, in order to demonstrate the capabilities of the candidates to potential employers (Z).

A - university staff

T - candidates -> degree holders

W - the belief that awarding diplomas is a good way of demonstrating the qualities of candidates

to potential employers

O - the university governing body

E - national educational and

E<sup>1</sup> - are degrees and diplomas awarded ?

E<sup>2</sup> - how many degrees and diplomas, of

what standard, are awarded for

of

the

resources consumed?

E<sup>3</sup> - do employers find the degrees

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ...

83

## Group Work

Produce complete conceptual models for the recruitment agency case!

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ...

84

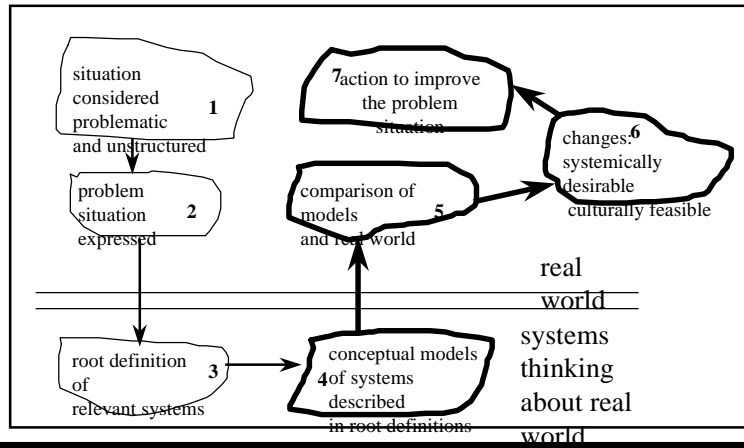
## Definition of Information Systems

- The Systems Approach
- -
- Conceptual Models and Organisational Reality



# Definition of Information Systems

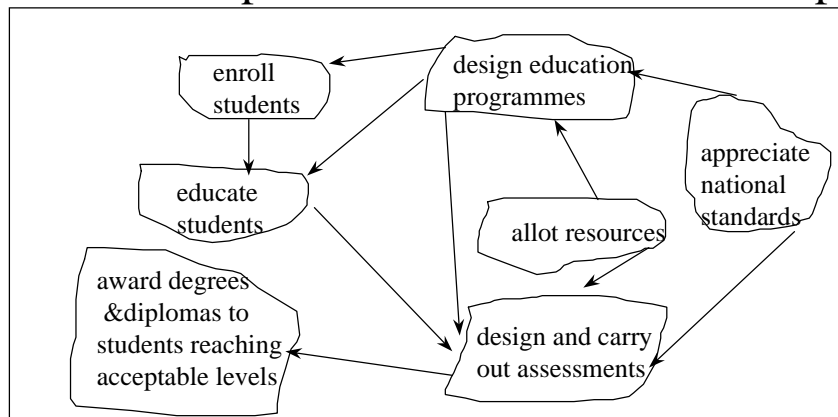
## Conceptual Models and Reality



Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

87

## Conceptual Models - Another Example



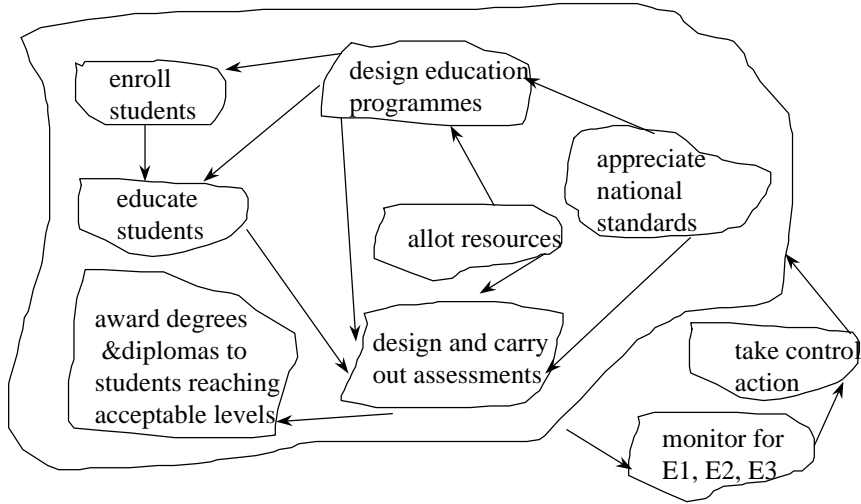
A university owned and operated system to award degrees and diplomas to suitably qualified candidates (X), by means of a suitable assessment (Y), in conformance with national standards, in order to demonstrate the capabilities of the candidates to potential employers(Z).

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

88

# A complete conceptual product : part

## I



# A complete conceptual product : part

## II

A university owned and operated system to award degrees and diplomas to suitably qualified candidates (X), by means of a suitable assessment (Y), in conformance

with national standards, in order to demonstrate the capabilities of the candidates to potential employers (Z).

A - university staff  
T - candidates -> degree holders  
W - the belief that awarding diplomas is a good way of demonstrating the qualities of candidates to potential employers  
O - the university governing body  
E - national educational and assessment standards

E<sup>1</sup> - are degrees and diplomas awarded?  
 E<sup>2</sup> - how many degrees and diplomas, of what standard, are awarded for the resources consumed?  
 E<sup>3</sup> - do employers find the degrees

## Comparing Model and 'Reality'

using

- general discussions
  - record difference between models and happenings
- historical construction
  - write a history informed scenario
- question definitions
  - does the activity exist? ...

