

UNIVERSITETET I TRONDHEIM
NORGES TEKNISKE HØGSKOLE
FAKULTET FOR ELEKTRO- OG DATATEKNIKK



HOVEDOPPGAVE

Kandidatens navn: Tormod Holte
Fag: Datateknikk
Oppgavens tittel (norsk): Digitalt Fjernsynsarkiv
Oppgavens tittel (engelsk): Digital Television Archives

Oppgavens tekst:

Dagens fjernsynsarkiver er basert på en kombinasjon av (digitale) databaser med beskrivelser av fjernsyns-materialet og analog lagring av selve fjernsyns-materialet på videobånd. Denne organiseringen gjør det tid- og arbeidskrevende å søke etter fjernsyns-materiale, det begrenser det beskrivelsesnivået som er tjenlig og ikke minst begrenses tilgangen til materialet av videobåndenes fysiske plassering.

Utviklingen de siste årene gjør det mulig å se for seg heldigitale fjernsynsarkiver der digital video er sømløst integrert med resten av systemet og der arkivsystemet kan gjøres tilgjengelig over datanett, f.eks. til distriktskontorer. Oppgaven går ut på å lage et utkast til design av et digitalt fjernsynsarkiv basert på NRKs behov, å gå nærmere inn på en del viktige problemstillinger og å vurdere realismen i et slikt prosjekt.

Oppgaven gitt: 22.08.94
Besvarelsen leveres innen: 22.12.94
Besvarelsen levert: 22.12.94
Utført ved: Institutt for datateknikk og telematikk
Veiledere: Amanuensis Roger Midtstraum
Stipendiat Rune Hjelsvold

Trondheim, 22.12.94

Roger Midtstraum
faglærer

Forord

Diplomoppgaven handler om *digitale fjernsynsarkiv* og er utført for IDTs gruppe for databaseteknikk. Faglig sett har oppgaven medført varierte utfordringer på “høyere nivå” (prosessmodellering), “mellomnivå” (datamodellering) og “lavere nivå” (bl.a. databasesystemer og praktisk implementering under X-Windows). Personlig har det også vært interessant at jeg gjennom diplomarbeidet har kunnet fordype meg i to av mine hovedinteresser: videoproduksjon og datateknikk.

NRK har med sin ekspertise og lange erfaring innenfor fjernsynsproduksjon og -arkiv vært en svært nyttig samarbeidspartner. Jeg vil med dette benytte anledningen til å takke alle enkeltpersoner i NRK som har vært støttespillere, bidragsytere og inspirasjonskilder under diplomarbeidet. En spesiell takk retter jeg til *Tedd Johansen* (fjernsynsarkivleder), *Solbjørg Engeset* (overingeniør, teknisk tjeneste) og *Reidar Otto Johnsen* (producer, U-redaksjonen) for all tid, hjelpsomhet og entusiasme.

En takk rettes også til *Tore Dybå* (forsker, SINTEF DELAB) og *Arnt Sæter* (diplomstudent, IDT) for godt samarbeid under NRK-besøkene og ellers.

Videre takker jeg mine to veiledere ved IDT, *Roger Midtstraum* og *Rune Hjelsvold*, for god veiledning med fleksible veiledningstidspunkt, konstruktiv kritikk, og mange interessante diskusjoner. Dessuten takker jeg for alle forslag til forbedring av annoterings- og browseverktøyet. Alle gode idéer som fortsatt ikke er implementert, kan utvilsomt aktivisere studenter i flere år fremover.

Trondheim, 22. desember 1994

Tormod Holte

English Summary

The title of this diploma thesis, ‘Digital Television Archives’, indicates the *digital* storage of broadcasted television material and its associated content-descriptions, within in the context of a *television archive* (library).

A television archive contains TV-programs for the purpose of documentation and possible reuse of material in new productions. To facilitate the retrieval process, additional text-based information (i.e. a description of the contents) is recorded after the program is finished and/or has been broadcasted. However, the retrieval may still end up becoming a time-consuming and tedious task, since the video material is stored separately on video cassette tapes. Also, the relationship between a program and its description, is usually limited to a tape number only. The thesis illustrates this concept by means of authentic examples from the Norwegian Broadcasting Corporation (NRK).

It is highly relevant to examine the demands being placed on the television archive by its environment. The environment can be considered as a set of TV-production processes. Five such processes have been identified and modelled; *planning and research*, *recording*, *editing*, *broadcasting* and *cataloging*. NRK’s production of television news is used as an example, and it’s concluded that both the *content-descriptions*, the *video material*, and the *processes*, all have some deficiencies or shortcomings.

Digital video is a technology which is already penetrating the production and storage of video material. Combined with methods for structuring and thematically indexing the video material, digital video is an important constituent in a digital television archive. A computer-based software application for a television archive can extract all relevant information from the five production processes and store this information in a database in a structured manner. By also storing digital video in a database, this framework can be an approach to integrating the different processes. Tighter couplings between the video material and the descriptions may be established, thus making the stored video material readily accessible.

Finally, prototypes for three basic components of such a software application (*annotating*, *searching*, and *browsing*,) are implemented for the purpose of realistic experiments and demonstrations.

Sammendrag

Tittelen på denne diplomoppgaven, “Digitalt Fjernsynsarkiv”, henspiller på *digital lagring* av fjernsynsprogrammer og programbeskrivelser anvendt i et *fjernsynsarkiv*.

Hensikten med et fjernsynsarkiv er å lagre fjernsynsprogrammer for dokumentasjonsformål og for eventuell gjenbruk i nye programmer. For å lette gjenfinningen benyttes tekstbasert tilleggsinformasjon (innholdsbeskrivelser) som registreres etter at programmet er ferdiglaget og/eller kringkastet. Gjenfinningen kan likevel være en tid- og arbeidskrevende oppgave fordi selve materialet er lagret separat på videobånd. I tillegg er koblingen mellom et program og beskrivelsen av dette, begrenset til et båndnummer som er inkludert i beskrivelsen. Rapporten illustrerer dette med en rekke konkrete eksempler fra Norsk Rikskringkasting (NRK).

For å undersøke hvilke krav omgivelsene stiller til et fjernsynsarkiv, har en sett nærmere på aktiviteter i forbindelse med fjernsynsproduksjonen. De fem aktivitetene *planlegging, materialproduksjon, redigering, sending og arkivering* er identifisert og modellert, med nyhetsproduksjonen i NRK som et eksempel. En samlet vurdering viser at det er svakheter eller mangler både ved *innholdsbeskrivelsene, videomaterialet og aktivitetene*.

Digital video er en teknologi som allerede er i ferd med å revolusjonere produksjon og lagring av video. Sammen med bedre strukturering og beskrivelse av videomaterialet, er dette en teknologi som kan benyttes for å realisere et digitalt fjernsynsarkiv. Datamaskinbaserte *støtteverktøy* for et slikt digitalt fjernsynsarkiv kan registrere relevant tilleggsinformasjon fra hver av de fem produksjonaktivitetene og lagre dette i en database. Når en også lagrer digital video i en database, kan verktøyet benyttes for å integrere de ulike aktivitetene. Dessuten er det mulig å gi enklere tilgang til videomaterialet ved å etablere flere relasjoner mellom videomaterialet og de tilhørende beskrivelsene.

Tre grunnleggende deler av verktøyet (*annotering, søking og browsing*), blir dessuten implementert med tanke på praktiske forsøk og demonstrasjoner.

Innholdsfortegnelse

1. INNLEDNING	1
2. FJERNSYNSARKIV	3
2.1 NRKS FJERNSYNSARKIV	3
2.1.1 REGISTRERING AV TV-PROGRAMMER	4
2.1.2 SØKING ETTER TV-PROGRAMMER	7
2.1.3 DATAMODELL FOR NRKS FJERNSYNSARKIV	9
2.2 FJERNSYNSARKIVETS OMGIVELSER	12
3. FJERNSYNSPRODUKSJON	13
3.1 DAGSREVYENS NYHETSPRODUKSJON	13
3.2 AKTIVITETS- OG INFORMASJONSANALYSE	14
3.3 VERDIKJEDE	15
3.4 DATAFLYTDIAGRAMMER	16
3.4.1 PLANLEGGING	18
3.4.2 MATERIALPRODUKSJON	19
3.4.3 REDIGERING	20
3.4.4 SENDING	21
3.4.5 ARKIVERING	24
4. PROBLEMSTILLINGER - NRK I DAG	27
4.1 INNHOLDSBESKRIVELSER	27
4.2 VIDEOMATERIALE	28
4.3 AKTIVITETER	30
4.3.1 STØTTEVERKTØY	30
5. TEKNOLOGISK GRUNNLAG - NRK I MORGEN?	33
5.1 DIGITAL VIDEO	33
5.1.1 LAGRING	34
5.1.2 KOMPRIMERING	34
5.1.3 ANVENDELSER	35
5.1.4 HARDDISKBASERT REDIGERING	37
5.1.5 HARDDISKBASERTE KAMERAER	39
5.2 DATAMODELL FOR VIDEOMATERIALE	39
5.3 TILLEGGSMODELL FOR INNHOLDSBESKRIVELSER	42
5.3.1 RELASJONSDATABASER VS SIFT	43
5.3.2 FRA TEKSTLIGE TIL STRUKTURERTE INNHOLDSBESKRIVELSER	44
6. DIGITALT FJERNSYNSARKIV	47

6.1 ARKITEKTUR	47
6.2 INFORMASJON FRA PRODUKSJONSKJEDEN	48
6.2.1 PLANLEGGING	48
6.2.2 MATERIALPRODUKSJON	50
6.2.3 REDIGERING	51
6.2.4 SENDING	53
6.2.5 ARKIVERING	54
6.3 STØTTEVERKTØY	55
6.3.1 FUNKSJONELLE KOMPONENTER	55
6.3.2 ANNOTERING	57
6.3.3 SØKING	57
6.3.4 BROWSING	58
7. PRAKTISKE FORSØK	61
<hr/>	
7.1 STØTTEVERKTØY VED IDT - EN OVERSIKT	61
7.2 ANNOTERINGSVERKTØY	62
7.3 SØKEVERKTØY	63
7.4 BROWSEVERKTØY	65
7.4.1 IMPLEMENTERING AV BROWSEVERKTØY	66
7.5 PRAKTISK BRUK AV VERKTØYENE	68
8. KONKLUSJON	71
<hr/>	
8.1 VIDERE ARBEID	72
VEDLEGG	73
<hr/>	

1. Innledning

Arkiver med film- og videomateriale har i årenes løp vokst til å bli en stor kulturskatt, på linje med bøker og kunstsamlinger. De fleste fjernsynsselskap har minst ett slikt arkiv, som da gjerne kalles *fjernsynsarkiv*. Det er liten tvil om at et fjernsynsarkiv er en verdifull ressurs, noe som nylig også har blitt bemerket i media. I en notis i Dagbladet [49] nevnes det f.eks. at norsk TV2 planla et spørreprogram hvor arkivmateriale fra 1970-årene skulle være en av hovedingrediensene. Som en forholdsvis nyetablert kanal håpet TV2 å få fri tilgang til hele NRKs arkiv, hvilket ble avslått. *Interaktiv TV* er en teknologi som bl.a. kan realiseres gjennom *video-på-forespørsel* (*video-on-demand*), hvor brukeren kan “bestille” programmer for visning på sin egen TV. Denne teknologien blir av noen blir spådd stor suksess [17, 50], og et rikholdig fjernsynsarkiv med populære programmer vil utvilsomt være et viktig konkurransefortinn.

Programmene i et fjernsynsarkiv tegner et bilde ikke bare av samfunnsutviklingen, men også den teknologiske utvikling som har funnet sted for lagringsmedier de siste årene. Teknisk kvalitet på opptakene har blitt stadig bedre etter hvert som nye film- og båndformat er tatt i bruk.

Et fjernsynsarkiv har flere viktige funksjoner i et fjernsynsselskap. Den mest åpenbare er kanskje *dokumentasjon*, altså at alle produserte og sendte programmer lagres for ettertiden. Slik lagring er også lovfestet i “Lov om avleveringsplikt for allment tilgjengelige dokument (av 1. juli 1990)” [2], og kopier av alle kringkastede programmer sendes til Nasjonalbibliotekavdelinga i Rana (NBR). Fjernsynsarkivene er imidlertid mye mer enn “lagerrom i påvente av reprisesendinger”. Tidligere sendte programmer er nyttig bakgrunnsmateriell, f.eks. illustrert ved Dagsrevyens hyppige gjenbruk av tidligere innslag med samme tema. Et fjernsynsarkiv er derfor også en viktig *støttespiller for fjernsynsproduksjon*.

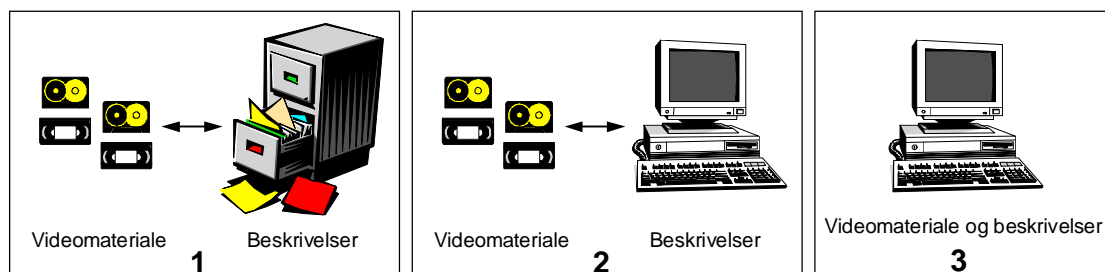
NRK er den institusjon i Norge som har lengst tradisjon med bruk av fjernsynsarkiv. Det er dessuten en omfattende programproduksjon i NRK, og i 1993 fylte egenproduserte programmer 56% av sendetiden [34]. Bedriftens opparbeidede ekspertise innen fjernsynsproduksjon og fjernsynsarkiv gjør den til en anerkjent aktør både nasjonalt og internasjonalt. Det har derfor vært både nyttig og inspirerende å kunne nyte godt av denne ekspertisen gjennom samarbeid med flere dyktige NRK-medarbeidere i dette diplomarbeidet. NRK blir et gjennomgangseksempel i denne rapporten for å illustrere praktiske problemstillinger som også har generell interesse og anvendelse.

Den omfattende egenproduksjonen i NRK gir en betydelig daglig tilvekst av arkivmateriale. Når antallet egenproduksjoner samtidig øker, blir det et stadig større behov for å finne igjen og benytte arkivopptak. Dette har (ifølge NRK) i den senere tid blitt forsterket i flere europeiske TV-selskap fordi det produseres en rekke programmer om den nyere historie, og for slike programtyper er arkivmateriale spesielt nyttig. Også egenproduksjonen i NRKs distriktskontorer er økende, men den fysiske oppbevaringen av arkivmaterialet på Marienlyst gir en viss leveringstid for kopier til distriktskontorene.

Digital video er et relativt nytt medium med potensiale for å revolusjonere både fjernsynsproduksjon og -arkiv. Lyd- og bildekvaliteten bevares ved kopiering, og med direkte tilgang (eng. *random access*) til materialet, kan videosekvenser avspilles i vilkårlig rekkefølge. Redigeringsystemer for digital video begynner å bli vanlig, og om kort tid kommer TV-kameraer med harddisker istedenfor videobånd [3].

For gjenfinning av arkivmateriale trenger en beskrivelser av innholdet. Utviklingen av fjernsynsarkiv kan beskrives i tre trinn, som også er illustrert i Figur 1.

1. *Analog video og analoge beskrivelser.* Tidligere var film det vanligste lagringsmediet, men analoge videobånd fikk etter hvert en ledende stilling. For beskrivelse av programinnhold var det vanlig å benytte maskinskrevne arkivkort [1].
2. *Analog video og digitale beskrivelser.* Dette er den mest vanlige løsningen i dag, hvor databaser inneholder tekstlige beskrivelser av videomaterialet. Analoge videobånd er fortsatt dominerende, selv om digitale båndformat begynner å ta over.
3. *Digital video, digital beskrivelse.* Med innføringen av digital video kan både programbeskrivelsene og videomaterialet lagres digitalt i databaser. Med et slikt "heldigitalt" fjernsynsarkiv kan en utnytte datastøtte for å tilby et integrert system som dermed forbedrer tilgangen til programbeskrivelsene og programmene.



Figur 1, Tre typer fjernsynsarkiv

NRKs Fjernsynsarkiv består i dag av analoge videobånd og digitale beskrivelser (vist i rute 2 i figuren). Denne rapporten vil bl.a. se nærmere på mulighetene for innføring av et fulldigitalt fjernsynsarkiv (vist i rute 3), og hvilke konsekvenser dette vil innebære.

Rapporten starter med en beskrivelse av NRK, basert på grundige studier av hhv. Fjernsynsarkivet (kapittel 2) og fjernsynsproduksjon (kapittel 3). Det er lagt særlig vekt på eksisterende prosesser og informasjon, og kapittel 4 oppsummerer erfaringer fra disse studiene. I den grad det er behov for forbedringer, omtales også eksisterende og fremtidig basisteknologi i kapittel 5. På denne bakgrunn presenterer kapittel 6 en arkitektur for et Digitalt Fjernsynsarkiv og foreslår støtteverktøy for et slikt system. Endelig blir praktiske forsøk og implementasjon av tre grunnleggende støtteverktøy omtalt i kapittel 7.

2. Fjernsynsarkiv

Fjernsyn er i utgangspunktet en *distribusjonsform* - en elektronisk overføring og spredning av signaler som representerer bilde og lyd. I sin opprinnelige form var fjernsynet bare en prosess for transport av signalene i *rom*. Ved bruk av film og senere video, ble det også mulig å *oppbevare* materialet, og på den måten også transportere det i *tid* [13].

Slik oppbevaring av TV-programmer for fremtiden, er en verdifull funksjon som ivaretas av et *fjernsynsarkiv*. I og med hjemmevideospillerens store utbredelse, kan en forsåvidt si at det i dag finnes mange “private fjernsynsarkiv” ute i de tusen hjem, selv om hvert av dem gjerne er begrenset til noen titalls kassetter. Det har imidlertid liten hensikt å lagre programmer som en ikke finner igjen senere, og gjenfinningen forenkles betraktelig dersom en skriver ned programtittel på etiketter eller i et separat register.

I større TV-selskap er fjernsynsarkivet langt mer omfangsrikt. Arkivfunksjonen er gjerne skilt ut som en egen organisasjon, og videokassetter (samt andre lagringsmedier, f.eks. film) blir lagret i store *magasin*. Med titusenvise arkiverte programmer, er det nødvendig med gode beskrivelser av innholdet for å effektivisere gjenfinningen.

2.1 NRKs Fjernsynsarkiv

Etter noen års prøvesendinger ble første ordinære sending fra NRK kringkastet 20. august 1960 [2]. Det var allerede fra starten behov for et fjernsynsarkiv, og dette ble formelt opprettet i 1961.

NRKs Fjernsynsarkiv har ansvaret for arkivering av TV-programmer og å utabeide de beskrivelser som er nødvendig for å finne dem igjen. Dessuten assisterer de brukerne med søking etter arkivmateriale. De åtte arkivarene ved Fjernsynsarkivet benytter omtrent halve arbeidsdagen til å beskrive innholdet i programmene og *klassifisere* dem etter tema. Arkivarene bruker også mye tid på å assistere de av arkivets brukere som trenger hjelp ved søking i databasen.

På grunn av sin geografiske avstand til fjernsynsarkivet på Marienlyst, kan reportere ved distriktskontorene ikke selv se arkivmateriale før de bestiller det. Eksempelvis sender en reporter ved NRK Tyholt sin bestilling av arkivmateriale til Fjernsynsarkivet, hvor arkivarene behandler bestillingen og ser gjennom arkivmateriale. I samråd med reporter blir aktuelt materiale kopiert og siden sendt med kurérpost til distriktskontorene. En dag i uka (for tiden torsdag) er satt av for slik kopiering av arkivmateriale for distriktskontorene.

I løpet av Fjernsynsarkivets 33-årige historie, har man opplevd en kontinuerlig utvikling av film- og båndformater¹. Selv om en del materiale har blitt kopiert til nye

¹ Som en kuriositet kan det nevnes at NRK allerede fra starten i 1961 hadde to videomaskiner [13]. Imidlertid var direktesendinger dominerende den første tiden.

formater², inneholder arkivet i dag et høyt antall forskjellige båndtyper. At kvaliteten på gamle opptak ikke er fullgod i forhold til dagens krav, er kanskje det minste problemet. Verre er det at overspilling til nye formater tar mye tid, og at man ikke lenger har avspillere til alle formatene. Film- og videomagasinet består av 4 store bomberom, hvert på ca. 500m², og rommer omkring 80.000 videobånd av forskjellig format og spilletid. Ett av rommene er forbeholdt filmmateriale. I tillegg finnes det et noe mindre rom med Sony Betamax påsynskassetter.

En tilsvarende utvikling har funnet sted for *beskrivelsene* av TV-programmene. Fra å ha benyttet kartotekkort av papir, ble EDB-basert arkiv innført i 1986. Siden 1988-89 har man benyttet fritekst-databasesystemet *SIFT Plus*. SIFT står for Søking i FriTekst, og dette systemet er nærmere beskrevet i Vedlegg A. Fjernsynsarkivets SIFT-database for beskrivelser av programmer kalles "*Fjernsynsarkivet*"³. Pr. august 1994 inneholder denne databasen drøyt 126.000 poster og har en størrelse på ca. 480 Megabyte.

SIFT er benyttet for flere av arkivsystemene i NRK, hvor nevnte database bare er en av flere som er tilgjengelig fra hovedmenyen "*NRKs arkivsystem*":

- | |
|---|
| <ul style="list-style-type: none"> • Platesamling - OSLO • Programsamling - OSLO (radio) • Fjernsynsarkivene (TV) <ul style="list-style-type: none"> - Fjernsynsarkivet - FLAK-databasen - Lillehammer OL-base - Distriktskontorenes arkiv • Nyhetsarkiv - Radio og Distriktskontor • Nyhetsarkiv - TV • Biblioteket |
|---|

Figur 2, Hovedmeny for NRKs arkivsystem

Fjernsynsarkivet er altså SIFT-basen som inneholder programbeskrivelser, mens *FLAK-databasen* inneholder fjernsynets sendeplaner (såkalte flak) og *Nyhetsarkiv - TV* inneholder kjøreplaner fra Dagsrevyens sendinger.

2.1.1 Registrering av TV-programmer

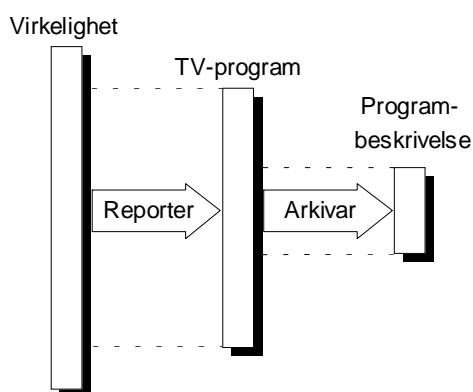
Alle programmer som er sendt, blir beskrevet og registrert (katalogisert) i SIFT-arkivet. I tillegg registreres de fleste Dagsrevy-innslag som ikke er sendt. (Det hender f.eks. at viktigere nyhetssaker kan føre til omprioriteringer. Nedprioriterte innslag kan

² Siste skudd på stammen er Digital Betacam (DigiBeta) - et båndformat for digitalt lagret video. I dag blir en del gammelt materiale overført til Digibeta.

³ "*Fjernsynsarkivet*" benyttes altså som betegnelse både på en organisasjon i NRK og på SIFT-databasen med innholdsbeskrivelser av programmer. I tillegg hender det at begrepet også benyttes om magasinet med videokassetter. For å redusere begrepsforvirringen mest mulig, blir det i denne rapporten forsøkt presisert "hvilket Fjernsynsarkiv" som menes, når dette ikke fremgår av sammenhengen.

likevel ha interesse senere). Originalt råmateriale blir *ikke* bevart⁴, og følgelig heller ikke beskrevet.

Arbeidet med å beskrive TV-programmer er ifølge Fjernsynsarkivet å betrakte som en *journalistisk* prosess. Forskjellen er at Dagsrevyens reportere rapporterer virkeligheten, mens Fjernsynsarkivets arkivarer rapporterer sendingene. Figur 3 illustrerer forenklet hvordan rapporteringen er en utvelgelse av visse aspekter, mens enkelte detaljer utelates i begge ledd.



Figur 3, To journalistiske prosesser

Det benyttes spesielle arbeidsrom med avspillingsutstyr og en terminal for registrering. Registreringsarbeidet blir først utført en stund etter at programmene er sendt, og det er derfor nødvendig å hente fram tidligere registrert informasjon om programmet. Slik informasjon hentes bl.a. fra andre SIFT-databaser. I tillegg hentes selve TV-programmet fra magasinet. Alle innholdsbeskrivelser lagres altså i SIFT-basen "Fjernsynsarkivet". Dagsrevyen og enkelte andre programmer blir behandlet som en serie innslag, hvor hvert innslag registreres for seg.

Et papirbasert skjema med *programstatistikk*, formelt kalt "Programopplysninger for fjernsyn", blir levert til Fjernsynsarkivet. Følgende informasjon inngår i denne statistikken:

- Programidentifisering
- Program/innslags-spesifisering
- Programansvarlige
- Opphavsrettigheter
- Programbeskrivelse
- Innslagsbeskrivelse

Nærmere detaljer fremgår av det ferdig utfylte skjemaet som er vist i Vedlegg B.

⁴Det hender likevel at enkeltpersoner gjemmer slike opptak av egen interesse. Faktisk er det ikke uvanlig at også slikt materiale blir benyttet om igjen i produksjoner - forutsatt at producer, reporter eller andre vet det eksisterer. Likevel er det slik at mange gode opptak som utmerket godt kunne benyttes i nye produksjoner, altså går tapt.

Registrering av programmet begynner som regel når programstatistikken ankommer Fjernsynsarkivet. Ofte får man ikke dette skjemaet før et par måneder etter at programmet er sendt. For Dagsrevyen benyttes istedet en *kjøreplan* (vist i Vedlegg E), og registreringene er for tiden 2-4 uker på etterskudd. Ved registreringen gir også de to SIFT-basene *FLAK-databasen* og *Nyhetsarkiv-TV* nyttig støtteinformasjon. Dessuten benyttes publikasjonen *Programbladet*.

For registrering av programinformasjon brukes følgende skjema:

```

--- Base: FJERNYSYN  Skjema: FJERNYSYN --- Reg.dato: 930926 -- Reg.nr: F
      .....          Signatur: ... / ... / ...           > 0
                        Prod. nr: .....

Tittel  : .....
Serie   : .....
Arb-titt: .....
Sendt   : .....   Tot-tid : .....   Del-tid : .....
Bånd.nr : .....
Båndtype: .....   I-versj.: .....
TC      : .....
Tekn.   : .....
Prod.   : .....
Klass.  : xxxx
Innhold : .....
.....

```

Figur 4, Skjema for registrering av programinformasjon

De fleste opplysningene hentes fra programstatistikken (Vedlegg B). *Tittel* er navnet på programmet eller, f.eks. i Dagsrevyens tilfelle, navnet på innslaget. I feltet *Serie* skrives "Dagsrevyen" dersom det gjelder et innslag i Dagsrevyen. Feltet *Klass.* inneholder en temabeskrivelse i form av nummer, basert på *Dewey desimal-klassifikasjonssystem* for tema. (Dette systemet brukes også vanligvis i fagbibliotek). En egen SIFT-database med Dewey-nummer og forklaring er til noe hjelp, men klassifiseringen er likevel en tung oppgave. Ideelt sett skal klassifiseringen gjennomføres samtidig med innholdsbeskrivelsen, men for nyhetsinnslag er man i dag flere år på etterskudd med klassifisering. *Innhold* er et felt av vilkårlig lengde, hvor programinnholdet beskrives i *fritekst*.

Fjernsynsarkivet har utviklet egne skriveregler som må følges under registreringen. Dette inkluderer bl.a. spesielle forkortelser og skrivemåter. *EA* betyr f.eks. "Eget Arkiv", *intv.* betyr "intervju", og etternavn skrevet med STORE bokstaver (f.eks. Ingvild BRYN) betyr at personen opptre i bildet. Arkivarene legger mao. inn mye *metainformasjon* i friteksten, dvs. informasjon det ikke er noen direkte støtte for i SIFT. Skjemaet er for øvrig gjengitt i sin helhet i Meroks diplomoppgave [2].

Reglene for innskriving er relativt omfattende⁵, men garanterer på ingen måte feilfrie beskrivelser. Dessuten motvirker de ikke at ulike arkivarer kan ha sin særpregede uttrykksform. Innholdsbeskrivelsen er lagret i et fritekst-felt, og selve teksten er da eneste mulige tilgang til denne beskrivelsen. Følgene av feilskrivning eller inkonsekvent terminologi kan bli at dokumentene vanskeligere gjenfinnes ved søk. Et konkret eksempel fra Fjernsynsarkivet, er søk etter "stridsvogn". Dette søket finner f.eks.

⁵ Versjonen som er gjengitt i [1] består av 15 sider.

ikke programmer hvor innholdet istedet er beskrevet som “panservogn” eller “tanks”. SIFT tilbyr ikke stavekontroll, og heller ikke registreringsstøtte (annet enn ledetekstene i skjemaet).

I og med at innholdet skrives som fritekst, blir det også stor variasjon på detaljnivået. Eksempelvis blir gjerne en uvanlig uttalelse (f.eks. “Sauer er ålreite dyr!”) skrevet i sin helhet, mens generelle bilder kan beskrives på formen “2 shots fra informasjonsstand i Spikersuppa”. Arkivarenes egen erfaring med hva som blir etterspurt, former også beskrivelsene.

Selv om registreringsarbeidet er en del på etterskudd, ønsker Fjernsynsarkivet likevel ikke å redusere detaljingsnivået på beskrivelsene. Dette kunne trolig fremskynde registreringen, men problemet ville da skyves over på brukerne, som senere ikke ville finne igjen programmene. Arkivarene forsøker heller å holde et relativt høyt detaljnivå. For å få til dette, benyttes av og til spesiallitteratur om det emnet programmet omhandler (f.eks. militære flytyper), hvor Fjernsynsarkiv har egne oppslagsverk.

De ferdige beskrivelsene lagres til slutt i SIFT-basen *Fjernsynsarkivet*, og alle brukere kan deretter søke i beskrivelsene for å finne ønsket materiale.

2.1.2 Søking etter TV-programmer

Alle NRK-ansatte har mulighet til selv å søke i Fjernsynsarkivets SIFT-base, eller de kan få hjelp av Fjernsynsarkivets personell. For søking benyttes SIFT’s kommandospråk. Kommandoen for søking er “finn”, og søkeordene kan kombineres som vist i avsnitt (2.1 SIFT). Det kan søkes i alle feltene i registreringskjemaet, også i kombinasjon med fritekst i innholdsfeltet. Imidlertid er det vanligvis bare arkivarene som søker i temafeltets Dewey-nummer (*Klass.*-feltet). Søk kan bare skrives med kommandospråket, og det finnes ingen søkeskjemaer.

Som et eksempel kan søkene “finn røyk*” eller “finn Oslo og statue” finne følgende programbeskrivelse:

```

--- Base: FJERN SYN   Skjema: FJERN SYN --- Reg.dat0: 900802 -- Reg.nr: F00/41639
                                         Signatur: eg /      / eg      >
                                         Prod. nr: FNYH 1151/90

Tittel   : Oslo : Verdens røykfrie dag
Serie    : Ettermiddagsnytt
Arb-titt:
Sendt    : 900531      Tot-tid : 00 00 58   Del-tid :
Bånd.nr  :
Båndtype: BETACAM SP          I-versj.: MAI 1657
TC       :
Tekn.    :
Prod.    : NRK   P: B. Amland  F: E. Haug
Klass.   : xxxx
Innhold  :
Verdens røykfrie dag markeres i Oslo sentrum med stor statue av sigarett-
stump. Avstand sigarettstump-statuen sett mellom to fontener i Spikersuppa.
Tilt ned statuen. Uid. kvinne uttaler seg off om statuen. Intv. uid.
yngre mann om statuen, synes røyk er sunnt for sjelen. Anti-røyker og
røykende mann diskuterer røykestopp. 2 shots fra informasjonsstand i
Spikersuppa. Uid. kvinne uttaler seg om røykekutt.

```

Figur 5, Søkeresultat

Søkeresultatet kan skrives ut på papir, og den fysiske videokassetten bestilles fra arkivet. Originalbånd lånes kun ut internt, mens distriktskontorene får en sendekvalitets kopi (bl.a. pga. risikoen med postgang). Det finnes dessuten flere *påsynsrom* for å se gjennom arkivmaterialet.

Innslaget vist i Figur 5 ligger på bånd nummer MAI 1657 sammen med en rekke andre innslag. *MAI*⁶ er en betegnelse som benyttes for bånd med Dagsrevyens originale innslag - den såkalte I-versjon (internasjonal versjon), som bare inneholder original video og lyd. Selve sendingen, slik den går på lufta, blir lagret på egne *dokumenasjonsbånd*. Disse får et eget båndnummer som skal lette gjenfinningen. For nyhetssendinger benyttes sendedato istedet for båndnummer, og feltet *bånd.nr* fylles derfor ikke inn for slike sendinger. Dette gjelder også for innslaget i Figur 5. Dokumentasjonsbånd er ikke like godt egnet for gjenbruk som MAI-bånd, da de også inneholder super⁷, kommentar og studiokommentar, slik Figur 6 viser.



Figur 6, MAI-bånd og dokumentasjonsbånd

På omslaget til MAI-båndet er det festet en innholdsfortegnelse som viser hvor på kassetten det søkte innslaget ligger. For de fleste programmer blir det ikke notert tidsangivelser i innholdsbeskrivelsene i SIFT⁸, slik at for lengre programmer kan det bety mye spoling og leting for å finne et bestemt innslag. Det kan altså være en møysommelig prosess fra et søk påbegynnes i SIFT til det ønskede arkivmaterialet er klart for gjennomsyn og evt. gjenbruk.

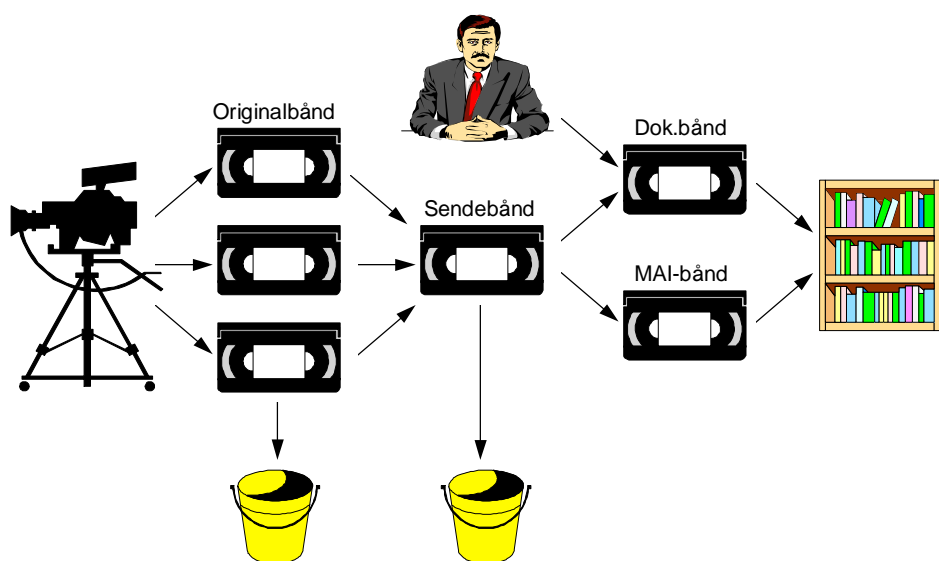


⁶ MAI er et glimrende eksempel på NRKs tilsynelatende forkjærlighet for akronymer. Kassetten måtte i sin tid gis et navn, og tre personer oppkalte dem ganske enkelt etter seg selv: Margareta, Arne og Ivar.

⁷ Tekst og grafikk lagt oppå bildet, (eng. *superimpose*).

⁸ Dette er likevel vanlig for sportssendinger, og det er da feltet TC (*Time Code*, tidskode) som benyttes.

Figur 7 oppsummerer hva som skjer med de ulike båndene, med spesiell vekt på Dagsrevyen:



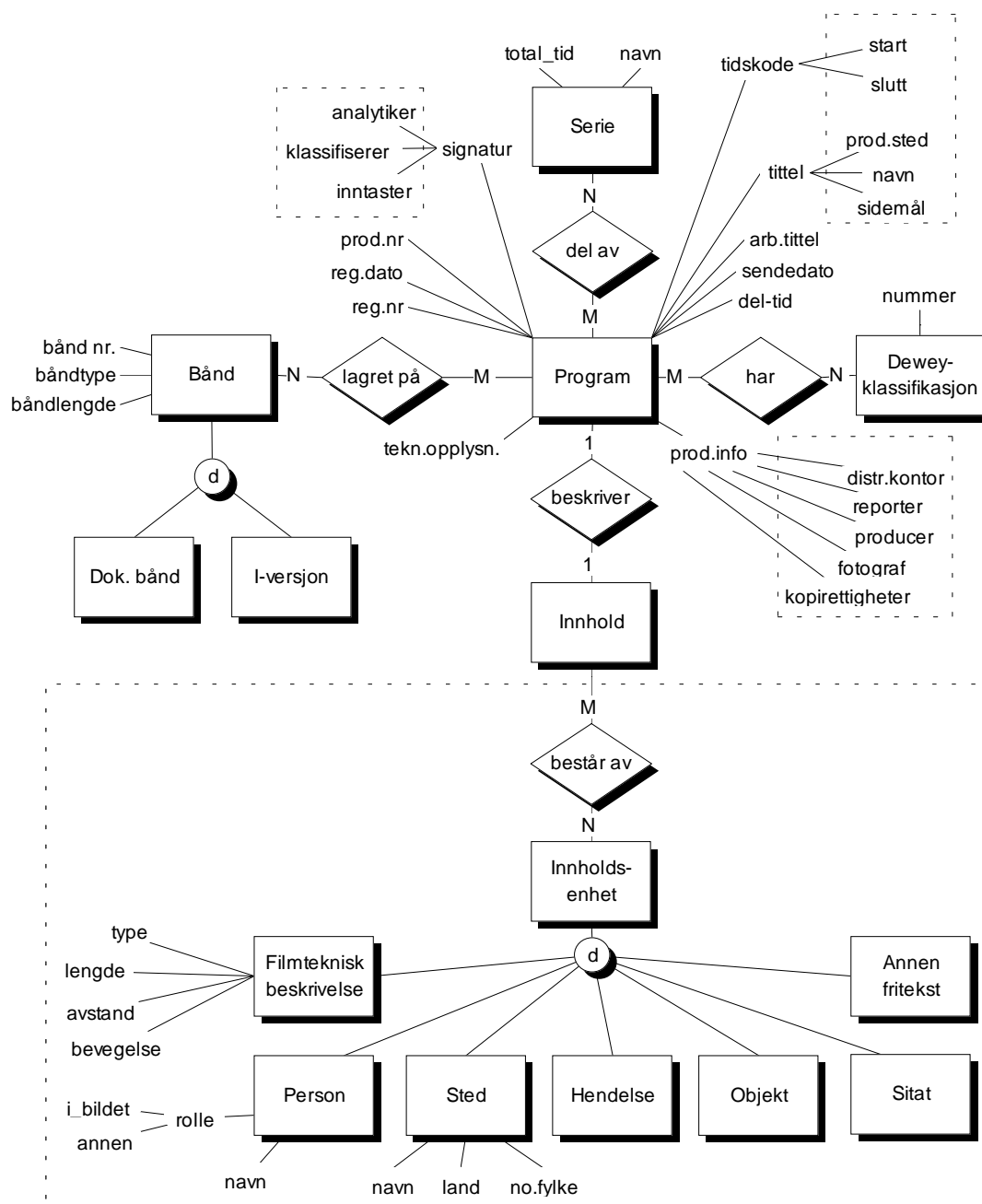
Figur 7, Videobånd

De aller fleste *originalbånd* blir “kastet”, som i praksis betyr at de blir overspilt med nytt materiale. Originalopptak som skal brukes, blir redigert og lagt på *sendebånd*. Noen sendebånd slettes, mens interessante Dagsrevy-innslag (som antas å kunne bli gjenbrukt senere) blir kopiert over til MAI-bånd. For Dagsrevyen leser også programlederen studiol kommentarer under direkte sending, som lagres på *dokumentasjonsbånd* sammen med innslag fra sendebånd. Dokumentasjonsbåndene inneholder altså en kopi av sendingen slik den sees i de tusen hjem. MAI-bånd og dokumentasjonsbånd blir lagret i magasinet. For Dagsrevyen er det pr. i dag bare innholdet i de enkelte *innslagene* (reportasjene) som blir registrert - ikke programleders introduksjon og studiol kommentarer.

