

florence

Et forskningsprosjekt om bruk av edb som hjelpemiddel i sykepleie,
ilknyttet det nordiske forskningsprogrammet SYDPOL.

Institutt for informatikk, Universitetet i Oslo, Postboks 1080 Blindern, 0316 OSLO 3.

Henrik Sinding-Larsen 45 43 21
Grete Thingelstad 45 41 66
Inger Sannes 45 41 68
Kristen Nygaard 45 41 65
Jens Kaasbøll 45 43 21
Tone Bratteteig 45 41 65
Gro Bjørknes 45 41 65

Å implementere en idé - samarbeid og konstruksjon i Florence-prosjektet

Rapport nr. 3 fra Florence-prosjektet
(juni 1986 - juli 1987)

FORORD.

Florence-prosjektet er et forskningsprosjekt om informatikk og sykepleie. Prosjektet ble initiert av Institutt for Informatikk, Universitetet i Oslo, i 1983. Fra høsten 1984 til våren 1985 var prosjektet et samarbeid mellom Institutt for Informatikk og Voksentoppen Institutt for barn med astma og allergi, ved Rikshospitalet. Fra desember 1985 til sommeren 1987 var prosjektet et samarbeid mellom Institutt for Informatikk og Kardiologisk seksjon ved Sentralsykehuset i Akershus.

Siste halvår av 1987 fortsetter prosjektets sykepleiergruppe ved Sentralsykehuset i Akershus med utprøving og bruk av prosjektets edb-system.

I deler av prosjektet har datamaskinleverandøren Norsk Data deltatt i samarbeidet. Norsk Data har bidratt med maskiner, programmer og konsulenter til en prototypingsperiode på 3 måneder våren 1985 og fra sommeren 1986 og ut prosjektperioden. Norsk Data samarbeider med Sentralsykehuset også siste halvår av 1987.

Utover disse har Norges Teknisk-Naturvitenskapelige Forskningsråd, Overordnet Styringsgruppe for EDB i helsevesenet (Sosialdepartementet), Akershus Fylkeskommune og Anders Jahres fond for medisinsk forskning bidratt med midler til prosjektet.

Dette er den tredje rapporten fra Florence-prosjektet, og den markerer avslutningen på prosjektet. Rapporten er i hovedsak skrevet av informatikerne i prosjektet: Gro Bjerknes og Tone Bratteteig. Vi kan imidlertid bare fortelle om våre egne erfaringer, og om hva vi tror sykepleierne mener. Unni Krogstad, avdelingssykepleier på Kardiologisk seksjon ved Sentralsykehuset i Akershus, har skrevet om sykepleiernes tanker om idéene i prosjektet. Vi håper at to beskrivelser av den samme prosessen vil illustrere forskjellene mellom de to gruppene som har samarbeidet, og at forskjeller kan være inspirerende som grunnlag for samarbeid.

Andre del av den avsluttende rapporteringen er teknisk dokumentasjon og foreligger i eget bind, som distribueres av Norsk Data. I tillegg regner vi med at det siste halvåret med utprøving på Sentralsykehuset i Akershus vil dokumenteres i en egen, mer sykepleieorientert rapport ved årets slutt.

Professor Kristen Nygaard var initiativtaker til prosjektet, og har hele tiden stått ansvarlig overfor de bevilgende myndigheter. Vi vil takke alle som har bidratt til prosjektet. Vi har hatt kontakt med hyggelige folk i Norsk Data og Unique, i Overordnet Styringsgruppe for edb i helsevesenet og Akershus Fylkeskommune.

Vi vil også takke samarbeidspartnerne våre ved Rikshospitalet og Sentralsykehuset i Akershus. Vi vil rette en spesiell takk til vår sekretær Grete Thingelstad, som har vist seg å være helt uunnværlig i prosjektet.

Blindern, 1/9 1987

Gro Bjerknes & Tone Bratteteig

INNHold.

FORORD

INNHold	s. 2
DEL I: OM FORSKNINGSPROSJEKTET FLORENCE.	s. 3
1: Fakta om prosjektet.	s. 3
2: Om yrkesorienterte språk.	s. 5
3: Om Florence som forskningsprosjekt.	s. 7
DEL II: OM SYSTEMUTVIKLING MED BRUKERE.	s. 9
1: Bruksperspektivet i praksis.	s. 9
2: Systemutviklingsprosessen fra informatikernes synspunkt.	s. 11
2.1: Arbeidsformer.	s. 12
2.2: Læringsaspekter ved prototyping.	s. 13
3: Systemutviklingsprosessen fra sykepleiernes synspunkt.	s. 15
DEL III: OM PRODUKTET: ARBEIDSARK-SYSTEMET.	s. 19
1: Arbeidsark som idé: informatisk og sykepleie-basert begrunnelse.	s. 19
1.1: Informatisk begrunnelse av Arbeidsark-idéen.	s. 19
1.2: Sykepleie-messig begrunnelse av Arbeidsark-idéen.	s. 24
2: Krav til Arbeidsark-systemet.	s. 25
3: Beskrivelse av Arbeidsark-systemet.	s. 26
4: Diskusjon av Arbeidsark-systemets datamodell.	s. 34
5: Valg av programvare og erfaringer fra programmeringen.	s. 39
LITTERATUR.	s. 44
VEDLEGG 1: Utkast til samarbeidsavtale mellom Institutt for Informatikk, Universitetet i Oslo og Sentralsykehuset i Akershus.	s. 47
VEDLEGG 2: Samarbeid med datamaskinleverandør.	s. 60
VEDLEGG 3: Innholdsfortegnelse i "Dokumentasjon av Arbeidsark-systemet. Rapport nr. 4 fra Florence-prosjektet".	s. 68

DEL I. OM FORSKNINGSPROSJEKTET FLORENCE.

Florence-prosjektet har hatt som mål å utvikle og spre kunnskap om hvordan sykepleiere kan bruke elektronisk databehandling som et hjelpemiddel i sitt daglige arbeid. Prosjektet har vært rettet mot post- og avdelingsnivå i helsevesenet.

Basis i prosjektet er tanken om at det er yrkesutøverne selv som skal ha kontrollen over innføringen av edb i yrket. Yrkesutøverne skal selv kunne påvirke og styre utviklingen av faget sitt. En viktig forutsetning for slik kontroll er at de er kunnskapsmessig forberedt til å vurdere de teknologiske og organisatoriske løsninger som tilbys. Yrkesutøverne må kunne formulere egne ønsker og krav til teknologien, og de må selv kunne endre de edb-systemer de bruker i sitt arbeid.

For å forske om informatikk og sykepleie har vi i Florence-prosjektet drevet delvis med analyse og delvis med design. Systemutviklingen ble gjennomført for å vinne kunnskap om systemutvikling, ikke for å utvikle et produkt. Denne rapporten skal dokumentere noen av våre læringsresultater som kan være interessante for andre. Denne første delen gir en kort innføring i hva prosjektet har handlet om, og hva vi har gjort. I den neste delen beskriver vi noen av erfaringene fra systemutviklingen, og vi vil her ta for oss arbeidsformene vi har brukt i prosjektet. I del III beskriver vi det edb-systemet som ble resultatet av Florence-prosjektets systemutvikling, og vi diskuterer de valgene vi har gjort underveis i design-prosessen.

1. FAKTA OM PROSJEKTET.

Florence-prosjektet ble initiert av Institutt for Informatikk, Universitetet i Oslo, i 1983. Prosjektet har pågått i nesten 4 år, fra høsten 1983 til høsten 1987, ved instituttet. Deler av tiden har prosjektet foregått som et samarbeid med et sykehus.

Florence-prosjektet har hatt ressurser fra flere kilder:

- Universitetet i Oslo, hvor vitenskapelige assistenter har brukt sin forskningstid,
- midler fra Norges Teknisk-Naturvitenskapelige Forskningsråd (NTNF), til drift av prosjektet og lønn til en forsker (sosialantropolog) og en sekretær,
- Overordnet Styringsgruppe for EDB i helsevesenet (OSG) i Sosialdepartementet og
- Akershus Fylkeskommune (i 1986) har bidratt til lønn til en sykepleierstilling.

I tillegg har vi fått penger til å kjøpe maskinutstyr av

- Anders Jahres fond for medisinsk forskning, og
- Norsk Data har bidratt med maskiner, programmer og konsulenter, spesielt i 1987.

I løpet av denne tida har vi vært på to sykehus, på to nokså forskjellige sykehusposter:

- høsten 1984 og våren 1985: Voksentoppen Institutt for barn med astma og allergi, som er en del av Rikshospitalets barneavdeling, men som ligger geografisk atskilt fra resten av sykehuset. Avdelingen er økonomisk selvstyrt.
- senhøsten 1985 og ut prosjektperioden: Kardiologisk seksjon - hjerteovervåkingen - ved Sentralsykehuset i Akershus (SiA)¹

Våre konklusjoner fra prosjektet stammer ofte fra det vi har opplevd av likheter og forskjeller mellom de to avdelingene. Vi skal derfor presentere dem noe nøye.

Vårt første møte med sykepleiere skjedde på Voksentoppen. Det er ca. 40 ansatte der, stort sett kvinner. Pasientene er kroniske allergikere som legges inn for å sjekke og evt. endre behandling, og de ligger inne til observasjon og testing 10 - 14 dager. Avdelingen

¹I vedlegg 1 finnes utkast til samarbeidsavtale mellom Institutt for Informatikk, Universitetet i Oslo og Sentralsykehuset i Akershus. Utkastet er tatt med for å gi et bilde av grunnlaget for samarbeidet i prosjektet, og sier noe om de planene og forventningene partene hadde før samarbeidet kom i gang.

er også akutt-avdeling for allergikere i Oslo-området. Det er vanligvis 10 - 20 pasienter i alderen 1 - 16 år inne, og det drives barnehage og skole ved avdelingen om dagen. Pasientene ligger med andre ord ikke i sengene sine, og fordi det er kronisk syke barn det er snakk om, er kontinuiteten i pleien viktig. Avdelingen driver primærsykepleie, og arbeidet er organisert i flerfaglige team som legger opp pleie og behandling for hver pasient. Utover medisinerings-apparatur bruker avdelingen lite teknologi.

Kardiologisk seksjon på SiA, SYD 5, er ganske forskjellig fra Voksentoppen. SYD 5 er en vanlig sykehusavdeling, med ca. 40 ansatte. Den tar imot akutt syke som har diagnose eller diagnoseforslag som berører hjertet. Det er 6 - 9 overvåkningspasienter og 9 - 15 noe mindre dårlige hjertesyrke pasienter. Pasientene ligger inne i 3 - 7 døgn, og de ligger stort sett i sengene sine. Pleien er organisert i to grupper, én har ansvar for overvåkningsrommene og én for resten av avdelingen. Avdelingen har høy status på sykehuset, og nesten halvparten av sykepleierne er menn. Alle pleierne på avdelingen er vant til å håndtere edb-teknologi (f.eks. EKG).

Det er store forskjeller mellom disse avdelingene, og det var til å begynne med vanskelig å se likhetpunkter fra SYD 5 til Voksentoppen. Likhetene mellom avdelingene ligger i at både hjertesyrke og allergikere / astmatikere er kronisk syke som kan få akutte anfall. Det er med andre ord sannsynlig at en pasient vil komme tilbake etter første gangs innleggelse. Ved senere innleggelser vil sykehistorien inneholde viktige opplysninger som letter klargjøringen av pasientens diagnose og pleiebehov. Videre er pasientenes psykiske helse viktig i sykdomsbildet, for å forstå og behandle sykdommen. Pleie av kroniske pasienter involverer også undervisning om å leve med sykdommen i dagliglivet.

Våre erfaringer i prosjektet varierte også av andre årsaker enn forskjeller mellom avdelingene. Vår gruppe på Institutt for Informatikk (IfI) var forskjellig organisert i de to periodene. I Voksentoppen-perioden hadde vi ansatt en sykepleier som forsker på IfI: Inger Sannes. Hun hadde tidligere jobbet på Voksentoppen, og brukte noe tid på avdelingen slik at andre sykepleiere skulle kunne ta ut tilsvarende tid til prosjektet. På SiA ble det tilført en stilling til SYD 5 og sykepleierne kunne ta ut ca. 38 timer i uka til prosjektarbeid. Fra sommeren 1983 til sommeren 1986 var sosialantropologen Henrik Sinding-Larsen ansatt i prosjektet, og det første året deltok Jens Kaasbøll deltok som den tredje informatikeren. Videre har tilsammen 7 hovedfagsstudenter bidratt med arbeid tilknyttet prosjektet.

En annen viktig forskjell mellom arbeidssituasjonen i de to periodene er avstanden mellom Universitetet og sykehusene. Til Voksentoppen brukte vi ca. 20 minutter og til SiA ca. 35 minutter med bil. Med offentlig kommunikasjon, som jo var aktuelt ettersom vi ikke har bil, brukte vi henholdsvis 45 minutter og 1 time og 15 minutter hver vei. Vi undervurderte hvordan reiselengden spilte inn på hvordan vi organiserte arbeidet, særlig når det gjaldt SiA: i utgangspunktet planla vi prosjektet som om prosjektstedet var nær. Dette medførte at vi trodde vi kunne greie oss med maskinutstyr ett sted, og at vi kunne ha korte og hyppige besøk på SYD 5. Virkeligheten ble annerledes. Siden vi brukte så mye tid på å reise, ble det til at vi kom sjelden, men at vi var der nokså lenge hver gang. Vi burde ha planlagt som om prosjektet hadde vært 7 mil unna. Det hadde vært en riktigere tolkning av situasjonen. Og det viste seg ganske riktig at produktiviteten økte da vi fikk duplisert utstyret, og kunne programmere på Universitetet.

Aktivitetene våre på de to postene var delvis nokså like og delvis forskjellige. Arbeidet i prosjektet har hele tiden vært organisert for å tilrettelegge for læring for begge parter. Vi har ikke gjennomført tradisjonelle undervisningsopplegg, men heller lagt vekt på at hver av gruppene skal formidle hva de oppfatter som viktig i faget sitt til de andre. Basis for å kunne forstå dette er en opplevelse av hva henholdsvis sykepleiere og informatikere gjør i arbeidet sitt. Det betyr at det har vært viktig å observere og intervju sykepleierne på jobb, mens de jobber. Sykepleierne på sin side fikk prøve å bruke edb-systemer, og var på ekskursjoner for å se edb i bruk. Til sammen dannet disse erfaringene grunnlag for mer teoretiske og prinsipielle diskusjoner om bruk av edb i sykepleie.

På Voksentoppen brukte vi mye tid for å forstå hva som foregikk, vi var på relativt mange besøk og var med på ganske mange aktiviteter. Formålet var deltakende observasjon, dvs. vi ville gjerne finne ut hva sykepleieres arbeid *egentlig* var. Vi brukte tid på en nokså teoretisk opplæring om edb, og vi var på ekskursjoner. Vi arrangerte et 2-dagers seminar etter ca. 1/2 år. På seminaret diskuterte vi organisatoriske konsekvenser av edb-systemer. Vi laget også systembeskrivelser sammen med sykepleierne, som vi etterpå diskuterte i plenum. Deretter laget vi to enkle prototyper, en edb-basert kardex og edb-basert prosedyrebok. Dette er beskrevet i "Gjensidig læring - rapport fra fase 1 i Florence-prosjektet".

På SiA brukte vi mye mindre tid på å observere hva som foregikk på posten. En grunn er nok reisetida, men en annen grunn er at vi hadde lært så mye om sykepleieres arbeid på Voksentoppen at vi visste hva vi ikke behøvde å kunne noe om. Vi var på ekskursjoner med sykepleierne fra SiA også, og vi arrangerte et 2-dagers seminar etter 3 måneder på avdelingen. Innholdsmessig var det ganske likt det første seminaret. Men siden vi kunne mer om sykehus, hadde vi bedre eksempler; vi hadde til og med laget oppgaver det gikk an å løse. Og vi hadde med oss sykepleie-baserte opplæringssystemer laget av hovedfagsstudenter ved Ifl. Vi har hatt mindre teoretisk opplæring på SiA. Opplæringen har ofte vært nokså teknisk, og den har vært gitt når noen har følt behov for undervisning. Vi har bare laget ett system på SiA, men det har vi til gjengjeld implementert. Aktivitetene på SiA fram til beslutningen om hvilket edb-system vi skulle lage, er dokumentert i "Læring som forutsetning for design - rapport nr. 2 fra Florence-prosjektet".

Vi kan si at aktivitetene på SiA i det store og hele har vært mer teknisk orientert enn aktivitetene på Voksentoppen. Vi brukte kortere tid på å bli kjent og å forstå hva sykepleierne gjorde på SiA, men vi gjorde mye av det samme som på Voksentoppen. Begge steder markerte 2-dagers seminaret et vendepunkt i prosessen, som en start på design-aktivitetene. Det er de tekniske sidene ved systemutviklingen som skiller de to periodene mest: begge ganger laget vi prototyper, men bare på SiA implementerte vi et edb-system beregnet på daglig bruk.

Fordi edb-systemet skulle kunne tas i bruk måtte vi kontakte Datatilsynet for å få konsesjon for å registrere personopplysninger i systemet. Både pasient-opplysninger og opplysninger om personalet ville være nødvendige. Vår første konsesjon gjaldt Florence-prosjektets personregister for året 1986, og vi fikk tillatelse som forskningsprosjekt. Spesifikasjonene var derfor nokså løse, og knyttet til det edb-systemet som skulle utvikles i prosjektet. Vi fikk senere forlenget konsesjonen til 1/7 1987 - vi hadde jo ikke rukket å lage noe register da konsesjonen utløp 31/12 1986!

2. OM YRKESORIENTERTE SPRÅK.

En grunnleggende idé i Florence-prosjektet var idéen om *yrkesorienterte språk*.² Motivasjonen for idéen var at fagforeninger var de eneste organisasjonene som var sterke nok til å møte utfordringene fra flernasjonale maskinleverandører. Kampen for alternative edb-systemer, og sikring av arbeidstakernes interesser, måtte skje gjennom fagforeninger. En slik yrkesorientering skulle avspeiles i begreper og språk.

Språkidéen omfattet mange "nivåer": Yrkesorienterte programmeringsspråk, så yrkesutøverne kunne endre programmene selv, yrkesorienterte systembeskrivelsesspråk som ga muligheter for å beskrive arbeid i termer som var kjent for yrkesutøverne, og til sist: utvidelse av yrkesspråkene så de omfattet generelle begreper for informatikk, ikke bare leverandør-avhengige begreper.

Det viste seg at det var vanskelig å gjennomføre dette i praksis. Etter en stund viste det seg også at ikke alle i prosjektgruppa syntes idéen om yrkesorienterte språk var så god. Vi var enige om at en yrkesorientering kunne være en bra måte å avgrense et edb-systems

²Denne idéen er også bærende for SYDPOL-programmet, se SYDPOL (1983).

bruksområde på, men å lage et språk var flere av oss skeptiske til. Lingvister³ har vist at brukere av edb-systemer bruker ord fra skjermbilder og rapporter i sin dagligtale, og at det derfor er en idé å hente slike ord fra bruksområdet. Men et helt språk?



Argumentasjonen om de forskjellige nivåene av språk var omtrent følgende:

Yrkesorienterte programmeringsspråk.

Det er mulig at det går an å lage noe som likner på et yrkesorientert programmeringsspråk. Men vil et slikt språk ha kvaliteter som er vesentlig forskjellig fra de programmeringsspråkene vi kjenner, og hva skal i så fall de være? Og hvis forskjellen ligger mer i hvilke benevnelser og begreper som brukes, betyr det at det fortsatt vil kreves programmeringskunnskaper for å endre og vedlikeholde programmene. Det går an å lage yrkesorienterte edb-systemer, men det er valget av hvilke edb-systemer vi lager, og ikke programmeringsspråkene, som gjør et edb-system yrkesorientert.

Yrkesorienterte systembeskrivelses-språk.

Hensikten med et yrkesorientert systembeskrivelses-språk må være å lette kommunikasjonen mellom systemutviklere og yrkesutøvere. Problemet er at begge parter blir "lurt". Yrkesutøvere blir tvunget til å beskrive jobben sin på informatikkens (informatikeres) premisser - uten informatikere hadde det jo ikke vært behov for et slikt systembeskrivelsesspråk. Og informatikere vil tro at de skjønner hva yrket handler om. Våre erfaringer er at det som er viktig i et yrke, blir tatt som en selvfølge. Det er ingen som snakker om det. Følgelig brukte vi minst ett år på å finne ut hva sykepleiere synes er viktig. Det andre poenget er at systembeskrivelser beskriver konkrete fenomener, og de vil bli preget både av arbeidsorganiseringen og hva som er aktuelle arbeidsoppgaver på arbeidsplassen. Det vil derfor være vanskelig å lage et generelt beskrivelsesspråk for en yrkesgruppe som har så mange forskjellige arbeidssituasjoner som sykepleiere.

Utvidelse av yrkes-språket.

Dette er vel nærmest en måte å si at "yrkesutøvere må få en generell edb-utdanning" på. Yrkesutøvere trenger generelle edb-kunnskaper for å påvirke hvordan edb brukes i deres yrke, dvs. de trenger kunnskap om edb som er uavhengig av hvilken maskinleverandør bedriften velger å bruke. Datamaskinleverandører gir oftest utdanning i å bruke deres maskiner på korte, funksjonsrettede kurs. Men selv om yrkesutøvere får generelle kunnskaper om edb, og dermed mer generelt anvendelige begreper som kan uttrykke denne kunnskapen, er ikke det noen garanti for at folk bruker ordene og språket slik "læreboka" sier⁴. Tvert imot tar de ofte etter våre (u)vaner, f.eks. pleier vanligvis sykepleiere, som

³ Se f.eks. Holmquist & Källgren (1981), Bøgh Andersen & Halskov Madsen (1986).

⁴ Se Kalverkämper (udat.). Igjen bidrar Holmquist & Källgren (1981) og Bøgh Andersen & Halskov Madsen (1986).

vi, å bruke det leverandør-spesifikke ordet "NOTIS" framfor det mer generelle "tekstbehandlingssystemet".

I vår jakt på "yrkesspråket" oppdaget vi også at språk i seg selv ikke er nok for å forstå det som skjer på en arbeidsplass. Det har også bidratt til at vi har fått svekket vår tro på yrkesorienterte språk.

Yrkesorienteringen.

Det kan også stilles spørsmål ved yrkesorienteringen som idé. I vårt samfunn er det stort sett menn som har yrker, og som er knyttet til fellesskap som laug og fagforeninger. Yrker er noe man lærer opp til, f.eks. ved å gå i lære. I mannsyrkene er det lange tradisjoner for laugs-virksomhet, og en yrkesorientering i systemutviklingen kan ses som et ledd i å styrke en yrkesgruppes stilling ved å bruke det etablerte fagforenings-apparatet til å kontrollere utviklingen av edb-bruk i yrket. For folk som ikke har yrker med tilhørende utdanning og organisering, vil ikke yrkesorienteringen være en mulig strategi for å vinne kontroll over bruken av teknologi i arbeidet: dvs. at strategien bare vil kunne brukes av de som har en viss grad av kontroll fra før. Det vil nok ikke overraske mange at dette utelukker en lang rekke kvinne-yrker, dvs. jobber som tradisjonelt utføres av kvinner. Kvinner befinner seg ofte innenfor tjenesteytende næringer, f.eks. som sekretærer, i varehandel og service-arbeid, og dette er liksom ikke "yrker" på samme måte som snekkere og leger er. Kvinner i tradisjonelt kvinnearbeid er vanligvis ufaglærte og uorganiserte, og har ofte forskjellige deltidsordninger⁵. En yrkesorientering vil ikke kunne brukes for å bidra til at flertallet av yrkeskvinner får mer kontroll over sin egen hverdag. Yrkesorienteringen kan sementere kjønnssegregeringen i arbeidsmarkedet fordi teknologi kan brukes som et middel til fortsatt å ekskludere kvinner fra de organisasjonene som utøver kontroll.⁶

Sykepleie er et unntak fra regelen i den forstand at sykepleiere har både et organisasjons-apparat knyttet til faglærte sykepleiere, og de har (delvis⁷) kontroll over utdanningen. Dette gjør strategien med yrkesorienteringen mulig å følge.

3. OM FLORENCE SOM FORSKNINGSPROSJEKT.

Som forskningsprosjekt har vi hatt noen problemer som skyldes forskning mer enn systemutvikling: et forskningsprosjekt kan være vanskelig å styre, siden målet er kunnskap og ikke på forhånd veldefinerte produkter. Vi har nok også på følelsen av at vi kunne ha fått mer for pengene enn det vi har gjort: Vi har ikke fått tilstrekkelig med penger noen steder. Det betyr at vi alltid har måttet jobbe - tildels nokså intenst - for å få det som manglet, enten det har dreid seg om datamaskiner eller lønn til sykepleiere. I noen perioder av prosjektet har det lyktes oss å ha alt vi trengte: det er i disse periodene vi har kunnet være aktive på sykehusene. Resten av tiden har ikke bare vært brukt til søknadsskriving, men også til refleksjon over egen praksis, med muligheter for å utvikle nye idéer.

Vi har også hatt en styregruppe tilknyttet prosjektet, kalt *følgegruppe* for å markere at gruppa ikke skulle styre forskningen for mye (se vedlegg 1). Det gjorde den heller ikke. Det viste seg å være vanskelig å motivere mulige interessenter til å delta i følgegruppa, og dermed til å diskutere og bidra til prosjektet når det var forskning og langsiktige mål som sto på dagsordenen. Følgegruppa fungerte atskillig bedre som interesseforum da det konkrete edb-systemet begynte å ta form. Vi tror at styregrupper der interessentene har sterkere økonomiske interesser i prosjektet, deltar og styrer mer. Ansvarsforholdene i prosjektet har vært nokså uklare, og vi tror det skyldes at prosjektet har vært et samarbeidsprosjekt finansiert fra flere hold. Det har vært enklest for styregruppa å ikke ta opp

⁵ Foged & Markussen (1984) og Balas et al(1982)

⁶ Cockburn(1983)

⁷ Se Martinsen & Wærness(1979) og Freidson (1970) s.76. Martinsen & Wærness skriver bl.a. "Med loven av 1948 hadde sykepleierne fått det formelle ansvaret for sin egen utdanning....Som tidligere påvist var det viktig for sykepleien å legitimere seg gjennom den mektige medisiner-profesjonen"(s 80).

en reell ansvarsdiskusjon. I tillegg til å passe på at følgegruppa møttes, måtte vi derfor styre oss selv og ordne opp i de interessekonfliktene som oppsto underveis.

Det kan være en motsetning mellom det å lage et edb-system som er forskningsmessig interessant, og et edb-system som kan tas i bruk raskt.