## Comparing Model and 'Reality': Define Questions

activity	does it exist in the real situation?	how is it done?	how is it judged?	comments
enrol student				
educate students				
.....				
<b>links</b>				
1 --> 2				
2 --> 3				
.....				

\*1 --> 2 enrol students --> educate student.

## Action For Improvement

Decide on whether and which relevant systems to analyse further and more formally and whether and what kind of IT support might be appropriate.

The conceptual products (root definitions and conceptual models) can be a starting point for the next analysis step.

## Group Work

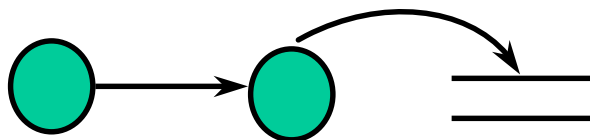
Produce conceptual models for the recruitment agency case and compare them with 'reality'!

What are the problems?

# Definition of Information Systems

Structured Analysis- Integrated Analysis

Overview and Basic Concepts



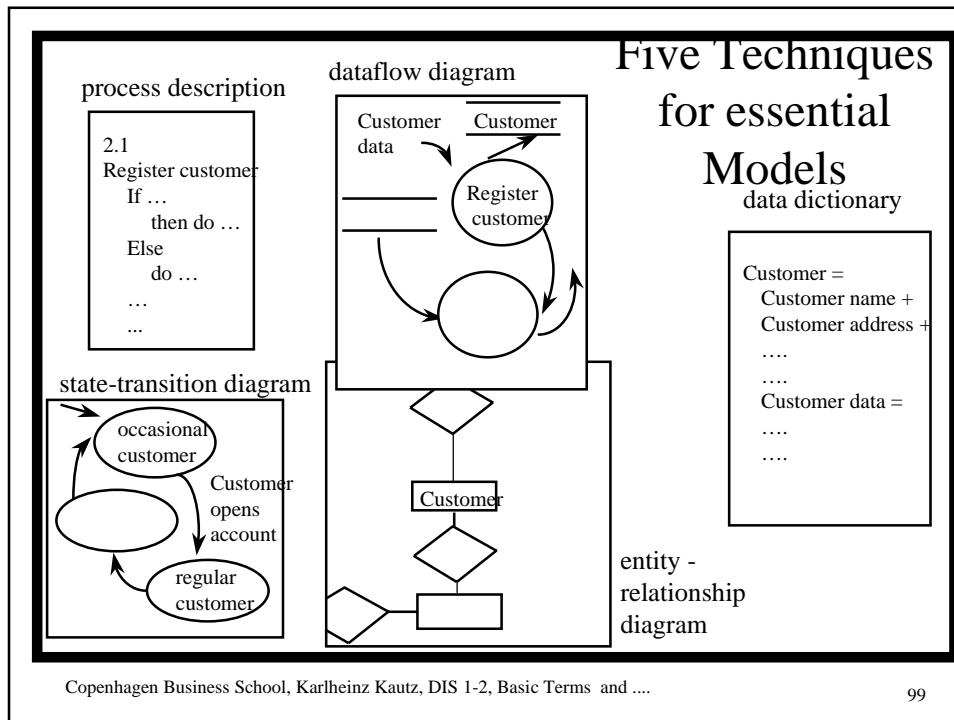
## Structured Analysis

- system analysis results in a part of a requirements specification which comprises an essential model of the future system and a design proposal for the system; the new physical system
- an essential model is technology neutral; it contains the requirements of the system, the functions which should be performed and the data which shall be stored
- a new physical model shows a system embedded in a particular technology and in a particular organization

## An essential model

is developed with the help of

- data flow diagrams
- entity-relationship diagrams
- state transition diagrams
- data dictionaries
- process descriptions



- ## The essential model - techniques and strategies
- event partitioning and functional decomposition
  - data integration
  - integrated system analysis; parallel work with functions and information
  - event driven; the systems events are starting point for a functional as well as an information analysis
- Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and .... 100

## The essential model - event partitioning

Comment:  
diagrams/graphical  
presentation can be found  
in original literature  
Delskov/Lange p.56

## Data flow diagrams

- **circle - activities, functions**
- **arrows - data flow**
- **parallel lines - data stores**
- **box - external process  
(source/terminator)**

Comment:  
diagrams/graphical  
presentation can be found  
in original literature  
Delskov/Lange p. 69

## Entity-relationship diagrams

Comment:  
diagrams/graphical  
presentation can be found  
in original literature  
Delskov/Lange p.90/94/95

## State diagrams

Comment:  
diagrams/graphical  
presentation can be found  
in original literature  
Delskov/Lange p. 102

## Data dictionaries

Comment:  
diagrams/graphical  
presentation can be found  
in original literature  
Delskov/Lange p. 106

## Process Descriptions

Comment:  
diagrams/graphical  
presentation can be found  
in original literature  
Delskov/Lange p. 112/113

## Group Work

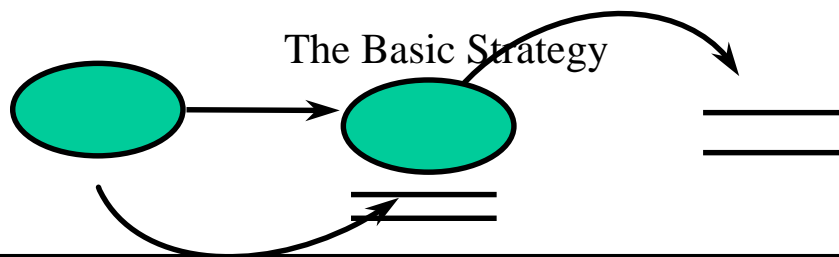
Produce a

data flow diagram  
entity-relationship diagram  
state-transition diagram  
data dictionary  
process

for a very small part of the bank case or  
the recruitment agency case!