2.1.3 Datamodell for NRKs Fjernsynsarkiv

Innholdet i Fjernsynsarkivets SIFT-database kan vises skjematisk i en datamodell. For Fjernsynsarkivets vedkommende er imidlertid situasjonen noe spesiell. Enkelte databasefelt benyttes for å lagre flere attributter, f.eks. inneholder ett av feltene (“*Prod.*”) informasjon om både distriktskontor, reporter, producer, fotograf og kopirettigheter. For kopirettigheter benyttes igjen forkortelser (EA=eget arkiv, AU=arkiv utenfra og C=copyright). Som tidligere nevnt, er også informasjon “skjult” i skrivemåten, f.eks. store bokstaver i etternavnet til avbildet person. Dette indikerer at databasedefinisjonen (og dermed registreringskjemaet) ikke dekker Fjernsynsarkivets behov godt nok - noe man altså har løst ved å benytte spesielle skriveregler.

For best mulig å reflektere den informasjonsmengden som faktisk registreres, må en datamodell også vise implisitt lagret informasjon. Følgende EER-datamodell⁹ forsøker å fange dette:



Figur 8, Datamodell for NRKs Fjernsynsarkiv.

⁹EER - Enhanced Entity Relationship. Modelleringskonsepter er hentet fra [14], men av layoutmessige hensyn avviker notasjonen noe fra foreslått standard. F.eks. er ellipsen rundt attributtene og symbolet "U" på subklasse-relasjonene utelatt. Nøkler er heller ikke satt opp, da det ikke er støtte for dette konseptet i SIFT. Dessuten er det innført stiplede bokser rundt attributter det ikke er noen direkte støtte for i databasen.

Modellen er satt opp med utgangspunkt i notatet “Fjernsynsarkivets skriveregler”, gjengitt i [1], og skjemaet i Figur 4. Entiteter, relasjoner og attributter innenfor de stiplede boksene er implisitt informasjon som er avledet av skrivereglene og innholdsbeskrivelsene.

De fleste attributtene er samlet under entiteten *Program*, mens særlig viktig informasjon er representert som egne entiteter. En *Serie* kan inneholde mange *Program*. For Dagsrevyens vedkommende registreres hvert innslag som et eget program og hver sending som en serie. Et program kan sendes flere ganger (være med i flere serier), eller et innslag kan inngå i flere nyhetssendinger (selv om det er svært uvanlig at innslaget gjentas i uendret form). Relasjonen *del_av* blir derfor av typen mange-til-mange (M:N).

Et *Program* kan være lagret på flere *Bånd*, og et *Bånd* kan inneholde mange *Program*. Dessuten kan et *Program* ha mange *Dewey-klassifikasjoner* (nummer), og mange *Program* kan svare til samme klassifikasjon. Relasjonene *lagret_på* og *har* blir derfor også mange-til-mange (M:N).

Selv om programmets *Innhold* er beskrevet som fritekst, kan innholdet betraktes som en samling av mange *Innholdsenheter*. Følgende typer innholdsenheter ser ut til å gå igjen i Fjernsynsarkivets SIFT-base:

1. *Filmteknisk beskrivelse*. Minst fire kategorier er her vanlig: *Type* (NRK-begreper: trick, grafikk, animation og super) *Lengde* (NRK-begreper: shot, strekk og scene - gjerne med flere varianter av hver type), *Avstand* (NRK-begreper: nært, halvært, fjernt, avstand) og *Bevegelse* (zoom inn/ut, pan høyre/venstre, tilt opp/ned).
2. *Person*. Navn på person (alternativt gruppe eller organisasjon) registreres dersom dette er kjent. Opptrer personen i bildet, lagres det også informasjon om dette (som nevnt vha. store bokstaver).
3. *Sted*. I tillegg til geografisk stedsnavn, lagres landets navn eller norsk fylke for steder i Norge. Ikke-navngitte (generiske) steder som “skog” eller “vann” kan i denne modellen betraktes som objekt eller sted, men dette er bare et spørsmål om hvilken representasjonsform man velger.
4. *Hendelse*. F.eks. intervju, diskusjon eller sabotasjeaksjon.
5. *Objekt*. Fysisk objekt, f.eks. statue, fontene eller røyk.
6. *Sitat*. F.eks. “Sauer er ålreite dyr!”. Skrives i sin helhet særlig når uttalelsen er uvanlig eller har spesiell interesse.
7. *Annen fritekst* kan være generelle kommentarer som ikke dekkes av de øvrige kategoriene, f.eks. “Norsk spillefilm fra 1948”.

En setning i fritekst vil typisk inneholde flere slike innholdsenheter. Setningen “Pan fra bebyggelse i Måløy til vegvesen-arbeidere i ferd med å reparere skader på brua” inneholder f.eks. informasjon om filmteknikk (pan), person (vegvesen-arbeidere), sted (Måløy), hendelse (reparere skader), og objekt (bebyggelse, brua). Det kan kanskje virke noe kunstig å “dekodde” en setning på denne måten. Hensikten er på ingen måte å presentere en allmengyldig, universell modell, men snarere en bevisstgjøring av den semantikken (meningen) som ligger bak ordene i beskrivelsen.

Sannsynligheten for to helt identiske beskrivelser av to forskjellige program er forsvinnende liten, og en kan derfor trygt si at det er et én-til-én forhold mellom *Program* og *Innhold* (unntatt ved reprisesendinger, men innholdet blir likevel ikke beskrevet på nytt). Imidlertid kan samme *Innholdsenhet* forekomme i mange beskrivelser av *Innhold*. Dette gjør det mulig for et søk, f.eks. etter “bebyggelse”, å finne alle programmer med ønsket innholdsenhet (dvs. alle programmer med innholdsenheten “bebyggelse”).

2.2 Fjernsynsarkivets omgivelser

Etter denne presentasjonen av fjernsynsarkiv, med NRKs Fjernsynsarkiv som eksempel, er det naturlig å se nærmere på arkivets omgivelser. Gjenfinning av videomateriale er altså viktig, og et sentralt spørsmål her, er *hvorfor* man ønsker å finne noe igjen. Et “privat hjemme-videoarkiv” er motivert av ønsket om å se favorittprogrammene om igjen, og når som helst. I et TV-selskap er imidlertid arkivets viktigste oppgave å kunne levere større eller mindre *deler* av programmer - som bakgrunn og arkivmateriale for nye produksjoner. En kan altså si at slike fjernsynsarkiv primært fungerer som en *serviceinstitusjon* for de som produserer nye TV-programmer. En grundig analyse av *produksjonsprosessen* er derfor et godt utgangspunkt for en bedre forståelse av arkivet og bruken av dette, samt et fundament for design av et digitalt fjernsynsarkiv. Fjernsynsproduksjon er temaet i neste kapittel.

3. Fjernsynsproduksjon

Målsetningen for all fjernsynsproduksjon er selvsagt å produsere programmer eller programinnslag for kringkasting. Det er store individuelle forskjeller mellom de ulike programtypene som produseres. Direktesendte sportssendinger, dokumentarprogrammer, dramaproduksjoner, debattprogrammer og nyhetsprogrammer er noen eksempler på programmer som jevnlig sendes på NRK¹⁰. Programmene er ikke bare forskjellige i form og innhold, men også produksjonsmåte. Vedlegg C viser noen karakteristiske faktorer ved egenproduksjon hos NRK.

Det er lite formålstjenlig å beskrive all fjernsynsproduksjon hos NRK i denne rapporten. En grundig studie av én produksjonsprosess er tilstrekkelig for å avdekke de fleste sentrale aspekter. Med det store mangfoldet av produksjonstyper, er det vanskelig å velge et “typisk” eksempel på en fjernsynsproduksjon. For denne rapporten er likevel bruk av Fjernsynsarkivet mest interessant, og Dagsrevyen utmerker seg her som den desidert største brukeren¹¹. Dagsrevyens nyhetsproduksjon blir derfor benyttet som *eksempel* på en produksjonsprosess i denne rapporten.



3.1 Dagsrevyens nyhetsproduksjon

Produksjonen av Dagsrevyen har sine særegne kjennetegn, f.eks. små produksjonsteam, kort produksjonstid og tidsbegrenset planlegging. Likevel er Dagsrevyen representativ for de viktigste produksjonsprosessene og bruken av Fjernsynsarkivet. Nyhetssendinger er dessuten allerede benyttet som gjennomgangseksempel i IDT's pågående forskning på videodatabaser, omtalt bl.a. i [16].

En rekke personer er involvert i produksjonen av en nyhetssending. Vedlegg D nevner noen av de mest sentrale aktørene og terminologi som benyttes hos NRK.

All sport- og nyhetsproduksjon i NRK, både i radioen og fjernsynet, er basert på verktøyet *Basys*. Dette er et dokumentstøttesystem for nyhetsproduksjon som blir benyttet av en rekke kringkastingsselskap. NRK benytter en versjon som er spesielt

¹⁰I tillegg kjøpes ferdige programmer og filmer inn fra forskjellige leverandører, som også blir beskrevet i Fjernsynsarkivet. I forbindelse med fjernsynsproduksjon, er det imidlertid egenproduksjonen som er mest interessant.

¹¹Fjernsynsarkivet har likevel ingen statistikk over bruken.

tilrettelagt deres behov og oversatt til norsk. NRKs Basys-system er en stor, sentral database plassert på Marienlyst. Systemet brukes både på Marienlyst og av distriktskontorene, og følgende funksjoner er tilgjengelig:

- NRK abonnerer på telegrammer, meldinger og artikler fra 5 telegrambyrå, bl.a. NTB og Reuter/AFP. Alt som mottas fra disse byråene lagres og kan leses i Basys. Meldingene oppbevares i 7 dager, unntatt NTB-meldinger, som oppbevares i 14 dager.
- Lage *kjøreplaner* for nyhets- og sportsprogrammer. Dagsrevyens kjøreplaner for de siste 60 dager kan til enhver tid leses fra Basys. (En kjøreplan fra Dagsrevyen er vist i Vedlegg E).
- En *skuff* er Basys-betegnelsen for en *newsgroup* eller katalog. Alle brukerne har en privat skuff hvor de kan skrive egne saker (nyhetsinnslag) og andre dokumenter. Når en reporter har skrevet en sak, kan den kopieres¹² over til en felles-skuff og evt. inkluderes i nyhetssendinger. Basys beregner fortløpende hvor lang tid det tar å lese opp en bestemt tekst, basert på den enkelte opplesers lesestil.
- Avhengig av gjeldende tilgangsrettigheter, er det mulig å lese og redigere sakene til andre reportere. Ansvarlig redaktør tilpasser alle sakene til kveldens sending, og programleder skriver sin egen introduksjon (studiokommentar) til saken, basert på reporterens forslag.
- Producer kan skrive *skuddliste*, som er en detaljert liste over kamerabruk og grafikk som skal inkluderes.

Alle saker til nyhetssendingen blir skrevet ut sammen med kjøreplan og skuddliste. Alle studiokommentarer blir også gjort tilgjengelig for programleder på *auto-cue* (lesemaskin montert på kameraet) under sending.

Basys går altså som en rød tråd gjennom Dagsrevyens nyhetsproduksjon, og legger på mange måter premissene for produksjonsaktivitetene. For ytterligere detaljer om Basys, vises det til [39] og [40].

Som omtalt i kapittel 2.1 og Vedlegg A, bruker Fjernsynsarkivet fritekstdatabase-systemet SIFT. Dagsrevyens kjøreplaner kopieres fra Basys til SIFT-basen "Nyhetsarkiv-TV" etter hver sending, slik at også kjøreplaner fra eldre Dagsrevy-sendinger enn 60 dager kan leses derfra.

¹² Strengt tatt blir ikke saken kopiert, men det opprettes en referanse til den.

3.2 Aktivitets- og informasjonsanalyse

Hensikten med en aktivitets- og informasjonsanalyse er å skaffe oversikt over virksomheten i en bedrift og hvilken informasjon som gjør denne virksomheten mulig. Dette kapittelet presenterer resultatet fra slik analyse gjennomført hos NRK som en del av dette diplomarbeidet høsten 1994. Analysen er utført i samarbeid med Arnt Sæter og Tore Dybå, men fremstillingen av produksjonsprosessen i dette kapittelet er likevel mer omfattende og detaljert enn [39] og [40]. Motivasjonen for slik analyse er, som tidligere nevnt, å få kunnskap om fjernsynsproduksjon og -arkiv, og et godt utgangspunkt for å utarbeide design av et digitalt fjernsynsarkiv.

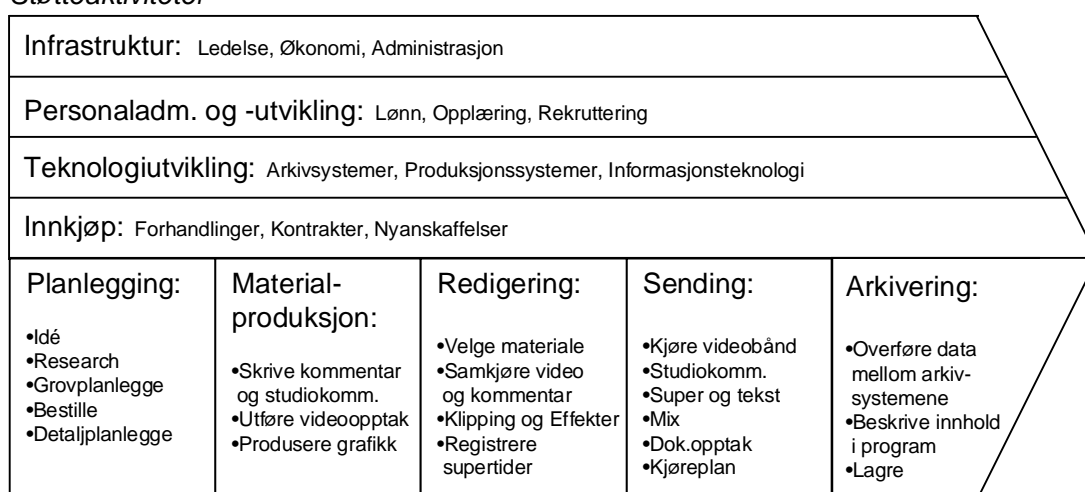
Arbeidsmetodikken har vært intervjuer, samtaler og møter med en rekke ressurspersoner fra ulike fagmiljøer hos NRK, og observasjon av deres daglige arbeid. Gjennom oppfølgende samtaler har dessuten hovedtrekkene i analyseresultatene blitt verifisert av NRK.

Det er utviklet mange metoder for slik analyse og presentasjon av resultater. Porter og Millar [10] benytter *verdikjeder* for å fokusere på bedriftens *essensielle (verdiskapende) aktiviteter* på overordnet nivå. *Dataflytdiagrammer (DFD)*, bl.a. beskrevet i [15], er en mer detaljert, mye benyttet representasjonsform for prosesser, lagre og dataflyt. Dette kapittelet benytter en kombinasjon av verdikjede og dataflytdiagrammer, slik det bl.a. er foreslått i [8].

3.3 Verdikjede

En verdikjede består av to deler: aktiviteter som direkte er en del av produktfremstillingen (*primæraktiviteter* - fjernsynsproduksjon i NRKs tilfelle) og aktiviteter som støtter opp om disse (*støtteaktiviteter*). På bakgrunn av diplomarbeidets studier av NRK, synes følgende verdikjede å være beskrivende og (i denne sammenheng) dekkende:

Støtteaktiviteter



Primæraktiviteter

Figur 9, Verdikjede for fjernsynsproduksjon

Støtteaktivitetene i øvre halvdel er tegnet parallelt med primæraktivitetene, som skal indikere at de foregår samtidig og fortløpende. I de fleste bedrifter utføres en rekke støtteaktiviteter, og selv for bedrifter som er svært forskjellige, er støtteaktivitetene relativt like. Det er vanlig å benytte fire “standard” støtteaktiviteter, som er tilstrekkelig for å gi en generell oversikt [8, 10]:

- *Infrastruktur*. Her inngår f.eks. ledelse, økonomi og administrasjon. Utarbeidelse av fjernsynets *sendeplan* og tildeling av ressurser og sendetid er eksempler på nødvendige aktiviteter av denne typen.
- *Personaladministrasjon og -utvikling*. Lønn, opplæring og rekruttering er eksempler på slike støtteaktiviteter.
- *Teknologiutvikling*. For NRK er arkivsystemer (f.eks. Basys og SIFT) og produksjonssystemer (f.eks. Basys) svært viktige, og under stadig utvikling. Generelt dekker denne kategorien all informasjonsteknologi, og den innsats som settes inn på å følge med i utviklingen.
- *Innkjøp*. Forhandlinger og kontrakter inngås om nyanskaffelser, f.eks. av eksternt produserte TV-programmer.

Primæraktivitetene i nedre halvdel leses fra venstre mot høyre, slik at produksjonskjeden starter med planlegging og slutter med arkivering. I praksis er aktivitetene ikke nødvendigvis så tydelig adskilt fra hverandre, og det er ofte avvik fra det sekvensielle forløpet som er tegnet her.

For NRK er primæraktivitetene altså de som inngår i fjernsynsproduksjonen. Følgelig er det primæraktivitetene som har størst interesse i denne rapporten, og neste avsnitt beskriver disse i nærmere detalj.

3.4 Dataflyttdiagrammer

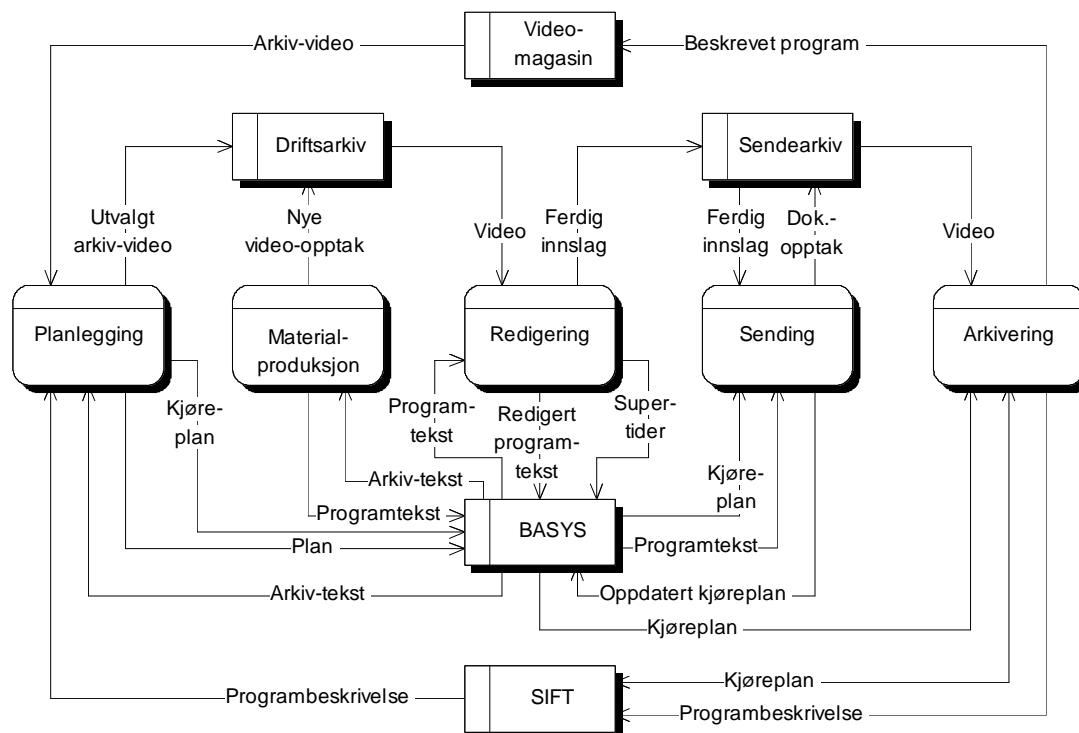
De fem primæraktivitetene fra Figur 9 kan settes opp i et overordnet dataflyttdiagram som viser sammenhengen mellom dem og de ulike arkivene. Figur 10 viser et slikt overordnet dataflyttdiagram.

For å redusere kompleksiteten noe, er bare de viktigste dataflyter vist på dette nivået, og eksterne entiteter (aktører) er heller ikke tatt med. Påfølgende figurer vil vise dekomponerte dataflyttdiagrammer for hver prosess.

I Figur 10 kan primæraktivitetene gjenfinnes langs en horisontal linje midt i figuren. Hovedrekkefølgen for utførelsen av aktivitetene er fra venstre mot høyre, selv om parallellitet og gjentakelser er mulig.

Diagrammet er organisert slik at arkivene på oversiden av aktivitetene inneholder video, mens arkivene på undersiden inneholder tekst. All dataflyt er lagt via arkivene. To av arkivene har i dag bare direkte betydning for planleggings- og arkiveringsprosessen:

- *Videomagasinet*. Dette er magasinet som inneholder videobånd med ferdige, sendte TV-programmer.
- *SIFT*. Som forklart i kapittel (2.1 SIFT), inneholder NRKs SIFT-system en rekke databaser. Relevant i denne sammenhengen er TV-programmenes innholdsbeskrivelser (*Fjernsynsarkivet* i SIFT) og fjernsynets sendeplaner (*FLAK-databasen* i SIFT).



Figur 10, Overordnet dataflytdiagram for fjernsynsproduksjon

De tre øvrige arkivene benyttes i forbindelse med flere av aktivitetene:

- *Driftsarkiv*. Dagsrevyen har et eget, mer uformelt internt videobåndarkiv som kan kalles “driftsarkiv”, da det er nødvendig for den daglige drift. I praksis er dette hyller i redaksjonslokalet hvor man oppbevarer *newsfeed* (opptak fra byråer og andre TV-kanaler) og egne *temakassetter* med standardscener som benyttes ofte. Blant tematassetene er det for tiden opptak bl.a. av Bosnia, satanisme, hvalfangst og banker. Temakassetene er med på å redusere “unødvendig” bruk av Fjernsynsarkivet. For å ikke operere med for mange arkiv i dataflytdiagrammene, vil driftsarkivet her for ordens skyld også inneholde videobånd med *innslag* som er under produksjon
- *Sendearkiv*. Dette arkivet inneholder ferdige innslag som er klare til sending. For å unngå å innføre flere arkiv enn høyst nødvendig i dataflytdiagrammene, blir også dokumentasjonsopptak av kringkastede programmer liggende i dette arkivet til de blir arkivert og beskrevet av Fjernsynsarkivet.

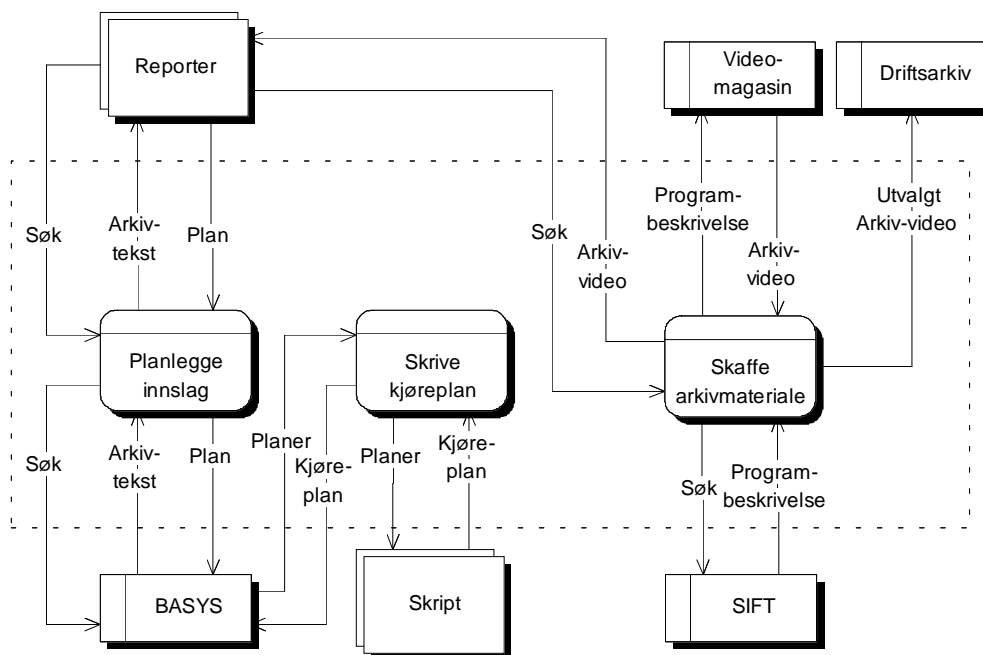
- *Basys* ble kort omtalt i kapittel 3.1, og er altså et dokumentstøtteverktøy for nyhetsproduksjon. I dataflytdiagrammene blir *Basys* representert som et arkiv, da *Basys*' evne til å lagre og gjenfinne informasjon er det mest vesentlige her.

De følgende avsnittene viser dekomponerte dataflytdiagrammer for hver aktivitet.

3.4.1 Planlegging

Første aktivitet i fjernsynsproduksjon er planlegging. Hos NRK er det planlegging på flere nivåer. På overordnet nivå tildeles sendetid og ressurser til de ulike avdelinger og programposter. En *sendeleder* har ansvar for gjennomføring av dagens programmer i henhold til fjernsynets sendeplan, og kan om nødvendig foreta omprioriteringer på oppsatt sendeplan. I verdikjeden (Figur 9) ble dette definert som en støtteaktivitet.

Også for hver programpost foregår det mye planleggingsarbeid, og i denne rapporten er det *slik* planlegging som er av størst interesse. For Dagsrevyen foregår det igjen planlegging på to nivåer; for sendingen som helhet og for hvert enkelt innslag. Dataflytdiagrammet i Figur 11 viser en dekomponering av planleggings-aktiviteten fra Figur 10, og gir et innblikk i planlegging av Dagsrevyen.



Figur 11, Dataflytdiagram for aktiviteten "Planlegging"

Den stiplede boksen angir hovedaktiviteten "Planlegging", og tre del-aktiviteter er tegnet inn:

Planlegge innslag. Den kreative idéprosessen foregår i reporterens (og/eller andres) hode. Det vil føre for langt å diskutere alle mulige opphav til en sak (tips, rykter, pressemeldinger, byråmeldinger, avisartikler, Dagsnyttssendinger, egne idéer osv.), men ofte vil reporter søke etter nærmere informasjon i *Basys*. Dersom søket er vellykket,

får han returnert ønsket arkivert tekst, som kan være av alle de typer Basys støtter (f.eks. meldinger, kjøreplaner og studiol kommentarer). Reporteren skrives et utkast eller forslag til sak, og dette lagres som en *Plan* i Basys. Med de knappe tidsfrister en opplever for nyhetsproduksjon, er det nærmest utenkelig å lage dreiebok (se vedlegg C).

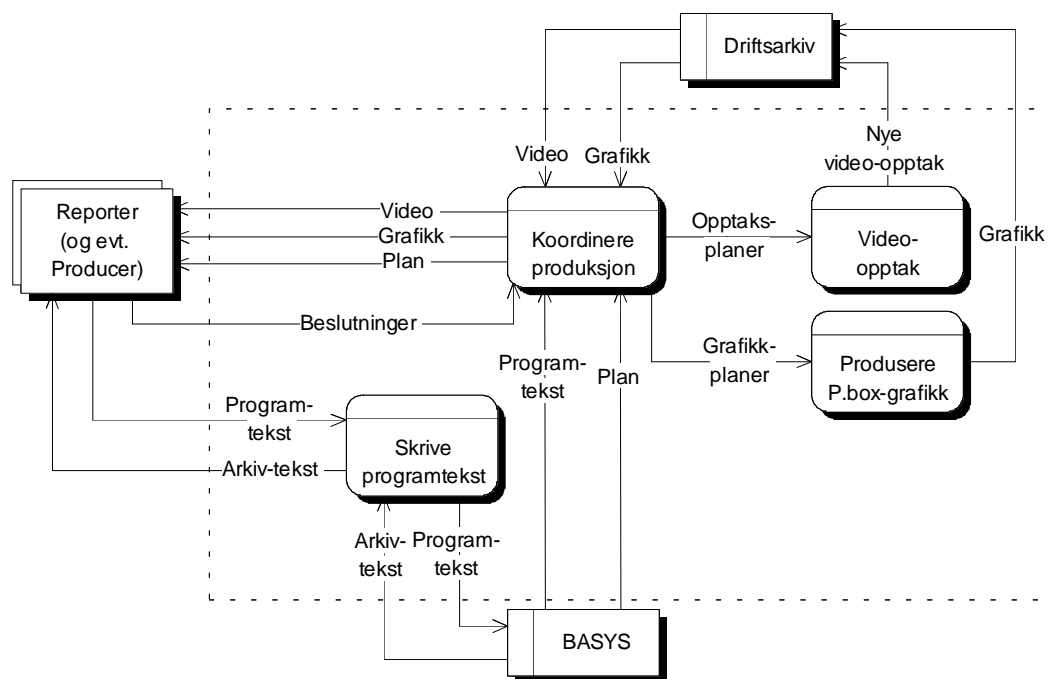
Skaffe arkivmateriale. Det er ofte ønskelig å benytte arkivmateriale fremfor å foreta nye opptak. Reporteren kan skaffe dette selv, eller få noen til å gjøre det for seg (f.eks. skript, producer eller redaksjonsassistent). Vellykkede søk i SIFT returnerer programbeskrivelser til passende programmer eller innslag, og beskrivelsene (eller en referanse til ønsket program) kan leveres til Fjernsynsarkivet for å få låne arkivert video fra magasinet. Interessant materiale blir gjennomgått, og passende programmer eller innslag legges i *Driftsarkivet* (som vanligvis ikke er mer formelt enn en stabel videokassetter).

Skrive kjøreplan. Det holdes en rekke redaksjonsmøter i løpet av dagen, og etter hvert skriver skripten en kjøreplan som lister opp de sakene som prioriteres til kveldens sendinger. Dette er en kontinuerlig prosess, og kjøreplanen kan endres helt opp til en halv time før sending.

I verdikjeden (Figur 9) ble dessuten *bestilling* nevnt i forbindelse med planlegging. Bestilling av f.eks. studio og redigeringskapasitet er vanlig for de fleste produksjoner [11], men Dagsrevyen er et unntak i så måte, da de har eget studio og egne redigerere.

3.4.2 Materialproduksjon

Neste prosess er materialproduksjon, og en dekomponering av denne er vist under.



Figur 12, Dataflytdiagram for aktiviteten "Materialproduksjon"

Fire delaktiviteter er her tatt med:

Skrive programtekst. Reporteren skriver, evt. på bakgrunn av tekst fra Basys, selve programteksten (kommentarer og studiokommentarer)¹³ til sitt innslag. Det er ikke uvanlig at producer bistår i denne prosessen. Mens programleders studiokommentarer *må* skrives i Basys, er det store individuelle forskjeller når det gjelder reporterens egne kommentarer. Noen reportere skriver disse for hånd, gjerne parallelt med at videoopptakene redigeres (kapittel 3.4.3)

Koordinere produksjon. Dette er en styrende aktivitet, hvor hovedoppgaven er å sørge for progresjon i produksjonen. Ut fra planen og programteksten i Basys vurderes hvilken video og grafikk som er nødvendig. Evt. arkivmateriale (som ble fremskaffet under planleggingsfasen) blir også nærmere vurdert, og påvirker planene for nye videoopptak eller grafiske illustrasjoner som skal produseres.

Video-opptak. Nye opptak i studio eller på *location* gjennomføres. Hele reportasje-teamet er av plasshensyn ikke tegnet inn, men er likevel kort omtalt i Vedlegg D.

Produsere Paintbox-grafikk. Enkelte konsepter lar seg vanskelig illustrere med video, og grafikk kan da være en løsning. NRK benytter *Quantel PaintBox* for dette formålet. Noe uformelt kan en si at også grafikken inngår i driftsarkivet.

3.4.3 Redigering

Redigering er en selvfølge i de aller fleste tilfeller, både av programtekst og video. Prosessen kan imidlertid også utelates, f.eks. ved direkte overføring og dersom særlig viktige nyheter ankommer like før eller under sending. Figur 13 viser likevel en dekomponering av den vanligste formen for redigering i forbindelse Dagsrevyen.

Koordinere redigering. Reporteren kontakter en av Dagsrevyens redigerere, og sammen setter de seg ned for å redigere aktuell sak. Noen reportere fokuserer mest på tekst, og andre på bilde. Dette resulterer i ulik redigeringsstil, men det kan være nødvendig å redigere både kommentarene og videomaterialet. Programteksten redigeres i Basys, og ansvarlig redaktør kan også påvirke dette om nødvendig. Det er store individuelle forskjeller mellom de ulike reporterens planlegging før selve redigeringsprosessen påbegynnes. Noen kommer til redigeringsrommet kun med en bunke videokassetter, mens enkelte har skrevet et detaljert forslag til klippeliste.

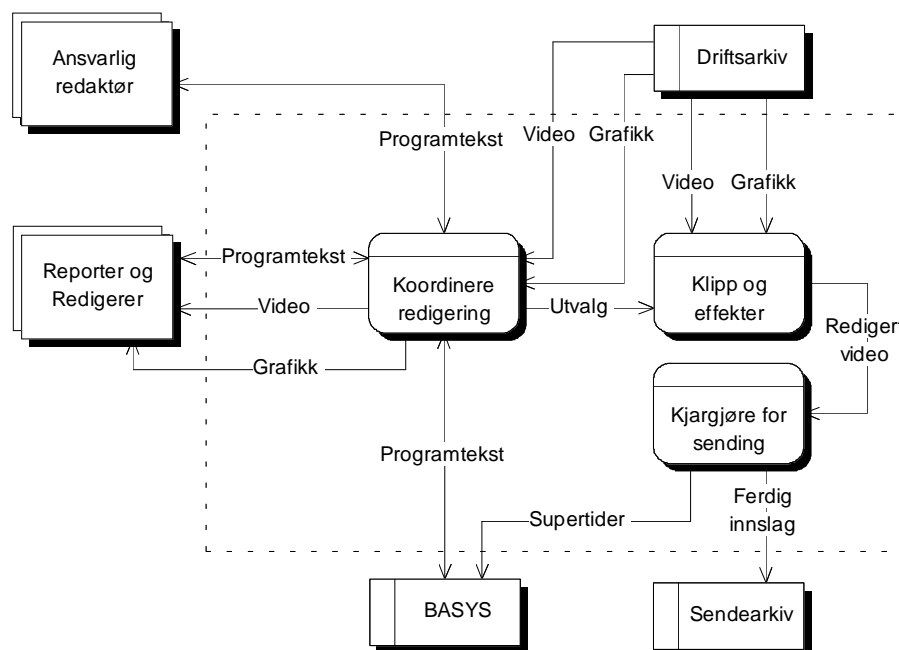
Klipp og effekter. På Dagsrevyens utstyr er det mulig å foreta enkle klipp, mens man overlater oppdraget til *Båndfilmsentralen* dersom ytterligere effekter er nødvendig. Dagsrevyen benytter ikke *for-redigering*¹⁴, så redigeringen foregår direkte på de

¹³ *Programtekst* er ikke et uttrykk som benyttes i Dagsrevyen, men det blir likevel brukt i denne rapporten som et samlebegrep på *studiokommentarer* og reporterens egne *kommentarer*.

¹⁴ Registrering av tidskode (start og stopp) for utvalgte scener i en klippeliste som senere overføres til redigeringsutstyret for automatisk kopiering. For-redigering kan foregå med papir og blyant eller med datastøtte.

originale opptakene. En betydelig andel av redigeringstiden går med til spoling og bytting av kassetter.

Klargjøre for sending. Når innslaget er ferdig redigert, må det registreres *supertider*, som er tidspunkt for manuell pålegging av tekst og grafikk under sending. Disse opplysningene legges inn i kjøreplanen i Basys. Endelig legges det ferdige innslaget i sendearkivet og er klart for kveldens sending¹⁵.