I vårt prosjekt har vi opplevd en konflikt mellom forskerne og "brukerne". På Voksentoppen var det konflikter mellom hva vi og sykepleierne oppfattet som interessante edb-anvendelser, og det lyktes oss ikke å komme til noe godt kompromiss. På SYD 5 kom vi fram til et kompromiss, så konfliktene gjaldt her mer utviklingen av systemet enn systemet i seg selv. Sykepleierne på SYD 5 hadde foreslått et edb-system som de mente ville være enkelt å bruke. Derfor trodde de at systemet ville være enkelt å lage, og at det dermed kunne bli innført om kun kort tid. Det ble det ikke. Selv om et edb-system virker enkelt fra sykepleiernes synspunkt, er det ikke nødvendigvis enkelt fra et informatisk synspunkt. I tillegg til at prosessen i seg selv gikk sent, hadde vi andre forsinkelser. Dette skapte store frustrasjoner hos sykepleierne. De mente vi burde ha avblåst hele idéen, og funnet på noe å lage som tok kortere tid. Det kunne vi sikkert også ha gjort. Men etter vår mening ville det lett ha ført til at vi tilpasset sykepleiernes behov til eksisterende edb-teknologi, i motsetning til å tilpasse teknologien til sykepleiernes behov. Og det var jo det siste som var vårt forskningsmessige mål. Som forskere er tid en ressurs vi kan tilby.

Det viste seg senere at sykepleierne ikke skjønnte hvorfor vi syntes idéen var god, og at de følte seg misfornøyd med sin egen innsats. De betraktet nemlig idéen som noe helt selv-følgelig, nærmest som den naturligste edb-anvendelse i verden. Men som informatikere er vi oppdratt til å tenke på informasjon i helt andre baner enn det sykepleiere gjør, og vi syntes deres idé var fundamentalt forskjellig fra noe annet vi hadde sett. Så vi syntes det var faglig spennende å implementere deres idé, og vi forsøkte å være så lojale mot idéen som det var teknisk mulig. Det gjorde at systemutviklingen tok lengre tid.

Nå ble vi ikke klar over hvor skarp denne motsetningen var før edb-systemet var så godt som ferdig. Og da ville det jo vært dumt å begynne på noe nytt. Personlig er vi glade for at vi slapp å ta konflikten. Det er mulig at vi hadde gitt etter, og tilpasset sykepleiernes behov til teknologien, siden det er begrenset hvor mye motgang vi kan tåle. På den annen side: Hvis vi hadde forklart sykepleierne grundig hvorfor idéen var god fra vårt synspunkt, kunne vi ha fått en dose moralsk støtte.

I et forskningsprosjekt vil forventningene og prioriteringene hos de forskjellige partene være gjenstand for både uenighet og forhandlinger. Dette skyldes dels at forskning har ny kunnskap som mål, mens "brukerne" ønsker seg et ferdig system som virker. I tillegg må vi nevne at forskning er ett av de få områdene hvor det er legitimt for informatikere å ha egne interesser, og et område hvor vi kan insistere på at våre interesser blir tatt alvorlig.

DEL II. OM SYSTEMUTVIKLING MED BRUKERE.

Denne delen vil i hovedsak dreie seg om systemutvikling med brukere, dvs. vårt (informatikernes) samarbeid med helsevesenet. Det dreier seg for det meste om våre erfaringer fra samarbeidet med sykepleierne i Florence-gruppa på Sentralsykehuset i Akershus (SiA), Kardiologisk seksjon: SYD 5.

Vi har delt delen i tre kapitler. Det første kapitlet handler om våre holdninger til systemutvikling, kalt bruksperspektivet, og hvilke erfaringer vi har med å anvende dette perspektivet. Kapittel 2 handler om erfaringene fra samarbeidet med sykepleierne det siste året. Erfaringene er til tider sammenliknet med de erfaringene vi har fra Voksentoppen. Det tredje og siste kapitlet handler om sykepleiernes erfaringer fra samarbeidet.

1. BRUKSPERSPEKTIVET I PRAKSIS.

Som informatikere har vi hatt holdningsmessig basis i *bruksperspektivet*¹. Dette synet på systemutvikling hevder at design av edb-systemer må ta utgangspunkt i folks arbeid. Edb-systemet må lages slik at det tas hensyn til både produktive og reproduktive sider av arbeidet (s.209-210). Vi antar at folk oppfatter sitt eget arbeid som en helhet, og at edb-systemet må passe inn i denne helheten. Vi går dessuten ut fra at både faglige og sosiale aspekter er viktige for arbeidsforholdene (s.216). Bruksperspektivet har også et *lokalt* utgangspunkt: det baserer seg på at det fins en felles sosial oppfatning om arbeidet på en arbeidsplass, m.a.o. at det går an å fokusere på enigheter, selv om det fins motsetninger. Vi avgrenser oss dermed til en relativt liten gruppe personer som samarbeider på samme sted. Personene i gruppa må ikke nødvendigvis ha samme yrke, f.eks. kan bruksperspektivet anvendes for flerfaglige team.

Dette synet på arbeid omfatter selvsagt både informatikeres og sykepleieres arbeid. En av konsekvensene er at systemutvikling må betraktes som et samarbeid mellom ulike ekspertgrupper, der alle er eksperter på sitt eget arbeid.

Bruksperspektivet antar også at edb-folkene kommer utenfra (s.208), enten det er fra en annen avdeling i organisasjonen eller fra en helt annen organisasjon. Vi vet godt at systemutvikling er en aktivitet som virker konfliktutløsende i organisasjoner, og vi vet også at vi som informatikere griper inn i prosesser og har mulighet for å påvirke dem². Men selv om vi kan påvirke situasjoner, kan ikke vi løse konflikter som måtte oppstå mellom forskjellige grupper.

Bruksperspektivet oppsto som idé da vi forberedte Florence-prosjektet: vi hadde behov for å klargjøre hvilket verdigrunnlag prosjektet skulle hvile på. I prosjektet har vi så valgt arbeidsformer som harmonerer med disse verdiene, og gjennom det praktiske arbeidet har vi fått nyansert idéen noe. Lyytinen karakteriserer bruksperspektivet som etno-metodologisk³, og vi synes karakteristikken passer både idé- og metode-messig. Den antropologiske tilnærmingen skyldes nok blant annet at vi har samarbeidet med en sosialantropolog store deler av prosjektperioden.

I Florence-prosjektet har vi fått anledning til å prøve ut bruksperspektivet i praksis. I det følgende skal vi beskrive hva har vi lært om anvendeligheten av dette prinsippet. Vi synes fremdeles bruksperspektivet gir en sunn basis for samarbeid i systemutviklingsprosessen.

¹ Bjerknes & Bratteteig (1984)

² Se Borum & Enderud (1981) og Lanzara & Mathiassen (1984)

³ Se Lyytinen (1986) s. 280-282

Det er ikke så lett virkelig å betrakte andre som eksperter på sitt eget arbeid.

Når det gjelder analyse av arbeid, er det forholdsvis enkelt å si at alle er eksperter på sitt eget arbeid. Men vi har brukt lang tid, og vi måtte lære mye om sykepleie, før vi turde å la dem ta avgjørelsen om hva slags edb-system de ønsket å bruke, dvs. hva slags edb-system vi skulle designe. Problemet vårt var delvis forskningsmessig: vi insisterte på at systemet skulle være for sykepleieres daglige arbeid. Vi hadde ikke lyst til å lage f.eks. et lite lagersystem for medisiner. Med andre ord: vi ville ikke lage et "tradisjonelt" edb-system, hvor dataene vi puttet inn tilfeldigvis ble brukt av sykepleiere, men som i prinsippet kunne vært brukt av pølseverkstedsmedarbeidere, møbelforretninger eller hva som helst. Det vi måtte ha erfaring for å se, var at det relativt enkle forslaget vi endte opp med, faktisk handlet om noe av kjernen i sykepleie. Med mindre erfaring hadde vi lett avvist systemet som uinteressant fra et forskningsmessig synspunkt. Med andre ord: vi kunne ha brukt dette forskningsargumentet til å si at systemet ikke dreier seg om sykepleieres daglige arbeid, og så hadde prosjektet handlet om hva vi mente sykepleieres daglige arbeid var. Dette er forøvrig alltid ett dilemma: selv om personer er eksperter på sitt eget arbeid, hender det at andre har velkvalifiserte spørsmål å stille til de begrunnelser som blir gitt. Disse spørsmålene kan ofte være grunnlaget for fruktbare konfrontasjoner mellom gruppene. Nå er det ikke bare informatikere som kan ha problemer med å anerkjenne andres ekspertise. Vi vil minne om at siden informatikere er eksperter på å lage edb-systemer, vil de kunne ha kvalifiserte meninger om design, uavhengig av bruksområdet for systemet.

Samarbeid mellom ekspertgrupper betyr ikke at alle skal gjøre det samme, eller gjøre like mye, hele tiden.

Dette virker som en nokså banal uttalelse. I praksis har det vist seg at det ikke alltid er så enkelt at noen er mer eksperter enn andre på et gitt tidspunkt.

Det har vist seg at forskjellig ekspertise har hatt forskjellig plass i prosjektforløpet. For oss var sosialantropologen en viktig person i analysedelen av prosjektet fordi han så andre ting enn oss, i tillegg til at han ofte kunne komme med alternative forklaringer eller andre måter å se tingene på. Selvsagt var sykepleierne og deres arbeid også viktig. I denne perioden var det viktig med *gjensidig læring*, siden vi tross alt skulle ende opp med et designforslag. Det vil si at informatikerne hadde ansvar for å legge opp prosjektet slik at de lærte noe om sykepleie, og at sykepleierne lærte noe om edb. Mot slutten av læringsperioden hadde informatikerne og sykepleierne møter hver for seg for å diskutere og prioritere forslag til edb-system for sykepleierne (beslutningsprosessen er beskrevet i "Læring som forutsetning for design. Rapport nr. 2 fra Florence-prosjektet"). Etter "fraksjons-møtene" møttes vi for å forhandle oss fram til et forslag vi alle kunne være enige om. Begge gruppene stilte med to forslag. Sykepleierne ønsket seg et Arbeidsark-system eller et pleie-tyngde-registrerings-system. Og vi ønsket oss et Arbeidsark-system eller et system som behandlet "harde facts". Vi ble enige om Arbeidsark-systemet, som i utgangspunktet var bygget på en idé fra sykepleierne.

Hva hendte så? Informatikerne overtok idéen. Dette førte til at sykepleierne følte seg ekskludert. De mistet eiendomsforholdet til idéen, og det skjedde ingen ting, i hverfall ikke noe de kunne se. For oss var situasjonen annerledes. Vi følte at vi var i en fase av prosjektet hvor vi hadde ansvaret. Når vi oppsøkte andre personer, var det personer med teknisk kompetanse, ikke sykepleierne, siden vi mente at vi hadde en god nok forståelse av det produktet vi skulle lage. Vi hadde ikke den samme oversikten over det utstyret vi skulle bruke. Selvsagt skjedde det ting, men de var ikke synlige for sykepleierne. Og et program som nesten virker er ikke egnet for å vise at det er framdrift i prosjektet. Dessuten syntes vi det var vanskelig å arrangere møter når vi ikke hadde noe særlig å si, spesielt fordi sykepleierne alltid klaget over at de hadde så mye å gjøre.

Dette førte raskt til en motivasjonskrise på sykehuset. Motivasjonskrisa på avdelingen ble ytterligere forsterket og vedlikeholdt ved at Florence-prosjektet, og den tilsynelatende manglende framdriften i prosjektet, ble brukt i interne konflikter på avdelingen. Vår rolle i arbeidet med å avverge og redusere krisa ble vanskeliggjort av dette. I ettertid tror vi at vi kunne bedret motivasjonen i prosjektets sykepleiergruppe hvis vi på et tidligere tidspunkt hadde gjort klart at dette var en periode av prosjektet hvor informatikerne jobbet mest og

hadde hovedansvar, og at ansvaret ville forskyve seg drastisk når tiden nærmet seg for innføring og bruk av systemet.

Innvinger mot bruksperspektivet.

Bruksperspektivet sier noe om hvilke verdier vi legger vekt på under systemutviklingen. Det har vist seg å fungere bra i prosjektet, men selvsagt har også dette perspektivet noen begrensninger. Vi vil trekke fram to kjennetegn her: perspektivet er informatisk, og det er et en-parts-perspektiv. La oss se litt nærmere på hva dette innebærer.

Med utgangspunkt i bruken av datamaskiner, tar vi utgangspunkt i de arbeidsoppgavene hvor datamaskiner skal brukes. Siden vi fokuserer på et helhetssyn på arbeidet, betyr det at vi også må se på samspillet mellom de forskjellige arbeidsoppgavene. Det er likevel viktig å forstå at datamaskinen på sett og vis fortsatt er i sentrum, men at vi ikke må glemme at den skal brukes av noen. Det er en viktig forskjell fra dette synet til et *brukerperspektiv*, som etter vår mening hevder at "brukerne er ekspertene". Med et brukerperspektiv blir ekspert-rollen forskjøvet fra informatikere til brukerne. For å lage en god brukssituasjon og for å lage nye og spennende edb-anvendelser, tror vi at det er viktig med både edb-kompetanse og, i vårt tilfelle, sykepleierkompetanse. Hvis ekspertisen sitter hos en enkelt yrkesgruppe, blir det vanskelig å få i stand konfrontasjoner med ekspertens grenser og dermed muligheter for grensesprengende nytenkning. Teknologien vil da lett kunne bidra til en konservering av den bestående arbeidsstrukturen.

Bruksperspektivet er et en-parts-perspektiv. På den måten vil edb-systemer laget med utgangspunkt i disse holdningene alltid kunne brukes for å forsterke maktposisjonen til dem vi lager edb-systemer for. Vi har med andre ord valgt å se bort fra konflikter mellom grupper i organisasjoner, for ikke å snakke om kundene og klientene til organisasjonen. I vårt tilfelle vil et edb-system kunne forandre sykepleiernes forhold til både hjelpepleiere og leger. Vi har valgt å si at det er organisasjonen selv som må ordne opp i forholdet mellom forskjellige interessegrupper. Informatikere har ansvar for å peke på problemene, men ikke for å løse dem.

Bruksperspektivet tar også utgangspunkt i én arbeidsplass. Det betyr at felles-systemproblematikken⁴ kan være vanskelig å diskutere innenfor rammene av bruksperspektivet. Vi har for eksempel ikke tatt stilling til hvordan alle sykepleiere, og mangfoldet av deres arbeidsplasser, kan være representert i systemutviklingsprosessen.

I Florence-prosjektet har vi fra vår side drevet informatisk en-parts-forskning, og laget edb-systemer som skal brukes lokalt på én sykehuspost. Vi har derfor ikke vært hemmet av begrensningene i bruksperspektivet.

2. SYSTEMUTVIKLINGSPROSESSEN FRA INFORMATIKERNES SYNSPUNKT.

I dette kapitlet skal vi presentere informatikerenes erfaringer fra Florence-prosjektet, i hovedsak erfaringer fra implementeringen av Arbeidsark-systemet. Erfaringene fra lærings- og analyse-delen av prosjektet er dokumentert i tidligere rapporter fra prosjektet. I tillegg sammenlikner vi noe av det vi har opplevd på SYD 5 med våre opplevelser på Voksentoppen.

Erfaringene presenteres i to hovedgrupper: først diskuterer vi arbeidsformene i prosjektet og de konsekvenser valg av arbeidsform har fått for styring av prosjektet, opplæring og spredning av informasjon. Deretter diskuterer vi læringsaspektet ved prototyping.

⁴Problemer knyttet til det å ta i bruk "samme" system i flere avdelinger i en stor organisasjon er bl.a. behandlet i SYDPOL (1987) s. 17.

2.1 Arbeidsformer

Vi synes at arbeidsformen i prosjektet har fungert bra.

Prosjektet har det siste 1 1/2 året vært organisert som et samarbeid mellom en gruppe med informatikere og sosialantropologer fra Universitetet, og en gruppe med sykepleiere fra SYD 5. Avdelingen har i prosjektet fått tilført lønn til en sykepleierstilling, og den ekstra arbeidskraften er blitt fordelt blant en gruppe ansatte på avdelingen, slik at flere kan drive med prosjektet i arbeidstiden.

Denne måten å organisere arbeidet på er forskjellig fra det vi gjorde i samarbeidet med Voksentoppen. Da var en sykepleier lønnet av prosjektet og trukket vekk fra avdelingen. De ansatte på avdelingen deltok delvis i sin fritid og delvis i arbeidstida. Sykepleieren i prosjektet deltok i arbeid på avdelingen, og dermed ble hennes innsats på avdelingen "byttet mot" innsats fra de ansatte på avdelingen. Men siden de hadde mye å gjøre på den avdelingen også, var det vanskelig for oss å kreve at sykepleierne skulle stille opp for fullt, og vi er imponert over hvor mye de stilte opp på fritida.

For oss har det vært en fordel at vi har hatt samarbeid med en gruppe på avdelingen, framfor at en person blir trukket ut og vekk fra avdelingen. Det er da mulig for flere sykepleiere å bruke tid på prosjektet. Det vil med andre ord si at det er helt legitimt å ha en "sykepleier-fraksjonsgruppe". På den måten stiller forskerne og sykepleierne likt: vi kan begge diskutere idéer med våre egne. I tillegg blir det lettere å ha et likeverdig samarbeid. Det er lettere å fordele både rettigheter og plikter når vi vet at alle deltakerne har anledning til å stille opp innenfor rammene av sitt daglige arbeid.

Det er rapportert at de "profesjonelle" brukerne blir mindre og mindre representant for de de er plukket ut for å representere, hvis de blir trukket ut av sitt daglige arbeid over lengre tid⁵. Dette har ikke vært noe problem med vår organisasjonsform. Tvert imot viser det seg at sykepleierne legger stor vekt på sykepleierne interesser og behov når de møtes til diskusjoner utenom det daglige arbeidet. Alle sykepleierne vi har samarbeidet med konkluderer med at det er sykepleie som er viktigst å kunne for sykepleiere som skal delta i systemutvikling.

Det er også ulemper ved å velge en arbeidsform som baserer seg på et samarbeid mellom grupper. Det er mye tyngre å motivere en gruppe enn å motivere én person. Og når gruppa kommer i krise, kan det være nokså tungt å komme over kneika. Men vi mener det er vel verdt å investere energi i å motivere en gruppe, fordi den er selv bærende når arbeidet kommer i gang. En gruppe på arbeidsplassen vil også ha bedre muligheter for å initiere en forandringsprosess enn det én person har. Dermed vil en gruppe som fungerer, kunne sørge for at prosessen fortsetter selv om vi som utenforstående trekker oss ut av prosjektet⁶.

Representasjon.

I prosjektet har vi hatt diskusjoner om arbeidsorganiseringen, og spesielt om forholdet mellom de to yrkesgruppene som har vært representert i prosjektet det siste året. Bakgrunnen for diskusjonene er forskjellen i forventninger, og dermed også forventninger til styringen av prosjektet. Sykepleierne syntes jo det gikk tregt, og mente at vi skulle ha avblåst utviklingen av det foreslåtte systemet når ikke resultatene kom på det forventete tidspunktet. Spørsmålet ble: kunne en mer tverrfaglig person bidratt til en raskere avskjæring, dvs. mer styring? Etter litt fram og tilbake ble vi enige om at det nok finnes personer med dobbel-kompetanse, men at de neppe blir tverrfaglige av den grunn. Uansett ville en slik "mekler" vært enten sykepleier eller informatiker "i sitt hjerte". Vi tror problemene kunne løses langt på vei med et tettere samarbeid mellom gruppene, ved at vi hadde brukt mer tid underveis på å finne de forskjellige forventningene i prosjektet. Vi må forsøke å

⁵Josefson gir et glimrende eksempel på dette i sin rapport fra et engelsk sykehus (Josefson (1987)). Se også Håndlykken (1982).

⁶Ved å overlate ansvaret for prosessen til en gruppe på arbeidsplassen, nærmer vi oss det Keen kaller en "Up-and In"-strategi for forandring, se Keen (1981).

velge en arbeidsform som tillater problemene å dukke opp. I forhold til det mer generelle diskusjonstemaet bruker-representasjon, blir konklusjonen altså at vi ikke trenger "profesjonelle brukere", men arbeidsformer som tillater løpende konfliktløsning.

Opplæring i bruk av edb-systemet.

Samarbeidet med en gruppe på avdelingen har vist seg å være et godt utgangspunkt for spredning av informasjon og for opplæring etter knoppskytingsprinsippet.

Bakgrunnen for opplæring etter knoppskytingsprinsippet er at innføringen av edb-systemet foregår på en "D-dag", og da må alle kunne bruke systemet. Årsaken til dette er systemets bruksområde: sykepleieres daglige arbeid. Dessuten må opplærings-opplegget være selvstående: folk må lære når de har tid, ikke når en eller annen tilfeldig "ekspert" er tilstede.

Opplæring tar utgangspunkt i den eksisterende organiseringen av arbeid. Sykepleierne er organisert i smågrupper på 3, som går vakter sammen, og på vakt er de delt opp i et inne-team og et uteteam. Teamene relaterer seg til ansvar for en "fysisk" del av avdelingen, og de pasientene som ligger der.

"Kjernegruppa" i Florence-prosjektet lærer opp 1 person i hver av smågruppene, som igjen lærer bort til de andre i smågruppa. Dette opplegget ser realistisk ut når det gjelder opplæring ved innføring av Arbeidsark-systemet. Opplæring og forutsetningene for opplæring er nærmere beskrevet fra sykepleiernes side i kapittel 3 i denne delen.

Et selvstående opplæringsopplegg vil også lett kunne brukes av nyansatte. Alternativet til ett opplæringsopplegg for alle, er å lage to opplæringsopplegg; ett for innføringen og ett som kan brukes av nyansatte. For eksempel kunne opplegget for innføring av systemet bestå i at de ansatte blir fjernet fysisk fra jobben for å lære og bruke systemet. Dette fører til at opplæringen vil ta lengre tid - og at de ressursene som faktisk finnes, ikke benyttes på noen god måte.

Spredning av kunnskap.

Arbeidsformen på SYD 5 har vært god, når vi betrakter den som en form for prosjekt-organisering. Strukturen gir muligheter for opplæring og arbeid lokalt på posten. Men arbeidsformen er ikke så god når det gjelder spredning av resultater i en litt større sammenheng. Vår holdning er at sykepleiere må spre kunnskap om edb i sykepleie blant sykepleiere, fordi de (selvsagt) kan formidle stoffet på en bedre måte til sykepleiere enn det vi kan. Men det kan være vanskelig å få tid til det innenfor rammene av det daglige arbeidet, spesielt når det pågår utviklingsarbeid på posten samtidig. På dette punktet hadde arbeidsorganiseringen fra Voksentoppen en klar fordel: en person som er trukket ut fra arbeidet for å representere sykepleiere i et prosjekt, prioriterer ofte formidling til andre høyt. Det gjorde i hvertfall vår sykepleier. Resultatene av hennes aktiviteter er at mange i Norge har fått vite hva som er gjort både i Norge og resten av verden om edb og sykepleie, samtidig som resten av verden har hørt om hva som skjer i Norge. Et konkret resultat er en arbeidsgruppe i SYDPOL-programmet som spesielt tar for seg edb og helsevesen.

2.2 Læringsaspekter ved prototyping.

Prototyping som bakgrunn for sykepleiernes valg av edb-system.

En del av sykepleiernes første opplæring bestod i å prøve og bruke prototyper laget på Apple's Macintosh⁷. Prototypene var laget på bakgrunn av erfaringene fra første del av prosjektet, som foregikk på Voksentoppen. Det var tre prototyper som viste forskjellige måter å presentere samme informasjon på. En var hovedsakelig basert på tekst og var ganske tradisjonell. En annen prototype kombinerte tekst og grafisk framstilling, og brukte kurver og diagrammer sammen med tekst. Den tredje prototypen hadde organisert informasjonen som et kart over posten, og brukte enda mer symboler og bilder i sin pre-

⁷Beskrevet detaljert i "Læring som forutsetning for design. Rapport nr. 2 fra Florence-prosjektet", kap 5.

sentasjon. Programmene brukte Macintoshens muligheter til å framstille grafikk, og hensikten med prototypene var å demonstrere forskjellige måter å presentere informasjon på. Med annet maskinutstyr ville vi hatt helt andre (og færre) muligheter for å lage prototyper. Fordi informasjonen som ble presentert var sykepleie-faglig, og dermed kjent for sykepleierne, kunne sykepleierne konsentrere seg om å vurdere presentasjonsformene framfor innholdet / meningen i informasjonen.

Det viste seg da sykepleierne foreslo edb-system, at prototypene hadde påvirket deres forestillinger om hva edb kan brukes til, og hvordan informasjon kan presenteres. De foreslo nemlig at informasjonen skulle presenteres som et kart over posten. Vi er nokså sikre på at de ikke hadde foreslått denne løsningen hvis de bare hadde hatt mulighet til å prøve edb-systemer med vanlige linje-orienterte skjermer, og følgelig linje-orienterte demonstrasjonssystemer.

Vi kan trekke flere konklusjoner av dette. For det første er vår erfaring at prototyper er utmerkete læringsmidler - for ikke-informatikere. Forskjellige eksempler og demonstrasjoner setter i gang fantasien. Samtidig begrenser de fantasien.⁸ Hvis eksemplene er hentet fra én type anvendelser, skal det mye oversikt til for å komme på helt andre løsninger for dette anvendelsesområdet.

Prototyping var ikke godt egnet for å lage kravspesifikasjon til edb-systemet.

Vi lagde også en prototype i design-fasen av prosjektet, og da i hovedsak for å prøve teknikken prototyping. Vi trodde prototypingen kunne hjelpe oss med å lage en fastere kravspesifikasjon til Arbeidsark-systemet. Så vi laget skjermbilder og menyer ved hjelp av skjermhåndterings-verktøyet FOCUS og programmeringsspråket COBOL. Vi brukte noen uker på det, siden vi var ukjente både med maskina og med FOCUS. Da vi endelig var ferdige, vekket ikke skjermbildene den helt store interessen. Det var ingen som satte seg ned og "lekte" med systemet. Men prototypen var OK for å vise de sykepleierne som ikke var med i prosjektgruppa hvordan vi hadde tenkt å lage systemet, altså for å demonstrere hva vi konkret arbeidet med og for å signalisere at vi faktisk jobbet med saken. I tillegg brukte vi tekstbehandling for å lage en "prototype" av papir-rapporten. Den fikk vi faktisk mer respons på. En av sykepleierne hadde skrevet inn alle pasientene for en vakt, og delte ut Arbeidsarket på rapporten. Det førte til diskusjoner om hva som burde stå på arket, og resulterte i endringer i layouten. En av sykepleierne laget så en ny, endret versjon av rapporten. Men heller ikke denne "papir-prototypen" ble brukt i særlig grad for å prøve ut mulighetene.