## Definition of Information Systems

Structured Analysis -  
Integrated Analysis:



## Basic Strategy

- Analyze existing system
- Outline events
- Develop information model
- Analyze essential activities
- Integrate models
- Imbed new requirements
- Develop new physical model

## Event Partitioning

1. An essential activity is a collection of processes which represent the reaction to an event.

~~2. There is no data flow between activities.~~

### Outline events:

3. They are only connected by data stores.

1.step:

identify events

2.step:

for each event

outline the course of the essential activity

3.step:

define each activity's interaction with the environment

### Results:

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ...

111

## Information Modeling

Event list and/or context diagram by means of the identified data flow provide immense knowledge about entities and relations.

1.step

brainstorm entities:

*(Subject of the system: products, services, resources*

*Stakeholders: persons, organizations*

*Administration: categories, organizational units, places*

*Procedure/functions: events, contracts)*

2.step

for each event

investigate whether the event results in some action

produce a result model which contains data

about one event or data about actions

which are an outcome of the events

3 step

integrate result models into a synthesized model

delete redundancies

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ...

112

## Information Modeling

### 4.step

- for each (external) data flow
- for each data element
- assign element as a characteristic to an entity or a relation
- actualize with missing entities and relation
- develop data dictionary for the information model

### 5.step

- for each entity with more than one state
- develop a state-transition diagram
- control that all states can be found in the information model or the data dictionary

The model with the information model might identify some entities that the event lists and the support data flow diagrams have to be actualised as a consequence

113

## Detailed Analysis of Essential Activities

- develop activity diagrams (data flow diagrams) for each essential activity

### 1.step

- define fundamental processes (compute new data)
- define preserving processes (create new data instances, changes, attributes, status, etc.)
- define physical processes (control, transport, translate in relation to environment)

for each process

- draw an unconnected circle
- with incoming and outgoing data flow and
- write processes on data stores
- connect processes in the essential order

### 2.step

- balance processes
- define read accesses

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ...

114

## Integrate Models

### 1.step

- balance essential model
  - balance each entity and relation
    - check that entity (relation) is used (read)
    - check that the model contains the necessary preserving processes
    - check whether entities (relations) miss or have unnecessary attributes
    - pay attention to data which is used by other systems
    - check and mark shared processes
  - actualize E/R diagram, data dictionary, data flow diagrams and process descriptions

### 2.step

- develop high-level data flow diagrams
  - group entities/relations in entity groups
  - group activities in activity groups
  - group data-flows in data channels
  - draw high level diagrams

## Imbed new Requirements

### 1.step

- for reach essential requirement
  - identify new events
  - identify new data (entities, relations, states, attributes)
  - identify new preserving processes
  - identify add-ons (update of existing essential activities)

### 2.step

- modify information model
  - with new entities, relations, attributes
  - analyse new essential activities
  - update existing essential activities and process descriptions
- actualise high-level data flow diagrams

## Develop new physical Model

### 1.step

for each entity/relation  
decide storage medium and method

### 2.step

for each essential activity  
determine processor  
develop sketch diagram or physical diagram by  
incorporating  
intermediate stores  
physical data flows (data flow transmitted by  
communication media)  
error situations and correction procedures

for each physical activity

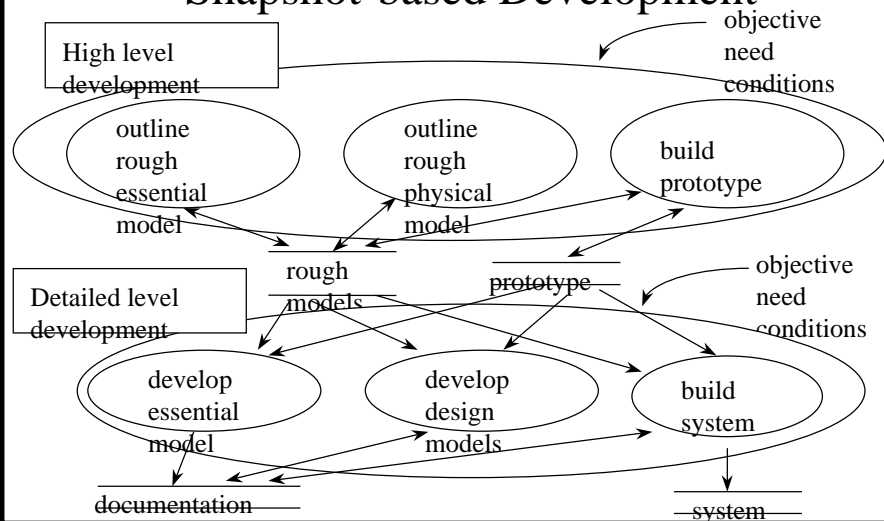
develop maxi process description n or  
physical detail diagram for each physical activity

by incorporating

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....  
control, translation and transport