Figur 13, Dataflytdiagram for aktiviteten "Redigering"

3.4.4 Sending

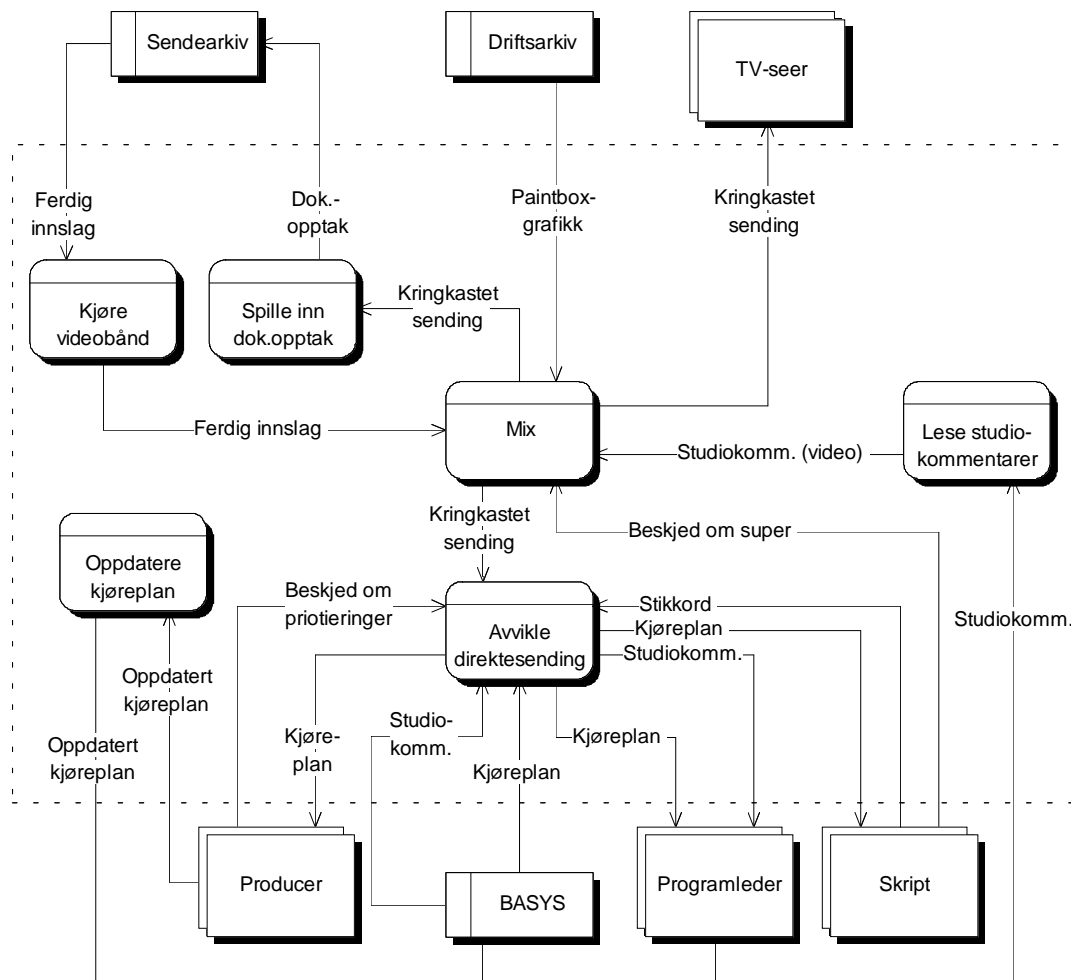
Bak kulissene virker en direkte sending av Dagsrevyen som "velorganisert kaos" for en utenforstående. Kontrollrommet for studio 5 (Dagsrevy-studio) inneholder ca. 75 monitører (bl.a. med bilder fra TV-kameraene, PaintBox-illustrasjoner, tekstemaskin og signaler mottatt utenfra). I kontrollrommet sitter følgende personer:

- Producer, som er ansvarlig for regien og konsentrerer seg om det som sendes i øyeblikket
- Skript, som bl.a. har ansvar for sendingens kontinuitet og hva som er neste hendelse (såkalt *preview*)
- Teknisk leder
- Lys- og kameraansvarlig
- Stillstoreansvarlig (grafikk og stillbilder)
- Skriftoveratør
- Bildemixer

¹⁵ Det hender imidlertid at innslag blir klare like før de skal sendes. En person *springer* da fra redigeringsrommet til avspillingsrommet (VB-rommet) med kassetten. Erfaring har vist at denne løypa tar omkring ett minutt...

I tillegg finnes en rekke personer i studio, bl.a. innspillingsleder og (selvsagt) programleder. I eget VB-rom sitter videobånd-operatører som kjører de ferdiglagde innslagene på producers kommando.

Følgende dataflyttdiagram viser en del aktiviteter i forbindelse med sending:



Figur 14, Dataflyttdiagram for aktiviteten "Sending"

Kjøre videobånd. Ferdige innslag spilles av på producers kommando, og går via en siste prosess (*mix*) før det kringkastes.

Lese studiokommentar. Studiokommentarene til kveldens sending gjøres tilgjengelig for programleder som papirutskrift og tekst på lesemaskin (også kalt *prompt* eller *auto-cue*) montert på kameraet. Programleder leser opp studiokommentarer foran kamera, og også dette videosignalet går til *mix*. Programleder har også kjøreplan, både som papirutskrift og på Basys-terminal (som for tiden kan sees ved programleders høyre side). Eksempel på studiokommentarer til en Dagsrevy-sending er for øvrig vist i Vedlegg F.

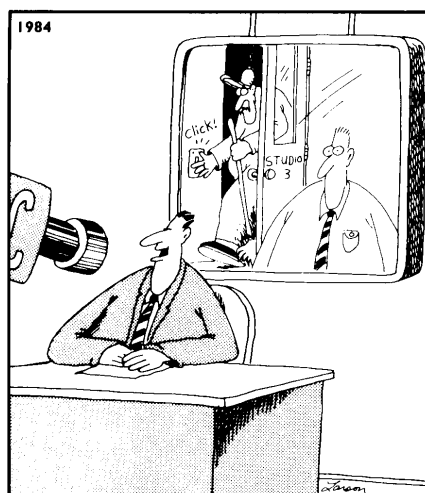
Avvikle direkte-sending. Dette er en kompleks prosess hvor hensikten er å styre avviklingen av en direkte-sendt Dagsrevy kombinert med de ferdiginnspilte innslagene. Producer tar de nødvendige beslutninger, og selv om det ikke er tegnet inn i Figur 14,

påvirker dette alle de andre aktivitetene. Beslutningene gjelder f.eks. kamerautsnitt og -vinkler (hentes fra Basys' *skuddliste*, som ikke er vist i diagrammene), kjøring av videobånd og henstillinger til programleder om å endre lesetempo hvis nødvendig. Skripten benytter stoppeklokke for å kontrollere sendingen i henhold til kjøreplanen, og gir i god tid beskjed om *slutt-stikkord* (de siste ordene fra innslaget eller i programleders studiokommentar) som indikerer når et innslag er ferdig.

Mix. Som en siste prosess før bildet går på lufta, blir videokildene pålagt *super* (f.eks. navn på reporter eller teksten "Arkiv" for å indikere at man viser arkivmateriale), *tekst* (f.eks. undertekster på utenlandske innslag) og *mix* (f.eks. *wipes* - overgang mellom to etterfølgende innslag i bestemt mønster - eller montasje av programleder og intervjuobjekt i annet studio). Skripten holder oversikt over supertider, og gir i god tid beskjed om når super skal legges på (basert på informasjon som tidligere er skrevet inn i kjøreplanen). I slutten av kjeden blir sendingen kringkastet, og kommer til slutt til TV-seeren¹⁶.

Spille inn dokumentasjonsopptak. Dagsrevyen er direktesendt, og for å bevare hele sendingen blir det derfor spilt inn et dokumentasjonsopptak. Dette er et opptak hvor super, tekst, mix og studiokommentarer er med, altså det samme som kringkastes. Dokumentasjonsopptaket (som for øvrig er av full sendekvalitet) legges i sendearkivet sammen med innslagene. Dersom innslaget også er kopiert inn på MAI-bånd (kapittel 2.1.2), har en altså to forskjellige versjoner av innslaget. Som tidligere nevnt, er de originale innslagene på MAI-bånd best egnet for gjenbruk.

Oppdatere kjøreplan. En sending blir ikke alltid avviklet som planlagt. Tekniske feil kan oppstå, eller det kan komme inn ferske nyheter underveis som får høy prioritet. Producer oppdaterer derfor kjøreplanen etter at sendingen er ferdig, slik at den best mulig stemmer med den sendingen som faktisk gikk på lufta.

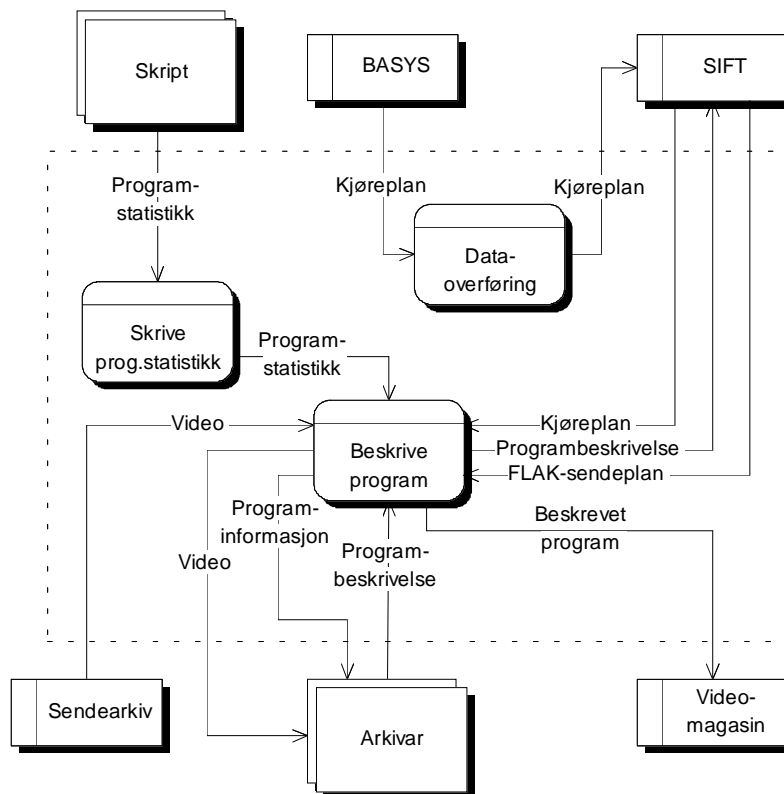


"My next guest, on the monitor behind me, is an organized crime informant. To protect his identity, we've placed him in a darkened studio— so let's go to him now."

¹⁶ Kringkasting ut til TV-seer er i seg selv en interessant og omfattende prosess, men som likevel ikke berøres i denne omgang. Noen aspekter er omtalt i [17] og [39].

3.4.5 Arkivering

Siste primæraktivitet fra Figur 10 er arkivering. Det gjelder her å sikre at programmet eller innslaget senere lett kan finnes igjen og om ønskelig benyttes som arkivmateriale for nye produksjoner.



Figur 15, Dataflytdiagram for aktiviteten "Arkivering"

Dette siste dekomponerte dataflytdiagrammet viser to delaktiviteter:

Dataoverføring. Dagsrevyens kjøreplan blir overført fra Basys til SIFT (Nyhetsarkivet) når den er oppdatert etter sendingen. Hensikten er langtidslagring, både så reporterne kan søke i dem, og til bruk for Fjernsvarsarkivet når programmene skal beskrives.

Skrive programstatistikk. For de fleste produksjoner må skripten fylle ut programstatistikk (forklart i avsnitt 2.1.1 og vist i Vedlegg B) som leveres Fjernsvarsarkivet. Dette gjelder imidlertid ikke for Dagsrevyen, hvor kjøreplanen fra Basys erstatter programstatistikken. Fjernsvarsarkivet starter registreringsarbeidet først når statistikken mottas - ofte opptil to måneder etter at programmet er sendt.

Beskrive program. Videobånd hentes ut fra sendearkivet, og kjøreplaner hentes ut fra SIFT (alternativt papirbasert programstatistikk fra skripten) slik at programmets innhold kan beskrives i SIFT's Fjernsvarsarkiv-database. Fjernsynets sendeplaner i SIFT's FLAK-database blir også benyttet ved registreringen, sammen med evt. annen

informasjon arkivaren kan skaffe. Etter registreringen blir videobåndet permanent lagret i magasinet¹⁷. Denne delprosessen ble for øvrig også omtalt i kapittel 2.1.1.

Med dette er alle de viktigste aktivitetene i produksjonskjeden for Dagsreveyn beskrevet.

¹⁷ Selv om det kan ta en stund før programbeskrivelsen blir skrevet, stilles programmet til disposisjon for brukerne straks det er sendt.

4. Problemstillinger - NRK i dag

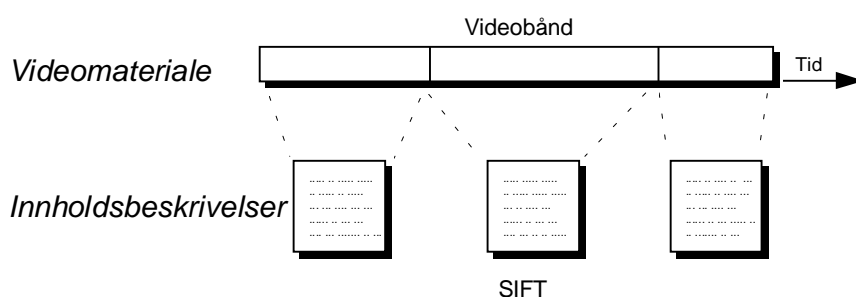
De to foregående kapitler har hhv. beskrevet fjernsynsarkiv og -produksjon, konkretisert gjennom eksempler fra NRK. Møter og samtaler med NRK-ansatte på Marienlyst (Oslo) og Tyholt (Trondheim) under informasjons- og aktivitetsanalysen, avdekket en del svakheter ved dagens situasjon hos NRK. Sammen med de tanker man gjør seg som utenforstående observatør, er dette nyttig ballast ved senere design av støtteverktøy for digitalt fjernsynsarkiv.

Problemstillingene er relatert til tre hovedområder, som blir nærmere omtalt i resten av dette kapitlet:

- *Innholdsbeskrivelser.* Selv om de tekstlige beskrivelsene av arkiverte programmer forenkler gjenfinningen av arkivert videomateriale betraktelig, er det fortsatt rom for en del forbedringer.
- *Videomateriale.* Tilgangen til videobånd fra arkivet er begrenset av båndenes fysiske plassering, og gjenfinningen av ønskede innslag er begrenset av deres sekvensielle struktur.
- *Aktiviteter.* Bare en brøkdel av programinformasjonen som genereres i løpet av produksjonsaktivitetene når fram til Fjernsynsarkivet. Både under produksjonen og ved arkiveringen setter dessuten støtteverktøyene (hhv. Basys og SIFT) unødvendige begrensninger.

4.1 Innholdsbeskrivelser

En forutsetning for å benytte arkivert videomateriale, er at en finner det en leter etter. Innholdsbeskrivelsene i SIFT er knyttet til videomaterialet, og til stor hjelp når dette skal lokaliseres. Figur 16 viser denne sammenhengen, hvor videobåndet som et eksempel inneholder tre programmer eller innslag.



Figur 16, Videomateriale med innholdsbeskrivelser

Koblingen mellom videomaterialet og innholdsbeskrivelsene er imidlertid relativt "løs". Dette skyldes både videobåndets natur, forklart i neste avsnitt, og at innholdsbeskrivelsene har en del svakheter.

Det viser seg at det ikke alltid er godt nok samsvar mellom innholdsbeskrivelsen og det som faktisk ligger på videobåndet. Opptakene kan være så gamle at de er uaktuelle for gjenbruk, uten at beskrivelsen gir noe hint om dette. Ofte viser opptakene uønskede værforhold eller årstider¹⁸. F.eks. er arkivstoff til en sommerreportasje fra Oslo sentrum uegnet dersom bildene viser det første snøfallet. Dette er ganske selvfølgelig, men det er ikke alltid at beskrivelsene sier noe om slike forhold.

Det har også vært påpekt at beskrivelsene av musikkprogrammer ikke alltid inneholder ordet “musikk” og at ungdomsprogrammer ikke alltid inneholder ordet “ungdom” - som burde være et minimum. Mens arkivarene gjør sitt beste for å detaljbeskrive et program, hender det altså at beskrivelsen ikke er *generell nok*. I slike tilfeller er søkeren prisgitt sin egen eller andres hukommelse for å finne igjen programmet. F.eks. kan det som en siste utvei være en hjelp å huske omtrent *når* et bestemt program ble sendt, og søke på sendedato.

Registreringsskjemaet i SIFT (Figur 4) er utgangspunktet for det som skrives inn om hvert program. Som datamodellen i Figur 8 viste, legges det likevel inn mye informasjon som ikke direkte støttes av registreringsskjemaet. En kan derfor slå fast at skjemaet i seg selv ikke er optimalt tilpasset Fjernsynsarkivets behov. Et symptom som underbygger denne påstanden, er det omfattende regelverket for hvordan registreringene skal utføres. Med en databasestruktur som i større grad reflekterer virkelig informasjonsbehov, vil arkivarene kunne fritas for mye “mental belastning”. Sammen med bedre datastøtte, kan dette være med på å øke kvaliteten på beskrivelsene og dermed antall vellykkede søk.

Innholdsbeskrivelsene er i dag uttrykt som setninger i naturlig språk. Dette gir hver enkelt arkivar en viss frihet til å velge personlig uttrykksform. Det ble nevnt at arkivarene utfører en form for journalistisk arbeid (kapittel 2.1.1). *Hensikten* med beskrivelsene er likevel å finne igjen programmer senere - ikke en høyverdig skjønnlitterær opplevelse. Det er stor fare for inkonsekvent ordbruk og uttrykksform, som senere kan gjøre det vanskelig å finne igjen arkivmaterialet. En mer strukturert tilnærming til arkiveringsprosessen vil bli vurdert i kapittel 5.2, hvor innholdsbeskrivelsene forsøkes uttrykt som strukturerte objekter istedenfor uforutsigbare formuleringer.

Ved Fjernsynsarkivet blir det fortalt at det i NRK-miljøet finnes mye udokumentert kunnskap om gamle programmer. En del personer har vært med helt siden starten (f.eks. Erik Bye og Erik Diesen), og har verdifull informasjon som aldri er blitt skrevet ned i innholdsbeskrivelsene. Det ville være ønskelig å registrere slik informasjon før det er for sent, men med dagens verktøy og kapasitet er dette dessverre umulig.

4.2 Videomateriale

De aller fleste program spilles inn på magnetiske videobånd (kassetter). Fjernsynsarkivet arkiverer kontinuerlig slike videobånd i magasin. Som regel blir det lagret to

¹⁸ Med *uønskede* menes ikke personlige preferanser i forhold til klimatiske forhold (eksempelvis varme, solfylte mai-dager), men arkivmaterialets anvendelighet i en gitt sammenheng.

eksemplar: det originale sendebåndet og et dokumentasjonsbånd av lavere kvalitet. I tillegg lagres, som nevnt i avsnitt 2.1.2, en I-versjon av enkelte programmer (f.eks. en del Dagsrevy-innslag på MAI-bånd). For å unngå slitasje på originale bånd, benyttes dokumentasjonsbånd for påsyn.

Akkurat som for bøker i et bibliotek, er det mulig at det bestemte båndet en søker, for øyeblikket er utlånt. Dagsrevyen opplever stadig at MAI-bånd er utlånt, og det hender en må nøye seg med dokumentasjonsbånd av selve sendingen, som også inneholder studiókommentarer og super.

Distriktskontorenes tilgang til videomaterialet er i tillegg begrenset av geografisk avstand til Marienlyst. Det er en økende produksjonsaktivitet hos distriktskontorene, og som en konsekvens opplever Fjernsynsarkivet stadig flere henvendelser derfra. Kopiering og distribusjon av dokumentasjonsbånd er allerede i dag en aktivitet som tar mye tid, og med en økende etterspørsel kan dette vokse til et stort problem.

Både analoge og digitale videobånd blir ofte pålagt *tidskode*, som er et nummererings-system for å identifisere hver ramme (hvert enkeltbilde) i videoen. Standardformatet SMPTE¹⁹ for tidskode benytter formatet *tt:mm:ss:rr*, hvor *tt* = timer, *mm* = minutter, *ss* = sekunder og *rr* = rammenummer. Tidskoden representerer enten tidspunktet på dagen da opptaket ble utført, eller den angir hvor lang tid som har gått siden starten på båndet [43].

Videobånd er et lagringsmedium hvor lagringen og tilgangen til videomaterialet er *sekvensiell*. Dette innebærer at en ikke kan hoppe vilkårlig rundt i materialet, men må spille det av samme rekkefølge som det ligger på båndet. Også frem- og tilbakespoling må utføres sekvensielt, og tar mer tid jo lengre en må spole. Tidskoden på et videobånd endrer ikke på dette, men det går raskere å finne igjen et innslag på båndet dersom en kjenner tidskoden til starten på innslaget.

For å finne igjen ønsket videomateriale, må en altså først finne riktig bånd, og deretter lete seg fram til riktig sted på båndet. Dette tar mye tid, både ved redigering og ved generelle søk etter arkivert videomateriale. Innholdsbeskrivelsene i SIFT angir sjelden tidskoder, selv om et felt i databasen er avsatt for dette. Dette kunne redusert søketiden en del for bånd med tidskode, men dokumentasjonsbåndene som vanligvis benyttes for påsyn, har ikke slik tidskode.

Som en konsekvens av stadig flere videobånd som skal arkiveres, har Fjernsynsarkivet også et fysisk plassproblem. Dessuten er det usikkerhet blant ekspertene når det gjelder holdbarheten av videobånd. Arkivmaterialets levetid kan forlenges ved at båndene kopieres, men for analoge bånd oppstår det et generasjonstap for hver kopiering. Etter OL har NRK kjøpt inn et stort antall digitale båndspillere (Digital Betacam), og inntil videre er digitale bånd eneste måte å sikre at kvaliteten ikke forringes ved kopiering.

Som tidligere nevnt, har ikke NRK noe eget organisert arkiv for råopptak som ikke er brukt i ferdige programmer. Dette er i og for seg forståelig, siden en i dag har mer enn

¹⁹ SMPTE - Society of Motion Picture and Television Engineers.

nok med å beskrive og finne igjen programmene. Likevel ville det ideelle være å spare *alle* opptak. Stoff som ikke passer i en bestemt produksjon, kan likevel være anvendelig i en annen sammenheng senere. Dessuten kan opptak som i dag er “uinteressante”, ha historisk interesse om noen år.

4.3 Aktiviteter

Fjernsvarsarkivet har ansvaret for å forfatte innholdsbeskrivelsene. For å kunne utføre dette arbeidet, er det imidlertid nødvendig å få best mulig informasjon fra de som lager programmene. Det eksisterer mye informasjon om TV-programmene mens de er under produksjon, enten denne skrives ned eller ikke. En stor andel av denne informasjonen når aldri frem til arkiverings-prosessen, hvor den kunne være nyttig for å utarbeide innholdsbeskrivelsene.

Selv om en del avdelinger er flinke til å fylle ut programstatistikk-skjemaet, er det ofte en tendens til at dette arbeidet utsettes til det er absolutt nødvendig. Erfaringsmessig er det også vanlig med skrivefeil og faktiske feil i programstatistikker og særlig i Dagsrevy-kjøreplaner. Dette medfører mye unødvendig dobbeltarbeid, siden arkivarene må rette feilene. Arkiveringen ender opp som en “opprydningssprosess” hvor Fjernsvarsarkivet retter feil, genererer ny informasjon og regenererer informasjon som tidligere eksisterte, men har gått tapt på veien.

Som nevnt ønsker Fjernsvarsarkivet bedre informasjon om programmene som skal arkiveres. Samtidig ønsker brukerne at det skal bli lettere å få tak i arkivmateriale. Evt. endringer, som i dette tilfellet ville berøre flere fagområder, forutsetter nødvendigvis mer overordnede beslutninger av ledelsen. For en vellykket omlegging, er det også svært viktig at alle involverte kan se en klar nytteeffekt for sitt daglige virke.

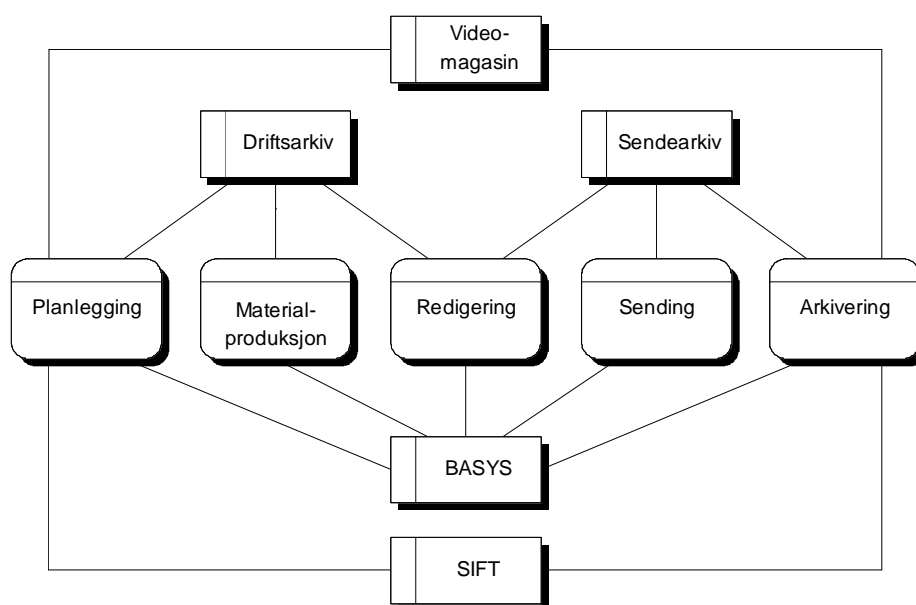
Det må også understrekes at en systematisk beskrivelse av produksjons- og arkiveringsaktivitetene, som dataflytdiagrammene i kapittel 3.4, ikke fanger opp de mange og viktige *uformelle* prosessene som foregår. Reporterens research kan f.eks. inkludere telefonsamtaler til kolleger i ulike avisredaksjoner eller til kontakter i næringslivet. Ved søk etter arkivmateriale er det også nyttig å støtte seg til NRK-kolleger med god oversikt over det aktuelle temaet. Dette er nødvendige aktiviteter, som også har en sosial funksjon. Iblant viser det seg at enkeltpersoner på eget initiativ har tatt vare på råopptak som ellers ville forsvunnet. Skal det lages en reportasje om Svalbard, kan det f.eks. være nyttig å kontakte en reporter som nettopp kom tilbake med opptak derfra. I mangel av et råstoff-arkiv, er dette en løsning som fungerer greit. Likvel ville det trolig være fruktbart å “synliggjøre” slik uformell informasjon ved at den ble registrert i en eller annen form.

4.3.1 Støtteverktøy

Både produksjons- og arkiveringsaktivitetene benytter støtteverktøy (hhv. Basys og SIFT). Både Basys og SIFT ser ut til å fungere greit hver for seg, og brukerne har ingen store innvendinger mot noen av dem. Et unntak er søketiden i Basys, som virker

uforholdsmessig lang²⁰. Dessuten blir mye av innholdet (studiokommentarer, kjøreplaner og byråmeldinger) slettet etter kort tid, slik at søkbar informasjonsmengde er relativt begrenset. Både Basys og SIFT er dessuten begrenset til tekstlige data. Det er forståelig at tilvendte brukere ikke reagerer på dette, men med digital video som integrert medium, åpnes det for nye muligheter. Disse vil bli nærmere forklart i kapittel 5 og 6.

Basys og SIFT er to separate systemer, og det er svært begrenset datautveksling mellom dem. Kjøreplanen fra Basys er eneste data som overføres til SIFT, og ingen data går motsatt vei. Figur 17 viser en forenklet utgave av dataflytdiagrammet i Figur 10.



Figur 17, Forenklet dataflytdiagram

Her fremgår det tydelig at SIFT og videomagasinet bare har direkte betydning for planleggings- og arkiveringsprosessen. Basys påvirker alle prosessene, selv om arkiveringsprosessen altså bare kopierer Basys-kjøreplanen over til SIFT. Det eksisterer altså et klart verktøymessig skille mellom Basys og SIFT. Dette kan være en av grunnene til manglende kommunikasjon mellom hhv. produksjon (de fire første aktivitetene) og arkivering, eller ihvertfall en årsak til at situasjonen ikke bedres.

Manglende integrasjon mellom Basys og SIFT fører til at brukerne må forholde seg til to systemer. Dessuten er det mange adskilte skuffer i Basys og mange adskilte databaser i SIFT, som kompliserer situasjonen ytterligere.

Både SIFT og Basys er kommandobaserte systemer. Dette betyr at brukerne må skrive til dels kryptiske kommandoer for å få utført det de ønsker. I Basys er det riktig nok

²⁰ Ved et konkret forsøk på søk i Basys tok det et drøyt minutt å søke etter fritekst i en av telegrambyrå-skuffene, og da var bare meldingene fra det siste døgnet gjennomført! Det er dessuten bare mulig å søke i én skuff om gangen.

mulig å benytte funksjonstaster for viktige kommandoer, men begge systemene mangler både menyer og tilpassede skjermbilder²¹. Med opplæring og erfaring kan en kan selvsagt vende seg til det meste, men både SIFT og Basys har mildt sagt lite brukervennlige grensesnitt. At systemene kjører på tekstbaserte terminaler utelukker selvsagt grafiske vindus-grensesnitt, men det er rom for forbedringer også innenfor dagens infrastruktur.

Direktesending av Dagsrevyen fungerer stort sett greit, men krever iherdig konsentrasjon av alle involverte. Med så mange samspillende faktorer som skal koordineres til rett tid, er det forståelig at problemer oppstår med jevne mellomrom. For en utenforstående virker det likevel noe påfallende at skripten manuelt benytter stoppeklokker for å holde orden på alle hendelser definert i Basys-kjøreplanen, f.eks. når super skal “kjøres inn og ut”. I NRK har man vært skeptisk til mer automatisert styring av sendingen, i motsetning til f.eks. TV2 [41]. Likevel burde det være mye å hente på en større grad av datastøtte, innenfor rammene gitt av de nåværende rutiner og roller.

²¹ En tenker ikke her på menyer for innlogging og valg av ulike baser på overordnet nivå, men på det nivået hvor selve arbeidet utføres.

5. Teknologisk grunnlag - NRK i morgen?

Forutgående diskusjon av problemstillinger indikerer at det er rom for forbedringer på flere områder hos NRK. Det er derfor interessant å se nærmere på tilgjengelig og fremtidig teknologi, og om dette kan bidra til å løse noen av problemene.

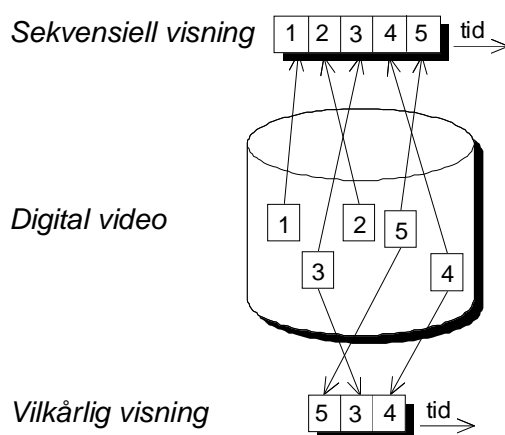
Det teknologiske fundamentet som presenteres i dette kapittelet, er relevant ikke bare for innføring av Digitalt Fjernsynsarkiv hos NRK, men i alle sammenhenger hvor en ønsker å benytte digitale videoarkiv.

5.1 Digital video

Til forskjell fra analog video, er digital video lagret på binært format. Med digital video oppnår en de samme fordeler som for all annen digital informasjon, f.eks. "kopier" av samme kvalitet som originalen, og mulighet til digital signal- og databehandling.

Mens analog eller digital video er et spørsmål om lagringsformat, spiller også lagringsmediet en stor rolle. Ved å benytte harddisk får en direkte tilgang (eng. *random access*) til det digitale videomaterialet, slik at det umiddelbart kan spilles av fra vilkårlig sted og i vilkårlig rekkefølge [2, 17]. Det er en tilsvarende forskjell for avspilling av lyd, hvor kassetter spilles sekvensielt, mens det på en CD-plate også er mulig å hoppe fritt mellom melodiene. Det må likevel nevnes at det finnes analoge medier med direkte tilgang (f.eks. videoplater, ofte kalt laserdisk) og digitale medier med sekvensiell tilgang (f.eks. digitale bånd).

Mens tidskode er en stor fordel for sekvensielt lagret video, er dette en absolutt *nødvendighet* for video med direkte tilgang (f.eks. lagret på harddisk), hvor rammene kan være spredt rundt på lagringsmediet. Unike rammenummer for hver enkelt ramme muliggjør datastøttet adressering av digital video. Dette sikrer at en fortsatt kan tilby tadisjonell, sekvensiell visning av video, samtidig som det er mulig å vise vilkårlige deler. Disse to visningsmåtene er illustrert i figuren under.



Figur 18, Sekvensiell eller vilkårlig visning av digital video

Tallene indikerer blokker av digitalt lagret video. For sekvensiell visning kan blokkene gjerne inneholde én ramme, men for at vilkårlig visning skal være meningsfylt, må

blokkene inneholde flere rammer som kan sees i sammenheng. Strukturering av digital video blir for øvrig nærmere beskrevet i avsnitt 5.2.

Slik rammenummer-adressering av video er mulig også for (sekvensielle) digitale bånd, men en må da påregne noe ventetid før en får tilgang til ønsket del av materialet.

5.1.1 Lagring

Digital video krever mye lagringsplass, og plassbehovet øker proporsjonalt med kravet til bildekvalitet. Et digitalt videosignal av *studiokvalitet* (NTSC 4:2:2 på komponentnivå med 8 bit pr. sample, definert i CCIR 601-standarden) opptar hele 216 Mbit [6]. Den europeiske PAL-standarden har en oppløsning på 720x576 *pixels* (bildelementer) [22], og lagringsbehovet for ett sekund ukomprimert digital video omkring 160 Mbit/s [26]. En snakker her bare om selve bildet, så lyden kommer i tillegg.

Dersom en antar en datarate på 160 Mbit/s, vil en typisk Dagsrevy-sending på 35 minutter oppta 42 GB (GigaByte). Dagens pris på ca. kr. 36.000,- for en (Seagate Elite) harddisk med 9 GB kapasitet gir en lagringspris på 4.000 kr/GB, og det ville koste rundt kr. 168.000,- å lagre en ukomprimert sending av Dagsrevyen! I 1993 sendte NRK Fjernsynet 348 timer nyheter [34], som alene ville fylle 25 TB (TeraByte). Selv om trenden hittil har vært større harddisker og lavere priser, er dette beløp som raskt dreper eventuelle visjoner om harddisk som permanent lagringsmedium - ihvertfall i overskuelig fremtid.

Ukomprimert video stiller ikke bare krav til lagring, men også til båndbredde. Harddisken må være i stand til å "pumpe ut" ca. 20 MB/s. Dagens harddisker kan bare levere omkring 3-5 MB/s [44], så spesielle teknikker som f.eks. *disk striping* er nødvendig [23, 24, 25]. Dette er en teknikk hvor en oppnår den nødvendige båndbredde ved å fordele videoen over flere harddisker.

Digitale båndautomater er et alternativt lagringsmedium. Slike selges for kr. 700.000,- og kan mates med inntil 256 bånd [50]. Prisen for et bånd er så lav som kr. 300,- og et bånd rommer 20 GB. Fylt med bånd tilbyr en slik enhet *online* adressering av over 5,1 TB (TeraByte) til 152 kr/GB (båndautomat medregnet). Lagring av ett års ukomprimerte nyhetssendinger i et slikt system ville kreve 5 båndroboter og koste omkring 4 millioner kroner. Følgelig er også dette et svært dyrt alternativ.

For slike båndbaserte systemer er aksessidene i størrelsesorden 1-2 minutter. Dersom bestemte deler av videomaterialet skal benyttes ofte, kan det derfor være hensiktsmessig å kopiere dette til harddisk, hvor aksessidene bare er noen millisekunder. En mulighet er også å lagre nytt materiale på harddisker og flytte dette til digitale bånd f.eks. etter et halvt år.

5.1.2 Komprimering

Til tross for dystre økonomiske scenarier, er det mulig å ta i bruk teknologien allerede i dag. Nøkkelordet er *komprimering* av video. Tapsfri komprimering er mulig med en faktor på 1,5:1 [6], mens høyere komprimeringsgrad er mulig dersom en aksepterer noe tap av bildekvalitet.

For komprimeringen benyttes gjerne den velkjente og utbredte standarden *JPEG*²², hvor hvert enkeltbilde komprimeres separat. Brukt på vanlig video kalles metoden ofte *MJPEG* (Motion JPEG). Komprimeringsgraden avhenger av den ønskede bildekvalitet, men ifølge CCITT og ISO er komprimeringsfaktorer på 25:1 og 100:1 egnet for hhv. stillbilde- og VHS-kvalitet [21]. Dersom en aksepterer VHS-kvalitet, er det altså mulig å redusere lagringsbehovet 100 ganger.

Det finnes også komprimeringsteknikker som i tillegg utnytter likhet mellom påfølgende enkeltbilder ved å lagre endringer fra ett bilde til et annet. *MPEG*-standard²³ er basert på dette prinsippet, og gir ytterligere økning i komprimeringsgraden. For dagens TV-standard²⁴ gir MPEG-1 påsynskvalitet med en datarate på omkring 1,5 Mbit/s, og MPEG-2 gir "god kvalitet" med 3-5 Mbit/s [26]. MPEG-1 komprimerer altså videomaterialet omkring 100 ganger, og reduserer lagringskostnadene tilsvarende²⁵. Et års nyhetssendinger vil da utgjøre 250 GB, slik at en båndstasjon med bånd (til under kr. 800.000,-) kan romme 20 år med nyhetssendinger eller 2 år med alle programmer som sendes fra NRK Fjernsynet (forutsatt samme produksjonsvolum hvert år). Med en engangsinvestering på kr. 1.000.000,- kan en i tillegg kjøpe inn harddisker for å lagre siste års nyhetssendinger i MPEG-1 kvalitet.

5.1.3 Anvendelser

Digital video er i seg selv en interessant teknologi, men enda mer interessant er det å se på hva dette faktisk kan brukes til. Digital video av full studiokvalitet utgjør som nevnt så store datamengder at en i lang tid fremover må benytte vanlige videobånd eller båndautomater for dette. Derimot kan sterkt komprimert video av *påsynskvalitet* på harddisk åpne for interessante muligheter allerede i dag. "Påsynskvalitet" er ikke noe entydig begrep, men benyttes gjerne om video som har tilstrekkelig kvalitet til å gi et godt inntrykk av innholdet. Påsynskvalitets-video vil derfor være et glimrende supplement til tekstlige beskrivelser av videoinnhold. Ved søking etter arkivmateriale, omtalt i kapittel 2.1, vil det f.eks. være mulig å *vis*e videomaterialet umiddelbart, hvor en i dag må nøye seg med en tekstutskrift. Dette er ikke minst interessant for Distriktskontorene, siden videoopptak på denne måten også kan gjøres tilgjengelig over datanett.

Studiokvalitet kan reduseres til påsynskvalitet på flere måter. Komprimering ble omtalt i forrige avsnitt, og dette kan om nødvendig benyttes sammen med tre andre teknikker.

- *Lavere oppløsning*. En halvering av antall *pixels* horisontalt og vertikalt vil f.eks. redusere lagringsbehovet fire ganger.

²² JPEG er en forkortelse for Joint Photographic Experts Group.

²³ MPEG er en forkortelse for Motion Picture Experts Group.

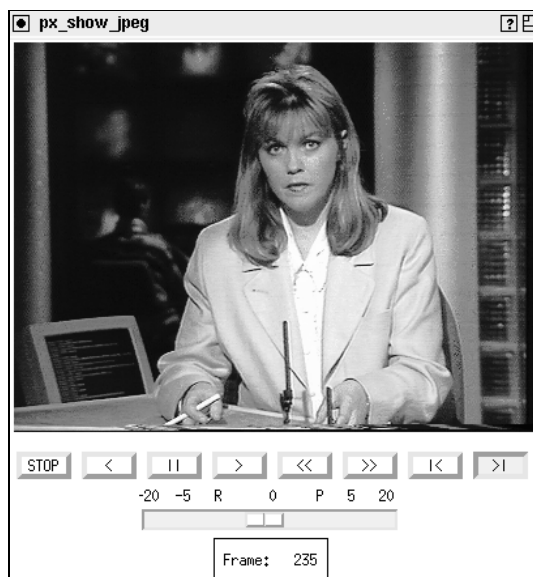
²⁴ MPEG-2 er også utarbeidet med tanke på *HDTV* (*High Definition TeleVision*).