Det har slått oss at akkurat det samme skjedde på Voksentoppen. Sykepleierne brukte edb-basert kardex når vi demonstrerte for dem, men det ble lite brukt av sykepleierne på egen hånd. Den gangen konkluderte vi med at sykepleierne ikke brukte systemet fordi de ikke trodde på idéen. Det er sikkert også en grunn, men vi tror ikke det er den eneste. Sykepleierne på Voksentoppen brukte nemlig samtidig Prosedyrebok-systemet, som var en enkel variant av tekstbehandling. Vår nåværende hypotese er at sykepleierne ikke brukte edb-basert kardex fordi det var en prototype, dvs. det var ikke "brukbart". Prosedyrebok-systemet var enkelt å bruke, og i tillegg var det nyttig: prosedyrene ble skrevet og endret ved hjelp av tekstbehandling. Selv om Prosedyrebok-systemet ikke virket som totalsystem, virket utskriftsdelen, så systemet var nyttig, og det var lett å se nytten av det med en gang. Og flere gamle prosedyrer ble erstattet.

Vi har videre erfart at prototypingen, prosessen altså, er lærerik for oss informatikere, men ikke for sykepleierne⁹. Gjennom prototypingen lærer vi noe om hvilke teknologiske grenser vi må forholde oss til i en senere implementasjon. Vi lærer også av at de forslagene vi kommer med forkastes - særlig av begrunnelsene for at de må forkastes. Sykepleierne lærer selvsagt ikke så mye av dette, men kan ha utbytte av å se idéer manifestert i en datamaskin.

⁸ Se bl.a. Floyd(1983)

⁹ Se SYDPOL (1987) ss. 62-63

Ut fra disse erfaringene konkluderer vi med at prototyping er brukbart til demonstrasjon, idégenerering og diskusjon, men lite egnet som utgangspunkt for kravspesifikasjon. Dette er en konklusjon som er i motstrid med de rådende oppfatningene av hva prototyping er brukbart til¹⁰. Vi kan tenke oss tre mulige forklaringer på at prototyping ikke er velegnet for sykepleier-systemer:

- 1) arbeidsforhold på sykehuset: sykepleierne jobber ofte under tidspress, og det er vanskelig å stikke av fra de arbeidsoppgavene som må gjøres,
- 2) sykepleiere er kvinner, og kvinner leker ikke så lett med teknologien, men lærer seg å bruke den når de ser behov for det¹¹, eller
- 3) det er vanskelig å få en ordentlig utprøving av et edb-system som skal støtte samarbeid siden det er vanskelig å lage eksperimenter og laboratorieforsøk av slike arbeidssituasjoner - det er ikke så lett å late som en samarbeider.

Vi tror at alle disse faktorene har betydning for det vi har opplevd. Dette kan antakeligvis overføres til andre situasjoner: prototyping som laboratorieforsøk er kanskje ikke velegnet for kvinnelige brukere, og kanskje heller ikke for å teste ut situasjoner hvor edb-systemer skal støtte samarbeid.

Kanskje vi også kan konkludere med at den "bruker-vennlige teknikken" prototyping er oppskrytt, og at vi stiller oss tvilende til begrunnelsene for å bruke teknikken. Prototyping viser seg å være lærerikt for informatikere, men vi skal ikke overdrive utbyttet brukerne kan få av å delta i prototyping. På den annen side skal vi heller ikke undervurdere signaleffekten prototyping har: med prototyping signaliserer vi for oss selv og omverdenen at vi er interesserte i hva brukerne mener om det produktet vi lager. Og prototypen blir et symbol på prosjektets framdrift.

3. SYSTEMUTVIKLINGSPROSESSEN FRA SYKEPLEIERNES SYNSPUNKT.

I dette kapitlet lar vi sykepleierne slippe til med sine tanker om arbeidet i prosjektet. Det følgende handler derfor om den tiden prosjektet har foregått på SYD 5, SiA.

Læring.

Et tilbakeblikk på halvannet år med Florence innebærer forventning, entusiasme, skuffelser, frustrasjon, håp og læring...

Forventning	- om læring, og om å spare tid på rutinearbeid, til investering i sykepleie.
Entusiasme	- ny vinkling på vårt arbeid som informasjonsbehandlere, og nysgjerrighet på ny teknologi.
Skuffelser	- maskiner som ikke virka, program som ikke kom...
Frustrasjon	- spørsmål fra resten av avdelingen / andre på sykehuset: "skal vi ikke begynne med noe snart?", "Hva får dere ut av Florence-prosjektet, da?"...
Håp	- etter oppsummeringsmøter, - kanskje blir det noe likevel...
Læring	- som ble helt anderledes enn vi hadde forventet.

Om organisasjonen.

Arbeidsgruppa.

Posten fikk i løpet av prosjektiden en ekstra sykepleier i engasjementstilling. Denne ble ikke brukt spesielt på prosjektet, men skulle dekke opp for noe av den tid

¹⁰ For eksempel hevder Lantz at prototyping er brukbart til det meste, spesielt til å lage kravspesifikasjoner, se kapittel 1 i Lantz (1986). Bødker (1987) gir en noe annerledes drøfting av hva prototyping er brukbart til.

¹¹ Se Cockburn (1983) og (1985), og Microsyster (1986).

som arbeidsgruppa brukte til dette. Denne modellen tror vi er en god løsning på flere arbeidsplasser enn vår. Det vil alltid være en fordel å ha flere enn en person engasjert i en slik oppgave. Flere strenger å spille på, økt kreativitet, diskusjoner, gjensidig oppmuntring, (det trengs!) osv.

At tiden som går med til et slikt prosjekt må erstattes, er en selvfølge. Det er viktige, både for å muliggjøre utviklingsarbeid, og for å hindre negative ringvirkninger og dermed holdninger i resten av arbeidsmiljøet.

En arbeidsgruppe i hvert av fagmiljøene, tror vi også er å foretrekke, i forhold til å engasjere en person med kompetanse i begge fag. Det siste høres kanskje mer rasjonelt ut, men det skal godt gjøres å finne en som er *like mye* sykepleier som informatiker. Den faglige tyngden vil vippe til en av sidene og begges interesser vil ikke i like stor grad bli ivaretatt.

"Gjensidig læring"-perioden fungerte bra så lenge vi hadde regelmessige møter. Etter at ideen om Arbeidsarket var unnfanget, satte informatikerene seg til å jobbe med programmering og møtene ble sjeldnere. Vi så ikke mye til aktiviteten, og ble vel nokså utålmodige etter hvert.

I denne perioden burde vi hatt flere møter. For det første fordi vi ikke hadde innsett betydningen av idéen i informatisk sammenheng. For oss var det en idé blant mange mulige. Vi ville bare begynne å gjøre noe. Det hadde utvilsomt høynet vår motivasjon om vi hadde fått følelsen av at vi var med på et slags "pionerarbeid". Vi hadde ingen spesiell stolthet eller eierforhold til idéen, og det tror vi det er viktig å ha. Særlig hvis prøveprosjektet er av en slik art at det kan forventes å medføre en del nybrottsarbeid, som da er vanskelig å tidsberegne.

For det andre ville diskusjoner om informatikernes vanskeligheter i denne fasen kunne gitt oss nyttig kunnskap om teknologiens begrensninger, og ulikhetene produkter imellom. Kanskje hadde vi kommet nærmere å forstå hva det innebærer å bevege seg på grensene av hva et produkt kan tåle.

Vi hadde kanskje tidligere sett hvilke utfordringer som ligger i å stille krav til leverandører om utvikling på våre premisser. I utgangspunktet hadde nok vi vært mer villige enn informatikerne til å tilpasse oss de eksisterende mulighetene. Tettere diskusjoner kunne gitt oss økt forståelse, økt interessefelleskap, og større muligheter for gjensidig støtte.

Lokalmiljøet

Informasjonsmøter på posten burde ha vært holdt jevnlig. Vi så det som viktig at disse ble holdt av informatikerne. Det ble for vanskelig for oss i Florence-gruppa på avdelingen å forklare / forsvare informatikernes programmeringsvansker / utfordringer når vi i liten grad forsto dem selv. Vi skjønner godt at det ikke er lett å holde informasjonsmøter når det ikke skjer noe nytt og alt er bare problemer, men dette var virkeligheten - også for oss. Posten som helhet ble stående "utenfor" og dermed forsvant interessen.

Følgegruppa

Ansvar for gjennomføring av prøveprosjektet lå på arbeidsgruppa på SYD 5, men ansvar og oppfølging fra sykehuset sentralt fungerte uklart. Det var tydelig at ansvarsfordelingen mellom IfI og SiA ikke var klarlagt bl.a. med hensyn til hvem som skulle ta initiativ til møter, kontakt med leverandører, osv. Følgegruppemøtene var få og sjelden fulltallige. Sykepleiere er ganske uerfarne når det gjelder prosjektarbeid. Dessuten bidrar den daglige arbeidssituasjonen ofte til at alt som ikke *må* gjøres med en gang, må utsettes.

Vi følte i alle fall at oppfølgingen ble for svak.

Det er imidlertid mulig at vi ved et tettere samarbeid gruppene imellom ikke hadde følt behovet for "kontroll" så stort.

Om opplæring.

For det første erfarte vi jo at den type teknologi som blir brukt i opplæringen former fantasien fra første stund. Det blir derfor viktig å se ulike typer utstyr, og å bli gjort oppmerksom på de forskjellige mulighetene / begrensningene. Det er også et poeng at denne informasjonen kommer på et tidspunkt hvor folk er modne for å "ta" den. Språket som blir brukt i all informasjon bør referere seg til kjente begreper (i den lokale sammenhengen).

Vi visste, da vi begynte ikke forskjell på "hardware" og "software". Mange av de forklaringene vi fikk ble ikke meningsfulle før etter en god stund.

Så lenge oppgaven er å utvikle fantasien er det lov å tenke seg uendelige mengder med utstyr / ressurser. Det er også med på å forme fantasien. Det er naturligvis helt andre muligheter med en skjerm ved hver seng, enn det er med en PC på vaktrommet. Diskusjonene om ressurser i forhold til ønsker, bør komme inn i løpet av en slik prosess.

Erfaringene fra andre innen helsesektoren viser vel at det tross alt ikke er bare lurt å satse på det første og beste system som kommer i ens vei...

Planlegging av opplæring på postplan må ta i betraktning:

Spesiell arbeidssituasjon (her akuttavdeling). Dvs. vanskelig / umulig å planlegge møter / trening. Opplæring må foregå *når* det er tid. Det vil igjen si at det ikke nytter å satse på "profesjonelle" lærere som kommer til avtalte tider. Det er også umulig å sende flere på kurs samtidig.

Læringen må foregå mest mulig på arbeidsplassen, men en kombinasjon av ulike metoder er gjennomførbar. Hos oss gikk 3 av medlemmene i arbeidsgruppa på kurs. Til forskjellig tid og tildels på ulike kurs. Andre av gruppa fikk undervisning på posten (informatikerne måtte vente til vi hadde tid). Ressurskrevende, ja. Men etter vår mening er dette noe som må taes hensyn til, dersom ikke belastningen på resten av arbeidsmiljøet skal bli for stor.

Turnus er et annet viktig "hinder" som må forseres i opplæringssammenheng. Det betyr i praksis at alle informasjonsmøter må taes flere ganger. Man kan jo basere seg på referater, men det viser seg at dette ikke er på langt nær så effektivt i forhold til informasjon, engasjement og aktivitet.

Motivering

En god del tid bør sannsynligvis brukes på diskusjoner om fagutvikling, teknologisk utvikling, mål med eventuelle prosjekter. Det er viktig at flest mulig i det lokale miljøet blir delaktig i denne diskusjonen. Det å etablere et interessefellesskap og et felles "eier-forhold" til prosjektet var vel det vi lyktes minst med på SYD 5.

"Avlæring"

Mange av oss nybegynnere i EDB-sammenheng trenger en viss tid på å bli fortrolig med teknologien. Vi må bl.a. bli kvitt en del vrangforestillinger om hva en data-maskin er og kan gjøre. Til dette bruk er det bra å ha en del utstyr å prøve seg på. Vi fikk som kjent 2 Macintosh-maskiner som "øvingsutstyr"¹². Disse var utstyrt med enkle oppgaver, spill og flere muligheter til grafikk, tegning osv.

Det viste seg at spillene ble brukt, nesten uten unntak av de mannlige sykepleierne på posten, mens de kvinnelige nesten ikke rørte dem. Jentene så litt på de mulighetene som var lagt inn angående EDB-basert kardex, men sluttet fort når det viste seg at det ikke var det vi skulle bruke.

Det ble satt opp et enkelt tekstbehandlingskurs som alle på posten gikk gjennom. De som har en del skrivearbeid i forbindelse med, eller i tillegg til, jobben har hatt nytte av dette.

Vi skal ikke trekke vidtgående konklusjoner av at det så ut til at bare de mannlige sykepleierne var tiltrukket av spill i læringssammenheng. - Men det kan jo være

¹²Dette er beskrevet i Florence (1987).

verd å merke seg at kanskje er det kjønnsforskjeller som bør tas hensyn til i planleggingen av læreprogrammer.

Selve opplæringen hos oss foregikk på den måten at gruppas medlemmer satte opp en liste hvor de fordelte resten av arbeidsstokken mellom seg. Undervisningen foregikk når det var tid. Dette var en brukbar modell.

Arbeidsmessig er personalet delt inn i små grupper på 3 og 3 som går i den samme turnus. Dvs. at de alltid har nattevakter, senvakter og helgejobbing sammen. Planen var da videre å lære opp 1 person (den mest motiverte!) i hver av disse smågruppene.

De fleste har nå gått igjennom et oppgavesett. Vi har fortsatt ikke kommet så langt at alle kan Arbeidsark-programmet. Manual er laget, men enda er det stadige feil i programmet (eller utilstrekkelige kunnskaper hos oss). Fremdeles er vi på et slikt nivå at vi aldri er sikre på om det er oss eller maskinene det er noe i veien med.

Vi har altså lært noe:

Vi kan mer om prosjektarbeid, om organisering, samarbeid og styring.

Vi har lært om forutsetninger for læring og opplæring.

Om EDB har vi lært at det er langt fram...

- at utvikling vanligvis ikke foregår på brukernes premisser,
- at utvikling er dyrt og tar tid.

Men at mulighetene er mange.

Uten at Arbeidsarket er prøvet ut har nye idéer om bruk tatt form. At teknologien kan være en hjelp i det interne informasjonsopplegget bør også være klart. Fantasien hos noen av oss er i alle fall satt i gang.

DEL III. OM PRODUKTET: ARBEIDSARK-SYSTEMET.

Denne tredje delen av rapporten er viet produktet av systemutviklingen i prosjektet: Arbeidsark-systemet. I denne delen beskriver vi systemet, idéene bak det og hvordan det er implementert. I tillegg finnes en beskrivelse av design-prosessen som vedlegg: i vedlegg 2 beskriver vi hvordan vi opplevde samarbeidet med Norsk Data i denne fasen av prosjektet.

En mer detaljert beskrivelse av Arbeidsark-systemet finnes i "Dokumentasjon av Arbeidsark-systemet. Rapport nr. 4 fra Florence-prosjektet". Innholdsfortegnelsen til denne rapporten er tatt med som vedlegg 3. Rapporten distribueres av Norsk Data.

1. ARBEIDSARK SOM IDÉ: INFORMATISK OG SYKEPLEIE-BASERT BEGRUNNELSE.

I dette kapitlet skal vi forklare hvorfor vi mener at idéen som ligger til grunn for Arbeidsark-systemet er en god idé. Kapitlet er delt i 2 deler, og vi starter med å gi en informatisk begrunnelse for idéen. Vi presenterer da 3 sider ved idéen: sett i forhold til sykepleie, til sykehus og til informatikk.

Sykepleiernes begrunnelse kommer til etterpå, i punkt 1.2.

Vi håper dermed å illustrere at samarbeidet mellom sykepleierne og informatikerne i Florence-prosjektet har gitt et konkret resultat begge gruppene kan si seg fornøyd med.

1.1 Informatisk begrunnelse av Arbeidsark-idéen.

Sykepleiemessige argumenter.

To faktorer spilte en viktig rolle når vi i prosjektet skulle velge hva slags edb-system som skulle lages: Edb-systemet skulle være for vanlige sykepleiere, og det skulle brukes i daglige oppgaver i sykepleiernes arbeid. Vi måtte derfor velge et edb-system hvor utgangs-punktet var relativt enkelt, både bruksmessig og rent funksjonelt; vi ville ikke ha et edb-system som krevde en totalendring i arbeidssituasjonen. I tillegg måtte det være lett for sykepleierne å se nytten av systemet.

Arbeidsark-systemet oppfyller disse kravene. Systemet erstatter de håndskrevne lappene som skrives av sykepleierne ved hvert vaktskift. Med andre ord griper systemet direkte inn i sykepleiernes daglige arbeid. Samtidig er det enkeltstående system. Hvis systemet svikter, eller viser seg å være ubrukelig, er det ikke noe problem å gå tilbake til de gamle rutinene. Vi erstatter med andre ord ikke noe av det "formelle" papirarbeidet, så det blir ikke store forandringer i de daglige pasient-administrative rutinene. I tillegg er det lett for sykepleiere å se at selv en liten tidsinnsparing på rapporten ved vaktskiftene kan få konsekvenser. Først og fremst tror vi måten å rapportere på blir endret, og vi tror innføringen av Arbeidsark-systemet vil få konsekvenser for innholdet i hva som blir diskutert på møtene. Videre tror vi at systemet kan føre med seg endringer i arbeidsrutiner og i fordeling av arbeidsoppgaver mellom de forskjellige vaktene.

Sett i lys av bruksperspektivet er Arbeidsark-systemet en god idé fordi det støtter noe vi oppfatter som vesentlig i sykepleie: forholdet mellom 1) planlegging og organisering av arbeidet slik at alle pasientene får best mulig behandling fra avdelingen / posten sett som en helhet, og 2) planlegging og pleie av pasientene, sett fra en enkelt pasient sitt synspunkt.

Vi er klar over at dette ikke helt svarer til mange sykepleieres definisjon av "egentlig"

sykepleie¹. Men etter det vi har sett av sykepleieres arbeid, mener vi at det er viktig for sykepleiere å kunne kombinere pleie av pasienter, planlegging av denne pleien og administrasjonen av den - i forhold til én og til alle pasientene.

Arbeidsark-systemet kan brukes til å distribuere informasjon som er relevant for planlegging, administrering og utføring av sykepleie også til andre yrkesgrupper på sykehuset. Dette utvider spredningen av informasjon i forhold til hvordan dagens private papirlapper brukes. Vi mener det er en aktuell utvidelse fordi sykepleiere er profesjonelle observatører. Sykepleierne observerer pasienter for å kunne ta pleiemessige beslutninger. Men de er også observatører for andre yrkesgrupper på sykehus, f.eks. leger og fysioterapeuter. Det betyr at de ikke bare samler inn, men også formidler pasientinformasjon.

I tillegg til å administrere seg selv, har sykepleierne ofte som oppgave å administrere legene. Dette er nok delvis knyttet til kjønnsroller som avspeiles i yrkesroller, men også til at sykepleierne er knyttet til en post og de pasientene som ligger der, mens legene løper mer rundt². Dermed blir det sykepleierne som må administrere samarbeidet mellom ymse typer leger, siden de får merke konsekvensene først hvis legene ikke samarbeider. Arbeidsark-systemet kan brukes som huskelapp for legene også, slik at de får støtte til å administrere sin egen hverdag, og selv kan skrive ned det som skal gjøres med pasientene.

Arbeidsark ser også ut til å kunne bli brukt til andre sykepleiemessige funksjoner, som f.eks. å dokumentere arbeidsbelastning på posten, uten at det var forutsett av oss. Hvis systemet har en god struktur (for sykepleiere) vil det kunne dukke opp flere nye forslag til bruksområder etterhvert.

Alt i alt kan vi si at Arbeidsark-systemet er sykepleiernes system. Det tar utgangspunkt i deres arbeid. Det er fint hvis andre også kan bruke systemet, men det er ingen betingelse for at systemet skal bli vurdert som vellykket fra vår side.

Sykehusmessige argumenter.

På sykehus har edb til nå stort sett vært brukt til "vanlig" administrasjon, dvs. lønn, lager etc., til å holde orden på prøvesvar på laboratorier, og til pasientinformasjon. Det har også vært laget noen medisinske systemer, men de har vært mer enkeltstående fenomener. De medisinske dokumentene, som journalen og deler av den, har for det meste vært urørt. Og med god grunn. Det er en mengde dokumenter som brukes i pleie og behandling av pasienter. Alle dokumentene har en historie, de brukes på en spesiell måte, og gjerne forskjellig av ulike yrkesgrupper. Pasientinformasjonen er med andre ord differensiert i forhold til ulike yrkesgrupper og forskjellige bruks- og ansvarsområder. Dette fører igjen til at den "samme" informasjonen lagres flere steder. I praksis betyr det at mange pasientopplysninger skrives ca. 5 ganger når en pasient legges inn på en post, fordi det er dette antall skjemaer som følger pasienten på posten. Det er klart at dette tar tid, og oppleves som unødvendig "administrasjon".

I tillegg er det et problem at selv om det eksisterer (skrevet) informasjon om en pasient, er ikke det noen garanti for at det er lett å finne denne informasjonen. Ønsket om et

¹ "Egentlig" sykepleie uttrykker det sykepleiere ideelt skal gjøre: pleie av hele personer. I sammenhenger der sykepleie blir undervist og / eller omtalt i mer abstrakte og generelle termer, beklages det at sykepleiere ofte ikke får tid til "egentlig sykepleie" på grunn av personellmangel og arbeidsdeling på sykehusene. Uttrykket "egentlig sykepleie" finnes brukt f.eks. på seminar om sykepleieforskning i Stavanger i oktober 1984, se Florence (1985).

² Dette er også knyttet til kjønn. Cockburn (1985) viser at menn generelt beveger seg over ett mye større fysisk og geografisk område enn kvinner.

"samlet" pasientinformasjons-system skyldes altså både et ønske om mindre arbeid ved registrering, og et ønske om raskere gjenfinning.

Foreløpig har det vært vanskelig å overføre pasientinformasjon til edb-systemer, selv om mange drømmer om det. En av grunnene er at dokumentene og informasjonen henger sammen. Det vil ikke lette arbeidssituasjonen om ett enkelt skjema blir overført til et edb-system. Det betyr bare at noe arbeid gjøres ved terminal, og noe ved skrivebordet. Hvis det skal monne, må alle skjemaene kunne nås gjennom edb-systemet. Og da nytter det ikke å ta utgangspunkt i ett og ett skjema, da må all informasjon på sykehus planlegges og tenkes fra bunnen av. Å erstatte eksisterende papirer og skjemaer på sykehuset med "edb-baserte skjemaer" vil ikke nødvendigvis resultere i noe godt edb-system³. Datamaskiner gir tross alt noen andre muligheter og begrensninger enn papir og blyant / skrivemaskin.

Utviklingen av et "heldekkende" informasjonssystem ville kreve store ressurser, og det er vel kanskje ikke sykehusene som har mest å bidra med når det gjelder tid og penger til å delta i systemutvikling i våre dager. Selv om det teoretisk sett går an å lage et edb-system som kunne gi relevant informasjon til ulike grupper, ville det være nokså halsbrekkende å innføre et slikt edb-system. Systemet måtte virke fra første sekund, og det måtte være tilgjengelig 24 timer i døgnet. I tillegg ville det kreve enorme ressurser å lære opp alle til å bruke det (inkludert vikarer og ufaglært personell). Problemer ville kunne oppstå ikke bare fordi mange er ukjent med å bruke slikt utstyr, men også fordi det kreves en omstilling i måten å tenke informasjon på, og omstilling i forhold til den profesjonelle utdannelsen.

Med andre ord: vi tror ikke noe på å ta utgangspunkt i ett eksisterende dokument, og vi tror det tar lang tid før "heldekkende" pasientinformasjons-systemer blir introdusert på sykehus.

Hva har så dette med Arbeidsark-systemet å gjøre?

Vi tror at Arbeidsark-systemet kanskje kan være en kile inn i det eksisterende "papir-baserte" informasjonssystemet. Arbeidsark tar utgangspunkt i organiseringen av pleie og behandling mer enn i selve utførelsen av pleie og behandling av hver enkelt pasient. Kanskje det kan være et fruktbart utgangspunkt for en diskusjon om pasientinformasjon på sykehus?

Det nåværende Arbeidsark-systemet åpner for muligheter for informasjons-spredning til flere yrkesgrupper, der de samme dataene kan brukes i forskjellige sammenhenger, av ulike folk. Med systemet introduserer vi "engangs-papiret" til sykepleierne, og mulighetene og fleksibiliteten i systemet ligger i spennet mellom å betrakte informasjonen som øyeblikksaktuell - og uaktuell neste vakt -, og likevel kunne oppbevare informasjonen til en eventuell nyinnleggelse og akutt-situasjon. Arbeidsark-systemet vil ikke, i sin nåværende form, kunne brukes til å dokumentere den pleie og behandling som er gitt pasienten (slik som journal og kardex nå gjør).

I tillegg tror vi at det er mulig - på sikt - å kople Arbeidsark-systemet mot noen av de edb-systemene som allerede eksisterer, som f.eks. systemer til pasientmottak, laboratorier, pleietyngde-registrering og eventuelt turnusplanlegging. Utvidelsen av Arbeidsark-systemet kan med andre ord gå både i retning av informasjon om enkelt-pasienter, og mot mer administrativt orienterte systemer, f.eks. dokumentasjon av pleie og organisering av arbeidet.

³ Newman (1983) kaller dette "paper +"-strategien: å "oversette" et papir-baserte rutiner direkte til et edb-basert system og bare utnytte edb-systemenes muligheter som "pynt", som + til den gamle rutinen.

Informatiske argumenter.

Fra et informatisk synspunkt er Arbeidsark-systemet interessant fordi det dreier seg om et system som støtter samarbeid. For det første er det en del felles trekk ved samarbeid som gjør at noen av egenskapene ved Arbeidsark-systemet kan anvendes i helt andre typer yrker og på andre arbeidsplasser. For det andre stiller edb-systemer for samarbeid andre krav til utformingen av systemet, til muligheter for endring, vedlikehold og opplæring, enn det systemer for enkeltpersoner gjør. I begge tilfeller illustrerer Arbeidsark mer generelle spørsmålsstillinger i informatikk.

Samarbeid, som vi oppfatter det, dreier seg ikke bare om å utføre handlinger som bidrar til at et felles mål blir nådd. Det dreier seg også om at planlegging og administrasjon skjer i nær tilknytning til utføring av arbeidet, slik at planlegging ikke uten videre kan skilles ut som en egen oppgave. Dette illustreres lett ved å se på service-yrker, hvor en del av jobben er å være mottakelig for avbrudd og å justere planene i forhold til avbruddene⁴. I tillegg kan planlegging være en kollektiv prosess. Å delta i en gruppe, i et samarbeid, betyr å ta del i arbeidet og i ansvaret for at arbeidet blir gjort. Når gruppearbeid dreier seg om å gi service, vil hvert enkelt gruppemedlem ta ansvar for at alle kunder får samme service uavhengig av hvilken person i gruppa de kommer ut for. Informasjonsutveksling og administrasjon av arbeidet må her ta andre former enn hvis en tar utgangspunkt i samarbeid om f.eks. å produsere hver sin bit til ett produkt.