117

## 'Blitzing' Snapshot-based Development



Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

118

## Blitzing' Snapshot-based Development

### Precision/Limitation

find system's objectives

### Brain storming

find system's entities

find system's events

find system's environment

### Model building

develop result models

develop state-transition diagrams

delimit essential activities

evaluate system's procedures

## Group Work

Produce an

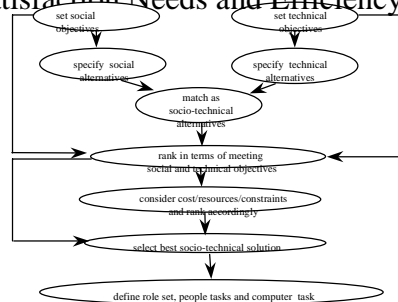
event list

for a very small part of the

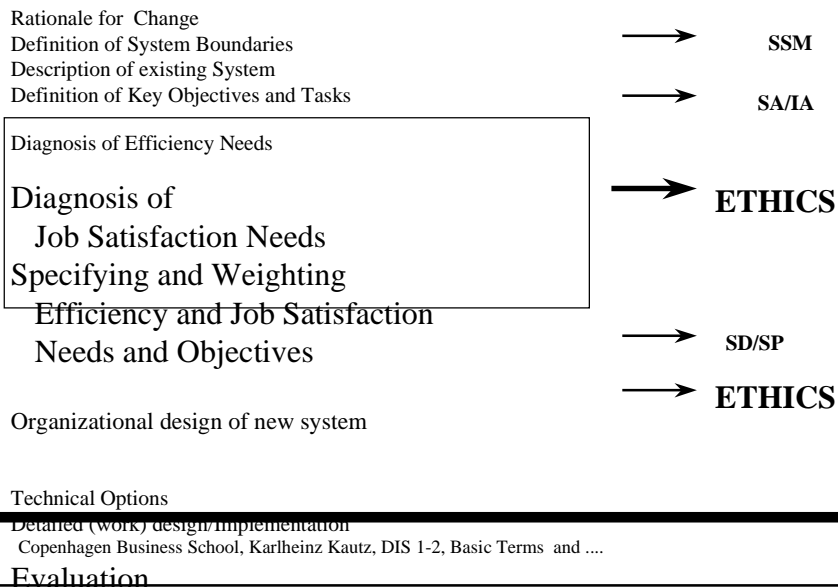
bank case or

the recruitment agency case!

# Definition of Information Systems Socio-technical Analysis: Job Satisfaction Needs and Efficiency Objectives

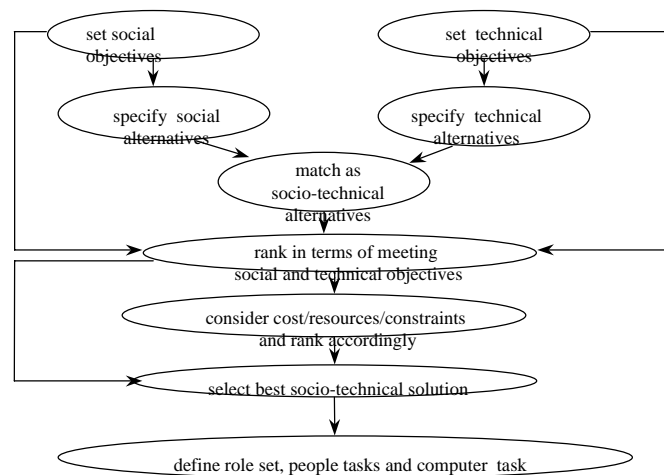


## Socio-Technical Analysis and Design:



123

## Socio-Technical Analysis and Design:



Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

124

## Basic Measuring Job Satisfaction

There are 5 areas of measurement:

- knowledge fit
- psychological fit
- efficiency fit
- task-structure fit
- ethical fit

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ...  
These should/can be checked before and after the design of a new system to find solutions matching efficiency

## Measuring Job Satisfaction

- knowledge fit:  
a good fit exists when the employees think that their skills are adequately used and that their knowledge develops and makes them more competent

Possible questions/statements

agree disagree neither/nor

1. My skills and knowledge are fully used in my present job
2. I find my work extremely interesting

3. I do not like my present job

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ...

## Measuring Job Satisfaction

- **psychological fit:**

a good fit exists when the employees think that their personal aspirations for recognition, achievement, responsibility, status and other psychological need factors are adequately met within the work situation.