²⁵ Dersom en ønsker å lagre alle programmer *online* på båndstasjoner (roboter), må det også kjøpes inn en ny stasjon når den forrige er full.

- *Redusert antall farger.* Ved å benytte 256 farger (8 bit) istedenfor hele fargespekteret (24 bit), blir lagringsbehovet redusert ytterligere tre ganger.
- *Redusert bilde-rate,* altså antall bilder som vises hvert sekund. Mens den europeiske TV-standarden *PAL* benytter 25 bilder/s, er det mulig å gå ned til ca. 16 bilder/s uten at en får inntrykk av ujevne bevegelser.

Disse “sparemetodene” kan kombineres, og sammen med komprimering, er det mulig å lagre store mengder video på en vanlig harddisk.

Til praktiske eksperimenter ved institutt for datateknikk og telematikk (IDT), er video digitalisert i oppløsningen 400x300 pixels med 24 bit fargegjengivelse og ca. 20 bilder/s. Med JPEG-komprimert video og ukomprimert PCM-lyd²⁶, opptar f.eks. en sending av *Ettermiddagsnytt*²⁷ (en konkret sending på 3 minutter og 35 sekunder) ca. 53 MB bilder og 1,8 MB lyd. Dette betyr at dataraten blir omkring 2Mbit/s, og for “påsynsbruk” er den subjektive bilde kvaliteten upåklagelig. Skeide utviklet høsten 1993 den digitale videospilleren som benyttes i eksperimentene [46], mens Wasskog og Bekkadal våren 1994 utvidet funksjonaliteten i grensesnittet [47]. Videospilleren kjører i et eget vindu som vist i figuren under.



Figur 19, Digital videospiller ved IDT

Spillerens trykknapper gir en funksjonalitet som også finnes på analoge videospillere: avspilling, hurtigspoling og bilde-for-bilde fremover og bakover, pause og stopp. Hurtigspolingen gir et støyfritt bilde uten de horisontale stripene en opplever med alle analoge spillere. Under trykknappene er det også lagt på en såkalt *slider*, hvor hastigheten avhenger av posisjon for denne. Når spilleren står i pause-modus, vises også *rammenummer*.

²⁶ *PCM - Pulse Code Modulation.* Nærmere detaljer er omtalt i bl.a. [45] og [46].

²⁷ Programposten *Ettermiddagsnytt* er nå utgått, men gamle sendinger er allerede digitalisert ved IDT. Disse sendingene er velegnede for testformål, siden innslagene gjerne er kortere enn i en typisk Dagsrevysending, og dermed opptar mindre plass.

Selv om digitale videoopptak åpner nye muligheter, kan de likevel ikke *erstatte* tekstbaserte innholdsbeskrivelser. Det vil fortsatt være nødvendig å kunne finne fram til videoopptakene på en enkel måte, og det er raskere å søke i en tekst enn direkte i video. Dagens datamaskiner og programvare kan heller ikke tolke digital video godt nok til at en direkte kan søke etter ønsket innhold - selv om det forskes på dette [4].

Innholdsbeskrivelser og digital video må altså inntil videre eksistere side om side. Rent praktisk kan digital video inkluderes som et *Binary Large Object (BLOB)* i en database, dersom databasesystemet støtter en slik datatype. Alternativt kan databasen inneholde referanser til digital video i separate filer [16]. I databasen lagres også relevant tekstlig informasjon om innholdet.

5.1.4 Harddiskbasert redigering

Redigering av vanlig (optisk) film er en prosess hvor filmen fysisk blir klippet opp og skjøtet sammen i riktig rekkefølge. Dersom resultatet ikke svarer til forventningene, går det an å arbeide videre med den redigerte filmen.

For videoredigering foregår “klippingen” ved at de ønskede scenene kopieres over til et annet bånd. Skal det gjøres endringer, betyr dette som regel at hele redigeringsarbeidet må gjøres på nytt. For å unngå dette, benyttes forredigeringssystemer, hvor en liste med hvert enkelt klipp blir lagret i en datamaskin. Klippelista styrer videospillene under kopieringen, og endringer kan enkelt utføres ved at en forandrer informasjonen i denne lista og starter kopieringen på nytt.

Harddiskbasert redigering er en konkret anvendelse av digital video som begynner å få stor utbredelse. Dette omtales også som *nonlineær redigering*, siden videoen som skal redigeres er digitalt lagret på harddisk. At redigeringen er “nonlineær”, betyr at redigereren har direkte tilgang til videomaterialet på harddisken. Med nonlineær redigering er det enkelt å gjøre endringer, prøve forskjellige klipp og se resultatet umiddelbart.

Siden originalopptakene blir gjort på analoge videobånd, må disse digitaliseres og lagres på harddisk før redigeringen kan begynne. Denne digitaliseringen er altså inngangsbilletten til nonlineær redigering, men samtidig en stor flaskehals, siden alt materialet må spilles av én gang i sanntid før en i det hele tatt kan begynne å redigere. Dette begrenser anvendeligheten for produksjoner hvor en har knappe tidsfrister. Nonlineær redigering har likevel fått stor utbredelse, særlig i større produksjoner med lengre tidsrammer.

Nonlineære redigeringssystemer kan klassifiseres som *off-line* eller *on-line*²⁸ [27]:

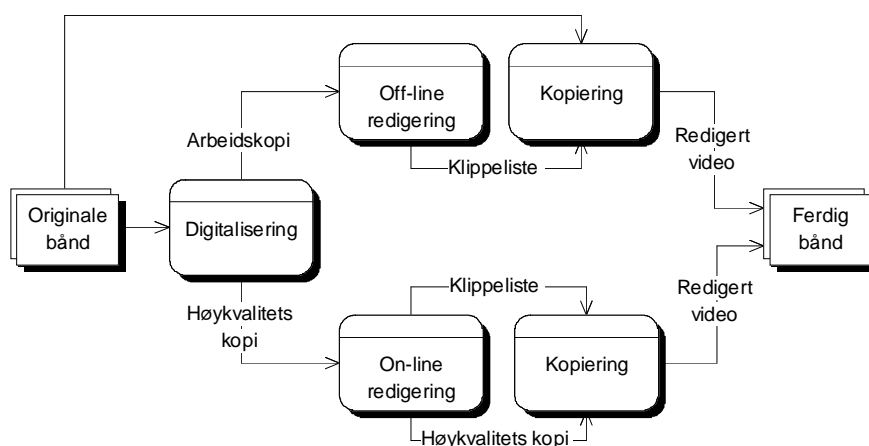
- I *off-line* systemer benyttes digital video som en arbeidskopi, og vha. redigeringssystemet utarbeides en klippeliste. Når klippelista er ferdig, styrer den kopiering av de analoge originalbåndene. Nonlineær redigering av denne typen kalles ofte *forredigering*. Det stilles ikke så strenge krav til kvaliteten

²⁸ *Nonlineær on-line redigering* må ikke forveksles med *tradisjonell on-line redigering*, omtalt bl.a. i [43], hvor redigeringen utføres direkte på *råmaterialet*.

på den digitale videoen, siden det er originalbåndene som benyttes for den endelige kopieringen. Off-line redigering er også velegnet for vanlig film, dersom det digitaliseres en arbeidskopi. Flere av de nye norske filmene er blitt redigert på denne måten.

- Også med *on-line* systemer utarbeides en klippeliste. Arbeidskopien er imidlertid her av så høy kvalitet at det ikke er nødvendig å benytte videobåndene igjen ved den endelige kopieringen. Når en er fornøyd med resultatet, kan den ferdige, digitale videoen kopieres til et bånd sammen med tekster og andre effekter som er lagt inn.

Disse to typene nonlineær redigering kan også illustreres i et dataflytdiagram:



Figur 20, Prinsippene for nonlineær redigering

Det finnes i dag en rekke kommersielle systemer for nonlineær redigering, bl.a. *AVID NewsCutter* [3, 32], *O.L.E. Lightworks* og *Heavyworks* [27, 32], *Quantel Newsbox* [32], *Sony Destiny* [28] og *Matrox Studio* [30]. Dette er systemer i prisklassen en halv til én million kroner [32], men det finnes også langt rimeligere løsninger, som f.eks. *Fast Video Machine* [29]. Dette er hovedsaklig komplette systemer som består av PC, harddisker, komprimeringskort, input-enhet ("klippebord") og programvare. Av rene programvareløsninger har bl.a. *Adobe Premiere 3.0* blitt godt mottatt [31]. De fleste systemene benytter JPEG-komprimering av video. Grensesnitt og funksjonalitet i programvaren varierer, men er som regel basert på en grafisk tidslinje med mulighet til bl.a. å vise første bilde i hvert kutt. I *Quantel Newsbox* er det også implementert en enkel *browser* (strukturert søkeverktøy).

Dagsrevyen vurderer å kjøpe inn slikt utstyr, og teknisk avdeling ved NRK har utført flere brukertester [32]. Redigererne har påpekt at slike systemer krever omstillinger i måten å arbeide på, men de ser potensialet for raskere redigering. En viktig innvending gjelder umodne input-enheter og grensesnitt, i tillegg til at systemene ikke tilbyr video av fullgod kringkastingskvalitet. Sony har uttrykt skepsis til det de kaller "utvanning av kvalitetsnormene" (definert i *CCIR601*-standarden), da bransjen ser ut til å godta lavere bildekvalitet ved bruk av slike systemer [33].

Klippelista er et biprodukt fra den nonlineære redigeringsprosessen som har interesse utover å være en ren liste over klipp. De ulike klippene er samtidig en *segmentering* av materialet, og dette er informasjon det er nyttig å ta vare på. Enkelte av de omtalte

systemene gjør det mulig å skrive inn en kort beskrivelse av hvert klipp. En slik “annotert klippeliste” er et godt utgangspunkt for senere søking etter klipp, og svært nyttig *input* til mer avanserte søkeverktøy.

5.1.5 Harddiskbaserte kameraer

Digital video er ikke noe nytt fenomen, og mange produsenter har for lengst utviklet nonlineære redigeringsverktøy. Likevel blir video fremdeles spilt inn på analoge videobånd, som først må digitaliseres. Denne digitaliseringsprosessen er som nevnt en plagsom flaskehals som forsinker videre bearbeiding og redigering. Redningen er digital innspilling (*Digital News Gathering*), og det foreligger allerede konkrete planer om fulldigitale kameraer som lagrer video direkte på harddisk [3, 35]. AVID, som i dag produserer det nonlineære redigeringsystemet *NewsCutter*, har alliert seg med kameraprodusenten *Ikegami*. De har lovet leveringsklare løsninger til neste års NAB95-messe²⁹. Også Sony arbeider med utvikling av harddiskbaserte kameraer.

På årets NAB94-messe i Las Vegas presenterte AVID og Ikegami konseptene for sitt nye harddiskbaserte kamera. Kameraet vil bestå av et bakstykke med to utskiftbare harddisker, som hver skal kunne lagre 20-25 minutter med lyd og JPEG-komprimert video av kringkastingskvalitet (komprimert 10 ganger). Det blir mulig å bytte ut den ene harddisken mens en spiller inn til den andre, slik at lange sekvenser kan spilles inn kontinuerlig. Redigeringen kan påbegynnes umiddelbart etter opptak, og om nødvendig kan opptakene overføres fra fjerntliggende steder til redigeringsstasjonen over tele-nettet.

5.2 Datamodell for videomateriale

Som det ble påpekt i avsnitt 4.1, har NRK en forholdsvis “løs” kopling mellom videomaterialet og innholdsbeskrivelsene. Innholdsbeskrivelsene i SIFT inneholder et båndnummer, og dette er eneste holdepunkt for å finne det videobåndet som inneholder ønsket videomateriale.

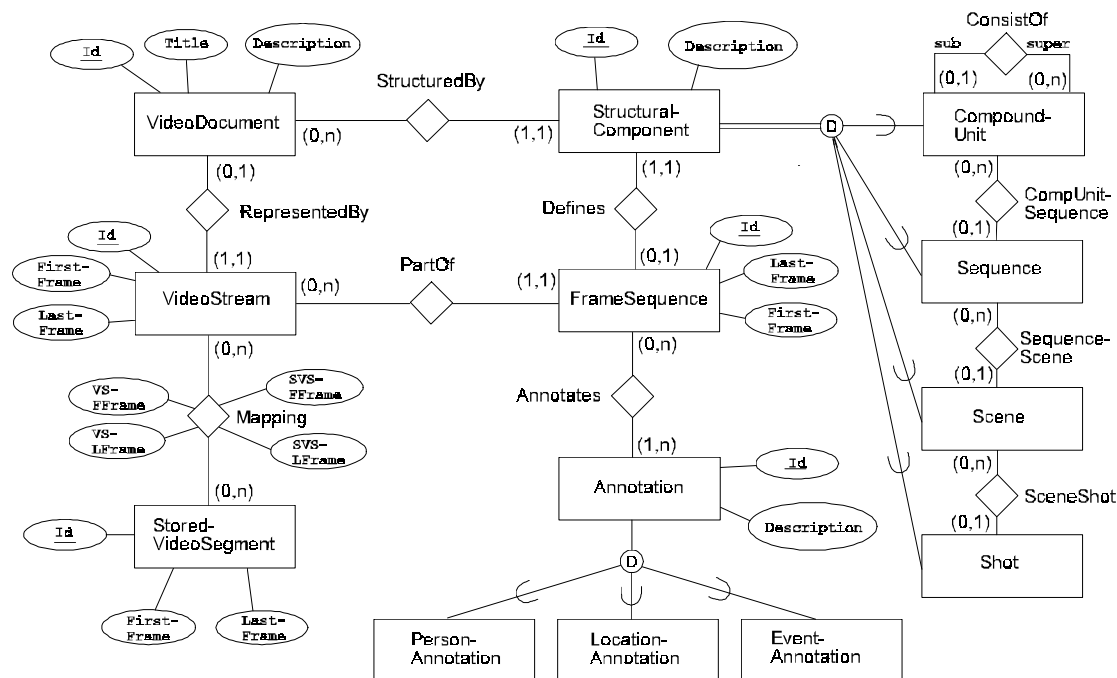
Gruppe for databaseteknikk ved institutt for datateknikk og telematikk, NTH, forsker bl.a. på alternative måter å knytte beskrivelser til videomaterialet. Hjelsvold og Midtstraum presenterer i [16] en datamodell hvor en skiller mellom videomaterialets *struktur* og *innhold*, og som dessuten muliggjør *deling* og *gjenbruk* av videomateriale. Filosofien i denne modellen ligger til grunn for de påfølgende kapitler i denne rapporten, og modellen er derfor gjengitt i sin opprinnelige form i EER-diagrammet i Figur 21. Modellen er særlig velegnet for digital video, omtalt i forrige avsnitt.

Videomaterialet er lagret digitalt i en database som grupper av påfølgende rammer³⁰ (såkalte *lagrede videosegment*, *StoredVideoSegments*). Alle rammer har et eget rammenummer, som kan sammenliknes med tidskoden på et videobånd. Dette

²⁹ NAB - National Association of Broadcasters.

³⁰ *Ramme* betegner den minste visbare enhet i en videofilm, altså et enkeltbilde. Ett sekund video av PAL-standard består av 25 rammer.

nummeret gjør det mulig å referere til hver enkelt ramme. Et *videodokument* (VideoDocument) representeres som en *videostrøm* (VideoStream), som igjen består av ett eller flere lagrede videosegment. En *rammesekvens* (FrameSequence) defineres som et intervall i en videostrøm, varierende i omfang fra en enkeltramme til et helt videodokument (f.eks. et TV-program). Rammesekvenser knytter videodokumentet til *strukturelle elementer* (StructuralComponents) og innholdsbeskrivelser, i modellen kalt *annotasjoner* (Annotations).



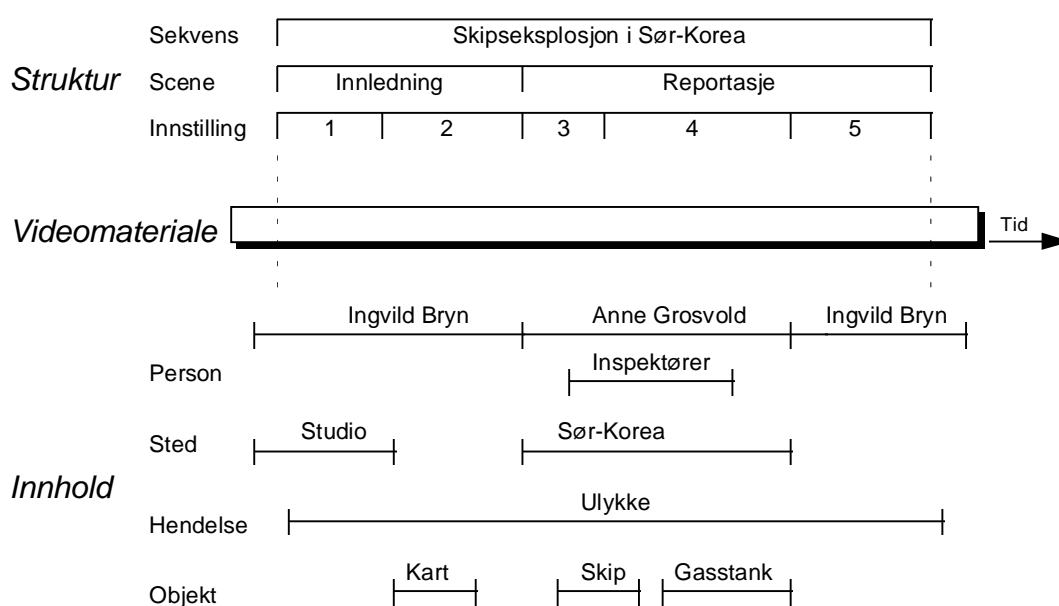
Figur 21, Datamodell for videomateriale

De strukturelle elementene i et videodokument er hierarkisk ordnet, og har fellestrekk med struktureringen av en tekst, som f.eks. en bok eller denne diplomoppgaven. En tekst kan struktureres i logisk sammenhengende deler som f.eks. kapitler og avsnitt for å bedre lesbarheten. Med direkte tilgang til digital video, blir det mulig med en tilsvarende strukturering av et videodokument, altså en slags “innholdsfortegnelse”. Modellen i Figur 21 benytter de strukturelle elementene *sekvens* (Sequence), *scene* (Scene) og *innstilling* (Shot) for dette. Innstilling betraktes som minste meningsfylte enhet, og er en samling rammer spilt inn etter hverandre uten opphold. Relaterte innstillinger kan grupperes i scener, og relaterte scener kan grupperes i sekvenser. Øverste nivå, som kan gjentas rekursivt, kalles *sammensatt enhet* (CompoundUnit). Dette kan f.eks. være et helt program, som igjen kan inngå i en serie eller bestå av flere deler.

Også innholds-annotasjonene til et videodokument finner sin parallell i vanlig tekst. På samme måte som stikkordregisteret (indeksen) bakerst i en bok refererer direkte til sidetall i teksten, knytter annotasjonene representative stikkord til bestemte rammesekvenser i videodokumentet. I datamodellen benyttes tre typer annotasjoner: *person*, *sted* og *hendelse*.

Alt videomateriale er av natur temporalt (sekvensielt), og bare meningsfylt dersom det spilles av i “riktig” rekkefølge. Selv om en registrerer struktur- og innholdsinformasjon, sørger rammenumrene for at video fortsatt kan vises sekvensielt. (Dette er analogt med at den sekvensielle strukturen i en bok ikke forstyrres av innholdsfortegnelse og stikkordregister). De to foreslåtte indekseringsmetodene åpner i tillegg for raskere og mer effektiv gjenfinning av arkivmateriale. En slik organisering av videomaterialet har derfor potensiale for å løse flere av problemene som er relatert til innholdsbeskrivelser (omtalt i avsnitt 4.1). Datamodellen blir derfor en viktig komponent når det senere (i kapittel 6) utarbeides forslag til nye støtteverktøy for et Digitalt Fjernsynsarkiv.

For å illustrere bruken av modellen, kan en se nærmere på et konkret eksempel fra den nå utgåtte programposten Ettermiddagsnytt. Figur 22 viser en forenklet utgave av innslaget “Skipseksplasjon i Sør-Korea” fra Ettermiddagsnytt 29.november 1993.



Figur 22, Videomateriale med struktur og innhold

Videomaterialet er vist som en rekke sammenhengende rammer midt i figuren. Grunnet figures målestokk, vises bare ett innslag (sammen med noen tilstøtende rammer før og etter innslaget), og de enkelte rammer er ikke tegnet inn. Tidsaksen er dessuten vilkårlig skalert. Med datamodellens terminologi, kan en betrakte videomaterialet som en videostrøm eller en rammesekvens (like lang som hele sendingen).

Innslagetets struktur er tegnet inn i øvre del av figuren. Fordelingen av de ulike strukturelle elementer avhenger av programtypen, og titlene (sekvens, scene og innstilling) kan evt. byttes ut med begreper som best mulig beskriver den aktuelle programtypen. For nyhetssendinger er det f.eks. naturlig å betrakte et innslag som en *sekvens*. Innslaget i figuren består av to *scener*, hvor den første er en innledning med lesning av studiokommentarer og den andre er en reportasje fra Sør-Korea. Hver av scenene kan igjen deles opp i mindre *innstillinger*, som i figuren er indikert med nummer. Med denne hierarkiske struktureringen inngår hver ramme i én sekvens, én scene og én innstilling.

Nederst i figuren er videomaterialets innhold skissert som annotasjoner. De tre annotasjonstypene modellen foreslår (*person*, *sted* og *hendelse*), er å betrakte som eksempler. En kan definere andre annotasjonstyper som best mulig passer programtypen og brukers spesifikke behov. F.eks. kan en i tillegg beskrive *objekter*, som vist i figuren, og evt. andre annotasjonstyper (eksempelvis de som ble vist i datamodellen for dagens Fjernsynsarkiv i Figur 8). I motsetning til den hierarkisk ordnede strukturinformasjonen, kan innholds-annotasjoner knyttes til vilkårlige rammesekvenser. Annotasjoner av samme type kan overlape, og det kan forekomme rammer som ikke er annotert i det hele tatt. I reportasjen i Figur 22 kommenterer f.eks. Anne Grosvold et innslag samtidig som skipsinspektører forekommer i bildet.

5.3 Tilleggsmodell for innholdsbeskrivelser

NRK benytter i dag SIFT som plattform for sin Fjernsynsarkiv-database, som nevnt i kapittel 2. Et tilbakeblikk på NRKs SIFT-innholdsbeskrivelser vist i Figur 16, sammenliknet med datamodellen i Figur 22, viser to fundamentalt forskjellige måter for beskrivelse av videoinnhold. For det første blir strukturinformasjon eksplisitt uttrykt i sistnevnte modell. Dessuten refererer både struktur og innhold direkte til den delen av videomaterialet som beskrivelsen gjelder. Beskrivelsene i SIFT refererer derimot til båndet hvor videomaterialet ligger lagret, og bare unntaksvis er tidskode angitt.

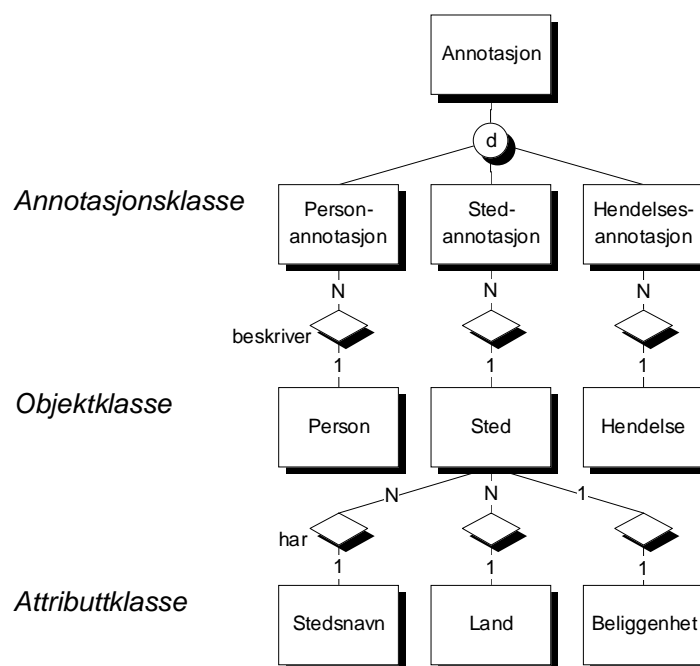
Datamodellen i Figur 22 er også et godt utgangspunkt for en alternativ modellering av innholdet. Annotasjonene kan betraktes som et grensesnitt (“koblingspunkt”) mot innholdsbeskrivelser lagret som selvstendige objekter³¹. Et enkelt forslag til utvidelse av datamodellen, hvor innholdet modelleres eksplisitt med egne *objektklasser*, er vist i Figur 23.

Entiteten *Annotasjon* med de tre spesialiserte *annotasjonsklassene* fra den opprinnelige datamodellen er tatt med her. Som et eksempel er det tatt med tre objektklasser (*Person*, *Sted* og *Hendelse*), som passer til hver av de tre annotasjonsklassene. En *Person*annotasjon *beskriver* ett personobjekt, og et personobjekt kan bli beskrevet av mange *Person*annotasjoner (og dermed forekomme i mange rammesekvenser). Tilsvarende argumentasjon for *sted*- og *hendelses*objekt forklarer hvorfor relasjonen er av typen én-til-mange.

Hver objektklasse har en rekke attributtklasser, som for *Sted* er illustrert med *Stedsnavn*, *Land* og *Beliggenhet* i figuren. For *Person* ville det være naturlig å bl.a. ta med attributten *Avbildet* (med verdien “Ja” eller “Nei”), siden dette i dag legges inn som implisitt informasjon (avsnitt 2.1.1). Relasjonene mellom objekt- og attributtklassene kan være én-til-én (1:1), én-til-mange (1:N) eller mange-til-mange (M:N). Et sted har f.eks. bare én geografisk beliggenhet, og andre steder kan ikke ha nøyaktig samme beliggenhet (1:1). Et sted har som regel bare ett navn, mens også andre steder kan ha samme navn (1:N). Det kan også tenkes at en person har flere yrker, samtidig som et yrke kan innehas av flere personer (M:N).

³¹ Begrepet *objekt* brukes her i betydningen “data av samme type”, og antyder *ikke* en objektorientert modelleringsmetode. Dette ble kalt *innholdsenhet* i datamodellen for NRKs Fjernsynsarkiv i Figur 8.

Med denne utvidelsen av modellen, kan en f.eks. representere Oslo som et eget stedobjekt. Dette stedobjektet har stedsnavnet “Oslo”, ligger i landet “Norge” og har en beliggenhet som f.eks. kan angis med geografiske koordinater. Når Oslo forekommer i et bestemt nyhetsinnslag, kan det opprettes en stedannotasjon som knyttes til aktuell rammesekvens i videostrømmen. Denne stedannotasjonen kan evt. inneholde en nærmere presisering av sted i Oslo, f.eks. “Stortinget”.



Figur 23, Datamodell for innholdsbeskrivelser

SIFT er basert på fritekstdokumenter, og derfor ingen egnet plattform for lagring av informasjon om videostruktur og strukturerte innholdsbeskrivelser. Relasjonsdatabaser er et alternativ.

5.3.1 Relasjonsdatabaser vs SIFT

Av de ulike databasesystem som tilbys, er relasjonsdatabasesystem i dag en klar markedsleder. Noen årsaker til den store utbredelsen er bl.a. rask og enkel databaseutvikling og stor fleksibilitet. Dessuten har det de siste årene kommet en rekke implementasjoner for mindre datamaskiner og PC'er.

I relasjonsdatabaser blir data lagret som atomiske verdier i tabeller (relasjoner). Søk etter data kan utføres i en bestemt tabell eller flere tabeller i kombinasjon. For søking er det vanlig å benytte det strukturerte spørrespråket *SQL (Structured Query Language)*. Brukeren kan likevel skjermes mot dette spørrespråket gjennom grafiske skjermbilder med f.eks. trykknapper og menyvalg.

I motsetning til SIFT, hvor dokumentene lagres på fri form, er relasjonsdatabaser konstruert med tanke på organisering og strukturering av data. Ved innleggelse er det dessuten mulig å kontrollere data på ulike måter, slik at sannsynligheten for feilregistrering elimineres eller ihvertfall blir svært lav.

Statens Datasentral (SDS) markedsfører *søkehurtighet* som SIFT's store styrke. Uten struktur på registrerte data, er man da også helt avhengig av hurtige søk. Relasjonsdatabaser strukturerer data på en slik måte at søkeområdet kan begrenses til de tabeller hvor aktuelle data befinner seg. Kombinering (såkalt *joining*) av flere store tabeller ved søk er relativt ressurskrevende, men med gode indekseringsmuligheter kan en likevel oppnå akseptable responstider. Også i relasjonsdatabaser er det mulig å søke etter fritekst (i tabellene), nærmere bestemt med operatoren *LIKE* og bruk av jokertegn³². (Det er vanlig at '%' eller '*' erstatter et vilkårlig antall tegn og at '_' erstatter ett tegn [14]). Ved å kombinere flere søkeord, kan en oppnå samme funksjonalitet som i SIFT.

SIFT er et glimrende system for hurtig søking i tekstdokumenter, men ble ikke primært utviklet med strukturert informasjon i tankene. (Se evt. også omtale av SIFT i Vedlegg A). Relasjonsdatabaser er særlig velegnet nettopp for systematisk organisering av strukturert informasjon, og de implementerte databasene er dessuten fleksible med tanke på vedlikehold og endringer. Flere leverandører tilbyr også etter hvert relasjonsdatabasesystemer med egne video-datatyper for integrert lagring av video i databasen. Oracle har f.eks. med sin Oracle Media Server blitt foretrukket som databaseleverandør i en rekke praktiske forsøk med video-på-forespørsel (eng. *video-on-demand*) [42].

5.3.2 Fra tekstlige til strukturerte innholdsbeskrivelser

En kan ikke komme utenom det faktum at NRK allerede har registrert over 126.000 poster (videodokumenter) med innholdsbeskrivelser i SIFT. Spørsmålet for NRK er derfor hvordan en kan utnytte all denne informasjonen best mulig, enten en beholder SIFT-systemet eller på sikt går over til strukturerte beskrivelser i en relasjonsdatabase.

De fritekstlige beskrivelsene som i dag ligger i SIFT, er altså ikke tilstrekkelig for å tilby den funksjonaliteten datamodellen i Figur 21 åpner for. Som et minimum kreves et nivå som *mapper* (relaterer) deler av teksten til deler av videomaterialet. En mulig strategi, er å fortsette med innholdsbeskrivelser i SIFT og utnytte SIFTs indekseringskraft for å koble disse til strukturerte objektklasser i en relasjonsdatabase³³.

Det er mulig å benytte datakraft for å *parse* innholdsbeskrivelsene, altså lete etter kjente ord i teksten og opprette referanser til den objektklassen ordet tilhører. Dersom en starter med tomme objektklasser, altså tabeller uten innhold, må disse fylles med data på forhånd eller samtidig som parsingen pågår. En kan da ignorere alle utfyllende ord som ikke tilfører innholdsinformasjon (f.eks. "og, men, for, vi ser bilder fra..."). For hver objektklasse kan det defineres passende attributtklasser, og det kan legges inn attributter (attributtklassens verdier) for hver av disse. Dette arbeidet

³² Følgende SQL-setning vil f.eks. finne ordene "rødvin", "vindu" og "pingviner": `SELECT ord FROM innhold WHERE ord LIKE '%vin%'`. Nærmere beskrivelser kan finnes i fagbøker om SQL og f.eks. [14].

³³ Som nevnt i Vedlegg A, er det mulig å benytte SIFT for å indeksere eksternt lagrede filer. Denne muligheten er likevel ikke undersøkt nærmere i forbindelse med dette diplomarbeidet.

må utføres manuelt, men er da gjort en gang for alle (selv om det kan være nødvendig å endre verdiene senere).

Det er *ikke* mulig å “importere” alle gamle data fra SIFT til en ny relasjonsdatabase uten videre, eller å få generert referansene helt automatisk. Dette skyldes at SIFT-beskrivelsene ikke inneholder nødvendig semantisk informasjon om innholdet (kontekstuell informasjon eller *metadata*). Ikke noe sted er det f.eks. indikert at Harald er en person, at Oslo er et sted eller at eksplosjon er en (forholdsvis dramatisk) hendelse. Dette betyr at parsingen må kontrolleres av mennesker som kjenner betydningen av de ulike ordene, f.eks. Fjernsynsarkivets arkivarer. Alternativt kan utfyllende tekst også legges inn som annotasjoner og redigeres manuelt i ettertid. Etter hvert som objektklasse-tabellene fylles med data, kan parsingen av SIFT-beskrivelser gradvis automatiseres. I tvilstilfeller må likevel arkivarene gjøre det klart hvilken objektklasse et ord tilhører, f.eks. om en bestemt forekomst av ordet Washington gjelder den amerikanske byen (sted), staten (sted) eller presidenten (person).

På denne måten er det altså mulig å utnytte fordelene ved strukturert innholdsindeksring av digitalt videomateriale uten å måtte forkaste de mange innholdsbeskrivelsene som allerede eksisterer. Dessuten vil en gjennom konverteringsfasen samtidig ha generert et rammeverk som letter fremtidig overgang til et rendyrket relasjonsdatabasesystem, dersom dette er ønskelig.

6. Digitalt Fjernsynsarkiv

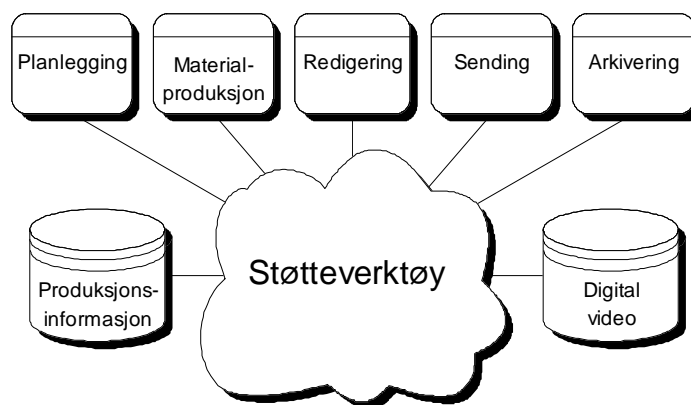
Et datasystem som benytter digitale innholdsbeskrivelser og digital video for å støtte virksomheten i et fjernsynsarkiv, kan ganske betegnende kalles et *Digitalt Fjernsynsarkiv*. Bakgrunnen for at et slikt system i dag er mer aktuelt enn noen gang, er forsøkt indikert i flere sammenhenger tidligere i denne rapporten. Utbredelsen av digital video er en grunnleggende forutsetning, og som beskrevet i avsnitt 5.1, er denne teknologien for lengst i ferd med å forandre fjernsynsmediet. NRK planlegger f.eks. i løpet av de nærmeste årene å gå over til å benytte digital, kringkastingskvalitets video under produksjonsarbeidet [34]. Forskning på beskrivelse av innhold og struktur for slik digital video, bl.a. rapportert i [4], [16] og [38] og omtalt i avsnitt 5.2, har dessuten gitt resultater en kan utnytte ved utviklingen av et slikt Digitalt Fjernsynsarkiv.

“Digitalisering” av gamle, manuelle prosesser og arkiv er ikke i seg selv en garanti for vellykkede løsninger. Et Digitalt Fjernsynsarkiv åpner for nye, mer effektive måter å arbeide på, men dette forutsetter at en klarer å frigjøre seg fra “inngrodde arbeidsvaner”. Det er nyttig å igjen rette fokus på verdiskapningen i bedriften, siden dagens eksisterende løsninger gir verdifull kunnskap som kan utnyttes i et nytt system.

Dette kapittelet vender derfor tilbake til produksjonskjeden for fjernsynsnyheter i NRK, for å se hvilken informasjon som blir generert i de i ulike prosessene. Samtidig blir det vurdert hvordan en på beste måte kan beskrive og ivareta denne informasjonen i et Digitalt Fjernsynsarkiv.

6.1 Arkitektur

Følgende arkitektur ligger til grunn for det Digitale Fjernsynsarkivet som blir beskrevet i de påfølgende avsnitt:



Figur 24, Arkitektur for Digitalt Fjernsynsarkiv

Produksjonskjeden består av de fem (etter hvert velkjente) aktivitetene øverst i figuren. All informasjon om en produksjon blir lagret i en sentral database og gjort tilgjengelig

for alle deltakere på alle trinn i produksjonskjeden. Videre benyttes det digital video³⁴ i alle ledd i produksjonskjeden, og videomaterialet blir gjort tilgjengelig gjennom lagring på harddisk og distribusjon over datanett. Et eller flere støtteverktøy knytter sammen de ulike prosessene og databasene. Enten video og data lagres separat eller i samme database, blir resultatet det samme, nemlig at alle arbeider mot felles data og video.

Med tilgang til sentralt lagret data og video over datanett, kan støtteverktøyene også benyttes ved distriktskontorene. (Enkle støtteverktøy for samarbeid mellom geografisk adskilte produksjonsteam er omtalt bl.a. i [5]).

Databasene og produksjonsaktivitetene blir beskrevet i neste avsnitt, mens støtteverktøy er temaet i avsnitt 6.3.

6.2 Informasjon fra produksjonskjeden

For å kunne utarbeide effektive støtteverktøy, er det nyttig å oppsummere hvilken informasjon som faktisk blir generert i hver av disse prosessene. Det er viktig å presisere at selv om informasjon blir *generert*, betyr ikke dette nødvendigvis at den blir *bevart og foredlet* i dag. Likevel vil en slik oppsummering klargjøre potensialet for nye støtteverktøy og være et utgangspunkt for forbedringer.

Datamodellen i Figur 21 er som nevnt et velegnet utgangspunkt for et Digitalt Fjernsynsarkiv. Derfor vil den videre fremstillingen relatere informasjonen fra produksjonsprosessene til konseptene i denne datamodellen³⁵ samt tilleggsmodellen i Figur 23. Der det er nødvendig, vil det dessuten innføres supplerende entiteter og relasjoner.

6.2.1 Planlegging

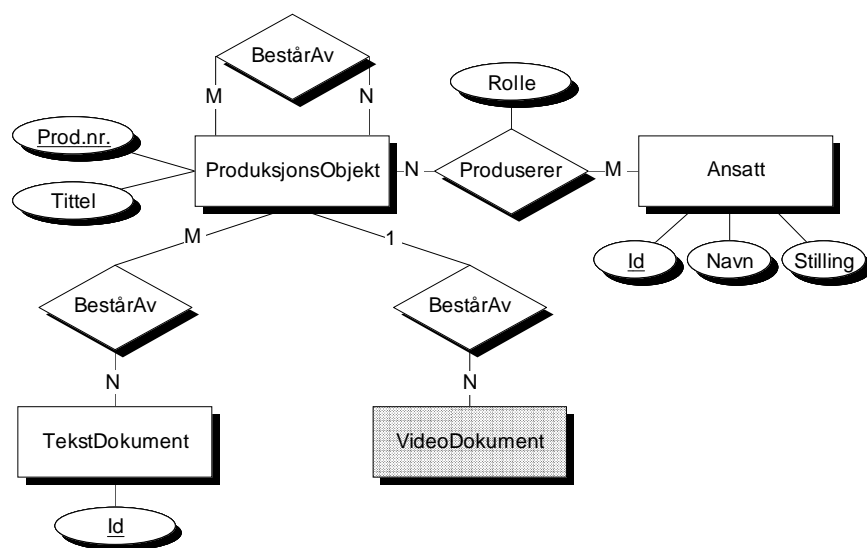
Det kan opprettes et `ProduksjonsObjekt` for en planlagt produksjon (f.eks. nyhetsinnslag eller nyhetssending). Denne entiteten, vist i ER-modellen i Figur 25, kan betraktes som et “øverste nivå” som knytter sammen all informasjon om produksjonen. Produksjonsnummeret identifiserer denne entiteten. Medlemmene av *produksjonsteamet* (`Ansatte`) representeres som egne entiteter og har en bestemt rolle under arbeidet med et `Produksjonsobjekt`. Denne informasjonen, altså hvem som skal delta i produksjonen og deres oppgave, bestemmes typisk i løpet av planleggingsfasen. Det tidligere omtalte `VideoDokument` er “tilkoblingspunktet” mot resten av modellen, og er derfor skravert i ER-diagrammet³⁶. Før en kan legge inn nye opptak i

³⁴ Bruk av digital video og en vurdering av konsekvensene ble omtalt i kapittel 5.