Arbeidsark-systemet brukes som grunnlag for enkeltpersoners beslutninger, og er samtidig et hjelpemiddel for distribusjon av informasjon. Disse funksjonene er ikke spesielle for sykepleie. Vi (Gro og Tone) kjenner igjen vårt eget "Arbeidsark" på kontoret, realisert ved hjelp av den mer gammeldagse tavle-teknologien. Vi bruker tavla for 1) huske på individuelle avtaler, 2) lage kollektive avtaler om samarbeid og 3) vanligvis en plan over det vi jobber mest med i øyeblikket, enten det nå er et program, et edb-system, en artikkel eller en rapport. For oss er tavla et godt hjelpemiddel fordi vi sitter på samme kontor. Hvis det hadde vært større geografisk avstand mellom oss eller vi hadde vært flere personer, kunne vi tenkt oss å lage vår egen versjon av Arbeidsark-systemet. Vi burde kanskje presisere forskjellen på et Arbeidsark-system og et vanlig mail-system, eller en felles kalender: poenget er å stille sammen forskjellige opplysninger som er relevante for samarbeidet. Det er vanskelig å gi generelle retningslinjer for hva slags informasjon som skal stilles sammen, siden dette er avhengig av organiseringen av arbeidet og av de aktuelle arbeidsoppgavene. For eksempel er tidsperspektivet nokså forskjellig for oss og sykepleiere: Vi planlegger neste uke, mens sykepleierne planlegger neste vakt.

Et edb-system for samarbeid er nytt i forhold til den utviklingen vi har sett innenfor informatikken de siste årene. Da interaktive systemer ble introdusert, ble de i begynnelsen laget over samme mal som de satsvise systemene. Dette førte til unødvendig rigide systemer, som heller ikke var spesielt "bruker-vennlige". Reduksjon av pris og størrelse på datamaskiner gjorde at et nytt ideal oppstod: det "personlige" systemet, eller "saksbehandler-systemer". Systemene skulle kunne tilpasses av den enkelte og til den enkeltes arbeidssituasjon. Dette førte til at brukergrensesnittene ble mer fleksible, i tråd med idealet at edb-systemet skulle være et "verktøy" som ble brukt i arbeidet, eller et "medium" for kommunikasjon mellom individer.

Et edb-system som skal støtte samarbeid, må ha egenskaper som er forskjellig fra et individ-orientert edb-system. Det må f.eks. være en viss grad av standardisering som vil avspeile enighet om hvordan edb-systemet skal brukes. Et eksempel på dette kan være struktureringsgraden av Arbeidsark-systemet. Foreløpig er det lite informasjon som er "eksplisitt" representert, dvs. det er en lav struktureringsgrad, siden meste-

⁴ Se Berger & Offe (1981).

parten av pasientinformasjonen skrives i fritext. Årsaken til dette er en enighet om at sykepleierne ikke helt vet hva som er nyttig som "felles-informasjon" om pasientene. Etterhvert kan det hende at det blir enighet om å bruke Arbeidsarket på bestemte måter, f.eks. at enzymverdier skal skrives på nederste linje i det siste tekstfeltet. Da kan vi implementere dette, og dermed øke struktureringsgraden⁵.

Standardiseringen avspeiler med andre ord noe av den enigheten samarbeidet bygger på. Hvis edb-systemet skal endres, kreves det forhandlinger om dette. Det er upraktisk å vedlikeholde mange forskjellige grensesnitt, og samtidig vedlikeholde programmer som gjør at hvert enkelt grensesnitt tilpasses "sine" brukerne. Saksbehandler-idéen forutsetter dessuten at brukerne har hver sin terminal, eller at de logger seg inn / "gjør noe" hver gang de skal bruke systemet. I vårt tilfelle er dette nokså upraktisk, siden det er flere som bruker samme terminal, og de logger stort sett ikke ut når de forlater terminalen.

Arbeidsark som edb-system for samarbeidsstøtte er altså på flere punkter i strid med den "individuelle" ideologien som er så utbredt i dag, og som også implementeres i edb-produkter⁶.

Det siste punktet i vår lovtale om Arbeidsark-systemet følger av dette. Arbeidsark-systemet er også interessant fordi det utfordrer våre begreper om hva som er en "teknologisk utfordring". Når vi har diskutert edb-systemet i informatikk-kretser, blir systemet ofte avvist som uinteressant fordi det ikke innebærer teknologisk nyutvikling, men bruk av eksisterende teknologi på nye måter. Dette sier noe om holdninger i tekniske miljøer om hva som er interessant. Men relativt enkle tekniske løsninger kan være vel så nyttige som teknisk avanserte løsninger⁷, og det kan ligge mange tekniske utfordringer i å realisere et tilsynelatende enkelt system for en ny type bruksområde ved hjelp av eksisterende teknologi. En løsning som er teknisk avansert i tradisjonell forstand, vil ofte dreie seg om et lite område av en persons yrkeskunnskap, enten det gjelder tekniske beregninger eller mer "ekspertsystem-aktige" aktiviteter. Når en fordyper seg i ett lite område, vil det ofte være vanskelig å se sammenhengen mellom det spesialiserte området og det helhetlige i yrket. Det gjør at det (for informatikere, selvfølgelig!) kan være vanskelig å skjønne kompleksiteten i andre yrker, fordi kompleksiteten ligger i kombinasjonen av aktiviteter mer enn i én enkelt arbeidsoppgave.

Informatisk sett har Arbeidsark-systemet vært faglig spennende å lage, selv om vi altså ikke har konstruert tekniske nyvinninger i tradisjonell forstand. Våre begreper om "god systemutvikling" har også blitt utfordret. Edb-systemer for samarbeid stiller andre krav både til utvikling og innføring av systemet enn individ-orienterte edb-systemer. Det er vanskeligere å lage et system for en gruppe enn for én person, f.eks. tar det lengre tid å spesifisere hva systemet skal gjøre fordi vi må gjøre eksplisitt den enigheten i gruppa som systemet baserer seg på.

Å konstruere et Arbeidsark-system som beholdt det idé-messige gode sett fra de tre sidene vi har beskrevet her: sykepleie, sykehus og informatikk, har vært overraskende vanskelig og spennende. Vi har hatt stort faglig utbytte av å utforske og overskride vanskelighetene, og har derfor valgt å bruke den tiden vi har hatt behov for, framfor å forlate idéen. Vi mener selv vi har tjent på det, og er stolte over å kunne si at idéene i

⁵ Dette er forøvrig i tråd med idéen om "stepwise structuring": at konstruksjon av regler og struktur bør foregå skrittvis fordi en kan komme til å introdusere for mye og feil struktur dersom en skal lage den på forhånd, før systemet er prøvet brukt. Både for mye og for lite struktur kan gjøre et edb-system ubrukelig, se Sandewall et.al(1982) og Florence(1985).

⁶ Se III.5 om NOTIS-RG/RP. Individ-orienteringen er utbredt fordi den teknologiske utviklingen med PCer og arbeidsstasjoner, med ferdigpakker med muligheter for individuell tilpasning og fleksible menysystemer, gjør det mulig å "skreddersy" et edb-system for den enkelte bruker. Det anses som nyskapende å forske på disse nye teknikkene.

⁷ Dette er blant annet dokumentert i Perby (1987).

Florence-prosjektet faktisk kan sies å være implementert i Arbeidsark-systemet.

1.2 Sykepleie-basert begrunnelse av Arbeidsark-idéen.

I dette punktet gir vi igjen ordet til sykepleierne.

Etter en vinter (1986) med jevnlige "bli kjent"-møter, ble det gjennomført et 2-dagers seminar, med sykepleiere og informatikere som deltagere, og "sykepleie og EDB" som tema. Arbeidsgruppa på SYD 5 hadde store forventninger til dette seminaret. Mange av oss gikk nærmest og ventet på en slags aha-opplevelse om endelig å gjennomskue dette med EDB.

Program for seminaret var:

- 1) Diskutere tanker om EDB og sykepleie
- 2) Læring / trening på Macintosh-maskiner
- 3) Se på informasjonsprosesser og prosedyrer på sykehuset

Vi gikk igjennom våre daglige informasjonsmøter og forsøkte og se systematisk på informasjonsflyten i avdelingen. Veggrafer ble laget for å illustrere ulike typer informasjonsmøter og prosesser⁸.

Dette var en ny måte å analysere hverdagslige gjøremål på, for oss. Vi fikk tid til å se i system, rutiner som mange av oss bare har fått overlevert. Nødvendighet og hensiktsmessighet, og ikke minst gjensidig avhengighet de ulike rutinene imellom, trådte klart fram.

Vi gikk systematisk igjennom hele bunken med ulike kurver, ark og skjemaer som flyter til og fra våre informasjonsmøter, og vårt håp om å skjære drastisk ned på skjernamengden sank. Det kom klart fram at våre ulike måter å dokumentere observasjoner på henger nøye sammen med den organisasjonsformen vi har valgt å jobbe etter.

Vi kunne ikke fjerne et eneste skjema uten at dette hadde store konsekvenser for strukturen i arbeidet, og *det* var ikke meningen med et slikt prøveprosjekt.

Etter å ha gått igjennom papirhaugen på denne måten sto vi da tilbake med det minst formaliserte av alle våre ark, - nemlig våre håndskrevne notater fra hver rapport.

Alle vakter på sykehus begynner med rapport. Dette vil si et informasjonsmøte hvor ansvarlige for det "avgående" arbeidsteam gir en oversikt over tilstand, observasjoner, beslutninger og planer for hver enkelt pasient. Dette mottas og noteres av det påtroppende arbeidslag, og brukes som beslutningsgrunnlag for fordeling av neste vaks ressurser.

Idéen vi ble sittende igjen med etter alle diskusjonene var: Hva om vi laget et "ferdig" rapportark med standardopplysninger om pasientene, lagt inn etter hvert som de kom. Dette kunne da deles ut til alle ved vaktskifte, og den muntlige rapporten kunne begrenses til å fortelle om "ekstra" behov, observasjoner, planer osv. på de pasientene hvor dette var aktuelt.

I tillegg fant vi ut at ved å gi dette arket en form som et kart over hvert team på avdelingen, plassere pasientene i de riktige "sengene", og i tillegg føre opp hvilket personell som hadde hvilke arbeidsoppgaver, ville vi få en organisatorisk oversikt som ikke alle uten videre har idag.

Slik oppsto idéen om Arbeidsarket.

⁸ Veggrafings-aktivitetene er nøye beskrevet i "Læring som forutsetning for design. Rapport nr. 2 fra Florence-prosjektet" (Florence(1987)).

Nå i ettertid ser vi at våre forestillinger om hva en "datamaskin kan" i høy grad var preget av at vår første læring på maskin foregikk på Macintosh-produkter. Vi trodde kort og godt at en datamaskin var en datamaskin. Erfaringene med denne typen verktøy (som var den eneste erfaring vi hadde) gjorde det vanskelig for oss å fatte de store problemene som skulle vise seg å oppstå når det gjaldt programmering.

Vi syntes at vi hadde funnet fram til et svært enkelt prøveprosjekt, som kanskje kunne gi oss en gevinst i form av tid.

Vi følte vel imidlertid ikke at dette var noe så helt spesielt, så hvis det var så vanskelig kunne vi da alltså finne på noe annet...

Telefoner fra informatikerene om "snarlige gjennombrudd" sa oss i grunnen ikke så mye. Informasjon om at de beveget seg på "grensen" av de ymse verktøyene, enda mindre.

Vi synes jo da at man burde bruke et annet redskap eller spørre oss om å forandre på oppgaven, vi var da ikke urimelige heller...

Etter hvert som våren skred fram og problemene stadig ikke var løst, synes vi nok at tiden hadde vært bedre nyttet ved å ta en grunnleggende diskusjon på idéen. Vi følte oss litt satt ut av spill og syntes det var ergerlig at tiden skulle gå uten at vi fikk lært og brukt noe av alt utstyret som sto på avdelingen. Riktignok var vi noen som hadde nytte av det som tekstbehandlingsutstyr, men det var jo ikke bare det vi skulle lære.

Nysgjerrigheten og interessen blant personalet på posten forsvant fullstendig. Florence ble nå bare nevnt i spydige kommentarer. Motivasjonen hos arbeidsgruppa var på et nullpunkt, vi syntes det var vanskelig å svare når folk spurte om hvordan det gikk.

Slik var situasjonen når sommerens avsluttende oppsummeringsmøte fant sted i juni -87.

Dette møtet følte vi fungerte bra for oss (det gjorde forresten de fleste fellesmøtene med informatikerne).

Det ble lufting av frustrasjoner fra alle sider, - som alltid gjør godt! - og, vi fikk tatt en forholdsvis grundig gjennomgang av hvilke erfaringer vi hadde gjort.

Vi gikk tilbake til de to gruppenes målsettinger. Det var jo klart at informatikerne var mer fornøyd med prosjektet så langt enn det vi var. Vi hadde jo som mål å lære EDB og å gjennomføre et prøveprosjekt. Dit var vi ikke kommet.

For folk som jobber i et system hvor effektivitet og produktivitet er et konstant krav, og mangel på ressurser er et daglig problem, fortøner "forskning" seg som den reneste luksus. Det var vanskelig for oss å se at prosjektet som helhet hadde gitt noen resultater. Det var ingenting som syntes, ikke noe "produkt". Vi er vant til å se resultatene av et arbeid. Det var klart at vi hadde lært, men helt andre ting enn det vi hadde forventet.

2. KRAV TIL ARBEIDSARK-SYSTEMET.

Funksjonalitet:

- erstatter håndskrevne notater ved vaktskifte, "engangspapir"
- standardisering av privat informasjon
- distribusjon av opplysninger til andre personellgrupper, f.eks. leger
- "åpen-dør"-system: hjertepasienter har ofte en kronisk sykdom (angina pectoris), og de kan risikere å bli lagt inn når sykdommen er akutt / livstruende. Pasienten kan komme tilbake fra andre poster / avdelinger, og gjentatte innleggelser kan forekomme, derfor kan det være praktisk hvis "gammel" pasientinformasjon kan hentes fram. Dette svarer til et av forslagene sykepleierne på Voksentoppen hadde: Et arkiv med

opplysninger om de som vil kunne komme ved akutte astma / allergi-anfall: Åpen-dør-systemet (som vi husker, kommer de fleste pasientene til Voksentoppen når sykdommen er i en roligere fase).

Bruk:

- pasientdata skrives inn ved innkomst samtidig med at de andre papirene fylles ut
- ajourføring og endring av data: foreløpig ikke bestemt hvem og når
- sletting av data: gjøres ved prøveperiodens slutt
- makulering av "brukte" Arbeidsark: makuleringsmaskin?
- sikring:
 - de organisatoriske og fysiske forholdene gjør at det er en utstrakt sosial kontroll med hvem som kan oppholde seg på de rommene hvor terminaler og datamaskin står.
 - Pasientinformasjon kan fjernes fra skjermen ved behov.
 - Det er passord for å bruke systemet, men dårlig "disiplin": folk logger seg ikke alltid ut når de er ferdige med det de skal gjøre.
 - Programmene ligger på et annet område i maskina enn det brukerne har adgang til. Dermed er det vanskelig å endre eller slette programmer.

Krav til innføring / opplæring:

- systemet må innføres på en gitt dato (i motsetning til gradvis innføring) siden det er et system for samarbeid
- opplæring tar utgangspunkt i den eksisterende organiseringen av arbeid.

Konsekvenser:

- kortere morgen-rapport. Dette kan føre endring av morgenrutinene, og antakeligvis til at innholdet i morgenrapporten blir endret.
- sykepleierne slipper å være "barnepiker" for legene: hvis legene mottar et ark med administrativ informasjon, slipper de å minne legene på hva de skal gjøre og administrere dagen for dem.

Tekniske krav:

- flerbrukersystem: oppslag fra forskjellige terminaler
- lett å flytte pasientinformasjon fra en seng til en annen (ikke dobbeltregistrering av pasientinformasjon)
- mulig å hente opplysninger om pasienter som har ligget på avdelingen før
- systemet behøver IKKE være oppe 24 timer i døgnet
- lagring av pasientinformasjon i X måneder. Dette er ikke nærmere bestemt siden all pasientinformasjon slettes når prøvetiden er over.
- samsvar mellom oversikts-skjermbilder og oversikts-papirrapporter så langt det er mulig
- sammenstilling av informasjon er et absolutt krav til papirrapportene, men kan fravikes i skjermbildene

Uavklarte tekniske / organisatoriske spørsmål:

- kopling mot andre registre på sykehuset, f.eks. i mottakelsen eller på laboratoriet?

3. BESKRIVELSE AV ARBEIDSARK-SYSTEMET.

Arbeidsark-systemet er et edb-system for sykepleiere på post- og avdelingsnivå. Systemet vil gi alle sykepleiere på vakt en kort oppsummering av de viktigste pasientopplysningene for de pasientene som ligger til overvåking. Informasjonen kan også gis til andre personer på avdelingen, f.eks. legene. Hovedpoenget med systemet er en ny måte å spre opplysninger på, både opplysninger om organiseringen av arbeidet, om pasientene og om sammenhengen mellom pasientpleie og administrasjon.

Arbeidsark-systemet skal erstatte de håndskrevne notatene sykepleierne bruker i sitt daglige arbeid. Sykepleierne skriver notatene under rapportene ved hvert vaktskifte, på små ark, gjerne A5 eller A6 format. De bruker notatene i løpet av dagen både til å memorere informasjon om pasientene, huske på spesielle ting som skal gjøres og til å notere på. Eksempler på slike håndskrevne notater finnes i figur 1.

Arbeidsark-systemet tar altså utgangspunkt i dagens uformaliserte praksis på SYD 5. Systemet vil støtte opp under hverdagens aktiviteter ved å gi støtte til det viktige samspillet mellom pleie og administrasjon. På denne måten vil systemet bidra til å utvide mulighetene i dagens realitet. Samtidig vil ikke systemet bidra til en ytterligere arbeidsdeling på post / avdelingsnivå.

Det viktigste resultatet av Arbeidsark-systemet er det arket som erstatter sykepleiernes lapper. Det blir laget ett ark for overvåkningspasientene (se figur 2), og ett eller to for resten av pasientene (avhengig av om det er korridor-pasienter eller ikke). Dette arket innebærer noen forandringer i forhold til de håndskrevne lappene. For det første vil edb-systemet innebære en standardisering av pasientinformasjon. Informasjon som ble skrevet ned på en "privat" måte før, vil bli delt og felles nå. Vi regner med at det vil bli diskusjoner om hva som er relevant pasientinformasjon når Arbeidsark-systemet har vært i bruk en tid. For det andre er ikke pasientopplysningene sekvensielle, de er ordnet geografisk, som et kart over avdelingen. Og for det tredje inneholder arket informasjon om organisering av arbeidet på denne vakta. Hittil har slike opplysninger vært å finne på håndskrevne lapper, festet på lett synlige steder (f.eks. "Scopet").

Pasient-opplysningene skal ikke bare presenteres på et papirark, de må også skrives inn i edb-systemet. Dette gjøres via terminal. Det står terminaler på vaktrommet, kontoret og overvåkningssentralen. Vi skal gi en kort presentasjon av skjermbildene som blir brukt og hva de inneholder. De tilgjengelige skjermbildene presenteres i en meny, se figur 3.

Pasientinformasjonen skrives inn i det bildet som heter PI, se figur 4. Det er også her en endrer og fjerner pasientinformasjon, og finner informasjon om pasienter som har ligget på posten tidligere. Ved hjelp av dette skjermbildet kan en hente alle pasientopplysningene som er lagret i edb-systemet, enten det dreier seg om pasienter som er innlagt på posten i øyeblikket eller om pasienter som har vært innlagt tidligere.

En kan få skrevet ut en sekvensiell oversikt over alle sengene på posten med informasjon om navn og fødselsår på pasientene i de respektive sengene, ved å gå til det bildet som heter OS (Oversikt Senger), se figur 5.

En kan også få geografisk oversikt over pasientene ved å taste OO (Oversikt Overvåkning), OU (Oversikt Uterom) eller OK (Oversikt Korridor). OO er vist i figur 6. Disse skjermbildene inneholder noe av den informasjonen som finnes i Arbeidsarkene, dvs. de rapportene som skrives ut. Opprinnelig var det meningen at disse bildene også skulle inneholde team-opplysninger. Men det viste seg å være en teknisk umulighet. Så teamopplysningene må skrives inn ved hjelp av et eget skjermbilde, TEAM, se figur 7.

Ett av poengene med Arbeidsark-systemet er at sykepleierne skal slippe å skrive de samme pasientopplysningene på nytt selv om pasienten for eksempel skulle bli flyttet rundt på posten (det skjer 8-10 flyttinger av pasienter hver dag). Flytting medfører at pasientene ikke bare må trilles rundt i korridorene, informasjonen om dem må også flyttes rundt. Dette gjøres i det skjermbildet som heter FLYTT, se figur 8.

507-2	-55	508	-20	510	-53	512-1	-51
Duck, Anton		Aas, Peder		Dalton, Averell		Dalton, Joe	
Diag:xxxx		Diag:xxxxxxxxxxxx		Diag:Sulten		Diag:herskesyk	
Sykehist:xxx		Sykehist:		Sykehist:		Sykehist:	
IV:		IV:		IV:		IV:	
US:		US:		US:		US:	
507-1	-63		ARBEIDSGRUPPE	ARBEIDSOPPGAVER		512-2	-52
Duck, Donald		Florence-gruppa:	finne fler deltakere	Dalton, Jack		Dalton, Jack	
Diag:xxxx		Unni	spre entusiasme	Diag:bror til Joe		Diag:bror til Joe	
Sykehist:xxxx		Frøydis	selge prosjektet	Sykehist:		Sykehist:	
IV:		Øyvind	?	IV:		IV:	
US:		Helene	følge med!!!	US:		US:	
		Harald					

Figur 2: Arbeidsark for overvåkningsrommene - det arket som produseres av edb-systemet og erstatter de håndskrevne lappene.

Florence - versjon fra 13/5-87

VELKOMMEN TIL ARBEIDSARK-SYSTEMET

Kommandoer: 00: Oversikt over Overvåknings-rommene
OU: Oversikt over "Ute-rommene" - ER IKKE FERDIG!!!!
OS: Oversikt over Senger
PI: Pasient-Informasjons-bilde
PI-RP: Pasient-Informasjons-bilde slik det ser ut i RapPorten
FLYTT: FLYTTer pasienter inn og ut og mellom senger
TEAM: TEAM-opplysninger

HJELP-knappen: generell informasjon om kommandoer du kan bruke i
ARBEIDSARK-systemet.

SHIFT og

HJELP (samtidig): informasjon om innholdet i skjermbildene i
systemet.

Figur 3: Menyen for edb-systemet viser hvilke kommandoer som kan gis og hva som kan gjøres.

01/09/87

Florence, 23/4-87

REGISTRE PASIENT-INFORMASJON I ARBEIDSARK-SYSTEMET

Efternavn: Dalton

Fornavn: Edward

Fødselsår: 53

Diagnose: bror av Joe, Jack og Averell

bykehast: fetter av de "ekte" Dalton brødrene

IV:

US:

Vil du slette pasient-informasjon? (J/N)

NB! Du kan bare slette info om pasienter som ikke er ligger i en seng

Figur 4: PI - bildet for PasientInformasjon. Pasientinformasjon skrives inn, endres og slettes i dette bildet.

Dato: 01/09 87 Florence - versjon 17/3-07

OVERSIKT OVER HVILKE PASIENTER SOM LIGGER I SENGENE PÅ POSTEN.

Sengnr: 507-2	Etternavn: Duck	Fornavn: Anton	Født: 55
Sengnr: 508	Etternavn: Aas	Fornavn: Peder	Født: 20
Sengnr: 510	Etternavn: Dalton	Fornavn: Averell	Født: 53
Sengnr: 512-1	Etternavn: Dalton	Fornavn: Joe	Født: 51
Sengnr: 507-1	Etternavn: Duck	Fornavn: Donald	Født: 63
Sengnr: 512-2	Etternavn: Dalton	Fornavn: Jack	Født: 52

Vil du SLETTE pasientdata? (J/N)

Du kan bare slette pasientdata for pasienter som IKKE ligger i en seng

Pasienter i en seng må flyttes først: Vil du FLYTTE pasient? (J/N)

Figur 5: OS - Oversikt over Sengene. Dette skjermbildet viser en sekvensiell oversikt over sengene på posten med tilhørende pasienter.

Dato: 01/09 87 Florence - versjon 3/4-87

ENKEL VERSJON AV OVERSIKTBILDE AV OVERVÅKNINGS-ROMMENE ARBEIDSARK

507-2	55	508	20	510	53	512-1	51
Duck		Aas		Dalton		Dalton	
Anton		Peder		Averell		Joe	
Diag:		Diag:		Diag:		Diag:	
XXXX		XXXXXXXXXX XXXXXX		Dalton		hørskesyk	
		XXX					

507-1	- 63		512-2	- 52
Duck		For opplysninger om team	Dalton	
Donald		om hvem som er på vakt	Jack	
Diag:		og hvem som har	Diag:	
xxxx		hvilke oppgaver	bror til Joe	
		gi menykommando TEAM		

Meny-kommandoer: OversiktUterom (OU), PasientInfo (PI), FLYTT og TEAM

HUSK!! TRYKK FØRST 'C' OG SÅ 'F' FOR Å RYDDE SKJERMBILDET.

Figur 6: OO - Oversikt Overvåkning. Dette skjermbildet gir en geografisk oversikt over overvåkningsrommene og svarer utseende-messig til en av de papir-rapportene som blir laget av edb-systemet (se figur 2).

Dato: 01/09-87		Florence - versjon 22/4-87	
INFORMASJON OM TEAM			
Pleiere:		Arbeidsoppgaver:	
Ole		passer på hakkespettene	
Dole		ansvar for 507	
Doffen		ansvar for 508 og 510	
Hetti		ansvar for 512	
Letti		EKG	
Netti		medisiner	
Kommandoer: OversiktOvervåkningsrom (OO) OversiktUterom (OU), PasientInfo (PI), FLYTT			

Figur 7: TEAM - gir oversikt over hvem som er på vakt og hvilke arbeidsoppgaver de har. Alle teamopplysninger legges inn, endres og slettes via dette skjermbildet.

En starter med å lete fram den senga som skal flyttes, i vårt eksempel seng nr. 510. Vi kan se at det er Lucky Luke som ligger i denne senga. Deretter gjør vi et oppslag i pasientregisteret. Vi finner den pasienten vi ønsker å legge i seng 510: Averell Dalton. Deretter sier vi at vi ønsker å legge inn Dalton ved å skrive "I", og resultatet blir som i figur 9. Nå betyr dette at Luke er å betrakte som utskrevet av posten (men dataene hans finnes fremdeles i pasient-registeret). Han kan lett legges inn i en annen seng, eller i den samme, for den saks skyld, ved hjelp av FLYTT. For at en person skal kunne legges inn, flyttes eller fjernes fra en seng, er det en forutsetning at hun / han er registrert som pasient i pasient-registeret.