### Possible questions/statements

ag dis n/n

1. Status is important for me. I like to be respected
2. Working here gives me a feeling of status
3. I carry considerable responsibility in my job
4. I should like to carry more responsibility
5. If I do good work, I feel management recognizes this
6. I should like my work to receive more recognition
7. My job is a very severe one
8. I enjoy the opportunity for making friends which my work provides

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

127

9. There are not sufficient opportunities for promotion

## Measuring Job Satisfaction

- **efficiency fit:**

a good fit exists when the employees think that they achieve the support services, rewards, and controls which they require

### Possible questions/statements

ag dis n/n

1. I could earn more money if I did not work here
2. I am paid adequately for the work I do
3. I have to work too hard in my job
4. We are expected to be too accurate in our work
5. Methods for checking are too lax
6. Supervision is very strict
7. I like to be left to get on with my work without interferences from supervision
8. My supervisors give me all the help I need

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

128

9. The bonus system is fair to everyone

## Measuring Job Satisfaction

- task-structure fit

a good fit exists when the employees think that technology and task structure produce a work situation wherein they have sufficient work variety and opportunities for the use of discretion which broadly fit their personal requirements in terms of skill level, meaningful tasks, autonomy and control

### Possible questions/statements

ag dis n/n

1. There is not much variety in my work
2. I should like a job which was less routine
3. There is little scope for me to use my own initiative
4. I should like a job which expected me to use my initiative more

5. In my job I have to rely on my own judgement

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

129

## Measuring Job Satisfaction

- task-structure fit (cont.)

### Possible questions/statements

ag dis n/n

10. I should like clearer target s to aim at
11. How well I can do my work depends on my co-workers
12. It would be better for me if I could do my work without depending on others
13. I can organise and carry out my work the way I want
14. I should prefer more freedom to plan my work myself

15. I believe my job to be important and others recognise its importance

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

130

## Measuring Job Satisfaction

- ethical fit

a good fit exists when the employees think that the organisation meets their values which concern communication, consultation, participation, and other aspects of human relationships

### Possible questions/statements

ag dis n/n

1. Top management here is too ruthless
2. Managers and workers here are very friendly
3. The organization puts production above the interests of its employees
4. The organisation looks after the welfare of its staff very well
5. A person's character and experience count for more here than formal qualifications

6. I believe that character and experience are more important than qualifications

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

131

## Measuring Job Satisfaction

- ethical fit (cont.)

### Possible questions/statements

ag dis n/n

9. Senior management is out of touch with the way the employees feel
10. We have sufficient say in the way the organization is run
11. Employees should be consulted more about major decisions that are to be made

12. We cannot sufficiently influence decisions

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

132

## Measuring Job Satisfaction

- an alternative approach

### A. Contents of Work

very specialised job  
I \_\_\_\_\_ I

varying, different job tasks

strong division of labour  
I \_\_\_\_\_ I

no division of labour

structured, pre-determined  
I \_\_\_\_\_ I

determined methods, and sequence work tasks of techniques are self-

rule-based  
I \_\_\_\_\_ I

contingency based

## Measuring Job Satisfaction

- an alternative approach

### B. Autonomy and steering

control of work situation  
I \_\_\_\_\_ I

self-steering/control

stressing work pressure  
I \_\_\_\_\_ I

no requirements for performance

no influence on own job  
I \_\_\_\_\_ I

big influence on own job

no influence on the company's work conditions  
I \_\_\_\_\_ I

big influence on the company's work conditions

## Measuring Job Satisfaction

- an alternative approach

### C. Social relationships

no security \_\_\_\_\_ complete security  
I \_\_\_\_\_ I

little opportunity \_\_\_\_\_ god opportunity  
for self-realisation for self-realisation  
I \_\_\_\_\_ I

no contact with others \_\_\_\_\_ much contact with others  
I \_\_\_\_\_ I

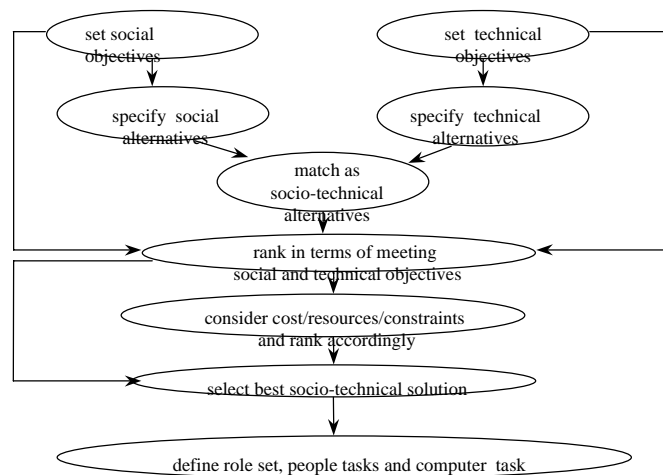
alienated \_\_\_\_\_ well-integrate  
I \_\_\_\_\_ I

### D. Education and personal development

no educational \_\_\_\_\_ many educational  
requirements requirements  
I \_\_\_\_\_ I

I \_\_\_\_\_ I

## Socio-Technical Analysis and Design:



## Group Work

Start developing a method for

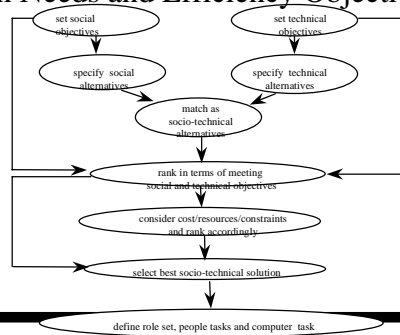
setting social objectives  
social alternatives

technical objectives  
technical alternatives

matching the socio-technical alternatives  
ranking them

# Definition of Information Systems

- **Socio-technical Analysis:**
- **Job Satisfaction Needs and Efficiency Objectives**



Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

139

## Socio-Technical Analysis: An Example

### **A root definition:**

A system owned by the company and applied by customer consultants with the purpose of as fast as possible satisfying the consultants' needs for information possibly held by other consultants to provide good customer service and to increase their individual and the company's turnover within the company's business guidelines.