³⁵ Navn på entiteter og relasjoner blir imidlertid oversatt til norsk for at fremstillingen skal få et konsistent, enhetlig preg.

³⁶ Bare en del attributter er vist. Det er også mulig å inkludere entiteter, relasjoner og attributter i datamodellen for NRKs Fjernsynsarkiv (Figur 8), bortsett fra *innhold* og *innholdsenhet*, som heller bør knyttes direkte til vedkommende rammesekvens.

database, må det opprettes et slikt VideoDokument, som kan sies å være “tomt” inntil opptakene blir lagt inn.



Figur 25, Produksjonsrelatert informasjon

Et ProduksjonsObjekt kan også bestå av flere ProduksjonsObjekter. Eksempelvis inneholder Dagsrevyen mange separate innslag, og programserier inneholder en rekke episoder. Dersom en vet hvilken sending et nyhetsinnslag skal inngå i, kan vedkommende ProduksjonsObjekt kobles direkte til dette. Den opprinnelige datamodellen (Figur 21) uttrykker gjenbruk av videomateriale gjennom overlappende RammeSekvenser. Med konseptene vist i ovenstående figur, kan slik gjenbruk i tillegg representeres mer eksplisitt ved at et ProduksjonsObjekt BestårAv de VideoDokumentene som det (gjen)bruker.

Til ProduksjonsObjekter kan det også knyttes TekstDokumenter, f.eks. kommentarer og studiol kommentarer. Annen informasjon som har betydning for produksjonen kan f.eks. være avtalebøker, telefonlister og kontrakter (for større produksjoner).

Kjøreplanen slik en kjenner den fra Dagsrevyen i dag, vil uttrykkes som en kombinasjon av de viste entiteter og relasjoner samt informasjon som er relatert til VideoDokumentet. Sammensetningen av innslag vil f.eks. uttrykkes med den rekursive relasjonen BestårAv mellom ulike ProduksjonsObjekt, og produksjonspersonell er eksplisitt modellert som egen entitet. Dagens kjøreplan inneholder også informasjon om varigheten til hvert innslag. Slik temporal (tidsrelatert) informasjon gjelder egentlig videomaterialet, og det er naturlig å uttrykke dette i den opprinnelige datamodellen.

Disse endringene gir derfor en velstrukturert kjøreplan, men det må understrekes at det er *datamodeller* som her diskuteres. Det er fremdeles mulig å skrive ut en kjøreplan lik den som benyttes i dag. For andre programmer gjelder det samme for *programstatistikken*, da denne i hovedsak inneholder samme type informasjon som kjøreplanen.

På reportasjenivå blir det dessuten klart hvilket *innhold* en tenker seg f.eks. i en nyhetsreportasje. Sett i lys av datamodellen for innholdsbeskrivelser i Figur 23, betyr dette at en planlegger hvilke objekter som skal inngå i reportasjen. En stiller seg bl.a. følgende spørsmål som samtidig velger ut de aktuelle objekter for reportasjen:

- Hva reportasjen skal handle om, dvs. hvilke *hendelser* som skal dekkes.
- Hvor det foregår noe aktuelt, dvs. hvilke *steder* som skal inngå i reportasjen.
- Hvem som skal medvirke, altså hvilke *personer* som skal delta i reportasjen.

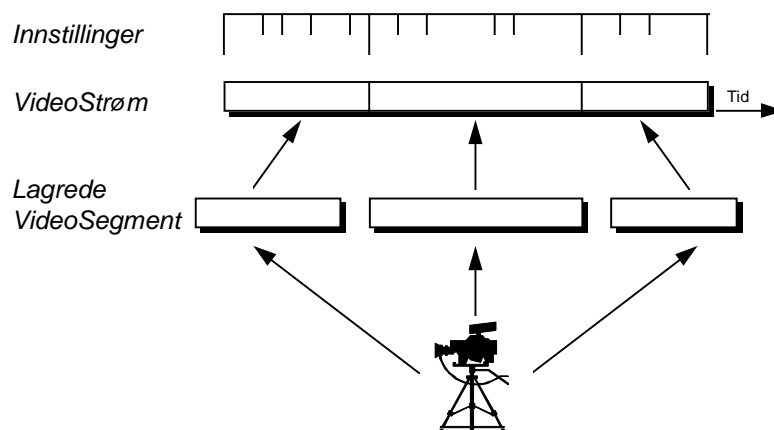
Planleggingen gir altså svar på disse spørsmålene, og dermed blir det klart hvilket *arkivmateriale* en skal søke etter og hvilke nye opptak en senere må foreta. Siden objektene velges fra databasen, kan det automatisk utføres syntaktisk og semantisk kontroll (hhv. stavekontroll og kontroll av at “riktig” begrep benyttes). For en konkret reportasje kan en f.eks. tenke seg at følgende objekter er relevante:

- *Hendelse*: EU-debatt, fiskeriforhandlinger, intervju, fisketur
- *Person*: Jan Henry T. Olsen, Gro Harlem Brundtland, fisker
- *Sted*: Stortinget, Lofoten, Brussel

Reporteren som arbeider med en reportasje har førstehånds informasjon, og derfor best forutsetning for å registrere riktige opplysninger. Imidlertid er det ikke uvanlig at opprinnelige planer blir endret. I slike tilfeller må en endre objektene tilsvarende.

6.2.2 Materialproduksjon

Eventuelle `TekstDokument` som produseres i denne prosessen, kan relateres til resten av produksjonsinformasjonen, som forklart i forrige avsnitt. Dersom separate system som `Basys` benyttes, kan det opprettes referanser til de eksterne filene.



Figur 26, Videomateriale - fra opptak til segmentering

Mer interessant i denne sammenheng, er de nye *video-opptakene* som produseres med tanke på senere bruk i reportasjen. I samme øyeblikk som kameraet fanger opptakene, blir hver ramme tildelt en tidskode. Analog video som spilles inn på videobånd må digitaliseres, og tidskoden må da ivaretas eller *mappes* over til et tilsvarende internt format. Den digitale videoen lagres i databasen som `LagredeVideoSegment` (se evt. Figur 21), og en samling av slike segment kan altså representeres som en `VideoStrøm`. Grunnlaget er dermed lagt for å koble opptakene til et Video-

Dokument (og dermed det tidligere omtalte `ProduksjonsObjekt`) og til `RammeSekvenser`. Denne sammenhengen er illustrert i Figur 26.

Figuren viser samtidig at videomaterialet er segmentert i *innstillinger*. Denne “fin-segmenteringen” kan utføres automatisk ved å benytte absolutt tid (tid på døgnet) som tidskode for opptakene. En diskontinuitet i tid indikerer et kutt, altså et skille mellom to innstillinger. På denne måten kan en “abstrahere seg opp” ett nivå i videostruktur allerede når videomaterialet legges inn i databasen, og slipper å definere kuttgrenser mellom innstillinger senere. Segmentering på grovere granularitetsnivå (sekvenser og scener) kan vente til redigeringsprosessen.

Også når det gjelder informasjon om *innholdet*, har en på dette tidspunkt mer detaljerte og nøyaktige opplysninger. Det er dermed mulig å oppdatere objektene som ble definert i planleggingsfasen og registrere tilleggsinformasjon. I nær fremtid kan en forestille seg løsninger hvor harddiskbaserte kameraer kan levere tidskode til en medbrakt bærbar PC på *location*³⁷, slik at det er mulig å registrere informasjon om innholdet mens opptaket pågår eller umiddelbart etterpå. Ved “hjemkomst”, kan registrert informasjon overføres til databasen for videre bearbeiding.

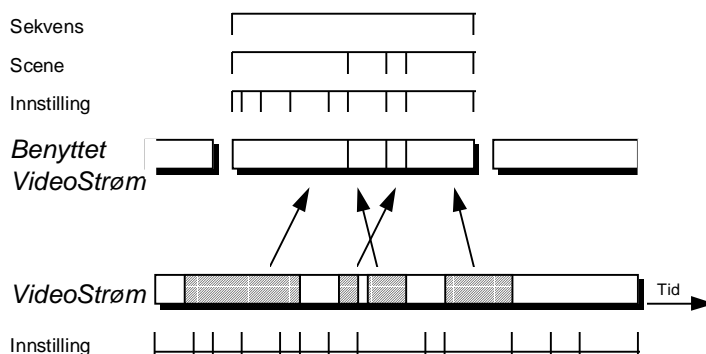
Av mer eksotiske løsninger, kan en f.eks. på kameraene montere sensorer som benytter det globale posisjoneringssystemet GPS for å logge geografisk posisjon ved opptak. Koblet til et geografisk informasjonssystem (GIS), kan en f.eks. automatisk lagre navnet på stedet hvor opptaket ble utført, samt opprette relasjoner til riktig sted-objekt i databasen. Sensorer kan også detektere kamerabevegelser (*panorering*, *tilting* osv.) og *zooming*, som dermed gir filmteknisk tilleggsinformasjon om opptakene. Flere forskere prøver å trekke ut informasjon om kutt, zooming og kamerabevegelser gjennom digital bildebehandling av videomaterialet i ettertid [4]. Det virker likevel mer nærliggende å loggføre slik informasjon på en enkel måte mens opptaket utføres, fremfor å bruke mye regnekraft på å gjøre det i ettertid - særlig når informasjonen bare blir riktig i 90% av tilfellene [38].

6.2.3 Redigering

Mens videomaterialet hittil har bestått av en `VideoStrøm` med alle opptak til f.eks. et nyhetsinnslag, blir det i redigeringsprosessen bestemt hvilke *Innstillinger* som skal benyttes i den ferdige produksjonen. De *benyttede* delene av `Videostrømmen` definerer en annen (logisk) `Videostrøm`, og er vist som skraverte `RammeSekvenser` i Figur 27.

Figuren viser et forenklet eksempel hvor fire `RammeSekvenser` (deler av `VideoStrømmen`) benyttes. Den første av disse delene består av fem *Innstillinger*, mens de øvrige består av én. Som det fremgår, kan en også benytte *deler* av *Innstillinger* i et ferdig innslag.

³⁷ Det kan kanskje virke upraktisk å dra med seg en bærbar PC i tide og utide, men med alt utstyr et reportasjeteam ofte sleper omkring, skulle ikke en liten PC gjøre noe stort utslag...

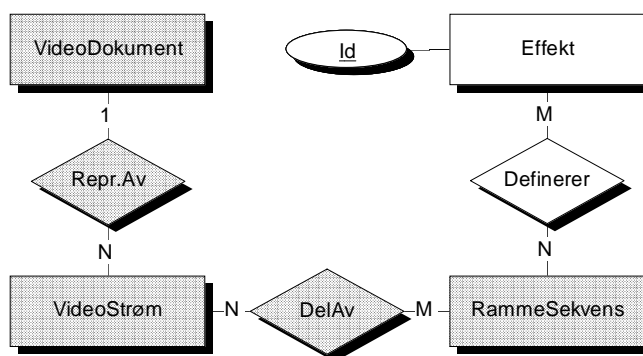


Figur 27, Redigering av innstillinger i en videostrøm

Siden alle Innstillinger allerede er definert i forrige prosess før redigeringen begynner, er det mulig å utvikle redigeringsverktøy som utnytter dette. Dersom informasjon om innholdet er registrert under eller etter opptak, kan dette benyttes for søke etter relevante innstillinger. I stedet for tidkrevende leting og spoling, kan redigereren hoppe direkte til ønskede Innstillinger og konsentrere seg om det kreative arbeidet.

Start- og sluttrammenummer for de benyttede RammeSeksvensene kan benyttes for å gruppere sammenhørende Innstillinger i Scener, som vist i Figur 27. Imidlertid er det ikke sikkert at Scene-segentering fremkommet på denne måten tilfører så mye informasjon. De tre siste Scenene i figuren inneholder f.eks. bare (deler av) én Innstilling hver, slik at de to segmenteringsnivåene blir helt identiske. Redigereren kan evt. definere Scenene slik at de uttrykker logisk sammenhørende konsepter. Dette er ikke noe omfattende arbeid, siden det bare er et spørsmål om å gruppere allerede definerte Innstillinger. En Sekvens er en RammeSeksvens som dekker hele det ferdigredigerte innslaget, og innslaget altså er ferdig segmentert på tre nivåer når redigeringsjobben er utført.

Et VideoDokument kan representeres av flere VideoStrømmer, f.eks. én for råopptakene og en annen for et ferdig redigert innslag. En kan også redigere flere alternative versjoner av et innslag, som alle benytter det samme fysiske lagrede materialet, men satt sammen på forskjellig måte. (Dette er et eksempel på såkalte virtuelle produksjoner, omtalt i [2]). Et utdrag av tidligere omtalte datamodell, gjengitt i figuren under, viser dette.



Figur 28, Utdrag av datamodell for videomateriale

Relasjonen `DelAv` mellom `VideoStrøm` og `RammeSekvens` kan sammenliknes med *klippelistene* som tradisjonelt blir generert i redigeringsystemer (omtalt i avsnitt 5.1.4). Denne relasjonen er også illustrert med pilene i Figur 27, og angir altså de enkelte “klipp” som skal skal inngå i det ferdigredigerte innslaget.

`RammeSekvenser` kan pålegges effekter, f.eks. ned- eller opptoning (*fade*), eller pålegging av *grafikk* [12]. Alternativt kan en effekt være en overgang mellom to påfølgende `RammeSekvenser`, f.eks. overgang i et bestemt mønster (*wipe*) eller gradvis overtoning (*dissolve*). Endelig kan effekter inkludere mange `RammeSekvenser`, f.eks. ved at de vises samtidig i adskilte *vinduer* i bildet. Hver enkelt `Effekt` definerer, i likhet med strukturert digital video, en `RammeSekvens` og lagres i databasen som `LagredeVideoSegment`. Effektene må da genereres og lagres i forbindelse med redigeringen, og en “effektliste” er i Figur 28 modellert med entiteten `Effekt`.

Supertider er i dag inkludert som meldinger i Dagsrevy-kjøreplanen. Manuell håndtering av supertider er ikke lenger nødvendig dersom en lagrer `Effekter` i databasen. Mengden av de enkelte `Effekter` utgjør en *effektliste* som også kan inneholde informasjon om f.eks. tekst, grafikk og titler som skal genereres og legges på under sending.

6.2.4 Sending

Før sending av Dagsrevyen er det mulig å benytte tidligere registrert informasjon for å skrive ut en kjøreplan. Det er altså mulig å gjennomføre direkte-sendinger på samme måte som i dag. Med all tidligere omtalt informasjon lagret i en database, har en imidlertid gode forutsetninger for å automatisere sendeavviklingen³⁸.

Strukturen i sendingen (hvilke innslag den inneholder), er uttrykt som relasjoner mellom de ulike `ProduksjonsObjektene`. Siden disse er direkte koblet til relevante `VideoDokument`, kan de ulike innslagene (`VideoStrømmene`) spilles av i planlagt rekkefølge. Selv med analoge videobånd pålagt tidskode kan slik informasjon benyttes for å starte og stoppe avspillingen til rett tid. Som nevnt i avsnitt 5.1, har ikke dagens harddisker tilstrekkelig båndbredde for å spille av ukomprimert video av kringkastingskvalitet. Komprimert video kan imidlertid benyttes allerede i dag.

Studiokommentarer er lagret som `TekstDokument`, og disse kan legges opp på lesemaskinen til rett tid. Før en evt. får gode systemer for *talegjenkjenning*, er det likevel (som i dag) nødvendig at en operatør manuelt styrer progresjonen, tilpasset programleders lesehastighet.

Selv om ferdige innslag er segmentert allerede før sending (Figur 27), gjenstår logging av tider for direkte-sendte studiokommentarer mellom innslag. For dette formålet må det benyttes en “global” tidskode som gjelder for *sendingen*. F.eks. kan sensorer på

³⁸ Det har fra NRK-hold vært uttrykt en viss skepsis til automatisert sendeavvikling, bl.a. fordi man selv vil ha “full kontroll” over alle hendelser. I en teknisk rapport som denne, er det likevel naturlig å påpeke de muligheter som eksisterer og evt. kan innføres på et senere tidspunkt.

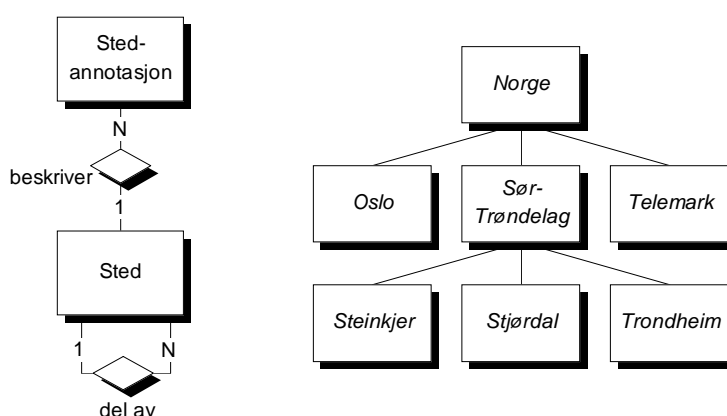
miksebordet automatisk registrere start- og sluttid for direktesendte studioinnslag. Ved også å lagre direktesendt materiale i databasen og dessuten registrere “globale” start- og sluttider for hvert ferdiglaget opptak, har en nok informasjon til å kunne referere til den ferdige sendingen i ettertid. Nærmere bestemt innebærer dette at en f.eks. kan spille av eller søke i sendingen også slik den ble kringkastet.

For andre programtyper enn Dagsrevyen, må det i dag skrives programstatistikk. Dette er en jobb mange misliker, og som derfor utsetter i det lengste. Informasjonen som i dag skrives inn i programstatistikken, er med det Digitalte Fjernsynsarkivet allerede loggført i løpet av de foregående produksjonsprosessene. Dersom en liker produksjonsarbeid bedre enn papirarbeid, vil det trolig vekke begeistring dersom en slipper å skrive programstatistikk.

6.2.5 Arkivering

I avsnitt 4.3 ble det påpekt at Fjernsynsarkivet i dag får svært begrenset informasjon om programmene som skal arkiveres, og at informasjonen ofte er mangelfull eller inneholder feil. Mye tid går også med til å regenerere informasjon som tidligere eksisterte, men har gått tapt på veien. Et Digitalt Fjernsynsarkiv endrer på dette. Produksjonsinformasjonen blir registrert i prosessen hvor den oppstår, og følgelig er mest korrekt. Fjernsynsarkivet vil motta samme informasjon som i dag, men på kjøpet får en nyttig strukturinformasjon som hittil aldri har blitt registrert. Gode støtteverktøy vil også redusere mulighetene for feil beskrivelser av innhold.

Hovedhensikten med arkiveringsprosessen er, og vil fortsatt være, å sørge for at en på best mulige måte kan søke etter (og gjenfinne) arkiverte programmer. Siden en ved dette tidspunkt allerede har mye informasjon om ProduksjonsObjektet (fra de foregående prosessene), kan arkivarene konsentrere seg om å kontrollere at innholdsbeskrivelsene og evt. registrere tilleggsinformasjon om disse. Relevante objektklasser (f.eks. Person, Sted og Hendelse) er allerede koblet til aktuelt videomateriale, og tilleggsinformasjon kan uttrykkes med Annotasjoner (se evt. Figur 21).



Figur 29, Hierarkisk ordning av innholds-objekter

En del objektklasser er rekursivt hierarkisk ordnet. Sted er et eksempel på dette, da et sted kan bestå av flere steder, samtidig som det er en del av et annet sted. Byen Trondheim ligger f.eks. i fylket Sør-Trøndelag, som igjen ligger i landet

Norge. Dette konkrete eksempelet på hierarkisk ordnet informasjon er vist i Figur 29 sammen med relasjonen “del av” i datamodellen, som beskriver slike sammenhenger.

En tilsvarende ordning kan en finne for fysiske Objekter, hvor et objekt kan bestå av delobjekter samtidig som det selv inngår i et større objekt. Personer kan til en viss grad plasseres i et hierarki, men dette hierarkiet ville være situasjons- og rollebetinget, og trolig lite egnet for eksplisitt modellering pga. stadige endringer. Hendelser har ingen naturlig hierarkisk struktur, men er derimot knyttet til et tidspunkt eller en tidsperiode. Hensikten med å modellere og registrere slike relasjoner eksplisitt, er muligheten dette gir for bedre støtte ved søk. Relasjonene legges inn bare en gang, og deretter er det bare nødvendig å oppdatere dem dersom hierarkiet endres³⁹.

6.3 Støtteverktøy

I NRK er det i dag, som tidligere nevnt, liten integrasjon av produksjon og arkiv, og dels av de ulike produksjonsprosessene. Den foreslåtte arkitekturen for Digitalt Fjernsynsarkiv (Figur 24) er med på å rive ned disse barrierene. Dette er også demonstrert gjennom beskrivelsene av ulike former for produksjonsinformasjon og konkrete eksempler i forrige avsnitt.

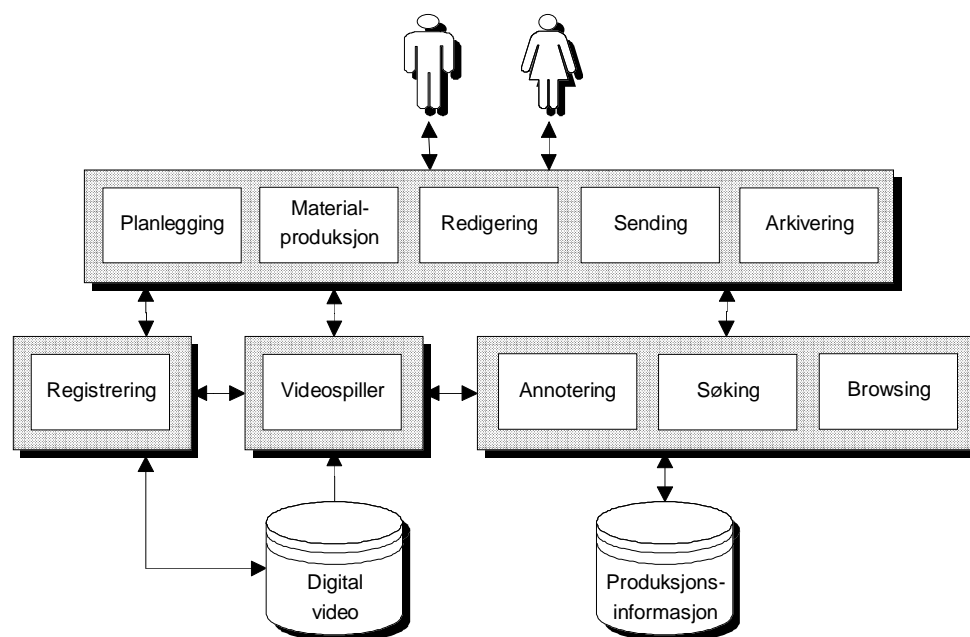
6.3.1 Funksjonelle komponenter

Et Digitalt Fjernsynsarkiv skal altså støtte de fem prosessene i verdikjeden. Støtteverktøy (programvare) for et slikt system blir såpass omfattende at det er utenfor denne rapportens rekkevidde å forklare alle detaljer ved et slikt verktøy. Likevel gir den overordnede fremstillingen i dette avsnittet, sammen med beskrivelsen av praktiske forsøk i neste kapittel, en viss oversikt over de mange mulighetene kombinasjonen av digital video og produksjonsinformasjon tilbyr.

Figur 30 viser de funksjonelle komponenter eller *moduler* som kan inngå i et støtteverktøy.

Det øverste nivået i figuren tilbyr et grensesnitt mellom brukerne og et underliggende sett av mer grunnleggende moduler, slik at systemet overfor brukerne fremstår som et enhetlig, integrert verktøy. Øverste nivå består derfor av produksjonsrettede verktøy tilpasset hver av de fem aktivitetene med samme navn. Aktørene i produksjonskjeden blir systemets brukerne. Siden alle har forskjellige oppgaver og behov, vil det være naturlig å definere bruksrettigheter som samsvarer med dette. Ulike brukere kan derfor arbeide med forskjellige deler av verktøyet, slik at f.eks. en redigerer vil oppleve et “annet” verktøy enn en reporter eller arkivar.

³⁹ Konkrete eksempler på endringer i et sted-hierarki, er f.eks. samlingen av Tyskland og delingen av tidligere Jugoslavia. På mer hjemlige trakter kan en nevne sammenslåing av norske kommuner.



Figur 30, Funktionelle komponenter i et støtteverktøy

Følgende grunnleggende moduler er tegnet inn på nederste nivå i figuren:

- *Registrering*, som tilbyr innlegging og vedlikehold av digital video. Med digitale (harddiskbaserte) kameraer, kan video legges direkte inn i videodatabasen, mens med analoge kameraer må det i tillegg være støtte for digitalisering av video.
- *Digital videospiller* er en tjener (eng. *server*) som bl.a. spiller av digital video fra databasen og viser den for brukerne i et eget vindu. Spilleren kan tilby enkle standardfunksjoner, f.eks. pause, avspilling, hurtigspoling og bilde-for-bilde fremover og bakover. IDTs digitale videospiller er et konkret eksempel som ble vist i Figur 19. Dessuten kommuniserer spilleren med registreringsmodulen og modulene for annotering, søking og browsing, som alle er klienter eng. *clients*). I praksis innebærer dette at klientene (modulene) kan styre videospilleren og få informasjon om hva som spilles av i øyeblikket (rammenummer eller tidskode).
- *Annotering* er en modul som knytter informasjon til bestemte rammesekvenser. Informasjonen kan eksempelvis være de tidligere omtalte strukturelle komponentene (sekvens, scene og innstilling) eller innholdsobjektene (person, sted og hendelse).
- *Søking*. Denne modulen finner rammesekvenser som er knyttet til bestemte innholdsobjekt (Figur 23) eller annen produksjonsrelatert tekstinformasjon, f.eks. om produksjonsobjekt, tekstdokument og ansatte (Figur 25). En kan sammenlikne funksjonaliteten med SIFT-søk som benyttes i dag, men en får i tillegg mer sofistikert søkefunksjonalitet som ikke støttes av SIFT.

- *Browsing*⁴⁰. Denne modulen finner rammesekvenser på bakgrunn av strukturelle komponenter, altså hvordan en rammesekvens er relatert til andre rammesekvenser. I klartekst betyr dette at en f.eks. kan finne bestemte sekvenser, scener eller innstillinger i et nyhetsinnslag.

De produksjonsrettede verktøyene er allerede beskrevet i forbindelse med aktivitetene i avsnitt 6.2. *Registrering* av digital video er omtalt i avsnitt 6.2.2 og *digital videospiller* er kort beskrevet i avsnitt 5.1.3. Annotering, søking og browsing er derimot nye konsepter, som trenger en nærmere forklaring.

6.3.2 Annotering

Annoteringsmodulen gjør det som nevnt mulig å legge inn informasjon om rammesekvenser. Strukturell organisering av videomaterialet (såkalt segmentering) kan utføres i forbindelse med opptak og redigering, slik det ble forklart i avsnitt 6.2. Slike rammesekvenser kan også tildeles et beskrivende navn som letter gjenfinningen.

Den andre formen for annotering er altså tilknytning av innholds-objekter. Første gang et objekt benyttes, kan det opprettes med denne modulen. Brukeren må da angi alle relevante attributter, f.eks. stedsnavn og land dersom objektet er et sted. Etter hvert som databasen fylles med data, vil det være mulig å f.eks. velge objektet fra en liste eller referere til det på annen måte.

Tilleggsinformasjon kan registreres om koblingen mellom innholds-objekt og rammesekvens, i form av *annotasjoner* (Figur 23). Slik informasjon vil neppe bli registrert i en travel produksjonsprosess, slik at dette må legges på i ettertid - f.eks. av Fjernsynsarkivets arkivarer.

Annoteringsmodulen er altså en modul i støtteverktøyet som vil bli benyttet i flere prosesser.

6.3.3 Søking

Kriteriene for et søk kan være bestemte attributter i *produksjonsobjektet*, f.eks. produksjonsteam eller sendedato. Dette tilsvarer dagens søk i de strukturerte feltene i SIFT. Mer vanlig vil det imidlertid være å søke i *innholds-objektene* og *annotasjonene*, som tilsvarer søk i fritekstfeltet i SIFT.

Siden det lagret informasjon om typen (objektclassen) til innholds-objektet, er det mulig å utnytte dette ved søking. En kan f.eks. søke etter alle rammesekvenser fra Oslo (“Sted = Oslo”) filmet om vinteren (“StedAnnotasjon = Vinter”) som *ikke* er annotert med person-objekter (“Ikke Person”)⁴¹.

⁴⁰ Det engelske ordet *browse* betyr å “skumlese” eller “bla gjennom”. Fremfor å innføre enda en meningsløs norsk oversettelse, benyttes det engelske ordet i denne rapporten.

⁴¹ Den viste syntaksen er å betrakte som et eksempel, da søkmodulen bare *utfører* søket. Hvordan brukeren *angir* søket (f.eks. vha. tekst, valglister, eller grafiske symboler), bestemmes av brukergrensesnittet i de overliggende støtteverktøyene.

Den hierarkiske relasjonen mellom steder (vist i Figur 29) kan brukes for å avgrense eller utvide søk. Anta som et eksempel at søk etter “Sør-Trøndelag” resulterer i en liste med flere tusen rammesekvenser som resultat. Det kan da være ønskelig å *avgrense* søket ved å presisere hvilket sted i Sør-Trøndelag som er aktuelt, f.eks. “Stjørdal”. Alternativt kan det tenkes at en i første omgang har søkt etter “Stjørdal”, men bare får noen få treff. I et slikt tilfelle, kan det være aktuelt å *utvide* søket ved å gå opp et nivå i hierarkiet. Siden steder er hierarkisk ordnet, er veien oppover i hierarkiet entydig, slik at gjentatte utvidelser av søk er mulig uten å angi noe kriterium.

Dersom en finner en rammesekvens som oppfyller søkekriteriene, er det mulig å “be” den digitale videospilleren om å spille av denne. Videospilleren må da få vite hvilken *videostrøm* rammesekvensen inngår i, samt *hvor* i videostrømmen denne ligger (f.eks. tidskode eller nummer for første og siste ramme).

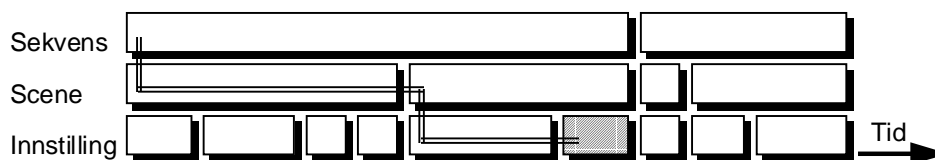
En får altså ikke bare et tekstlig søkeresultat som i dag, men kan se de opptakene som oppfyller søkekriteriet. På denne måten kan en straks avgjøre om videoopptakene har interesse. Dersom den digitale videoen bare er av påsynskvalitet, er det fortsatt nødvendig å bestille kringkastingskvalitet av de ønskede opptakene. Også i et slikt tilfelle er det nyttig å få se videomaterialet før en bestiller det. En slipper da å bestille flere bånd enn absolutt nødvendig, og det gir en stor tidsgevinst at en slipper å bestille og lete gjennom bånd som likevel ikke kan benyttes. Med digital video av full kringkastingskvalitet, kan arkivmaterialet i databasen benyttes uten videre i en ny produksjon.

6.3.4 Browsing

I en nyhetssending gir programlederen hint om sendingens struktur, f.eks. med informasjon av typen “Nå til skipseksplasjonen i Sør-Korea”. Dagsrevyen starter dessuten med en kort innledende presentasjon av kveldens nyheter og slutter med en oppsummering av de viktigste sakene.

Dersom den samme sendingen er lagret digitalt, kan de strukturelle komponentene *Sekvens*, *Scene* og *Innstilling* eksplisitt uttrykke slik strukturell informasjon og benyttes for å finne frem til ønsket del av programmet.

Browsing kan betraktes som en metode for “navigering” i lagret videomateriale (rammesekvenser) gjennom å benytte de strukturelle komponentene. Dette kan best illustreres med et eksempel.



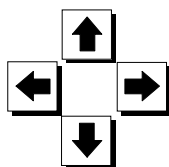
Figur 31, Browsing i rammesekvenser

Figur 31 viser strukturen i et tenkt program. Normalt vil hver av de strukturelle komponentene ha fått en beskrivende tittel (ikke vist i figuren), f.eks. “Skipseksplasjon i Sør-Korea” eller “Innledning”. Ved å vise disse titlene i kronologisk ordnede lister, er

det eksempelvis mulig å velge ønsket rammesekvens, f.eks. den skraverte innstillingen i figuren. Denne innstillingen kan så spilles av vha. videospilleren på samme måte som søkeresultatene fra søkemodulen.

En alternativ måte for valg av nevnte innstilling, kan være å vise strukturen grafisk (som i figuren), sammen med tittelen på hver rammesekvensene. Ved å peke på den skraverte innstillingen, f.eks. med en musepeker, kan videospilleren vise tilhørende videomateriale (rammesekvens).

Strekene i figuren antyder en tredje måte for å velge den skraverte innstillingen, som er velegnet ved *visning* av videomateriale. For hvert nivå i strukturen er det mulig å gå til neste eller forrige rammesekvens, på samme måte som en skifter melodi på en CD-plate. I tillegg er det mulig å navigere opp eller ned mellom nivåene. En kan benytte retningstastene på et tastatur eller lage tilsvarende trykknapper i skjermbildet som kan velges med mus, illustrert i Figur 32.



Figur 32, Retning som navigeringsmekanisme ved browsing

Ved et trykk på en knapp, vil avspillingen umiddelbart og uten ventetid hoppe til nærmeste rammesekvens i angitt retning. For å komme til den skraverte innstillingen, ville en f.eks. med startpunkt i første sekvens velge følgende hopp: ↓, →, ↓, →. For lengre programmer med mange rammesekvenser (f.eks. Dagsrevyen) er dette en rask måte å navigere på. Hoppene vil som regel være størst på sekvensnivå og minst på innstillingsnivå.

7. Praktiske forsøk

For å illustrere de ulike funksjonelle komponentene som inngår i et Digitalt Fjernsynsarkiv og høste ytterligere erfaringer for utviklingen av et fullverdig støtteverktøy, er implementering av prototyper til stor hjelp. Slike prototyper kan benyttes for demonstrasjonsformål, samtidig som de gjør det mulig å oppleve ulike aspekter av et system i praksis.

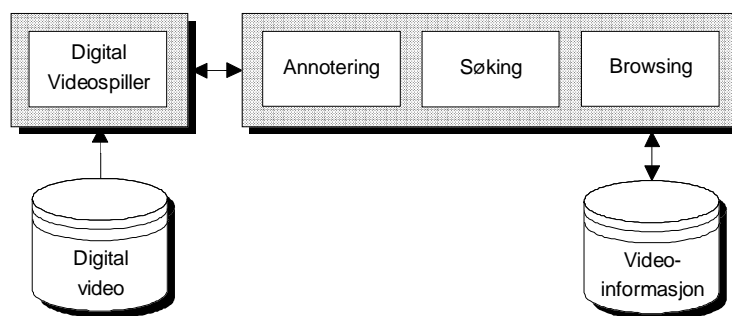
Databasegruppen ved institutt for datateknikk (IDT) har de siste to årene bl.a. arbeidet med utvikling av digitale videoarkiv og modellering av film- og videoinformasjon. Denne diplomoppgaven inngår sammen med tidligere og pågående arbeid som en naturlig del av gruppens aktiviteter på dette området.

7.1 Støtteverktøy ved IDT - en oversikt

For eksperimenter ved IDT blir det benyttet en Sun SPARCstation LX med Parallax XVIDEO-kort. Digital video lagres som to filer: én for bilder og én for lyd. En tredje fil benyttes for å indeksere de to andre [46]. Det ble i 1993 digitalisert drøye to uker med NRKs programpost "Ettermiddagsnytt", som fortsatt brukes ved forsøkene.

Av de grunnleggende modulene i Figur 30, er det tidligere utviklet en digital videospiller (vist i Figur 19) og et annoteringsverktøy. Det er dessuten utviklet et kombinert søke- og browseverktøy som illustrerer enkel søking samt browsing på ett nivå (valg av innslag).

Arkitekturen for IDTs støtteverktøy er derfor som vist i følgende figur:



Figur 33, Arkitektur for støtteverktøy ved IDT

De fire komponentene (modulene) er implementert som frittstående applikasjoner som kjører under X-Windows. Annoteringsverktøyet lagrer videoinformasjon som to filer (for hver nyhetssending): én for innholdsinformasjon og én for strukturelle komponenter. Disse to filene blir benyttet av hhv. søke- og browseverktøyet. Kommunikasjonen mellom videospilleren og de tre verktøyene foregår som *meldingsutveksling*, hvor en melding kan bestå av atomiske kommandoer og/eller parametre.

Tidligere implementerte verktøy for annotering, søking og browsing hadde begrenset funksjonalitet, og støttet bare delvis det rammeverket som datamodellen for videomateriale i Figur 21 definerer. Det er derfor utviklet nye versjoner av samtlige

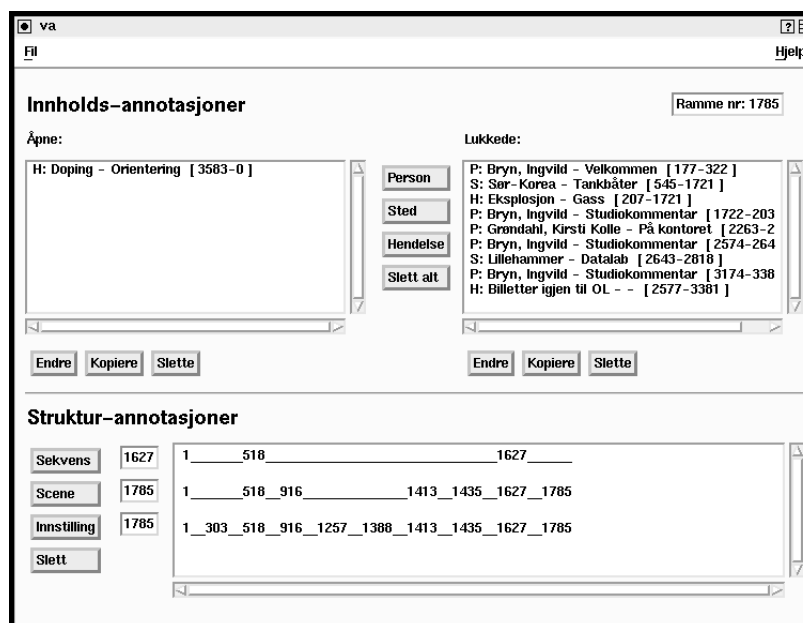
verktøy, bl.a. i forbindelse med denne diplomoppgaven. For utvikling av verktøyene er det benyttet *X-* og *Motif-toolkits* og programmeringsspråket *C*.