Dato: 01/09-87. Florence versjon 20/3-87

FLYTTING AV PASIENTER

Sengnr: 510 Etternavn: Luke
Fornavn: Lucky Født:

OPPSLAG I PASIENT REGISTERET:

Etternavn: Dalton
Fornavn: Averell Født: 53

Skal pasienten fra pasient-registeret legges inn i denne senga (I)
eller skal pasienten i senga flyttes ut (U)?

Figur 8: FLYTT - flytter pasientinformasjon fra en seng til en annen. Alle forbindelser mellom senger og pasienter håndteres i dette skjermbildet. I dette eksempelet ligger Lucky Luke i seng 510, og vi har funnet fram opplysninger om Averell Dalton fordi vi ønsker å legge ham i seng 510 istedenfor Luke.

Dato: 01/09-87. Florence versjon 20/3-87

FLYTTING AV PASIENTER

Sengnr: 510 Etternavn: Dalton
Fornavn: Averell Født: 53

OPPSLAG I PASIENT-REGISTERET:

Etternavn: Dalton
Fornavn: Averell Født: 53

Skal pasienten fra pasient-registeret legges inn i denne senga (I)
eller skal pasienten i senga flyttes ut (U)? I

Figur 9: FLYTT - etter at Dalton er lagt inn i seng 510.

4. DISKUSJON AV ARBEIDSARK-SYSTEMETS DATAMODELL.

Hittil i denne delen har vi presentert Arbeidsark-systemet ideologisk og konkret. I de 2 følgende kapitlene vil vi gå inn på de tekniske sidene ved systemutviklingen. Vi starter med å se på valg vi har gjort i forhold til strukturen i Arbeidsark-systemet: vi diskuterer den datamodellen som ligger til grunn i systemet. Med dette ønsker vi å begrunne den overordnede, tekniske design i systemet. I neste kapittel går vi videre med tekniske diskusjoner, men da legger vi vekt på valg og beslutninger vi har tatt i underveis i design-prosessen, dvs. endringer i systemet som skyldes tekniske forutsetninger mer enn overordnede betraktninger. Arbeidsark-systemet er, som alle andre edb-systemer, et kompromiss mellom idéer og den teknologien som eksisterer i øyeblikket. Vi ønsker å vise hvilke valg som skyldes hva.

Det var ganske enkelt å lage den datamodellen som ligger til grunn for Arbeidsark-systemet. Grunnlaget for modellen var en beskrivelse av ønsket utseende på en rapport som sykepleierne laget umiddelbart etter opplæringsperioden, våren 1986. Denne beskrivelsen tjente som spesifisering av edb-systemet. Vi har hele tiden hatt den samme datamodellen, og grensene for hvilke data som skulle behandles av systemet og hvordan databehandlingen skulle foregå, ble satt tidlig.

For å illustrere og tydeliggjøre de valgene vi har gjort i Arbeidsark-systemets datamodell, vil vi sammenlikne denne modellen med en annen, eksisterende datamodell til bruk i helsevesenet (se figur 10). Vi har her valgt Petter Hurlens modell for informasjonsutveksling i helsevesenet⁹. Hurlens datamodell er mer generell, den er større og gjelder hele helsevesenet, ikke bare sykepleieres arbeid. Vi vil derfor bare referere de delene av Hurlens modell som er relevante for Arbeidsark-systemet, dvs. bare de delene av modellen som har noe med sykepleiernes daglige arbeid å gjøre. Ved å diskutere hensiktene med de to modellene, forklarer vi langt på vei forskjellene mellom dem. Vi starter derfor med en overordnet diskusjon, før vi ser på modellenes innhold.

Hensikten med modellene.

Hurlens datamodell.

Hurlens datamodell er en *generell modell til bruk i helsevesenet*, for at data felles for forskjellige rutiner kan gis felles struktur. Hensikten er å lette utveksling av data mellom edb-systemer og rutiner, dvs. at modellen er laget for å standardisere administrasjon av informasjon i helsevesenet. Den skal inneholde administrative pasientdata, både beregnet på administrasjon på et sykehuset og generelt i landet. Utenom de rent administrative datene gir modellen rom for medisinske pasientdata, dvs. data som først og fremst brukes av leger.

Hurlens interesser er i første rekke *helsepolitisk styring og planlegging i helsevesener*¹⁰. Han åpner derfor også for at det kan konstrueres lokale databaser som

⁹ Se Hurlen, P. (1986): *En modell for informasjonsutveksling i helsevesenet*, hovedfagsoppgave ved Institutt for Informatikk, Universitetet i Oslo.

¹⁰ Hurlen (1986) sier i forordet til hovedfagsoppgaven:

"Denne oppgaven inneholder et forsøk på å lage en modell av helsevesenet. Modellen har som formål å bidra til å trekke opp retningslinjer for utnyttelsen av moderne edb-teknologi til informasjonsutveksling i helsevesenet. En forutsetning for oppgaven er at hjelp til informasjonsformidling i helsevesenet vil kunne ha en positiv effekt for pasientene. Jeg tror at:

- Bedre og mer komplett informasjon om en pasient er et bedre grunnlag for å ta korrekte avgjørelser om utredning og behandling.
- Bedre informasjon om en helseinstitusjons eller et helsesenters drift kan effektivisere driften og gi

inneholder data tilknyttet deler av helsevesenet som ikke omfattes av en sentral datamodell.

	Hurlens modell	Arbeidsark
Anvendelses- område	generell modell, helse- vesenet som helhet	lokalt tilpasset modell, en arbeidsplass
Hensikt	helsepolitisk styring og planlegging	støtte for sykepleieres daglige arbeid
Tids- perspektiv	statistikk, summarisk informasjon	"engangspapir"

Figur 10: Oversikt over hovedforskjellene mellom de to modellene

Hurlen legger vekt på at alle data i helsevesenet skal passe i en felles struktur. Fra edb-systemer basert på Hurlens datamodell skal det være enkelt å produsere rapporter som kan brukes av sentrale planleggere i helsevesenet. Enklere tilgang til summarisk informasjon vil kunne gjøre rapportering i helsevesenet bedre. En av årsakene til at tidligere forsøk på standardisert rapportering ikke har vært så vellykket, kan være at rapporteringen har krevd spesial-innsamling og -innlegging av informasjon. Dersom den sentralt interessante informasjonen kan trekkes ut fra data som av andre årsaker er lagt inn i edb-systemet, gir en standardisering av edb-systemet bedre muligheter for en vellykket sentral rapportering.

Arbeidsark-systemets datamodell.

Arbeidsark-systemet er et eksempel på et edb-system med lokal database, og systemets datamodell er forskjellig fra den sentralt orienterte modellen.

I Arbeidsark-systemets datamodell finner vi noen av de samme typene data som i Hurlens modell, men de brukes forskjellig og er derfor definert forskjellig. Informasjon fra Arbeidsark-systemet er beregnet for *lokal bruk*, først og fremst bruk i *sykepleiernes daglige arbeid*. Informasjonen skal ikke være historisk: hver rapport som tas ut trenger bare være aktuell for en vakt. Vi kan derfor kalle rapportene fra Arbeidsark-systemet "*engangspapir*" og papirene kan kastes når sykepleierne går fra vakt.

Informasjonen i Arbeidsark-systemet skal brukes som bakgrunn for de administrative, pleiemessige beslutninger sykepleierne tar i arbeidet. Systemet brukes for å sammenstille informasjon som sykepleierne mener er nødvendig for dette. Det er sykepleierne som trekker konklusjonene fra dataene edb-systemet presenterer; systemet produserer

grunnlag for en bedre prioritering av ressursene, både lokalt og sentralt.

- Bedre informasjon om befolkningens helse og om effekten av tiltak er viktig i forebyggende helsearbeid. Innføring av edb-teknologi vil alltid bygge på en avveining mellom positive og negative effekter. Mitt håp er at modellen vil være en hjelp til å foreta slike avveininger."

ingen "svar". En grunn til dette er at samme data brukes forskjellig av forskjellige sykepleiere, og i ulike brukssituasjoner, i tillegg til at andre personellgrupper kan bruke de samme dataene i andre sammenhenger.

Sykepleierne trenger informasjon om pasientene og om de personellmessige ressursene for å kunne administrere og planlegge pleien for posten og for hver enkelt pasient med de ressursene som er tilgjengelige på en bestemt vakt. Informasjon om pasientene betyr her informasjon om enkelt-pasienter, presentert i den tilstanden de er i i øyeblikket. Sykepleierne står selv for en evt. "summering" av informasjonen. Personellinformasjon betyr i Arbeidsark-systemet informasjon om hvem som er på vakt, og hvilke kategorier oppgaver den enkelte har ansvaret for.

Informasjon fra Arbeidsark-systemet vil i første rekke bli brukt til å spre informasjon om pasienter og personell til personellet som er på vakt. Den informasjonen systemet gir, kan brukes i mange sammenhenger, på forskjellige måter, og graden av struktur er derfor minimal.

Datamodellenes innhold.

Informasjon som skal støtte generelle administrative funksjoner i helsevesenet, kan ifølge Hurlen grupperes i 4 kategorier etter hva den dreier seg om:

1. Pasienter.

Pasient-informasjon gjelder både den enkelte pasient og pasienter sett som sykehusets klientell. Informasjon som gjelder den enkelte pasient brukes når behandling og pleie av pasienten skal bestemmes og gjennomføres. Informasjon om sykehusets pasienter er summarisk, evt. statistisk bearbeidet pasientinformasjon, beregnet på administrasjonen på sykehuset og i helsevesenet.

2. Medisin.

Medisinsk informasjon har først og fremst betydning for helsearbeidernes daglige arbeid, men vil også kunne bli brukt i planleggingen av sykehusets drift, f.eks. gjennom opprettelse av isolasjons-avdelinger o.l.

3. Helsearbeidere.

Vi finner noe av det samme skillet mellom enkeltstående og summarisk informasjon hos pasient-informasjon også når det gjelder helsearbeidere.

4. Ressurser, økonomi og materiell.

Ressurs-informasjon er viktig for planlegging (og gjennomføring) av pleie og behandling, både lokalt og sentralt.

Vi skal se på hvordan disse kategoriene data finnes i Arbeidsark-systemets modell, og sammenlikne det med hva Hurlens modell spesifiserer.

Pasient-data: personlige og medisinske data om pasienter.

Pasientdata dreier seg ifølge Hurlen om 3 forhold:

- 1) identifikasjon av pasienten som person,
- 2) identifikasjon av hennes plager og problemer, og
- 3) identifikasjon av tiltak og behandling av pasienten.

Identifikasjon av pasienten som person (1) er viktig for flere formål. Dels er det viktig for administrative formål, i forhold til trykkesystemet, dels er det viktig å forsikre seg om at pasienten er én bestemt person (på sykehuset). I administrative sammenhenger vil ikke navn, adresse og fødselsdato være tilstrekkelig for en entydig identifikasjon av en person, særlig ikke om vi tar med i betraktning at folkeregisteret opererer med en grense på 26 tegn for personnavn. For administrasjonen er fødselsnummer eller et sykehus-internt pasientnummer det eneste som sikrer entydig identifikasjon av en

person¹¹.

For en post regner vi det som usannsynlig at det samtidig ligger inne to pasienter med samme navn, adresse og fødselsdato. For å identifisere en pasient tar sykepleierne utgangspunkt i hvor pasienten ligger, så brukes (etter-)navn og fødselsår¹². Vi valgte å bruke sykepleierens måte å identifisere pasientene på til rapportene fra Arbeidsark-systemet.

Men dataene er mer bestandige enn vaktene, og vi ønsket at pasient-opplysninger som var lagt inn i systemet skulle være lette å hente fram igjen innenfor en gitt tidsramme. Vi valgte å holde fast ved troen på at det ikke samtidig skrives ut fra posten pasienter med samme navn, adresse og fødselsår. For å sikre entydighet innførte vi derfor en tidsparameter slik at hver pasient får registrert et tidspunkt for innleggelse, som etter hvert erstattes med tidspunkt for utskriving.

Vi skiller mellom inneliggende og utskrevne pasienter, relativt til posten. Dette er forskjellig fra det skillet Hurlen lager (s. 47-48) mellom inneliggende og polikliniske pasienter. Vi betrakter ikke polikliniske pasienter overhodet, og skiller derfor heller ikke mellom forskjellige typer forhold mellom sykehuset og pasienten¹³. Det som er interessant er om en pasient er inneliggende på denne vaken eller ikke.

Arbeidsark-systemets behov for identifikasjon av pasientenes plager og problemer (2) er svært forskjellig fra Hurlens syn. Hurlen opererer med en lege-orientert forståelse av pasientene og diskuterer deres problemer som lege (ss. 56-59). Problemene beskrives ved hjelp av en entydig diagnosekode¹⁴. Arbeidsark-systemet har også data om pasientens *diagnose*, men her er den en tekst på noe under 60 tegn. I tillegg inneholder systemet data om *sykehistorie*, som en tekst på ca. 170 tegn. Datene i disse kategoriene i Arbeidsark-systemet er sykepleie-relevante data, dvs. det er data som gir sykepleierne et bilde av pasienten. Sykepleierne bruker begreper som er kjent av de andre som er ansatt på posten, som "P3" eller "AMIN", men som er ubegripelige for andre. Det er en sosial enighet om betydningen av disse begrepene. Dermed kan vi si at kodene som brukes er lokale, og de er bestemt av faglig-sosiale regler, ikke av regler i edb-systemet.

Sykepleierne trenger også å vite om tiltak for pleie og behandling av pasientene (3). Hurlen kobler tiltak for pleie og behandling til undersøkelse for å fastsette eller bekrefte diagnose, dvs. til legens arbeid. Tiltak er da forskjellige former for målinger, som resulterer i funn som brukes i diagnose-stilling og beslutninger om pleie og behandling (s. 57). Andre tiltak er behandling, f.eks. medikamentell behandling (s. 61). I Arbeidsark-systemet vil noen av tiltakene være gitt i form av at diagnose er gitt, dvs. ved at sykepleierne knytter visse behandlingstiltak til bestemte diagnoser. I tillegg

¹¹ Hurlen (1986) sier ss. 44 - 45:

"Alternativet må være å benytte en nummerserie for entydig identifikasjon. ..

Fødselsnummeret er det eneste som kan benyttes for å identifisere informasjon om pasienter og de fleste andre personer i helsevesenet. De aller fleste vil nemlig ha et fødselsnummer. Unntak er nyfødte og utenlandske statsborgere som ikke er bosatt i Norge."

Hurlen foreslår altså fødselsnummeret som identifikasjon, og bemerker at det i nesten alle tilfelle er stabilt for en person (unntatt er sjeldne kjønnsskifter). Folk skifter navn relativt oftere. Enda mer ustabilt er en persons adresse, der offisiell adresse er den adressen som er registrert i folkeregisteret, mens personens reelle adresse kan være en annen. Dette gjør entydigheten i identifikasjonen tvilsom.

¹² Å bruke seng som første identifikator forutsetter muligheten at pasientene ligger i sengene sine. Vi må også bemerke at sykepleierne til daglig ikke bruker hele fødselsdatoen, men bare fødselsåret.

¹³ Se Hurlen (1986) ss. 54-55.

¹⁴ Hurlen (1986) s. 59:

".. det i dag eksisterer internasjonale sykdomsklassifikasjoner med tilhørende kodeverk .. Disse er alment akseptert og brukt. Det faller derfor naturlig for helsepersonell å beskrive et problem ved hjelp av en diagnosekode."

finnes kategoriene *undersøkelse* og *intravenøs* som skal gi sykepleierne informasjon om arbeidsoppgaver knyttet til pasienten. Disse kategoriene er, som diagnose, tekst-felter på ca. 60 tegn.

Pleie-administrative data: data om helsearbeidere og andre ressurser.

Hurlens modell inneholder selvsagt mye informasjon om personalet på sykehusene; modellen skal bl.a. kunne brukes i administrasjonen av sykehuset. Arbeidsark-systemet skal gi informasjon om hvilke personer som er tilstede på en vakt. I tillegg skal det gi opplysninger om arbeidsoppgaver og fordelingen av disse. I Arbeidsark-systemets datamodell er pleieadministrative data en tekst på ca. 250 tegn, av teknisk-praktiske årsaker delt noe opp. Dette er svært forskjellig fra Hurlens betraktninger om pleieadministrasjon.

For identifikasjon av helsepersonell trekker Hurlen fram helsepersonellnummeret, et offisielt nummer som tildeles autorisert helsepersonell (s. 45). Nummeret angir ikke personens profesjon, men brukes bare for leger, tannleger og veterinærer (!). Det sier seg selv at denne administrative tilnærmingen til identifikasjon ikke er så aktuell for Arbeidsark-systemet.

Hurlens behandling av administrative data er ikke forskjellig fra administrasjon andre steder enn sykehus. Det synet som ligger til grunn for Arbeidsark-systemets datamodell ser på sykepleiernes administrasjon (og data nødvendig for administrasjon) som integrert i sykepleiernes arbeid. Fordi administrative beslutninger og tolkninger av administrative data dermed blir sykepleiefaglig, overlater vi til sykepleierne å lage oversikten og oppdelingen av de administrative opplysningene.

Vi har valgt å la dataene ha en lav struktureringsgrad, noe vi har hatt mulighet til fordi det er snakk om små datamengder. Vi har definert postene som så store felter at de skal kunne brukes som felter av variabel lengde. Den videre strukturering og standardisering av dataene vil sykepleierne gjøre selv etter hva de finner praktisk i bruken av systemet.

Bruksområdet for et edb-system gir føringer til datamodellen i systemet.

Sammenlikningen mellom de to modellene viser at hensikten med modellene, dvs. hva de skal brukes til og hvem som skal bruke dem, påvirker hvilke data som betraktes som relevante. Dette påvirker også tilsynelatende tekniske valg, fordi datamodellen for et edb-system bestemmer databasen i systemet: hvilke data skal være mulig å ha i systemet, og på hvilke måter er det mulig å bruke dem. Det er *før* vi starter med å sette ord på hvilke kategorier av data som skal være viktige i systemet vårt, at vi må tenke over hva systemet skal brukes til.

Oppdelingen i kategorier av informasjon ser enkel og tilforlatelig ut. Det viser seg at denne oppdelingen ikke er så uproblematisk som det virker. Vi skal illustrere dette ved å ta for oss noe så prosaisk som en seng.

I Arbeidsark-systemet er det en kobling mellom sengene på posten og organiseringen av arbeidet, i form av en kobling mellom en gruppe senger og et team med sykepleiere. Dette avspeiler den faktiske planleggingen og organiseringen av sykepleiernes arbeid som foregår på den avdelingen Arbeidsark-systemet er laget til.

Hos Hurlen, derimot, er senger materiell ved sykehuset. I tillegg er sengenummer en slags adresse som forteller hvor på sykehuset pasienten befinner seg. Derfor kan en seng også kategoriseres som pasientadministrative data.

Både Hurlens modell og Arbeidsark-systemets modell tar utgangspunkt i "den verden helsearbeiderne og pasientene opplever" (s. 26). Vi har ovenfor sett at verden kan se forskjellig ut til tross for dette. Hensikten med en datamodell påvirker hvordan modellen blir. Tilsvarende vil også en eksisterende modell ha et innebygget syn på hva som

er viktig og relevant i en organisasjon. Dette vil også påvirke hva modellen kan brukes til.

5. VALG AV PROGRAMVARE OG ERFARINGER FRA PROGRAMMERINGEN

Teknologien vi hadde til rådighet påvirket det produktet vi laget og de valgene vi måtte ta. Generelt gjelder det at eksisterende teknologi begrenser valgmulighetene i teknisk design. I dette avsnittet eksemplifiserer vi dette med erfaringer fra implementeringen av Arbeidsark-systemet, og vi refererer til databaser, filer og programmer i dette systemet.

Maskin- og programvare:

På SYD 5 har vi hatt følgende maskin- og programvare:
NORD 100 Compact, SIBAS E10, NOTIS-RG versjon B
På Ifi har vi brukt:
NORD 570, SIBAS F02, NOTIS-RG versjon C

SIBAS er databasesystemet, og NOTIS-RG er rapportgeneratoren vi har brukt. Vi har brukt SINTRAN III VSX versjon K (operativsystem), NOTIS-WP-M (tekst-behandlingssystem) og UNIQUE versjon B (applikasjongsgenerator) begge steder.

Vi har hatt tilgang til mer programvare, for eksempel har vi hatt hele "DIALOGUE"-pakken fra ND. Foruten SIBAS og UNIQUE består denne pakken av BRUKERMIL-JØ, som brukes for å lage grensesnitt, ACCESS, som er et spørrespråk mot databasen og ABM, som brukes for å lage transaksjonssystemer. I tillegg har vi hatt skjermhåndteringsspråket FOCUS, som kan brukes sammen med SIBAS og programmeringsspråk som COBOL, FORTRAN og PLANC.

Vi valgte å bruke et databasesystem for å slippe å programmere sikkerhetsrutinene nødvendige for å sikre korrekte data i flerbruker-systemet ved hjelp av COBOL.

Dermed følte vi at vi reelt sett hadde følgende valgmuligheter:

For skjermbilder:

SIBAS og 1) FOCUS og COBOL eller 2) UNIQUE

For rapport:

SIBAS og 1) COBOL eller 2) NOTIS-RG / RP

Ettersom vi visste lite om NDs produkter før vi satte i gang, presenterte vi det vi skulle lage for forskjellige personer i ND, og de fleste anbefalte oss å bruke UNIQUE framfor FOCUS og COBOL - av effektivitetsgrunner. Derfor valgte vi SIBAS, UNIQUE, NOTIS-RG / RP i håp om at programmeringen skulle gå raskt.

I databasesystemer kan vi bare representere det som kan uttrykkes ved tegn.

Vi valgte å bruke SIBAS databasesystem som har innebygget sikkerhetskontroller etc. slik at det er rimelig enkelt å lage sikre flerbrukersystemer. Fordi vi har en enkel data-struktur, har vi ikke utnyttet alle mulighetene som ligger i SIBAS.

Ulempen ved å bruke et databasesystem er nettopp at vi bare kan lagre "data", dvs tegn. Dette skaper et problem når vi anvender databaser i sykehus-verdenen. Der kan "mangel" på data være en viktig informasjon. F.eks. er det en viktig informasjon at pasienter ikke har allergiske reaksjoner på antibiotika etc. I termer av databaser er dette et tomt felt, eller "ikke-informasjon", som ikke skal lagres og som helst ikke bør forekomme. Tilsvarende kan det være viktig at det ikke er en relasjon mellom to filer. For eksempel kan en pasient være innlagt, og da er det en relasjon mellom en pasient og en seng. Men det fins relevant pasientinformasjon om pasienter som ikke er innlagt,

akkurat som at det er relevant at det ikke ligger en pasient i en seng.

Denne forutsetningen i databasesystemet kan føre til mye omstendlig programmering, særlig fordi applikasjongsgeneratoren UNIQUE ikke gir muligheter for å lage programmer som bruker "ikke-informasjon". Vi hadde planer om å lage poster som ikke skulle inneholde "egentlig" informasjon, som skulle brukes til å sikre at vi fikk avbildet det vi anså som vesentlig. Med en applikasjongsgenerator er det vanskelig å skjule - og sikre - slike poster for brukerne, og det er en nokså lite elegant løsning. Heldigvis slapp vi å gjøre det.

Applikasjonsgeneratorer har implisitt algoritmestruktur som det er vanskelig å overskride.

Vi brukte UNIQUE for å slippe tidkrevende programmering mot databasen. UNIQUE er en del av DIALOGUE-pakken, som blir markedsført som NDs "familie av database-systemer". I følge ND er UNIQUE et 4. generasjonsspråk, og "et verktøy for utvikling av applikasjoner. Det kan brukes til å utvikle skjermbilder og spesifisere transaksjoner direkte på skjermen. Det sparer ca. 90% av utviklingstiden og av vedlikeholdsressursene"¹⁵. Alternativet hadde vært å bruke COBOL sammen med skjermhåndterings-språket FOCUS. Dette hadde gitt oss større muligheter med hensyn til valg av algoritme-struktur, men hadde kostet mer med hensyn til programmering mot databasen.

Forholdet mellom å bruke applikasjongsgeneratorer og "vanlige" programmeringsspråk krever muligens en forklaring.

Applikasjonsgeneratorer er laget på bakgrunn av en vanlig type anvendelser av edb-systemer, og for å forenkle produksjonen av slike systemer. Vi kan kalle denne anvendelsen "edb-systemer for saksbehandlere". Det er systemer som brukes av én person av gangen, dvs. hver saksbehandler bruker sin egen terminal. Oppgavene som blir gjort er typisk registrering og framfinning. Framfinningen kan f.eks. dreie seg om mange forskjellige opplysninger om en kunde, slik at informasjon som er knyttet til Marianne Johansen lett kan finnes fram fra alle filer / registre. I mer tekniske termer: søking i mange forskjellige registre, men bare med én nøkkel. En skal bare hente én post fra hver fil. Den andre bruksmåten er å skaffe seg en enkel oversikt. Hvis kunden f.eks. heter Johansen, er det fint å kunne få ut en oversikt over alle kundene som heter Johansen, og finne ut hvem en egentlig leter etter. En oversikt svarer til en sekvensiell listing av poster i en fil, gjerne innenfor et avgrenset søkeområde.

Applikasjonsgeneratorene er "verktøy" som forenkler arbeidet med å lage slike programmer / systemer. De er blitt laget fordi noen har sett at det blir laget utrolig mange programmer / systemer av denne typen. Applikasjongsgeneratoren er en form for skjelett-program. Programmene som ligger i bunnen håndterer databasen (inn / ut / sjekking ved kall), de kan brukes til å søke i flere filer på en nøkkel, eller til å få ut en sekvensiell liste over flere poster i samme fil. Det er enkelt å bruke applikasjongsgeneratorer hvis det systemet en ønsker å lage har samme algoritmestruktur som skjelettet i applikasjongsgeneratoren. Og det er vanskelig å programmere med applikasjongsgeneratorer hvis den algoritmestrukturen vi ønsker oss er annerledes enn skjelettet.

Nå er det jo ingen tvang å bruke applikasjongsgenerator hvis en mener at de ikke passer til oppgaven - det vil alltid være informatikerens ansvar å velge de verktøyene som passer til oppgaven. Men samtidig må informatikeren forholde seg til krav om effektivitet. Siden alle oppgaver kan løses på mange måter, vil det være fristende å strukturere oppgaven så den passer til et effektivt verktøy. På denne måten kan applikasjons-

¹⁵ Dette sitatet og følgende sitater i dette punktet er hentet fra NDs manualer for UNIQUE og NOTIS RG / RP.

generatorer bidra til en standardisering av edb-anvendelser¹⁶.