Copenhagen Business School, Karlheinz Kautz, DIS 1-2, Basic Terms and ....

140

## Evaluating 'Job' Satisfaction

Factor	Consultant 1		Consultant 2		Consultant 3		Consultant 4	
	Present stage	Import ance						
Use of individual capabilities	A	1	B	1	A	2	B	1
Further education	A	1	A	1	A	1	A	2
Variety of work tasks	B	1	A	1	B	3	A	1
Clear goals	A	1	B	2	A	3	A	2
Continuous feedback	A	1	A	1	A	2	A	2
Recognition	A	3	A	2	A	2	A	1
Co-determination	C	1	C	2	B	3	C	1
Responsibility	C	1	C	2	B	3	C	1
Rewards	A	3	C	2	C	3	C	2
Control and monitoring	A	1	A	3	A	4	A	3
Access to support services	D	1	C	2	D	1	D	1
Status	A	3	B	3	B	4	B	3
Promotion	B	1	A	2	A	2	B	1
In agreement with service demands	A	1	A	1	B	1	A	2
In agreement with Sales demands	A	1	A	2	B	2	A	2

A = very satisfied .....C= on a middle level ..... E= very problematic  
1= very important ..... 3 = neutral .....5= not important at all

## Social and Technical Objectives

- SO1: Access to support
- SO2: Use of one's own capabilities
- SO3: Job promotion
- SO4: Variety of work tasks
- SO5: Further education
- SO6:
- TO1: Provision of more support
- TO2: Fast access to information
- TO3: Access to correct information
- TO4: Profitable for the company

## Social Alternatives

- SA1: Self-Determined Work Tasks  
(Combined with Individual Feedback)  
SO2, SO3, SO6
- SA2: Reduction of Administrative Tasks  
SO2, SO4
- SA3: An 'informal' Group Debate Forum  
SO1, SO2, SO5, SO6
- SA4: A 'formal' Support System  
SO1, SO2, SO3, SO5, SO6

## Technical Alternatives

- TA1: A Bulletin Board  
TO1, TO4
- TA2: (Individual) Folders  
TO1, TO4, TO3 (partly)
- TA3: Additional Competent Staff  
TO1, TO2, TO3
- TA4: A (distributed) IT application  
TO1, TO2, TO3, TO4

## Matching and Ranking Alternatives

### Step 1

Technical Alternatives/ Social Alternatives	TA 1: A Bulletin Board	TA 2: (Individual) Folders	TA 3: Additional Competent Staff	TA 4: A (distributed) IT application
SA 1: Self-Determined Work Tasks (Combined with Individual Feedback)	15	-	19	20
SA 2: Reduction of Administrative Tasks	13	-	-	18
SA 3: A n 'informal' Group Debate Forum	19	-	23	24
SA 4: A 'formal' Support System	23	24	27	28

The numbers in the cells are computed as follows:  
SA 1, TA 1 f. ex. satisfies SO 2, SO 3, SO 6, as well as TO 1  
and TO 4

There are 6 social objectives, the most important one  
receives 6 'points', the 2<sup>nd</sup> 5 etc.

There are 4 technical objectives, the most important one  
receives 4, the 2<sup>nd</sup> 3 etc,

Thus

## Ranking Alternatives

Social and Technical ...	Costs	Resources	Constraints/Restrictions
SA4/TA3: A 'formal' Support System — Additional Competent Staff	Staff salary/wages Staff time	Competent staff	Competent staff are a scarce resource
SA4/TA4: A 'formal' Support System — A (distributed) IT application	Reduced social contact Implementation costs: Development costs Introduction costs Educational costs Development of 'Administrative' routines	'Technical' and 'Administrative' Support system Competent staff have to formalise 'questions' and 'answers'	Implementation Costs

- Analyze costs/resources/constraints vs. value
- Choose alternative(s)
- Define roles

Computer Aided Software Engineering

-

Computer Aided Systems Engineering

-

CASE

Part I

What is CASE ?

## Definition of CASE Tools

### What is a CASE tool?

any computer software and/or system which is specifically designed to support any sub-stage, stage or stages of the information system development process of a computer-based information system or any aspect of the management of these tasks and processes

(Avison & Fitzgerald, 1995)

## CASE Tools: A Brief History

### 1. Generation (1984-1986)

simple drawing tools implemented as stand-alone applications to be used by individuals

### 2. Generation (1987-1989)

improved graphical features; simple code generation possibilities, interfaces to other tools, clusters of several machines integrated