7.2 Annoteringsverktøy

På oppdrag fra IDTs gruppe for databaseteknikk, har undertegnede implementert et annoteringsverktøy parallelt med diplomarbeidet. Verktøyet er nærmere beskrevet i Vedlegg G, og en kort oppsummering av funksjonaliteten følger.

Hovedvinduet i annoteringsverktøyet består av to deler, som vist i Figur 34.



Figur 34, Brukergrensesnitt for annoteringsverktøyet

Hver gang videospilleren settes i pause-modus eller programmet kjøres en ramme fram eller tilbake, sender videospilleren en melding om rammenummeret til annoteringsverktøyet. Dette nummeret kan sammenliknes med en tidskode, og vises i øverste høyre hjørne av vinduet.

Øvre halvdel av vinduet, "Innholds-annotasjoner", benyttes for å knytte objektene *person*, *sted* eller *hendelse* til vilkårlige rammesekvenser. En rammesekvens er avgrenset av et start- og sluttrammenummer. Slike rammesekvenser (og dermed de tilhørende annotasjoner) kan kalles *lukkede* fordi de definerer et lukket intervall, i motsetning til *åpne* annotasjoner, som foreløpig mangler start- eller sluttramme-

nummer. Åpne og lukkede innholds-annotasjoner vises i hver sin liste med følgende informasjon:

- Objektets type, enten Person, Sted eller Hendelse.
- Objektets navn, f.eks. Lillehammer
- Annotasjon av objektet som gjelder denne rammesekvensen, f.eks. Datalab.
- Start- og sluttrammenummer på formen [startramme - sluttramme].

En innholds-annotasjon kan opprettes ved å trykke på aktuell knapp (Person, Sted eller Hendelse) plassert mellom de to listene. Et eget dialogvindu (vist i Vedlegg G) benyttes for å registrere nødvendig informasjon. Innholds-annotasjoner kan dessuten endres, kopieres eller slettes.

Det er i verktøyet registrert ni lukkede og én åpen innholds-annotasjon. Sistnevnte er hendelsen `Doping`, som (via hendelses-annotasjonen `Orientering`) er knyttet til en rammesekvens. Som vist starter denne rammesekvensen med ramme 3583, og den har foreløpig ikke fått tildelt noen sluttramme (angitt med 0).

Nedre halvdel av vinduet, “Struktur-annotasjoner”, benyttes for å definere rammesekvenser som angir programmets struktur (inndelt i *sekvenser*, *scener* og *innstillinger*). Tre knapper benyttes for å markere overgangen mellom rammesekvenser. Aktuelt rammenummer blir registrert, og programstrukturen vises i et eget felt etter hvert som den blir definert. Det forutsettes at registreringen utføres sekvensielt, altså fra begynnelsen av programmet og stigende rammenummer.

Av hovedvinduet i Figur 34 kan en se at videospilleren i øyeblikket viser videoramme nummer 1785. En kan dessuten se at det nettopp er markert en scene-overgang, siden siste scene (og innstilling) også slutter på ramme 1785. Siste sekvens, scene og innstilling startet på ramme 1627, og en kan se at det foreløpig ikke er definert noe sluttrammenummer for sekvensen.

Struktur-annotasjoner og lukkede innholds-annotasjoner lagres i hver sin fil som senere kan brukes av hhv. søke- og browseverktøyet. For utførlig forklaring av benyttede filformat, vises det til Vedlegg G.

7.3 Søkeverktøy

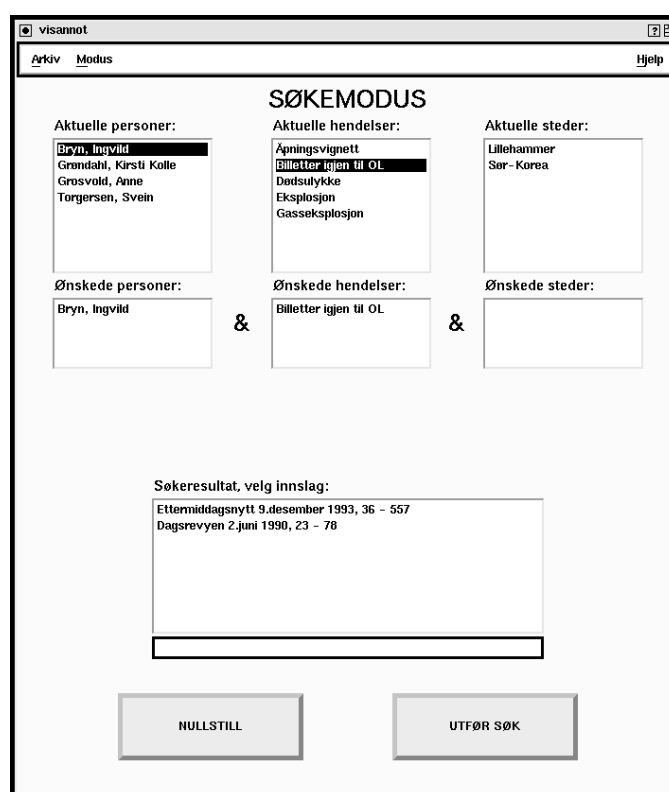
Arnt Sæter har i sitt diplomarbeid utviklet et søkeverktøy som benytter innholds-annotasjoner fra annoteringsverktøyet [39]. Programmet kan operere i to modi med tilnærmet like grensesnitt.

- Søkemodus (søk i innholds-annotasjoner)
- Annotasjonsvisnings-modus (visning av innholds-annotasjoner)

I *søkemodus* benyttes hovedvinduet som er vist i Figur 35. De tre listene øverst i vinduet viser de tre typene objekter (person, sted og hendelse) som er registrert for alle

programmer⁴². Ved å velge ut de ønskede objektene av hver type, blir disse lagt inn i korresponderende liste. I det viste eksempelet er personen “Bryn, Ingvild” og hendelsen “Billetter igjen til OL” valgt⁴³. Ved å trykke på knappen merket “UTFØR SØK”, vil søkeresultatet bli vist i lista nederst i vinduet. I dette eksempelet ble resultatet to rammesekvenser. Dersom en velger et av søkeresultatene fra resultatlista, vil videospilleren vise dette.

Det er også mulig å velge flere objekter av samme type. Disse kombineres med den logiske operatoren “eller” (f.eks. “Bryn, Ingvild” eller “Grøndahl, Kirsti Kolle”), mens delresultatet (ønskede personer) kombineres med de andre delresultatene med operatoren “og”. Et mulig søk vil f.eks. være (Bryn, Ingvild *eller* Grøndahl, Kirsti Kolle) *og* (Billetter igjen til OL *eller* Eksplosjon)⁴⁴.



Figur 35, Brukergrensesnitt for søkeverktøyet (i søkemodus)

Annotasjonsvisnings-modus er verktøyets andre modus med et nesten identisk hovedvindu, bortsett fra at vinduet i tillegg inneholder et felt som viser rammesekvensenes tekstlige annotasjoner. Når videospilleren settes i pausestilling eller

⁴² Som en ser, er det vist svært få objekter av hver type. Dette skyldes at verktøyet i det viste tilfellet opererte på *testdata*, som også forklarer hvorfor beskrivelsene ikke er særlig realistiske. Etter hvert som flere nyhetsinnslag og -sendinger er annotert, er disse listene blitt korrigert og betydelig større.

⁴³ Strengt tatt er ikke dette en hendelse, men som nevnt ble det

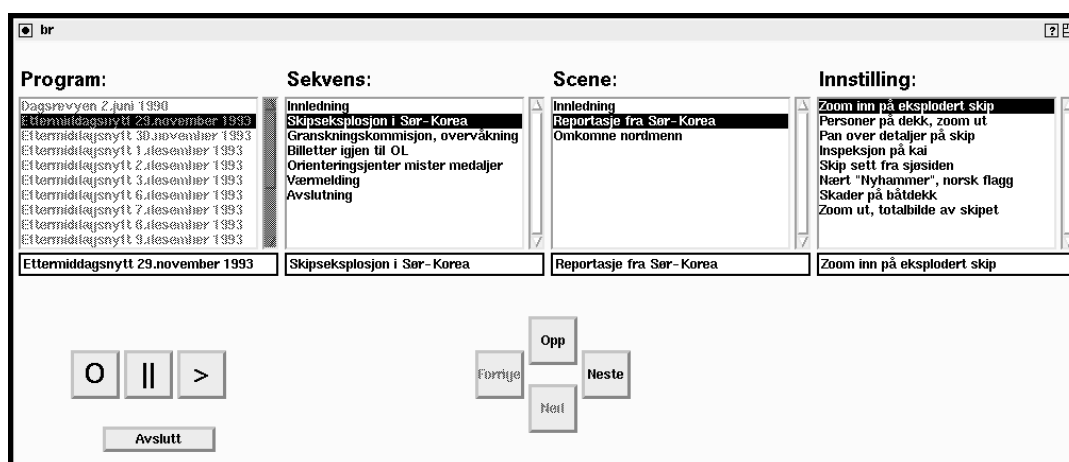
⁴⁴ En kan imidlertid betvile nytteverdien av akkurat dette konkrete søket.

programmet kjøres en ramme fram eller tilbake, sender videospilleren en melding om rammenummeret til søkeverktøyet. Verktøyet finner da alle rammesekvenser hvor aktuell ramme inngår, og lister opp alle objekter som er knyttet til disse rammesekvensene - sammen med evt. fritekst-annotasjoner. Det er mulig også her mulig å søke etter ønskede objekter, og listene øverst i vinduet viser da bare objekter som er gyldige for det aktuelle rammenummeret.

7.4 Browseverktøy

Mesteparten av implementeringsarbeidet i denne diplomoppgaven har vært konsentrert om et browseverktøy for å tilby strukturert tilgang til og visning av digital video. Dette verktøyet blir derfor noe mer detaljert beskrevet.

Det teoretiske grunnlaget for browsing av videomaterialet ble beskrevet i avsnitt 6.3.4, blant annet ble valglister og retningsknapper omtalt. Begge disse metodene for browsing er implementert i browseverktøyet. Verktøyet har fått et hovedvindu som er vist i Figur 36.



Figur 36, Brukergrensesnitt for browseverktøyet

Det mest iøynefallende med vinduet er de fire listene som hhv. viser navn på programmer, sekvenser, scener og innstillinger. Ved oppstart av browseverktøyet viser den første lista hvilke program som er tilgjengelig, mens de tre andre listene er tomme. Når et program blir valgt, vises det hvilke sekvenser dette inneholder. Når en sekvens blir valgt, vises det hvilke scener denne inneholder, osv.

Videre inneholder vinduet et lite kontrollpanel som gjør det mulig å starte, pause og stoppe avspillingen av et program direkte fra browseverktøyet. En kan enkelt bygge ut dette kontrollpanelet, men flere trykknapper finnes allerede i videospiller-applikasjonen (Figur 19), og dette er heller ikke hovedhensikten med et browseverktøy.

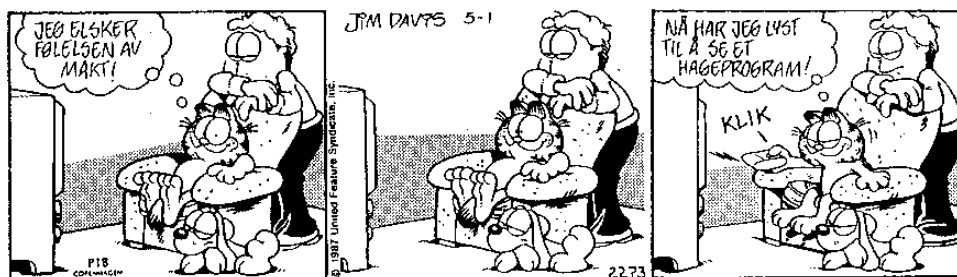
Nederst i vinduet er det lagt inn fire retningsknapper (*opp*, *ned*, *forrige* og *neste*), som hopper til neste rammesekvens i angitt retning. Fra en scene er det mulig å gå opp (til en sekvens) eller ned (til en innstilling), mens fra en sekvens eller innstilling er det bare mulig å gå hhv. ned eller opp. På hvert av de tre nivåene kan en dessuten gå til neste eller forrige rammesekvens, helt til en kommer til starten eller slutten på nivået over.

Dersom en med IDTs videospiller ønsker å skifte til et annet program, er det nødvendig å stoppe videospilleren (dvs. avslutte videospiller-applikasjonen) før den igjen kan startes med et annet program. Lista for valg av program er derfor sperret inntil videospilleren stopper, noe som blir angitt med grå skrift.

I Figur 36 har brukeren valgt programmet “Ettermiddagsnytt 29.november 1993”. Videre har han valgt sekvensen “Skipseksplisjon i Sør-Korea”, scenen “Reportasje fra Sør-Korea” og innstillingen “Zoom inn på eksplodert skip”. De avmerkede listeinnslagene og teksten under listene viser til enhver tid hvilken del av videomaterialet som blir avspilt, i øyeblikket altså en innstilling hvor det zoomes inn på et eksplodert skip.

Brukeren kan ha valgt denne innstillingen gjennom å “klikke seg fram” gjennom listene eller ved gjentatte trykk på retningsknappene. Browsingen starter alltid i første sekvens av et program, slik at brukeren i så fall har trykket på følgende knapper for å komme til aktuell innstilling: Neste (sekvens) - Ned (til scene) - Neste (scene) - Ned (til innstilling). Brukeren kan skifte mellom de to navigeringsmetodene etter eget ønske, og listene blir kontinuerlig oppdatert for å gi brukeren visuell tilbakemelding om hvor i materialet han “befinner seg”. I tillegg blir ugyldige trykknapper blokkert og vist med grå skrift. For innstillingen som vises i øyeblikket er det f.eks. umulig å gå lengre ned (fordi innstilling er nederste nivå) eller til forrige innstilling (fordi dette er den første).

Mens tradisjonell visning av analog video ikke eksplisitt gir strukturell informasjon om videomaterialet, gir browseverktøyet brukeren full oversikt og en følelse av kontroll.



Med tanke på evt. senere re-implementasjoner følger en kort beskrivelse som letter forståelsen av kildekoden og beskriver hovedprinsippene bak utviklingen av browseverktøyet. For spesielt interesserte er det også inkludert en full utlisting av kildekoden for browseverktøyet i Vedlegg H.

7.4.1 Implementering av browseverktøy

Browseverktøyet er, i likhet med de andre to støtteverktøyene, utviklet for å kjøre under X-Windows. Hovedvinduet er konstruert med såkalte *widgets* (grafiske elementer) som ved aktivisering kaller opp en brukerdefinert funksjon [48].

Browseverktøyet leser en *katalogfil* som inneholder følgende informasjon om hvert enkelt program (på et standardformat vist i Vedlegg G):

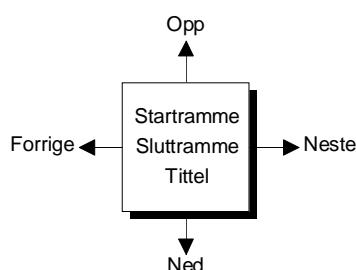
- Programnavn
- Videofilnavn
- Annotasjonsfilnavn

For å holde disse tre navnene er det definert en intern *datatype*, kalt *programPost*. Alle programposter blir koblet sammen i en *lenket liste* som bl.a. sørger for at alle programnavn kan vises i listeboksen på skjermen. Videofilnavnet er nødvendig for at videospilleren kan starte vise riktig program, og annotasjonsfilnavnet sørger for at riktig struktur-annotasjonsfil blir lest inn.

Hver struktur-annotasjonsfil inneholder informasjon om de ulike rammesekvensene:

- Type (sekvens, scene eller innstilling)
- Startramme
- Sluttramme
- Tittel

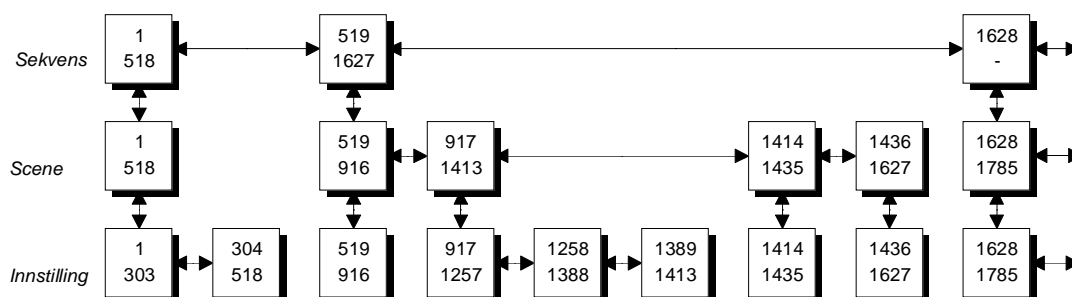
Også for denne informasjonen er det opprettet en egen datatype, som er kalt *strukturPost* og vist i Figur 37.



Figur 37, Strukturpost i browseverktøyet

Strukturposten er så generell at den kan benyttes for både sekvens, scene og innstilling. Det er ikke inkludert noe typefelt i posten, siden browseverktøyet selv holder orden på dette. De fire pilene representerer pekere, som knytter hver strukturpost til andre strukturposter.

Strukturen for et program kan dermed bygges opp som en samling av strukturposter i tre nivåer. Skjermbildet fra annoteringsverktøyet i Figur 34 viste struktur-annotasjonene (rammesekvensene) for et konkret program. Figur 38 viser hvordan de samme struktur-annotasjonene er representert i browseverktøyet.



Figur 38, Rammesekvenser representert som strukturposter

Øverste nivå angir sekvenser, midterste nivå scener og nederste nivå innstillinger. Tittelfeltet er utelatt i figuren, slik at strukturpostene bare viser start- og sluttramme-

nummer. Dessuten er ikke alle pekerne oppover i strukturen tegnet inn, men i oppoverretning er det valgt å lenke alle strukturpostene til første strukturpost i nivået over. F.eks. er både innstilling [917-1257], [1258-1388] og [1389-1413] knyttet oppover til scenen [917-1413]. For den tredje sekvensen [1628-] har en foreløpig ikke definert noe sluttrammenummer (som vist også i Figur 34).

Verktøyet benytter bl.a. tre pekere for hhv. *første* sekvens, scene og innstilling, altså de tre strukturpostene som er vist lengst til venstre i figuren og en “browse-peker” som peker på den strukturposten som er valgt. Denne generelle datastrukturen legger grunnlaget for browsing både med “retningsknappene” og listene.

For at videospilleren skal kunne starte avspilling av riktig rammesekvens i en videostrøm, sender browseverktøyet rammesekvensens startrammenummer til videospilleren. Siden IDTs videospiller på forhånd ikke var forberedt for browsing, var det også nødvendig å utvide videospillerens funksjonalitet til å kunne motta rammenummer.

7.5 Praktisk bruk av verktøyene

Med tanke på praktiske forsøk og demonstrasjoner av verktøyene, har en del av de tidligere digitaliserte nyhetsendingene blitt annotert i forbindelse med diplomarbeidet. Annoteringsverktøyet er benyttet for å registrere og lagre innholds- og struktur-annotasjoner.

For *logging* (registrering) av rammenummer, kjører en videospilleren til det bildet som innleder eller avslutter aktuell rammesekvens. Videospilleren sender som nevnt rammenumre til annoteringsverktøyet, hvor selve annoteringen foregår. Brukeren kan dermed konsentrere seg om videomaterialet mens verktøyet holder orden på rammenummer.

Når annoteringsverktøyet benyttes isolert på denne måten, må all informasjon om det ferdiglagede videomaterialet legges inn på en gang, og en kan (med rette) bli overveldet av alt registreringsarbeidet som må utføres. Betrakter en derimot annoteringsverktøyet som en *funksjonell komponent* slik det ble foreslått i Figur 30, vil det istedet inngå som en av delene i et større verktøy. Registreringsoppgaven blir da fordelt mellom de ulike produksjonsaktivitetene (aktiviteten *arkivering* regnes også som en del av produksjonskjeden), og kan utføres med de tilhørende produksjonsrettede verktøyene. Hver aktivitet vil da bidra med ulike deler av de nødvendige annotasjonene, som forklart i avsnitt 6.2. Dessuten kan en under annoteringen benytte søke- og browsefunksjonaliteten som foreløpig er implementert gjennom separate søke- og browseverktøy.

Etter hvert som digital video får større utbredelse, er det sannsynlig at stadig flere brukere vil oppdage de fordelene en oppnår gjennom beskrivelse og strukturering av videomaterialet.

De utviklede verktøyene er demonstrert for Norsk Folkemuseum, som ser et stort potensiale for bruk av digital video i forbindelse med kulturhistorisk materiale. Det ble uttrykt at verktøyene virket svært nyttige og anvendelige, og en kunne nærmest tenke seg å ta disse i bruk umiddelbart. Som en konsekvens ønsker Norsk Folkemuseum å

gjennomføre prosjekter i samarbeid med IDT for å undersøke disse mulighetene nærmere.

NRK har i dette diplomarbeidet tilført svært nyttig og relevant informasjon om fjernsynsproduksjon og fjernsynsarkiv. De tre utviklede verktøyene er allerede presentert og demonstrert ved NRK Marienlyst for et bredt panel NRK-medarbeidere. Siden demonstrasjonen ble arrangert dagen før diplominnlevering, er det dessverre umulig å inkludere et fylldig referat. Imidlertid gjorde verktøyene det mulig å forklare abstrakte konsepter gjennom praktiske anvendelser i en reell situasjon.

Fra NRK-hold ble det konkludert med at demonstrasjonen hadde gitt god forståelse av en del muligheter som et Digitalt Fjernsynsarkiv kan tilby, og man så klart nytteverdien av å ta slike verktøy i bruk - i første omgang for videomateriale av påsynskvalitet. Demonstrasjonen blir dessuten ett av flere utgangspunkt for videre intern diskusjon på flere fronter i NRK som vil avklare ønsker, fremtidige økonomiske prioriteringer og langtidsstrategier.

8. Konklusjon

Dette diplomarbeidet har sett nærmere på digitale fjernsynsarkiv. Utgangspunktet har vært NRKs Fjernsynsarkiv slik det fremstår i dag. Dersom en i dag betrakter Fjernsynsarkivet fra utsiden, er det et sted hvor en henvender seg for å få godt arkivmateriale til nye produksjoner: “Etter at programmet er spilt inn på bånd, kan en søke etter det i SIFT-databasen og be om å få en kopi”. Betrakter en derimot arkivet fra innsiden, vil en se at uten omfattende registreringsarbeid, omhyggelig utført av Fjernsynsarkivets arkivarer, ville en slik tjeneste være umulig.

Det er som regel nødvendig å bestille mye materiale for å være på den sikre siden, så se gjennom dette, og til slutt gjerne oppdage at bare en brøkdel likevel kan brukes. *Søking, bestilling, kopiering* og *gjennomsyn* tar mye tid, men er nødvendige rutiner i et tradisjonelt fjernsynsarkiv. Bestilling og kopiering er nødvendig fordi videomaterialet er lagret på analoge videobånd, mens gjennomsyn er nødvendig fordi programbeskrivelsene aldri kan gi et helt sikkert inntrykk av innholdet.

Et digitalt fjernsynsarkiv angriper alle disse problemområdene. Med digital video av full sendekvalitet, vil bestilling og kopiering istedet utføres i et integrert støtteverktøy som benytter databaser med digital video og innholdsinformasjon. *Søking* og *bestilling* utføres i en database, og istedenfor *kopiering* blir det opprettet en logisk referanse til fysisk lagret videomateriale. *Gjennomsyn* utføres samtidig med søkingen, og videomaterialet kan umiddelbart vises med en digital videospiller i det samme verktøyet.

En kan i utgangspunktet innføre et digitalt fjernsynsarkiv av denne typen uten at en nødvendigvis endrer på programproduksjonen. Skal en imidlertid få fullt utbytte av evt. investeringer i et slikt system, kreves det en tettere kobling mellom *videomaterialet* og *beskrivelsene* av dette. *Videomaterialet* i dag er et resultat av produksjonsaktivitetene og *beskrivelsene* er et resultat av arkiveringsaktiviteten. Det er følgelig nødvendig med en tilsvarende tettere kobling mellom disse aktivitetene. Fjernsynsarkivet trenger derfor mye informasjon fra produksjonsfasen som de ikke får i dag. Dette betyr at informasjon må fanges opp underveis i produksjonen, og det er urealistisk at Fjernsynsarkivet skal løse denne oppgaven alene. Utfordringen er å få tak i denne informasjonen uten at producer, reporter og andre aktører opplever det som en ekstra belastning i en allerede stresset hverdag.

Med det integrerte støtteverktøyet som er foreslått i rapporten, er grunnlaget lagt for en tettere integrasjon av alle aktiviteter. Verktøyet kan nærmest ubemerket samle inn informasjon samtidig som alle aktører utfører samme arbeid som i dag, dog med mer datastøtte. Ved å legge inn funksjonalitet i verktøyet som støtter brukerne i deres daglige virke, kan en faktisk heller oppleve at alle kan utføre oppgavene *bedre* enn i dag.

NRK er primært en “produksjonsbedrift for TV-programmer”, og ved at et slikt støtteverktøy støtter hele produksjonskjeden, kan dette på sikt føre til enda bedre programmer. NRK følger også aktivt med i den pågående utviklingen og innføringen av interaktive medier. For produksjon av gode interaktive TV-programmer, er det nødvendig å endre de nåværende aktivitetene. Det som i det hele gjør et program *interaktivt*, er ikke selve programinnholdet, men den *tilleggsinformasjonen* en knytter

til programmet. Siden støtteverktøyet kan trekke ut mye tilleggsinformasjon direkte fra aktivitetene, er det også et viktig skritt på veien mot interaktiv TV.

8.1 Videre arbeid

Beskrivelsen av de ulike aktivitetene i fjernsynsproduksjonen har gjort det mulig å gi støtteverktøyet et grensesnitt mot hver aktivitet. Som en naturlig oppfølging av dette diplomarbeidet, kan en gå videre i minst to retninger:

- Modellere og prototype funksjonaliteten i de produksjonsrettede delene av støtteverktøyet.
- Videreutvikle basisfunksjonaliteten (annotering, søking og browsing) i støtteverktøyet.

Av produksjonsrettede delverktøy har Bekkadal i sin diplom sett på redigeringsverktøy [43]. En tilsvarende innsats er mulig for hver av de andre prosessene, hvor definering av funksjonelle egenskaper er et naturlig utgangspunkt. Teknologer kan alltid foreslå løsninger, men det er opp til brukerne om en virkelig *ønsker* de mulighetene som teknologien tilbyr. Tett samarbeid med fremtidige brukere av et støtteverktøy er helt sentralt, og NRK vil på mange områder være en svært egnet og kompetent samarbeidspartner, dersom NRK har interesse av dette.

En kan også forbedre de tre basisverktøyene på en rekke områder. *Annoteringsverktøyet* trenger f.eks. bedre støtte for struktur-annotasjoner, *søkeverktøyet* trenger kraftigere søkefunksjonalitet og *browseverktøyet* kan utstyres med et mer intuitivt, grafisk orientert grensesnitt.

Det vil også være fornuftig med en økt bevisstgjøring av verktøyene som generelle utvidelser av *databasefunksjonalitet*. Gjennom å utvikle verktøyene som en standard pakke av generelle basisfunksjoner, åpnes det for nye anvendelsesområder og effektiv gjenbruk av funksjonalitet. Dersom en vinner erfaringer fra flere bruksområder, vil en lettere se hvilken funksjonalitet som er ønskelig, og som i neste omgang også kan inkluderes i et Digitalt Fjernsynsarkiv.

Vedlegg A: SIFT

SIFT Plus (for korthets skyld ofte bare kalt SIFT) er basisapplikasjonen i et informasjonssøkesystem som kom på markedet i 1988. Systemet er utviklet og blir markedsført i Norge av Statens Datasentral (SDS). SIFT er et system som kan “ta imot, lagre, gjenfinne og presentere informasjon av vidt forskjellig struktur og innhold, slik at den blir tilgjengelig samtidig og raskt”, ifølge SDS [18]. SIFT leveres for de fleste plattformer, inkludert UNIX. SDS fremhever særlig den korte søketiden som et viktig fortrinn, i tillegg til få restriksjoner på informasjonens struktur.

SIFT har i dag et titalls større brukere, bl.a. Nasjonalbibliotekavd. i Rana (NBR) og NRK. At NRKs Fjernsynsarkiv i dag benytter SIFT, gjør systemet spesielt interessant i denne sammenheng.

SIFT står for *Søking I FriTekst*, og fritekst er en viktig del av systemet. Det kan legges inn tekstdokumenter av vilkårlig lengde, og det er mulig å søke etter alle ord og ordkombinasjoner i teksten. I tillegg har også SIFT trekk fra relasjonsdatabaser, hvor informasjonen legges inn i tabellstrukturer. I SIFT er egenskaper fra fritekstsøkesystem og relasjonsdatabasesystemer kombinert, og SDS kaller dette *informasjons-søkesystemer* [19]. I praksis betyr dette at vanlige dokumenter kan legges rett inn, og i tillegg kan det legges inn informasjon om dokumentenes struktur. Dokumentstrukturen samsvarer med faste felt, som bl.a. kan benyttes ved søking, sortering, visning og utskrift.

Dokumenter lagres normalt internt i SIFTs database, men kan også lagres i eksterne filer. Filene blir uansett indeksert i SIFT, slik at søketiden ikke blir påvirket. Det blir påstått at SIFT kan håndtere 15-20 millioner dokumenter uten at det går ut over søke- og oppdateringstiden [19].

SDS lager også applikasjoner i SIFT, og leverer pr. i dag en rekke standard database-løsninger. SIFT/Bibl er f.eks. et ferdig bibliotekssystem og SIFT/Noark er et komplett system for saksadministrasjon og -oppfølging tilpasset offentlig forvaltning og Riksarkivets krav. En egen applikasjon er også utviklet for NRKs Fjernsynsarkiv.

I forhold til “rene” relasjonsdatabasesystemer, innrømmer SDS over telefon at SIFT har en del mangler. SIFT er i utgangspunktet ikke annet enn et fil- og indekseringssystem, og mangler derfor grunnleggende konsepter som f.eks. tabeller, nøkler og fremmednøkler. Gyldigheten av data som legges inn blir heller ikke kontrollert, f.eks. tillatte verdier som gyldighetsintervall og datoformat, eller unike verdier (duplikat-kontroll). Siden SIFT ikke støtter slike funksjoner, overlates dette til den enkelte applikasjon. Ulike implementasjoner (databaser) kan derfor ha svært ulike egenskaper.

SIFT benytter et eget søkespråk, og det finnes tre typer søk [19]:

- *Boolsk søking* kan benytte operatorene “og”, “eller” og “og ikke”. Slik søking benyttes når man vet nøyaktig hvilke krav som skal stilles til søkeresultatet, f.eks. “Brundtland og Stortinget og ikke EU”. Det er dessuten mulig å stille krav til søkeordenes nærhet i teksten, f.eks. i samme

setning/avsnitt eller i gitt avstand til hverandre. Feltsøk og fritekstsøk kan blandes vilkårlig.

- *Konseptorbasert søking* er spesielt rettet mot fritekstsøking. Her rangeres dokumentene i databasen ut fra deres innbyrdes sannsynlighet for å være relevant.
- *Assosiativ søking* finner alle dokumenter som har en viss likhet med de dokumenter som allerede er funnet.

Søk kan høyre-, midt- og venstretrunkeres med vilkårlig antall bokstaver eller med én bokstav. F.eks. kan * (stjerne) benyttes for vilkårlig antall bokstaver etter, foran eller midt i det søkeordet som oppgis. Søk basert på 3-4 trunkerte søkeord vil som regel gi responstider på under ett sekund, selv i databaser med over én million dokumenter [19]. Det er mulig å gjenbruke og utvide spørsmål, og søkemakroer for hyppig benyttede søk kan defineres. En kan bla i resultatlister og skrive ut til skjerm, skriver eller fil, evt. via en innebygget rapportgenerator.

Det er nylig utviklet et Windows-basert grensesnitt mot SIFT Plus[20]. Dette grensesnittet skal være mer brukervennlig, men søkefunksjonaliteten er den samme som for tradisjonell SIFT.

Vedlegg B: Programstatistikk

Vedlegg C: Fjernsynsproduksjon

Det er ikke bare form og innhold som er forskjellig for de ulike programtypene. Selve produksjonsmåten er også ganske forskjellig, bl.a. når det gjelder følgende faktorer (basert på [11]):

- *Produksjonsteam.* Det varierer hvor mange personer som deltar i produksjonsarbeidet. Nyhetsteam kan være små, f.eks. 3 personer, mens for større sports- eller dramaproduksjoner kan opptil 30 personer være involvert.
- *Økonomiske rammer.* Store, omfattende produksjoner koster mer enn små. Dessuten kan prioriteringer og valg påvirke det endelige budsjettet.
- *Produksjonstid.* Det kan ta måneder og år å produsere f.eks. drama- eller dokumentarserier. For Dagsrevyen er selvsagt så lange produksjonstider helt utenkelig, fordi stoffet skal presenteres i daglige sendinger mens det ennå har nyhetens interesse.
- *Planlegging.* Hvor omfattende planleggingsarbeid som nedlegges i en produksjon, er direkte begrenset av produksjonstiden. Kort produksjonstid begrenser mulig planlegging, mens for f.eks. en dramaproduksjon utarbeides det gjerne *dreiebok* (detaljert billedbeskrivelse av hver scene), *dekorasjonsbreakdown* (detaljert planlegging av interiør og eksteriør), *produksjonsplan* (detaljert ukeplan) og *storyboard* (tegninger av hver scene).
- *Opptakssted.* Noen programmer spilles inn i studio, andre på *location*.
- *Arkivmateriale* er velegnet for enkelte programtyper (f.eks. Dagsrevyen og dokumentarprogrammer), og mindre aktuelt for andre (f.eks. nye fjernsynsfilmer eller dramaserier).
- *Medvirkende.* Enkelte programmer har et stort persongalleri som også inkluderer skuespillere, mens andre programmer ikke har dette behovet.
- *Redigering.* En rekke faktorer bestemmer hvor mye redigering som er nødvendig, f.eks. programtype, opptakskvalitet og programmets lengde. Redigering har alltid vært en flaskehals i NRKs produksjonskjede, og tilgjengelig redigeringskapasitet har derfor også betydning.
- *Sending.* Programmer kan være opptak, direktesendinger eller en kombinasjon av begge. Dagsrevyen består f.eks. av direktesendte intervjuer og *studio-kommentarer* (programleders opplesning), mens innslagene er opptak.
- *Andre forhold* skiller også de ulike programtypene fra hverandre: *Rammebetingelser*, f.eks. tariffier eller internasjonale avtaler. *Møter* med eksterne kontakter, f.eks. med arrangører av sportsbegivenheter. *Markedsføring*, f.eks. produksjon av *trailere* (kort forhåndsreklame som vises mellom de ordinære programmene).

Vedlegg D: Produksjon av Dagsrevyen

Følgende aktører er sentrale ved produksjon av Dagsrevyen:

- *Telegrambyrå*. Leverer nyhetsmeldinger som ofte er utgangspunkt for saker.
- *Reporter*. Utarbeider manuskript (tekst) og innslag (video), som f.eks. baseres på nyhetsmeldinger fra telegrambyråer, saker fra radioens Dagsnytt eller avisene, eller egen *research* (undersøkelser av arkivmateriale, litteratur, personintervjuer etc).
- *Producer*. (Også kalt *regissør* og *sendeleder*). Bistår reporter med produksjon av billedmateriale, og utarbeider *skuddliste*, som er de mest nødvendige opplysninger om avviklingen av sendingen, f.eks. beskrivelse av kamerabruk, bildekomposisjon og *stillstore* (grafikk). I Dagsrevyen er producer også sendeleder, altså ansvarlig for gjennomføring av selve sendingen. *Director* er den vanlige internasjonale betegnelse på det som i NRK kalles *producer*.
- *Skript*. Noe forenklet kan skripten sies å være producers sekretær. Skripten sørger bl.a. for å skaffe medvirkende og å utarbeide *kjøreplan*, som er en kortfattet oversikt over innslag og rekkefølge på disse. Under sending er skripten dessuten ansvarlig for såkalte *supertider*, som er tidspunkt for pålegging av tekst og grafikk oppå bildet.
- *Vaktsjef*. (Også kalt *ansvarlig redaktør* eller *kveldsredaktør*). Redigerer reporterne saker (tekst) og prioriterer mellom dem.
- *Reportasjeteam*. (For nyhetsproduksjon også kalt *ENG-team*, *Electronic News Gathering*). Produksjonsteam for utendørs énkamera-opptak. Ansvar for lyd og bilde ivaretas av to personer, eller en person har ansvaret for begge deler. Utendørs produksjonsteam som benyttes for andre programtyper er *EFP-team* (Electronic Film Production, énkamera-team som også inkluderer person med lys-ansvar) og *OB-team* (Outdoor Broadcasting, flerkamera-team, typisk 2-28 kameraer).
- *Redigerer*. Dagsrevyen har 12 egne redigerere som med Dagsrevyens utstyr kan utføre klippearbeid. For bruk av effekter, film eller ukurante båndformat, overlates oppdraget til Båndfilmsentralen i NRK.
- *Programleder*. Setter sitt personlige preg på studiókomentarene⁴⁵, og leser disse under sending. Programleder presenterer også resultatet av det omfattende produksjonsarbeidet som ligger bak.

⁴⁵ Programleder starter med reporters forslag til innledning, og skriver tilpasser selv en versjon som benyttes under sendingen. Enkelte programledere *skriver* f.eks. slike studiókomentarer på sin egen dialekt.

- *Andre*. Et stort antall andre personer er i tillegg til de ovenfor nevnte personer uunnværlige i nyhetsproduksjon, f.eks. *TOM* (Technical Operation Manager, teknisk leder), *redaksjonsassistenter* (skaffer bl.a. arkivmateriale for reporter eller producer), *mikser*, *tekster* og *studiopersonell*.

Vedlegg E: Kjøreplan for Dagsrevyen

Vedlegg F: Studiokommentarer, Dagsrevyen

Vedlegg G: Video-Annoteringsverktøy

G.1 Kontekst

Første versjon av annotasjonsverktøyet ble implementert av Wasskog og Bekkadal [47]. Verktøyet grensesnitt og funksjonalitet hadde en del begrensninger, og dette dokumentet beskriver arbeidet med å forbedre en del svakheter.