Kuvaja (1987) sier at programmerings-effektiviteten øker ved hjelp av applikasjons-generatorer. Dette gjelder spesielt for uerfarne programmerere. Dermed kan slike verktøy bidra til en spesialisering innenfor edb-yrkene: de som er erfarne programmerere og kan velge mellom vanlige programmeringsspråk og applikasjonsgeneratorer, og de uerfarne, som programmerer forttere ved hjelp av verktøyene, og som kanskje aldri får anledning til å bli erfarne programmerere fordi det ikke "lønner seg". Hvis applikasjonsgeneratorer bidrar til en spesialisering av edb-yrkene, må vi regne med at denne spesialiseringen faller sammen med kjønnsdeling av arbeidet. Cockburn (1985) hevder at kvinner bruker og menn utvikler og vedlikeholder teknologi. Foreløpig er det ingen grunn til å tro at dette ikke også skulle gjelde programmeringsspråk.

Problemer underveis i applikasjons-programmeringen.

Vi ønsket å lage objekt-orienterte skjermbilder i Arbeidsark-systemet: vi ønsket å ha informasjon knyttet til en seng som et objekt på skjermen, der vi i hvert objekt presenterte dataene under hverandre. I oversiktsbildet for overvåkningsrommene ønsket vi å skrive ut informasjon om flere pasienter, men ikke sekvensielt (se figur 6). Til å lage disse skjermbildene brukte vi applikasjonsgeneratoren UNIQUE. I UNIQUE blir variablene deklart etterhvert som de brukes, og siden det vanligvis bare blir lest fra én post pr. register, betyr det at det gjelder å fange alle interessante felt fra en post før neste post leses. Vi simulerte objektorienteringen ved å bruke REPEAT, dvs. lese en post, legge dataene i "riktige" hjelpevariable, og lese igjen, til alle postene / objektene var plassert i hjelpevariable. Deretter flyttet vi dataene fra hjelpevariablene til det stedet de faktisk var deklart på skjermbildet. (Rein COBOL-programmering!)

Det er videre umulig å lese flere poster fra samme fil uten å bruke REPEAT. Dette gjorde at vi måtte endre struktur på FLYTTing av pasienter. I utgangspunktet tok vi utgangspunkt i en PASIENT, og gikk derfra til SENGeregisteret for å finne ut hvor pasienten lå. Ved flytting måtte en først sjekke SENGeregisteret, og eventuelt gjøre noe med den pasienten som lå der (dvs. 2 oppslag i både seng og pasient-fil). Så det gikk ikke, og det ville i tilfelle ha ført med seg en del problemer i forhold til datastrukturen, f.eks. hvordan tomme senger og pasienter skulle representeres.

Vi måtte derfor lage en ny algoritmestruktur. Der starter vi med oppslag i SENGefila, hvor også pasientidentifikasjon er lagret. Deretter slår vi opp i PASIENTfila for å finne den nye pasienten som skal legges inn, evt. endre utskrivningsdato (statusdato) for den pasienten som flyttes ut. Vi greier oss med ett oppslag i SENG og med ett i PASIENT-fila.

Algoritmestrukturen i UNIQUE-programmene er delvis implisitt, og delvis knyttet til utseendet på skjermbildene. Dette knytter programstruktur til layouten på skjermbildet. Det er for eksempel vanskelig å utvide "inneteams" arbeidsområde med 1 seng, fordi det betyr endringer i layout både på skjermbilde og i rapport. Team, som begrep, er knyttet til den fysiske utformingen og overvåkningsteknologien på avdelingen, både i avdelingen og i Arbeidsark-systemet. Et annet problem som oppstod når vi forsøkte å lure oss rundt den implisitte algoritmestrukturen, var at vi fylte UNIQUEs buffer til randen. Alle våre skjulte hjelpevariable tar utrolig mye plass, og gjør at programmet går tregt. Så rent teknisk er det ikke mange bokstaver til det er plass til på dette skjermbildet. Og hvis vi hadde fått plass, hadde kanskje systemet gått så tregt at det ikke hadde vært å anbefale av den grunn.

På grunn av algoritmestrukturen har Arbeidsark-systemet blitt mindre fleksibelt og

¹⁶ Se SYDPOL (1987) ss. 62-63.

generelt enn det vi opprinnelig hadde tenkt oss. For eksempel vil det kreve mye omprogrammering for å organisere pasient-informasjonen etter andre prinsipper, som f.eks. etter legens ansvar, eller i henhold til sykepleieransvar. I et Arbeidsark-system som skal kunne overføres til andre poster og avdelinger, må det være flere generelle prinsipper for å organisere informasjonen enn det er nå. I det nåværende Arbeidsark-systemet er noe generelt og noe spesielt, men vi kan ikke skille ut det generelle fra bare dette ene systemet. Nyten av Arbeidsark-systemet vil alltid være avhengig av at sammenstillingen av informasjon passer til organiseringen av arbeidet på det stedet hvor edb-systemet brukes. Et generelt system må nok derfor programmeres i et "vanlig" programmeringsspråk, slik at det er rom for lokale tilpasninger.

Det generelle i systemet er sammenstilling av pasientinformasjon og informasjon om ansvar og arbeidsoppgaver. Den konkrete utformingen vil være avhengig av arbeidsplassen og arbeidsorganiseringen der.

Rapportgeneratorer har også implisitt algoritmestruktur som det kan være vanskelig å gå rundt.

Rapportgeneratoren vi har brukt er NOTIS-RG. "Med RG kan man på en enkel måte definere avanserte rapporter ved å tegne det ønskede utseendet på skjermen". En rapportgenerator er laget etter samme prinsipp som applikasjongsgeneratorer: I administrativ databehandling er det typisk masse-programmering av en "klasse" med rapporter. De kan f.eks. skrive ut alle kunder og hva de har kjøpt siste kvartal i forhold til varegrupper. Dette skal være sortert etter geografisk distrikt, og deretter alfabetisk på kundenavn. Det skal typisk skrives ut noe oppsummering etter hver kunde, etter hvert distrikt og når hele rapporten er ferdig. Algoritmen som ligger i bunnen er typisk en sekvensiell utskrift av poster i ett / flere registre, hvor hver forekomst som skrives ut, har samme utseende. Og når det blir såkalt brudd, dvs. ny kunde eller nytt distrikt, markeres dette i rapporten ved en "særbehandling". Men den er også regelmessig og symmetrisk. Med NOTIS-RG er det en smal sak å lage slike rapporter. Men hvis en ønsker en rapport med en mer utradisjonell layout, får en problemer. Et annet problem med NOTIS-RG er at det også er "saksbehandler-orientert". Det gjør det vanskelig å lage en god dokumentasjon av hvordan de forskjellige parametrene settes, siden det er meningen at hver enkelt person som har behov for rapporter bare lager dem, og ferdig med det.

I det følgende avsnittet bruker vi betegnelsen "rapport" på en rapport som produseres ved hjelp av NOTIS-RG, dvs. vi bruker deres rapport-begrep. Produktet som lages, dvs. et ferdig Arbeidsark, kaller vi "dokument". Dette dokumentet blir ofte kalt rapport andre steder, for å skille mellom skjermbilder og papirrapporter. Årsaken til begrepsbruken i dette avsnittet er at et dokument kan bestå av flere rapporter.

Den rapportstrukturen som ligger til grunn for NOTIS-RG er forskjellig fra de dokumentene vi ønsket å lage. Dette har ført til triksing og rett og slett grisete programmering fra vår side.

2 av våre 3 dokumenter er ikke "symmetriske", dvs. de kan ikke lages ved at et såkalt detaljområde repeteres N ganger. Men alle dokumentene kan deles opp i flere rapporter, slik at del-rapportene blir symmetriske. Derfor er et dokument satt sammen av flere rapporter, og alle rapportene skrives ut på samme dokument-fil. NOTIS-RG lager sideskift mellom hver rapport. Dette skaper et bruks-problem: En må gå inn i den tekstfila som dokumentet ligger på, og fjerne sideskiftene mellom rapportene, før en skriver ut dokumentet. Det gjør det med andre ord tungvint å produsere det ferdige dokumentet. Vårt tilfelle av grisete programmering skyldes at rapporter vanligvis legges etter hverandre, de kan ikke skrives ut ved siden av hverandre.

Dette skapte problemer når vi skulle lage et Arbeidsark for inneteamet (se figur 2). I den øverste delen skal vi skrive ut informasjon om 4 senger og de pasientene som

ligger der, ved siden av hverandre. Dette lar seg gjøre ved å lage et detaljbilde over en seng med tilhørende pasient, og si at den skal repeteres 4 ganger på samme linje. Resultatet legges ut på en tekstfil.

Den nederste delen av rapporten er verre. Der skal vi skrive ut informasjon i et A-B-A-mønster, der A svarer til detaljbildet i den første rapporten, og B svarer til team-opplysninger. Problemet oppstår fordi en ikke kan skrive ut forskjellige detaljområder på samme linje. Vi hadde tenkt å lage en tom TEAMpost, som skulle skrives ut på den nederste delen av rapporten og bli klippet bort manuelt. Det er én grisete løsning. Vi valgte en annen. Vi fant ut at vi kunne utnytte at NOTIS-WP har atskilt editering og formattering. Det gjorde vi ved å lage et detaljområde med mønster A-B og repetere dette to ganger på en linje. Dette blir så lagt ut på en tekst-fil. Deretter setter vi standard-utskriftsformat på fila slik at bare A-B-A blir skrevet ut. Vi utnytter med andre ord at tekstfila har en bredde på ca. 200 tegn. Når vi sier at bredden på utskrift er 132 tegn, blir bare de første 132 tegnene pr. linje skrevet ut. Fordi formatteringen av fila må gjøres "manuelt" (med en eksplisitt kommando), blir de resterende 68 tegnene ikke flyttet til neste linje eller noe sånt, de forblir "skjult".

Forslaget ND hadde til løsning av vårt A-B-A-problem, var at hvert eneste felt i rapporten måtte defineres som et navngitt felt og deretter kalles opp og skrives ut som "gruppeslutt". Vi syntes dette hørtes tungvint ut, og vi har ikke prøvd det. Vi kan derfor ikke si om det virker.

NOTIS-RG gjorde også at vi måtte lage en kopling mellom TEAM og SENG vi ikke hadde planlagt. Årsaken er at det må finnes en forbindelse mellom alle de "strømmene" (filene) som brukes i en rapport. Så da måtte vi lage det for å få skrevet ut team-opplysninger og pasient-opplysninger på samme rapport.

LITTERATUR.

Litteratur-referansene er delt i 2 deler. Først refererer vi litteratur vi har brukt i rapporten, deretter gir vi en oversikt over litteratur som er laget av Florence-prosjektets medarbeidere, og/eller handler om Florence-prosjektet, gjennom tidene.

1. LITTERATUR-REFERANSER I RAPPORTEN.

- Balas, G. et. al. (1982): Faglig Kvinnepolitikk - hvor går LO?, Pax forlag, Oslo
- Bjerknes, G. & Bratteteig, T. (1984): "The Application Perspective - An Other Way of Conceiving Edp-based Systems and Systems Development" i Sääksjärvi, M. (red.): Report of the Seventh Scandinavian Research Seminar on Systemengineering, Helsinki 1984
- Berger, U. & Offe, C. (1981): "Das Rationalisierungsdilemma der Angestelltenarbeit", i Kocka (red.): Angestellte im europäischen Vergleich, Vandenhoeck & Ruprecht, Göttingen
- Borum, F. & Enderud, H. (1981): Konflikter i organisationer Nyt Nordisk Forlag, København
- Bødker, S. (1987): "Prototyping revisited - design with users in a cooperative setting", kommer i Järvinen, P. (red.): Report of the 10th Information systems Research seminar In Scandinavia, Universitetet i Tampere
- Bøgh Andersen, P. & Halskov Madsen, K. (1986): Metaforer i design og brug af EDB-systemer, arbeidsnotat i SYDPOLs arbeidsgruppe 2
- Cockburn, C. (1983): Brothers. Male Dominance and Technological Change Pluto Press, London
- Cockburn, C. (1985): Machinery of Dominance Pluto Press, London
- Feldman, M. S. & March, J. G. (1981): "Information in Organizations as Signals and Symbols" ss. 305-324 i Administrative Science Quarterly, Vol. 25
- Foged, B. & Markussen, R. (1984): Det fleksible køn Tiderne skifter, Viborg
- Floyd, C. (1983): "A Systematic Look at Prototyping" i Preceedings for Working Conference on Prototyping, Namur Belgia
- Freidson, E. (1970): Profession of Medicine Harper & Row, New York
- Holmquist, B. & Källgren, G. (1981): Postgirot som språkmiljø, Institut för lingvistik, Stockholms Universitet
- Hurlen, P. (1986): En modell for informasjonsutveksling i helsevesenet., hovedoppgave ved Institutt for Informatikk, Universitetet i Oslo 1986
- Håndlykken, P. (1982): "Bruk av beskrivelser og modeller ved utvikling av edb-systemer", i Fossum, E. (red): EDB og arbeidsliv Tanum-Norli, Oslo
- Josefson, I. (1987) Samtal med sjukskoterskor på St. Thomas Hospital i London våren 1987, upublisert arbeidsnotat
- Kalverkämper, H. (udatert): Die Axiomatik der Fachsprachen-Forschung, udatert notat
- Keen, P. G. W. (1981): "Information Systems and Organizational Change" ss. 24-33 i Communications of the ACM, Vol. 28, No. 1
- Kuvaja, P. (1987): "A literature analysis of the production of application generators", kommer i Järvinen, P. (red.): Report of the 10th Information systems Research seminar In Scandinavia, Universitetet i Tampere
- Lantz, K. (1986): The Prototyping Methodology, Prentice-Hall, Englewoods Cliffs
- Lanzara, G. F. og Mathiassen, L. (1984): Mapping Situations Within A System Development Project, MARS-rapport no 6, DAIMI PB-179, Århus
- Lyytinen, K. (1986): Information Systems Development as Social Action: Framework and Critical Implications, Jyväskylä Studies in Computer Science, Economics and Statistics 8, University of Jyväskylä
- Martinsen, K. & Wærness, K. (1979): Pleie uten omsorg? Pax Forlag, Oslo
- Microsyster (1986): Computerising women's groups: Microsyster's experiences from April 1984 to March 1986 presentert på IFIP WG9.1 konferanse om "Women, Work and Computerization - Oppurtunities", Dublin 1986

- Munk-Madsen, A. (1984). "Practical Problems of System Development Projects". i Sääksjärvi, M. (red.), Report of The Seventh Scandinavian Research Seminar on Systemeering, Part II, ss. 280-298. Helsinki School of Economics.
- Newman, W. (1983). "Chapter 4: Procedures and Problem Solving". Upubl. notat.
- Perby, M-L. (1987): "Computerization and Skill in Local Weather Forecasting" i Bjerknes, G. et al (red): Computers and Democracy - a Scandinavian Challenge, Gower Publishing Company Ltd, Hampshire, 1987
- Sandewall, E. et al (1982): Stepwise Structuring - A Style of Life for Flexible Software, Rapport LiTH-MAT-R-82-18, Linköping
- SYDPOL (1983): Forskningsprogrammet SYDPOL - SYstem Development environments and Profession Oriented Languages, NORDFORSKs Publikationsserie 1983:2
- SYDPOL (1987): Arbejdsformer sat på dagsordenen, SYDPOL Rapport nr. 1

2. LITTERATUR OM FLORENCE-PROSJEKTET.

Florence-prosjektet:

- Gjensidig læring. Rapport fra fase 1, Oslo 1985
- Status for Florence-prosjektet august 1986 (intern-rapport)
- Status i Florence-prosjektet desember 1986 (intern-rapport)
- Læring som forutsetning for design - rapport nr. 2 fra Florence-prosjektet, Oslo 1987
- Status i Florence-prosjektet mai 1987 (intern-rapport)

I tillegg er Florence-prosjektet beskrevet eller brukt som eksempel i:

Bjerknes, Gro & Bratteteig, Tone:

- "The Application Perspective is a Frame for Understanding Development of Computer-based Systems for Nurses" i Lassen, M. og Mathiassen, L. (red): Report of The Eight Scandinavian Research Seminar on Systemeering, Århus 1985
- "Florence in Wonderland" i Bjerknes, G. et al (red): Computers and Democracy - a Scandinavian Challenge, Gower Publishing Company Ltd, Hampshire, 1987
- Female Technologists Working with a Female Dominated Occupation - Opportunities and Alternatives presentert på IFIP WG9.1 konferanse om "Women, Work and Computerization - Opportunities", Dublin 1986

Bjerknes, Gro og Kaasbøll, Jens:

- "Profession Oriented Languages for Nurses" i Nurminen, M og Gaupholm, H (red): Report of the Sixth Scandinavian Research Seminar on Systemeering, Bergen 1983

Einarsrud, Anne Kristin og Granholm, Torill:

- Spesifikasjon og implementasjon av deler av programmoduler for sykepleiere, hovedfagsoppgave ved Institutt for Informatikk, Universitetet i Oslo 1986

Kaasbøll, Jens:

- "Observation of Human Information Processing as a Technique in System Development" i Lassen, M. og Mathiassen, L. (red): Report of The Eight Scandinavian Research Seminar on Systemeering, Århus 1985
- "Intentional Development of Professional Language Through Computerization. A Case Study and Some Theoretical Considerations", i Docherty, P. et al (red) System Design for Human Development and Productivity: Participation and Beyond, North-Holland, IFIP 1987
- "Intentions and Conventions in the Design of Computer Systems for Nursing Practice: A Poster Comparing 8 Designs from the Florence Project", presentert på The Ninth Scandinavian Research Seminar on Use-and-Development of Information Systems, Lund, 1986

- "An AI-Based System in the Development of Nursing Knowledge" kommer i Buchberger, Gøranzon og Nygaard (red) **AI: Perspectives and Implications. Selected Papers from the Vienna Conference on AI and Language, Knowledge and Responsibility**, Oslo 1987

Nygaard, Kristen:

- "User-oriented Languages" i **Medical Informatics Europe**, 1984

Sannes, Inger:

- "Skal edb bli et "hjelpemiddel" for sykepleiere?" i **Sykepleien** nr. 10/84
- **Will Computer Technology lead to more and better nursing? May nurses direct this development?** foredrag ved "International Symposium on Nursing Use of Computers and Information Science", Calgary, Canada mai 1985
- "Bedre sykepleie med EDB?" i **Sykepleien** nr. 10/85

VEDLEGG 1.

UTKAST TIL SAMARBEIDSAVTALE MELLOM INSTITUTT FOR INFORMATIKK, UNIVERSITETET I OSLO, OG SENTRALSYKEHUSET I AKERSHUS.

Dette er et utkast til samarbeidsavtale mellom partene i Florence-prosjektet. Vi har tatt det med her fordi utkastet dannet basis for samarbeidet, og fordi det illustrerer hvilke forventninger som eksisterte i prosjektet høsten 1985, før samarbeidet mellom Institutt for Informatikk og Sentralsykehuset i Akershus kom igang. Det har vært interessant å se hvordan tankene fra den gang er blitt fulgt opp.

29.10.85

Det følgende er utkast til en avtale mellom Sentralsykehuset i Akershus og Florence-prosjektet ved Institutt for informatikk, Universitetet i Oslo. Avtalen omfatter de ulike partenes rettigheter og plikter i forhold til det planlagte samarbeid om Florence-prosjektets fase II og III.

<Selv om mye av teksten kanskje virker bastant og definitiv i formen, betyr det ikke at vi ikke ønsker endringer. Den endelige teksten bør være mest mulig klar og entydig. Derfor har vi valgt denne formen også i første utkastet. Men en avtale er jo alltid en tekst som to parter skal stå inne for, så vi er selvsagt åpne for alle typer endringer og kommentarer.>

Disposisjon

1. Overordnede mål for prosjektet.
2. Prosjektorganisasjon
3. Fordeling av ansvar i forhold til de ulike aktivitetene.
 - Arbeidsbetingelser for prosjektet, ledelse og arbeidsorganisering
 - Utvikling, evaluering og systemspesifikk opplæring
 - Generell opplæring, informasjon, rapporter
4. Utstyr og programmer.
5. Tidsplaner.

1. Overordnede mål for prosjektet

Florence-prosjektet ser det som et viktig prinsipp er at yrkesutøverne selv har kontrollen over innføringen av edb. En av de viktigste forutsetningene for en slik kontroll er at yrkesutøverne er kunnskapsmessig forberedt til å vurdere de teknologiske og organisatoriske løsninger som tilbys, og til å formulere egne ønsker og krav. Florence-prosjektets mål er å utvikle strategier for å sikre sykepleierprofesjonen en slik kunnskapsmessig forberedelse. I tillegg er det et mål for informatikerne å samle kunnskap om alternative system-utviklingsstrategier.

For å utvikle strategier og kunnskap trenger vi konkrete erfaringer fra en systemutvikling som kan forgå under andre og bedre betingelser enn de vanlige. Fordi Florence-prosjektet mottar forskningsmidler fra NTNF, har vi muligheter til prioritere områder som ofte må lide kortsiktig produktrettede systemutviklinger. Gjennomarbeidede undervisningsopplegg om edb som setter yrkesutøvernes behov i sentrum er et slikt område. Alt for ofte blir brukere presentert for generell edb-undervisning som ikke har noe med deres arbeidssituasjon å gjøre.

Florence-prosjektet ønsker videre å bidra til utforming av undervisning for sykepleierstudenter, men først og fremst vil vi undervise de sykepleierne vi skal samarbeide med. Prosjektets andre hovedområde er å lage programmeringsverktøy og andre hjelpemidler for systemutvikling med tanke på at sykepleiere selv skal kunne lage og endre edb-løsninger som de finner nyttige for deres yrkesutøvelse. Undervisning og tilgjengelig teknologi må utvikles slik at edb blir et redskap som sykepleiere og annet helsepersonell opplever at de aktivt kan forme og bruke for å bedre kvaliteten og effektiviteten i den pleie og omsorg som blir gitt.

Florence-prosjektet betrakter profesjonene som viktige forvaltere av kunnskap fordi de kontrollerer utdanningssiden, og fordi de organiserer helsearbeiderne på tvers av det den enkelte institusjon. Hvis ikke profesjonene tar opp den utfordringen som datateknologien representerer, kan de stå i fare for at en stadig viktigere del av de kunnskapene som brukes i yrkesutøvelsen blir kontrollert av edb-firmaer. Vi mener det er nødvendig å bruke profesjons-organisasjonene som ramme for å bygge opp en strategi blant helsearbeidere som er slagkraftig nok til å kunne ivareta de faglige utfordringer som vil følge av edb-teknologiens muligheter og farer. En utstrakt bruk av edb vil drastisk kunne endre den arbeidssituasjonen som de ulike profesjonene har bygget opp.

2. Prosjektorganisasjon

Hovedstruktur og ressurser:

Florenceprosjektet mottar sine viktigste driftsmidler fra NTNF (Norges Teknisk Naturvitenskapelige Forskningsråd). Disse midlene blir kanalisert til Florence-prosjektet gjennom Institutt for informatikk ved Universitetet i Oslo. Universitetet i Oslo bidrar også direkte ved å lønne flere av de vitenskapelige ansatte som bruker forskningsdelen av sin stilling til arbeid med prosjektet. Florence-prosjektet er et prosjekt innen det nordiske SYDPOL-programmet, og deltakere fra Florence deltar i nordiske arbeidgruppemøter hvor prosjektet blir diskutert.

Selv om prosjektet mottar støtte fra flere hold kan vi for denne avtalens vedkommende betrakte Institutt for informatikk (IfI) ved Universitetet i Oslo som den ene part og Sentralsykehuset i Akershus (SiA) som den andre.

Institutt for informatikk skal inngå en avtale med Norsk Data og Rogalandsdata for å skaffe det nødvendige utstyr og programmer for prosjektperioden. I tillegg til avtalen IfI-ND/RD, som er begrenset til samarbeidsprosjektets varighet, bør SiA inngå en avtale med ND om betingelsene for at SiA kan bruke/overta utstyret etter at Florenceprosjektet er avsluttet.

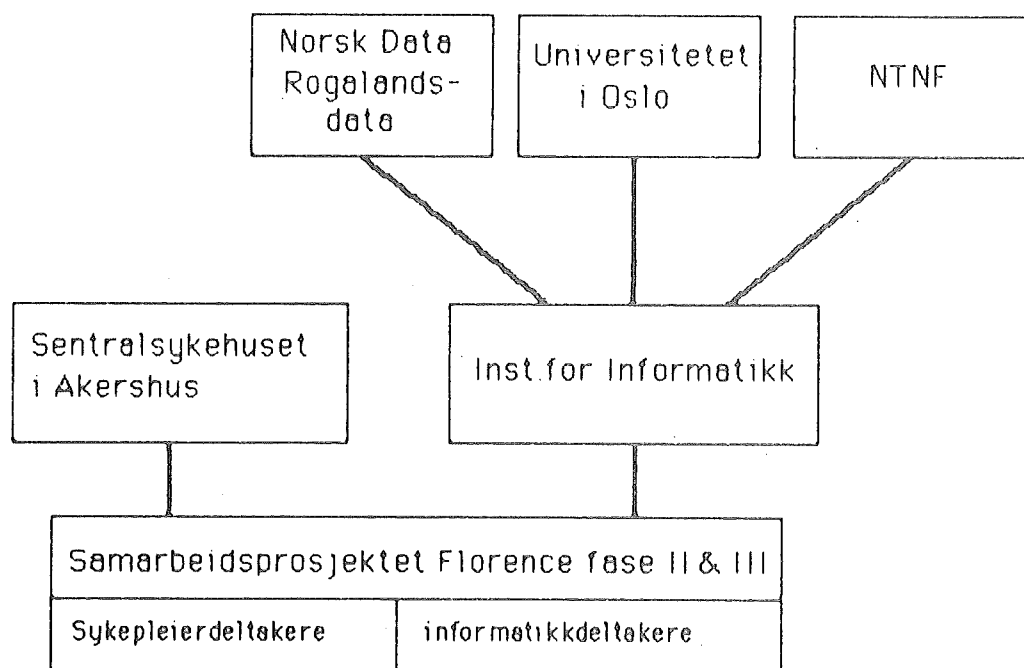


Fig. 1 Partene i avtalen og de viktigste ressurskildene.

NTNF kommer til å støtte prosjektet ut 1986, som er den tidrammen denne avtalen er begrenset av. I løpet av denne tiden kommer tre forskere til å ha Florence-prosjektet som hovedarbeidsområde i sin forskningstid: Gro Bjerknes (informatiker), Tone Bratteteig (informatiker og daglig leder av prosjektet) og Henrik Sinding-Larsen (sosialantropolog). Kristen Nygaard (professor i informatikk) er faglig ansvarlig overfor prosjektets bevilgende myndigheter). Tre hovedfagsstudenter vil skrive hovedoppgave i tilknytning til prosjektet. De blir veiledet av Jens Kaasbøll (informatiker).

Det daglige arbeidet skal utføres av **prosjektgruppa** som består av informatikkdeltakere og sykepleierdeltakere. Både sykepleiersiden og informatikksiden skal velge en person som kan fungere som kontaktperson og daglig leder for prosjektaktivitetene i de respektive miljøer.