### 3. Generation (1990 - ...)

multi-tool integration, shared repositories, enterprise-wide CASE usage

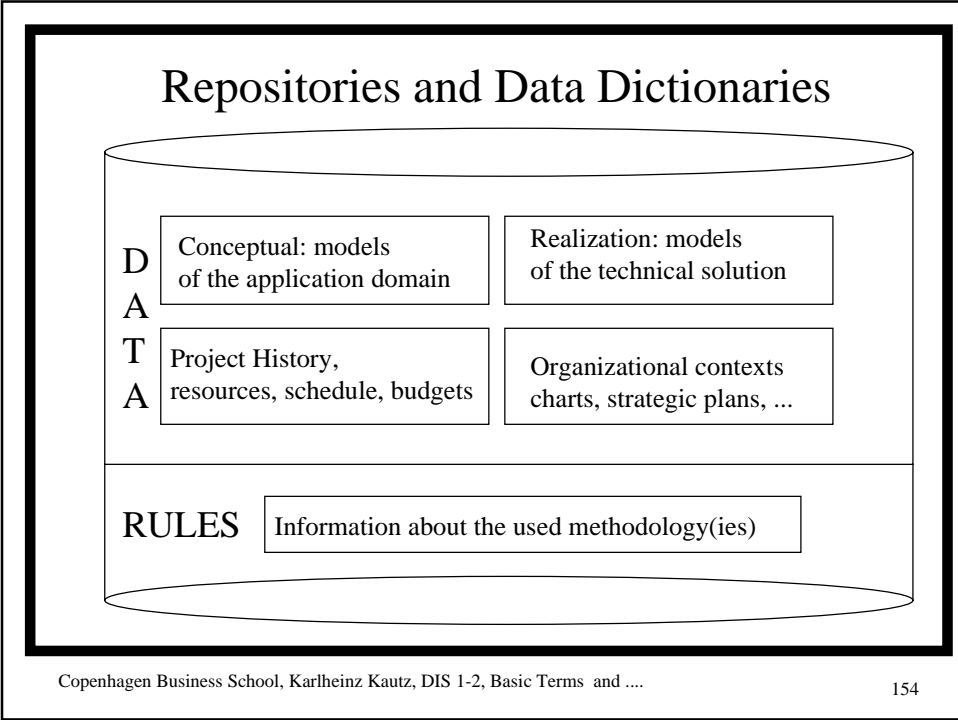
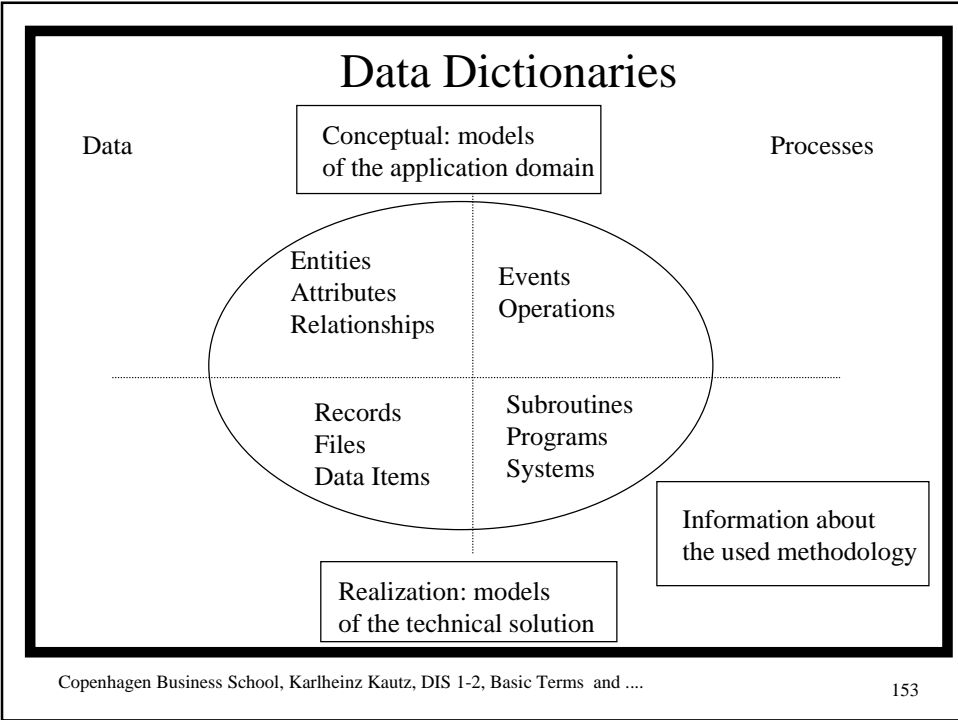
First Case tools in Denmark around 1987.

## CASE Tools - Main Features:

- Creating models of user requirements in graphical form: data flow diagrams, entity relationship diagrams
- Creating software design models for the data and the procedures
- Analysing and checking consistency and cross references of all analysis and design, identifying errors
- Building prototypes and allowing simulation
- Establishing and enforcing organizational development standards
- Storing and administration of project management information

## Case Tools and Methodologies

- Case tools as methodology copycats
- Case tools as methodology companions
- Case tools as methodology neutral aids



## Repositories: Some Key Questions

- Does a metamodel for the repository exist?
- What reports are provided? Are online browsing and searching capabilities provided ?
- How the diagrams stored? Does topological information get lost?

## Diagramming Capabilities of CASE Tools

- What control do users have about placement, size and font of text in diagrams?
- How quickly can diagrams be changed and redrawn?
- How easily can diagrams be saved and retrieved?
- How many objects can be placed in one diagram/on one diagram page?
- How many different diagrams of the same type and of different types (different models) can be seen at the same time?

## Diagramming Capabilities of CASE Tools

- What capabilities for in- and out-zooming do exist?
- What capabilities for free form drawings and annotations do exist?
- What kind of control do users have about placing, shaping and 'rubber-banding' arrows?
- How closely do the diagrams approximate WYSIWYG?