Annotasjonsverktøyet bygger på og benytter en videoklient (spiller/grensesnitt) som igjen benytter en videoserver. Server og grensesnitt ble først implementert av Skeide [46] (basert på basisfunksjoner fra Televerkets Forskningsinstitutt), og grensesnittet ble senere forbedret av Wasskog og Bekkadal [47]. Sammenhengen mellom annotasjonsverktøy, videoklient og videoserver kan noe forenklet betraktes slik (for nærmere detaljer om videoserver-arkitektur henvises til [45]):

Videoserver ↔ Videoklient/Grensesnitt/Videospiller ↔ Annotasjonsverktøy

For nåværende implementasjon av videospilleren er digital video organisert som separate *filer* for hvert TV-program. Av ytelseshensyn er det en fordel å fordele filene over flere disk og arbeide mot logiske *strømmer*. Selv om nåværende videospiller bare kan arbeide mot filer, blir den nye versjonen av annotasjonsverktøyet også forberedt for videostrømmer. Langørgen har i sommer arbeidet med design og eksperimenter på en parallell maskinplattform [45], og senere videreført dette arbeidet i diplomoppgaven.

G.2 Forbedringer

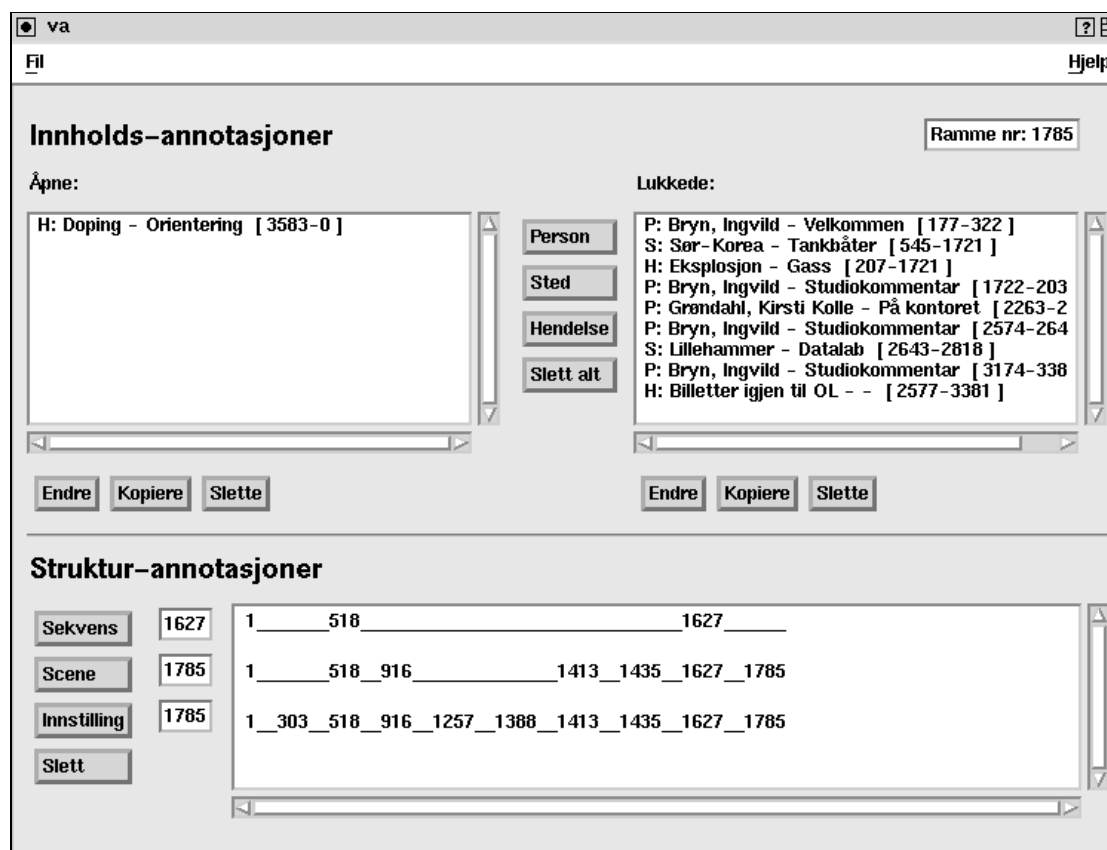
Følgende forbedringer var ønskelige i utgangspunktet eller oppsto i løpet av arbeidet:

1. Forbedre *grensesnittet* for annotering. Dialogbokser for registrering, endring og kopiering av annotasjoner.
2. Opprette databaser for personer, steder og hendelser. Annotasjoner kobles til objekter fra disse databasene ved at objekter velges fra en valgliste, eller det kan opprettes nye objekter. Dette reduserer nødvendig inntasting og faren for feilregistrering. Tidligere radio-buttons for valg av de tre annotasjonstypene erstattes med tre tilsvarende trykknapper, som hver åpner en dialogboks med riktig valgliste.
3. Lagre og hente annotasjonsfiler for evt. å kunne arbeide videre med dem senere.
4. Kopiere og endre registrerte annotasjoner - både åpne og lukkede.
5. Navngi strukturannotasjoner.

For å utføre en del av disse forbedringene, var det nødvendig med en omfattende endring av data- og filstrukturer, som blir nærmere omtalt i dette notatet. I det følgende beskrives verktøyet slik det nå foreligger i *ny versjon*, mens gammel versjon altså er beskrevet i [47].

G.3 Grensesnitt og funksjonalitet

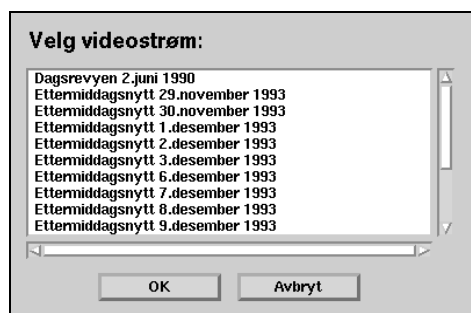
Hovedvinduet for verktøyet ser ut nå som følger:



Figur G.1, Hovedvinduet for annotasjonsverktøyet

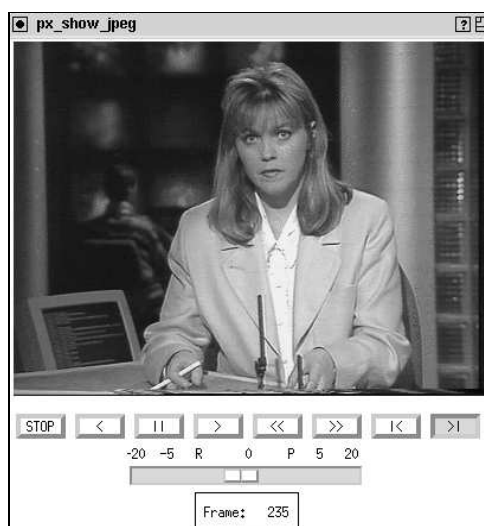
Vinduet har fått en meny, og de to hoveddelene for hhv. innholds- og strukturannotasjoner er beholdt. Layouten er imidlertid forbedret noe, og det er laget en ressursfil som bl.a. setter farger og skrifttyper.

Fra Fil-menyen velges "Start video", som åpner dialogboksen i Figur G.2. I denne dialogboksen velges et program, og videospilleren starter. Oppstart av video er dermed blitt mer brukervennlig siden forrige versjon, hvor brukeren selv måtte finne riktig filnavn. Selv om hvert navn fortsatt svarer til én fil, kan en evt. på denne måten senere fremstille en samling videofiler som én videostrøm.



Figur G.2, Dialogboks for valg av program

Når videospilleren startes, åpnes samtidig tilhørende fil med innholds-annotasjoner, dersom denne eksisterer. Videospilleren er uendret fra forrige versjon, og grensesnittet for denne ser ut som følger:



Figur G.3, Digital videospiller

For innholds-annotasjoner benyttes øvre del av hovedvinduet. For å opprette en ny annotasjon, trykker en på knappen “Person”, “Sted” eller “Hendelse”. Dialogboksen i Figur G.4 kommer da til syne.

Figur G.4, Dialogboks for registrering av innholds-annotasjoner

I dette konkrete tilfellet var det “Hendelse” som ble valgt. Rammenummeret som videospilleren viste ved trykk på “Hendelse”, blir automatisk fylt inn i startramme-feltet, men nummeret kan også overstyres av brukeren. Databaseobjektet “Doping” ønskes, men dette står ikke i valglisten (fordi objektet ennå ikke registrert i databasen). Brukeren skriver derfor inn teksten selv, og neste gang vil objektet kunne velges direkte. Som en presisering skriver brukeren også inn annotasjonen “Orientering”. Ved trykk på “OK”, blir posten lagt i listen over “åpne annotasjoner” i hovedvinduet. (Er også sluttramme tastet inn, legges den istedet i listen over “lukkede annotasjoner”).

Samme dialogboks benyttes dersom brukeren ønsker å “Endre” eller “Kopiere” en annotasjon i hovedvinduet. Teksten “Ny” i dialogboksen blir da erstattet med hhv. “Endre” eller “Kopiere”. Feltene er dessuten fylt inn med de verdier som gjelder aktuell post.

For struktur-annotasjoner benyttes nedre del av hovedvinduet. Et trykk på “Sekvens”, “Scene” eller “Innstilling” markerer overgangen mellom to påfølgende strukturelement (kalt ramme-

sekvenser i [16]). For alle knappetrykk (unntatt for starten på første sekvens), “lukkes” den tilhørende struktur-annotasjon. Siden elementene er strengt hierarkisk ordnet, lukkes samtidig struktur-annotasjoner på lavere nivå. Ved trykk på “Scene” vil f.eks. både en scene og en innstilling lukkes, og disse to struktur-annotasjonene kan da navngis i følgende dialogboks:

Figur G.5, Dialogboks for navngivning av struktur-annotasjoner

Det ble besluttet at struktur-annotasjoner skulle ha lavere prioritet under implementeringen. Bortsett fra navngivning, er derfor funksjonaliteten for struktur-annotasjoner uendret fra forrige versjon. Dette betyr at det fortsatt ikke er mulig å endre eller kopiere struktur-annotasjoner, og en kan heller ikke hente fram lagrede struktur-annotasjoner for å arbeide videre på dem. Endring og kopiering forutsetter interaksjon med brukeren, men visningslisten for struktur-annotasjoner (til høyre for trykknappene), er bare en avbildning av strukturen, og derfor ikke tilstrekkelig for slik interaksjon.

G.4 Widgets

Hovedvinduet og dialogboksene er konstruert av en rekke større og mindre grafiske elementer (*widgets*). Av disse er to typer av særlig betydning for verktøyets funksjonalitet og brukerinteraksjon: *visningslister* i hovedvinduet (Figur G.1) og *valglister* i dialogboksene.

Hovedvinduet inneholder to lister som viser de registrerte annotasjonene:

- `aapenAnnot`. Annotasjoner som foreløpig bare har fått registrert start- eller sluttramme.
- `lukketAnnot`. “Ferdigregistrerte” annotasjoner som både har start- og sluttramme.

Visningslisten for struktur-annotasjoner er uendret fra forrige versjon.

Dialogboksene har som nevnt valglister:

- `dVideostreams`. Valgliste i dialogboks (Figur G.2) for valg av program.
- `dAnnot`. Valgliste i dialogboks (Figur G.4) for valg av databaseobjekt.

G.5 Fil- og datastruktur

Wasskog og Bekkadals versjon av verktøyet [47] benytter én stor fil for alle annotasjoner, og annotasjoner blir kun skrevet til disk - aldri lest inn igjen. Da det ikke er ønskelig å lese inn annotasjoner til *alle* videostrømmene hver gang (og siden skrive disse tilbake igjen), er den tidligere data- og filstruktur lite egnet. Dessuten bør annotasjonsverktøyet på sikt ikke være begrenset til en bestemt database. En mer fleksibel data- og filstruktur, bedre tilpasset den

utvidede funksjonaliteten, blir presentert i det følgende. Strukturen er utarbeidet med utgangspunkt i datamodellen presentert i [16].

G.5.1 Katalog

Det er definert en egen datatype, `katalogPost`, for kataloginformasjon om programmer:

<code>videostreamID</code>	Integer. Unik ID for videostrøm. Blir foreløpig ikke benyttet.
<code>videostreamNavn</code>	String. Vilkårlig navn for en videostrøm, f.eks. navn på program.
<code>annotFilnavn</code>	String. Filnavn for tilhørende innholds-annotasjoner. Struktur-annotasjoner benytter samme navn, men med tillegg '.s' etter filnavnet.
<code>next</code>	Peker til neste post (NULL hvis siste).

`katalogPtr` er en peker til starten på en liste av slike katalogposter.

Katalogen leses inn fra fila `vs-ann.kat` på disk, og formatet er som følger:

```
<nesteAnnotID>
<nesteDbID>
<antallPoster> /* Disse tre feltene skrives én gang */

<videostreamNavn.1>
<videostreamID.1>
<annotFilnavn.1>
<videostreamNavn.2>
:
:
```

Her viser `nesteAnnotID` og `nesteDbID` hva som hhv. er neste ledige felles, unike (integer) ID for annotasjoner og databaseobjekter. `antallPoster` viser hvor mange poster (integer) som følger i katalogfila, dvs. hvor mange videostrømmer som er beheftet med annotasjonsfiler. Alle disse tre feltene er høyrejustert og har lengden 10 (med blanktegn på venstre side). Dette for å lette skrivning til katalogfil under kjøring av programmet, da disse tre feltene blir *overskrevet* med nye verdier.

`videostreamNavn.i` er et fritt valgt navn på videostrøm nummer *i*, f.eks. "Ettermiddagsnytt 17.desember 1993" eller "Tatt av vinden". `videostreamID.i` identifiserer entydig denne videostrømmen. I annotasjonsverktøyet benyttes en videofil pr. videostrøm, slik at `videostreamID.i` istedet inneholder filnavn. Dette gjør det mulig å foreløpig benytte standard videofiler, samtidig som verktøyet er åpent med tanke på fremtidige løsninger for videoklient og -server (se f.eks. [45]). `annotFilnavn.i` er navnet på tilhørende annotasjonsfil.

I en begrenset prototyp som annotasjonsverktøyet, med bare et fåtall digitaliserte nyhets-sendinger, vil ikke katalogfila endre seg over tid. Pr. i dag er katalogfila derfor hardkodet, og må skrives inn manuelt ved evt. endringer.

G.5.2 Innholds-annotasjoner

Det er definert en egen datatype, `annotPost`, for informasjon om innholds-annotasjonsposter:

annotID	Integer. Unik ID som følger annotasjonen i hele dens levetid. ID er felles for alle annotasjonstyper og annotasjonsfiler.
annotType	Char. Mulige verdier: { P S H } (Person, Sted, Hendelse).
startramme	Integer.
sluttramme	Integer. (Constraint: startramme < sluttramme).
dbID	Integer. Fremmednøkkel inn i aktuell database, knytter annotasjonen til riktig databaseinnslag. Hvilken base det gjelder, er gitt av annotType. For strukturannotasjoner er dbID = 0.
annotasjon	String. Annotasjon på fritekst-form (Optional).
next	Peker til neste post (NULL hvis siste).
prev	Peker til forrige post (NULL hvis første).

De to pekerne `aapenAnnotPtr` og `lukketAnnotPtr` peker til starten på lister av hhv. åpne og lukkede innholdsannotasjoner. Innholdet i disse to listene vises hhv. i visningslistene `aapenAnnot` og `lukketAnnot`.

Da det bare er aktuelt å arbeide med én videostrøm om gangen, er det tilstrekkelig å lagre og hente de annotasjoner som er direkte relatert til denne. Det er derfor praktisk å benytte én annotasjonsfil pr. videostrøm. Annotasjonsfilene kan ha vilkårlig navn (definert i katalogfil), og formatet er som følger:

```
<antallPoster> /* Dette feltet skrives én gang */
<startramme> <sluttramme> <annotID> <annotType> <dbID>
<annotasjon>
:
:
```

`antallPoster` (integer) viser hvor mange poster som følger i annotasjonsfila, dvs. hvor mange annotasjoner som er lagret. Feltene på andre linje samsvarer direkte med feltene i de interne annotasjonslistene.

G.5.3 Struktur-annotasjoner

Siden struktur-annotasjoner hadde lavere prioritet ved implementeringen, ble den enkle datastrukturen fra forrige versjon bare utvidet med en streng for å holde strukturelementets (rammesekvensens) tittel. Datatype for informasjon om struktur-annotasjonsposter er definert som følger:

frameNr	Integer. Rammenummer som angir overgangen mellom to struktur-element (rammesekvenser).
tittel	String. Strukturelementets navn (annotasjon).
next	Peker til neste post (NULL hvis siste).

Det benyttes tre lister, som hver har en peker til første og siste element:

- Sekvenser: `struktSekvensStart` og `struktSekvensSlutt`
- Scener: `struktSceneStart` og `struktSceneSlutt`
- Innstillinger: `struktInnstillingStart` og `struktInnstillingSlutt`

Innholdet i disse tre listene vises i visningslisten `strukturVis`.

Struktur-annotasjonsfilene har samme filnavn som sekvens-annotasjonsfilene, men med tillegg '.s' etter filnavnet. Formatet er som følger:

```
<antallPoster> /* Dette feltet skrives én gang */
<type> <startramme> <sluttramme>
<tittel>
:
:
```

`antallPoster` (integer) viser hvor mange poster som følger i struktur-annotasjonsfila, dvs. hvor mange strukturelementer (rammesekvenser) som er lagret. `type` er enten SE (sekvens), SC (scene) eller IN (innstilling). `startramme` og `sluttramme` markerer hhv. start og slutt på et strukturelement, og to påfølgende poster undersøkes for å skrive begge disse feltene.

G.5.4 Databaseobjekter

Det er definert en egen datatype, `dbPost`, for informasjon om databaseobjekter:

<code>dbID</code>	Integer. Unik ID for hvert databaseinnslag. ID'en er felles for alle databasefiler.
<code>dbInnslag</code>	String. Innslag i databasen (Personnavn, Stednavn eller Nøkkelord). Dette feltet kan evt. utvides for å få med flere attributter.
<code>antallAnnot</code>	Integer. Antall annotasjoner som refererer til dette databaseobjektet. Feltet settes til 1 når objektet opprettes, men er foreløpig ikke i bruk, og blir ikke oppdatert.
<code>next</code>	Peker til neste post (NULL hvis siste)
<code>prev</code>	Peker til forrige post (NULL hvis første)

De tre pekerne `dbPersonPtr`, `dbStedPtr` og `dbHendelsePtr` peker til starten på hver av de tre databaselistene. Innholdet i den aktuelle av disse tre listene, vises i valglisten `dAnnot` (dialogboksen i Figur G.4).

De tre databaselistene lagres i separate filer med fast navn:

- `db-person`
- `db-sted`
- `db-hendelse`

Alle databasefilene har følgende format:

```
<antallPoster> /* Dette feltet skrives én gang */
<dbInnslag>
<dbID> <antallAnnot>
:
:
```

`dbID` er felles for alle de tre databasefilene, og gjør det mulig å senere benytte andre databasesystemer og -tabeller. `antallAnnot` viser hvor mange annotasjoner som refererer til databaseobjektet, men feltet er altså foreløpig ikke i bruk. I annotasjonsverktøyet kan databaseobjekter bare opprettes, ikke slettes eller endres.

Alle poster i databasefilene er sortert alfabetisk etter `dbInnslag`. For evt. annen indeksering, må det f.eks. opprettes egne filer.

G.5.5 Sammenheng

Det er altså følgende sammenheng mellom filene på *disk*, de tilsvarende strukturer i *memory* og hvilke *widgets* de vises i:

<i>DISK:</i>	Katalogfil vs-ann.kat	Annotasjonsfiler annotfil.1 annotfil.2 annotfil.3 : annotfil.n	Databasfiler db-person db-sted db-hendelse	Strukturfiler annotfil.1.s annotfil.2.s annotfil.3.s : annotfil.n.s
	↑	↑	↑	↑
<i>MEM:</i>	Katalogliste katalogPtr ↔	Annotasjonslister aapenAnnotPtr ↔ lukketAnnotPtr ↔	Databaselister dbPersonPtr dbStedPtr dbHendelsePtr	Strukturlister struktSekvensStart struktSceneStart struktInnstillingStart
		↑	↑	↑
<i>WIDGETS:</i>		Visningslister aapenAnnot lukketAnnot	Valgliste dAnnot	Visningsliste strukturVis

G.6 Beskrivelse av implementasjon

Funksjonene i programmet er forsøkt navngitt slik at de best mulig reflekterer den oppgaven som utføres, og programlistingen er dessuten kommentert for å lette lesbarheten. Programmet er i hovedsak delt opp i tre filer:

- `globalt.h`: Header-fil som definerer konstanter, datatyper, globale variable og globale widgets.
- `va.c`: Video Annotering, som definerer grensesnittet og callback-funksjoner.
- `liste.c`: Liste- og filbehandlingsfunksjoner.

De følgende avsnitt beskriver hovedtrekkene i programmet.

G.6.1 Oppstart

Hovedvinduet bygges opp, og det legges *callbacks* (funksjoner) til de ulike *widgets* (grafiske elementer). Katalogfil leses for å bygge katalogliste med tilgjengelige programmer. ID'er som er lagret i katalogfil blir også lest inn. Databasefilene leses og legges inn i databaselistene. Hovedvinduet er sperret for bruker-input inntil et program velges.

G.6.2 Starte video

Brukeren velger fra en dialogboks det programmet som ønskes annotert. Dersom det allerede eksisterer en innholds-annotasjonsfil, leses denne inn i visningslisten for lukkede annotasjoner. Videospilleren starter det valgte programmet.

G.6.3 Registrere innholds-annotasjoner

Registrering av innholds-annotasjoner tillates bare når videospilleren er aktiv (viser et program). Dialogboksen i Figur G.4 benyttes for dette, og når brukeren godkjenner innholdet, blir annotasjonsposten lenket inn i `aktivAnnotPtr` (dersom bare startramme er gitt) eller `lukketAnnotPtr` (dersom også sluttramme er gitt). Visningslister i hovedvinduet viser de viktigste data som er registrert.

G.6.4 Registrere struktur-annotasjoner

Registrering av struktur-annotasjoner tillates bare når videospilleren er aktiv (viser et program). Trykknappene i hovedvinduet registrerer rammenummer som videospilleren viser i øyeblikket, og når en slik struktur blir “lukket”, benyttes dialogboksen i Figur G.5 for å angi tittel. Brukeren kan om ønskelig bare navngi høyeste nivå, slik at denne tittelen også blir benyttet for lavere nivå. Alternativt kan navn utelates helt, slik at tittelen hhv. får det generiske navnet “Sekvens”, “Scene” eller “Innstilling”. Visningsliste i hovedvinduet viser alle rammenummer for overganger mellom strukturelementer (rammesekvenser).

G.6.5 Slette annotasjoner

Det er mulig å slette alle annotasjoner (innhold og strukturer) ved å velge “Slett alt”. Det er dessuten mulig å slette alle struktur-annotasjoner eller en bestemt innholdsannotasjon ved å trykke på de respektive knapper. Dersom en innholdsannotasjon slettes, blir denne lenket ut av aktuell annotasjonsliste. Databaseinnslag blir aldri slettet, selv om alle tilhørende annotasjoner slettes.

G.6.6 Lukke innholds-annotasjoner

En åpen innholds-annotasjon kan “lukkes” vha. et dobbelt museklikk i listen for åpne annotasjoner. Annotasjonen flyttes da over til listen for lukkede annotasjoner med sluttramme lik den spilleren viser i øyeblikket. For endring av annotasjonen i forbindelse med lukkingen, benyttes trykknappen “Endre”.

G.6.7 Endre innholds-annotasjoner

Ved endring av innholds-annotasjoner benyttes dialogboksen i Figur G.4 med alle felt *default* fylt inn med kilde-annotasjonens verdier. Annotasjonen kan knyttes til et annet databaseinnslag av samme type (valgliste er tilgjengelig), og selve annotasjonsteksten kan endres. Det er også mulig å manuelt endre start- og sluttramme, og annotasjonen lenkes til slutt tilbake til riktig annotasjonsliste. Annotasjonen blir lukket dersom både start- og sluttramme er registrert, og startamme < sluttramme.

G.6.8 Kopiere innholds-annotasjoner

En innholds-annotasjon kan “åpnes” ved å dobbeltklikke på en lukket annotasjon. Dette oppretter en kopi hvor startamme settes til det rammenummer videospilleren viser i øyeblikket.

Trykknappene “Kopiere” tilbyr mer sofistikerte kopieringsmuligheter, med mulighet for endring i forbindelse med kopieringen. Kopieringen kan foregå i begge retninger, og fører til at det blir opprettet en ny annotasjon (med ny `annotID`), samtidig som den opprinnelige annotasjonen blir bevart. Funksjonen kan sammenliknes med en registrering hvor alle felter *default* er fylt inn med kilde-annotasjonens verdier.

G.6.9 Lagre annotasjoner

Når videospilleren stopper, får brukeren tilbud om å lagre annotasjonene. Strukturannotasjoner og lukkede innholdsannotasjoner blir lagret i to separate filer. Åpne annotasjoner blir ikke lagret. Skal disse bevares, må de først lukkes. I forbindelse med lagring blir også databasen oppdatert ved at databaselistene skrives til databasefilene.

Vedlegg H: Kildekode for browseverktøy

```

/*****
Browse-verktøy for digital video
Tormod Holte, 29.11.94 - 14.12.94
*****/

/*** Header-filer ***/
#include <stdio.h>
#include <string.h>
#include <Xm/Xm.h>
#include <Xm/Label.h>
#include <Xm/Form.h>
#include <Xm/TextF.h>
#include <Xm/List.h>
#include <Xm/Frame.h>
#include <Xm/PushB.h>
#include <Xm/MainW.h>
#include <Xm/MessageB.h>
#include <Xm/SelectioB.h>
#include <Xm/Separator.h>
#include <Xm/RowColumn.h>
#include <Xm/FileSB.h>
#include <Xm/DialogS.h>
#include <Xm/PanedW.h>
#include <X11/Xatom.h>

/*** Konstanter ***/
#define VINDU_X 1000 /* vinduets bredde */
#define VINDU_Y 400 /* vinduets hoyde */
#define KNAPP_X 110 /* knappens bredde */
#define NAV_X 53 /* x-koordinat for navigeringsknappenes senter */
#define NAV_Y 77 /* y-koordinat for navigeringsknappenes senter */
#define NAV_S 50 /* sidekantenes lengde for navigeringsknappene */
#define MAXSTRLEN 200 /* maksimal lengde som tillates for strenger */
#define KATALOGFIL "vs-ann.kat"
#define VIDEOSPILLER "show_video"
#define HYPHEN_F "-F"
#define HYPHEN_A "-A"
#define HYPHEN_B "-B"

#define XtNbrowseWindow "Annotation Window"
#define XtNframeNumber "Frame Number"
#define XtNstop "Stop"
#define XtNvideofunksjon "Videofunksjon"
#define XtNplayerWindow "Player"
#define XtNstartramme "Startramme"
#define XtNsluttramme "Sluttramme"

const char FWD='1',REV='2',HLT='3',FFWD='4',FREV='5',INC='6',
DEC='7',STP='8',SLIDE='9',BROWSE='B';

/*** Type-definisjoner ***/
typedef struct programPost programPost;
struct programPost
{
    char *tittel;
    char *videoFilnavn;
    char *annotFilnavn;
    programPost *neste;
};

typedef struct strukturPost strukturPost;
struct strukturPost
{
    int startramme;
    int sluttramme;
    char *tittel;
    strukturPost *neste;
    strukturPost *forrige;
    strukturPost *opp;
    strukturPost *ned;
};

/*** Variable ***/
programPost *startProgramPtr=NULL; /* Starten paa programliste */
strukturPost *startSekvensPtr=NULL; /* Starten paa sekvensliste */
strukturPost *startScenePtr=NULL; /* Starten paa sceneliste */
strukturPost *startInnstillingPtr=NULL; /* Starten paa innstillingliste */

strukturPost *sisteSekvensPtr=NULL; /* Siste registrerte sekvens */
strukturPost *sisteScenePtr=NULL; /* Siste registrerte scene */
strukturPost *sisteInnstillingPtr=NULL; /* Siste registrerte innstilling */

programPost *valgtProgramPtr=NULL; /* Peker til valgt program */
strukturPost *valgtSekvensPtr=NULL; /* Peker til valgt sekvens */
strukturPost *valgtScenePtr=NULL; /* Peker til valgt scene */
strukturPost *valgtInnstillingPtr=NULL; /* Peker til valgt innstilling */
strukturPost *browsePtr; /* Peker til browse-element */

int videoSpilling = FALSE;
int rammeNummer = 0;

/*** Widgets ***/
Widget topLevel, form, vindu;
Widget avsluttKnapp;
Widget navigering, opp, ned, neste, forrige;
Widget stop, pause, play;

Widget programRamme, programLedetekst, programListe,
programValgt;
Widget visProgram;
Widget sekvensRamme, sekvensLedetekst,
sekvensListe, sekvensValgt;
Widget visSekvens;
Widget sceneRamme, sceneLedetekst,
sceneListe, sceneValgt;
Widget visScene;
Widget innstillingRamme, innstillingLedetekst,
innstillingListe, innstillingValgt;
Widget visInnstilling;

/*** Funksjoner ***/

/* Legg path foran filnavn */
char *fulltFilnavn(char *path, char *fil)
{
    char *fulltNavn = (char *) malloc(strlen(path) + strlen(fil) + 1);

    strcpy(fulltNavn, path);
    strcat(fulltNavn, fil);
    return fulltNavn;
}

void fyllStrukturliste(Widget listeboks, strukturPost *rotnode)
{
    XmString string;
    strukturPost *ptr;
    int antall = 0;
    int listepos = 1;

    ptr = rotnode;

    while (ptr != NULL) /* gjenta for hele lista paa aktuelt nivå */
    {
        string = XmStringCreateSimple(ptr->tittel);
        XmListAddItemUnselected(listeboks,
            string, listepos); /* legger inn i programliste */
        ptr = ptr->neste;
        listepos++;
        XmStringFree(string);
    } /* fyllStrukturliste */

    /*** Registrer vinduet i serveren ***/
    void registrerVindu(Widget w)
    {
        Window window;
        Display *display;
        Atom BROWSEVINDU;
    }
}

```

```

window = XtWindow(w);
display = XtDisplay(w);
BROWSEVINDU = XInternAtom(display, XtNbrowseWindow,
FALSE);

XChangeProperty(display, DefaultRootWindow(display),
BROWSEVINDU, XA_WINDOW,
32, PropModeReplace,
(unsigned char *) &window, 1);
}

void avsluttCB(w, client_data, call_data)
Widget w;
XtPointer call_data, client_data;
{
Window window;
Display *display;
Atom BROWSEVINDU, type;
unsigned long nitems, left;
Window *retdata1;
int format;

window = XtWindow(w);
display = XtDisplay(w);
BROWSEVINDU = XInternAtom(display, XtNbrowseWindow,
FALSE);

/* Slett property fra X-serveren */
XGetWindowProperty(display,
DefaultRootWindow(display),
BROWSEVINDU,
0, 4, TRUE, XA_WINDOW,
&type, &format, &nitems, &left,
&retdata1);
exit(0);
}

void browseVideo (int startramme, int sluttramme)
{
char frame_str[6];

Display *display;
Window spillerVindu;
Atom PLAYER_WINDOW, STARTRAMME_ATOM,
SLUTTRAMME_ATOM, type;
unsigned long nitems, left;
Window *retdata;
int format, status;
XClientMessageEvent clientMessageEvent;

/* Finn spillerens id fra X-serveren (display) */
display = XtDisplay(vindu);
PLAYER_WINDOW = XInternAtom(display, XtNplayerWindow,
FALSE);
if (XGetWindowProperty(display,
DefaultRootWindow(display),
PLAYER_WINDOW,
0, 4, FALSE, XA_WINDOW,
&type, &format, &nitems, &left,
&retdata) == Success && type == XA_WINDOW)
{
spillerVindu = *retdata;

clientMessageEvent.display = display;
clientMessageEvent.window = spillerVindu;
clientMessageEvent.type = ClientMessage;
clientMessageEvent.format = 8;
STARTRAMME_ATOM = XInternAtom(display, XtNstartramme,
FALSE);
clientMessageEvent.message_type = STARTRAMME_ATOM;

/* Data som skal sendes (startramme) */
clientMessageEvent.data.l[0] = startramme;

/* Send event til spilleren */
status = XSendEvent(display, spillerVindu,
TRUE, NoEventMask, &clientMessageEvent);

/* Flush display slik at spilleren detekterer eventen */
XFlush(display);
}
}

}

void videoFunksjon(Widget w, XtPointer client_data, XtPointer
call_data)
{
Display *display;
Window spillerVindu;
Atom PLAYER_WINDOW, FUNKSJON_ATOM, type;
unsigned long nitems, left;
Window *retdata;
int format, status;
XClientMessageEvent clientMessageEvent;
char funksjon = (char) client_data;

/* Finn spillerens id fra X-serveren (display) */
display = XtDisplay(vindu);
PLAYER_WINDOW = XInternAtom(display, XtNplayerWindow,
FALSE);
if (XGetWindowProperty(display,
DefaultRootWindow(display),
PLAYER_WINDOW,
0, 4, FALSE, XA_WINDOW,
&type, &format, &nitems, &left,
&retdata) == Success && type == XA_WINDOW)
{
spillerVindu = *retdata;

clientMessageEvent.display = display;
clientMessageEvent.window = spillerVindu;
clientMessageEvent.type = ClientMessage;
clientMessageEvent.format = 8;
FUNKSJON_ATOM = XInternAtom(display, XtNvideofunksjon,
FALSE);
clientMessageEvent.message_type = FUNKSJON_ATOM;

/* Data som skal sendes (funksjon) */
clientMessageEvent.data.l[0] = funksjon;

/* Send event til spilleren */
status = XSendEvent(display, spillerVindu,
TRUE, NoEventMask, &clientMessageEvent);

/* Flush display slik at spilleren detekterer eventen */
XFlush(display);
}
}

void oppdaterValgtFelt (strukturPost *ptr)
{
XmString innslag = XmStringCreateSimple (ptr->tittel);
XmString blank = XmStringCreateSimple (" ");
int *finnElementsListePosisjon(strukturPost *element);

/* Hvis innstilling */
if (ptr->ned == NULL)
XtVaSetValues(innstillingValgt, XmNlabelString, innslag, NULL);
XmListSelectPos(innstillingListe, finnElementsListePosisjon(ptr),
FALSE);

/* Hvis scene */
if ((ptr->ned != NULL) && (ptr->opp != NULL))
{
XmListDeleteAllItems(innstillingListe);
XmListSelectPos(sceneListe, finnElementsListePosisjon(ptr),
FALSE);
XtVaSetValues(sceneValgt, XmNlabelString, innslag, NULL);
XtVaSetValues(innstillingValgt, XmNlabelString, blank, NULL);
fylListe(innstillingListe, browsePtr->ned);
}

/* Hvis sekvens */
if (ptr->opp == NULL)
{
}
}

```

```

    XmlListDeleteAllItems(sceneListe);
    XmlListDeleteAllItems(innstillingListe);
    XtVaSetValues(sekvensValgt, XmNlabelString, innslag, NULL);
    XmlListSelectPos(sekvensListe, finnElementsListePosisjon(ptr),
FALSE);
    XtVaSetValues(sceneValgt, XmNlabelString, blank, NULL);
    XtVaSetValues(innstillingValgt, XmNlabelString, blank, NULL);
    fyllStrukturliste(sceneListe, browsePtr->ned);
}
}

void oppdaterNavigeringsknapper()
{
    /* Hvis videoen kjoerer og rammesekvens er valgt: enable relevante
knapper */
    if ((videoSpilling) && (browsePtr != NULL))
    {
        if(browsePtr->opp == NULL)
            XtSetSensitive(opp, FALSE);
        else
            XtSetSensitive(opp, TRUE);
        if(browsePtr->ned == NULL)
            XtSetSensitive(ned, FALSE);
        else
            XtSetSensitive(ned, TRUE);
        if(browsePtr->neste == NULL)
            XtSetSensitive(neste, FALSE);
        else
            XtSetSensitive(neste, TRUE);
        if(browsePtr->forrige == NULL)
            XtSetSensitive(forrige, FALSE);
        else
            XtSetSensitive(forrige, TRUE);
    }
    /* Hvis ikke: disable alle knapper */
    else
    {
        XtSetSensitive(opp, FALSE);
        XtSetSensitive(ned, FALSE);
        XtSetSensitive(neste, FALSE);
        XtSetSensitive(forrige, FALSE);
    }
}

void oppCB(Widget w, XtPointer client_data, XtPointer call_data)
{
    if (browsePtr != NULL)
    {
        if(browsePtr->opp != NULL)
        {
            browsePtr=browsePtr->opp;
            oppdaterValgtFelt(browsePtr);
            browseVideo (browsePtr->startramme, browsePtr->sluttramme);
        }
        oppdaterNavigeringsknapper();
    }
}

void nedCB(Widget w, XtPointer client_data, XtPointer call_data)
{
    if (browsePtr != NULL)
    {
        if(browsePtr->ned != NULL)
        {
            browsePtr=browsePtr->ned;
            oppdaterValgtFelt(browsePtr);
            browseVideo (browsePtr->startramme, browsePtr->sluttramme);
        }
        oppdaterNavigeringsknapper();
    }
}

void forrigeCB(Widget w, XtPointer client_data, XtPointer call_data)
{
    if (browsePtr != NULL)
    {
        if(browsePtr->forrige != NULL)
        {
            browsePtr=browsePtr->forrige;
            oppdaterValgtFelt(browsePtr);
            browseVideo (browsePtr->startramme, browsePtr->sluttramme);
        }
    }
}

    oppdaterNavigeringsknapper();
}
}

void nesteCB(Widget w, XtPointer client_data, XtPointer call_data)
{
    if (browsePtr != NULL)
    {
        if(browsePtr->neste != NULL)
        {
            browsePtr=browsePtr->neste;
            oppdaterValgtFelt(browsePtr);
            browseVideo (browsePtr->startramme, browsePtr->sluttramme);
        }
        oppdaterNavigeringsknapper();
    }
}

/* Navigeringsknapper for browser */
lagNavigering (Widget vindu)
{
    XmString tekst = XmStringCreateSimple("Opp");

    opp = XtVaCreateManagedWidget("navigering",
        xmPushButtonWidgetClass,
        vindu,
        XmNlabelString, tekst,
        XmNtopAttachment, XmATTACH_POSITION,
        XmNtopPosition, NAV_Y-13,
        XmNleftAttachment, XmATTACH_POSITION,
        XmNleftPosition, NAV_X-5,
        XmNwidth, NAV_S,
        XmNheight, NAV_S,
        NULL);
    XtAddCallback(opp, XmNactivateCallback, oppCB, NULL);
    XmStringFree(tekst);

    tekst = XmStringCreateSimple("Ned");
    ned = XtVaCreateManagedWidget("navigering",
        xmPushButtonWidgetClass,
        vindu,
        XmNtopAttachment, XmATTACH_POSITION,
        XmNtopPosition, NAV_Y+3,
        XmNleftAttachment, XmATTACH_POSITION,
        XmNleftPosition, NAV_X-5,
        XmNlabelString, tekst,
        XmNwidth, NAV_S,
        XmNheight, NAV_S,
        NULL);
    XtAddCallback(ned, XmNactivateCallback, nedCB, NULL);
    XmStringFree(tekst);

    tekst = XmStringCreateSimple("Forrige");
    forrige = XtVaCreateManagedWidget("navigering",
        xmPushButtonWidgetClass,
        vindu,
        XmNtopAttachment, XmATTACH_POSITION,
        XmNtopPosition, NAV_Y-5,
        XmNleftAttachment, XmATTACH_POSITION,
        XmNleftPosition, NAV_X-10,
        XmNlabelString, tekst,
        XmNwidth, NAV_S,
        XmNheight, NAV_S,
        NULL);
    XtAddCallback(forrige, XmNactivateCallback, forrigeCB, NULL);
    XmStringFree(tekst);

    tekst = XmStringCreateSimple("Neste");
    neste = XtVaCreateManagedWidget("navigering",
        xmPushButtonWidgetClass,
        vindu,
        XmNtopAttachment, XmATTACH_POSITION,
        XmNtopPosition, NAV_Y-5,
        XmNleftAttachment, XmATTACH_POSITION,
        XmNleftPosition, NAV_X,
        XmNlabelString, tekst,
        XmNwidth, NAV_S,
        XmNheight, NAV_S,
        NULL);
    XtAddCallback(neste, XmNactivateCallback, nesteCB, NULL);
    XmStringFree(tekst);
}
}