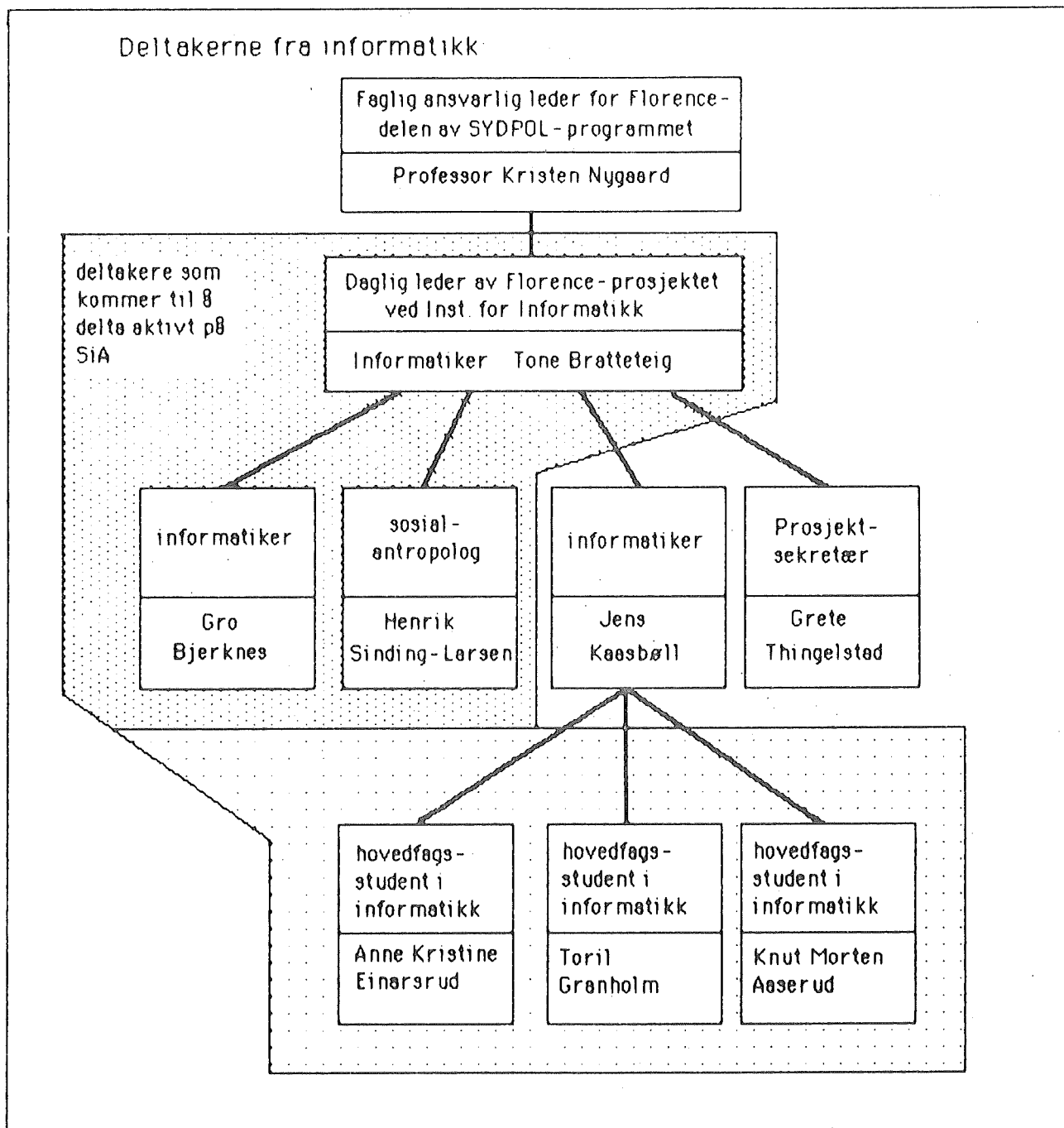


fig. 2 Deltakerne fra informatikk

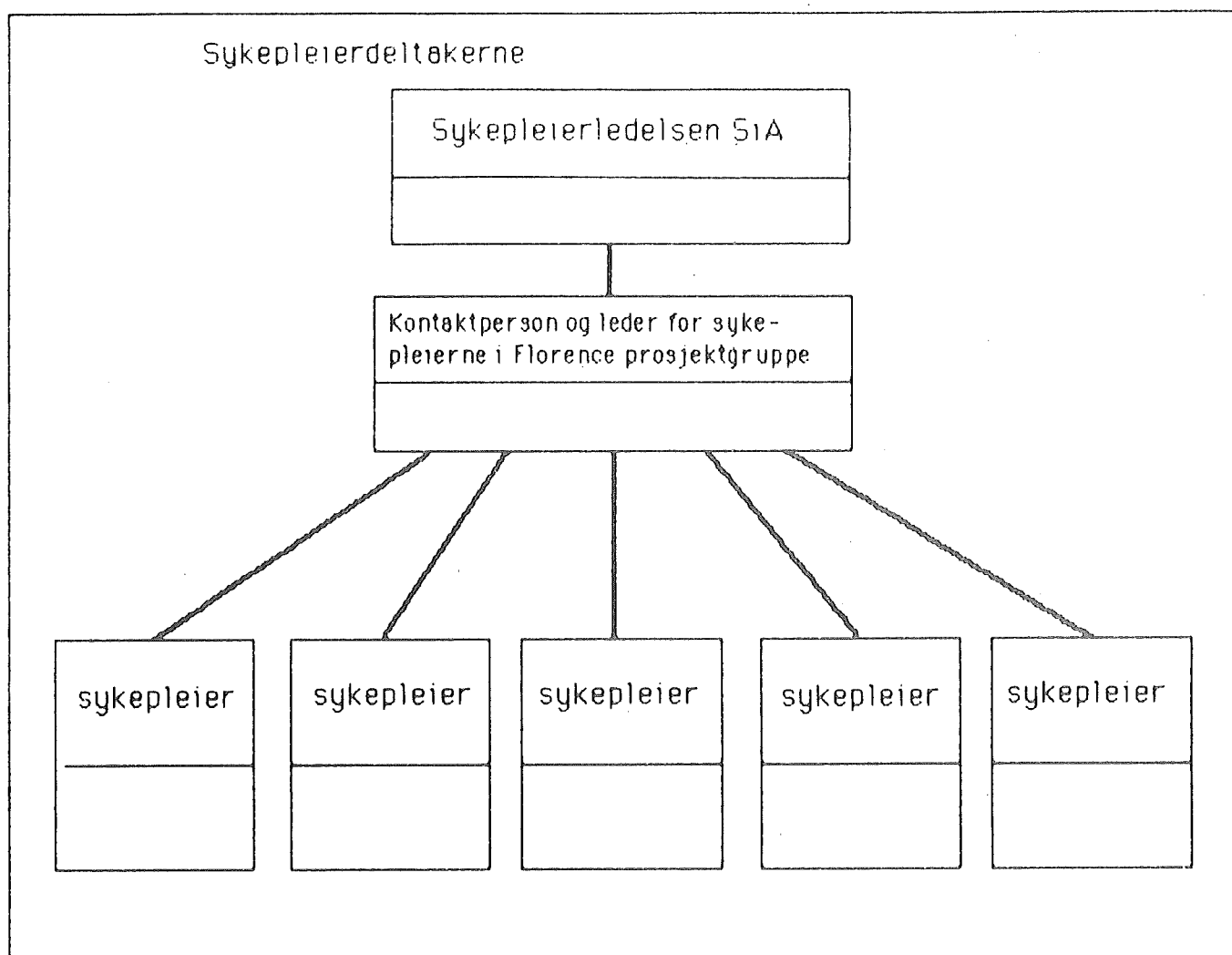


fig 3. Sykepleierdeltakerne

Fordi prosjektet vil ha konsekvenser for en rekke ulike instanser skal det opprettes en følgegruppe som skal overvåke prosjektet. Den skal også være et forum for kommunikasjon mellom de partene som blir berørt av prosjektet.

Følgegruppa er ikke en styringsgruppe som skal ha ansvaret for direkte styring og planlegging av prosjektgruppas arbeid.

Gruppas funksjoner er altså: 1) å gi råd og støtte til prosjektgruppa slik at arbeidet blir konstruktivt og ikke kommer i konflikt med de berørte parter, 2) å være et formelt forum for kommunikasjon fra prosjektgruppa til de berørte parter mens prosjektet pågår og 3) organisere (og representere) de ulike partene til felles innsats når det er behov for dette (f.eks. for å søke om støtte og midler til prosjektet).

Et forslag til følgegruppas sammensetning er den følgende:

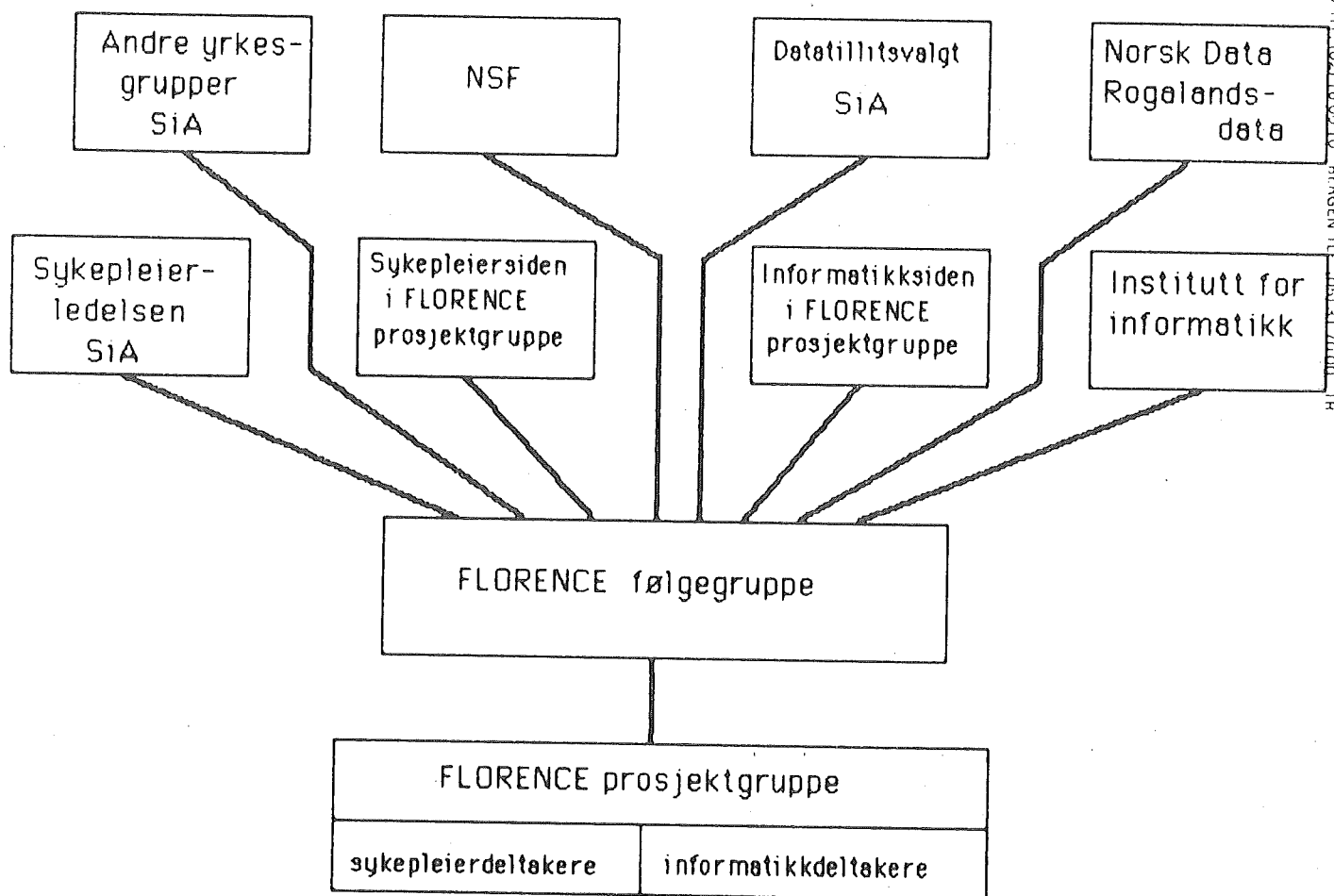


fig. 4

Følgegruppa skal være konstituert før prosjektet starter arbeidet ved den aktuelle posten.

3. Fordeling av ansvar for de ulike aktivitetene.

Arbeidsbetingelser for prosjektet.

For å avlaste avdelingen i prosjektperioden, er det nødvendig med en ekstra sykepleierstilling på avdelingen. Institusjonene har i fellesskap ansvar for å skaffe denne stillingen, mens SiA har ansvaret for å administrere den. Den tiden avdelingen får til disposisjon gjennom den ekstra stillingen, skal brukes til prosjektaktiviteter av de sykehusansatte som er med på prosjektet.

Informatikkdeltakerne må ha muligheter til å gjøre seg kjent på avdelingen, både ved å følge sykepleierne på vakter og ved å kunne "intervjue" sykepleierne om deres arbeid.

Hvis andre enn sykepleiere blir direkte berørt av de edb-systemene vi lager, kan prosjektgruppa trekke representanter for disse yrkeskategoriene inn i prosjektarbeidet.

Ansvaret for aktiviteter og framdrift av prosjektet er delt mellom institusjonene. En person fra hver institusjon har leder- og kontaktansvar.

Utvikling og evaluering av edb-systemet.

Både sykepleierne og informatikerne i prosjektgruppa har ansvaret for å utvikle edb-systemet. Informatikerne har et spesielt ansvar for at det blir utviklet en prototype, og for at det blir utviklet programmeringsverktøy som sykepleierne kan bruke til å endre senere versjoner av systemet. Sykepleierne har et spesielt ansvar for å evaluere edb-systemet, og for videre tilpasninger av edb-systemet på avdelingen. I tillegg har informatikerne ansvar for å skaffe edb-utstyr til forsøksvirksomheten.

Opplæring og informasjon.

Informatikerne har ansvar for å gi sykepleierne i prosjektgruppa den tilstrekkelige opplæringen for å delta i utviklingen og utprøving av edb-systemet. Dette skal omfatte både generell og mer systemspesifikk edb-opplæring. Den delen av den generelle opplæringen som egner seg til det skal være åpen for andre interesserte ved SiA. Sykepleierdeltakerne har ansvar for å lære opp de andre på posten/avdelingen i å bruke et evt ferdig system, men informatikerne skal bistå med å lage et evt opplæringsopplegg.

Prosjektgruppa som helhet har ansvar for å tilrettelegge og gi informasjon til resten av sykehuset, og prosjektgruppa har også ansvaret for å skrive rapportene fra prosjektet.

Sykepleierdeltakerne i prosjektet har ansvaret for å gi informasjon om prosjektet til de andre ansatte (sykepleiere? andre yrkesgrupper?) på avdelingen (posten? skolen? sykehuset?).

4. Utstyr og programmer.

Institutt for informatikk ved Florence-prosjektet har ansvar for utstyr og programmer så lenge forsøksvirksomheten pågår. For videre bruk av programmer og utstyr, se avtalen mellom IfI og RD/ND.

5. Forslag til tidsplan:

Fase 2:

- 1/11 - 31/12-85: Informatikerne besøker posten. 2 dagers undervisningsseminar for sykepleierne på posten. Åpent orienteringsmøte på SiA. Installasjon av dataanlegg på posten.
- 1/1 - 28/2-86: Periode for endring og utprøving av edb-basert kardex. Evt samordning med utvikling av edb-system for pleietyngderegistrering.
- 1/3 - 31/3-86: Evaluering av fase 2.
Rapport fra fase 2.
Beslutning om hovedaktivitet i fase 3.

Fase 3:

- 1/4 - 30/6-86: Eksperimentell systemutvikling.
Utvikling av programmeringsverktøy for sykepleiere.
Opplæring i bruk av verktøyet.
- 1/7 - 30/9-86: Forberedelse til eventuell drift:
Videre opplæring og muligheter for forandring av systemet.
- 1/10 - 31/10-86: Rapport fra fase 3.
- 1/11 - 31/12-86: Evaluering av prosjektet som helhet.
Sluttrapporter.
Evt. etterundersøkelse i juli -87.

Telefon: (02) 70 22 00 - Adr.: 1474 Nordbyhagen

Institutt for Informatikk
v/professor Kristen Nygaard
Blindern
Universitetet i Oslo
0313 OSLO 3

Nordbyhagen, 7. november 1985

IS/th

Vår ref

Deres ref :

PROSJEKT OM UTVIKLING OG BRUK AV EDB I SYKEPLEIE "FLORENCEPROSJEKTET"

Vi viser til henvendelse 11.09., orienteringsmøte 7.10. og konferanse om kontraktutforming 1.11.85.

Etter at prosjektforslaget har vært drøftet med berørte parter i sykehuset, stiller vi oss positivt til samarbeid på de premisser som er skissert.

Cardiologisk avdeling er innstilt som prosjektsted og forbereder seg til at arbeidet kan komme i gang etter den foreslåtte tidsplan, det vil si november/desember 1985.

Bakgrunnen for at vi har valgt cardiologisk avdeling er stabilitet og faglig sikkerhet i sykepleiegruppen. Det er et godt samarbeid mellom yrkesgruppene, slik at de programmene som utvikles vil både preges av og vurderes med henblikk på hele pleie- og behandlingstilbudet pasienten får.

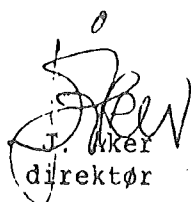
Det er allerede i gang arbeid for å finne frem til gode instrumenter for hvordan ressursene forvaltes internt, gjennom driftsplanleggingsprosjektet. Eksempelvis har avdelingssykepleier fra januar 1985 fortløpende foretatt en sammenligning mellom pleietyngde/behov og personalressurser.

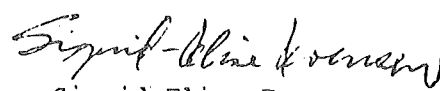
PREMISSER FOR PROSJEKTET

1. Sykehuset kan ikke selv stille utstyr eller programmeringshjelp til disposisjon for prosjektet, og det er derfor en forutsetning at dette er en del av "prosjektpakken" utenfra.
2. Prosjektperioden bør etter vår oppfatning være ut 1987, slik at vi får en realistisk evaluering.
Kontrakt om utstys- og programmeringshjelp må sees i lys av dette. Det er behov for avklaring i eget møte mellom SiA, Institutt for Informatikk, Norsk Data/ - og Rogalandsdata så snart som mulig.
3. Sykehuset ønsker å stå fritt med hensyn til eventuell overdragelse av utstyr etter at prosjektperioden er over.
Betingelser for eventuell overtagelse ønskes likevel avklart før prosjektet kommer i gang. Sykehuset må derfor delta i de aktuelle forhandlingene.

4. Det vil være nødvendig å tilføre en ekstra sykepleierstilling til den posten hvor prosjektet skal gjennomføres. Det forutsettes at lønnsmidler for stillingen blir dekket innen Florenceprosjektet.
5. En vil arbeide for at erfaringene fra Florenceprosjektet får ringvirkninger utenfor cardiologisk avdeling. Det vil derfor bli tilrettelagt målrettet informasjon, undervisning, demonstrasjon, seminar o.l. også for sykehusets datautvalg og andre aktuelle innen sykepleietjenesten og sykehuset for øvrig. En forutsetter at Institutt for Informatikk bistår med slike opplegg i nødvendig grad.

Vi ser frem til et fruktbart samarbeid gjennom Florenceprosjektet, og er forberedt på at arbeidet kommer igang så snart som mulig. Vi ber om å bli kontaktet vedrørende utarbeidelse av nødvendige kontrakter i forhold til premissene og for å få en avklaring/kontrakt om prosjektets gjennomføring.


J. Aker
direktør


Sigrid-Elise Evensen
sjefsykepleier

VEDLEGG 2.

SAMARBEID MED MASKINLEVERANDØR.

I dette vedlegget vil vi beskrive vår opplevelse av ett aspekt av systemutviklingen: vårt forhold til maskinleverandøren i prosjektet. Vi skal gjengi vår historie om samarbeidet med Norsk Data, siden den etter vår mening beskriver mange vanlige problemer i systemutvikling. Fortellingen er i store trekk hentet fra "Status i Florence-prosjektet, mai 1987".

SAMARBEIDET MED LEVERANDØREN - VED 2 AV BRUKERNE

Beskrivelse av programmering og implementering av Arbeidsark-systemet.

Dette er historien om hvordan organisasjonsstrukturen i Norsk Data (ND) har påvirket og forsinket vårt arbeid. Vi har "døpt om" de fleste av personene vi refererer til, fordi det er ND som organisasjon vi ønsker å si noe om. Alle de vi har truffet har vært svært hyggelige og hjelpsomme. Og vi tror historien ville vært den samme uansett hvilken leverandør vi hadde vært i kontakt med, fordi vi ville hatt samme type ikke-standard problemer. Det er feil-situasjonene som har gjort at vi har sett NDs organisasjonsstruktur, for vanlige kunder med vanlige feil fungerer den sikkert bra og er derfor ikke synlig.

Problemer med NORD 570.

Vi fikk programvare til NORD 570 den 17/12 -86. 19/12 begynte vi å installere programvaren vi hadde fått: NOTIS WP, NOTIS DS og SIBAS, og gjorde forsøk på å installere UNIQUE. Vi fikk ikke til å installere UNIQUE.

Først trodde vi at vi hadde satt feil CPU-nummer, altså at vi hadde gitt feil kode under installeringen. Det var ikke tilfelle: feilen viste seg å være at det var blitt satt et feil bit da SINTRAN ble installert, altså før "vår" tid. Da det ble rettet virket kopieringsbeskyttelsesprogrammet Software-keys, og vi kunne lese inn UNIQUE.

Men da vi leste inn UNIQUE-diskettene viste det seg at 3 av de 4 diskettene var like, og ca. halvparten av programmene manglet. Vi fikk med andre ord ikke installert UNIQUE da heller. Vi fikk tilsendt nye disketter, men denne gangen hadde vi fått disketter med kompilierungsfeil i programmene.

Da forbarmet noen fra ND seg over oss, og Espen kom for å ordne opp. Og vi skulle få siste versjon av produktet UNIQUE (versjon B). Samtidig skulle de installere den samme UNIQUE-versjonen på SiA. Men det fikk de ikke til. Rettere sagt: de fikk installert, men de kunne for lite om UNIQUE til å prøve det nok til å finne ut at det ikke virket - noe vi etterhvert fant ut.

Først 24/2 fikk vi installert UNIQUE på SiA sammen med Marianne fra ND. Det manglet noen filer for at det skulle være mulig å kjøre. (Det samme skjedde oss tidlig høsten 1986 da programmene til produktet BRUKERMILJØ ikke virket. Også da ble programvaren installert av en person som ikke kunne nok til å teste om det var gjort riktig og som ikke sjekket at alle filene var på plass. Ukjente med ND-utstyret som vi var brukte vi litt tid på å finne ut at det faktisk var NDs "feil" og ikke manglende kompetanse fra vår side.)

Vi satte igang med programmering av Arbeidsark-systemet og laget en enkel database med tilhørende UNIQUE-programmer. Vi fikk raskt problemer fordi vår oppgave krevde at vi måtte gå ut over den typen oppgaver UNIQUE er laget for å ta seg av. Vi tilkalte hjelp fra ND, og 13/3 kom Marianne hit for å hjelpe oss med UNIQUE-problemene våre. Hun tipset oss om noen flere tommelfinger-regler, og mente vi burde kontakte firmaet Unique med våre ikke-standard-problemer.

Marianne skulle samtidig installere NOTIS RG/RP slik at vi også kunne få skriftlige rapporter med data fra databasen vår. Det fikk hun ikke til: NOTIS RG/RP virket ikke til tross for at innlesingen hadde gått helt bra. 24/3 kom hun tilbake sammen med Espen. Sammen gjentok de installerings-prosedyren, men det samme skjedde. Innlesingen gikk bra, men NOTIS RG/ RP virket ikke. En uforklarlig feil. Vi muntret dem opp med å fortelle om andre morsomme og uforklarlige ting ved maskina.

Samtidig med at NDs folk strevde med rare feil på maskina, hadde vi kontakt med Unique både i Oslo og Sandnes om programmerings-problemene våre. Vi lærte mye av det, og vi skal komme tilbake til det litt senere. Etter hvert som programmeringen skred fram og vi laget større og annerledes programmer, oppsto nye feil. Ca. 24/3 oppsto en særlig alvorlig og uforklarlig feil som gjorde at vi bare sporadisk fikk tilgang til databasen, og som følgelig gjorde programmering mot databasen vanskeligere. Vi meldte derfor umiddelbart fra til Service-avdelingen i ND om denne nye feilen (siden det etter vår mening dreide seg om en ND-feil, og ikke en UNIQUE-feil). Det var snakk om 2 alvorlige feil som hindret oss i videre programmering. Den ene var at NOTIS RG/ RP ikke virket: RG virket av og til mens RP aldri hadde gått. Den andre feilen oppsto under noen bestemte UNIQUE-kjøringer, der vi gikk ut av programmet (i en såkalt ND-500-TRAP (altså en felle!)).

6/4 kom en person fra NDs Service-avdeling for å ordne opp i dette. Men hun var fersk og kunne mindre enn de andre, og tilbrakte derfor 3 timer med å vente på telefoner fra NDs mer erfarne eksperter. Så skrev hun opp feilmeldingene vi hadde fått, og som vi allerede hadde rapportert, leste inn NOTIS RG/RP og gikk. Samme dag fikk vi telefon fra Jan, som svarte på henvendelse fra 1/4. Vi hadde fått oppgitt Jan som leder av "vår" service-gruppe. Han er kjent som en svært dyktig person - som ikke alltid tar telefonen sin. Vi orienterte da om feilen(e) og om Service-personen som hadde vært og sett på feilmeldingene tidligere på dagen. Han skulle snakke med henne og de andre "kjent-folkene" på ND.

Både vi og våre kontakter syntes etterhvert at det var plagsomt mye feil med vår maskin. NDs Øyvind Fjell tok derfor initiativet til et møte om organiseringen av kontakten mellom oss og ND.

7/4 møtte representanter fra Florence- og Scala-prosjektene på Universitetet NDs representanter Øyvind Fjell, Espen og Marianne for å snakke om dette. Vi syntes alle det var synd at ingen fra NDs planlagte Universitets-service-gruppe var tilstede. ND lovt å "forbedre seg", og ville jobbe for å forbedre sitt rykte hos Universitetets informatikk-miljø.

På møtet diskuterte vi blant annet mulige forklaringer på feilene på maskina, og vi kom fram til at hardware-feil var en mulighet som burde sjekkes.

Vi kom tilbake fra seminar i Danmark mandagen etter, den 13/4. Da var det mer feil enn noen sinne. Datoen på maskina var satt tilbake til 15.mai 1986 (SINTRANS installerings-dato), noe som skjedde med ujevne mellomrom. Vi fikk vite av vår (Universitetets) systemansvarlige at vi hadde fått Telefix + modem, og i tillegg var det kommet noen nye filer på "systemområdet". Vi ringte følgelig til Jan, som var inne. Han fortalte oss via telefonen hvordan vi skulle få databasen opp igjen.