## Diagramming Capabilities of CASE Tools

- What kind of consistency checks and error checks are done (within one diagram, within the different levels of one diagram, between different diagrams (models))?
- Can the users control when consistency and/or error checking is done?
- How does the user move between levels of diagrams?

## Group work:

Test the tool PowerDesigner:

PowerAnalyst - basically dataflow diagrams

DataArchitect - basically information model

MetaWorks - checking consistency between models?

Which of the five models of the SA/IA are supported and in what way?

Use the examples from Delskov/Lange!

## Part II: Diffusion of and Experience with CASE Tools

in  
Denmark and Finland  
(from 1992)

## Who reported?

Denmark: 222 organisations were asked - 64 gave valid answers

Finland: 72 organisations were asked- 38 gave valid answers

33% general managers-29% analysts-21% project managers

most Case using organisations are fairly large

average project team size 3-5 persons

majority in projects of 1 - 5 person-years

## How much are the tools used?

most respondents have 2 or less years of CASE experience and have used the tools in at most 2 projects

the tools are generally poorly diffused in the organisations

half of the enterprises have used their tool for 2 or less projects, less than a third in 6 or more projects

in the majority of enterprises only up to 25% of the analysts use the tool

after 3 years purchasing/initial use many are still not in a routine use situation

it looks as if it takes a long time from buying to extensive use!

why? poor management support, poor method qualifications ...

## How was the CASE tools introduced?

- in more than 50% IT management took initiative; in 27% analysts took the initiative data flow diagrams
- the tools evaluated in a trial project, by management only, by an expert group, by customers or by methodology experts
- the tool was consistent with the standard techniques in 50% of the cases and in 45% of the IS organisations it was the standard
- only in 50% of the cases a post trial period evaluation took place

## Difficulties during Introduction

- lack of experience with methods was significant in 50% and very significant in 25% of the cases
- problems with adapting to the methodologies supported by the tool were significant in more than 20% of the cases
- Inadequate user qualifications, problems with the working procedures and organisational culture problems were significant in more than 20% of the cases as were software and installation problems; novices had more problems here
- vendor problems and resistance to change were also significant in more than 20% of the cases; experienced users had more problems here

## Changes through CASE

- new working procedure were developed in more than 50% of the cases
- 2/3 who had no new working procedures were considering it
- drawing conventions were developed in more then 65%
- naming conventions in more than 50%
- standard reporting (milestones) were introduced in nearly 50%
- changes to the tools were only made in less than 20%

## Training and Support for the Introduction

- self study in more than 80% of the cases
- internal training/supplier course in more than 60% of the cases
- tutorial in more than 50% of the cases
- estimated time to be productive 2 weeks - 3 month in more than 75% of the case; 2 weeks - 1month more than 50% of the cases
- hotline service, central help desk, user group in more than 50% of the case
- novices are not organised in user groups and use central helpdesk

## CASE Tool Usage

- the tools are mainly used to support analysis and design; where Danes seem to follow SA quite closely, but in general use less techniques
- more than 80% use the tools to produce data flow diagrams and entity-relationship diagrams (no info how they use the tool)
- more than 50% used these techniques already before the tool was introduced
- on average one technique supported by the tool is not used and the organisations use at least one technique manually (as it is not supported by the tool)
- > 80% use the tools for drawing and consistency checking;  
> 70% for report generation ;> for central data-management,  
< 20% to produce input for other tools

## Performance Evaluation

- very positive:  
check capabilities, central data dictionary, quality of diagrams, reliability, user friendliness, coverage of life cycle activities (and this was what they expected ...)
- positive:  
time to master tool, vendor support, tool documentation, reporting capabilities, response time
- not so positive:  
text editors, interfaces to code generation
- mixed feelings:  
database interface, growth potential, price

## Meeting the Objectives

Before:

- Improved Quality
- Improved Productivity
- Improved Procedures
- Standardisation (methods)
- Leading Edge

After:

- Improved Quality : > 55% significantly met
- Improved Procedures : > 55% significantly met
- Standardisation (methods) : >50% significantly met
- Improved Productivity
- Leading Edge

## Further Experiences

- increased project and product standardisation
- common development environment
- more iterative development process (55%), less iterative (21%)
- only indirect use of other tools (word processors, 4GLs. ...)
- a well planned strategy is prerequisite to benefit from CASE
- substantial training is a prerequisite
- real beneficial CASE should support all life cycle activities
- management support is necessary
- education in methodology

## Main Strength

More than 50% state that

check consistency

support software evolution

improvement of understandability

automate routine procedures

introduce discipline in the development process

see as main strength

## Productivity, Quality, Profitability Gains

- 46% indicate that project duration has slightly decreased
- 42% disagree that development has been sped up
- the saved time might be used for improved documentation:  
81% agree that documentation has improved
- 50% said that functionality had been enhanced;  
50% refuted this statement
- 31% indicate poor return-on-investment;  
47% good return on investment
- however 95% had not performed specific calculations

## Conclusion

The diffusion of CASE is dependant on a mixture of organisational and technological conditions like

- technical characteristics of the tool
- tool user qualification
- acceptability
- profitability
- scale of investment
- change agents
- etc .... (to be identified later)