```

```

lagVideopanel (Widget vindu)
{
    XmString tekst;

    tekst = XmStringCreateSimple("O");

    stop = XtVaCreateManagedWidget("videopanel",
        xmPushButtonWidgetClass,
        vindu,
        XmNtopAttachment, XmATTACH_POSITION,
        XmNtopPosition, NAV_Y-5,
        XmNleftAttachment, XmATTACH_POSITION,
        XmNleftPosition, 5,
        XmNlabelString, tekst,
        XmNwidth, NAV_S,
        XmNheight, NAV_S,
        NULL);
    XtAddCallback(stop, XmNactivateCallback, videoFunksjon, STP); /*
STOP */
    XmStringFree(tekst);

    tekst = XmStringCreateSimple("||");

    pause = XtVaCreateManagedWidget("videopanel",
        xmPushButtonWidgetClass,
        vindu,
        XmNtopAttachment, XmATTACH_POSITION,
        XmNtopPosition, NAV_Y-5,
        XmNleftAttachment, XmATTACH_POSITION,
        XmNleftPosition, 10,
        XmNlabelString, tekst,
        XmNwidth, NAV_S,
        XmNheight, NAV_S,
        NULL);
    XtAddCallback(pause, XmNactivateCallback, videoFunksjon, HLT);
/* PAUSE */
    XmStringFree(tekst);

    tekst = XmStringCreateSimple(">");

    play = XtVaCreateManagedWidget("videopanel",
        xmPushButtonWidgetClass,
        vindu,
        XmNtopAttachment, XmATTACH_POSITION,
        XmNtopPosition, NAV_Y-5,
        XmNleftAttachment, XmATTACH_POSITION,
        XmNleftPosition, 15,
        XmNlabelString, tekst,
        XmNwidth, NAV_S,
        XmNheight, NAV_S,
        NULL);
    XtAddCallback(play, XmNactivateCallback, videoFunksjon, FWD); /*
PLAY */
    XmStringFree(tekst);

    /* Disable kontrollpanel for video inntil program er valgt */
    XtSetSensitive(stop, FALSE);
    XtSetSensitive(pause, FALSE);
    XtSetSensitive(play, FALSE);
}

/* Listeboks for valg av program */
Widget lagProgram (Widget vindu)
{
    XmString tekst = XmStringCreateSimple("Program:");
    void programCB();

    programRamme = XtVaCreateWidget("Ramme",
xmRowColumnWidgetClass, vindu, NULL);

    XtVaSetValues(programRamme,
        XmNtopAttachment, XmATTACH_POSITION,
        XmNtopPosition, 5,
        XmNleftAttachment, XmATTACH_POSITION,
        XmNleftPosition, 0,
        XmNbottomAttachment, XmATTACH_POSITION,
        XmNbottomPosition, 80,
        XmNrightAttachment, XmATTACH_POSITION,
        XmNrightPosition, 25,
        NULL);

    programLedetekst = XtVaCreateManagedWidget("Ledetekst",
        xmLabelWidgetClass, programRamme,
        XmNlabelString, tekst,
        NULL);
    XmStringFree(tekst);

    programListe = XmCreateScrolledList(programRamme, "Liste",
    NULL, 0);

    XtVaSetValues(programListe,
        XmNvisibleItemCount, 10,
        XmNselectionPolicy, XmSINGLE_SELECT,
        XmNhighlightThickness, 0,
        XmNscrollBarDisplayPolicy, XmSTATIC,
        NULL);

    tekst = XmStringCreateSimple(" ");
    programValgt = XtVaCreateManagedWidget("Valgt",
xmLabelWidgetClass,
        programRamme,
        XmNlabelString, tekst,
        NULL);
    XmStringFree(tekst);

    XtAddCallback(programListe, XmNdefaultActionCallback,
        programCB, programValgt);
    XtManageChild(programListe);

    XtManageChild(programRamme);

    return programRamme;
}

void ryddVisningslister()
{
    XmString tekst = XmStringCreateSimple(" ");

    /* Slett innholdet i strukturlistene */
    XmListDeleteAllItems(sekvensListe);
    XmListDeleteAllItems(sceneListe);
    XmListDeleteAllItems(innstillingListe);

    /* Slett innholdet i feltene */
    XtVaSetValues(sekvensValgt, XmNlabelString, tekst, NULL);
    XtVaSetValues(sceneValgt, XmNlabelString, tekst, NULL);
    XtVaSetValues(innstillingValgt, XmNlabelString, tekst, NULL);
    XmStringFree(tekst);
}

void nullstillStrukturpekere()
{
    startSekvensPtr = NULL;
    sisteSekvensPtr = NULL;
    startScenePtr = NULL;
    sisteScenePtr = NULL;
    startInnstillingPtr = NULL;
    sisteInnstillingPtr = NULL;
    browsePtr = NULL;
}

/* Callback for valg av program */
void programCB(Widget w, XtPointer client_data,
XmListCallbackStruct *cbs)
{
    int listepos, antall;
    programPost *tempPtr = NULL;
    int teller=0;
    XmString innslag;

    /* Funksjons-prototyper */
    void lesAnnotasjoner();
    void toemSubstruktur (strukturPost *topnode);
    void startProgram();

    /* Oppdater "valgt program"-feltet */
    XtVaSetValues(client_data,
        XmNlabelString, cbs->item,
        NULL);

    /* Husk hvilket program som er valgt */
    listepos = cbs->item_position-1; /* valgt element's posisjon i liste */
    tempPtr = startProgramPtr;
}

```

```

while (teller < listepos)
{
    tempPtr = tempPtr->neste;
    teller++;
}
valgtProgramPtr = tempPtr;

/* Toem strukturlister for evt. innhold */
if (startSekvensPtr != NULL)
    toemSubstruktur (startSekvensPtr);
nullstillStrukturpekere();

/* Les annotasjonsfil for valgt program */
lesAnnotasjoner();
browsePtr = startSekvensPtr; /* Sett browse-element til første sekvens
*/

ryddVisningslister();

valgtSekvensPtr = NULL;
valgtScenePtr = NULL;
valgtInnstillingPtr = NULL;

fyllStrukturliste(sekvensListe, startSekvensPtr);

innslag = XmStringCreateSimple (startSekvensPtr->tittel);
XtVaSetValues(sceneValgt, XmNlabelString, innslag, NULL);
startProgram();
}

/* Callback for visning av program */
void startProgram()
{
    int pid; /* prosess id */

    /* Hvis program ikke er valgt, saa returner */
    if (valgtProgramPtr == NULL)
        return;

    if ((pid = fork()) < 0)
    {
        perror("Kan ikke forke av prosess for video");
        exit(1);
    }
    if(pid == 0)
    {
        /* Kjoer videospiller */
        if (execl (fulltFilnavn(getenv("PLAYER_PATH")),
VIDEOSPILLER),
fulltFilnavn(getenv("PLAYER_PATH")), VIDEOSPILLER),
HYPHEN_F, fulltFilnavn(getenv("VIDEOFILE_PATH")),
valgtProgramPtr->videoFilnavn),
(char *)0) < 0)
        {
            perror("Kan ikke starte video");
            exit(1);
        }
    }
    printf ("STARTER\n");
    videoSpilling = TRUE;
    oppdaterNavigeringsknapper();

    /* Disable programliste og enable kontrollpanel for video */
    XtSetSensitive(programListe, FALSE);
    XtSetSensitive(stop, TRUE);
    XtSetSensitive(pause, TRUE);
    XtSetSensitive(play, TRUE);
}

/* Listeboks for valg av sekvens */
Widget lagSekvens (Widget vindu)
{
    XmString tekst = XmStringCreateSimple("Sekvens:");
    void sekvensCB();
    void visSekvensCB();

    sekvensRamme = XtVaCreateWidget("Ramme",
xmRowColumnWidgetClass, vindu, NULL);

    XtVaSetValues(sekvensRamme,
        XmNtopAttachment, XmATTACH_POSITION,
        XmNtopPosition, 5,
        XmNleftAttachment, XmATTACH_POSITION,
        XmNleftPosition, 25,
        XmNbottomAttachment, XmATTACH_POSITION,
        XmNbottomPosition, 70,
        XmNrightAttachment, XmATTACH_POSITION,
        XmNrightPosition, 50,
        NULL);

    sekvensLedetekst = XtVaCreateManagedWidget
        ("Ledetekst", xmLabelWidgetClass, sekvensRamme,
        XmNlabelString, tekst,
        NULL);
    XmStringFree(tekst);

    sekvensListe = XmCreateScrolledList(sekvensRamme, "Liste", NULL,
0);

    XtVaSetValues(sekvensListe,
        XmNvisibleItemCount, 10,
        XmNselectionPolicy, XmSINGLE_SELECT,
        XmNhighlightThickness, 0,
        XmNscrollBarDisplayPolicy, XmSTATIC,
        NULL);

    tekst = XmStringCreateSimple(" ");
    sekvensValgt = XtVaCreateManagedWidget("Valgt",
xmLabelWidgetClass,
        sekvensRamme,
        XmNlabelString, tekst,
        NULL);
    XmStringFree(tekst);

    XtAddCallback(sekvensListe, XmNdefaultActionCallback,
        sekvensCB, sekvensValgt);
    XtManageChild(sekvensListe);

    XtManageChild(sekvensListe);
    XtManageChild(sekvensRamme);

    return sekvensRamme;
}

/**/ Callback for valg av sekvens ***/
void sekvensCB(Widget w, XtPointer client_data,
XmListCallbackStruct *cbs)
{
    int listepos, antall;
    strukturPost *tempPtr = NULL;
    int teller=0;
    XmString tekst;
    void visSekvensCB();

    /* Husk hvilken sekvens som er valgt */
    listepos = cbs->item_position-1; /* valgt element's posisjon i liste */
    tempPtr = startSekvensPtr;
    while (teller < listepos)
    {
        tempPtr = tempPtr->neste;
        teller++;
    }
    valgtSekvensPtr = tempPtr;
    browsePtr = valgtSekvensPtr; /* Sett browse-element til valgt sekvens
*/
    browseVideo (browsePtr->startramme, browsePtr->sluttramme);
    oppdaterNavigeringsknapper();

    /* Oppdater "valgt sekvens"-feltet */
    XtVaSetValues(client_data,
        XmNlabelString, cbs->item,
        NULL);

    /* Slett etterfoelgende listebokser */
    XmListDeleteAllItems(sceneListe);
    XmListDeleteAllItems(innstillingListe);
    tekst = XmStringCreateSimple(" ");
    XtVaSetValues(sceneValgt, XmNlabelString, tekst, NULL);
    XtVaSetValues(innstillingValgt, XmNlabelString, tekst, NULL);
    XmStringFree (tekst);

    fyllStrukturliste(sceneListe, valgtSekvensPtr->ned);
}

void visVideoSegment(strukturPost *strukturPtr)

```

```

{
    int pid; /* prosess id */
    int resultat = -1;
    char start[6], slutt[6];

    /* Hvis program ikke er valgt, saa returner */
    if (strukturPtr == NULL)
        return;

    if ((pid = fork()) < 0)
    {
        perror("Kan ikke forke av prosess for video");
        exit(1);
    }
    if (pid == 0)
    {
        sprintf (start, "%d", strukturPtr->startramme);
        sprintf (slutt, "%d", strukturPtr->sluttramme);

        /* Kjoer videospiller */
        resultat = execl(VIDEOSPILLER, VIDEOSPILLER, HYPHEN_F,
            valgtProgramPtr->videoFilnavn,
            HYPHEN_A, start,
            (char *)0);
        if (resultat < 0)
        {
            perror("Kan ikke starte video");
            exit(1);
        }
        printf ("STARTER\n");
        videoSpilling = TRUE;
        oppdaterNavigeringsknapper();
    }

    void visSekvensCB(w, client_data, call_data)
        Widget w;
        XtPointer call_data, client_data;
    {
        visVideoSegment(valgtSekvensPtr);
    }

    /* Listeboks for valg av scene */
    Widget lagScene (Widget vindu)
    {
        XmString tekst = XmStringCreateSimple("Scene:");
        void sceneCB();
        void visSceneCB();

        sceneRamme = XtVaCreateWidget("Ramme",
            xmRowColumnWidgetClass, vindu, NULL);

        XtVaSetValues(sceneRamme,
            XmNtopAttachment, XmATTACH_POSITION,
            XmNtopPosition, 5,
            XmNleftAttachment, XmATTACH_POSITION,
            XmNleftPosition, 50,
            XmNbottomAttachment, XmATTACH_POSITION,
            XmNbottomPosition, 70,
            XmNrightAttachment, XmATTACH_POSITION,
            XmNrightPosition, 75,
            NULL);

        sceneLedetekst = XtVaCreateManagedWidget("Ledetekst",
            xmLabelWidgetClass,
                sceneRamme,
                XmNlabelString, tekst,
                NULL);
        XmStringFree(tekst);

        sceneListe = XmCreateScrolledList(sceneRamme, "Liste", NULL, 0);

        XtVaSetValues(sceneListe,
            XmNvisibleItemCount, 10,
            XmNselectionPolicy, XmSINGLE_SELECT,
            XmNhighlightThickness, 0,
            XmNscrollBarDisplayPolicy, XmSTATIC,
            NULL);

        tekst = XmStringCreateSimple(" ");
        sceneValgt = XtVaCreateManagedWidget("Valgt",
            xmLabelWidgetClass,
                sceneRamme,
                XmNlabelString, tekst,
                NULL);
        XmStringFree(tekst);

        XtAddCallback(sceneListe, XmNdefaultActionCallback,
            sceneCB, sceneValgt);
        XtManageChild(sceneListe);

        XtManageChild(sceneListe);
        XtManageChild(sceneRamme);

        return sceneRamme;
    }

    /* Callback for valg av scene */
    void sceneCB(Widget w, XtPointer client_data, XmListCallbackStruct
        *cbs)
    {
        int listepos, antall;
        strukturPost *tempPtr = NULL;
        int teller=0;
        XmString tekst;
        void visSceneCB();

        /* Husk hvilken scene som er valgt */
        listepos = cbs->item_position-1; /* valgt element's posisjon i liste */
        tempPtr = valgtSekvensPtr->ned;
        while (teller < listepos)
        {
            tempPtr = tempPtr->neste;
            teller++;
        }
        valgtScenePtr = tempPtr;
        browsePtr = valgtScenePtr; /* Sett browse-element til valgt scene */
        browseVideo (browsePtr->startramme, browsePtr->sluttramme);
        oppdaterNavigeringsknapper();

        /* Oppdater "valgt scene"-feltet */
        XtVaSetValues(client_data,
            XmNlabelString, cbs->item,
            NULL);

        /* Slett etterfoelgende listeboks */
        XmListDeleteAllItems(innstillingListe);
        tekst = XmStringCreateSimple(" ");
        XtVaSetValues(innstillingValgt, XmNlabelString, tekst, NULL);
        XmStringFree (tekst);

        fyllStrukturliste(innstillingListe, valgtScenePtr->ned);
    }

    void visSceneCB(w, client_data, call_data)
        Widget w;
        XtPointer call_data, client_data;
    {
        visVideoSegment(valgtScenePtr);
    }

    /* Listeboks for valg av innstilling */
    Widget lagInnstilling (Widget vindu)
    {
        XmString tekst = XmStringCreateSimple("Innstilling:");
        void innstillingCB();
        void visInnstillingCB();

        innstillingRamme = XtVaCreateWidget("Ramme",
            xmRowColumnWidgetClass, vindu, NULL);

        XtVaSetValues(innstillingRamme,
            XmNtopAttachment, XmATTACH_POSITION,
            XmNtopPosition, 5,
            XmNleftAttachment, XmATTACH_POSITION,
            XmNleftPosition, 75,
            XmNbottomAttachment, XmATTACH_POSITION,
            XmNbottomPosition, 70,
            XmNrightAttachment, XmATTACH_POSITION,
            XmNrightPosition, 100,
            NULL);

        innstillingLedetekst = XtVaCreateManagedWidget
            ("Ledetekst",
            xmLabelWidgetClass, innstillingRamme,

```



```

        XmNlabelString, tekst,
        NULL);
XmStringFree(tekst);

innstillingListe = XmCreateScrolledList(innstillingRamme, "Liste",
NULL, 0);

XtVaSetValues(innstillingListe,
XmNvisibleItemCount, 10,
XmNselectionPolicy, XmSINGLE_SELECT,
XmNhighlightThickness, 0,
XmNscrollBarDisplayPolicy, XmSTATIC,
NULL);

tekst = XmStringCreateSimple(" ");
innstillingValgt = XtVaCreateManagedWidget("Valgt",
xmLabelWidgetClass,
innstillingRamme,
XmNlabelString, tekst,
NULL);
XmStringFree(tekst);

XtAddCallback(innstillingListe, XmNdefaultActionCallback,
innstillingCB, innstillingValgt);
XtManageChild(innstillingListe);

XtManageChild(innstillingListe);
XtManageChild(innstillingRamme);

return innstillingRamme;
}

/* Callback for valg av innstilling */
void innstillingCB(Widget w, XtPointer client_data,
XmListCallbackStruct *cbs)
{
    int listepos, antall;
    strukturPost *tempPtr = NULL;
    int teller=0;
    XmString tekst;
    void visSceneCB();

    /* Husk hvilken innstilling som er valgt */
    listepos = cbs->item_position-1; /* valgt element's posisjon i liste */
    tempPtr = valgtScenePtr->ned;
    while (teller < listepos)
    {
        tempPtr = tempPtr->neste;
        teller++;
    }
    valgtInnstillingPtr = tempPtr;
    browsePtr = valgtInnstillingPtr; /* Sett browse-elem. til valgt
innstilling */
    browseVideo (browsePtr->startramme, browsePtr->sluttramme);
    oppdaterNavigeringsknapper();

    /* Oppdater "valgt innstilling"-feltet */
    XtVaSetValues(client_data,
XmNlabelString, cbs->item,
NULL);
}

void visInnstillingCB(w, client_data, call_data)
Widget w;
XtPointer call_data, client_data;
{
    visVideoSegment(valgtInnstillingPtr);
}

strukturPost *finnListeStart(strukturPost *liste)
{
    if (liste != NULL)
    {
        while (liste->forrige != NULL)
        {
            liste = liste->forrige;
        }
        return liste;
    }
}

int *finnElementsListePosisjon(strukturPost *element)
{
    strukturPost *temp = element;
    int posisjon = 1;

    /* Tell bakover for å finne elementets posisjon */
    if (temp != NULL)
    {
        while (temp->forrige != NULL)
        {
            temp = temp->forrige;
            posisjon++;
        }
        return posisjon;
    }
}

void skrivSubstruktur (strukturPost *topnode)
{
    strukturPost *temp;

    temp = topnode;

    while (temp != NULL)
    {
        printf ("%d - %d] %s",
temp->startramme, temp->sluttramme, temp->tittel);
        if (temp->ned != NULL)
            skrivSubstruktur(temp->ned);
        else
            printf ("*****\n");
        temp = temp->neste;
    }
}

void skrivListe (strukturPost *liste)
{
    strukturPost *temp;

    temp = liste;

    while (temp != NULL)
    {
        printf ("%s[%d - %d]\n",
temp->tittel, temp->startramme, temp->sluttramme);
        temp = temp->neste;
    }
}

void toemSubstruktur (strukturPost *topnode)
{
    strukturPost *temp;

    temp = topnode;

    while (temp != NULL)
    {
        if (temp->ned != NULL)
            toemSubstruktur(temp->ned);
        else
            free(temp);
        temp = temp->neste;
    }
}

void toemListe (strukturPost *liste)
{
    while (liste != NULL)
    {
        free (liste);
        liste = liste->neste;
    }
}

strukturPost *settInnPost (strukturPost *liste, strukturPost *over,
strukturPost *post)
{
    post->ned = NULL; /* Dette er foreloepig nederste post */
}

```

```

post->neste = NULL; /* Dette er foreloepig siste post i "strukturen"
*/

/* Forbind posten oppover... */
post->opp = over;
/* Hvis 'over' ikke allerede er forbundet, saa forbind den med 'post' */
if (over != NULL)
    if (over->ned == NULL)
        over->ned = post;

/* Hvis lista er tom */
if (liste == NULL)
{
    post->forrige = NULL;
    return post;
}

/* Hvis ny sekvens */
if (over == NULL)
{
    liste->neste = post;
    post->forrige = liste;
    return post;
}

/* Hvis ny scene eller innstilling */
if (over != NULL)
{
    /* Hvis ny struktur begynner */
    if (post->startramme == over->startramme)
        post->forrige = NULL;
    else
    {
        liste->neste = post;
        post->forrige = liste;
    }
    return post;
}

/* Les (og ignorer) en linje fra fil */
void readln(FILE *f)
{
    int tegn = ' ';

    while ((!feof(f)) && (tegn != '\n'))
        tegn = getc(f);
}

void lesAnnotasjoner()
{
    char *strukturAnnotFilnavn;
    strukturPost *lestPost;
    strukturPost *startPost = NULL;
    FILE *fil;
    int teller, antallPoster = 0;
    char dummy[MAXSTRLEN];
    char type[3];

    /* Legger til dir. og extension '.s' for struktur-annot */
    strukturAnnotFilnavn = (char *) malloc(strlen(valgtProgramPtr->
        annotFilnavn)+50);
    sprintf (strukturAnnotFilnavn, getenv("ANNOTFILE_PATH"));
    strcat (strukturAnnotFilnavn, valgtProgramPtr->annotFilnavn);
    strcat (strukturAnnotFilnavn, ".s");

    fil = fopen(strukturAnnotFilnavn, "r");
    if (fil == NULL) /* Hvis filen ikke kan leses */
    {
        printf("Kan ikke åpne %s\n", strukturAnnotFilnavn);
        return;
    }

    fscanf (fil, "%d\n", &antallPoster);

    for (teller = 0; teller < antallPoster; teller++)
    {
        lestPost = (strukturPost *) malloc(sizeof(strukturPost));
        fscanf (fil, "%s%d%d\n",
            &type, &(lestPost->startramme), &(lestPost->sluttramme));
        /* Inkrementer startramme for aa unngaa overlappende
        rammesekvenser */
        lestPost->startramme++;

        fgets (dummy, MAXSTRLEN, fil);
        lestPost->tittel = (char *) malloc(strlen(dummy));
        sprintf(lestPost->tittel, dummy);

        /* Hvis sekvens */
        if (strcmp(type, "SE") == 0)
        {
            sisteSekvensPtr = settInnPost(sisteSekvensPtr, NULL, lestPost);
            /* Hvis dette er foerste sekvens, saa husk den */
            if ((startSekvensPtr == NULL) && (sisteSekvensPtr->forrige ==
                NULL))
                startSekvensPtr = sisteSekvensPtr;
        }

        /* Hvis scene */
        if (strcmp(type, "SC") == 0)
            sisteScenePtr = settInnPost(sisteScenePtr, sisteSekvensPtr,
                lestPost);

        /* Hvis innstilling */
        if (strcmp(type, "IN") == 0)
            sisteInnstillingPtr = settInnPost(sisteInnstillingPtr,
                sisteScenePtr, lestPost);
    } /* for */

    startScenePtr = startSekvensPtr->ned;
    startInnstillingPtr = startScenePtr->ned;
} /* lesAnnotasjoner */

programPost *lesProgram()
{
    FILE *katalog = NULL; /* filpeker */
    int teller = 0, antall = 0, lengde;
    programPost *startPtr = NULL, *lestPtr = NULL, *sistLestPtr =
    NULL;
    char dummy[MAXSTRLEN];

    katalog = fopen(fulltFilnavn(getenv("ANNOTFILE_PATH"),
        KATALOGFIL, "r");
    if (katalog == NULL) /* Hvis umulig aa aapne */
        printf ("** Kan ikke åpne katalogfil **\n");
    else
    {
        readln(katalog); /* Hopp over to foerste linjer */
        readln(katalog);
        fscanf (katalog, "%d\n", &antall);

        for (teller=0; teller<antall; teller++)
        {
            lestPtr = (programPost *) malloc(sizeof(programPost));
            lestPtr->neste = NULL;

            /* Videotittel */
            fgets (dummy, MAXSTRLEN, katalog);
            lengde = strlen(dummy);
            lestPtr->tittel = (char *) malloc(lengde);
            strncpy(lestPtr->tittel, dummy, lengde-1); /* fjern '\n' */
            strcat(lestPtr->tittel, "\0");

            /* Video-filnavn */
            fgets (dummy, MAXSTRLEN, katalog);
            lengde = strlen(dummy);
            lestPtr->videoFilnavn = (char *) malloc(lengde);
            strncpy(lestPtr->videoFilnavn, dummy, lengde-1); /* fjern '\n' */
            strcat(lestPtr->videoFilnavn, "\0");

            /* Annotasjons-filnavn */
            fgets (dummy, 200, katalog);
            lengde = strlen(dummy);
            lestPtr->annotFilnavn = (char *) malloc(lengde);
            strncpy(lestPtr->annotFilnavn, dummy, lengde-1); /* fjern '\n' */
            strcat(lestPtr->annotFilnavn, "\0");

            if (startPtr == NULL) /* Hvis dette er foerste element */
                startPtr = lestPtr;
            else
                sistLestPtr->neste = lestPtr;
            sistLestPtr = lestPtr;
        } /* for */
    } /* else */
    fclose(katalog);
    return startPtr;
}

```

```

void fyllProgramliste()
{
    XmString string;
    programPost *ptr;
    int antall = 0;
    int listepos = 1;

    ptr = startProgramPtr;
    while (ptr != NULL) /* gjenta for hele lista */
    {
        string = XmStringCreateSimple (ptr->tittel);
        XmListItemUnselected(programListe,
            string, listepos); /* legger inn i programliste */
        ptr = ptr->neste;
        listepos++;
        XmStringFree (string);
    }
}

/* Event-handler */
void client_message_handler(Widget w,
    caddr_t client_data,
    XClientMessageEvent *event)
{
    Atom STOP_ATOM;
    XmString tekst = XmStringCreateSimple(" ");

    if (event->type != ClientMessage)
    {
        return;
    }

    STOP_ATOM = XInternAtom(XtDisplay(w), XtNstop, FALSE);

    /* VIDEOEN STOPPER */
    if (event->message_type == STOP_ATOM)
    {
        videoSpilling = FALSE;
        oppdaterNavigeringsknapper();
        ryddVisningslister();

        /* Nullstill program */
        XtVaSetValues(programValgt, XmNlabelString, tekst, NULL);
        valgtProgramPtr = NULL;
        nullstillStrukturpekere();

        /* Enable programliste og disable kontrollpanel for video */
        XtSetSensitive(programListe, TRUE);
        XtSetSensitive(stop, FALSE);
        XtSetSensitive(pause, FALSE);
        XtSetSensitive(play, FALSE);

        printf("VIDEO STOPPET\n");
    }
}

/* main */
void main (argc, argv)
    int argc;
    char **argv;
{
    XtAppContext applikasjon;
    Widget sekvens, scene, innstilling;
    Arg args[1];
    int n=0;

    startProgramPtr = lesProgram();

    /* Initialiser applikasjonen */
    topLevel = XtVaAppInitialize(&applikasjon,
        "Br",
        NULL, 0,
        &argc, argv,
        NULL,
        NULL);

    /* Event-handler (for events fra videospiller) */
    XtAddEventHandler(topLevel, NoEventMask, TRUE,
        (XtEventHandler) client_message_handler, NULL);

    /* Hovedvinduet */
    form = XtVaCreateWidget("Form", xmMainWindowWidgetClass,
        topLevel,
            XmNwidth, VINDU_X,
            XmNheight, VINDU_Y,
            NULL);

    vindu = XtVaCreateWidget("Vindu", xmFormWidgetClass, form,
        XmNfractionBase, 100,
        NULL);

    /* Navigeringsknapper */
    lagNavigering(vindu);

    /* Kontrollpanel for video-funksjoner */
    lagVideopanel(vindu);

    /* Programliste */
    programRamme = lagProgram(vindu);

    /* Sekvensliste */
    sekvens = lagSekvens(vindu);

    /* Sceneliste */
    scene = lagScene(vindu);

    /* Innstillingliste */
    scene = lagInnstilling(vindu);

    avsluttKnapp = XtVaCreateManagedWidget("Avslutt",
        xmPushButtonWidgetClass,
        vindu,
        XmNtopAttachment, XmATTACH_POSITION,
        XmNtopPosition, 90,
        XmNleftAttachment, XmATTACH_POSITION,
        XmNleftPosition, 8,
        XmNwidth, KNAPP_X,
        NULL);
    XtAddCallback(avsluttKnapp, XmNactivateCallback,
        avsluttCB, NULL);

    XtManageChild(vindu);
    XtManageChild(form);

    XtRealizeWidget(topLevel);
    registrerVindu(topLevel);

    /* Fyll programliste med leste programmer */
    fyllProgramliste();

    /* Disable navigeringsknapper */
    oppdaterNavigeringsknapper();

    XtAppMainLoop(applikasjon);
}

```


Vedlegg I: Referanser

- [1] Merok, P. "Datamodeller for film", Prosjektoppgave, Institutt for datateknikk og telematikk, NTH, April 1993
- [2] Merok, P. "Datamodeller for digitale film- og video-arkiv", Diplomoppgave, Institutt for datateknikk og telematikk, NTH, Desember 1993
- [3] Lerbæk, F. "Digital nyhetsfangst med Avid NewsCutter", *Audiovisuelle medier*, MultiMediaForlaget AS, Oslo, Nr. 5, August 1994, side 22-23
- [4] Smoliar, S.W. and HongJiang, Z. "Content-Based Video Indexing and Retrieval", *IEEE Multimedia*, Vol. 1, No. 2, 1994, side 62-72
- [5] Gidney, E. et. al. "CSCW for Film and TV Preproduction", *IEEE Multimedia*, Vol.1, No. 2, 1994, side 16-26
- [6] Cherry, G. et. al. "High Speed Networks and the Digital TV Studio", *Proceedings, 3rd International Workshop on Network and Operating System Support for Digital Audio and Video*, California, November 1992
- [7] NTNF, "Strukturerte intervjuer - Teknikknotat nr. 4", *Bedriftsutvikling med ny teknologi (BUNT)*, NTNF 1989
- [8] NTNF, "Aktivitets- og informasjonsanalyse - Teknikknotat nr. 21", *Bedriftsutvikling med ny teknologi (BUNT)*, NTNF 1989
- [9] NTNF, "Analyse av informasjon og eksisterende systemer - Teknikknotat nr. 24", *Bedriftsutvikling med ny teknologi (BUNT)*, NTNF 1989
- [10] Porter, M.E. and Millar, V.E. "How information gives you competitive advantage", *Harvard Business Review*, July-August 1985, side 149-160
- [11] Baggerånsås, L. et. al. "Produksjonsplanlegging og produksjonsledelse i fjernsyn", *Vett og Viten A/S* 1993, ISBN 82-412-0093-5
- [12] Millerson, G. "Effective TV Production", *Focal Press*, Second Edition, 1983, Reprinted 1989, ISBN 0-240-51209-X
- [13] Pujik, R. "Virkeligheter i NRK", *Gjøvik Trykkeri A/S*, 1990, ISBN 82-992256-0-4
- [14] Elmasri, R, Navathe S.B, "Fundamentals of Database Systems", *The Benjamin/Cummings Publishing Company, Inc.*, Second Edition, 1994, ISBN 0-8053-1748-1
- [15] Sølvsberg, A and Kung, D.C.H. "Information Systems Engineering", Institutt for datateknikk og telematikk, Foreløpig utgave, NTH, 1993

- [16] Hjelsvold, R. and Midtstraum, R. "Modelling and Querying Video Data", *Proceedings of the 20th VLDB Conference*, Santiago, Chile, 1994
- [17] Holte, T. og Sæter, A.H. "Interaktiv TV og video", Prosjektoppgave, Institutt for datateknikk og telematikk, NTH, April 1994
- [18] SDS, "SIFT Informasjons-søkesystemet", Informasjonsbrosjyre, Statens Data-sentral
- [19] SDS, "SIFT Produktinformasjon", Informasjonsbrosjyre, Statens Datasentral
- [20] SDS, "WinSIFT - Windowsbasert søkegrensesnitt", Informasjonsbrosjyre, Statens Datasentral
- [21] Gonzalez, R.C, Woods, R.E. "Digital Image Processing", *Addison-Wesley*, June 1992, ISBN 0-201-50803-6
- [22] Lerbæk, F. "Faren ved kompresjon", *Audiovisuelle medier*, MultiMedia-Forlaget AS, Oslo, Nr. 6, September 1994, side 6
- [23] Lougher, P. and Shepherd, D. "The Design of a Storage Server for Continuous Media", *Published in The Computer Journal (special issue on multimedia)*, pp 32-42, 36(1), February 1993, Lancaster University, UK, 1993. Tilgjengelig via WWW; URL = <ftp://ftp.comp.lancs.ac.uk/pub/mpg/MPG-92-27.ps>
- [24] Lougher, P. et. al. "Scalable Storage Servers for Digital Audio and Video", *Presented at the IEEE International Conference on Storage and Recording Systems 1994, University of Keele, Keele, UK, 5-7 April, 1994*, Lancaster University, UK, 1994. Tilgjengelig via WWW; URL = <ftp://ftp.comp.lancs.ac.uk/pub/mpg/MPG-94-17.ps>
- [25] Lougher, P. et. al. "The Impact of Digital Audio and Video on High Speed Storage", *Invited paper to the 13th IEEE Symposium on Mass Storage Systems, Annecy, France, 12-17 June, 1994*, Lancaster University, UK, 1994. Tilgjengelig via WWW; URL = <ftp://ftp.comp.lancs.ac.uk/pub/mpg/MPG-94-01.ps>
- [26] Danielsen, R. "Digital TV", Temaartikkel i publikasjonen *Televerkets Forskningsintitutt - Forskningsaktiviteter 1993*, Televerkets Forsknings-institutt, Kjeller, 1993
- [27] Hagen, Ø. "Lightworks blir tyngre", *Audiovisuelle medier*, MultiMediaForlaget AS, Oslo, Nr. 4, Mai 1994, side 20-21
- [28] Lerbæk, F. "Skjebnesymfonien", *Audiovisuelle medier*, MultiMediaForlaget AS, Oslo, Nr. 4, Mai 1994, side 16-17

- [29] Lerbæk, F. "Fet og Fast", *Audiovisuelle medier*, MultiMediaForlaget AS, Oslo, Nr. 4, Mai 1994, side 16-17
- [30] Hagen, Ø. "3D video via DOS", *Audiovisuelle medier*, MultiMediaForlaget AS, Oslo, Nr. 4, Mai 1994, side 18
- [31] Lerbæk, F. "Bedre multi-medieverktøy", *Audiovisuelle medier*, MultiMediaForlaget AS, Oslo, Nr. 1, Januar 1994, side 20-21
- [32] Engeset, S. "Vurdering av AVID Newscutter, Quantel Newsbox og O.L.E. Heavyworks", Teknisk notat, NRK TKAV, 27.06.94
- [33] Lerbæk, F. "Sony maner til kresenhet", *Audiovisuelle medier*, MultiMediaForlaget AS, Oslo, Nr. 3, April 1994, side 18-19
- [34] Styret for Norsk rikskringkasting, "NRKs melding 1993-1995", Styrets årsmelding, Avdeling for informasjon og samfunnskontakt, Juni 1994, ISSN 0802-1236
- [35] Hagen, Ø. "Ambisiøs AVID i sterke allianser", *Audiovisuelle medier*, MultiMediaForlaget AS, Oslo, Nr. 4, Mai 1994, side 12
- [36] Koegel, J.F. "On the Design of Multimedia Interchange Formats", *Third International Workshop on Network and Operating System Support for Digital Audio and Video*, November 12-13, 1992, La Jolla, California
- [37] Buford, J.F. and Gopal, C.B. "Standardizing a Multimedia Interchange Format: A Comparison of OMFI and MHEG", Interactive Media Group, Department of Computer Science, University of Massachusetts - Lowell. Tilgjengelig over World Wide Web; URL = <ftp://imgftp.uml.edu/pub/Papers>
- [38] Tonomura, Y. et. al. "Structured Video Computing", *IEEE Multimedia*, Vol. 1, No. 3, 1994, side 62-72
- [39] Sæter, A. "Interaktiv TV", Diplomoppgave, Institutt for datateknikk og telematikk, NTH, Desember 1994
- [40] Dybå, T., Holte, T. og Sæter, A. "LAVA: Analyse av fjernsynsproduksjon", Rapport, SINTEF DELAB, Desember 1994
- [41] Leirvåg, A. "Newstar for bilde, lyd, tekst og grafikk - PCer styrer nyheter på TV2", *Datatid*, Mai 1993, side 55-57
- [42] Ryvarden, E. og Kamsvåg, I. "Video over telenettet - Prøveprosjekt startes i år", *ComputerWorld Norge*, Nr. 9, 1994
- [43] Bekkadal, A.E. "Videoredigering som en database-anvendelse", Diplomoppgave, Institutt for datateknikk og telematikk, NTH, Desember 1994

-
- [44] Bratbergsengen, K. "Filsystemer - Lagring og behandling av store datamengder", Foreløpig utgave, NTH, August 1994
- [45] Langørgen, S. "ELVIRA - Eksperimentell videotjener for ATM", Diplomoppgave, Institutt for datateknikk og telematikk, NTH, Desember 1994
- [46] Skeide, S. "High Quality Video Server", Diplomoppgave, Institutt for datateknikk og telematikk, NTH, Desember 1993
- [47] Bekkadal, A.E. og Wasskog, T.A. "Grensesnitt for digital video", Prosjekt-oppgave, Institutt for datateknikk og telematikk, NTH, April 1994
- [48] Heller, D. "Motif Programming Manual - for OSF/Motif version 1.1", *O'Reilly & Associates, Inc.*, Vol. 6, July 1992, ISBN: 0-937175-70-6
- [49] Bratten, B. "Nektes å vise NRK-klipp", *Dagbladet*, 30.juni 1994
- [50] Christensen, G. og Østvold, R. "PACSUS - PACS for Ullevål sykehus", Diplomoppgave, Institutt for datateknikk og telematikk, NTH, Desember 1994