Men feilene var der fremdeles! Så vi ringte sporenstreks tilbake til Jan siden vi hadde truffet ham inne (for en gangs skyld). Overveldet av iveren vår lovt han å komme og hjelpe oss allerede dagen etter. Vi anbefalte da diverse "kjentfolk" han kunne snakke med, som han tydeligvis gjorde. Han ringte dagen etter, 14/4, og sa at det nok kanskje var en hardware-feil. (Denne "diagnosen" ble som vi husker foreslått på møtet 7/4, der bla. vår "kjentmann" Espen var tilstede.) Jan sa videre at han ikke hadde tid til å komme før påske, så det skulle komme noen for å kjøre Diagnostics på maskina første natta etter påske.

Nils, fra vår service-gruppe, kom 21/4. Men den samme dagen, før han kom, ringte også en annen person fra ND som gjerne ville ordne opp med NOTIS RG/RP. Vi ba henne

snakke med de andre som skulle hjelpe oss slik at de kanskje kunne samordne framstøtene...

22/4 litt ut på dagen dukket Nils opp igjen fra kjelleren der maskina vår står. Han sa at hardwaren var utmerket - ND hadde jo skifta ut nesten hele da maskina brant ifjor vår, og da hadde de sjekka og dobbelt-sjekka alt. Han forsto ikke hva som var årsaken til våre feil, og mente de ikke skyldtes hardware. Han rapporterte at:

"Kjørte 7 harde 500-tester over natta (ca. 20 timer). Fikk en feil i Cobol-kompilering, men klarte ikke å få den til å feile på nytt."

Nils mente med andre ord vi sto ovenfor en software feil, som han ikke kunne hjelpe oss med. Men han lovte å snakke med Jan om det, og trodde nok ikke det var noen vei utenom at han måtte komme.

"Feilen i Unique oppstår etter et bestemt mønster - 2. gang et bestemt register kalles opp. Samme adresse hver gang."

Verken databasens struktur eller programmene kunne forklare feilen.

Vi måtte sette oss ned og ventet på lyd fra Jan - som alle mente nå måtte komme, og som *helt sikkert* kunne hjelpe oss. Det hadde tatt oss noen uker å forsere noen av hindringene som er lagt ut for å hindre at Jan reiser ut i utide. Vår framgang kunne vi (for å si det litt grovt) måle ved å se på kunnskapsnivået hos de som kom for å se på feilene. Vi forserte en "skanse" pr. uke.

Men Jan kom ikke denne gangen heller. Derimot ringte Nils og sa at Jan hadde sagt at det ville hjelpe hvis vi fikk en ny versjon av SIBAS, nemlig versjon F02. For det hadde vært så mye tøys med både F01 og F00 - som vi altså hadde. Nils var noe skeptisk, men han mente det i hvert fall ville si oss om det var SIBAS-feil eller ikke. Vi var også noe skeptiske.

Jan hadde videre gitt Lise, som tok kontakt for å reparere på NOTIS RG/RP, "lov" til å komme til oss da det ble klart at det ikke var en hardware-feil vi hadde. Lise kom, installerte NOTIS RG/RP til en forandring, sa at programmet var i orden, og gikk. Men det var fortsatt ikke mulig å bruke NOTIS RG/RP.

24/4 fikk vi noen SIBAS-disketter fra Nils. Vi fikk ikke til å installere den nye SIBAS-versjonen. Vi ringte først til Nils, som ikke var inne, og deretter til Jan. Jan ønsket tydeligvis ikke å gå utenfor huset, for han ble glad og lovte å hjelpe oss med en gang, da vi fortalte at vi hadde Telefix. Etter litt fram og tilbake fikk han kobla seg på, så da overlot vi saken til Jan.

Mandag morgen (27/4) ringte Nils. Da virket SIBAS, men alle feilene var der fortsatt. Så da visste vi at det ikke var SIBAS-versjonen vår det var noe i veien med.

Nils fortsatte feil-søkingen via modem, og triksa på bit-nivå for å få NOTIS RG/RP til å gå. Om natta kjørte de ny hardware-test. De fikk ingen feil-meldinger, men våre feil var der jo fremdeles. Nils hadde likevel mistanke om at det kunne være noe med hardware, nærmere bestemt TRAP-kortet. Han kom og skiftet det - og da virket NOTIS RG/RP og vi fikk installert nye programmer.

Men feilene vi fikk under UNIQUE-kjøringer var der fremdeles. Nils sa han trodde han hadde vært borti noe liknende en gang med FICS. Da hadde det vært FICS det var noe feil med. Han tok kopi av hele disken, og dro.

Vi ble nysgjerrige på om det faktisk var UNIQUE som var årsak til feilene våre. Vi flyttet rundt på programmer og bruksområder på maskina for å sjekke om vi fikk feil på samme sted. Det gjorde vi. Vi opprettet alle data-filene på nytt (i tilfelle vi hadde dratt med noe grums i kopieringen), men fikk fortsatt samme feil. Vi fortsatte (også dagen etter) med å gjøre noen endringer i databasen og i programmene, og fant ut at det er vår UNIQUE-versjon sin håndtering av felt, spesielt lange felt, som gir oss feil.

Vi ringte og fortalte om denne oppdagelsen til Nils, som sa at han ikke kunne noe om UNIQUE. Han lovte derfor å snakke med Jan om dette.

Jan sendte problemet videre til Bjarne (29/4).

For oss gikk livet videre med NOTIS RG/RP som viste seg å inneholde en del begrensninger, men ikke verre enn at vi mente vi skulle klare å lure oss rundt de fleste av dem. En support-gruppe på ND ble engasjert i saken, men vi fant ut at vår egen triksing med reglene var enklest og lettest å leve med¹.

Problemer med UNIQUE.

Problemene i historien vår dreier seg som nevnt ovenfor også om programmeringen i UNIQUE. Vi har valgt å skille ut disse problemene fordi UNIQUE-problemene ikke skyldes maskin-feil, men at programmering av Arbeidsark-systemet beveger seg langs grensene på produktet. Vi skal også ta disse problemene kronologisk.

Vi ervervet våre basis-kunnskaper i UNIQUE på kurs hos ND. Etter å ha brukt disse kunnskapene i ca. 2 uker sto vi fast. Vi fikk hjelp av Marianne i ND, som bidro med sine erfaringer i bruk av produktet. Vi kom et stykke videre, men sto snart fast igjen. Det ble klart for oss at Arbeidsark-systemet ikke var av den typen oppgaver UNIQUE er laget for å ta seg av. 13/3 fikk vi Mariannes tips om å kontakte firmaet Unique for å få hjelp.

Etter noen forsøk fikk vi svar hos Unique i Sandnes (20/3). Problemet vårt var å presentere flere pasienters data samtidig, ved siden av hverandre. I Unique mente de at det vi skulle gjøre, var umulig. Men vi var uenige i det, og det viste seg å være mulig å simulere "vinduer" med UNIQUE.

Underveis støtte vi på diverse problemer i form av grenser i UNIQUE, og vi hadde tidvis tett kontakt med Unique i Oslo. Siste uke i mars ringte vi spesielt ofte fordi vi fikk en feilmelding som verken vi eller Unique i Oslo forsto. Unique Oslo kontaktet Unique Sandnes, der de som har laget UNIQUE sitter. Etter en ukes tid hadde Oslo-kontoret fått svar fra Sandnes om at det hørtes umulig ut. De foreslo at vi skulle sende en utskrift av programmet til Sandnes og forklare hva som gikk galt. Det gjorde vi den 7/4.

Og vi fikk svar tilbake 13/4 fra Eirik Simonsen (vi har hørt at det han ikke vet om UNIQUE er ikke verdt å vite). Vi skal sitere litt fra brevet:

"Som nevnt er kilden til problemet at dere prøver å få til noe som ON-LINE ikke skal kunne klare. ...

.. Belastningen på CPU vil være enorm i forhold til et standardprogram. ...

.. Jeg er forundret over den standhaftighet sluttbrukergruppen deres viser. Uansett hvilket verktøy som brukes i utvikling av informasjonssystemer så trengs en balanse mellom de muligheter og behov som finnes.

Min vurdering er at ON-LINE ikke kan klare å realisere den skjermbildelayout (dvs. 6 senger) som sluttbrukeres vil ha. Konkret betyr dette at utseende må endres hvis ON-LINE skal brukes til å vise aktuelle data."

Til tross for stadige tilpasninger til UNIQUEs forutsetninger om problem-spesifikasjon, så det ut til at vi måtte gi avkall på å lage den skjermbilde-presentasjon av dataene som vi ønsket. Vi forenklet oversikts-skjermbildet ved å legge data om arbeidsorganiseringen på et annet, eget skjermbilde. Med disse endringene virket oversikts-bildet. Men vi fikk fortsatt samme feilmelding (ND-500-TRAP) når vi prøvde å nå registeret med data om arbeidsorganiseringen ("team-registeret").

Det var flere regelmessige uregelmessigheter som ikke forsvant med nytt TRAP-kort. Vi fikk (som nevnt ovenfor) blant annet problemer med felt som er lengre enn 128 tegn(!), og vi fant ut at denne feilen ikke var knyttet til ett bestemt register, som vi trodde i utgangspunktet, men at det var generelt. Vi ringte til Petter i Unique, Oslo. Han sa at det ikke var noen SIBAS-restriksjoner på felt-lengder når det gjaldt UNIQUE, og han skjønnte heller ikke hvorfor deler av feltene ble repetert. Han syntes vi hadde en tung data-struktur, og det var vi enige i.

¹ Hvordan vi løste problemene med NOTIS RG/RP står beskrevet i kap. III.5.

På grunn av forenklingen i oversikts-skjermbildet kunne vi forenkle strukturen i team-registret. O UNDER! feilen forsvant. Dermed ble en feilmelding erstattet med en annen: "Protect violation" ble byttet ut med "feil etter 128. tegn, karakteren på slutten av 1. linje i et oppdelt felt blir blank".

Ivrig som vi var, tok vi med oss programmene til SiA, installerte dem og prøvde om feilene var der også. De var det. Vi konkluderte at UNIQUE versjon B03 hadde underlige feil. Vi mente at dette ikke var heldig dersom dataene handlet om "ekte" hjerte-pasienter. Da vi ringte for å mase på Bjarne sa han seg enig i dette, og han ville undersøke i ND om noen kjente til denne feilen.

Jan overlot, som vi husker, problemet til Bjarne 29/4. Bjarne hadde sendt den videre til en Support-gruppe. Denne Support-gruppa er svært hemmelig, og det er visst ikke så mange i denne gruppa som har UNIQUE-kompetanse. Vi foreslo for Bjarne at vi kunne slutte å mase på ham og heller begynne å mase på noen i Support-gruppa. Men Bjarne mente Support-gruppa skulle slippe mas fra oss. Det er få andre i ND som vet om denne Support-gruppa. Marianne fortalte oss at hun pleier å ringe til Unique, Oslo, når hun har problemer. Og på NDs UNIQUE-kurs lærte vi også at vi burde kontakte Unique hvis vi hadde problemer.

Da en uke var gått, og ingen fra ND hadde ringt for å intervju oss nærmere om feilene, ringte vi til Unique i Oslo, 6/4. Der traff vi Morten. Vi fortalte ham at vi hadde snakket med ND, men at det virket som om de hadde litt liten brukserfaring, så nå følte vi at vi måtte ringe til dem. Morten sa at ND hadde like mye supportansvar for UNIQUE som for NOTIS-produktene, men var enig i at ND kanskje kunne hatt bedre kompetanse på UNIQUE. Han intevjuet oss straks om feilene og om datastrukturen og lovet å ringe igjen. Vi lovet på vår side at vi skulle få de som jobbet med saken på ND til å kontakte ham.

7/4 kom Bjarne fra ND for å se på programmene våre etc., og vi fortalte at han måtte ringe til Morten, så de kunne samordne framstøtene. 8/4 ringte vi til Morten. Da hadde han ikke snakket med Bjarne, men han hadde forslag til hva som kunne være feil. Han mente bla. at output-felt i UNIQUE har en øvre grense på 127 tegn!

Men fortsatt var det uoppklarte feil, og han lovet å jobbe videre med saken. Vi ringte sporenstreks til Bjarne for å fortelle om dette. Han fortalte på sin side at han skulle på kurs mesteparten av uka etter, så vi kunne bare fortsette å mase... Morten ringte igjen 11/4, og kom med enda et forslag til problemløsning. Og denne gangen virket det! Glade og lykkelige som vi var, ringte vi til Jan for å fortelle denne gode nyheten. Det virket ikke som han ble noe glad i det hele tatt. Så vi repeterte de gode nyhetene til Bjarne da han ringte dagen etter for å fortelle oss om stoda.

Feilene skyldtes vår programmering i UNIQUE. Problemene dreide seg om å dele lange felt i flere mindre felt for skjerm-presentasjon av dataene. Det finnes 2 måter å behandle lange felt i biter på: grovt sagt setter den ene måten bitene sammen til et felt, mens den andre lar feltet bestå av biter. Manualen er uklar på valg-kriterier for den ene eller andre måten, og vi valgte altså den gale. En årsak til at vi ikke forsto hva som var feil var at det nesten virket, og at vi hadde fått feil på programmer som brukte den "riktige" måten før (men da trolig av andre årsaker). Men den viktigste årsaken var at vi ikke visste nok om hvordan UNIQUE som programmeringsspråk håndterer data.

For å repetere: feilen skyldtes ikke maskina eller UNIQUE, men en semantisk feil i programmeringen. Vi hadde rett og slett brukt primitivene galt. For oss er det første gang vi har hatt problemer med semantikken i programmeringsspråk, og vi tror det skyldes at vi tidligere har lært hvordan de programmeringsspråkene vi bruker, er implementert.

Forholdet mellom ND og Unique.

Ett av prosjektets problemer har vært at vi bruker et produkt som selges av ND, men som lages av noen andre. Dette har til tider gjort oss engstelige, og med god grunn, har det vist seg.

Så lenge de feilene vi hadde var maskinfeil, var ansvarsforholdet klart. Da vi (med god hjelp av Nils) fant ut at det var UNIQUE-feil - på begge ND-maskinene, ble ansvarsforholdet mer uklart. Skulle vi si ifra til begge service-gruppene? Ville de greie å samordne framstøtetene? Og ville de samarbeide med Unique? Og hvem hadde ansvar for oss som prosjekt, med to forskjellige maskiner? Vi ringte rundt til forskjellige personer, og de fleste mente at ND ihverfall hadde et formelt ansvar, siden vi hadde fått produktet gjennom dem. Og da vi endelig fikk tak i Jan, sa han at han hadde ansvaret for at feilen ble rettet. Vi pustet lettet ut, for da visste vi ihvertfall hvem vi kunne mase på.

Det virker som om kontakten mellom ND og Unique er nokså dårlig. Det virker ikke som om de bruker hverandre for å finne løsninger på feil som blir meldt. Selv internt i ND ringer folk til Unique når de har problemer. Antakeligvis vil både ND og Unique tjene på et nærmere samarbeid. Problemet er vel bare å få det i gang. Samtidig kan det kanskje være en fordel at samarbeidet ikke er så godt. For oss har det vært en fordel å kunne ringe til Unique.

Unique er et relativt lite firma hvor de fleste kjenner hverandre. Det gjør at problemløsning går raskt, fordi folk vet om hverandre og om hverandres kunnskaper. Og det er UNIQUE de har, og trenger å ha, kompetanse på. Forøvrig vil vi si at Uniques dyktige og entusiastiske personer har hatt en usvikelig tro på produktets fortreffelighet, og det har bidratt til å holde motet oppe hos oss i sorte stunder.

ND er for det første en mye større organisasjon enn Unique og trenger i tillegg en mye bredere kompetanse. De trenger kompetanse både for å finne ut feilene dreier seg om UNIQUE og til å rette dem opp. Siden få i ND har erfaring med bruk av UNIQUE, tar det lang tid å løse ikke-trivielle problemer med dette produktet.

NDs organisasjon sett i forhold til forsknings-prosjektet Florence.

Til tross for at feilene med NORD 570 har bidratt til betydelige forsinkelser, har vi lært mye av samarbeidet med ND, og spesielt har vi lært ting om ND vi neppe kunne ha lært på noen annen måte. NDs organisasjon er blitt tydelig fordi vi har hatt uvanlige feil. Da vil det reelle kompetanshierarkiet bli synlig.

Vi tolker problemene vi har opplevd i samarbeidet med ND, dithen at ND har et internt kommunikasjonsproblem og et kompetanseproblem. Begge disse typene problemer blir gjort til våre problemer, fordi vi på sett og vis faller utenfor den organisasjons-strukturen det er lagt opp til i ND.

Vi faller utenfor den opptrukne strukturen på flere måter:

- Vi har avtale om at produktene skal leveres ferdig installert. Det betyr at vi har sortert under salgsavdelingen store deler av prosjektperioden, og delvis under serviceavdelingen (siden det tok lang tid til å installere NOTIS RG/RP).
- Vi er en "kunde" med liten ND-kompetanse, men med gode generelle kunnskaper. Vi har derfor behov for kontakt med ND-folk med bred og generell erfaring.
- I tillegg løser vi problemer som gjør at vi beveger oss på grensa av hva produktene kan tåle. Dette fører selvsagt til feilmeldinger som vi ikke kan klassifisere, selv om vi ofte har meninger om hva slags feil det dreier seg om.

ND er ikke organisert på en måte som gjør vi effektivt får svar på våre spørsmål og/eller hjelp til å rette opp feilene våre. Salgsavdelingen tar seg av konfigurasjon og installasjon. Når dette er unnagjort overtar service-avdelingen ansvaret. Vår Universitets-maskin sorterer under én service-gruppe pga. CPU-nummeret, mens maskina på SiA sorterer under en annen service-gruppe. Dette er et problem på grunn av rutinen for melding og løsning av feil i ND.

Feilmeldinger fra kunder skal rettes til Service generelt, som så skal "rute" dem videre. Hvis de som sitter på Service ikke greier å "rette" feilen, blir melding sendt til service-gruppa. Hver service-gruppe har en leder, og er forøvrig delt opp i en hardware og en

software gruppe. Feilen blir "distribuert" internt i gruppa ved at "noen" foreslår en mulig løsning på problemet. Med litt flaks blir feilen rettet ved første forsøk. Hvis ikke, foreslår "noen" andre mulige løsninger. Med nye forslag kommer det inn nye personer, med stadig økende kompetanse, og atter nye forslag til mulige feil-"diagnoser". Hvis det viser seg at feilen ligger i et produkt, blir support-gruppa for dette produktet kontaktet, av service-gruppa. Tilslutt blir kanskje feilen rettet. Men det er mange skanser å trenge igjennom, og det tar sin tid.

Service er den delen av ND som kundene skal kontakte, for at spesialiseringen internt i ND ikke skal bli kundenes problem. Men personene på Service har ikke alltid kompetanse nok til å klassifisere feil, slik at det ofte kan ta litt tid før noen skjønner feilmeldingene, enn si gjør noe med dem. Det tar tid for folk å finne ut at de kan for lite til å løse et problem og så sende det videre til en som kan mer.

NDs organisasjon - og dermed deres forutsetninger om hvordan kunder er - gjør at det ikke er så ofte noen er interessert i våre meninger om hva som er feil med maskina.

Vårt prosjekt har hatt problemer som skyldes manglende ND-kompetanse i fagmiljøet vårt. Noe av årsaken til dette ligger i at vi ikke har hatt anledning til å gå på ukeslange kurs i behandling av ND-produkter pga. undervisningsbelastningen ved Universitetet. Vi kunne f.eks. ønsket oss en kort introduksjon til NDs datamaskin med alle dens forskjellige produkter, som ikke tok opp emner nødvendige for ikke-databehandlere.

Vi mener at det tar lang tid å få rettet feil fordi kompetansenivået i ND generelt er for lavt. Spisskompetansen vernes, og dette bidrar til å danne og vedlikeholde et kompetanse-hierarki i ND. Jo mer en person kan, desto vanskeligere er det å få snakke med henne, siden det er så mange lag i pyramiden som verner henne mot "påtrengende" brukere. I tillegg føles det som om alle ansatte blir "voktet" av sekretærene slik at det er vanskelig å få telefonkontakt hvis en ikke kjenner det direkte telefonnummeret. (Som de fleste andre, ringer nemlig heller ikke personer fra ND tilbake med en gang de mottar en lapp om det.) Vi synes selvsagt det er flott at personer med høy kompetanse kan delegerer ansvar og oppgaver. Men etter en stund blir vi lei av at det alltid er vår tid det går utover, f.eks. når 4 forskjellige personer fra ND har skrevet ned den samme feilmeldingen uten at feilen er blitt rettet noe på. Riktignok betyr det at vi vet mer om hva som ikke er feil, men vi synes vår telefoniske rapport kunne greie seg.

Generelt kan en si at ND er organisert etter funksjonalitet - og spesialitet. Dette fører til kommunikasjonsproblemer. For ND oppstår problemene fordi spesialiseringen krever kommunikasjon på tvers av organisasjonen når "ubestemte" feil skal rettes, og fordi det ikke fins noen formelle kanaler som støtter opp under slik kommunikasjon. For oss opptrer problemet ved at vi ikke vet hvem vi skal ringe til - og det tar ofte litt lang tid å gå "tjenestevei". Det er ikke lett å vite hvem som kan svare på hva, så en er avhengig av personlige kontakter, personer som igjen kjenner personer i ND. Dette er en privat - og trolig nokså vanlig - løsning på NDs interne kommunikasjonsproblem. Vi har hatt kontakt med flere deler av ND samtidig, men de vet ikke om hverandre, så ting blir ofte dårlig koordinert.

NDs service-apparat er ikke tilpasset de behov for bistand som naturlig vil oppstå i et forsknings-prosjekt. Forskning i informatikk vil i stor grad si å tilpasse bruk av edb-teknologi til nye, uprøvde bruksområder eller nye bruksmåter innenfor kjente felt. Grensene som eksisterer i teknologien er et resultat av tidligere bruksområders behov og krav, og med nye bruksmåter oppstår nye krav til teknologien. En må regne med å strekke de eksisterende produktene langt, og det bør ikke overraske noen at grensene i teknologien blir påtrengende synlige. På grunn av dette vil forskningsprosjekter ofte befinne seg i grenselandet mellom service og utvikling, service for å finne ut hva som er feil, og utvikling for å finne ut hvilke muligheter for å strekke produktene som fins. Det er mulig at NDs supportgrupper også befinner seg i dette grenselandet. Supportgruppene yter imidlertid service til service-gruppene. Det er med andre ord nokså vanskelig for menige brukere å få kontakt med dem.

Et service-apparat som skal takle "kunder" med disse typene problemer må ha bred, generell kunnskap om teknologien de leverer. Problemene vil stort sett være forbundet med diagnose av feil. Når feil-kilden er lokalisert, eller feilen er klassifisert, vil det vanligvis være enkelt å rette den opp. Først da kan folk med spesialisert kompetanse på NDs forskjellige produkter komme inn i bildet.

Vi vil også bemerke at det, ut fra det vi kan lese av NDs organisasjonsstruktur, ser ut til at det er i service-gruppene og i utviklings-avdelingen de faglige utfordringene ligger. På disse stedene i organisasjonen er det bruk for bred informatikk-kompetanse, på et eller annet nivå. Bredden i kompetanse gjør den mer generell, og den kan derfor brukes som referansebakgrunn for et bredere felt innenfor databehandling enn spesialisert, snever og dyp, kompetanse.

Derfor er det heller ikke overraskende at vi finner få damer disse stedene.

Som en oppsummering vil vi si at for å ha et godt prosjekt-samarbeid med en datamaskin-leverandør er det vår erfaring at det er viktig:

- å ha maskin- og programvare installert og testet. Det er viktig at datamaskinen i utgangspunktet virker som den skal fordi utviklingen av edb-systemet i praksis ikke kan ta til før systemet er i orden. Vi har brukt lang tid på å finne ut om det er programmene i vårt Arbeidsark-system det er noe i veien med, om det er UNIQUE og UNIQUEs grenser, eller om det er NDs maskin- og/eller programvare som er skyld i feilene som oppstår. Vi har hatt tilfeller av alle disse feilkildene, og det har tatt tid og krefter fra oss. Læring ved feiling er svært slitsomt i lengden.
- å ha sanksjonsmuligheter. Fordi ND ikke har hatt økonomiske interesser i prosjektet utover utlånet av utstyr, er den eneste sanksjonsmuligheten vi har hatt i prosjektet nå å gi ND dårlig rykte. Det skader muligens ND på lang sikt, men hjelper ikke oss på kort sikt.
- å ha med en person fra leverandøren som er interessert i prosessen i prosjektet, og som føler seg forpliktet overfor oss, brukerne og produktet. Vår erfaring er at det ikke har vært noen hos ND som har hatt ansvaret for å følge opp prosjektet til daglig, og at NDs organisasjonsstruktur medfører en pulverisering av dette ansvaret. Det har gjort at ting har tatt tid, og at NDs organisasjonsstruktur ble vårt problem. NDs representant i følge-gruppa hadde lite med prosjektet å gjøre.

VEDLEGG 3.

INNHALDSFORTEGNELSE I "DOKUMENTASJON AV ARBEIDSARK-SYSTEMET. RAPPORT NR. 4 FRA FLORENCE-PROSJEKTET".

Vi har valgt å ta med innholdsfortegnelsen i den 4.de rapporten fra prosjektet fordi den også er en del av den avsluttende rapporteringen fra prosjektet. Årsaken til at avslutnings-dokumentasjonen fra prosjektet kommer i 2 bind er at vi ønsker å skille mellom dokumentasjon av prosjektet og av det edb-systemet prosjektet har resultert i. Rapport nr. 4 er først og fremst drifts- og vedlikeholds-dokumentasjon, og er derfor mest interessant for de som skal bruke og vedlikeholde, og evt. videreutvikle, Arbeidsark-systemet.

**DOKUMENTASJON AV ARBEIDSARK-SYSTEMET. RAPPORT NR. 4 FRA
FLORENCE-PROSJEKTET.**

INNHold.

FORORD

INNHold

1. KORT BESKRIVELSE AV ARBEIDSARK-SYSTEMET

Beskrivelse av bruk av systemet

Krav til systemet

2. BRUKERVEILEDNING TIL ARBEIDSARK-SYSTEMET

3. OVERSIKT OVER ARBEIDSARK-SYSTEMET PR. 1/9 -87

Oversikt over systemet og databasen

Datamodell, pasient

Datamodell, seng

Database-definisjon

4. UNIQUE-PROGRAMMER

Oversikt over sammenhengen mellom applikasjonene

Hovedmeny (UNIQUE-II-S-B-N)

OO

OU

OK

OS

PI

PI-IT-RP

PI-UT-RP

FLYTT

TEAM

INIT

5. DOKUMENTER OG RAPPORTER LAGET MED NOTIS RG / RP

Oversikt over dokumenter og rapporter

OO:TEXT

OU:TEXT

OK:TEXT

